

**MODELING AND SIMULATING CAUSAL DEPENDENCIES ON
PROCESS-AWARE INFORMATION SYSTEMS FROM A COST PERSPECTIVE**

Bela Mutschler

Graduation committee:

Prof. dr. R. J. Wieringa, University of Twente, The Netherlands (promotor)
Dr. M. U. Reichert, University of Twente, The Netherlands (assistant promotor)
Prof. dr. J. van Hillegersberg, University of Twente, The Netherlands
Prof. dr. ir. M. Aksit, University of Twente, The Netherlands
Prof. Dr. M. Nüttgens, University of Hamburg, Germany
Prof. Dr. J. Becker, University of Münster, Germany
Dr. J. Gordijn, VU University Amsterdam, The Netherlands
Dr. F. Houdek, Daimler Group Research, Germany



SIKS Dissertation Series No. 2008-05

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN: 978-90-365-2578-7

Copyright ©2008, Bela Mutschler, Neu-Ulm, Germany

English Correction: Renate McGrath (Vermont, US)

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photography, recording, or any information storage and retrieval system, without prior written permission of the author.

**MODELING AND SIMULATING CAUSAL DEPENDENCIES ON
PROCESS-AWARE INFORMATION SYSTEMS FROM A COST PERSPECTIVE**

DISSERTATION

to obtain
the doctor's degree at the University of Twente,
on the authority of the rector magnificus,
prof. dr. W. H. M. Zijm,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday 17 January 2008 at 16.45

by

Bela Mutschler

born on January 2, 1979
in Lindau am Bodensee, Germany

This dissertation has been approved by:
Prof. dr. R. J. Wieringa (promotor)
Dr. M. U. Reichert (assistant promotor)

Preface

Without my family's support and continuous encouragement, it would have been impossible for me to get this far. In particular, I want to thank my wife *Simone* for her outstanding belief in my success and her perpetual support by keeping my back free.

I am also deeply grateful to my supervisor *Manfred Reichert* for being such a great mentor. Manfred, I really appreciated all the valuable advice you gave me in the last years. You helped me through more than one fundamental crisis and always convinced me to go on. In particular, you gave me the necessary freedom for developing all my ideas.

Special thanks also go to my promotor Roel Wieringa for his valuable suggestions, advice, and support during the last stage of writing my dissertation.

My special thanks also go to *Johannes Bumiller* for his deep interest in my thesis. Johannes, you helped me to come up with many good ideas and suggestions, to evolve them, and you gave me valuable advice whenever I needed it. The work with you forms one basis for the success of my thesis.

I also want to thank my partners at DaimlerChrysler, namely *Bärbel Hörger* for her continuous support of my work, and *Michael Offergeld* for the fruitful and pleasant cooperation and the level of responsibility he trustfully put on me in doing my research.

Finally, I also want to thank the following people for their support during the work on this thesis: Dr. Barbara Weber, Dr. Stefanie Rinderle-Ma, Novica Zarvic, Lianne Bodenstaff, Chen Li, Dominic Müller, Stefan Miller, Dr. Zlatko Zlatev, Dr. Ramin Tavakoli, Dr. Eduard Metzker, Heiko Ziegler, Dr. Axel Dold, Prof. Dr. Tobias Häberlein, Dr. Frank Houdek, Dr. Karin Schmidt, Carsten König, Michael Stupperich, Manuela Roos, Dr. Klaas Sickel, Dr. Pascal van Eck, Suse Engbers, Prof. Dr. Franz Schweiggert, Ulrich Kreher, Stanislav Pokraev, Peter Manhart, and Renate McGrath.

Abschließend möchte ich noch einige Dankesworte in deutscher Sprache anfügen, um mich auch bei denen direkt bedanken zu können, die nicht mit der englischen Sprache vertraut sind. Ohne die fortwährende Unterstützung meiner Familie – hier insbesondere meiner Frau Simone – hätte ich diese Dissertation nicht in dem mir selbst gesteckten Zeitraum beenden können. Simone war es die mir zu ihren Lasten den Rücken frei gehalten hat und die mir in mancher fundamentaler Krise immer beruhigend zur Seite gestanden hat. Gerade auch mein 3-monatiger Aufenthalt an der Universität Twente im Frühjahr 2007, den meine Familie stillschweigend und ohne zu Murren als Notwendigkeit akzeptiert hat, hat einen entscheidenden Anteil an meinem Erfolg. Danken möchte ich auch meinen Eltern und Großeltern für ihre Unterstützung während der 3 harten Jahre als Doktorand.

Abstract

Providing effective IT support for business processes has become crucial for enterprises to stay competitive in their market. Business processes must be defined, implemented, enacted, monitored, and continuously adapted to changing situations. Process life cycle support and continuous process improvement become critical success factors in contemporary and future enterprise computing.

In this context, *process-aware information systems* (PAIS) adopt a key role. Thereby, *organization-specific* and *generic* process support systems are distinguished. In the former case, the PAIS is build "from scratch" and incorporates organization-specific information about the structure and processes to be supported. In the latter case, the PAIS does not contain any information about the structure and processes of a particular organization. Instead, an organization needs to configure the PAIS by specifying processes, organizational entities, and business objects.

To enable the realization of PAIS, numerous process support paradigms, process modeling standards, and business process management tools have been introduced. The application of these approaches in PAIS engineering projects is not only influenced by technological, but also by organizational and project-specific factors. Between these factors there exist numerous causal dependencies, which, in turn, often lead to complex and unexpected effects in PAIS engineering projects. In particular, the costs of PAIS engineering projects are significantly influenced by these causal dependencies.

What is therefore needed is a comprehensive approach enabling PAIS engineers to systematically investigate these causal dependencies as well as their impact on the costs of PAIS engineering projects. Existing economic-driven IT evaluation and software cost estimation approaches, however, are unable to take into account causal dependencies and resulting effects. In response, this thesis introduces the *EcoPOST framework*. This framework utilizes evaluation models to describe the interplay of technological, organizational, and project-specific evaluation factors, and simulation concepts to unfold the dynamic behavior of PAIS engineering projects. In this context, the EcoPOST framework also supports the reuse of evaluation models based on a library of generic, predefined *evaluation patterns* and also provides governing guidelines (e.g., model design guidelines) which enhance the transfer of the EcoPOST framework into practice. Tool support is available as well.

Finally, we present the results of two online surveys, three case studies, and one controlled software experiment. Based on these empirical and experimental research activities, we are able to validate evaluation concepts underlying the EcoPOST framework and additionally demonstrate its practical applicability.

Contents

Preface	i
Abstract	iii
I Introduction and Related Work	1
1 Motivation	3
1.1 Process-aware Information Systems	4
1.2 Problem Statement	9
1.3 Contribution	12
1.4 Research Methodology	12
1.5 Outline of the Thesis	13
2 Related Work	15
2.1 Motivation	15
2.2 IT Evaluation Terminology	17
2.2.1 Costs	17
2.2.2 Benefits	18
2.2.3 Risks	19
2.3 A Framework for Comparing IT Evaluation Approaches	19
2.3.1 Existing Frameworks	19
2.3.2 Our Comparison Framework	20
2.4 Financial Viewpoint	23
2.4.1 Static Measures	24
2.4.2 Dynamic Measures	25
2.4.3 Cost-oriented Approaches	26
2.5 Process Viewpoint	29
2.5.1 Times Savings Times Salary Approach	29
2.5.2 Hedonic Wage Model	30
2.5.3 Activity-based Costing	32
2.5.4 Business Process Intelligence	33
2.6 Strategic Viewpoint	37
2.6.1 Nolan's Approach	38
2.6.2 Porter's Competitive Forces Model	39
2.6.3 Parson's Approach	40
2.6.4 Approach of McFarlan/McKinney	41
2.6.5 Real Option Theory	41

2.7	Software Cost Estimation Techniques	42
2.7.1	Algorithmic Approaches	42
2.7.2	Expertise-based Approaches	43
2.7.3	Learning-oriented Approaches	44
2.7.4	Regression-based Approaches	45
2.7.5	Composite Approaches	46
2.8	Other Approaches	47
2.8.1	Value-based Software Engineering	47
2.8.2	e3-value Framework	48
2.8.3	Value-based IT Alignment	48
2.9	Discussion	48
2.10	Summary	49
3	Requirements Analysis	53
3.1	Motivation	53
3.2	General Requirements	54
3.3	Specific Requirements	57
3.4	Summary	59
II	The EcoPOST Framework	61
4	The EcoPOST Methodology	63
4.1	Motivation	63
4.2	Basic Terminology	63
4.3	The Methodology in a Nutshell	65
4.4	Running Example	67
4.5	Understanding the Evaluation Scenario	68
4.6	Static Cost Factors	69
4.7	Dynamic Cost Factors	69
4.8	Impact Factors	71
4.8.1	Identifying Impact Factors through a Survey	72
4.8.2	Organization-specific Impact Factors	73
4.8.3	Project-specific Impact Factors	74
4.8.4	Technology-specific Impact Factors	74
4.9	Evaluation Models	79
4.9.1	Model Notation	79
4.9.2	Illustrating Example	81
4.9.3	Model Design Rules	82
4.9.4	Model Validation	86
4.10	Simulating Evaluation Models	88
4.11	Deriving Conclusions	89
4.12	Summary	89

5	Simulating Evaluation Models	91
5.1	Motivation	91
5.2	Need for Simulation	92
5.3	Specifying a Simulation Model	93
5.3.1	Constituting Elements	93
5.3.2	Specifying SCFs and static ImFs	94
5.3.3	Specifying DCFs and dynamic ImFs	94
5.3.4	Specifying Rate and Auxiliary Variables	96
5.4	Computing a Simulation Model	101
5.5	Illustrating Examples	102
5.5.1	Costs for Business Process Redesign	103
5.5.2	Costs for Maintaining Process Logic	104
5.6	Discussion	108
5.7	Summary	108
6	Evaluation Patterns	111
6.1	Motivation	111
6.2	Primary vs. Secondary Evaluation Patterns	111
6.3	Primary Evaluation Patterns	113
6.3.1	Business Process Redesign Costs	113
6.3.2	Process Modeling Costs	115
6.3.3	Requirements Definition Costs	116
6.3.4	Process Implementation Costs	117
6.3.5	Process Adaptation Costs	118
6.4	Secondary Evaluation Patterns	119
6.4.1	End User Fears	119
6.4.2	Process Knowledge	120
6.4.3	Domain Knowledge	122
6.4.4	Process Evolution	124
6.4.5	Process Complexity	125
6.4.6	Process Maturity	126
6.4.7	Work Profile Change	127
6.5	Customizing Evaluation Patterns	128
6.6	Discussion	130
6.7	Summary	131
7	Methodology Governing	133
7.1	Motivation	133
7.2	Guidelines for Designing Evaluation Models	134
7.3	Guidelines for Developing Simulation Models	136
7.4	Guidelines for Handling Dynamic Evaluation Factors	138
7.5	Guidelines for Evaluation Patterns	139
7.5.1	Identifying Evaluation Patterns	140
7.5.2	Applying Evaluation Patterns	141

7.6	Discussion	141
7.7	Summary	142
8	Tool Support	143
8.1	Motivation	143
8.2	Tool Architecture	143
8.3	The EcoPOST Cost Benefit Analyzer	145
8.3.1	Cost Evaluations	145
8.3.2	Benefit Evaluations	146
8.3.3	Model Repository	146
8.3.4	Business Ratios	148
8.4	Summary	149
III	Validation	151
9	Model Validation through Experimental Research	153
9.1	Motivation	153
9.2	Background Information	154
9.3	Experimental Framework	156
9.3.1	Basic Issues	157
9.3.2	Experiment Design	158
9.3.3	Risk Analysis	159
9.4	Performing the Experiment	161
9.4.1	Experiment Preparation	161
9.4.2	Experiment Execution	162
9.4.3	Data Analysis Procedure	162
9.4.4	Results	165
9.5	Discussion	167
9.6	Related Work	169
9.7	Summary	170
10	Case Study 1: Process Design and Work Change	173
10.1	Motivation	173
10.2	The Case Study	174
10.2.1	Research Design	174
10.2.2	Results	175
10.2.3	Methodical Soundness	178
10.3	Discussion	179
10.4	Related Work	180
10.5	Summary	180

11 Case Study 2: Using EcoPOST in Practice	183
11.1 Motivation	183
11.2 Background Information	183
11.3 The Case Study	184
11.3.1 Research Design	184
11.3.2 Analyzing Process Management Costs	187
11.3.3 Analyzing IT System Realization Costs	190
11.3.4 Analyzing Specification and Test Costs	193
11.4 Discussion	195
11.5 Summary and Lessons Learned	197
IV Discussion and Summary	199
12 Discussion	201
13 Summary and Outlook	211
Appendices	215
A Glossary	215
B Simulation Models from Chapter 6	217
Primary Evaluation Patterns	217
Secondary Evaluation Patterns	224
C Simulation Models from Chapter 11	233
Process Management Costs	233
IT System Realization Costs	236
Specification and Test Costs	238
References	253
Samenvatting	255
SIKS Dissertation Series	257

List of Figures

1.1	The Business Process Life Cycle [200].	3
1.2	Survey Background Information.	6
1.3	Degree of Process Support and IS Adaptation.	6
1.4	Process Changes.	7
1.5	Drivers of Evolution.	7
1.6	Considering Process Requirements.	8
1.7	Process-aware Information Systems.	9
1.8	The EcoPOST Research Plan.	11
1.9	Research Methodology.	13
1.10	Layout of the Thesis.	13
1.11	Answering the Research Questions: A Chapter-Oriented Overview.	14
2.1	The Curves of Technology Adoption.	15
2.2	Related Work at a Glance.	16
2.3	Different Types of Costs.	18
2.4	Our Criteria Framework at a Glance.	20
2.5	Combining Existing and New Criteria.	21
2.6	Accuracy of Estimations [29].	22
2.7	Return on Investment and Payback Period.	25
2.8	Accounting Rate of Return and Break Even Analysis.	26
2.9	Net Present Value and Internal Rate of Return.	27
2.10	Zero Base Budgeting and Cost Effectiveness Analysis.	28
2.11	Total Cost of Ownership and Target Costing.	28
2.12	Times Savings Times Salary Approach and Hedonic Wage Model.	30
2.13	1st Work Profile Matrix.	31
2.14	Linear System of Equations.	31
2.15	2nd Work Profile Matrix (Changes = Gray Cells).	31
2.16	Activity-based Costing and Business Process Intelligence.	33
2.17	Reference Architecture for Business Process Intelligence.	34
2.18	Realization of BPI Benefits.	36
2.19	Applying Nolan's Approach.	38
2.20	Nolan's Approach and Porter's Competitive Forces Model.	40
2.21	The Strategic Grid.	41
2.22	Parson's Approach and the Approach of McFarlan/McKinney.	42

2.23	Algorithmic Cost Estimation Approaches.	43
2.24	Expertise-based Cost Estimation Approaches.	44
2.25	Learning-oriented Cost Estimation Approaches.	45
2.26	Regression-oriented Cost Estimation Approaches.	46
2.27	Composite Approaches.	47
2.28	Systemic Evaluation of PAIS Engineering Projects.	50
2.29	The Big Picture.	51
3.1	Deriving Requirements.	53
3.2	Requirements for Economic-driven IT Evaluation.	54
3.3	Developing a Business Case.	54
3.4	Pressure for Economic Justification.	56
3.5	Trustworthiness of Vendor Data.	56
3.6	Difficulties when Accomplishing Economic-driven IT Evaluations.	57
3.7	Requirements for the Economic-driven Evaluation of PAIS.	58
3.8	Derivation of Requirements: Four Pillars.	59
3.9	Identified Requirements at a Glance.	60
4.1	Terminology in the EcoPOST Framework.	64
4.2	Main Steps of the EcoPOST Methodology.	66
4.3	The EcoPOST Evaluation Philosophy.	66
4.4	Understanding an Evaluation Scenario.	68
4.5	Overview of Potential Static Cost Factors [25].	69
4.6	Overview of Potential Dynamic Cost Factors.	71
4.7	Survey Background Information.	72
4.8	Organization-specific Impact Factors (Survey Results).	75
4.9	Project-specific Impact Factors (Survey Results).	76
4.10	Selected Organization- and Project-specific Impact Factors.	77
4.11	Technology-specific Impact Factors (Survey Results).	78
4.12	Selected Technology-specific Impact Factors.	79
4.13	Evaluation Model Notation and Initial Examples.	80
4.14	Dealing with the Impact of End User Fears.	82
4.15	Using Flows and Links in our Evaluation Models.	83
4.16	Examples of Incorrect Modeling.	84
4.17	Transitive Dependencies (Simplified Evaluation Models).	85
4.18	Interpreting Simulation Outcomes.	88
5.1	Step 6 of the EcoPOST Methodology.	91
5.2	Feedback in Evaluation Models – Overview of Potential Dynamic Effects.	93
5.3	Elements of a Simulation Model.	93
5.4	Constant Equations.	94
5.5	Integral Equations.	94
5.6	Integrating the Net Flow of DCFs and dynamic ImFs.	95
5.7	Integral Equations (Examples).	95

5.8	Rate Equations.	96
5.9	Auxiliary Equation.	97
5.10	Auxiliary Variables for Representing Nonlinear Dependencies.	97
5.11	Specifying Table Functions and Reference Values in <i>Vensim</i> [208].	98
5.12	Table Functions for Quantifying Impact Factors.	99
5.13	Multiplicative and Additive Equation Formulation.	100
5.14	Computing a Simulation Model.	102
5.15	Evaluation and Simulation Model.	103
5.16	Simulation Outcomes.	103
5.17	Assumed Process Adaptation Costs.	104
5.18	Evaluation Model.	105
5.19	Simulation Model.	106
5.20	Simulation Outcomes.	107
6.1	Primary Evaluation Patterns.	113
6.2	Primary Evaluation Pattern: Business Process Redesign Costs.	114
6.3	Validating the Importance of Business Process Redesign.	115
6.4	Primary Evaluation Pattern: Process Modeling Costs.	116
6.5	Primary Evaluation Pattern: Requirements Definition Costs.	117
6.6	Primary Evaluation Pattern: Process Implementation Costs.	117
6.7	Primary Evaluation Pattern: Process Adaptation Costs.	118
6.8	Secondary Evaluation Patterns.	119
6.9	Secondary Evaluation Pattern: Dealing with the Impact of End User Fears.	120
6.10	Validating the Impact of End User Fears.	121
6.11	Validating the Impact of Communication.	121
6.12	Secondary Evaluation Pattern: Process Knowledge.	122
6.13	Validating the Impact of Process Knowledge.	123
6.14	Secondary Evaluation Pattern: Domain Knowledge.	123
6.15	Validating the Impact of Domain Knowledge.	124
6.16	Primary Evaluation Pattern: Business Process Evolution.	125
6.17	Secondary Evaluation Pattern: Process Complexity.	125
6.18	Secondary Evaluation Pattern: Process Maturity (Continuous Representation).	126
6.19	Primary Evaluation Pattern: Work Profile Change.	127
6.20	Step-by-Step Customization of Evaluation Patterns.	128
6.21	Merging Evaluation Patterns.	130
6.22	Related Issues in the Context of Evaluation Patterns.	131
7.1	EcoPOST Governance Guidelines.	133
7.2	Naming Feedback Loops (Simplified Diagrams).	135
7.3	Embedded Constants.	137
8.1	EcoPOST Tool Architecture.	144
8.2	Tool Support for Evaluation Scenarios.	145
8.3	Benefit Evaluation Module (Hedonic Wage Model).	147

8.4	Model Repository.	148
8.5	Net Present Value Module and Break Even Point Module.	149
9.1	The eTravel Business Process.	154
9.2	Data-driven Case Handling.	155
9.3	Selected Criteria for Comparing Workflow Management and Case Handling.	156
9.4	Single Factor Experiment.	157
9.5	Main Groups, Teams, and Students.	158
9.6	Experiment Design, Mathematical Model, and TimeCatcher Tool.	160
9.7	Performing the Experiment.	161
9.8	Distribution of Samples (Box-Whisker-Plot Diagrams).	163
9.9	Distribution of Samples (Scatter-Plot Diagrams).	163
9.10	Paired Comparisons.	164
9.11	Experiment Results (Overall Implementation Efforts).	165
9.12	Experiment Results (Base and Change Implementation).	166
9.13	Understanding Effort Distribution.	167
9.14	Selected Questionnaire Results.	168
9.15	Validation Effort Distribution for Process Implementation Costs.	169
9.16	Understanding the Distribution of Efforts.	170
10.1	Validating the Impact of Work Profile Change.	174
10.2	General Usefulness (Site 1).	176
10.3	General Usefulness (Site 2).	177
10.4	Impact on Job Dimensions (Site 1).	177
10.5	Impact on Job Dimensions (Site 2).	178
11.1	Automotive E/E System Development [122].	184
11.2	The Business Case.	185
11.3	Data Collection (Information Sources).	186
11.4	Identification of relevant Impact Factors.	187
11.5	Process Management Costs (Evaluation Model).	189
11.6	Process Management Costs (Simulation Results).	191
11.7	IT System Realization Costs (Evaluation Model and Simulation Results).	192
11.8	Specification and Test Costs (Evaluation Model and Simulation Results).	194
11.9	Efforts Caused by the new Process Designs.	196
11.10	Efficiency of the new Process Designs.	197
12.1	Requirements at a Glance.	203
12.2	Experimental and Empirical Research in the EcoPOST Framework.	207
12.3	Software Technologies used Today and Tomorrow.	209
12.4	Evaluation of the EcoPOST Framework.	210

List of Tables

2.1	Business Process Intelligence Tools.	35
6.1	Merge Algorithm for Evaluation Models.	129
7.1	Guidelines for Designing Evaluation Models.	134
7.2	Guidelines for Developing Simulation Models.	136
7.3	Guidelines for Handling Dynamic Evaluation Factors.	138
7.4	Guidelines for Identifying and Applying Evaluation Patterns.	140

Part I

Introduction and Related Work

Chapter 1

Motivation

Providing effective IT support for business processes has become crucial for enterprises to stay competitive in their market [8, 43]. In the automotive domain, for example, a broad spectrum of business processes ranging from simple administrative procedures to complex, knowledge-intense engineering processes has to be effectively supported [101, 122]. Similar requirements exist in many other domains like finance [70], transportation [18], or healthcare [106]. All processes must be defined, implemented, enacted, monitored, and continuously adapted to changing situations. Thus, *process life cycle support* and continuous process improvement adopt a key role in contemporary and future enterprise computing.

The process life cycle [200, 213] begins with the *design phase*, i.e., with the conceptual (re)design of a business process (cf. Fig. 1.1). Tools for modeling, analyzing, and mining are used in this context. Based on the chosen process design, the business process is implemented. As a result of this *implementation phase* we obtain a process-oriented *information system* (IS) which provides business functions for facilitating the process. As a characteristic example of such an IS consider a *product data management* (PDM) system, which usually offers a broad range of business functions for the integrated support of engineering processes and product life cycle management [127]. In the automotive domain, such PDM systems are used to deploy models and documentation of vehicles (the managed product) to involved user groups (e.g., engineers, managers, suppliers).

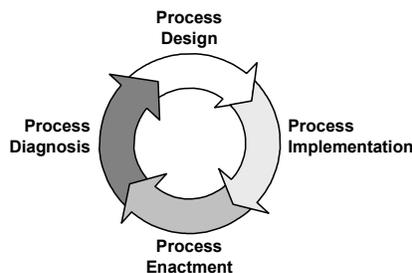


Figure 1.1: The Business Process Life Cycle [200].

During the following *enactment phase*, multiple instances of the implemented process are created and executed. Process enactment, in turn, results in a multitude of valuable real-

time process data that can be analyzed and evaluated in the *diagnosis phase*. Based on this, optimized business processes can be derived.

IS supporting the entire process life cycle are often referred to as *process-aware information systems* (PAIS). In general, PAIS orchestrate processes of a particular type (e.g., handling of a customer order) based on a predefined process model. Such a model defines the *tasks* to be executed (i.e., activities), their *dependencies* (e.g., control and data flow), the *organizational entities* performing these tasks (i.e., actors), and the *business objects* which provide or store activity data. Unlike conventional IS, PAIS strictly separate process logic from application code [160]. This provides promising perspectives, e.g., regarding the effective adaptation of process changes.

For implementing PAIS, numerous process support paradigms (e.g., workflow management [202], service flow management [107, 199], case handling [204]), process modeling standards [59] (e.g., EPC, WS-BPEL, BPML), and tools (e.g., ARIS Toolset, Staffware, Websphere) have been introduced. More precisely, build-time components support the (graphical) modeling of "as is" and "to be" processes as well as comprehensive process analysis (e.g., based on simulations). Run-time components, in turn, support the execution of processes and the analysis of process performance based on logged execution data [203].

Projects dealing with the introduction of PAIS are usually associated with high costs. These costs are not only influenced by technological factors [202], but also by organizational [102, 172, 173] and project-specific [15, 113, 210] ones. Hidden dependencies and interactions between the different factors result in additional, complex effects making cost evaluations for PAIS engineering projects a challenging task to accomplish. In particular, existing economic-driven IT evaluation approaches and cost estimation techniques are unable to capture such effects as they mainly rely on static evaluation models.

What is needed is a comprehensive approach enabling PAIS engineers to systematically analyze the complex causal dependencies between organizational, project-specific, and technological factors arising in PAIS engineering projects as well as their effects on project costs. In response to this need, this thesis introduces the **EcoPOST framework**.

1.1 Process-aware Information Systems

We will consider IT support for business process as being *effective*, if the following two goals are achieved: (1) the cost-effective implementation and customization of processes, and (2) the availability of a technical infrastructure supporting all phases of the process life cycle (cf. Fig. 1.1). In practice, however, many IS fail to provide effective business process support [52, 206, 225]. In order to better understand the rationales for this situation, we have conducted an *exploratory case study* in the automotive domain. Results shall serve as further motivation for this thesis.

Exploratory Case Study. Over a period of three months we analyzed two characteristic automotive processes (a release management process and a data provision process) as well as their IT support (e.g., by a PDM system with more than 5000 users). We conducted 26 interviews with software developers, domain experts, and end users. Interviews were based

on a predefined, semi-structured protocol comprising two parts. The first one addressed the investigated process, whereas the second part dealt with specific problems of the supporting IS. Besides, we analyzed existing process documentation and organizational handbooks.

Altogether, we collected more than 120 shortcomings related to the development and operational use of the investigated IS. These included both organizational and technological aspects. Due to lack of space, we cannot present all of these 120 shortcomings in detail. Instead, we summarize major results along five main *problem areas* [137]:

- **Problem Area 1: *Process Evolution*.** According to our case study, many problems are related to the evolution of business processes and their variability. In the analyzed domain, frequent process changes require the continuous adaptation of the supporting IS. However, realizing such adaptations is a difficult task to accomplish (cf. Problem Area 2 and Problem Area 3).
- **Problem Area 2: *Hard-coded Process Logic*.** The analyzed IS exhibit a "hard-coded" process logic, i.e., process logic is hidden in the application code and is not separately managed, e.g., by a *workflow management system* (WfMS). Each time a business process changes, deep inspections and customizations of source code modules become necessary. This, in turn, results in large efforts and inefficient adaptation of IS to process changes.
- **Problem Area 3: *Complex Software Customizing*.** The analyzed IS are realized based on standard software components. Insufficient customization features of these components also result in an ineffective adaptation of process changes. In particular, existing software components lack possibilities to customize process logic at a sufficiently flexible and detailed level. This complicates the alignment of process-oriented IS to organization-specific requirements.

Besides these fundamental problem areas, we have identified two additional ones which are not directly related to the development of IS, but which can be linked to the elicitation and analysis of requirements prior to IS implementation:

- **Problem Area 4: *Inadequate Business Functions*.** Our case study reveals that provided business functions do not effectively support business processes. Many of the implemented business functions are never used and are therefore without any "value". Other business functions provide more functionality than actually needed. Also, business functions which are actually needed for process support are missing, making the automation of certain process activities impossible.
- **Problem Area 5: *Missing Process Information*.** Some of the analyzed IS log event-based process execution data or status information (e.g., related to the start and completion of process activities). However, the structure of log data differs from system to system. Hence, keeping track of the processes or mining them generates large efforts (e.g., for normalize available data). In any case, missing process information makes it difficult to identify possible business process optimizations, process cycle times are longer than needed, and resources are not allocated in a cost-effective way.

In summary, our exploratory case study has provided initial insight into many practical problems related to the development and maintenance of process-oriented IS.

In order to investigate the most relevant results from our domain-specific case study in detail, we performed an additional online survey [94, 95, 96].

Online Survey. The survey does not only involve IT professionals from the automotive domain, but from other organizations and domains as well (cf. Fig. 1.2). 79 IT professionals (equating to a response rate¹ of 20.2%) have participated. See [136, 137] for details about the survey.

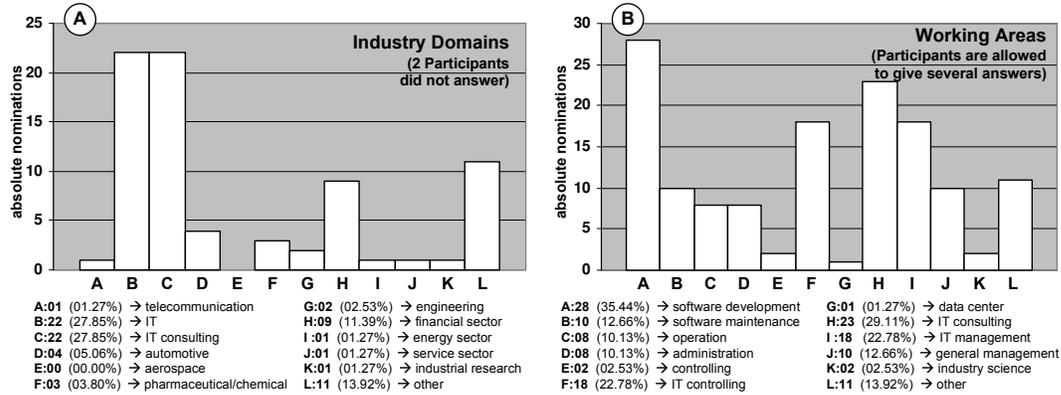


Figure 1.2: Survey Background Information.

We first asked the survey participants whether the current degree of process-orientation is sufficient. 25.32% of the participants state that IS *only partly* provide a sufficient degree of process-orientation (cf. Fig. 1.3A).

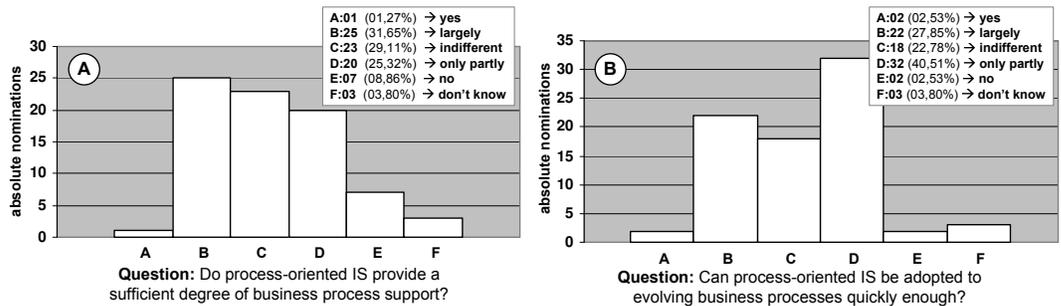


Figure 1.3: Degree of Process Support and IS Adaptation.

8.86% even state that current IS do not provide a sufficient degree of process orientation at

¹Mehta and Sivadas [115] describe that response rates for electronic surveys range from 40% to 64%. Bachmann et. al [10] identify response rates of 19% for email and 46% for mail surveys. Falconer and Hodgett [62] note that reasonable response rates for IS research are likely to be in the range of 10% to 35%. Thus, given the low response rates to IS and email surveys in general, and the large number of 29 questions, we regard the response rate to our survey as acceptable.

all. 29.11% of the participants consider the realized process support neither as problematic nor as advantageous. Only 32.92% of the participants consider available business process support as (largely or fully) sufficient.

One of the problem areas identified in our case study concerns process evolution (cf. Problem Area 1). Survey results confirm that the need to continuously adapt IS to evolving processes constitutes a problem in practice. 43.04% of the respondents (cf. Fig. 1.3B) answer the question whether their current IS can be adopted to evolving business processes (and therefore to evolving requirements) quickly enough with *no* (2.53%) or *only partly* (40.51%). Only 30.38% answer this question with *yes* (2.53%) or *largely* (27.85%).

More than 90% of the participants agree that business processes change *very often*, *often* or *sometimes* in their organization (cf. Fig. 1.4B). Additionally, 68.35% believe that the frequency of business process changes will increase in future (cf. Fig. 1.4B).

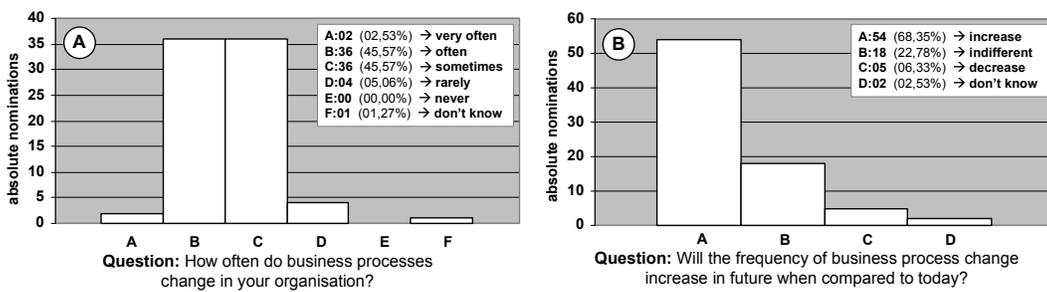


Figure 1.4: Process Changes.

We further analyzed drivers for process evolution. Participants state that the need for *process optimization* (65.82%) is the most important driver in this context (cf. Fig. 1.5). Others are *organizational engineering* (49.37%), *laws and policies* (46.84%), i.e., compliance issues, and *market developments and dynamics* (49.37%).

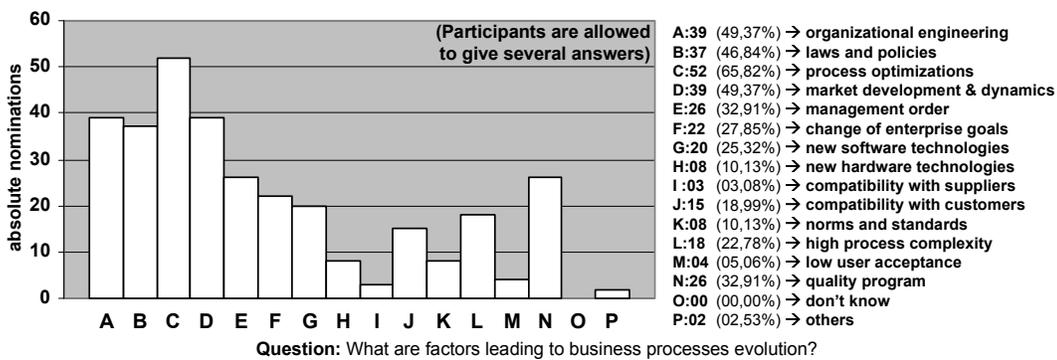


Figure 1.5: Drivers of Evolution.

Besides, we also investigate the problem of inadequate business function support in more detail (cf. Problem Area 4). 45.57% of the respondents share the opinion that business process requirements (specifying which business functions are to be implemented) must be

considered when developing an IS (cf. Fig. 1.6A). 41.77% state that respective requirements should be considered if possible. More precisely, 87.34% of the participants expect business process requirements to be considered when implementing an IS. However, and this is important, only 62.02% of the participants acknowledge that respective requirements are indeed (*yes* and *largely*) considered when developing IS (cf. Fig. 1.6B).

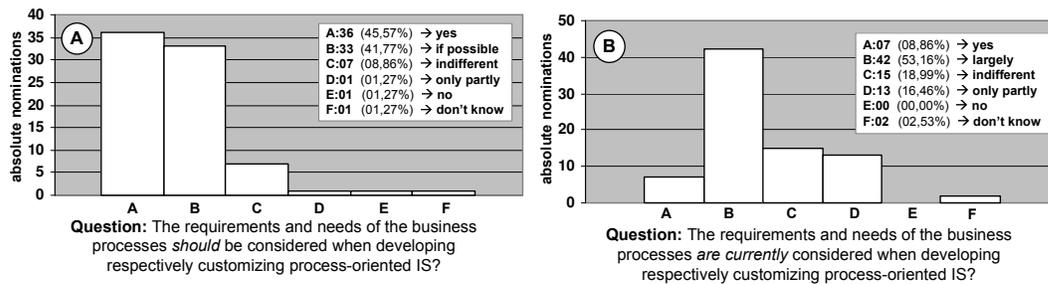


Figure 1.6: Considering Process Requirements.

Discussion. The results of both our case study and our online survey show that current IS are unable to provide business process support as needed in practice. In our case study, we have identified five major reasons for this drawback: (i) continuous evolution of business processes, (ii) hard-coded process logic of the supporting IS, (iii) complex software customization, (iv) inadequate business functions, and (v) missing process information. Our survey confirms these problems. Moreover, it provides further insights, e.g., into the drivers of process evolution or the compliance of IS with process requirements. Generally, these results show that the currently realized degree of "process-orientation" is scarce. Hence, enterprises demand technologies which enable them to implement effective business process support. PAIS are such a technology.

Process-aware Information Systems. Our empirical studies indicate that providing effective business process support by IS is a difficult task to accomplish. The currently realized degree of "process-orientation" in IS is by far not satisfactory. By contrast, enterprises more and more crave for approaches that enable them to improve their business process performance. Reflecting the aforementioned results, we conclude that conventional process-oriented IS are scarce. What we need instead are *process-aware IS* (PAIS), i.e., IS that support all phases of the business process life cycle.

PAIS can be implemented in two ways [59]: (1) by developing an organization-specific process support system, or (2) by configuring a generic process support system. In the former case, the PAIS is build "from scratch" and incorporates organization-specific information about the structure and processes to be supported. As an example consider an *enterprise resource planning* (ERP) system. In the latter case, the PAIS does not contain any information about the structure and processes of a particular organization. Instead, an organization needs to configure the PAIS by specifying processes, organizational entities, and business objects. As an example consider the configuration of a WfMS.

In any case, PAIS strictly separate process logic (comprising the activities to be executed) from application code [55], i.e., PAIS are driven by process models rather than program code (cf. Fig. 1.7). They are realized based on process engines which orchestrate processes at run-time [160]. These process engines also provide an extensive library of process-oriented functions at build-time, e.g., for accomplishing automatic process analysis. Empirical studies [99, 100] confirm that PAIS enable the fast and cost-effective implementation and customization of new and of existing processes (cf. Problem Areas 2 + 3).

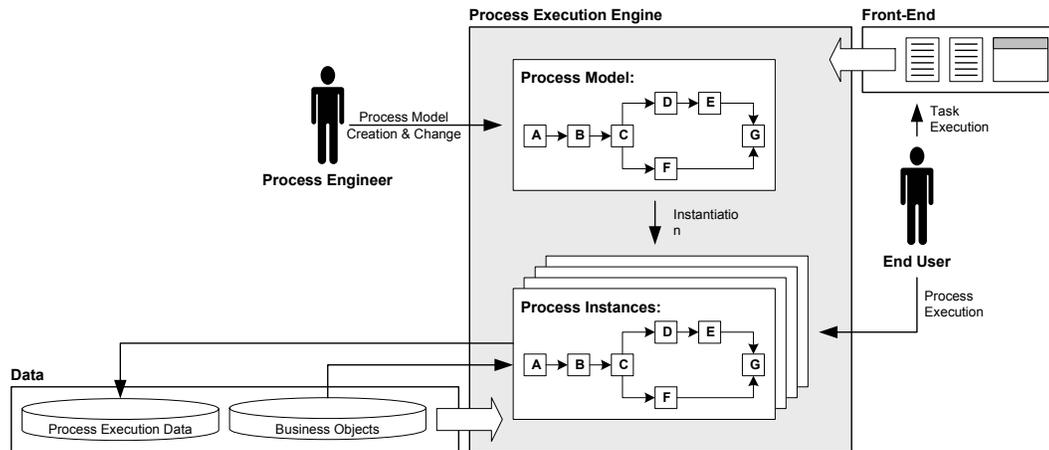


Figure 1.7: Process-aware Information Systems.

Realizing PAIS also implies a significant shift in the field of IS engineering [133]. Traditional IS engineering methods and paradigms (e.g., procedural programming) have to be supplemented with engineering principles particularly enhancing the operational support of business processes. This is crucial to tie up those requirements that have been neglected by current process-oriented IS so far. However, such a shift is difficult to accomplish as software projects often use software technologies – at least today – that do not support the needed degree of process-orientation.

1.2 Problem Statement

Realizing PAIS is a complex task and PAIS engineering projects are usually associated with high costs [211, 222]. When evaluating these costs, a lot of PAIS-specific cost factors have to be considered which do only partly emerge in projects dealing with function- and data-centered IS. As examples consider costs for conducting interview-based process analysis (in order to elicit process requirements), costs for redesigning business processes and defining optimized processes, costs for creating process models and documentation, or costs for implementing the PAIS as well as process changes based on process management technology.

In order to justify these costs, it is often referred to assumed benefits of PAIS installations, e.g., to improved process performance [45, 165], cheaper process implementation [99, 100], and gained process flexibility [66, 168].

However, though economic-driven IT evaluation and software cost estimation have received considerable attention during the last decades (cf. Chapter 2) – and have become an essential task in IS engineering – it is difficult to apply existing evaluation techniques to PAIS engineering projects. This difficulty particularly stems from the inability of existing techniques to take into account the numerous technology-, organization-, and project-specific factors and their dependencies and interactions, which specifically arise in the context of PAIS engineering projects.

For example, activities for *business process redesign* – as often conducted prior to the introduction of a PAIS – may be influenced by an intangible factor "Willingness of Staff Members to support Redesign Activities". Obviously, if staff members do not contribute to a redesign project by providing needed information (e.g., about process details), any redesign effort will be ineffective and will increase costs. If staff willingness is additionally varying during the redesign activity (e.g., due to a changing communication policy), business process redesign activities will be subject to even more complex effects.

What PAIS engineers need is a comprehensive approach that enables them to investigate such dependencies and their impact on the costs of PAIS engineering projects. Developing such an approach, a number of challenges have to be taken into account. *First*, we need to identify those factors that influence the costs of PAIS engineering projects. *Second*, we have to identify causal dependencies between these factors. This may be a difficult task to accomplish as these may not be transparent. *Third*, it is necessary to investigate the effects originating from these causal dependencies. *Fourth*, we need to analyze in which way this interplay is influencing the overall costs of PAIS engineering projects. *Finally*, we have to integrate all these issues into a comprehensive approach for deriving qualitative and quantitative conclusions.

Picking up these challenges, we define a number of research questions which guide the research described in this thesis (cf. Fig. 1.8 for a more detailed research plan). Thereby, we distinguish between *knowledge problems* (KP) and *design problems* (DP) [217]²:

- **Research Question 1 (KP):** What are existing approaches that can be used to evaluate PAIS? What are criteria that can be used to compare these approaches? Are existing approaches really suitable to evaluate PAIS engineering projects?
- **Research Question 2 (KP):** Which technological, organizational, and project-specific factors determine the complex cost structure of PAIS engineering projects?
- **Research Question 3 (KP):** Which relationships and causal dependencies exist between these technological, organizational, and project-specific factors? How can the interplay of these technological, organizational, and project-specific factors be analyzed? How can it be described?

²A *knowledge problem* is a difference between what we know about the world and what we would like to know [216]. Knowledge problems can be solved by asking others, by searching the literature, or by doing research. Knowledge problems have stakeholders, namely the people who would like to acquire the desired knowledge. Research problems typically are knowledge problems in which we search for true propositions. *Design problems*, in turn, are engineering problems, in which we search for an improvement of the world with respect to some goals. The evaluation criteria for answers to both kinds of problems are quite different: *truth* in the case of research problems, *goal achievement* in the case of design problems.

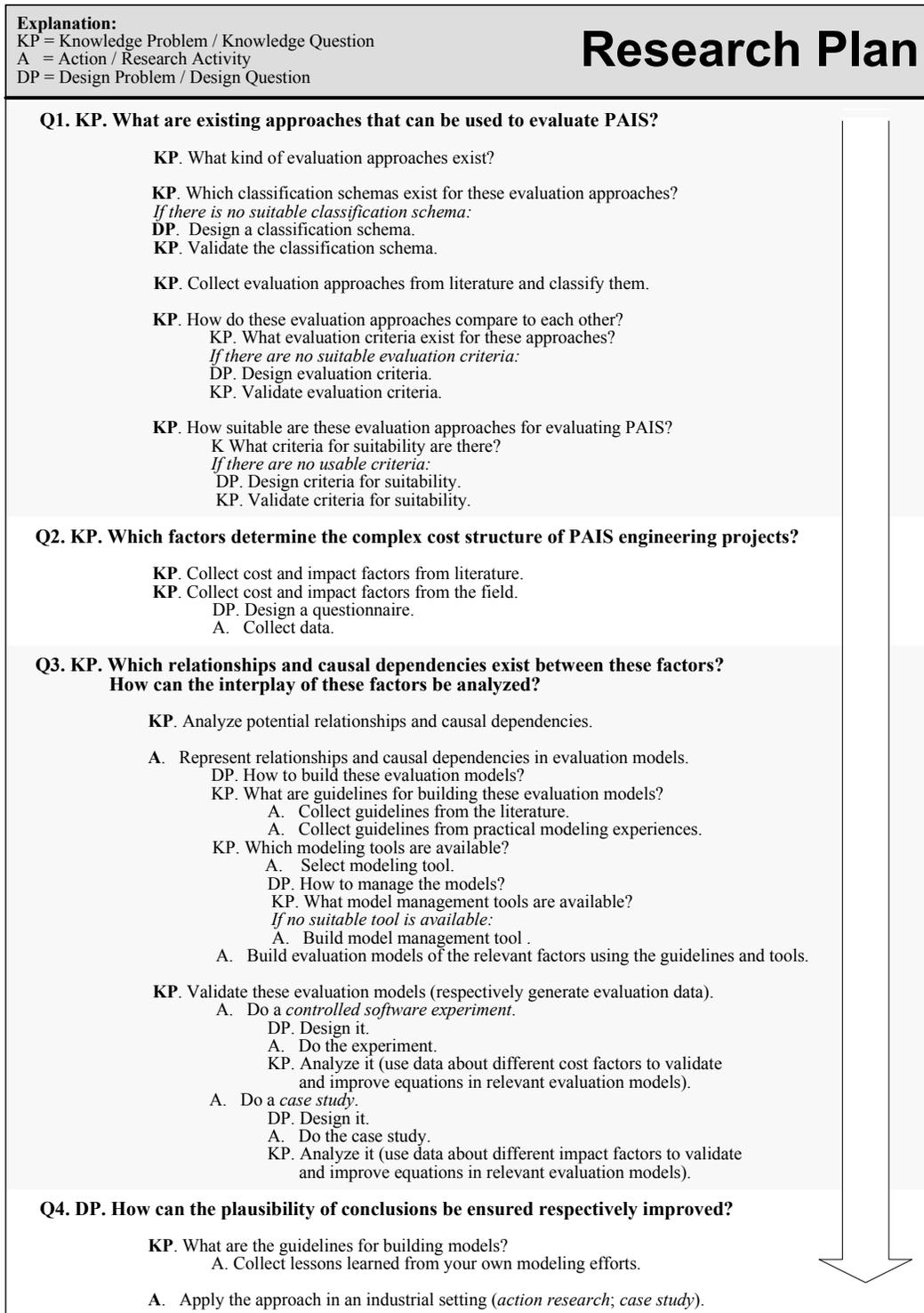


Figure 1.8: The EcoPOST Research Plan.

- **Research Question 4 (DP):** How can qualitative and quantitative conclusions regarding the costs of PAIS engineering projects be derived? How can the plausibility of conclusions be ensured respectively improved?

These research questions guide the research presented in this thesis. Note that Section 1.5 discusses which research questions are addressed in which of the following chapters.

1.3 Contribution

The contributions of this thesis are as follows:

- We identify a set of technology-, organization-, and project-specific impact factors which influence the costs of PAIS engineering projects.
- We introduce a comprehensive approach (called EcoPOST framework) for systematically investigating causal dependencies and resulting cost effects in PAIS engineering projects. Our approach utilizes evaluation models and simulation and also supports the reuse of evaluation models.
- We describe a basic set of generic and customizable evaluation models called *evaluation patterns*. Each evaluation pattern specifies one cost or impact factor and related causal dependencies in detail.
- Based on our experiences in applying the EcoPOST framework, we describe governing guidelines enhancing the transfer of the EcoPOST framework into practice (e.g., model design guidelines).
- We present the *EcoPOST Cost Benefit Analyzer*, a tool which supports various administrative tasks related to the application of the EcoPOST framework. Among other things, this tool includes a model repository enabling the reuse of evaluation and simulation models.
- We illustrate how our evaluation models can be validated based on empirical and experimental research activities. More precisely, we describe the results of two online surveys, three case studies, and one controlled software experiment.

1.4 Research Methodology

Our research comprises four phases: (1) *problem analysis*, (2) *requirements analysis*, (3) *solution design*, and (4) *solution validation and implementation evaluation* (cf. Fig. 1.9).

We start with an analysis of the problem to be investigated (*Phase 1*). Related activities are an exploratory case study, an online survey, and a literature survey on IT evaluation approaches and software cost estimation techniques. Note that results of the first two activities have been already discussed (cf. Section 1.1).

Based on these activities as well as based on practical experiences gathered in the automotive domain, we derive requirements for evaluating PAIS engineering projects from an economic-driven perspective (*Phase 2*). Taking identified requirements, we develop our solution approach, i.e., the EcoPOST framework (*Phase 3*).

Finally, we validate our framework based on a second online survey, two case studies, and a controlled software experiment (*Phase 4*).

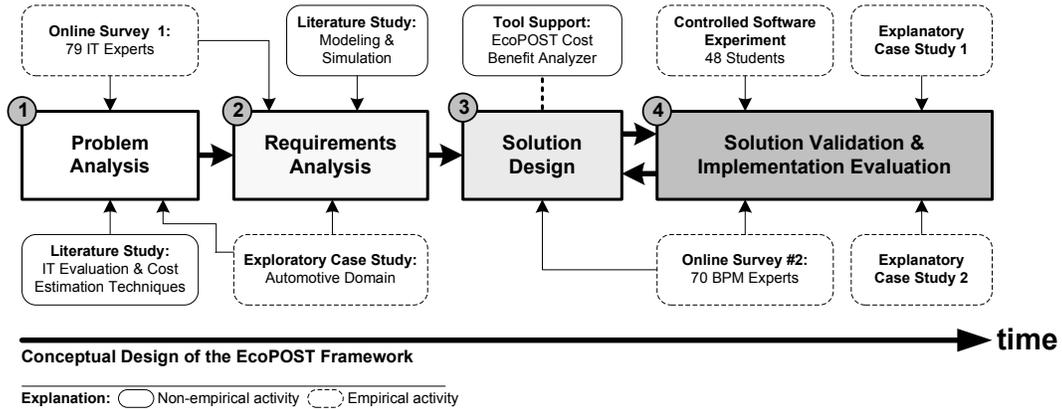


Figure 1.9: Research Methodology.

1.5 Outline of the Thesis

The remainder of this thesis is organized as follows. *Part I* unifies introductory chapters. More specifically, Chapter 2 discusses related work (and therewith addresses the first of the identified research questions, cf. Section 1.2). Chapter 3 summarizes requirements for evaluating PAIS engineering projects from an economic-driven viewpoint.

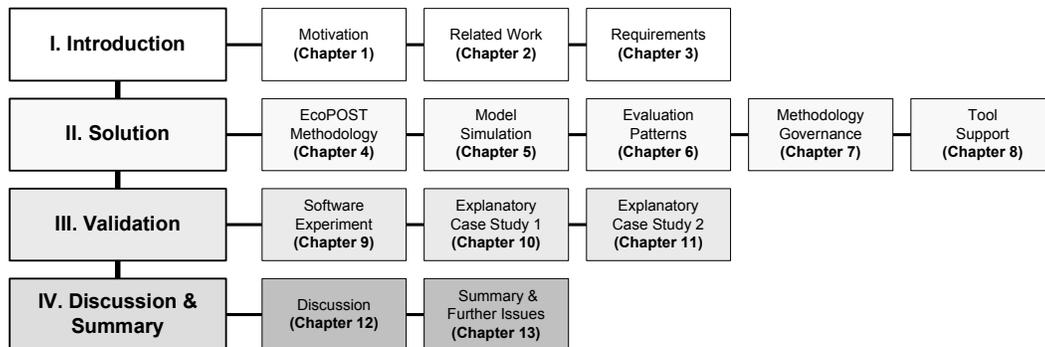


Figure 1.10: Layout of the Thesis.

Part II of this thesis introduces the EcoPOST framework. Chapter 4 introduces the EcoPOST methodology. Chapter 5 deals with the simulation of our evaluation models. Picking

up the demand to reuse evaluation models, Chapter 6 introduces the notion of *evaluation patterns*. Chapter 7 deals with the governing of the EcoPOST framework. Chapter 8 describes available tool support for the EcoPOST framework.

Part III of this thesis summarizes research activities which have been conducted to validate the EcoPOST framework. Chapter 9 illustrates – along a software experiment comparing workflow and case handling technology – how experimental research can be applied for validating evaluation and simulation models. Chapter 10 also deals with model validation, this time based on a case study investigating the correlation of PAIS and work profile change. Chapter 11 summarizes the results from a case study, in which we use the EcoPOST framework to investigate cost overruns in a large PAIS engineering project in the automotive domain.

Finally, *Part IV* discusses (Chapter 12) and summarizes (Chapter 13) the main contributions of the thesis. Chapter 13 also gives an outlook on future work. Fig. 1.11 illustrates which of the defined research question is addressed by which chapter.

		I		II				III		IV				
Chapters		Chapter 1*	Chapter 2	Chapter 3*	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Chapter 9	Chapter 10	Chapter 11	Chapter 12*	Chapter 13*
Research Questions	Research Question 1		X											
	Research Question 2				X					X	X	X		
	Research Question 3				X	X	X			X	X	X		
	Research Question 4				X	X	X	X	X	X	X	X		

* not relevant with respect to the defined research questions

Figure 1.11: Answering the Research Questions: A Chapter-Oriented Overview.

Chapter 2

Related Work

2.1 Motivation

Providing effective IT support has become crucial for enterprises to stay competitive in their market [8]. However, it remains a complex task for them to select the "right" IT investment at the "right" time, i.e., to select the best possible IT solution for a given context [36, 51].

Generally, the adoption of *information technology* (IT) can be described by means of an *S curve* (cf. Fig. 2.1A) [39, 40, 186]. When new IT emerges, it is unproven, expensive, and difficult to use. Standards have not been established, and best practices still have to emerge. At this point, only "first movers" start using new IT. They expect that the high costs and risks for being an innovator will be later compensated by gaining competitive advantage.

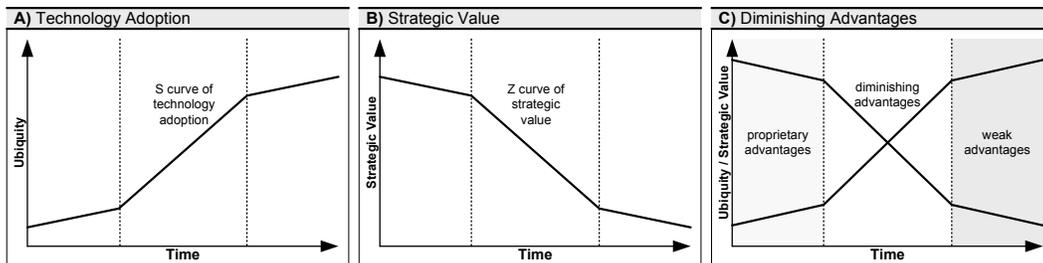


Figure 2.1: The Curves of Technology Adoption.

Picking up an emerging IT later, by contrast, allows to wait until it becomes more mature and standardized, resulting in lower introduction costs and risks. However, once the value of IT has become clear, both vendors and users rush to invest in it. Consequently, technical standards emerge and license costs decrease. Soon the IT is widely spread, with only few enterprises having not made respective investment decisions.

Factors that typically push a new IT up the S curve include standardization, price deflation, best practice diffusion, and consolidation of the vendor base. All these factors erode the ability of IT as a mean for differentiation and competitive advantage. In fact, when dissemination of IT increases, its strategic potential shrinks at the same time. Finally, once

the IT has become part of the general infrastructure, it is typically difficult to achieve further strategic benefits (though rapid technological innovation often continues). This can be illustrated by a *Z curve* (cf. Fig. 2.1B).

Considering the different curves of IT adoption, decisions about IT investments and the appropriate moment of their introduction constitute a difficult task [33, 67] (cf. Fig. 2.1C). Respective decisions are influenced by numerous parameters which are typically summarized in a business case [116, 161]. Examples for respective parameters are costs of an investment, assumed profit, its impact on work performance, business process performance, and the achievement of enterprise objectives. To cope with different evaluation goals, many evaluation approaches have been introduced in the last decades. This chapter gives an overview of existing methods (cf. Fig. 2.2) and discusses their ability to deal with the complex economics of IT investments in general and the economics of PAIS in particular.

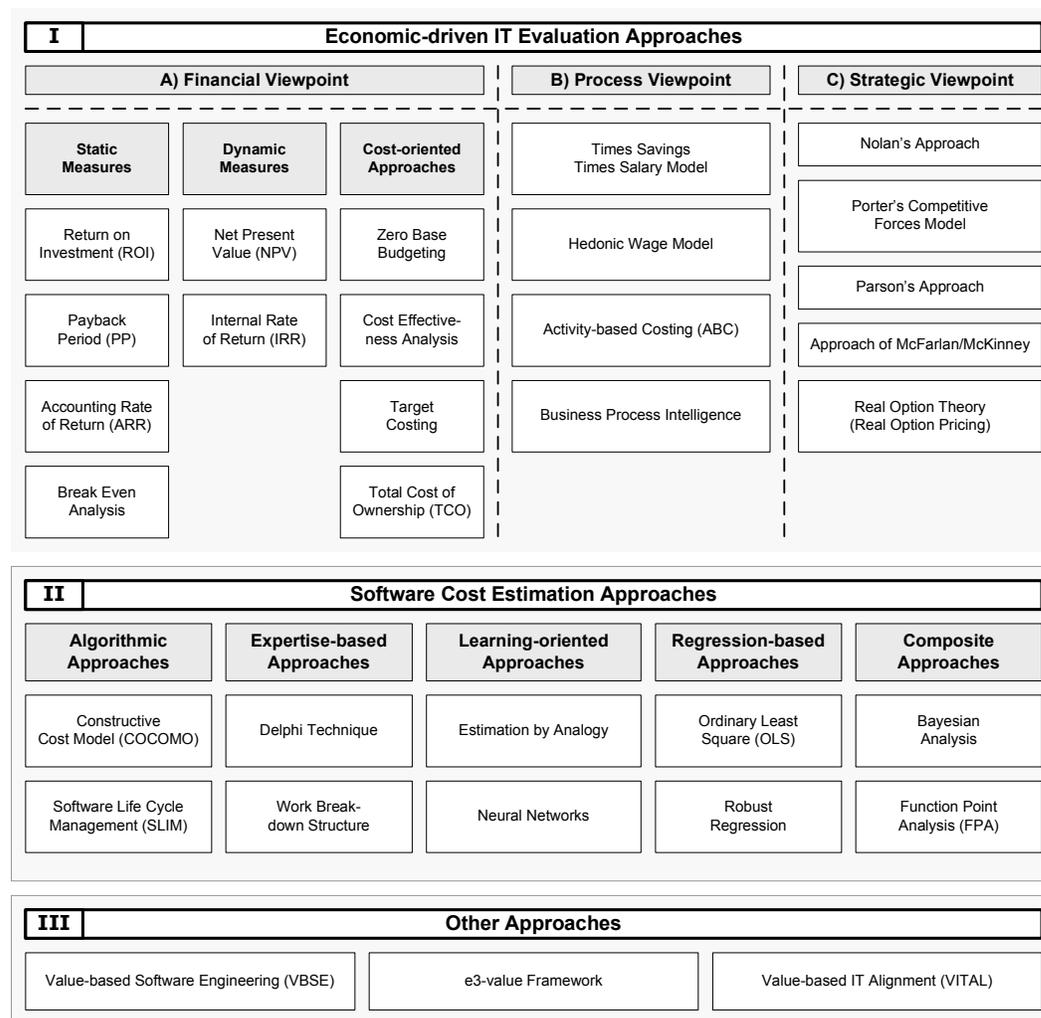


Figure 2.2: Related Work at a Glance.

The remainder of this chapter is organized as follows. Section 2.2 introduces basic terminology related to IT evaluation. Section 2.3 introduces the framework used in this thesis for discussing existing IT evaluation approaches. Section 2.4 deals with methods for conducting evaluations from a financial viewpoint. Section 2.5 discusses approaches for evaluating the impact of IT on process and work performance. Section 2.6 summarizes approaches for analyzing IT from a strategic viewpoint. Section 2.7 discusses software cost estimation techniques. Section 2.8 deals with other approaches. Section 2.9 discusses whether the presented approaches are suitable to evaluate PAIS engineering projects from an economic-driven viewpoint. Finally, Section 2.10 concludes with a summary.

2.2 IT Evaluation Terminology

Economic-driven IT evaluation typically focuses on the systematic analysis of *costs*, *benefits*, and *risks* [135] (though other aspects can be addressed as well). These three terms are characterized in the following.

2.2.1 Costs

Generally, *costs* can be defined as the total expenses for goods or services including money, time and labor. Literature distinguishes between different *cost types* [2]:

- **Historical Costs:** Describe the original monetary value of an economic item, i.e., the total amount of money spent for an investment at purchase or payment time.
- **Acquisition Costs:** Refer to the costs for purchasing an asset (e.g., IT system) after adjustments for incentives, discounts, or closing costs, but before any sales tax.
- **Opportunity Costs:** Denote the difference between the yield an investment earns and the yield which would have been earned if the costs for the investment had been placed into an alternative investment generating the highest yield available.
- **Internal and External Costs:** External costs occur outside an organization and can be controlled by contracts and budgets. Internal costs, by contrast, occur within an organization (e.g., related to a specific project).
- **Direct and Indirect Costs:** Direct costs can be associated to the production of a particular product or service or can be allocated to a particular cost center. Raw materials and the wages of those working on production lines are good examples. By contrast, indirect costs cannot be budgeted, i.e., they cannot be represented by explicit cost factors. Indirect costs include depreciation (where it is calculated related to output – e.g. machine hours), maintenance and certain labor costs.
- **Fixed and Variable Costs:** Fixed costs are not directly related to the level of production or output. In other words, even if the business has a zero output or high output, the level of fixed costs will remain broadly the same. As examples of fixed costs

consider rent rates, depreciations, or administration costs. Variable costs are those costs which vary directly with the level of output. As examples consider output-related inputs such as raw materials, direct labor, fuel and revenue-related costs such as commission.

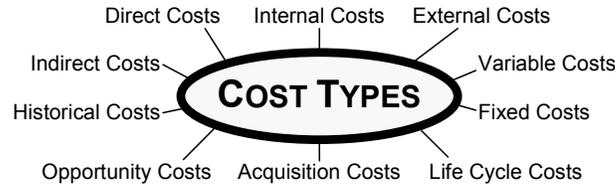


Figure 2.3: Different Types of Costs.

- **Life Cycle Cost:** Refer to the costs of an investment over its entire life cycle. This includes costs for planning, research, development, production, maintenance, disposal, and cost of spares and repair times.

This variety (cf. Fig. 2.3) makes it difficult to introduce a standard meaning for the term "costs". Instead, every evaluation approach addressing costs has to specifically describe the assumed semantics in the given context.

2.2.2 Benefits

In economic-driven IT evaluations, costs are typically justified by expected benefits to be gained through an IT investment. Generally, "benefit" is a term used to indicate an advantage, profit, or gain attained by an individual or organization. Basically, we distinguish between tangible and intangible benefits [6, 78, 79]:

- **Tangible Benefits:** Are measurable and quantifiable [196]. Typically, tangible benefits can be associated with monetary value. Thereby, one distinguishes between (i) *increased revenues* (i.e., resulting from increased revenues) and (ii) *decreased costs* (i.e., equating to cost savings).
- **Intangible Benefits:** Unlike tangible benefits, intangible benefits are typically not quantifiable in monetary terms. Instead, qualitative value (derived from subjective measures) is assigned to them [125]. As a typical example consider the impact of an investment on customer or employee satisfaction. Due to their complex quantification, intangible benefits are often not considered in a business case as they introduce a too great margin of error.

This categorization can be also applied when considering existing IT evaluation approaches. While some of them strictly focus on the quantification of tangible benefits [33, 174, 175], others consider intangible economic effects [79, 114, 144, 154].

2.2.3 Risks

Risk is the positive or negative impact an investment may have on some present situation or some future events. In professional risk assessment [20, 153], risk combines the probability of an event with the impact this event has on an assumed *risk scenario*. For example, financial risk is often considered as the unexpected variability or volatility of revenues, which can be worse or better than expected. Risk-oriented evaluation approaches are not further considered in this thesis.

2.3 A Framework for Comparing IT Evaluation Approaches

This section introduces the conceptual framework we use for classifying and comparing existing economic-driven IT evaluation approaches [141]. This framework has been developed based on existing comparison frameworks (see below), on a literature study on economic-driven IT evaluation, and an empirical study [136].

2.3.1 Existing Frameworks

In literature there exist several frameworks that aim at comparing IT evaluation approaches:

- **Andresen's Framework** [7]: This framework is based on nine criteria. Criterion 1 (*extent of involvement*) considers persons or user groups to be involved when applying the evaluation approach. Criterion 2 (*stage of IT evaluation*) addresses the question at which project stage an evaluation approach can be used. Criterion 3 (*type of impact*) considers the concrete effects that can be analyzed with an evaluation approach. Criterion 4 (*costs of a method*) deals with the effort related to the use of an approach. Criterion 5 (*number and type of evaluation*) allows to investigate the theoretical foundation of an evaluation approach. Criterion 6 (*type of investment*) helps to understand to what kind of IT investment an approach can be applied. Criterion 7 (*scope of IT evaluation*) characterizes the enterprise level an approach is tailored to (e.g., management, operational departments). Criterion 8 (*difficulty*) deals with the complexity related to the application of an evaluation approach. Finally, Criterion 9 (*type of outcome*) analyzes in which way evaluation results are presented.
- **Pietsch's Framework** [152]: This framework utilizes ten criteria, many of them addressing the same or similar issues as Andresen: (1) *theoretical foundation*, (2) *evaluation object and scope*, (3) *sources of evaluation data*, (4) *stage of IT evaluation*, (5) *flexibility*, (6) *costs of an approach*, (7) *tool support*, (8) *transparency and traceability*, (9) *completeness*, and (10) *relevance for practice*.
- **Enterprise Architecture (EA) Frameworks**: Besides, there are enterprise architecture frameworks, e.g., *Zachman's framework* [223] or the *GRAAL framework* [206, 218], which also address potential criteria for comparing IT evaluation approaches.

2.3.2 Our Comparison Framework

Though the aforementioned frameworks address many important characteristics of IT evaluation approaches, they also neglect a number of basic issues. As examples consider the data needed for using an evaluation method, the ability of an approach to allow for plausible conclusions, the liability of an approach to manipulation, or the sensitivity of evaluation methods regarding evolving information baselines (i.e., varying data quality).

For this reason we have developed a new conceptual comparison framework which combines criteria of the above frameworks with additional ones (cf. Fig. 2.4).¹ The additional criteria have been derived from a literature study on economic-driven IT evaluation and from practical needs identified in an empirical study [136] that was carried out in the context of this thesis. Specifically, we introduce one classification criteria (C-2: "Decision Support") and two evaluation criteria (E-2: "Objectiveness", E-3: "Sensitivity"), and detail the evaluation criteria "Practical Applicability" (E-4).

Comparison Framework					
I. Classification Criteria	C-1: Evaluation Viewpoint			II. Evaluation Criteria	
	Financial Viewpoint		o		E-1: Plausibility
	Work Performance Viewpoint		x		Interpretability of Evaluation Results
	Strategic Viewpoint				Transparency of Result Generation
	C-2: Decision Support				E-2: Objectiveness
					Degree of Objectiveness
	C-3: Evaluation Dimensions				E-3: Sensitivity
	Evaluation of Benefits		x		Error-Proneness of Evaluations
	Evaluation of Costs		x		Resistance against Manipulation
	Evaluation of Risks				E-4: Practical Applicability
	Evaluation of Work Performance				Flexibility
	C-4: Evaluation Scope				Efficiency (Effort)
	ex-ante		x		
	ex-post				Availability
	C-5: Evaluation Outcome				Heterogeneity
Quantification Absolute Measures		x	Data Quality		
Quantification Relative Measures			Overall Effort		
Visualization of Outcome		x	Effectiveness		
Qualitative Conclusions				Comparability	
			Correctness of Conclusions		
			E-5: Theoretical Foundation		
			Degree of Formalization		
			E-6: Tool Support		
			Simplicity of providing Tool Support		

Caption: x: supported o: optional ↑: positive ↓: negative ➔: neutral

Figure 2.4: Our Criteria Framework at a Glance.

Fig. 2.4 shows our overall comparison framework. Basic to it is the distinction between classification and evaluation criteria. *Classification criteria* (left side) allow for a general characterization of existing approaches (e.g., regarding supported evaluation viewpoints, covered evaluation dimensions, or provided evaluation outcomes). *Evaluation criteria* (right side), in turn, describe specific features of considered evaluation approaches (e.g., ability to derive plausible and objective results, theoretical foundation, and simplicity of providing tool support) and allow to assess these features.

¹This framework is partly based upon [123].

Existing Criteria	C-1	C-2	C-3	C-4	C-5	E-1	E-2	E-3	E-4	E-5	E-6
Andresen's Framework											
Extent of involvement									X		
Stage of IT evaluation			X	X							
Type of impact	X		X								
Costs of a method									X		
Number & type of evaluation										X	
Type of investment	X										
Scope of IT evaluation	X			X							
Difficulty									X		
Type of outcome					X						
Pietsch's Framework											
Theoretical foundation										X	
Evaluation object and scope	X		X								
Sources of evaluation data									X		
Stage of IT evaluation				X							
Flexibility									X		
Costs of an approach									X		
Tool support											X
Transparency and traceability						X					
Completeness						X					
Relevance for practice									X		
EA Frameworks			X	X					X		X

Figure 2.5: Combining Existing and New Criteria.

Fig. 2.5 illustrates which criteria of the existing comparison frameworks can be mapped on criteria of our own framework. In the following, we discuss the classification and evaluation criteria of our comparison framework in detail.

2.3.2.1 Classification Criteria

This section summarizes the criteria C-1 to C-5 (cf. Fig. 2.4, left side) which can be used to classify IT evaluation approaches:

- **Criterion C-1 (Evaluation Viewpoint):** We distinguish between three basic evaluation viewpoints. Methods for analyzing IT investments from the *financial viewpoint* deal with the evaluation, distribution, and consumption of financial value. Approaches for investigating IT investments from a *work performance viewpoint* evaluate the impact of IT on work and process performance. Approaches for conducting evaluations from a *strategic viewpoint* allow to analyze the impact of IT on the achievement of strategic enterprise objectives.
- **Criterion C-2 (Decision Support):** Most evaluation approaches aim at supporting decision making (e.g., investment decisions, project decisions, etc.) [117, 118]. Picking up this issue, we consider the suitability of an evaluation approach to support decisions as classification criterion.
- **Criterion C-3 (Evaluation Dimensions):** This criterion deals with the evaluation dimension that can be analyzed by an evaluation approach. In our framework, we distinguish between the evaluation of *costs*, *benefits*, *risks*, and *work performance*.

- Criterion C-4 (Evaluation Scope):** We distinguish between ex-ante and ex-post evaluations. *Ex-ante evaluations* aim at identifying the best solution in a given context. They focus on the economic feasibility of an investment, and they are typically conducted prior to an investment. However, they can also be used for evaluating an already initiated investment. Note that the accuracy of ex-ante evaluations increases with the number of available parameters during a software development project (cf. Fig. 2.6). *Ex-post evaluations*, by contrast, justify assumptions made during an ex-ante analysis, i.e., ex-post evaluations typically confirm or discard the results of a previous ex-ante evaluation.

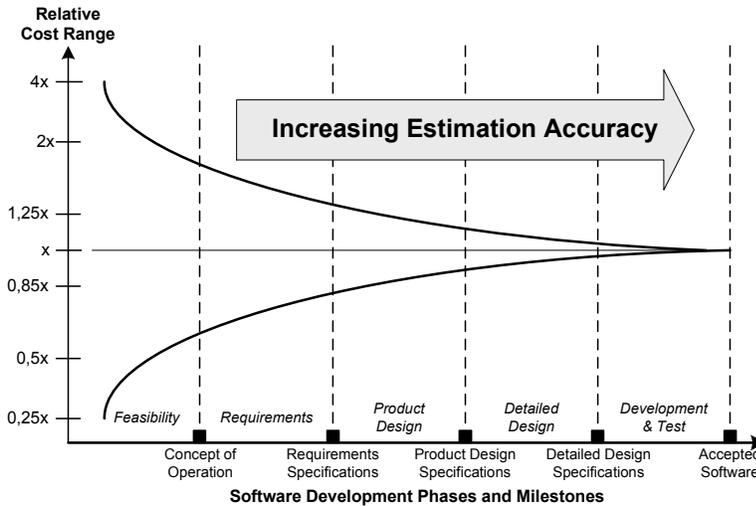


Figure 2.6: Accuracy of Estimations [29].

- Criterion C-5 (Evaluation Outcome):** We distinguish between four types of evaluation outcome: (1) *absolute figures* (i.e., single numbers, calculated sums or differences), (2) *relative figures* (relating two absolute figures and analyzing their correlation), (3) *graphical representations* (i.e., tables, charts, and outlines for illustrating and visualizing both absolute and relative figures), and (4) *textual evaluations*.

These five classification criteria allow for a general characterization of existing evaluation approaches. It is important to mention that our discussions of evaluation approaches in this chapter are organized along the three viewpoints of criterion C-1 (cf. Fig. 2.4).

2.3.2.2 Evaluation Criteria

Before discussing selected evaluation approaches, we describe the evaluation criteria E-1 to E-6 of our comparison framework in detail (cf. Fig. 2.4, right side):

- Criterion E-1 (Plausibility):** This criterion deals with the ability of an evaluation approach to derive plausible results. Plausibility is determined by two sub criteria:

(1) *interpretability of results* and (2) *transparency of result generation*. While the former addresses the clarity of evaluation results, the latter deals with the traceability of deriving an evaluation.

- **Criterion E-2 (*Objectiveness*)**: This criterion deals with the ability of an evaluation approach to produce the same or at least similar results when it is applied (to the same context) by different users. Approaches enhancing a high degree of objectiveness exhibit fewer opportunities for manipulation.
- **Criterion E-3 (*Sensitivity*)**: This criterion concerns changes in evaluation results when underlying evaluation data is modified. High sensitivity means that small modifications of evaluation data can result in significant changes. Low sensitivity, in turn, implies that even strong modifications of evaluation data do not lead to strong changes. Thus, this criterion will be a measure for *error-proneness* if evaluation data is incomplete. Furthermore, Criterion E-3 allows to draw conclusions regarding the resistance of an evaluation approach against manipulation.
- **Criterion E-4 (*Practical Applicability*)**: Three sub criteria determine the practical applicability of an evaluation approach: (1) the ability of an evaluation approach to meet the varying requirements of different application domains (i.e., its *flexibility*), (2) the efforts for accomplishing an evaluation (i.e., its *efficiency*), and (3) the ability of an evaluation approach to derive correct results (i.e., its *effectiveness*).
- **Criterion E-5 (*Theoretical Foundation*)**: Theoretical foundation deals with the availability of a constant notation stating formal rules and classes of expressions which are registered and performed in the same way by all users of an evaluation approach. Formalization contributes to make evaluations transparent and traceable. This additionally helps to sustain the plausibility of conclusions.
- **Criterion E-6 (*Tool Support*)**: Tool support is an important criterion as well, as the possibility to automate certain evaluation steps can significantly foster the practical applicability of an evaluation approach. However, providing tool support can be a difficult task to accomplish.

These six evaluation criteria allow to gain insights into the specific features of IT evaluation approaches. In the following sections we discuss existing IT evaluation approaches based on this comparison framework.

2.4 Financial Viewpoint

This section discusses approaches that can be used to accomplish evaluations from a financial viewpoint (cf. Criterion C-1). In particular, this viewpoint deals with the creation, distribution, consumption, and evaluation of economic value as well as with the prediction of revenues and expenses based on the exchange of valuable goods and services between multiple actors. Basically, one distinguishes between financial business ratios (or measures)

[153, 161] and cost-oriented evaluation approaches. While the former consider the monetary costs and benefits of an IT investment over a specified period of time (cf. Section 2.4.1 and Section 2.4.2), the latter allow to determine the costs and to better understand the cost structures related to IT investment (cf. Section 2.4.3).

2.4.1 Static Measures

Static business ratios ignore the time value of money. As examples consider *return on investment*, *payback period*, *accounting rate of return*, and *break even analysis*.

Return on Investment. Due to its simple calculation, *return on investment* (ROI) has become one of the most popular measures to understand, evaluate and compare the financial value of IT investment options. It measures the economic return of an investment, i.e., the effectiveness of using money to generate profit. More precisely, ROI describes how many times the net benefit of an IT investment (i.e., its *benefits* minus its *costs*) covers the original investment costs (whereas *benefits* are typically quantified as the money gained or lost through an IT investment):

$$\text{return on investment} = \frac{\text{benefits} - \text{costs}}{\text{costs}} * 100\%$$

There exist many variations of this definition considering the multiple interpretations and applications in different industry domains [2], e.g., the *return on invested capital* (ROIC) or the *financial ROI*. Fig. 2.7A characterizes² the ROI method based on our comparison framework (cf. Section 2.3.2). First, consider the classification criteria (left side). ROI constitutes an approach which can be clearly assigned to the financial viewpoint (C-1). It takes into account both costs and benefits and – in certain variants – risks (C-3). Further, ROI supports ex-ante as well as ex-post evaluations (C-4), and its outcome can be considered as an absolute measure (C-5).

The evaluation criteria (right side), in turn, allow to assess the ROI measure in more detail. Criterion E-1 in Fig. 2.7A, for example, indicates that result generation of the ROI method (based on the simple formula above) is easy to understand, while the interpretability of evaluation results can be considered neither as simple (i.e., positive) nor difficult (i.e., negative), but as neutral. Reason is that – though the evaluation outcome is simple – it is not always trivial to estimate the consequences and resilience of a calculated ROI. Furthermore, the degree of objectiveness is high (E-2) and providing tool support is simple (E-6).

Payback Period. The *payback period* (cf. Fig. 2.7B) is the length of time required to compensate the *initial costs* of an IT investment through its *benefits per year* (which are again quantified as the money gained or lost through the investment in one year):

$$\text{payback period} = \frac{\text{initial costs}}{\text{benefits per year}}$$

For example, a 4000\$ investment which returns 1000\$ per year would have a four year payback period. It is assumed that the investment with earliest payback period is the best one.

²In the following, we will not discuss these criteria in detail for the considered approaches.

A		Return On Investment (ROI)		
I. Classification Criteria	C-1: Evaluation Viewpoint			
	Financial Viewpoint	X		
	Work Performance Viewpoint			
	Strategic Viewpoint			
	C-2: Decision Support			
	Decision making is supported based on the calculated financial business ratio.			
	C-3: Evaluation Dimensions			
	Evaluation of Benefits			
	Evaluation of Costs	X		
	Evaluation of Risks	O		
	Evaluation of Work Performance			
	C-4: Evaluation Scope			
	ex-ante	X		
	ex-post	X		
	C-5: Evaluation Outcome			
Quantification Absolute Measures	X			
Quantification Relative Measures				
Visualization of Outcome				
Qualitative Conclusions				
II. Evaluation Criteria	E-1: Plausibility			
	Interpretability of Evaluation Results		⬆	
	Transparency of Result Generation		⬆	
	E-2: Objectiveness			
	Degree of Objectiveness		⬆	
	E-3: Sensitivity			
	Error-Proneness of Evaluations		⬆	
	Resistance against Manipulation		⬇	
	E-4: Practical Applicability			
	Flexibility		⬆	
	Efficiency (Effort)	Data Collection & Preparation		⬆
		Availability		⬆
		Heterogeneity		⬆
		Data Quality		⬆
	Effectiveness	Overall Effort		⬆
Comparability			⬆	
Correctness of Conclusions			⬆	
E-5: Theoretical Foundation				
Degree of Formalization		⬆		
E-6: Tool Support				
Simplicity of providing Tool Support		⬆		

B		Payback Period (PP)		
I. Classification Criteria	C-1: Evaluation Viewpoint			
	Financial Viewpoint	X		
	Work Performance Viewpoint			
	Strategic Viewpoint			
	C-2: Decision Support			
	Decision making is supported based on the calculated financial business ratio.			
	C-3: Evaluation Dimensions			
	Evaluation of Benefits	X		
	Evaluation of Costs	X		
	Evaluation of Risks	O		
	Evaluation of Work Performance			
	C-4: Evaluation Scope			
	ex-ante	X		
	ex-post	X		
	C-5: Evaluation Outcome			
Quantification Absolute Measures	X			
Quantification Relative Measures				
Visualization of Outcome	O			
Qualitative Conclusions				
II. Evaluation Criteria	E-1: Plausibility			
	Interpretability of Evaluation Results		⬆	
	Transparency of Result Generation		⬆	
	E-2: Objectiveness			
	Degree of Objectiveness		⬆	
	E-3: Sensitivity			
	Error-Proneness of Evaluations		⬆	
	Resistance against Manipulation		⬇	
	E-4: Practical Applicability			
	Flexibility		⬆	
	Efficiency (Effort)	Data Collection & Preparation		⬆
		Availability		⬆
		Heterogeneity		⬆
		Data Quality		⬆
	Effectiveness	Overall Effort		⬆
Comparability			⬆	
Correctness of Conclusions			⬆	
E-5: Theoretical Foundation				
Degree of Formalization		⬆		
E-6: Tool Support				
Simplicity of providing Tool Support		⬆		

Caption: x: supported o: optional ⬆: positive ⬇: negative ⬆: neutral

Figure 2.7: Return on Investment and Payback Period.

However, this is not always reasonable for investments with large benefits to be expected in future. The payback period is a simple measure, which has several limitations. It does not address the time value of money and only considers the compensation of initial investment costs. Fig. 2.7B shows the evaluation of the payback period approach based on our framework (cf. Section 2.3.2).

Accounting Rate of Return. The *accounting rate of return* is a measure of profitability that associates the expected *overall benefits* with the investment’s *overall costs*:

$$\text{accounting rate of return} = \frac{\text{overall benefits}}{\text{overall costs}}$$

In this approach (cf. Fig. 2.8A), costs and benefits for an investment’s entire life cycle are addressed (unlike in the payback period method, which considers only the period it takes to recoup the original investment). Like ROI and payback period, the ARR method ignores the time value of money.

Break Even Analysis. The *break even analysis* (cf. Fig. 2.8B) determines the point where the total financial benefit of an IT investment (i.e., the money gained) equals the total costs of the IT investment. It can be also used when (some) benefits are uncertain or intangible.

2.4.2 Dynamic Measures

Dynamic approaches consider the time value of money by comparing the cash outflows (or costs) prior to an investment with its expected cash inflows (or benefits). As examples consider *net present value* and *internal rate of return*.

Net Present Value. The *net present value* (NPV) (cf. Fig. 2.9A) is a technique where all expected cash outflows (or costs) and cash inflows (or benefits) are discounted to the

A Accounting Rate Of Return (ARR)		B Break Even Analysis	
I. Classification Criteria	C-1: Evaluation Viewpoint	II. Evaluation Criteria	E-1: Plausibility
	Financial Viewpoint		Interpretability of Evaluation Results
	Work Performance Viewpoint		Transparency of Result Generation
	Strategic Viewpoint		
	C-2: Decision Support		E-2: Objectiveness
	Allows to measure the profitability by associating the expected overall benefits with the expected overall costs.		Degree of Objectiveness
	C-3: Evaluation Dimension		E-3: Sensitivity
	Evaluation of Benefits		Error-Proneness of Evaluations
	Evaluation of Costs		Resistance against Manipulation
	Evaluation of Risks		
	Evaluation of Work Performance		E-4: Practical Applicability
	C-4: Evaluation Scope		Flexibility
	ex-ante		Efficiency (Effort)
	ex-post		Data Collection & Preparation
			Availability
	Heterogeneity		
	Data Quality		
	Overall Effort		
	Comparability		
	Correctness of Conclusions		
C-5: Evaluation Outcome	E-5: Theoretical Foundation		
Quantification Absolute Measures	Degree of Formalization		
Quantification Relative Measures			
Visualization of Outcome	E-6: Tool Support		
Qualitative Conclusions	Simplicity of providing Tool Support		

Caption: x: supported o: optional +: positive -: negative #: neutral

Figure 2.8: Accounting Rate of Return and Break Even Analysis.

present point in time. This is done by applying a discount rate to the difference of inflows and outflows (i.e., the NPV is calculated from the current time t_0 to some future point in time T):

$$NPV = \sum_{i=0}^T \frac{B_i - C_i}{(1+d)^i}$$

B_i constitutes the assumed benefit (i.e., money gained or lost through the investment) for the i^{th} period in future. C_i , in turn, denotes the assumed costs for the same period (i.e., the sum of the expected outflows). For each period, the value of all cash outflows is subtracted from the value of all expected cash inflows. The difference between inflows and outflows across all considered periods is the net present value (with d the assumed discount factor). Only investments with a positive NPV are acceptable as their return exceeds the discount rate.

Internal Rate of Return. As another example of a dynamic measure consider the *internal rate of return* (IRR) (cf. Fig. 2.9B). IRR is the annual rate at which an investment is estimated to pay off. It is important to mention that IRR and NPV are not equivalent (though related). In particular, IRR does not use a discount rate. Instead, IRR takes into account the time value of money by considering expected life cycle cash flows.

2.4.3 Cost-oriented Approaches

The approaches discussed in the following focus on the analysis and justification of IT investment costs: *zero base budgeting approach*, *cost effectiveness analysis*, *target costing approach*, and *total cost of ownership approach*. Note that all these approaches go beyond the scope of the approaches discussed in Section 2.4.1 and Section 2.4.2.

A		Net Present Value (NPV)		B		Internal Rate of Return (IRR)		
I. Classification Criteria	C-1: Evaluation Viewpoint	Financial Viewpoint	X	II. Evaluation Criteria	E-1: Plausibility	Interpretability of Evaluation Results	⬆	
		Work Performance Viewpoint					Transparency of Result Generation	⬆
		Strategic Viewpoint			E-2: Objectiveness	Degree of Objectiveness	⇒	
	C-2: Decision Support	Decision making is supported based on the calculated financial business ratio.			E-3: Sensitivity	Error-Proneness of Evaluations	⬇	
						Resistance against Manipulation	⬇	
	C-3: Evaluation Dimension	Evaluation of Benefits	X		E-4: Practical Applicability	Flexibility	⇒	
		Evaluation of Costs	X			Efficiency (Effort)	Data Collection & Preparation	⇒
		Evaluation of Risks	O				Availability	⇒
		Evaluation of Work Performance					Heterogeneity	⇒
							Data Quality	⇒
	C-4: Evaluation Scope	ex-ante	X		Overall Effort	⬆		
		ex-post			Comparability	⇒		
					Correctness of Conclusions	⇒		
	C-5: Evaluation Outcome	Quantification Absolute Measures	X	E-5: Theoretical Foundation	Degree of Formalization	⬆		
		Quantification Relative Measures						
		Visualization of Outcome		E-6: Tool Support	Simplicity of providing Tool Support	⬆		
		Qualitative Conclusions						

Caption: x: supported o: optional ⬆: positive ⬇: negative ⇒: neutral

Figure 2.9: Net Present Value and Internal Rate of Return.

Zero Base Budgeting. The *zero base budgeting* approach (cf. Fig. 2.10A) is a budgeting method. It assumes that all costs of an investment have to be justified for each new period (e.g., a month, a quarter, a year), i.e., fundings are continuously justified [152, 227].

Cost Effectiveness Analysis. A *cost effectiveness analysis* (cf. Fig. 2.10B) allows to compare and select the best out of several investment options [175]. A scoring model identifies key performance criteria for candidate investments, assigns a score to each criterion (based on subjective assessments), and finally computes a weighted overall score for each candidate investment.

Generally, cost effectiveness analysis can be applied to different scenarios. As a first scenario consider minimization of costs for a given level of effectiveness. An example may be the choice among several printers. Each printer may be equally effective, and the issue is to choose the one with the lowest expected life cycle costs. As a second example consider maximization of effectiveness for a given amount of costs. As example take the choice among several database management systems, where costs are identical, but features vary. As a final scenario consider maximization of effectiveness and minimization of costs at the same time. As an example consider the selection of an engineering workstation where both performance features and costs might differ significantly.

This approach will be particularly useful if the benefits of an investment are quantifiable mainly in non-monetary dimensions. It does not allow for justifying an investment, i.e., it does not explicitly address the question whether benefits exceed costs.

Total Cost of Ownership. The *total cost of ownership (TCO) approach* (cf. Fig. 2.11A) is a method to assess direct and indirect costs related to an investment [153]. A TCO assessment ideally results in a statement reflecting not only purchase costs, but also costs related to the future use and maintenance of the investment (as long as these costs can be made explicit). This includes costs caused by (planned and unplanned) failure or outage, costs for

A Zero Base Budgeting		B Cost Effectiveness Analysis	
I. Classification Criteria	C-1: Evaluation Viewpoint	II. Evaluation Criteria	E-1: Plausibility
	Financial Viewpoint		Interpretability of Evaluation Results
	Work Performance Viewpoint		Transparency of Result Generation
	Strategic Viewpoint		
	C-2: Decision Support		E-2: Objectiveness
	Decision making is supported in the sense of project monitoring and control.		Degree of Objectiveness
	C-3: Evaluation Dimensions		E-3: Sensitivity
	Evaluation of Benefits		Error-Proneness of Evaluations
	Evaluation of Costs		Resistance against Manipulation
	Evaluation of Risks		
C-4: Evaluation Scope	E-4: Practical Applicability		
ex-ante	Flexibility		
ex-post	Efficiency (Effort)		
	Data Collection & Preparation		
	Availability		
	Heterogeneity		
	Data Quality		
	Overall Effort		
	Comparability		
	Correctness of Conclusions		
C-5: Evaluation Outcome	E-5: Theoretical Foundation		
Quantification Absolute Measures	Degree of Formalization		
Quantification Relative Measures			
Visualization of Outcome	E-6: Tool Support		
Qualitative Conclusions	Simplicity of providing Tool Support		

Caption: x: supported o: optional ▲: positive ▼: negative ➔: neutral

Figure 2.10: Zero Base Budgeting and Cost Effectiveness Analysis.

diminished performance incidents (i.e., if users are kept waiting), costs for security breaches (in loss of reputation and recovery costs), costs for disaster preparedness and recovery, floor space, electricity, development expenses, testing infrastructure and expenses, quality assurance, incremental growth, and decommissioning.

A Total Cost of Ownership (TCO)		B Target Costing	
I. Classification Criteria	C-1: Evaluation Viewpoint	II. Evaluation Criteria	E-1: Plausibility
	Financial Viewpoint		Interpretability of Evaluation Results
	Work Performance Viewpoint		Transparency of Result Generation
	Strategic Viewpoint		
	C-2: Decision Support		E-2: Objectiveness
	Financial estimate for assessing the direct and indirect costs related to the purchase of a capital investment.		Degree of Objectiveness
	C-3: Evaluation Dimensions		E-3: Sensitivity
	Evaluation of Benefits		Error-Proneness of Evaluations
	Evaluation of Costs		Resistance against Manipulation
	Evaluation of Risks		
C-4: Evaluation Scope	E-4: Practical Applicability		
ex-ante	Flexibility		
ex-post	Efficiency (Effort)		
	Data Collection & Preparation		
	Availability		
	Heterogeneity		
	Data Quality		
	Overall Effort		
	Comparability		
	Correctness of Conclusions		
C-5: Evaluation Outcome	E-5: Theoretical Foundation		
Quantification Absolute Measures	Degree of Formalization		
Quantification Relative Measures			
Visualization of Outcome	E-6: Tool Support		
Qualitative Conclusions	Simplicity of providing Tool Support		

Caption: x: supported o: optional ▲: positive ▼: negative ➔: neutral

Figure 2.11: Total Cost of Ownership and Target Costing.

Target Costing. *Target costing* (cf. Fig. 2.11B) is a technique for planning and realizing a defined amount of costs at which a product with a specified functionality has to be produced to generate profitability [152, 182]. Besides, it also allows for identifying cost reductions by focusing on major "design drivers" that influence costs. Therefore, target costing integrates strategic business and profit planning, competitive research and analysis, market research

and customer requirements, research and development, technology advances, and product development. Target costing uses *product portfolio profit plans* to provide strategic summary schedules for product development, introduction and replacement, or IT investments. Generally, target costing is different from a simple expenditure control mechanism as it aims at determining market-based prices for envelopes of features based upon market and competitive conditions in which price/volume relationships are examined.

2.5 Process Viewpoint

This viewpoint focuses on evaluating operational work and business process performance [80, 117, 118, 119]. Characteristic to all approaches described in the following are quantifications based on information about process/work activities (e.g., start/completion times, average duration times, waiting and idle times), process/work resources (e.g., resources needed, input and output data, or size of work queues), and quality metrics (e.g., failed or successful processes/work activities). In the following, we describe four approaches: *times savings times salary approach* (Section 2.5.1), *hedonic wage model* (Section 2.5.2), *activity-based costing* (Section 2.5.3), and *business process intelligence* (Section 2.5.4).

2.5.1 Times Savings Times Salary Approach

The *times savings times salary* (TSTS) approach (cf. Fig. 2.12A) [174, 175] is based on the assumption that an employee's salary is a measure of his "contribution" or "value" to an organization. Its goal is to estimate the work time an IT investment will save, and then to multiply that time with the salaries of all affected employees. For example, if an employee with a wage of W [\$ per week] and a work time of T [hours per week] is currently devoting x_j hours per week to activity j , and if a new IT investment leads to a situation in which the employee now has to devote y_j hours per week to activity j , then the value of the IT investment regarding activity j can be calculated as follows:

$$Value = \frac{(x_j - y_j)}{T} * W$$

The TSTS approach is based on five premises. *First*, it assumes that an employee's value corresponds to his costs for an organization. *Second*, it assumes that saving x percent of an employee's time is worth x percent of the employee's costs. *Third*, it is based on the assumption that the resources of an organization are efficiently allocated, i.e., that the costs of additional employees are balanced against their value for an organization. Consequently, the number of employees would not be higher even if it had been possible to hire additional employees. *Fourth*, the TSTS approach assumes that work comparable in value to current work remains to be done. In other words, it is assumed that there is additional work to which any saved time could be devoted, and that the value of work is comparable to work currently done. *Fifth*, it assumes that saved time will be allocated among an employee's productive activities.

The TSTS approach is easy to accomplish. As it is time-based, it can be used for evaluating the impact of IT on work performance and also on business process performance.

However, there are three major problems derogating its use in practice. *First*, it is assumed that an employee’s value corresponds to his cost to an organization. This will be true if the organization is not resource-constrained and has hired the optimal number of employees. However, in general, the possibility that an employee’s value exceeds his costs should not be automatically dismissed. If his value is greater than his cost, then this approach will underestimate the true value of saved time. *Second*, and more important, the TSTS approach does not take into account how the saved time is used. Instead, it is implicitly assumed that saved time is efficiently reallocated among available work activities. Consequently, it cannot be assumed that a particular time allocation will take effect. *Third*, the calculation of the saved time implies that benefits are automatically realized. However, typically they are not, i.e., saved time may not result in economic benefit. The value of a new IT investment, for example, may be low or high, depending on how an organization and its flow of work is managed. The TSTS approach does not capture this variability.

A Times Savings Times Salary Model		B Hedonic Wage Model	
I. Classification Criteria	C-1: Evaluation Viewpoint	II. Evaluation Criteria	E-1: Plausibility
	Financial Viewpoint		Interpretability of Evaluation Results
	Work Performance Viewpoint X		Transparency of Result Generation
	Strategic Viewpoint		E-2: Objectiveness
	C-2: Decision Support		Degree of Objectiveness
	Allows to estimate the value of a new IT investment with respect to saved work time. Can be used for decision making.		E-3: Sensitivity
	C-3: Evaluation Dimension		Error-Proneness of Evaluations
	Evaluation of Benefits X		Resistance against Manipulation
	Evaluation of Costs		E-4: Practical Applicability
	Evaluation of Risks		Flexibility
	Evaluation of Work Performance		Efficiency (Effort)
	C-4: Evaluation Scope		Data Collection & Preparation
	ex-ante X		Heterogeneity
	ex-post		Data Quality
	C-5: Evaluation Outcome		Overall Effort
Quantification Absolute Measures X	Effectiveness		
Quantification Relative Measures	Comparability		
Visualization of Outcome	Correctness of Conclusions		
Qualitative Conclusions	E-5: Theoretical Foundation		
	Degree of Formalization		
	E-6: Tool Support		
	Simplicity of providing Tool Support		

Caption: **x**: supported **o**: optional **↑**: positive **↓**: negative **↔**: neutral

Figure 2.12: Times Savings Times Salary Approach and Hedonic Wage Model.

2.5.2 Hedonic Wage Model

Like the TSTS approach, the *hedonic wage model* (cf. Fig. 2.12B) assumes that employees perform activities of different intrinsic value [174]. The value of an IT investment, in particular, is determined by its ability to contribute to the restructuring of existing work patterns and to cause a shift in an employee’s work profile by replacing low-value-activities with tasks in a higher category. Such a restructuring can not only increase the efficiency (doing more of the same thing in the same amount of time) but also the effectiveness (doing more valuable work) of an organization and its employees.

The hedonic wage model assumes that employees carry out categories of activities (together making up their work profile). The aggregation of all work profiles into one *work profile matrix* (cf. Fig. 2.13) characterizes the work profile of an organization (with the *level in the job hierarchy* as the first dimension and the *type of activity* as the second one).

Level in Job Hierarchy	Type of Activity						Wage per Hour
	Management Activity (T1)	Specialist Activity (T2)	Routine Activity (T3)	Assistant Activity (T4)	Service Activity (T5)	Other Activity (T6)	
Manager (S1)	50%	15%	20%	5%	5%	5%	160€
Specialist (S2)	10%	60%	10%	10%	5%	5%	120€
Clerk (S3)	2%	15%	35%	20%	18%	10%	80€
Assistant (S4)	0%	0%	10%	55%	30%	5%	60€
Secretary (S5)	0%	0%	0%	15%	70%	15%	45€

→ Each clerk utilizes (in average) 15% of his work time for specialist activities

Figure 2.13: 1st Work Profile Matrix.

Note that both the number of job levels and activity types (i.e., the dimensions of the work profile matrix) are organization-specific, i.e., they may differ. In the example from Fig. 2.13, there exist five job levels (managers, specialists, clerks, assistants, and secretaries) and six types of activities (management activity, specialist activity, routine activity, assistant activity, service activity, and other activity). Each value in Fig. 2.13 specifies how much time (in %) an employee (belonging to one of the five job levels) uses to conduct the six considered activities.

Linear System:

$$\begin{array}{l}
 \mathbf{S1}: 0.50T1 + 0.15T2 + 0.20T3 + 0.05T4 + 0.05T5 + 0.05T6 = 160.00 \text{ €/h} \\
 \mathbf{S2}: 0.10T1 + 0.60T2 + 0.10T3 + 0.10T4 + 0.05T5 + 0.05T6 = 120.00 \text{ €/h} \\
 \mathbf{S3}: 0.02T1 + 0.15T2 + 0.35T3 + 0.20T4 + 0.18T5 + 0.10T6 = 80.00 \text{ €/h} \\
 \mathbf{S4}: 0.00T1 + 0.00T2 + 0.10T3 + 0.55T4 + 0.30T5 + 0.05T6 = 60.00 \text{ €/h} \\
 \mathbf{S5}: 0.00T1 + 0.00T2 + 0.00T3 + 0.15T4 + 0.70T5 + 0.15T6 = 45.00 \text{ €/h}
 \end{array}
 \left. \vphantom{\begin{array}{l} \mathbf{S1} \\ \mathbf{S2} \\ \mathbf{S3} \\ \mathbf{S4} \\ \mathbf{S5} \end{array}} \right\} \text{Value} \left\{ \begin{array}{l} 230.62 \text{ \$/h} \\ 130.55 \text{ \$/h} \\ 96.93 \text{ \$/h} \\ 63.87 \text{ \$/h} \\ 50.60 \text{ \$/h} \end{array} \right.$$

Figure 2.14: Linear System of Equations.

From this matrix, a linear system of equations is derived (cf. Fig. 2.14). Solving this system of equations (not shown here), it becomes possible to determine the value of the different activities [174]. In Fig. 2.14, for example, a manager (job level S1) has a value of 230.62\$/h for the organization, but generates only costs of 160\$ (his average wage per hour).

Level in Job Hierarchy	Type of Activity						Wage per Hour
	Management Activity (T1)	Specialist Activity (T2)	Routine Activity (T3)	Assistant Activity (T4)	Service Activity (T5)	Other Activity (T6)	
Manager (S1)	60%	20%	5%	5%	5%	5%	160€
Specialist (S2)	20%	60%	5%	5%	5%	5%	120€
Clerk (S3)	2%	15%	35%	20%	18%	10%	80€
Assistant (S4)	0%	0%	30%	35%	30%	5%	60€
Secretary (S5)	0%	0%	0%	15%	60%	25%	45€

Figure 2.15: 2nd Work Profile Matrix (Changes = Gray Cells).

The value of an IT investment is derived based on a second work profile matrix. This second matrix reflects the (assumed) change in the work profile of an organization which is caused

by the investment. Like before, this second matrix is also converted into a linear system of equations which is then solved. The value of the IT investment can be determined by comparing values of job levels before and after the IT investment.

The hedonic wage model is similar to the TSTS approach, but avoids certain restrictive assumptions. For example, it produces more accurate value estimates. By estimating pre- and post-implementation work profile matrices, the projected values can be audited. However, disadvantages like the insufficient evaluation of qualitative factors remain.

2.5.3 Activity-based Costing

Activity-based costing (ABC) (cf. Fig. 2.16A) is a method of allocating costs to products and services. ABC helps to identify areas of high overhead costs per unit and therewith to find ways to reduce costs. Generally, an ABC analysis comprises the following steps:

- **Step 1:** The *scope of the business activities to be analyzed* has to be identified (e.g., based on activity decomposition).
- **Step 2:** The identified *activities are classified*. Typically, one distinguishes between *value adding* or *non-value adding* activities, between *primary* or *secondary* activities, and between *required* or *non-required* activities. An activity will be considered as value-adding (compared to a non-value adding one) if the output of the activity is directly related to customer requirements, services or products (as opposed to administrative or logistical outcomes). Primary activities directly support the goals of an organization (whereas secondary activities support primary ones). Required (unlike non-required) activities are those that must always be performed.
- **Step 3:** For each activity creating the products or services of an organization, *costs are gathered*. These costs can be determined based on salaries and expenditures for research, machinery, or office furniture.
- **Step 4:** *Activities and costs are combined* and the total input cost for each activity is derived. This allows for calculating the total costs consumed by an activity. However, at this stage, only costs are calculated. It is not yet determined where the costs originate from.
- **Step 5:** The *"activity unit cost" is calculated*. Though activities may have multiple outputs, one output is identified as the primary one. The "activity unit cost" is calculated by dividing the total input cost (including assigned costs from secondary activities) by the primary activity output. Note that the primary output must be measurable and its volume or quantity obtainable. From this, a "bill of activities" is derived which contains a set of activities and the amount of costs consumed by each activity. Then, the amount of each consumed activity is extended by the activity unit cost and is added up as a total cost for the bill of activity.
- **Step 6:** The calculated activity unit costs and bills of activity are used for *identifying candidates for business process improvement*.

ABC is an approach to make costs related to activities transparent. However, the correct accomplishment of an ABC analysis causes significant efforts and requires a lot of experience. Often, it may be not transparent, for example, which costs are caused by which activity.

A		Activity-based Costing	
I. Classification Criteria	C-1: Evaluation Viewpoint		
	Financial Viewpoint	o	
	Work Performance Viewpoint	x	
	Strategic Viewpoint		
	C-2: Decision Support		
	Allows to allocate costs to products and services. Decision making is only implicitly supported.		
	C-3: Evaluation Dimension		
	Evaluation of Benefits	x	
	Evaluation of Costs	x	
	Evaluation of Risks		
Evaluation of Work Performance			
C-4: Evaluation Scope			
ex-ante	x		
ex-post	x		
C-5: Evaluation Outcome			
Quantification Absolute Measures	x		
Quantification Relative Measures			
Visualization of Outcome	x		
Qualitative Conclusions			
II. Evaluation Criteria	E-1: Plausibility		
	Interpretability of Evaluation Results		⬆
	Transparency of Result Generation		⬆
	E-2: Objectiveness		
	Degree of Objectiveness		⬆
	E-3: Sensitivity		
	Error-Proneness of Evaluations		⬇
	Resistance against Manipulation		➡
	E-4: Practical Applicability		
	Flexibility		⬆
Efficiency (Effort)	Data Collection & Preparation	Availability	➡
		Heterogeneity	⬆
		Data Quality	⬇
Effectiveness	Overall Effort		⬆
	Comparability		⬆
	Correctness of Conclusions		➡
E-5: Theoretical Foundation			
Degree of Formalization		➡	
E-6: Tool Support			
Simplicity of providing Tool Support		➡	

B		Business Process Intelligence	
I. Classification Criteria	C-1: Evaluation Viewpoint		
	Financial Viewpoint	o	
	Work Performance Viewpoint	x	
	Strategic Viewpoint		
	C-2: Decision Support		
	Applies business intelligence concepts to business processes. Decision making is only implicitly supported.		
	C-3: Evaluation Dimension		
	Evaluation of Benefits	x	
	Evaluation of Costs	x	
	Evaluation of Risks		
Evaluation of Work Performance	x		
C-4: Evaluation Scope			
ex-ante	x		
ex-post	x		
C-5: Evaluation Outcome			
Quantification Absolute Measures	x		
Quantification Relative Measures	x		
Visualization of Outcome	x		
Qualitative Conclusions			
II. Evaluation Criteria	E-1: Plausibility		
	Interpretability of Evaluation Results		⬆
	Transparency of Result Generation		⬆
	E-2: Objectiveness		
	Degree of Objectiveness		⬆
	E-3: Sensitivity		
	Error-Proneness of Evaluations		⬇
	Resistance against Manipulation		➡
	E-4: Practical Applicability		
	Flexibility		⬆
Efficiency (Effort)	Data Collection & Preparation	Availability	⬆
		Heterogeneity	⬆
		Data Quality	⬇
Effectiveness	Overall Effort		⬆
	Comparability		⬆
	Correctness of Conclusions		➡
E-5: Theoretical Foundation			
Degree of Formalization		➡	
E-6: Tool Support			
Simplicity of providing Tool Support		⬆	

Caption: x: supported o: optional ⬆: positive ⬇: negative ➡: neutral

Figure 2.16: Activity-based Costing and Business Process Intelligence.

2.5.4 Business Process Intelligence

Enterprises are aiming at continuous optimizations of their business processes [43, 66, 213]. An important factor in this context is the availability of adequate metrics. DeMarco stated in 1983: "You can't manage what you can't control, and you can't control what you can't measure" [56]. This also applies for business processes. In order to effectively manage them, process logic has to be explicitly defined at build-time and process instances have to be flexibly controlled during run-time (e.g. using process management systems). A promising approach in this respect is providing *business process intelligence* (BPI) concepts.

BPI applies business intelligence concepts (e.g., analytical applications) to processes [42, 71, 85, 176]. It is based on the analysis of process execution data (e.g., related to the start and completion of process activities, or the resources needed by a process activity) and the automatic derivation of (optimized) process models and performance characteristics from these data. It is implemented as a set of integrated tools providing features for the analysis, mining, prediction, control, and optimization of processes. Its overall goal is to extend performance management to business processes (cf. Fig. 2.16B). We discuss BPI in detail along a conceptual reference architecture [132, 128]. This architecture comprises three major levels (cf. Fig. 2.17).

Level 1 is responsible for the extraction of process execution data from the information systems supporting the monitored business processes. Typically, the implementation of a particular business process is scattered over heterogeneous information systems each of them using a different representation for process log files. While some IS provide event-based execution logs (audit trails [228]) with detailed information, others maintain only

simple process logs. Therefore, the syntactical and semantic integration of these log data is a challenging task. In practice, very often message brokers and "extract transform load" (ETL) modules (known from data warehousing) are used for this purpose. Furthermore, a central *repository* (*process warehouse*) stores collected control and application data generated during real process executions. Besides, estimated reference values (e.g., derived from process simulations) can be stored in a database (called *process warehouse* [228]) what allows for delta analysis (i.e., comparison of reference values and real process data).

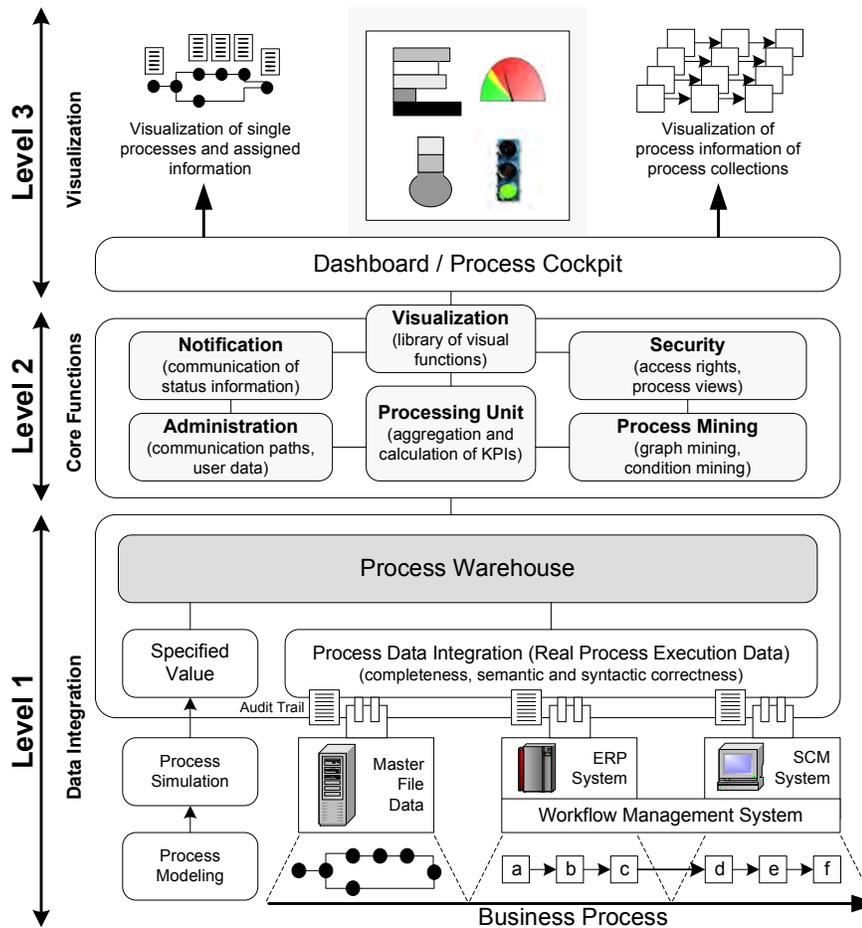


Figure 2.17: Reference Architecture for Business Process Intelligence.

Level 2 of our reference architecture (cf. Fig. 2.17) implements BPI *core functions*. In order to measure and evaluate process performance, the *processing unit* aggregates and calculates *key performance indicators* (KPI) (e.g., process cycle time or number of processes completed within a given period of time) based on the data provided by the process warehouse. To be able to quickly react to critical process events (e.g., lack of resources needed to complete a process step) the *notification component* provides functions to send messages to relevant persons (e.g., the process administrator). To deal with confidential data, a *security*

component controls the access to (aggregated) process data (e.g., through generated process views). Similar issues are known from data warehousing. The *process mining component* is responsible for the automated derivation of (optimized) process models based on logged execution data (e.g., by algorithms and tools). Thereby, the correct induction of process models depends on the completeness and quality of available process log data. Finally, the *administration component* provides support functions, e.g., for user management.

Level 3 (cf. Fig. 2.17) is responsible for the *visualization* of processes and aggregated process information (i.e., information about a collection of process instances). The *visualization component* is providing a library of presentation elements (e.g., traffic lights or bar charts) for the design of user-specific presentation forms (dashboards).

This reference architecture can be applied using contemporary BPI tools. Examples of such tools include *Websphere Business Integration Monitor*, *ARIS Process Performance Manager*, and *BizTalk Server Business Activity Monitoring Framework*. Table 2.1 benchmarks BPI features of these tools³ from very positive "++" to very negative "-". All of them assume the availability of event-based process execution data. Contemporary BPI tools, however, do not cover all aspects of the described architecture (e.g., data integration). Therefore, other software tools (e.g., message broker) are needed as well.

Evaluation Criteria	ARIS -PPM	WBI -Monitor	BizTalk (BAM)	ADO score
Degree of Details	+	+	++	-
Process Visualization	++	+	-	---
Modification	-	+	-	+
Delegation Possibilities	---	++	---	---
Analysis Possibilities	++	-	++	-
Simulation Possibilities	---	---	---	---
Information Transfer	+	+	+	+
Border Value Definition	+	+	-	-
Result Commentation	++	---	---	---
Cross-application Monitoring	++	---	-	+
Cross-organizational Monitoring	++	-	-	+
Portability	+	+	---	
Security Support	-	-		
System Requirements	+	+	+	
Administration	+			

Table 2.1: Business Process Intelligence Tools.

BPI tools utilize metrics to derive (aggregated) process information and to generate status reports. In the following we introduce three use cases for BPI and discuss the benefits arising in this context (cf. Fig. 2.18):

- **Use Case 1: Information System Alignment.** BPI can be used to support the development and maintenance of process-oriented information systems. In particular, it provides valuable information for aligning the information systems to the business processes (e.g., information about the adequacy of provided business functions).

³A detailed study including a comparison of the features, strengths, and limitations of existing BPI tools can be found in [104].

- Use Case 2: Business Process Optimization.** BPI can be used to identify "critical" scenarios that may occur during the execution of a business process (e.g., non-availability of resources, unnecessary waiting and idle times). Process mining [203, 73] as an important BPI concept allows for the continuous derivation of optimized process models. This, in turn, reduces the total effort necessary for "manual" process analyses. As optimizations are based on real data, their implementation tends to be much more effective than other approaches (e.g., the disclosure of optimization potentials by process simulation based on estimated data).

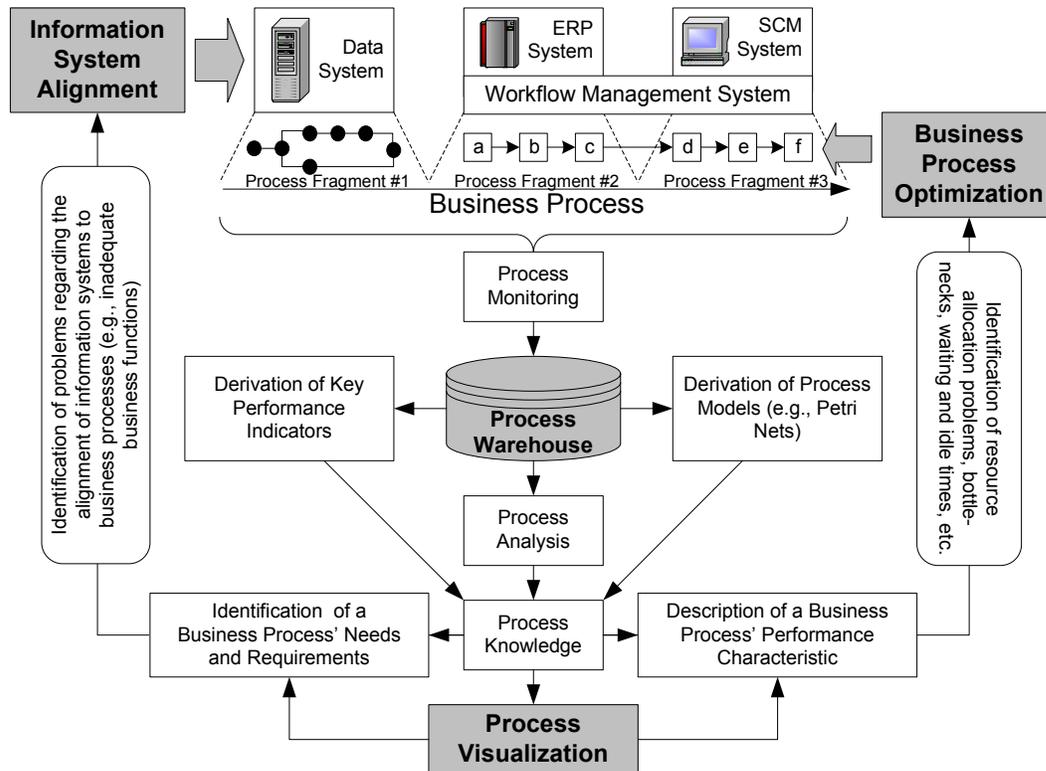


Figure 2.18: Realization of BPI Benefits.

- Use Case 3: Visualization of Process Information:** Due to the fragmented support of business processes, their control is distributed over several operational systems, i.e., we cannot always assure that controlled execution by one control system (e.g., process management system) is possible. Nevertheless, when collecting the respective log data from the different systems, it becomes possible to provide monitoring and visualization support for the overall business process [24, 110]. The information to be visualized include complete process schemas and process instances (e.g., control and data flows, activity states) as well as other process-related data (e.g., application data) [23, 169, 167]. Most BPI tools include features to visualize processes and related aspects. ARIS PPM, for example, offers a detailed tree view to illustrate the hierar-

chical relationships between processes and sub processes. Particularly the analysis of entire process maps becomes easier using such or comparable features.

However, contemporary visualization modules of BPI tools still lack some important issues: visualization of processes running on different platforms as well as features to customize process visualizations. Regarding the latter, for example, it is an objective to better adjust the process information to be visualized for different purposes. Process views, for example, target at providing either context-aware (e.g., visualization of control flow) or user-aware (e.g., high-level aggregated process data for managers) process information. Providing such advanced visualization features will further increase the advantages of this use case. In summary, the visualization and monitoring of distributed processes is a complex, though quite useful BPI use case.

Besides the benefits offered by these three use cases, other promising perspectives arise from the use of BPI tools:

- ***Real-time Enterprise***: The ability of organizations to process enterprise data as fast as possible is considered to be success-critical [177]. BPI tools are (partially) able to provide and process information in real-time and therefore enable faster reactions to internal or external events (e.g., responses to critical situations).
- ***On-demand Planning***: BPI tools do not only enable the continuous monitoring and control of business processes and their performance, but of the supporting IS and their effectiveness as well. Thus, performance problems (e.g., process deadlocks or resource allocation problems) can be identified and solved earlier.
- ***Reduced Planning Risks***: As the output of business processes (e.g., of production processes) can be predicted more precisely using BPI tools, planning risks can be reduced (e.g., regarding sales and distribution efforts).
- ***Effective Guidance of IT and Business Investments***: BPI tools can support enterprise investment decisions as they help to identify bottlenecks that can only be corrected making additional purchases of software, hardware or expertise.

BPI tools support a broad spectrum of use cases. Currently there is a growing interest in BPI tools of both vendors and customers. However, BPI is a technology that enables the measurement of business process performance. It is not a methodology like the TSTS approach or the hedonic wage model that allow for evaluating the effects of an IT investment prior to this investment. As it requires the availability of real process execution data, it can be mainly only used for a-posteriori assessments.

2.6 Strategic Viewpoint

In the previous sections we have discussed approaches that can be used to evaluate IT investments from either a financial or a work performance viewpoint. In this section, we discuss approaches that can be used to evaluate IT investments from a strategic viewpoint, e.g.,

regarding their contribution to the achievement of strategic enterprise objectives [47, 194]. However, strategy evaluation is a complex task as the strategic impact of IT investments is confronted with a considerable measure of uncertainty.

In the following, we describe four evaluation approaches: *Nolan's approach* (Section 2.6.1), *Porter's competitive forces model* (Section 2.6.2), *Parson's approach* (Section 2.6.3), and the approach provided by *McFarlan/McKinney* (Section 2.6.4).

2.6.1 Nolan's Approach

Nolan's approach [144, 152] (cf. Fig. 2.20A) is based on three consecutive evaluation steps (cf. Fig. 2.19):

- **Step 1:** First, the objectives of an organization are identified. Three objectives are of particular relevance: (1) to improve productivity (in order to reduce the time-to-market), (2) to strengthen the market position and competitive advantage of an organization, and (3) to improve the effectiveness of management functions. All identified enterprise objectives are subordinated to one of these three major goals.

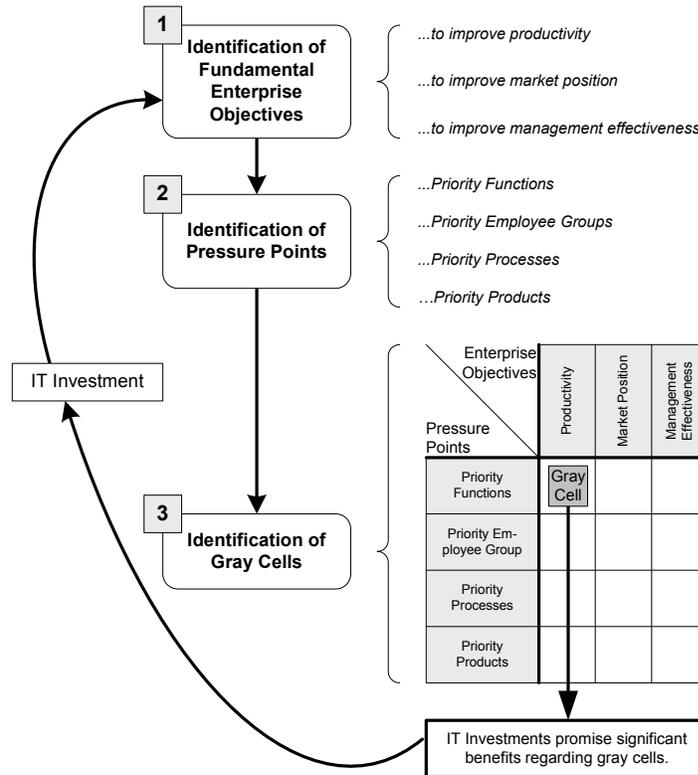


Figure 2.19: Applying Nolan's Approach.

- **Step 2:** Second, so called *pressure points* are derived. Each pressure point aggravates the achievement of one (or several) of the previously identified enterprise objectives.

In other words, a pressure point represents a critical business object that exhibits non-optimal (i.e., improvable) business performance (independent on how business performance is measured). Such critical business objects can be single business functions (*priority functions*), user groups (*priority employee groups*), business processes (*priority processes*), or even entire products (*priority products*).

- **Step 3:** In this third step, *gray cells* are identified. A gray cell describes those IT systems (e.g., an ERP system) that do not provide adequate support in the context of a specific pressure point (and that therewith hamper the achievement of the enterprise objectives). Gray cells are derived using a matrix with the pressure point as the first, and the fundamental enterprise objectives as the second dimension (cf. Fig. 2.19). Focusing on identified gray cells promises significant benefits regarding the achievement of enterprise objectives.

This approach is particularly suitable for the portfolio management of an organization, i.e., it helps to identify weak parts of an IT infrastructure and helps to guide new IT investment (by selecting an IT investment based on its suitability to address a gray cell). It can be also used to evaluate whether the introduction of IT is a reasonable investment.

2.6.2 Porter's Competitive Forces Model

Porter's *competitive forces model* [154] (cf. Fig. 2.20B) focuses on market position and competitive advantage. It is based on the analysis of five "competitive forces" which shape every industry and every market, and which determine the intensity of competition:

- **Force 1: Entry of Competitors.** This force deals with the question how easy or difficult it is to enter a market (e.g., through market barriers).
- **Force 2: Threat of Substitutes.** This force deals with the question how easy or difficult a product or service can be substituted by competing offers. A threat from substitutes will occur if there exist alternative products with lower prices and better performance parameters for the same purpose.
- **Force 3: Power of Buyers.** This force deals with the position of buyers (and their pressure on margins).
- **Force 4: Power of Suppliers.** This force deals with the questions how strong the position of suppliers is (which is likely to be high, for example, when a market is dominated by only a few large suppliers).
- **Force 5: Competition among existing Market Players.** This force deals with the questions whether there is a competition between existing market players.

The objective of corporate strategy is to modify these competitive forces in a way that improves the market position of an organization. Thereby, one distinguishes between *physical* and *informational* activities (which improve market position based on their impact on the

competitive forces). *Physical activities* embrace all activities that result in the creation or handling of a physical product (e.g., a car). *Informational activities*, by contrast, are based on experience knowledge, intuition and other intangible factors (e.g., the introduction of a new IT for automating the procurement process of an automotive manufacturer).

A Nolan's Approach		B Porter's Competitive Forces Model	
I. Classification Criteria	C-1: Evaluation Viewpoint	E-1: Plausibility	E-1: Plausibility
	Financial Viewpoint	Interpretability of Evaluation Results	Interpretability of Evaluation Results
	Work Performance Viewpoint	Transparency of Result Generation	Transparency of Result Generation
	Strategic Viewpoint	E-2: Objectiveness	E-2: Objectiveness
	C-2: Decision Support	Degree of Objectiveness	Degree of Objectiveness
	Particularly suitable for an organization's portfolio management. Can supports decisions in portfolio management.	E-3: Sensitivity	E-3: Sensitivity
	C-3: Evaluation Dimension	Error-Proneness of Evaluations	Error-Proneness of Evaluations
	Evaluation of Benefits	Resistance against Manipulation	Resistance against Manipulation
	Evaluation of Costs	E-4: Practical Applicability	E-4: Practical Applicability
	Evaluation of Risks	Flexibility	Flexibility
Evaluation of Work Performance	Efficiency (Effort)	Efficiency (Effort)	
C-4: Evaluation Scope	Data Collection & Preparation	Data Collection & Preparation	
ex-ante	Availability	Availability	
ex-post	Heterogeneity	Heterogeneity	
C-5: Evaluation Outcome	Data Quality	Data Quality	
Quantification Absolute Measures	Overall Effort	Overall Effort	
Quantification Relative Measures	Comparability	Comparability	
Visualization of Outcome	Correctness of Conclusions	Correctness of Conclusions	
Qualitative Conclusions	E-5: Theoretical Foundation	E-5: Theoretical Foundation	
	Degree of Formalization	Degree of Formalization	
	E-6: Tool Support	E-6: Tool Support	
	Simplicity of providing Tool Support	Simplicity of providing Tool Support	

Caption: x: supported o: optional ▲: positive ▼: negative ➔: neutral

Figure 2.20: Nolan's Approach and Porter's Competitive Forces Model.

Generally, Porter's approach has some weaknesses. *First*, in the economic sense, the model assumes a "perfect market". The more a market is regulated, the less meaningful insights can it deliver. *Second*, the model is particularly suitable for dealing with simple market structures. In complex market structures, however, the comprehensive description and analysis of all five forces tend to get difficult. *Third*, the model assumes static market structures (which is not the case today). *Fourth*, the model is based on the idea of competition. It assumes that organizations try to achieve competitive advantages over other market players, suppliers, and customers. With this focus, the model does not take into account strategic alliances, networked business constellations [12], or virtual enterprise-networks.

2.6.3 Parson's Approach

Parson's Approach [152] (cf. Fig. 2.22A) analyzes the strategic impact of IT along three levels: the *industry sector level* (also called *global level*), the *enterprise level*, and the *strategic level*. On the global level, IT investments influence the structure of entire industrial sectors, production environments, and markets. As an example consider the realization of cross-organizational IT solutions such as e-commerce marketplaces. On both the enterprise level and the strategic level, IT investments are analyzed with respect to their impact on Porter's competitive forces.

Parson's approach aims at supporting managers in evaluating the impact of an IT investments on the market position of an organization. Similar to Porter's competitive forces model, it not clear whether Parson's approach can be used to derive really useful evaluations. In any case, Parson's approach adopts a very broad and holistic viewpoint.

2.6.4 Approach of McFarlan/McKinney

The approach provided by McFarlan/KcKinney [114, 152] (cf. Fig. 2.22B) utilizes a *strategic grid* for assessing the achievement of strategic enterprise objectives through IT investments. This strategic grid allows for conclusions regarding the IT landscape of an organization. Therefore, any potential IT investment is assigned to one of the four quadrants of the strategic grid (cf. Fig. 2.21):

- **Low Current | Low Future Impact:** The IT investment has only little strategic relevance and only supports existing business processes.

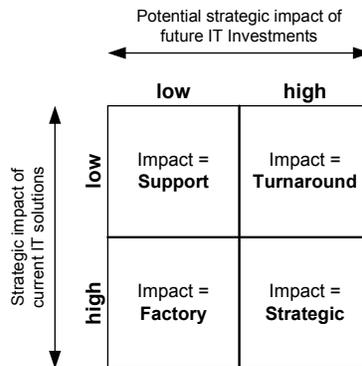


Figure 2.21: The Strategic Grid.

- **Low Current | High Future Impact:** The IT investment will have a major impact on future business models, i.e., it will be a key element of strategic planning.
- **High Current | Low Future Impact:** The IT investment is important in terms of day-to-day operations, but cannot be considered as fundamentally important. The key issue is maintaining existing IT solutions.
- **High Current | High Future Impact:** The IT investment is a success factor both for an organization's current and future strategy, i.e., it has strategic significance. Such an IT investment is success-critical in the way that an organization is not able to deliver products and services, i.e., to proceed in business, without it.
- **High Current | High Future Impact:** The IT investment is a success factor both for the current and future strategy of the organization, i.e., it has strategic significance.

Like Nolan's approach, this approach is particularly suitable to guide the portfolio management of an organization.

2.6.5 Real Option Theory

Real option theory applies *financial option theory* (e.g., the Black-Scholes option-pricing model) to IT investments and aims at quantifying the value of management flexibility in a

sider Boehm's *constructive cost model* (COCOMO) [27] (cf. Fig. 2.23A) or Putnam's *software life cycle management* (SLIM) [157] (cf. Fig. 2.23B). Algorithmic cost models combine exponential and linear scaling factors into one rating-driven cost model: $Costs = A * (Size)^B * EM$. A is an organization-dependent constant. A software system's size can be measured in *lines of code*, *functions points* or *object points*. B is an exponential scale factor to account for the relative economies or diseconomies of scale encountered as a software project increases its size (e.g., through communication overhead of large teams).

EM unifies a set of additional linear *effort multipliers* EM_i (with $EM = \sum(EM_i)$). EM usually combine attributes to further adjust the estimate obtained from calculating only with a project's size and exponential scale factors. EM_i typically cover *product attributes* (e.g., constraints and requirements placed upon the project), *platform attributes* (e.g., limitations caused by the hardware and operating system being used), *personnel attributes* (e.g., level of skills possessed by the project personnel), and *project attributes* (e.g., constraints and conditions under which project development takes place).

A Constructive Cost Model (CoCoMo)		B Software Life Cycle Management	
I. Classification Criteria	C-1: Evaluation Viewpoint	II. Evaluation Criteria	E-1: Plausibility
	Financial Viewpoint		Interpretability of Evaluation Results
	Work Performance Viewpoint		Transparency of Result Generation
	Strategic Viewpoint		
	C-2: Decision Support		E-2: Objectiveness
	Decision making is supported in the sense of project monitoring and control.		Degree of Objectiveness
	C-3: Supported Evaluations		E-3: Sensitivity
	Evaluation of Benefits		Error-Proneness of Evaluations
	Evaluation of Costs		Resistance against Manipulation
	Evaluation of Risks		E-4: Practical Applicability
	Evaluation of Work Performance		Flexibility
	C-4: Evaluation Type		Efficiency (Effort)
	ex-ante		Data Collection & Preparation
	ex-post		Availability
	C-5: Evaluation Outcome		Heterogeneity
Quantification Absolute Measures	Data Quality		
Quantification Relative Measures	Overall Effort		
Visualization of Outcome	Comparability		
Qualitative Conclusions	Correctness of Conclusions		
	E-5: Theoretical Foundation		
	Degree of Formalization		
	E-6: Tool Support		
	Simplicity of providing Tool Support		

Caption: x: supported o: optional ↑: positive ↓: negative ⇔: neutral

Figure 2.23: Algorithmic Cost Estimation Approaches.

2.7.2 Expertise-based Approaches

Expertise-based methods are particularly useful in the absence of quantifiable data. They capture the knowledge of practitioners and other experts to derive cost estimates. Typical representatives of expertise-based approaches are the *delphi technique* and the *work breakdown structure* approach.

Delphi Technique. The *delphi technique* (cf. Fig. 2.24A) guides a group of individuals to a consensus of opinion respectively estimation. In a first step, participants are individually asked to make a cost estimation for the software system to be analyzed (without consulting the other participants). Results are then collected, revised, and then returned to each participant for a second round. During the second run participants are again asked to make a

cost estimation, but this time with knowledge of what the other participants estimated in the first round. This usually results in a narrowing of estimation range and more substantial estimations. Note that there exist many variants which utilize, for example, group discussion between estimation rounds.

Work Breakdown Structure. The *work breakdown structure approach* (cf. Fig. 2.24B) reorganizes the costs of a software project into a hierarchy of cost factors. Then each cost factor is separately estimated. By assigning probabilities to all cost factors (forecasting their actual emergence), an overall cost estimation can be derived. Thereby, expertise comes into play when determining the most useful specification of the hierarchical work breakdown structure and when specifying the probabilities.

A		Delphi Technique		
I. Classification Criteria	C-1: Evaluation Viewpoint			
	Financial Viewpoint	X		
	Work Performance Viewpoint			
	Strategic Viewpoint	X		
	C-2: Decision Support			
	Decision support is only implicitly enabled.			
	C-3: Supported Evaluations			
	Evaluation of Benefits	X		
	Evaluation of Costs	X		
	Evaluation of Risks	X		
	Evaluation of Work Performance			
	C-4: Evaluation Type			
	ex-ante	X		
	ex-post			
	C-5: Evaluation Outcome			
Quantification Absolute Measures	X			
Quantification Relative Measures	X			
Visualization of Outcome				
Qualitative Conclusions				
II. Evaluation Criteria	E-1: Plausibility			
	Interpretability of Evaluation Results		↕	
	Transparency of Result Generation		↕	
	E-2: Objectiveness			
	Degree of Objectiveness		↕	
	E-3: Sensitivity			
	Error-Proneness of Evaluations		↔	
	Resistance against Manipulation		↕	
	E-4: Practical Applicability			
	Flexibility		↕	
	Efficiency (Effort)	Data Collection & Preparation		↕
		Availability		↕
		Heterogeneity		↕
		Data Quality		↕
	Overall Effort		↔	
Effectiveness	Comparability		↔	
	Correctness of Conclusions		↕	
E-5: Theoretical Foundation				
Degree of Formalization		↕		
E-6: Tool Support				
Simplicity of providing Tool Support		↕		

B		Work Breakdown Structure		
I. Classification Criteria	C-1: Evaluation Viewpoint			
	Financial Viewpoint	X		
	Work Performance Viewpoint			
	Strategic Viewpoint			
	C-2: Decision Support			
	Decision support is only implicitly enabled.			
	C-3: Supported Evaluations			
	Evaluation of Benefits			
	Evaluation of Costs	X		
	Evaluation of Risks			
	Evaluation of Work Performance			
	C-4: Evaluation Type			
	ex-ante	X		
	ex-post			
	C-5: Evaluation Outcome			
Quantification Absolute Measures	X			
Quantification Relative Measures				
Visualization of Outcome				
Qualitative Conclusions				
II. Evaluation Criteria	E-1: Plausibility			
	Interpretability of Evaluation Results		↔	
	Transparency of Result Generation		↔	
	E-2: Objectiveness			
	Degree of Objectiveness		↔	
	E-3: Sensitivity			
	Error-Proneness of Evaluations		↔	
	Resistance against Manipulation		↔	
	E-4: Practical Applicability			
	Flexibility		↕	
	Efficiency (Effort)	Data Collection & Preparation		↕
		Availability		↕
		Heterogeneity		↕
		Data Quality		↕
	Overall Effort		↔	
Effectiveness	Comparability		↔	
	Correctness of Conclusions		↕	
E-5: Theoretical Foundation				
Degree of Formalization		↕		
E-6: Tool Support				
Simplicity of providing Tool Support		↔		

Caption: x: supported o: optional ↕: positive ↘: negative ↔: neutral

Figure 2.24: Expertise-based Cost Estimation Approaches.

Expertise-based methods are useful for unprecedented software projects but encounter scalability problems, i.e., the larger a software project is the more difficult is it to derive expertise-based estimations. Their obvious drawback is that an estimate is only as good as the expert’s opinion, and there is no way to test that opinion until it will be too late to correct the damage if that opinion proves wrong.

2.7.3 Learning-oriented Approaches

Learning-oriented approaches use known outcomes of past projects when estimating costs. Typical representatives of learning-oriented approaches are *estimation by analogy* and *neural networks-based cost estimation*.

Estimation by Analogy. *Estimation by analogy* (cf. Fig. 2.25A) uses general lessons learned and estimation heuristics to extrapolate an estimate from past projects. The rule of analogy says that similar projects are likely to exhibit similar costs and schedules. Considered heuristics are, for example, (1) environmental conditions and constraints obtained

during previous software projects, (2) technical and managerial decisions, as well as (3) final successes or failures that resulted.

A		Estimation by Analogy					
I. Classification Criteria	C-1: Evaluation Viewpoint			II. Evaluation Criteria	E-1: Plausibility		
	Financial Viewpoint		X		Interpretability of Evaluation Results		⇒
	Work Performance Viewpoint				Transparency of Result Generation		⇒
	Strategic Viewpoint		X		E-2: Objectiveness		
	C-2: Decision Support				Degree of Objectiveness		⇒
	Decision support is only implicitly enabled.				E-3: Sensitivity		
	C-3: Supported Evaluations				Error-Proneness of Evaluations		⇒
	Evaluation of Benefits		X		Resistance against Manipulation		⇒
	Evaluation of Costs		X		E-4: Practical Applicability		
	Evaluation of Risks		X		Flexibility		⇕
	Evaluation of Work Performance				Availability		⇒
	C-4: Evaluation Type				Data Collection & Preparation		⇕
	ex-ante		X		Heterogeneity		⇕
	ex-post				Data Quality		⇕
	C-5: Evaluation Outcome				Overall Effort		⇕
Quantification Absolute Measures		X	Comparability		⇕		
Quantification Relative Measures		X	Correctness of Conclusions		⇒		
Visualization of Outcome			E-5: Theoretical Foundation				
Qualitative Conclusions			Degree of Formalization		⇕		
			E-6: Tool Support				
			Simplicity of providing Tool Support		⇕		

B		Neural Networks					
I. Classification Criteria	C-1: Evaluation Viewpoint			II. Evaluation Criteria	E-1: Plausibility		
	Financial Viewpoint		X		Interpretability of Evaluation Results		⇒
	Work Performance Viewpoint				Transparency of Result Generation		⇒
	Strategic Viewpoint				E-2: Objectiveness		
	C-2: Decision Support				Degree of Objectiveness		⇕
	Decision making is supported in the sense of project monitoring and control.				E-3: Sensitivity		
	C-3: Supported Evaluations				Error-Proneness of Evaluations		⇒
	Evaluation of Benefits				Resistance against Manipulation		⇒
	Evaluation of Costs		X		E-4: Practical Applicability		
	Evaluation of Risks				Flexibility		⇒
	Evaluation of Work Performance				Availability		⇕
	C-4: Evaluation Type				Data Collection & Preparation		⇕
	ex-ante		X		Heterogeneity		⇕
	ex-post		X		Data Quality		⇕
	C-5: Evaluation Outcome				Overall Effort		⇕
Quantification Absolute Measures		X	Comparability		⇕		
Quantification Relative Measures			Correctness of Conclusions		⇒		
Visualization of Outcome			E-5: Theoretical Foundation				
Qualitative Conclusions			Degree of Formalization		⇕		
			E-6: Tool Support				
			Simplicity of providing Tool Support		⇕		

Caption: X: supported O: optional ↑: positive ↓: negative ➔: neutral

Figure 2.25: Learning-oriented Cost Estimation Approaches.

Neural Networks. *Neural networks* (cf. Fig. 2.25B) can be "trained" to produce cost estimations [82, 192]. For this purpose an initial neural network is build from historical project data. Based on this "training set", the parameters of the neural network are automatically adjusted in order to reduce the delta between the historic cost estimation and the analyzed software project. Thereby, the specification of a suitable delta value is important. Without a suitable delta value, the neural network could theoretically become "overtrained". This means that its estimation is adjusted until it is very good at emulating the costs for the used training project, but cannot be applied to the new and actually to be analyzed software project (as it has become too specific). The most common form of a neural network used in the context of software estimation is a "back propagation trained feed-forward" network [82]. However, as large and accurate data sets are needed to accurately train neural networks, this approach is hardly used in practice as adequate training data is often missing.

2.7.4 Regression-based Approaches

Regression-based techniques apply regression models to derive cost estimations. As typical examples consider the *ordinary least square* (OLS) method and the *robust regression*.

Ordinary Least Square. The *ordinary least square* (OLS) (cf. Fig. 2.26A) uses standard linear regression [86]. Regression will be called "linear" if the relation of the response (i.e., the cost estimation) to the explanatory variables (i.e., the cost factors) is assumed to be a linear function. Based on historic project data, the OLS method attempts to find a function which closely approximates this data (a "best fit"). Thereby, it particularly attempts to minimize the sum of the squares of the ordinate differences (called *residuals*) between points

generated by the function and corresponding points in the historic project data. Mathematically, the OLS method is based on the following formula: $y = \alpha + \beta x + \epsilon$. The right side generally comprises a linear combination of parameters, here denoted α and β . The term ϵ represents the unpredicted or unexplained variation in the response variable; it is conventionally called the "error" whether it is really a measurement error or not, and it is assumed to be independent of x . The expected value of the error term is typically assumed to be equal to zero (a nonzero expected value could be absorbed into α).

Robust Regression. Alternatively, *robust regression* (cf. Fig. 2.26B) can be used as well. Doing so, the common problem of outliers in observed cost data can be mitigated.

A		Least Ordinary Square Method			
I. Classification Criteria	C-1: Evaluation Viewpoint				
	Financial Viewpoint	X			
	Work Performance Viewpoint				
	Strategic Viewpoint				
	C-2: Decision Support				
	Supports decision making. Allows to assess direct and indirect costs related to the purchase of an investment.				
	C-3: Supported Evaluations				
	Evaluation of Benefits				
	Evaluation of Costs	X			
	Evaluation of Risks				
	Evaluation of Work Performance				
	C-4: Evaluation Type				
	ex-ante	X			
	ex-post				
	C-5: Evaluation Outcome				
	Quantification Absolute Measures	X			
	Quantification Relative Measures				
	Visualization of Outcome				
	Qualitative Conclusions				
II. Evaluation Criteria	E-1: Plausibility				
	Interpretability of Evaluation Results				↑
	Transparency of Result Generation				↑
	E-2: Objectiveness				
	Degree of Objectiveness				↑
	E-3: Sensitivity				
	Error-Proneness of Evaluations				⇒
	Resistance against Manipulation				⇒
	E-4: Practical Applicability				
	Flexibility				⇒
	Efficiency (Effort)	Data Collection & Preparation	Availability	⇒	
			Heterogeneity	⇒	
			Data Quality	⇒	
			Overall Effort	⇒	
	Effectiveness	Comparability	Correctness of Conclusions	⇒	
				↑	
	E-5: Theoretical Foundation				
	Degree of Formalization				↑
	E-6: Tool Support				
Simplicity of providing Tool Support				⇒	

B		Robust Regression			
I. Classification Criteria	C-1: Evaluation Viewpoint				
	Financial Viewpoint	X			
	Work Performance Viewpoint				
	Strategic Viewpoint				
	C-2: Decision Support				
	Supports decision making. Allows to assess direct and indirect costs related to the purchase of an investment.				
	C-3: Supported Evaluations				
	Evaluation of Benefits	X			
	Evaluation of Costs	X			
	Evaluation of Risks				
	Evaluation of Work Performance				
	C-4: Evaluation Type				
	ex-ante	X			
	ex-post				
	C-5: Evaluation Outcome				
	Quantification Absolute Measures				
	Quantification Relative Measures				
	Visualization of Outcome				
	Qualitative Conclusions	X			
II. Evaluation Criteria	E-1: Plausibility				
	Interpretability of Evaluation Results				↑
	Transparency of Result Generation				↑
	E-2: Objectiveness				
	Degree of Objectiveness				↑
	E-3: Sensitivity				
	Error-Proneness of Evaluations				⇒
	Resistance against Manipulation				⇒
	E-4: Practical Applicability				
	Flexibility				⇒
	Efficiency (Effort)	Data Collection & Preparation	Availability	⇒	
			Heterogeneity	⇒	
			Data Quality	⇒	
			Overall Effort	↓	
	Effectiveness	Comparability	Correctness of Conclusions	↑	
				↑	
	E-5: Theoretical Foundation				
	Degree of Formalization				↑
	E-6: Tool Support				
Simplicity of providing Tool Support				⇒	

Caption: x: supported o: optional ↑: positive ↓: negative ⇒: neutral

Figure 2.26: Regression-oriented Cost Estimation Approaches.

2.7.5 Composite Approaches

Composite approaches combine two or more cost estimation techniques to derive a cost estimation. Typical representatives of composite approaches are *bayesian cost analysis* and *function point analysis* (FPA).

Bayesian Cost Analysis. *Bayesian cost analysis* (cf. Fig. 2.27A) is based on inductive reasoning [46, 58]. Thereby, expert-judgement (e.g., about the costs of a software project) is combined with sample data (e.g., from an already completed software project) in a logically consistent manner in order to produce a cost estimation. Bayes' theorem is used to transform initial values for the parameters of a cost model into a posterior distribution. This transformation can be considered as a learning process. The posterior distribution is determined by the variances of the expert-judgment and the sample data. If the variance of the expert-judgment is smaller than the variance of the sample data, then a higher weight is assigned to the expert-judgment. On the other hand, if the variance of the sample data is smaller than the variance of the expert-judgment, then a higher weight is assigned to the

sample data causing the posterior estimate to be closer to the sample data.

Function Point Analysis. The *function point analysis* (cf. Fig. 2.27B) measures the size of software systems [65]. More precisely, it measures the amount of functionality that is considered as being relevant (independent on the programming language, development methodology, software technologies or capabilities of the project team). Functionality (or size) is expressed by "function points" (FP). One FP is defined as one end-user business function. Each business function is specified based on its outputs, inquiries, inputs, used files, and interfaces. The overall number of estimated FP can then be used to estimate software development or project costs, to estimate the annual maintenance costs of a software system, or to determine project productivity (after completion of the project).

A		Bayesian Analysis	
I. Classification Criteria	C-1: Evaluation Viewpoint		
	Financial Viewpoint	X	
	Work Performance Viewpoint		
	Strategic Viewpoint		
	C-2: Decision Support		
	Decision making is supported in the sense of project monitoring and control.		
	C-3: Supported Evaluations		
	Evaluation of Benefits		
	Evaluation of Costs	X	
	Evaluation of Risks		
Evaluation of Work Performance			
C-4: Evaluation Type			
ex-ante	X		
ex-post			
C-5: Evaluation Outcome			
Quantification Absolute Measures	X		
Quantification Relative Measures			
Visualization of Outcome			
Qualitative Conclusions			
II. Evaluation Criteria	E-1: Plausibility		
	Interpretability of Evaluation Results		↑
	Transparency of Result Generation		↑
	E-2: Objectiveness		
	Degree of Objectiveness		↑
	E-3: Sensitivity		
	Error-Proneness of Evaluations		↑
	Resistance against Manipulation		→
	E-4: Practical Applicability		
	Flexibility		↑
Efficiency (Effort)		↑	
Data Collection & Preparation		↑	
Availability		→	
Heterogeneity		→	
Data Quality		→	
Overall Effort		→	
Effectiveness		→	
Comparability		→	
Correctness of Conclusions		↑	
E-5: Theoretical Foundation			
Degree of Formalization		→	
E-6: Tool Support			
Simplicity of providing Tool Support			↑

B		Function Point Analysis	
I. Classification Criteria	C-1: Evaluation Viewpoint		
	Financial Viewpoint	X	
	Work Performance Viewpoint		
	Strategic Viewpoint		
	C-2: Decision Support		
	Decision making is implicitly supported by using the number of FP as decision baseline.		
	C-3: Supported Evaluations		
	Evaluation of Benefits		
	Evaluation of Costs	X	
	Evaluation of Risks		
Evaluation of Work Performance			
C-4: Evaluation Type			
ex-ante	X		
ex-post			
C-5: Evaluation Outcome			
Quantification Absolute Measures	X		
Quantification Relative Measures			
Visualization of Outcome			
Qualitative Conclusions			
II. Evaluation Criteria	E-1: Plausibility		
	Interpretability of Evaluation Results		↑
	Transparency of Result Generation		→
	E-2: Objectiveness		
	Degree of Objectiveness		→
	E-3: Sensitivity		
	Error-Proneness of Evaluations		→
	Resistance against Manipulation		→
	E-4: Practical Applicability		
	Flexibility		↑
Efficiency (Effort)		↑	
Data Collection & Preparation		↑	
Availability		→	
Heterogeneity		→	
Data Quality		→	
Overall Effort		↓	
Effectiveness		↑	
Comparability		↑	
Correctness of Conclusions		↑	
E-5: Theoretical Foundation			
Degree of Formalization		↑	
E-6: Tool Support			
Simplicity of providing Tool Support			↑

Caption: X: supported O: optional ↑: positive ↓: negative →: neutral

Figure 2.27: Composite Approaches.

2.8 Other Approaches

There are other approaches that address IT from a value-based perspective. The goal of the approaches presented in this section is to develop fundamental knowledge and practical techniques to increase the value created over time by software and IT projects and products.

2.8.1 Value-based Software Engineering

Value-based Software Engineering (VBSE) integrates value considerations into software engineering principles and practices [28]. Seven key elements constitute the foundations for VBSE [30, 31]: (1) *benefits realization analysis*, (2) *stakeholder value proposition elicitation and reconciliation*, (3) *business case analysis*, (4) *continuous risk and opportunity management*, (5) *concurrent system and software engineering*, (6) *value-based monitoring and control*, and (7) *change as opportunity*.

To be able to cover all relevant aspects of a software system's life cycle various *value-based sub-models* are introduced. As an example consider the sub-model *value-based requirements reengineering* (VBRE), which includes principles and practices for identifying a system's success-critical stakeholders, for eliciting their value propositions with respect to the software system, and to reconcile these value propositions into a set of system objectives. The *value-based architecting* (VBA) sub-model, as second example, involves the reconciliation of a software system's objectives with achievable architectural solutions. The *value-based design and development* (VBDD) sub-model, as a last example, involves techniques for ensuring that the software system's objectives and value considerations are inherited by the software's design and development.

Due to the enormous number of integrated concepts VBSE is still in a conceptual stage. In particular, there exist no VBSE best practices. Consequently, it is hardly possible to transform VBSE into practice.

2.8.2 e3-value Framework

The *e3-value Framework* is a requirements engineering method that is based on analyzing e-commerce initiatives through stakeholder-based viewpoints [68, 69, 70]. Its goal is to derive and analyze multi-enterprise relationships, business cases and requirements.

The framework defines three evaluation perspectives, each of them representing one baseline to evaluate stakeholder's interests and to derive suitable requirements. The *business value viewpoint* focuses on economic value creation, distribution and consumption in multi-actor networks. It allows to set up a prediction of revenues and expenses, based on exchanges of valuable goods and services between multiple actors. The *business process viewpoint* puts the previous value viewpoint into operation in terms of business processes, i.e., it deals with the operational execution of business processes. Finally, the *IS viewpoint* focuses on information systems and their process support.

2.8.3 Value-based IT Alignment

The VITAL framework [209] (*Value-based IT Alignment*) investigates the problem of aligning IT services to business requirements [194]. It builds upon work in value-based requirements engineering [11, 70] (cf. Section 8.2) and ICT architecture design [206, 218].

With the advent of networked business constellations [225], this problem even gets a new dimension. In such networks, there is not a single decision point about IT support and different actors in the network may have conflicting requirements that need to be coordinated [26, 195]. Each actor has the goal to act in a profitable way.

2.9 Discussion

We have introduced existing (economic-driven) IT evaluation and software cost estimation approaches. However, none of the discussed approaches is able to deal with the specific challenges which arise in the context of PAIS engineering projects [135] (cf. Section 1.2).

Hence, other approaches are needed for evaluating (economic) effects related to PAIS and PAIS engineering projects. So far, focus has been on analyzing the impact of WfMS on business process performance. Oba et al. [146], for example, investigate the introduction of WfMS and particularly focus on the identification of factors that influence work efficiency, processing time, and business process standardization. A mathematical model is provided for predicting the reduction rate of processing times. An extension is the work of Reijers and van der Aalst [163, 165]. They use process simulation to compare pre- and post-implementations of information systems that rely on WfMS. Focus is on analyzing business process performance based on criteria such as lead time, waiting time, service time, and utilization of resources. In most cases, the use of workflow technology has resulted in a significant decrease of lead and service time.

Choenni et al. [45] present a model to measure the added value of WfMS to business processes. This model builds upon different *performance criteria*, i.e., parameters of a business process that are affected by the introduction of a WfMS (such as speed, quality, flexibility, and reliability). Aiello [3] introduces a measurement framework for evaluating workflow performance. The framework is defined in an abstract setting to enable generality and to ensure independence from existing WfMS. Becker et. al [19] introduce a framework to identify those processes that can be supported by a WfMS in a "profitable" way. Their framework can serve as guideline for evaluating processes during the selection and introduction of a WfMS. It contains three groups of criteria: technical, organizational and economic criteria. Designed as a scoring model, their approach enables users to systematically determine those business processes that can be automated using a WfMS.

A different approach is proposed by Abate et al. [1]. They introduce a novel measurement language to evaluate the performance of automated business processes: the "workflow performance query language" (WPQL). This language allows to define metrics independent from a specific workflow implementation.

However, these approaches also do not take into account the numerous evaluation factors, their causal dependencies and their interactions which arise in the context of PAIS engineering projects. What is particularly not addressed is the investigation of PAIS from a *systemic viewpoint*. In this context, "systemic" means that the introduction of PAIS is typically taking place in enterprise environments comprising organizational, project-specific and technological parts. These parts interact with each other and can result in complex effects which either hamper or facilitate the success⁴ of a PAIS engineering project. Together, these parts build a *system* "PAIS engineering project".

2.10 Summary

This chapter has discussed IT evaluation approaches and software cost estimation methods. IT evaluation approaches have been presented along three viewpoints: financial viewpoint, process viewpoint, and strategy viewpoint. Regarding software cost estimation approaches,

⁴"Success" is typically evaluated based on compliance with project key performance indicators such as project costs, project schedules, or resource allocation.

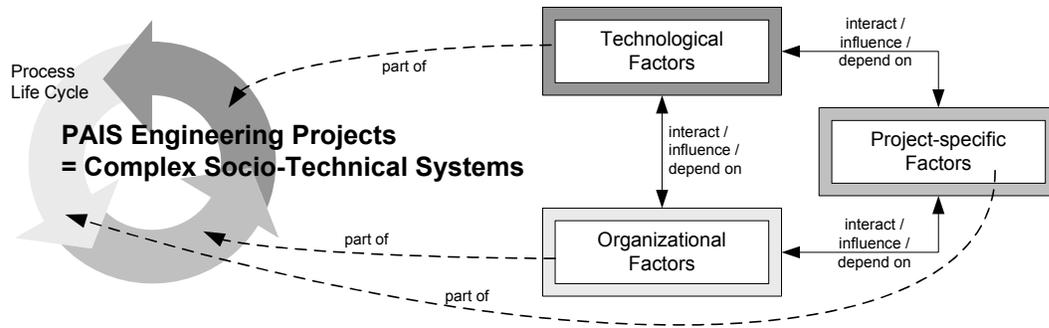


Figure 2.28: Systemic Evaluation of PAIS Engineering Projects.

we have distinguished between six major categories: algorithmic-based approaches, expertise-based approaches, learning-oriented approaches, regression-based approaches, composite approaches, and dynamic-based approaches.

Fig. 2.29 summarizes the strengths and weaknesses of all discussed approaches at a glance. Finally, we have discussed the applicability of the presented approaches in the context of PAIS engineering projects. We have concluded that existing approaches neglect the investigation of PAIS from a systemic viewpoint. However, this is necessary in order to analyze the complex effects in PAIS engineering projects.

The Big Picture		Economic-driven IT Evaluation Approaches																												
		Financial Viewpoint								Process Viewpoint		Strategic Viewpoint			Software Cost Estimation Techniques															
		Return On Investment (ROI)	Payback Period	Accounting Rate of Return	Breakeven Analysis	Net Present Value (NPV)	Internal Rate of Return (IRR)	Zero Base Budgeting	Cost Effectiveness Analysis	Total Cost of Ownership (TCO)	Target Costing	Times Savings Times Salary Model	Hedonic Wage Model	Activity-based Costing	Business Process Intelligence	Nolan's Approach	Porter's Competitive Forces Model	Parson's Approach	Approach of McFarlan and McKenney	Real Option Theory	Constructive Cost Model (COCOMO)	Software Life Cycle Management (SLIM)	Delphi Technique	Work Breakdown Structure	Estimation by Analogy	Neural Networks	Ordinary Least Square	Robust Regression	Bayesian Analysis	Function Point Analysis (FPA)
I. Classification Criteria	Evaluation Viewpoint																													
	Financial Viewpoint	X	X	X	X	X	X	X	X	X			O	O					X					X	X					
	Work Performance Viewpoint										X	X	X	X																
	Strategic Viewpoint														X	X	X	X	X			X	X	X	X	X	X	X	X	X
	Evaluation Dimension																													
	Evaluation of Benefits	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	Evaluation of Costs	X	X	X	X	X	X	X	X	X			X	X			X			X		X	X	X	X	X	X	X	X	X
	Evaluation of Risks	O	O	O	O	O	O									X						X						X		
	Evaluation of Work Performance											X	X	X	X						X						X			
	Evaluation Scope																													
ex-ante	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
ex-post	X	X	X	X								X	X		X					X							X			
Evaluation Outcome																														
Quantification Absolute Measures	X	X		X	X		X	X	X		X	X	X	X					X					X	X					
Quantification Relative Measures			X			X							X						X					X	X					
Visualization of Outcome		O		O			X		X		X	X	X						X				X	X						
Qualitative Conclusions								X	X						X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
II. Evaluation Criteria	Plausibility																													
	Interpretability of Evaluation Results	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
	Transparency of Result Generation	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
	Objectiveness																													
	Degree of Objectiveness	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
	Sensitivity																													
	Error-Proneness of Evaluations	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
	Resistance against Manipulation	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
	Practical Applicability																													
	Flexibility	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
	Efficiency (Effort)	Data Collection & Preparation	Availability	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
			Heterogeneity	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
			Data Quality	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	
Effectiveness	Comparability	Overall Effort	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔		
		Correctness of Conclusions	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔		
Theoretical Foundation																														
Degree of Formalization	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔		
Tool Support																														
Simplicity of providing Tool Support	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔	➔		

Caption: x: supported o: optional ⬆: positive ⬇: negative ➔: neutral

Figure 2.29: The Big Picture.

Chapter 3

Requirements Analysis

3.1 Motivation

The EcoPOST framework aims at the analysis of PAIS engineering projects from a systemic viewpoint. We consider PAIS engineering projects as complex socio-technical systems which are characterized by complex causal dependencies between cost and impact factors. Focus is on analyzing these causal dependencies and on investigating the impact of these dependencies on the costs of PAIS engineering projects.

Before we introduce the EcoPOST framework in detail, this chapter summarizes important requirements to be considered when evaluating PAIS engineering projects [126]. Thereby, we distinguish between requirements for evaluating IT in general and specific requirements related to the analysis of PAIS engineering projects.

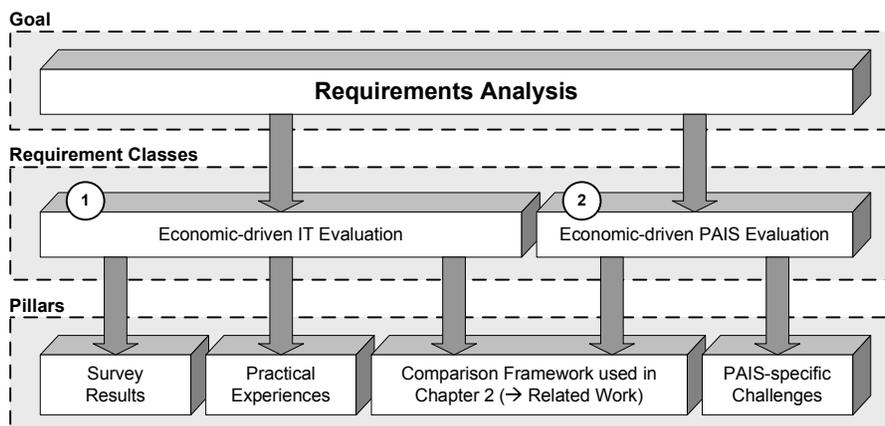


Figure 3.1: Deriving Requirements.

The remainder of this chapter is organized as follows. Section 3.2 describes general requirements for evaluating IT. These requirements are derived based on three pillars (cf. Fig. 3.1): results of our first online survey (cf. Section 1.1), practical experiences from the automotive domain, and criteria from the presented comparison framework (cf. Section 2.3.2).

Taking into account PAIS-specific evaluation challenges (cf. Section 1.3), Section 3.3 discusses specific requirements for the design of an evaluation framework for PAIS engineering projects. Section 3.4 concludes with a summary.

3.2 General Requirements

Fig. 3.2 shows five requirements we have identified as being fundamental when performing economic-driven IT evaluations. These requirements reflect major wishes and needs of stakeholders who are concerned with economic-driven IT evaluations such as (IT) controllers, project managers, policy makers, and senior managers. Identified requirements are derived based on results of our first survey (cf. Section 1.1) and on practical needs we observed in the automotive domain (cf. Fig. 3.1).

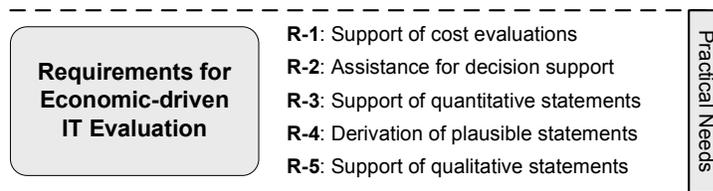


Figure 3.2: Requirements for Economic-driven IT Evaluation.

According to our initial online survey (cf. Section 1.1 for details), 72% of the 79 survey respondents confirm that the results of a business case significantly influence IT investment decisions. More specifically, 28% of the respondents state that a business case is *always* developed (cf. Fig. 3.3A). 43% of the participants point out that a business case is developed in most cases (i.e., *mostly*).

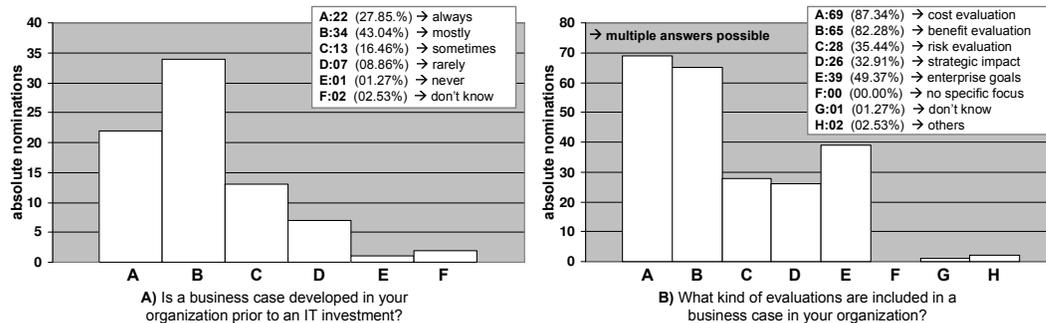


Figure 3.3: Developing a Business Case.

Thereby, 87.43% of the respondents consider *costs evaluations* as the most important part of a business case (cf. Fig. 3.3B). *Benefit evaluations* (82.28%) are also recognized as important. The *contribution of an IT investment to organizational goals* (49.37%), *risks* (35.44%) and the *impact of an IT investment on the overall enterprise strategy* (32.91%) are only considered as being of minor relevance.

These survey results do not only confirm the necessity to perform economic-driven IT evaluations, but also allow us to identify a first requirement for economic-driven IT evaluation.

R-1: *An economic-driven IT evaluation approach must support cost evaluation.*

Our practical experiences confirm this requirement, i.e., economic-driven IT evaluations need to be cost-driven. This even applies to corresponding benefits which are often quantified by means of "decreased costs" or "increased revenues". In fact, due to an immense cost pressure, a significant change has occurred regarding the understanding of "value". IT innovation, not long ago one major dynamic driver of both business and technology evolution, is no longer considered as an important value driver, but mainly as source of high costs. Thus, policy makers are increasingly demanding for tools and methods supporting them in decision making.

In this context, financial business ratios adopt a key role. In our survey, the majority of the 79 participants (55.70%) confirm the use of the *return on investment* ratio, directly followed by *organization-specific business ratios* (53.16%). Besides, *total cost of ownership* approaches and *break even analysis* are popular as well (with disseminations of 37.97% and 25.32%). Other business ratios we assumed as being intensively used (e.g., the *net present value* or the *internal rate of return*) have turned out to be of minor relevance. Approaches additionally mentioned include the *payback period* and the *shareholder value added*.

All these business ratios integrate available data (e.g., about the costs of an IT investment or the cash flow generated by it) into one measure in order to support decision making. The popularity of financial business ratios indicates that supporting decision making is important. Note that decision making is addressed in the comparison framework from Chapter 2 as well (cf. Criterion C-2).

R-2: *An economic-driven IT evaluation approach must support decision making.*

Another basic objective of financial business ratios is to indicate either positive or negative economic effects. As a matter of fact there is an increasing pressure to prove the added value of IT investments. 79.75% of the survey participants (cf. Fig. 3.4A) state that the need to prove the positive ROI of IT investments has increased within the last 12 months *strongly* (40.51%) or *rather strongly* (39.24%).

Thereby, it is no longer sufficient to rely on rough estimations. Instead, it becomes necessary to quantify the added value for an organization through IT investments as precisely as possible (cf. Fig. 3.4B).

R-3: *An economic-driven IT evaluation approach must support quantitative statements.*

In response to this requirement, more and more IT suppliers (e.g., hardware suppliers, software vendors, service providers) publish economic measures to substantiate the potential economic impact of their products (e.g., ROI measures). However, in most cases the correctness and plausibility of such data is neither understandable nor traceable. Thus, it is not surprising that most survey respondents (74.68%) consider such vendor information as not

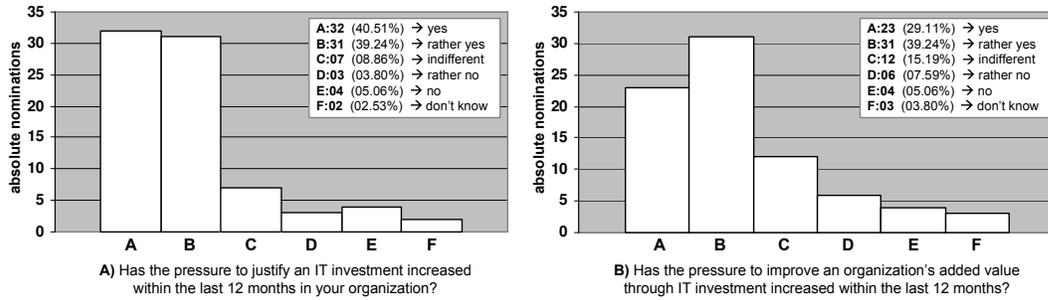


Figure 3.4: Pressure for Economic Justification.

trustworthy (cf. Fig. 3.5A). Consequently, 72.15% of the survey participants state that they do not incorporate vendor data when making investment decisions (cf. Fig. 3.5B). Anyhow, still 13.92% do in fact consider vendor data.

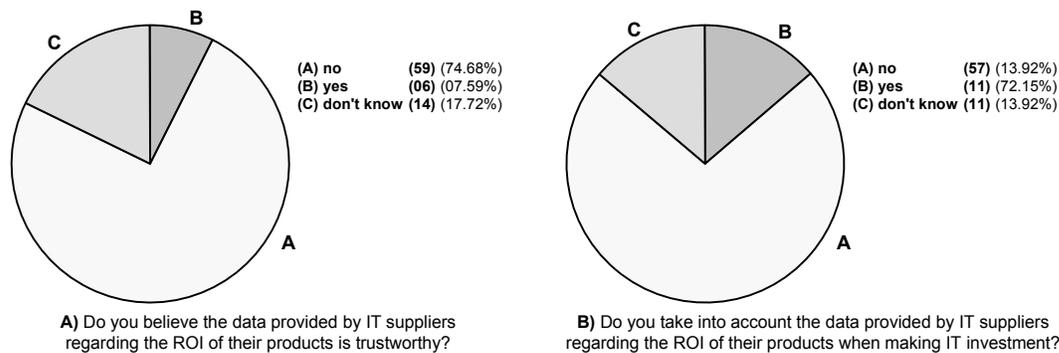


Figure 3.5: Trustworthiness of Vendor Data.

Taking survey results, we can conclude that deriving plausible evaluations is a key requirement. This issue has also been discussed in the comparison framework we used in Chapter 2 (cf. Criterion E-1).

R-4: *An economic-driven IT evaluation approach must allow for plausible statements.*

In practice, many IT evaluations lack a sufficient degree of plausibility. The predominant majority of the survey participants (82.28%) state that the *quantification of benefits* is particularly difficult (cf. Fig. 3.6). This has been clearly confirmed by our practical experiences. Besides there are other difficulties, e.g., the *identification of all relevant cost drivers* (45.57%), the *quantification of risks* (39.24%), and *uncertainties regarding the period over which an IT investment will be used* (27.85%). This last issue is further hampered by the fact that IT investments may not yield immediate returns and often exhibit a latency in value realization [67].

Another difficulty, however, is not confirmed as expected by the survey: the *missing understanding of the impacts of an IT investment*. Indeed, 21.52% of the participants state

this would be a problem, but we expected a significantly higher percentage here. The reason for this may be that traditional software engineering is increasingly replaced with novel engineering principles that inhere a high degree of complexity which, in turn, aggravates the understanding (and therewith the evaluation) of innovative IT projects. Our survey, however, indicates that understanding the impacts of IT investment is not a big problem.

Likewise, our survey indicates that the *availability* and *complexity* of business ratios is not considered as problematic (cf. Fig. 3.6). 17.72% of the participants state that available business ratios do not adequately support the derivation of conclusions. Finally, only 11.39% consider the complexity of business ratios as critical.

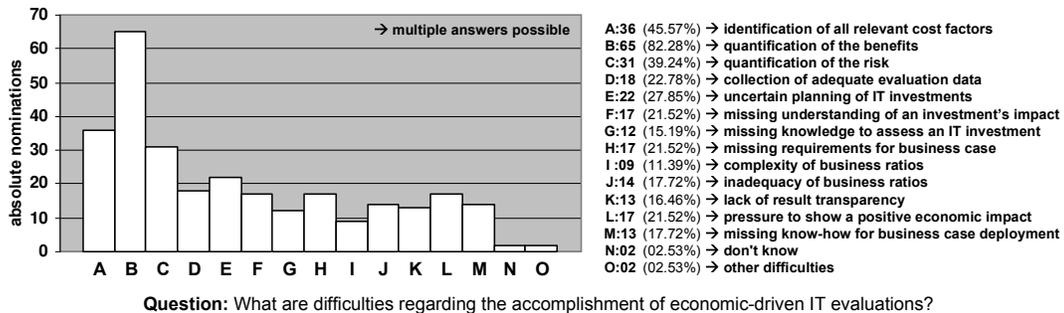


Figure 3.6: Difficulties when Accomplishing Economic-driven IT Evaluations.

Considering these difficulties, it becomes often impossible to quantify all aspects of IT investments. In fact, even if it can be managed to quantify some aspects, correspondent conclusions are – for most cases – not precise and afflicted with uncertainties. In response to this problem, we can identify another requirement.

R-5: *Besides the derivation of quantitative statements, economic-driven IT evaluation approaches must allow for qualitative statements as well.*

Altogether, survey results show that the economic justification of IT investments is important, but a difficult task to accomplish. However, the identified requirements are still very broad. Hence, the following section describes additional specific requirements for evaluating PAIS engineering projects.

3.3 Specific Requirements

The requirements we have discussed in the previous section are general requirements for evaluating IT from an economic-driven viewpoint. In the following, we describe six additional, specific requirements for evaluating PAIS engineering projects (cf. Fig. 3.7). These requirements are derived based on two pillars: (1) PAIS-specific challenges described in Section 1.2, and (2) criteria from the comparison framework we introduced in Section 2.3.2 for classifying and comparing existing IT evaluation approaches.

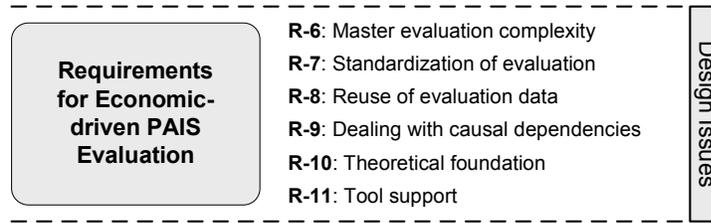


Figure 3.7: Requirements for the Economic-driven Evaluation of PAIS.

Ensuring Practical Applicability. Evaluating PAIS is complex, e.g., due to emergence of intangible impact factors, causal dependencies between cost and impact factors, and complex effects caused by these interdependencies. An approach for evaluating PAIS engineering projects has to master this complexity.

R-6: *An economic-driven evaluation approach for PAIS engineering projects must allow to incorporate and to manage the complexity of PAIS engineering projects.*

Further, it is important that a solution approach enables users to perform evaluations in a standardized way, i.e., the methodical steps to be executed need to be clearly defined.

R-7: *An economic-driven evaluation approach for PAIS engineering projects must allow for performing an evaluation in a standardized manner.*

Note that the requirements R-6 and R-7 are major prerequisites to foster the applicability of an evaluation approach (cf. Criterion E-4 in our comparison framework from Chapter 2).

An evaluation results in many efforts, e.g., for collecting needed data and preparing evaluation results. In the comparison framework from Chapter 2, this issue is addressed as sub-criterion E-4. In order to reduce the efforts related to an evaluation, the reuse of evaluation data should be possible.

R-8: *An economic-driven evaluation approach for PAIS engineering projects should support the reuse of evaluation data.*

Conceptual Requirements. In Section 1.2, we have pointed out that the economics of PAIS engineering projects are affected by numerous technological, organizational and project-specific factors, and that there exist many causal dependencies between these factors. Moreover, these dependencies result in complex economic effects which are difficult to identify and trace. Hence, in order to fully understand the costs of PAIS engineering projects, we have to identify, analyze, and understand causal dependencies and their effects.

R-9: *An economic-driven evaluation approach for PAIS engineering projects must provide means to describe and analyze causal dependencies between organizational, project-specific, and technological factors.*

In order to handle this requirement, one needs a precise and constant notation stating formal rules and classes of expressions. More precisely, we must specify a set of perceptions (i.e., abstractions) and actions which are registered and performed in the same way by all users

of an evaluation approach. This helps to ensure transparency and traceability of evaluations and to enhance plausibility of evaluation outcomes. Note that this theoretical foundation of an evaluation approach is also addressed in our comparison framework from Section 2.3.2 (cf. Criterion E-5).

R-10: *An economic-driven evaluation approach for PAIS engineering projects must have a theoretical foundation in order to enable the consistent and traceable derivation of an evaluation statement.*

Finally, as already discussed in Section 2.3.2, we identify the need to provide tool support for an evaluation approach as important requirement. This is necessary in order to facilitate the application of an evaluation approach in practice (cf. Criterion E-6).

R-11: *An economic-driven evaluation approach for PAIS engineering projects must include tool support for performing evaluations.*

Note that requirement R-11 is a particularly critical requirement as the application of many methodologies fails due to a lack of tool support.

Derivation of Requirements. Finally, Fig. 3.8 shows which requirements have been derived based on which source of information.

		General Requirements					Specific Requirements					
		R-1	R-2	R-3	R-4	R-5	R-6	R-7	R-8	R-9	R-10	R-11
Pillars	Survey Results	✓	✓	✓	✓	✓						
	Practical Experiences	✓	✓	✓	✓	✓						
	Comparison Framework (Related Work)		✓		✓				✓	✓	✓	✓
	PAIS-specific Challenges						✓	✓	✓	✓	✓	✓

Figure 3.8: Derivation of Requirements: Four Pillars.

Other Requirements. When evaluating PAIS, there may exist further requirements. As examples consider compliance with *workflow patterns* [205], *change patterns* [214] or change features [168, 66]. Besides, the quantification of benefits and risks of PAIS engineering projects may be a requirement as well. However, focus of this work is on investigating the causal dependencies of cost and impact factors and related (cost) effects. Thus, we only consider the eleven requirements described above.

3.4 Summary

Figure 3.9 summarizes the presented requirements at a glance. First, we have described general requirements for evaluating IT. These requirements have been derived based on results of our first online survey, practical experiences gathered in the automotive domain, and criteria we have encountered as particularly important from the presented comparison framework in Chapter 2. Taking into account the PAIS-specific evaluation challenges described in Section 1.3, we have discussed specific requirements for our solution approach.

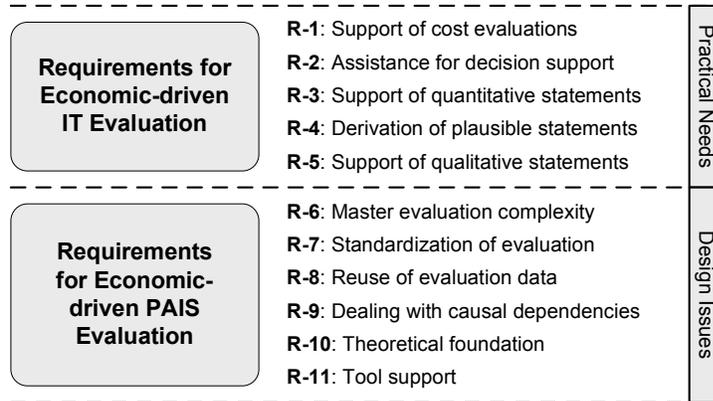


Figure 3.9: Identified Requirements at a Glance.

Chapter 12 will discuss which of these requirements are either fully or only partly addressed by our solution approach.

Part II

The EcoPOST Framework

Chapter 4

The EcoPOST Methodology

4.1 Motivation

PAIS engineering projects are characterized by a variety of organizational, project-specific, and technological cost and impact factors. While many of these factors particularly emerge in PAIS engineering projects, others also arise in projects concerning the development of function- and data-centered IS. Between these impact factors, there exist numerous causal dependencies which make PAIS engineering projects even more complex.

Though economic-driven IT evaluation and software cost estimation have received considerable attention during the last decades (cf. Chapter 2) – and have become an essential task in IS engineering – it is difficult to apply existing techniques to PAIS engineering projects (cf. Section 2.9). Picking up this problem, this chapter introduces the EcoPOST methodology, a comprehensive evaluation approach which supports PAIS engineers in understanding causal dependencies and resulting effects. Focus is on intangible impact factors and their effects on the costs of PAIS engineering projects.

The remainder of this chapter is organized as follows. Section 4.2 introduces basic EcoPOST terminology. Section 4.3 gives an overview of the EcoPOST methodology. Section 4.4 introduces a running example we use in the following to illustrate the main methodical steps of the EcoPOST methodology. These methodical steps are discussed in the following Sections 4.5 to 4.10. We present a basic set of impact factors, introduce the notion of evaluation models to investigate causal dependencies and related effects, and motivate the need to simulate evaluation models in order to unfold the dynamic implications of causal dependencies. Section 4.11 concludes with a summary.

4.2 Basic Terminology

In Section 2.2, we have already discussed basic terminology in the field of economic-driven IT evaluation. Considering our main evaluation goal, namely, to investigate causal dependencies and interactions between technological, organizational, and project-specific impact factors and their impact on the costs of PAIS engineering projects, these existing terms are not sufficient. In particular, they are too restricted and do not allow to address the spe-

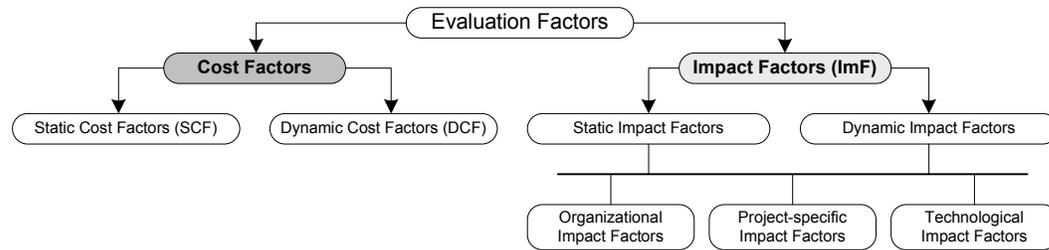


Figure 4.1: Terminology in the EcoPOST Framework.

cific characteristics of PAIS engineering projects we want to investigate (cf. Sections 1.2 and 2.9). Hence, we introduce the following EcoPOST-specific terminology (cf. Fig. 4.1), which is more useful in our context:

- **Cost Factors:** Represent costs that can be quantified in terms of money. Thereby, we distinguish between two kinds of cost factors:
 - *Static Cost Factors* (SCF) represent costs whose value does not change during a PAIS engineering project (except for their time value, which is not further considered in the following). As typical examples of SCFs consider software license costs, hardware costs, or costs for external consultants.
 - *Dynamic Cost Factors* (DCF), in turn, represent costs that are determined by activities related to a PAIS engineering project. The (re)design of business processes prior to the introduction of PAIS, for example, constitutes such an activity. As another example consider the performance of interview-based process analysis. These activities cause measurable efforts, which, in turn, vary due to the influence of intangible impact factors. A corresponding DCF "Costs for Business Process Redesign" may be influenced, for instance, by an intangible factor "Willingness of Staff Members to support Process (Re)Design Activities". Obviously, if staff members do not contribute to a (re)design project by providing needed information (e.g., about process details), any redesign effort will be ineffective and will result in increasing (re)design costs. If staff willingness is additionally varying during the (re)design activity (e.g., due to a changing communication policy), the DCF will be subject to even more complex effects. In our framework, intangible factors like "Willingness of Staff Members to support (Re)Design Activities" are represented by so called *impact factors*.
- **Impact Factors (ImF)** are intangible factors that influence DCFs and other ImFs. We distinguish between organizational, project-specific, and technological ImFs. ImFs cause the value of DCFs and dynamic ImFs to change, making their evaluation a difficult task to accomplish. As examples consider factors such as "End User Fears", "Availability of Process Knowledge", or "Ability to (re)design Business Processes". Opposed to SCFs and DCFs, the values of ImFs are not quantified in monetary terms. Instead, they are specified based on qualitative scales describing the degree of an ImF

ranging from "low" to "high". Such quantifications are also known from software cost estimation models such as Boehm's *constructive cost model* (COCOMO) [27, 29] (cf. Section 2.7.1). Like cost factors, ImFs can be further classified into *static* and *dynamic* ImFs:

- *Static Impact Factors*: Static ImFs do not change, i.e., they are assumed to be constant during a PAIS engineering project (e.g., certain degrees of user fears, process complexity, or work profile change).
- *Dynamic Impact Factors*: Dynamic ImFs, by contrast, may change during a PAIS engineering project, e.g., due to the interference of other ImFs. As typical examples of dynamic ImFs consider process and domain knowledge which is typically varying during a PAIS engineering project (or an subsidiary activity).

In practice, it becomes often possible to represent a cost factor either as SCF or as DCF. Likewise, it becomes possible to represent impact factors either as a static or as a dynamic impact factor. Note that the decision about how to represent a cost or impact factor solely depends on the user of the EcoPOST methodology. Notwithstanding, we will give examples of typical representatives of SCFs, DCFs and static and dynamic ImFs in the following. Moreover, Chapter 7 will provide guidelines supporting EcoPOST users in deciding whether a cost or impact factor is represented as a static or as a dynamic factor.

4.3 The Methodology in a Nutshell

Considering the given classification of cost and impact factors, dynamic evaluation factors (i.e., DCFs and dynamic ImFs) are particularly difficult to comprehend. Reason is that the implications of intangible ImFs (i.e., effects due to their emergence in PAIS engineering projects) are difficult to follow. DCFs and dynamic ImFs constitute a major source of misinterpretation and ambiguity when evaluating the costs of PAIS engineering projects (cf. Requirement R-1 in Chapter 3). To better understand and to investigate the dynamic behavior of DCFs and dynamic ImFs, we introduce the notion of *evaluation models* as basic pillar of the EcoPOST methodology. An evaluation model comprises a set of model variables – with SCFs, DCFs, and ImFs corresponding to model variables (cf. Section 4.9.1 for details).

EcoPOST Methodology Overview. When applying the EcoPOST methodology seven steps have to be accomplished (cf. Fig. 4.2). *Step 1* concerns the comprehension of an evaluation scenario. This is crucial for developing problem-specific evaluation models. The following three steps deal with the identification of SCFs (*Step 2*), DCFs (*Step 3*), and ImFs (*Step 4*). *Step 5* concerns the design of evaluation models and *Step 6* deals with their simulation. Both evaluation models and simulation results allow to gain insight into the dynamic behavior (i.e., evolution) of DCFs and dynamic ImFs (*Step 7*).

Fig. 4.3 illustrates the evaluation philosophy underlying our evaluation models. In this figure, costs of business process redesign activities are represented as a DCF (as they are assumed to be changing during the accomplishment of the process redesign activity). During

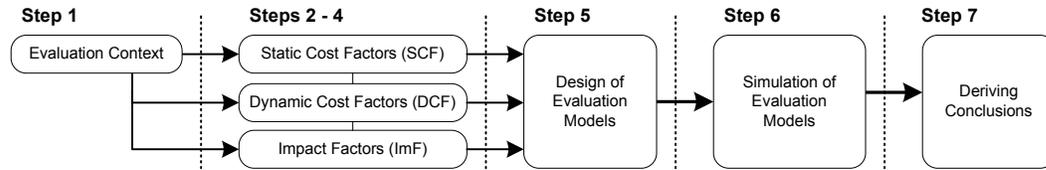


Figure 4.2: Main Steps of the EcoPOST Methodology.

business process redesign, one or several ImFs like "End User Fears" or "Process Knowledge" can influence this activity. While the former may hamper the discovery of information needed to effectively redesign processes (resulting in increasing costs), the latter may simplify the redesign of processes (resulting in decreasing costs). The combined effects of both ImFs make the analysis of the DCF a complex task to accomplish.

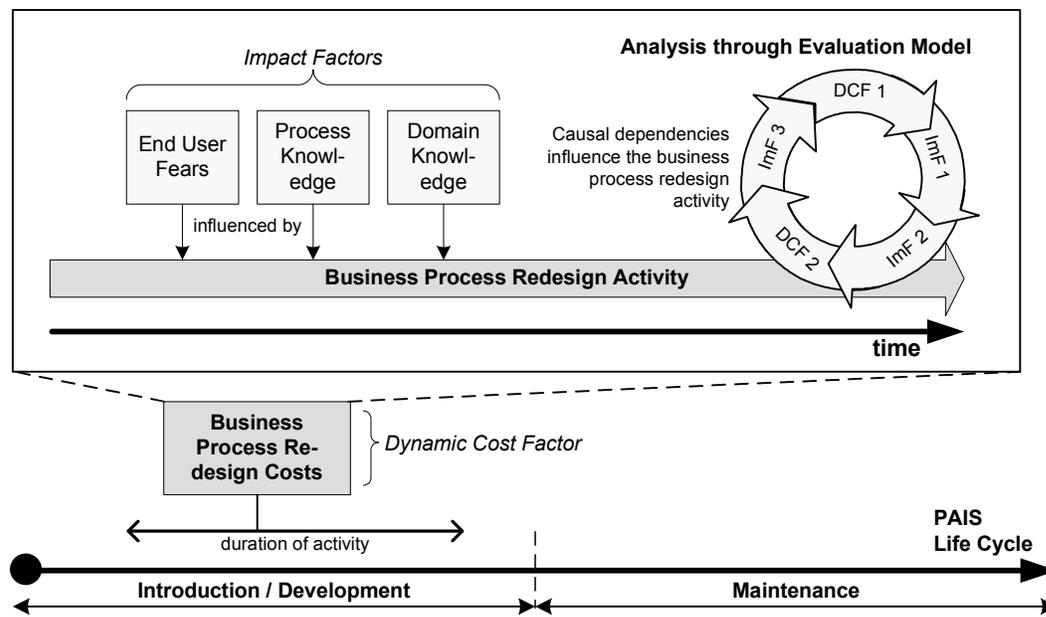


Figure 4.3: The EcoPOST Evaluation Philosophy.

Our evaluation models are based on the *System Dynamics* (SD) notation [50, 64, 147, 166, 189]. SD is a formalism for studying and modeling complex socio-technical systems as they can be found, for example, in biological, environmental, industrial, business, and social systems¹ [166, 64]. Its underlying assumption is that human mind is excellent in observing the

¹Several definitions or characterizations of the term "system" are given in literature. Generally, a system is characterized as a set of interdependent, but correlated variables. Rapoport [158] characterizes a system as "a part of the world, which is sufficiently well defined to be the object of an inquiry or also something, which is characterized by a structure." Shapiro et. al [183] characterize a system as a "family of relationships between its members acting as a whole." Also, purpose and interaction have played an important part in reflections on systems: systems are conceived, in the words of Forrester [64], as "wholes of elements, which cooperate towards a common goal."

elementary forces and actions out of which a system is composed (e.g., fears, delays, resistance to change), but unable to understand dynamic implications resulting from these forces and actions. In PAIS engineering projects we have a similar situation. Such projects are complex socio-technical feedback systems characterized by a strong nexus of organizational, technological, and project-driven factors. While the identification of these factors already constitutes a major problem, the understanding of causal dependencies between factors and resulting effects is even more difficult (cf. Requirement R-9 in Chapter 3).

Generally, "feedback" refers to situations in which a factor X (e.g., user fears) affects another factor Y (e.g., emotional resistance of end users), and factor Y, in turn, directly or indirectly affects X. SD denotes such cyclic chains of causes and effects as *feedback loops*. It particularly assumes that the causal dependency between X and Y cannot be studied without considering the entire system.

There are other formalisms that can be used to model complex systems of interacting factors. Causal *Bayesian Networks* (BN) [84], for example, promise to be a useful approach in this context as well. BN deal with (un)certainty and focus on determining probabilities of events. A BN is a directed acyclic graph which represents interdependencies embodied in a given joint probability distribution over a set of variables. In BN, one distinguishes between measurable variables, intangible variables, and random variables (which form the "Bayesian" aspect of a BN). In our context, we are interested in the interplay of the variables of a system and the effects resulting from this interplay. However, as BN do not allow to model feedback loops (cycles in BN would allow infinite feedbacks and oscillations that would prevent stable parameters of the probability distribution), we cannot use BN.

Agent-based modeling provides another approach for modeling complex socio-technical systems. Resulting models comprise a set of reactive, intentional, or social agents encapsulating the behavior of the various variables that make up a system [34]. By means of simulation, the behavior of these agents can be emulated according to defined rules [178]. System-level information (e.g., about intangible factors being effective in a PAIS engineering project) is thereby not further considered. Since the latter constitutes an important aspect of our approach, we have not further considered the use of agent-based modeling.

In the following sections, we discuss the seven steps of the EcoPOST methodology in detail. Thereby, we rely on a running example in order to clarify described issues.

4.4 Running Example

Assume that a PAIS shall be introduced and that the business process(es) to be supported shall be redesigned before implementing the PAIS [77, 221]. Due to lack of communication about the goals of a PAIS, due to a high degree of job (re)design, and due to changed social clues, introducing the PAIS causes end user fears. These end user fears lead to emotional user resistance. An increasing emotional resistance of end users, in turn, results in a decreasing ability to acquire process knowledge. Reason is that an increasing emotional resistance makes profound process analysis (e.g., based on interviews with process participants) a difficult task to accomplish. A decreasing ability to acquire process knowledge

results in a decreasing ability to (re)design business processes. Finally, an increasing ability to (re)design business processes can even enforce end user fears since end users often consider business process (re)design activities as a potential threat for their own job. This scenario (i.e., the interference of business process (re)design costs through end user fears) shall be investigated in more detail in the following.

4.5 Understanding the Evaluation Scenario

The first step of our methodology concerns the evaluation scenario to be investigated: "*Only by knowing the questions to be answered, we can safely judge the pertinence of factors to include in or omit from the system formulation*" [64]. Clarifying an evaluation scenario is important for two particular reasons. First, it can help to better understand and evaluate a PAIS engineering project. Second, and more important, it can help to select relevant cost and impact factors when building an evaluation model.

Simple questions describing an evaluation scenario and related subproblems can be used for clarifying an evaluation scenario [134]: How does ImF A influence DCF A? How does ImF A influence ImF B? How does business process evolution influence the costs of maintaining a PAIS? How do end user fears influence process analysis costs? How does emotional resistance of end users influence the ability to (re)design business processes?

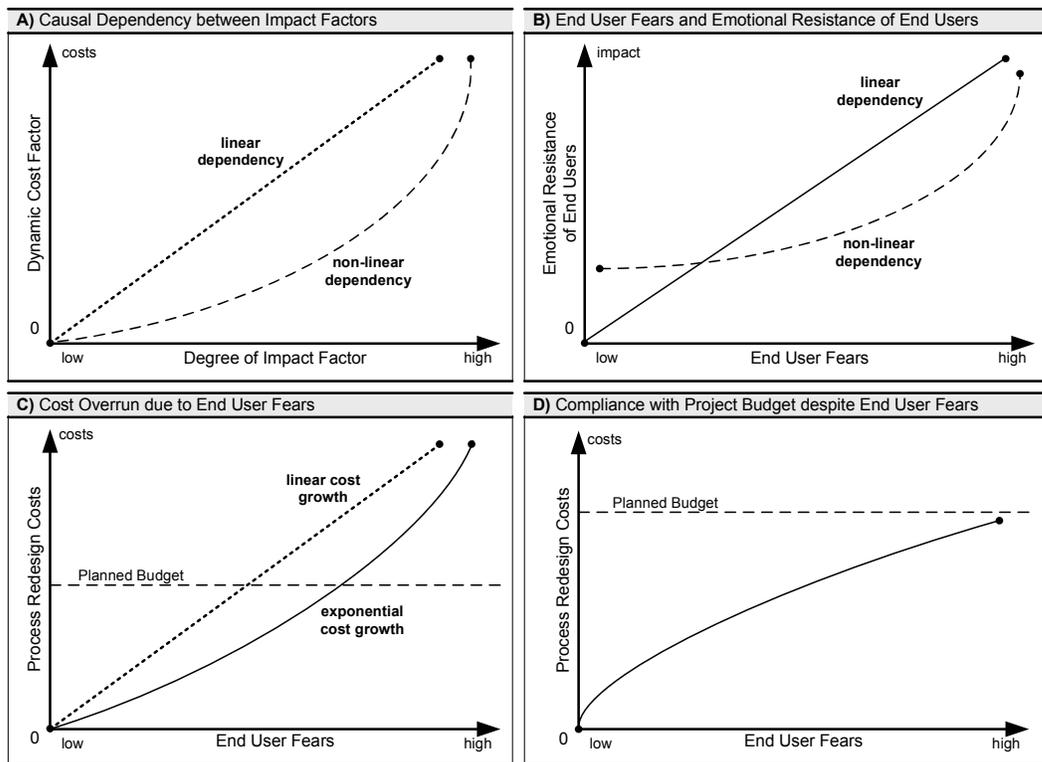


Figure 4.4: Understanding an Evaluation Scenario.

In this context, graphical diagrams illustrating dependencies between cost and impact factors (so called *behavior-over-time-graphs* [64]) can be a useful aid (cf. Fig. 4.4). A behavior-over-time-graph can be considered as a hypothesis which is to be investigated. Fig. 4.4A, for example, illustrates potential (either linear or non-linear) correlations between an arbitrary ImF and a DCF (i.e., the effect of an ImF on a DCF). Picking up our running example (cf. Section 4.4), Fig. 4.4B shows potential correlations between the two ImFs "End User Fears" and "Emotional Resistance of End Users". Generally, high end users fears result in high emotional resistance of end users. Fig. 4.4C and Fig. 4.4D further illustrate the assumed evolution of the DCF "Process Redesign Costs". In the first case (cf. Fig. 4.4C), the DCF is assumed to be growing (either linearly or exponentially) due to end user fears. Hence, the actually planned costs for the process redesign activity cannot be preserved. In the second case (cf. Fig. 4.4C), the DCF approximates the planned budget.

4.6 Static Cost Factors

Step 2 of the EcoPOST methodology deals with the identification of SCFs relevant for the given evaluation scenario. As aforementioned, SCFs represent costs that do not change during a PAIS engineering project, i.e., values of SCFs are constant:

Definition 4.1 (Static Cost Factor) *A cost factor is called static cost factor (SCF), if the costs it represents are constant (i.e., not changing) along the course of time. An SCF can be composed of other, subsidiary SCFs.*

Typical examples are summarized in Fig. 4.5A and include software costs, hardware costs, costs for external consultants, or other budgeted costs.

A) Software Costs	B) Hardware Costs	C) Other potential SCF
<ul style="list-style-type: none"> - Operating system software - Package and licence costs - Application development tools - Security and encryption software - Networking software - Communication software - Systems management software - Database management systems - Front-end applications 	<p>1) Front End Costs</p> <ul style="list-style-type: none"> - Monitors, printers, scanners, other peripherals <p>2) Networking Equipment</p> <ul style="list-style-type: none"> - Cabling, routers, switching devices, encryption devices <p>3) Backend Costs</p> <ul style="list-style-type: none"> - Servers, mainframes, desktop PS units 	<ul style="list-style-type: none"> - Costs for external consultants - Costs for external development - Costs defined by contracts - Costs for user training - Costs for training support personnel - Support costs - Compliance costs <p>Generally: Budgeted (Project) Costs</p>

Figure 4.5: Overview of Potential Static Cost Factors [25].

In our running example (cf. Section 4.4), the originally planned process redesign costs may be represented as SCFs (while the adjusted process redesign costs which are influenced by end user fears may be represented as DCF).

4.7 Dynamic Cost Factors

Step 3 of the EcoPOST methodology deals with the identification of DCFs relevant for a given evaluation scenario. Often, DCFs are composed of other cost factors (either SCFs or

DCFs). In our running example (cf. Section 4.4), the actually perceived process redesign costs may be represented as DCF "Adjusted Process Redesign Costs". This DCF, however, may be composed of other cost factors such as process analysis costs, process modeling costs, or costs for developing and introducing a new process design. We define:

Definition 4.2 (Dynamic Cost Factor) *A cost factor is called dynamic cost factor (DCF), if the costs it represents may change along the course of time. DCFs may be composed of other, subsidiary cost factors. Subsidiary cost factors can be both SCFs and DCFs.*

Thereby, it depends solely on the modeler whether subsidiary cost factors are represented as SCFs or DCFs. In particular, subsidiary DCFs can themselves be composed of other SCFs and DCFs. For this reason the boundaries between DCFs and their subsidiary DCFs often become blurred. While a DCF "Business Process Redesign Costs" can include a subsidiary DCF "Process Modeling Costs" in evaluation scenario A, the DCF "Process Modeling Costs" cannot be a subsidiary DCF but the DCF to be analyzed in another evaluation scenario B. Potential DCFs candidates are as follows:

- **Business Process Redesign Costs:** Costs caused by redesigning business processes can be represented as DCF. This activity implies, for example, costs for conducting interviews with process stakeholders, costs for generating process models and process documentation, or costs for developing and introducing a new process design. Besides, business processes actually not supported by PAIS are often (re)designed too as a consequence of the original (re)design activities. This results in additional costs. All these subsidiary cost factors can be represented either as SCFs or DCFs. Finally note that this DCF shall be analyzed in our running example (cf. Section 4.4).
- **Process Modeling Costs:** Costs caused by process modeling activities can be represented as DCF. This may include, for example, costs for conducting interviews with process stakeholders, costs for process modeling tools, and costs for the modeling activity itself.
- **Organizational Change Costs:** Introducing PAIS often implies changes in an organization. Realizing such changes results in costs which can be represented as DCF, e.g., costs for introducing new or adapting existing organizational structures like team structures and single jobs (respectively work profiles).
- **Requirements Definition Costs:** In PAIS engineering projects, requirements engineering is significantly influenced by characteristics (such as size, complexity) of the business process(es) to be supported [193]. Eliciting requirements results in costs which can be represented as DCF. This implies costs for conducting interviews with process stakeholders or costs for documenting requirements [188].
- **Requirements Management Costs:** Requirements management costs are caused by the activity of managing a given set of requirements throughout the life cycle of a PAIS. Such costs can be represented as DCF as well. They are caused, for example, by eliciting and managing requirement changes.

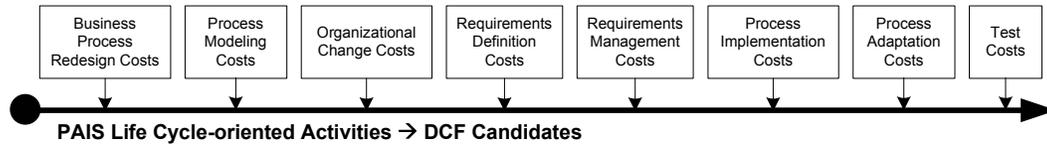


Figure 4.6: Overview of Potential Dynamic Cost Factors.

- **Process Implementation Costs:** Process implementation costs are caused by the activity of implementing the business processes to be supported.
- **Process Adaptation Costs:** Process adaptation costs are caused by adapting the process logic of a PAIS. In PAIS, such adaptations can be often handled at a high semantic level (by manipulating an explicit process model using a process modeling tool) and typically without the need for recoding the PAIS (or parts of it).
- **Test Costs:** Test costs are caused by activities related to the test of a PAIS. This implies not only costs for performing functional tests, but also costs for testing whether a business process is realized as originally planned.

In Section 4.3 we have described how DCFs are influenced by intangible ImFs. The identification of ImFs relevant for a given evaluation scenario is therefore another step in our evaluation methodology, which is discussed in the following section.

4.8 Impact Factors

Step 4 of the EcoPOST methodology deals with the identification of ImFs relevant for a given evaluation scenario. ImFs are intangible factors which influence DCFs. In particular, ImFs lead to the evolution of DCFs as well as dynamic ImFs and make the analysis of these factors a difficult task to accomplish.

ImFs can be composed of other, subsidiary impact factors. An ImF "Work Profile Change", for example, can be determined by other subsidiary ImFs such as "Change of Skill Variety", "Change of Task Significance" and "Change in Autonomy".

Regarding the "impact" of an ImF, we distinguish between two kinds of relationships. A *positive relationship* between x and y (with x the ImF and y the dependent DCF or dynamic ImF) indicates that y will tend in the same direction if a change occurs in x . For example, if a positive link connects the ImF "Process Complexity" to the DCF "Costs of Process Implementation", it will be indicated that a growing process complexity results in increasing costs. A *negative relationship*, by contrast, denotes that the dependent DCF will tend in the opposite direction if a change occurs in x . For example, if a negative link connects the ImF "Process Knowledge" to the DCF "Costs of (re)designing a Business Process", this will indicate that increasing process knowledge (e.g., gathered through interview-based process analysis) results in decreased business process (re)design costs.

4.8.1 Identifying Impact Factors through a Survey

So far, it is necessary to identify relevant ImFs for each evaluation scenario from scratch. To derive a basic set of ImFs which can be used as a baseline for selecting ImFs, we have conducted an online survey among *business process management* (BPM) experts [129].

Background Information. This survey was done over a period of two months in 2006. Due to the many benefits provided by commercial survey tools (e.g., automatic data collection, flexible questionnaire design, support of different question types, support of different analysis tools, etc.), we decided not to implement our own survey delivery platform, but distributed the questionnaire via a Web-based delivery platform the recipients were directed to. This decision was also supported by the fact that several studies (e.g., [115]) have described electronic surveys as advantageous when compared to traditional postal methods.

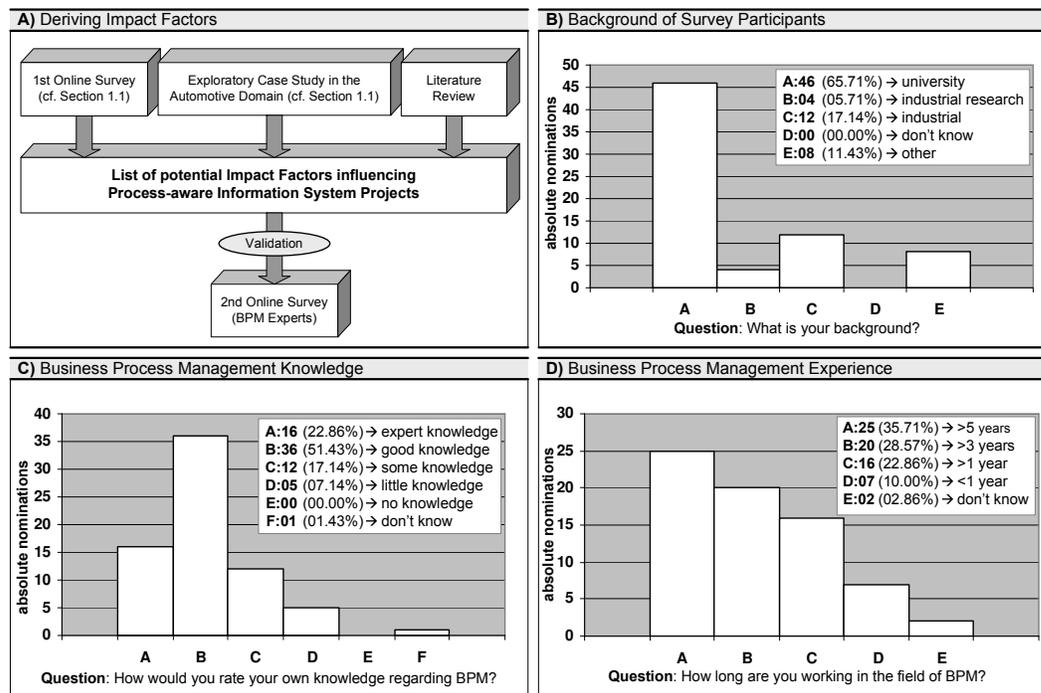


Figure 4.7: Survey Background Information.

Altogether, 70 BPM experts from more than 50 industrial and academic organizations participated² (corresponding to a response rate of 26.21%). Fig. 4.7B to Fig. 4.7D summarize important background information about the survey participants.

²In order to convince people to participate in our survey we conducted several initiatives. First, and most important, email messages were sent over various international computer science mailing lists requesting participants to visit the survey site and to fill out the questionnaire. Two weeks after sending out the initial mails we sent an additional reminder. The effect of this reminder was surprising. Within two days the number of participants raised from 25 to over 60. Secondly, we also used personal contacts to raise awareness for the survey. In doing so, we convinced additional people to participate in the survey.

Developing the Questionnaire. Based on our first online survey (cf. Section 1.1), practical experiences gathered in several large PAIS engineering projects in the automotive domain as well as a profound literature study (see below), we derived a basic set of potential ImFs (cf. Fig. 4.7A). We then empirically validated this list through our online survey.

As mentioned before, there exist several other studies dealing with ImFs in IS projects, mainly in the context of ERP systems. Yusuf et al [222], for example, investigate the introduction of an ERP system in a large manufacturing organization. Thereby, they particularly focus on technological and cultural factors and compare the expected and actual benefits of the ERP system. Vogt [210] analyzes failed ERP projects and identifies – from a software engineering viewpoint – factors which help to avoid such failures. A similar study is described by Voordijk [211], who investigates ERP implementations in large construction firms. Mandal et al [113] describe experiences gathered during the planning and implementation stages of an ERP implementation in a water corporation. Daneva and Wieringa use a "success model" for reasoning about the factors which enable successful ERP implementations [53]. Focusing on potential risk factors, Sumner [190] investigates the re-engineering of business processes prior to an ERP project, investments in recruiting and skilling technology professionals, and the use of external consultants (and the integration of their application-specific knowledge and technical expertise with existing teams). More generally, Davenport [54] deals with only one critical success factor, namely, an organization's ability to align its IS to its business processes and business strategy. Baroudi et al [15] investigate the impact of user involvement on IS usage and information satisfaction. Their results demonstrate that user involvement in IS development enhances both system usage and user satisfaction.

All these studies have contributed to our questionnaire. A particularly important role in this respect plays the research conducted by Parkes who identifies critical success factors for workflow implementations based on a survey [148] and a case study [149]. Many of the factors Parkes has identified have also been included in our questionnaire (e.g., management commitment, communication, and participation by end users).

Most important, our questionnaire (comprising 48 questions in total) investigates potential ImFs along three dimensions: organization-, project-specific, and technology-specific ImFs. In the following sections we discuss our survey results along these three dimensions.

4.8.2 Organization-specific Impact Factors

Organization-specific ImFs deal with attributes of an organization that influence PAIS engineering projects (or subsidiary activities). According to our survey, *end user participation* (47.14%) and *access to required information* (42.86%) are those organization-specific ImFs that aggregate most nominations as "essential factor" (cf. Fig. 4.8A):

- ***End user participation*** is important regarding two activities: (1) elicitation of (non-functional and functional) requirements and (2) (re)design of the business processes prior to the introduction of a PAIS. The costs of both activities will increase if the degree of end user participation decreases.

- **Access to required information** includes access to organizational handbooks, process documentation, application handbooks, and informal information. Unavailable information and devolution of relevant information can negatively affect PAIS engineering projects. Hence, the costs of a PAIS engineering project will increase if the access to required information decreases.

Besides, there are ImFs which are considered as "very important": *reorganization of information, availability of process documentation, ability to (re)design business processes, and the ability of an organization to adapt its IT governance*. A more detailed description of selected ImFs is given in Fig. 4.10A. In order to better understand the relevance of the analyzed ImFs, Fig. 4.8B also shows the mean for each ImF.

4.8.3 Project-specific Impact Factors

Project-specific ImFs (cf. Fig. 4.9A) deal with project-driven attributes which influence PAIS engineering projects. In our survey (cf. Fig. 4.9), *management commitment* (67.14%) and *communication with end users* (45.71%) aggregate most nominations as "essential" project-specific evaluation factors:

- **Management commitment** reflects management support for PAIS engineering projects. Specifically, lack of management commitment can result in increasing resistance to change (e.g., of end users), and, therefore, in increasing costs.
- **Communication with end users** is considered as being crucial while performing a PAIS engineering project. This factor is related to the organizational ImF "End User Participation". However, while the latter deals with the inclusion of end user know-how in PAIS engineering projects, the former concerns the information of end users about the goals and procedures related to the introduction of the PAIS. If communication decreases, the costs of a PAIS engineering project will increase.

Several other ImFs are considered as "very important", including *degree of job (re)design, information about existing processes, and project motivation*. Fig. 4.10B gives a more detailed description of selected project-specific ImFs. Fig. 4.9B shows the mean values.

4.8.4 Technology-specific Impact Factors

Technology-specific ImFs (cf. Fig. 4.11A) deal with the technical infrastructure for developing, maintaining, and introducing PAIS. Interestingly, none of the listed technology-specific ImFs is considered as really essential by the majority of respondents (cf. Fig. 4.11A). *Good documentation* (17.14%) and *usability of the process management platform* (20%), however, get most nominations as "essential factors". These two factors are considered also as "very important" by many survey participants (38.57% and 37.14%).

Other ImFs are considered as "very important" or "important". These ImFs include, for example, *technical maturity of the process management platform, supported degree of process flexibility, and support of standards and norms*. Fig. 4.11B shows the mean for each ImF. Fig. 4.12 summarizes selected technological ImFs.

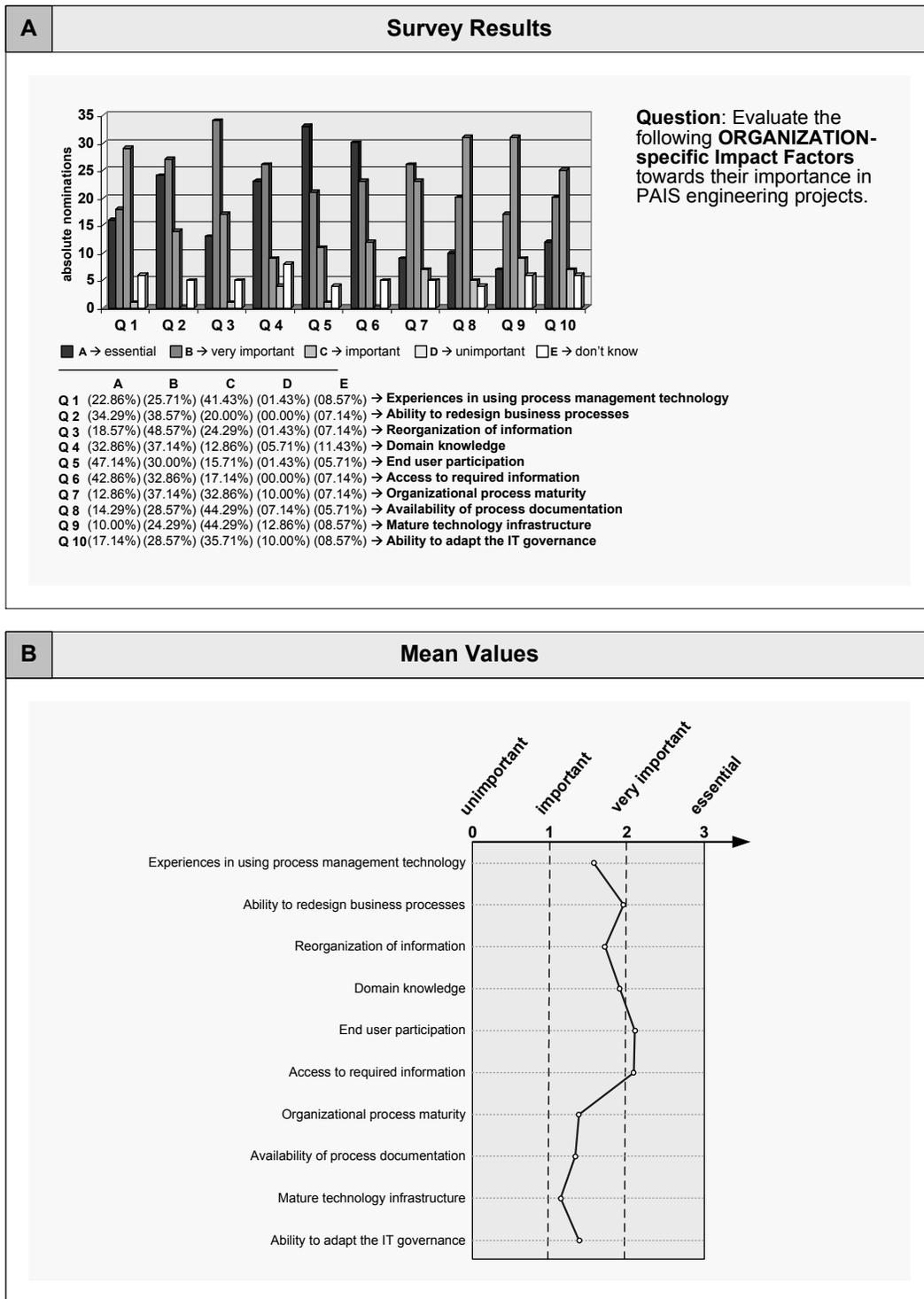


Figure 4.8: Organization-specific Impact Factors (Survey Results).

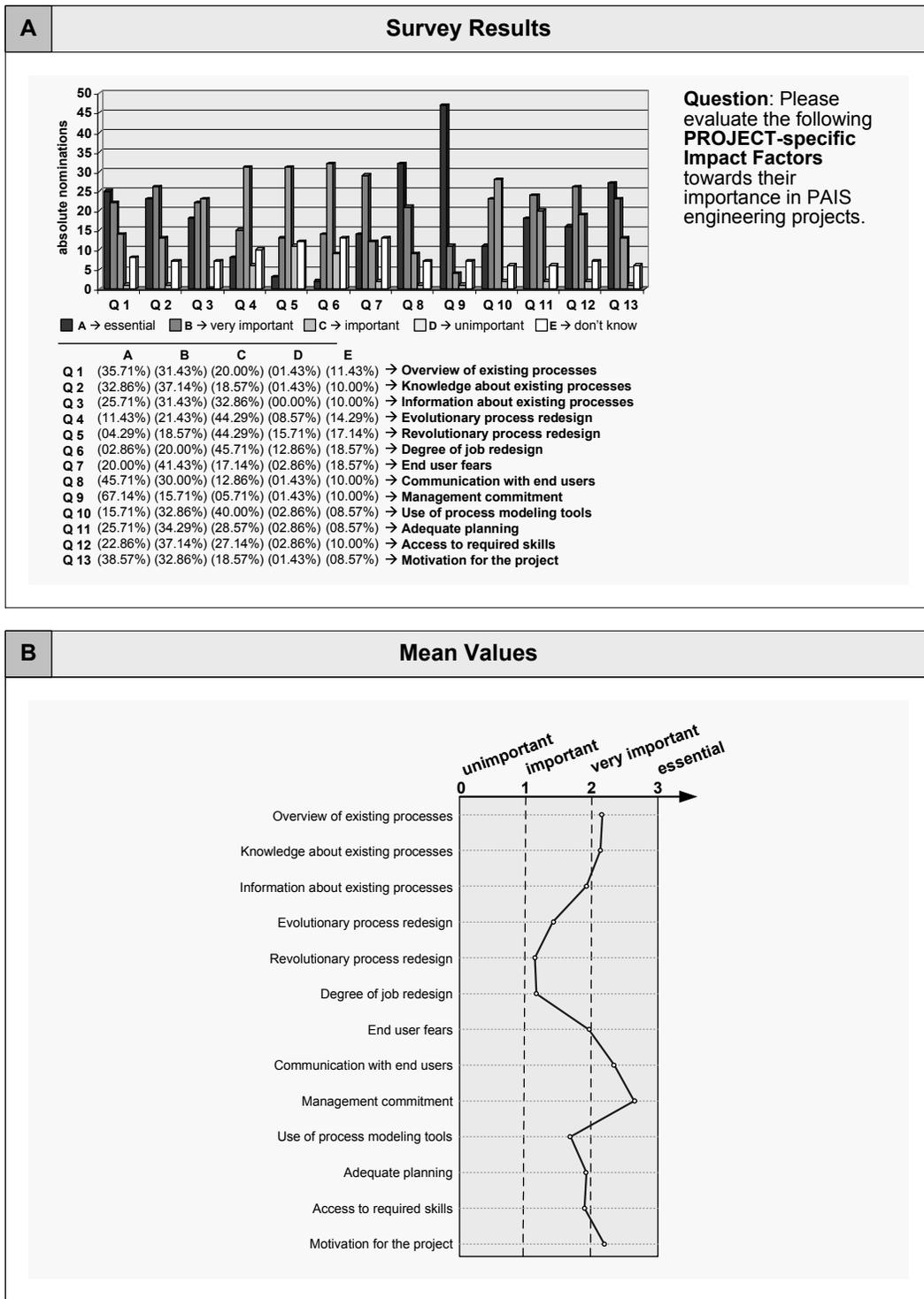


Figure 4.9: Project-specific Impact Factors (Survey Results).

A) Organizational Impact Factors (Selection)
<p>Experiences in using Process Management Technology: There exists a relationship between the experiences of an organization, of PAIS architects, and of PAIS developers in using process management technology on the one hand, and the costs of a PAIS engineering project on the other hand. In particular, if experiences increase, the costs of a PAIS engineering project will decrease.</p>
<p>Ability to (re)design Business Processes: There exists a relationship between the ability to (re)design business processes on the one hand, and the costs of a PAIS engineering project on the other hand. Successfully (re)designing business processes is a major success factor for efficiently realizing PAIS. If the ability to (re)design business processes decreases (e.g., due to strong user fears, missing management commitment, etc.), the costs of a PAIS engineering project will increase.</p>
<p>Reorganization of Information: The reorganization of information (e.g., of the data processed by a PAIS) has a significant impact on the costs of a PAIS engineering project. Reorganizing information (as well as other resources) can create conflicts within an organization. These conflicts, in turn, can influence a PAIS engineering project. On the other hand, an effective reorganization of information (though producing additional efforts) can reduce the costs of a PAIS engineering project.</p>
<p>Domain Knowledge: Domain knowledge can influence the costs of a PAIS engineering project. Thereby, domain knowledge is determined by the time a PAIS engineer has worked in a specific domain. In particular, domain knowledge increases with the time the PAIS engineer has worked in the domain. Domain knowledge is related to the ImF "Experiences in using Process Management Technology" (see above). However, it does not deal with experiences in using technology but with experiences in a specific organizational domain. If domain knowledge increases, the costs of a PAIS engineering project will decrease.</p>
<p>Organizational Process Maturity: Process maturity can be assessed by dedicated maturity models that describe characteristics of effective process organizations. Examples are the <i>capability maturity model integration</i> (CMMI) or the <i>software process improvement and capability determination</i> (SPICE) model. If the process maturity of an organization is low, it will be more difficult to implement (optimized) business processes, i.e., costs will increase.</p>
B) Project-specific Impact Factors (Selection)
<p>End User Fears: There exists a relationship between the degree of end user fears on the one hand, and the costs of a PAIS engineering project on the other hand. In particular, the introduction of a PAIS may cause end user fears, e.g., due to a high degree of job redesign and due to changed social clues. Furthermore, such end user fears can lead to emotional user resistance. This, in turn, can make it difficult to get support from the users while introducing a PAIS. Hence, increasing end user fears can result in increasing costs of a PAIS engineering project.</p>
<p>Knowledge about existing Processes (Process Knowledge): When developing a PAIS, detailed knowledge about the business processes to be supported is crucial. Specifically, process knowledge answers the question: What is the business process actually doing? This includes knowledge about the flow of data, control flows, as well as knowledge about process attributes (process owners, roles, etc.). Note that there exist several approaches to derive process knowledge, whereas the classic approach is to perform a manual process analysis based on interviews with process participants. More advanced process analysis features include the modeling and simulation of business processes. Business process intelligence tools (cf. Section 2.5.4) can be applied as well. Generally, increasing process knowledge results in decreasing costs.</p>
<p>Access to required Skills: There exists a relationship between the access to required skills (e.g., business process analysis skills, business process modeling skills, workflow management knowledge, etc.) on the one hand and the costs of a PAIS engineering project on the other hand. The unavailability of needed skills influences many activities like process (re)design activities or the overall implementation of the PAIS. Insufficient access to required skills can result in increasing costs of a PAIS engineering project.</p>
<p>Motivation for the Project: The motivation of project staff for a PAIS engineering project influences the costs of the respective project. If motivation decreases (e.g., due to low management commitment, insufficient project resources, etc.), this will result in lower productivity, and therefore in increasing costs.</p>
<p>Degree of Job Redesign: During a PAIS engineering project, existing work profiles are often redesigned. However, while expectations of policy makers and IT managers are carefully considered in most cases, the wishes of end users (i.e., the users working with the PAIS) are often neglected in this context. Consequently, the new work profile results in end user fears, e.g., due to changing skill variety or changes in the autonomy of workers (due to the automation of process fragments by means of workflow technology). End user fears, in turn, may lead to emotional resistance and missing (but highly needed) user support during process design activities (e.g., when conducting interview-based process analysis). This results in increasing costs for a PAIS engineering project. When taking into account the degree of work profile change, in turn, it becomes possible to avoid unexpected user resistance and to initiate counteractive measures at an early stage (e.g., through a better communication policy).</p>

Figure 4.10: Selected Organization- and Project-specific Impact Factors.

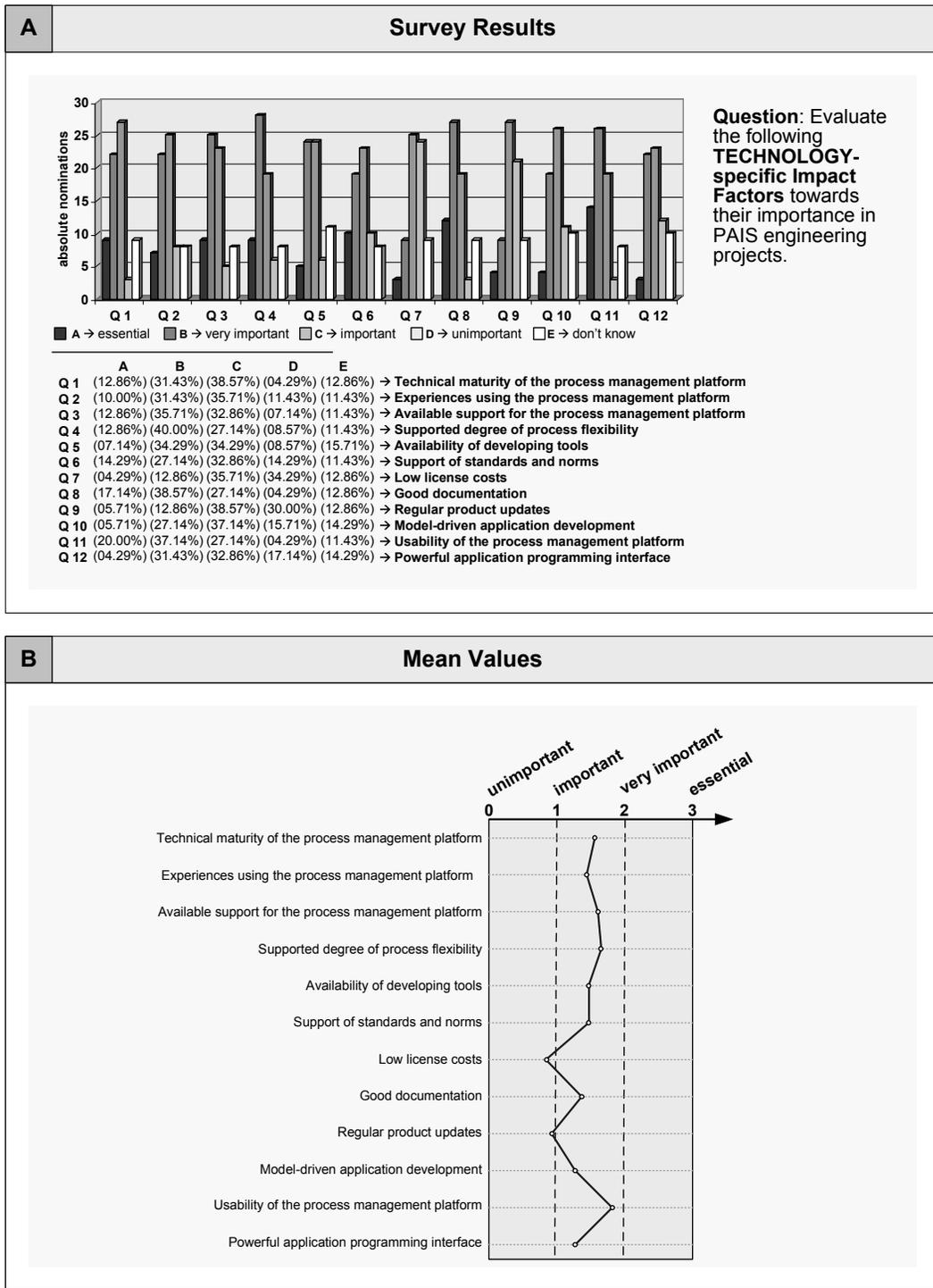


Figure 4.11: Technology-specific Impact Factors (Survey Results).

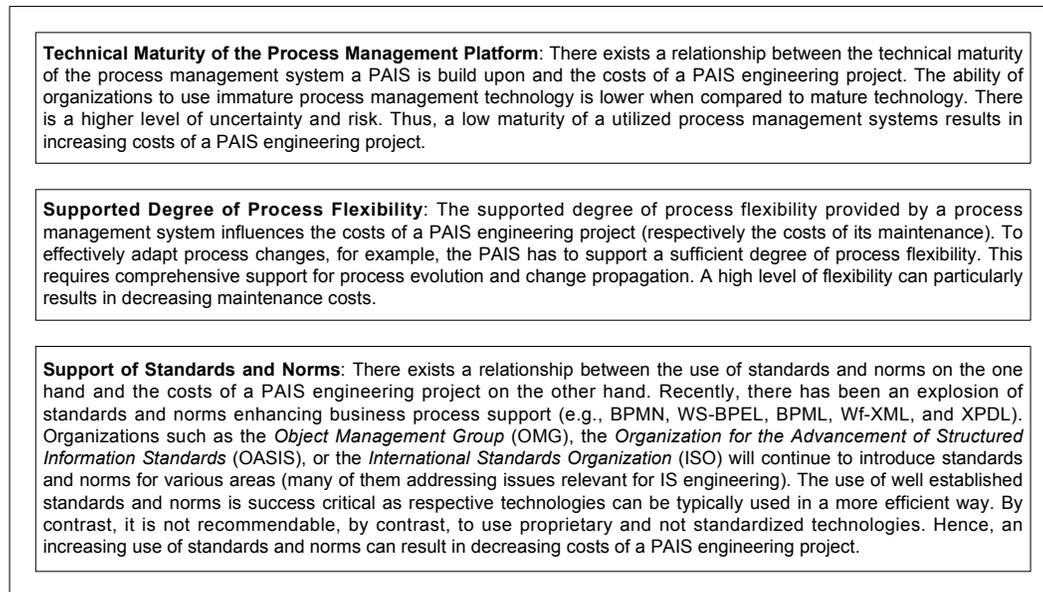


Figure 4.12: Selected Technology-specific Impact Factors.

4.9 Evaluation Models

So far, we have described SCFs, DCFs, and a basic set of organization-, project-, and technology-specific ImFs. In this section, we describe how these evaluation factors are combined into evaluation models in order to better understand their dynamic interplay [134, 135, 138].

Section 4.9.1 introduces the notation of our evaluation models. Afterwards, Section 4.9.2 describes a basic model construct for describing DCF interference through ImFs. Section 4.9.3 gives an illustrating example. Section 4.9.4 deals with model testing. Finally, Section 4.9.5 discusses the problem of model validation.

4.9.1 Model Notation

In our approach, DCFs and dynamic ImFs are analyzed by evaluation models. As already discussed in Section 4.3, these models are specified using the System Dynamics [166, 147] notation (cf. Fig. 4.13). An evaluation model comprises a set of model variables with SCFs, DCFs, and ImFs corresponding to model variables.

Different types of variables exist. *State variables* can be used to represent dynamic factors, i.e., to capture changing values of DCFs (e.g., the "Business Process Redesign Costs"; cf. Fig. 4.13A) and dynamic ImFs (e.g., "Process Knowledge"). A state variable is graphically denoted as rectangle (cf. Fig. 4.13A), and its value at time t is determined by the accumulated changes of this variable from starting point t_0 to present moment t ($t > t_0$); similar to a bathtub which accumulates – at a defined moment t – the amount of water which has been poured into it in the past. Typically, state variables are connected to at least one *source* or *sink* which are graphically denoted as cloud-like symbols (except for state

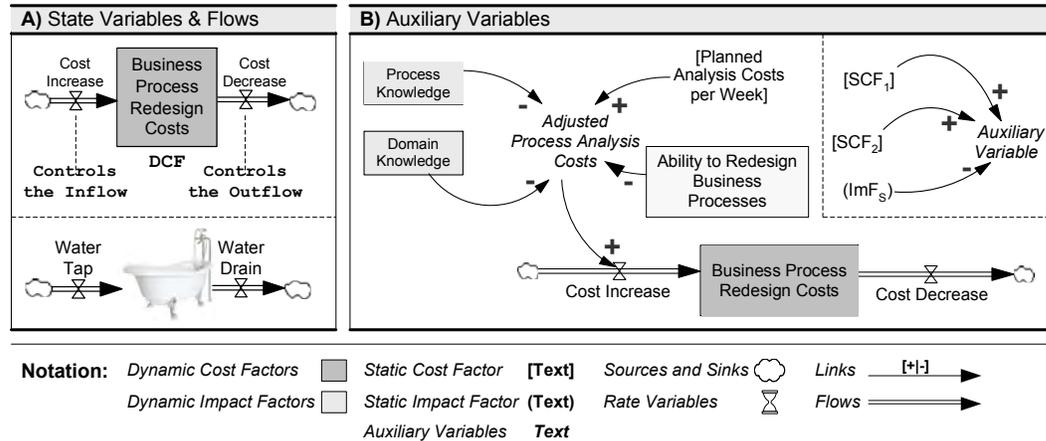


Figure 4.13: Evaluation Model Notation and Initial Examples.

variables connected to other state variables).

Values of state variables change through inflows and outflows. Graphically, both flow types are depicted by twin-arrows which either point to (in the case of an inflow) or out of (in the case of an outflow) the state variable (cf. Fig. 4.13A). Picking up again the bathtub image, an *inflow* is a pipe that adds water to the bathtub, i.e., inflows increase the value of state variables. An *outflow*, by contrast, is a pipe that purges water from the bathtub, i.e., outflows decrease the value of state variables. The DCF "Business Process Redesign Costs" shown in Fig. 4.13B, for example, increases through its inflow ("Cost Increase") and decreases through its outflow ("Cost Decrease"). Returning to the bathtub image, we further need "water taps" to control the amount of water flowing into the bathtub, and "drains" to specify the amount of water flowing out. For this purpose, a *rate variable* is assigned to each flow (graphically depicted by a valve; cf. Fig. 4.13A). In particular, a rate variable controls the flow it is assigned to based on those SCFs, DCFs, and static and dynamic ImFs which influence it. It can be considered as an interface which is able to merge SCFs, DCFs, and static and dynamic ImFs.

Besides state variables, evaluation models may comprise *constants* and *auxiliary variables*. Constants are used to represent static evaluation factors, i.e., SCFs and static ImFs. Auxiliary variables, in turn, represent intermediate variables and typically bring together – like rate variables – cost and impact factors, i.e., they merge SCFs, DCFs, and static and dynamic ImFs³. As an example consider the auxiliary variable "Adjusted Process Analysis Costs" in Fig. 4.13B. This auxiliary variable merges together the three dynamic ImFs "Process Knowledge", "Domain Knowledge", and "Ability to Redesign Business Processes" and the SCF "Planned Analysis Costs per Week".

Both constants and auxiliary variables are integrated into an evaluation model with *links* (not flows), i.e., labeled arrows. A *positive link* (labeled with "+") between x and y (with y as dependent variable) indicates that y will tend in the same direction if a change occurs in

³Besides, auxiliary variables can be also used to establish nonlinear dependencies between model variables. This special case is further discussed in Section 5.3.6.

x. A *negative link* (labeled with "-") expresses that the dependent variable y will tend in the opposite direction if the value of x changes. Altogether, we can define:

Definition 4.3 (Evaluation Model) A graph $EM = (V, F, L)$ is called *evaluation model*, if the following holds:

- $V := S \dot{\cup} X \dot{\cup} R \dot{\cup} C \dot{\cup} A$ is a set of model variables with
 - S is a set of state variables,
 - X is a set of sources and sinks,
 - R is a set of rate variables,
 - C is a set of constants,
 - A is a set of auxiliary variables,
- $F \subseteq ((S \times S) \cup (S \times X) \cup (X \times S))$ is a set of edges called *flows*,
- $L \subseteq ((S \times A \times Lab) \cup (S \times R \times Lab) \cup (A \times A \times Lab) \cup (A \times R \times Lab) \cup (C \times A \times Lab) \cup (C \times R \times Lab))$ is a set of edges called *links* with $Lab := \{+, -\}$ being the set of link labels, where
 - $(q_i, q_j, +) \in L$ with $q_i \in (S \dot{\cup} A \dot{\cup} C)$ and $q_j \in (A \dot{\cup} R)$ denotes a *positive link*,
 - $(q_i, q_j, -) \in L$ with $q_i \in (S \dot{\cup} A \dot{\cup} C)$ and $q_j \in (A \dot{\cup} R)$ denotes a *negative link*.

In our evaluation models, DCFs and dynamic ImFs are represented by state variables, whereas SCFs and static ImFs are represented by constants. Obviously, our evaluation models have a sufficient theoretical foundation, a requirement which has been identified in Chapter 3 (cf. Requirement R-9).

4.9.2 Illustrating Example

Fig. 4.14 shows the overall evaluation model for our running example. It describes the influence of the dynamic ImF "End User Fears" on the DCF "Costs for Business Process Redesign". Basic to this evaluation model is a cyclic structure connecting the four ImFs "End User Fears", "Emotional Resistance", "Ability to Acquire Process Knowledge", and "Ability to Redesign Business Processes". Their arrangement (cf. Fig. 4.14) illustrates the following coherence: Increasing end user fears (e.g., due to a high degree of job (re)design and due to changed social clues) result in increased emotional resistance of end users. This dependency is represented by a positive link from the ImF "End User Fears" to the "Resistance Growth Rate" (which controls the inflow of the ImF "Emotional Resistance"). An increasing emotional resistance of end users, in turn, results in a decreasing ability to acquire process knowledge. Reason is that an increasing emotional resistance makes profound process analysis (e.g., based on interviews with process participants) a difficult task to accomplish. This dependency is represented by a negative link from the ImF "Emotional

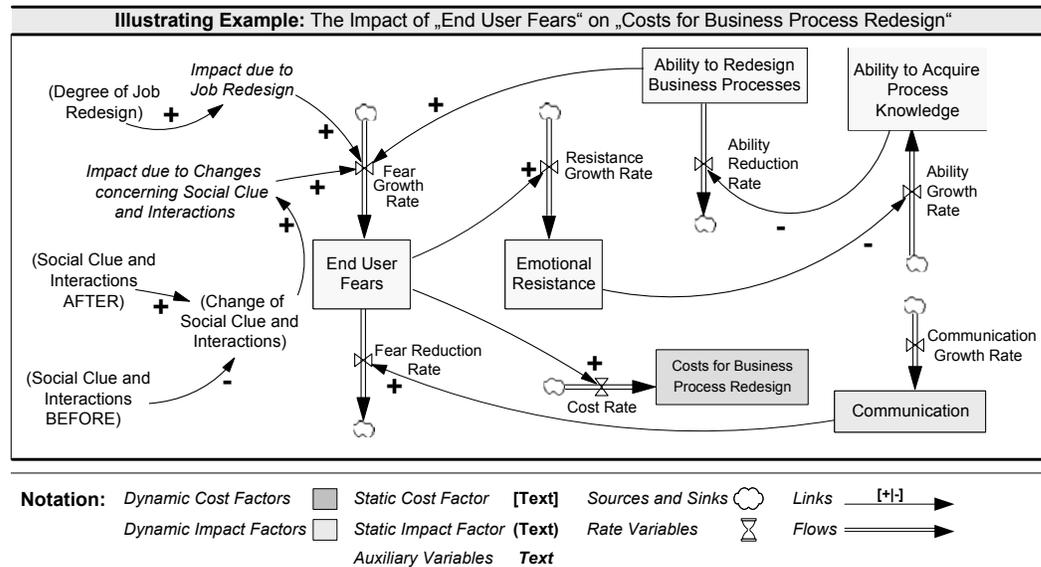


Figure 4.14: Dealing with the Impact of End User Fears.

Resistance" to the rate variable "Ability Growth Rate" (which controls the inflow of the ImF "Ability to Acquire Process Knowledge").

A decreasing ability to acquire process knowledge results in a decreasing ability to (re)design business processes. Again, this dependency is represented by a positive link. Finally, an increasing ability to (re)design business processes can even enforce end user fears since end users often consider business process (re)design activities as a potential threat for their own job. This dependency is represented by another positive link. Thereby, the "Fear Growth Rate" is not only biased by this link. It is also influenced by the "Impact due to Job Redesign" and the "Change of Social Clue and Interactions" (which is calculated from the social clues and interactions before and after the business process (re)design). Finally, "Communication" is considered as well. This ImF deals with the information of end users about the goals of introducing a PAIS.

So far, we have discussed the design of EcoPOST evaluation models. Such models are of significant value for PAIS engineers, e.g., due to their suitability to serve as a consciousness-raising tool about the economics in PAIS engineering projects.

We have not yet discussed correctness of evaluation models. Obviously, without providing further model design rules, incorrect evaluation models can be quickly modeled (aggravating the derivation of plausible evaluations; cf. Requirement R-4 in Chapter 3). The following section picks up this problem and describes respective model design rules.

4.9.3 Model Design Rules

This section discusses model design rules which ensure the syntactical and improve the semantical correctness of our evaluation models.

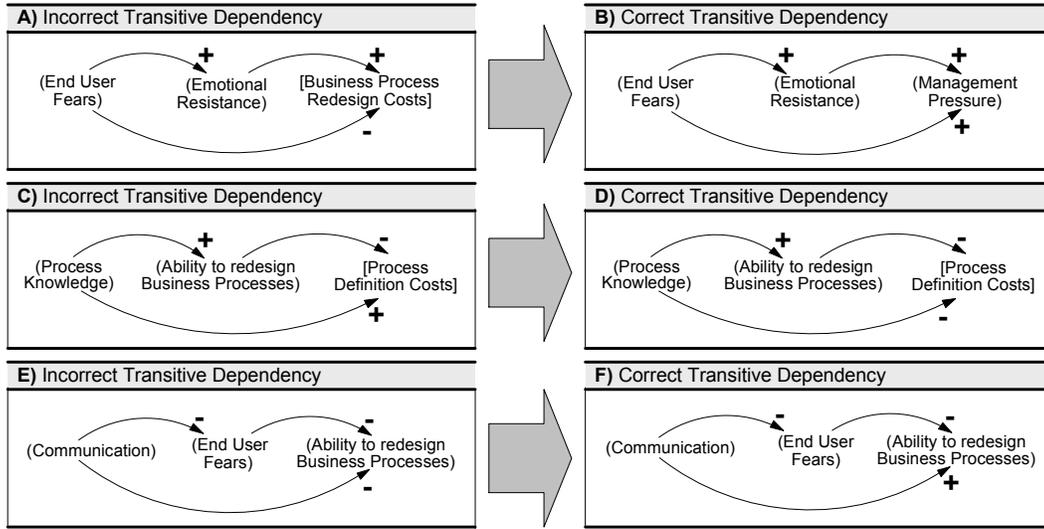


Figure 4.17: Transitive Dependencies (Simplified Evaluation Models).

Altogether, two causal relations ("+" and "-") are used in the context of our evaluation models. Correct transitive dependencies can be described based on a *multiplication operator* (i.e., by "*"). More precisely, transitive dependencies have to comply with the following three *multiplication laws* for transitive dependencies (for any $x, y \in \{+, -\}$):

$$+ * y = y \quad (4.7)$$

$$- * - = + \quad (4.8)$$

$$x * y = y * x \quad (4.9)$$

Considering these laws, the evaluation models shown in Fig. 4.17A and Fig. 4.17C violate the first law, whereas the model shown in Fig. 4.17E violates the second one. Law 3 states the "*" is commutative.

Design Rule 7 (Dual Links I) – A constant cannot be connected to the same auxiliary variable with both a positive and negative link:

$$\forall v \in C \wedge \forall q \in A : \neg \exists ((v, q, [-]) \wedge (v, q, [+])) \in L \quad (4.10)$$

Design Rule 8 (Dual Links II) – A state variable cannot be connected to the same auxiliary variable with both a positive and negative link:

$$\forall v \in S \wedge \forall q \in A : \neg \exists ((v, q, [-]) \wedge (v, q, [+])) \in L \quad (4.11)$$

Design Rule 9 (Dual Links III) – An auxiliary variable cannot be connected to another auxiliary variable with both a positive and negative link:

$$\forall v, \forall q \in A : \neg \exists ((v, q, [-]) \wedge (v, q, [+])) \in L \quad (4.12)$$

Besides, there exist some additional simple constraints:

Design Rule 10 (Representing Cost Factors) – A cost factor cannot be represented both as SCF and DCF in an evaluation model.

Design Rule 11 (Representing Impact Factors) – An impact factor cannot be represented both as static and dynamic ImF in an evaluation model.

4.9.4 Model Validation

Generally, the correctness of an (evaluation) model does not only depend on its compliance with model design rules, but also on its ability to represent the real world [207], i.e., on its useful domain of applicability [212]. In other words, a model has to be *valid*.

The roots of *model validation* have to be analyzed in a historical context, in particular in the historical development of the different theories of knowledge. Thereby, one can distinguish between two opposing philosophies of science [224]. One paradigm, originated with Descartes' *Rationalism* (1596-1650) and Locke's *Empiricism* (1632-1704), guided the major epistemological theories of the 16th through the 19th centuries. This traditional *logical/empiricist philosophy of science* assumes that knowledge is an objective representation of reality and that theory justification can be an objective, formal process. Besides, a different paradigm emerged based on Hegel's *Coherence Theory* (1770-1831) and Dewey's *Pragmatism* (1859-1952). Both articulated that knowledge is socially justified belief, rather than a product of mirroring nature and, thus, socially, culturally, and historically dependent. In [103], Kuhn argued that scientific progress is not directed toward an objective and absolute "truth" but simply toward "successful creative work". Others proposed that knowledge acquisition is holistic rather than atomistic. This second paradigm is also called *relativist/holistic philosophy of science*.

Our EcoPOST evaluation models (like SD models in general) are more closely related to the relativist/holistic viewpoint. Forrester [64], for example, embraces the relativist approach (i) proposing that the validity of a SD model cannot be discussed absent of its purpose, (ii) accepting "qualitative" model validation, and (iii) interpreting "data" in a broad sense (including non-numerical and verbal information, and mental data). In [63], Forrester and Senge stress that SD models are built for a purpose and that their validity is fundamentally determined by the extent to which they fulfill that purpose.

In [189], Sterman states that the validity of a model has to rest on the validity of its constituent parts (though this is no guarantee of validity). Sterman argues that all SD models are, in a sense, wrong because there could always exist a counter example to which the model did not conform completely. Barlas [13, 14] concludes that SD is a scientific method according to the standards placed by the relativist/holistic philosophy of science: "*Real-life experience has taught most system dynamics practitioners that models are inherently incomplete, relative, and partly subjective, and that model validity means usefulness with respect to a purpose. But, at the same time, many practitioners unaware of the recent relativist philosophical developments would think that their own view of model validity is*

not truly scientific. Thus, many practitioners, while experiencing that validation is bound to be relative, semi-formal, and conversational process, at the same time see this as a weakness of their modeling effort; System Dynamics practitioners do not have to be apologetic for not meeting a utopian [logical/empiricist] criterion of scientific inquiry." Model validation is also addressed by Coyle [49, 50]. Generally, he defines the validity of a model as being "*well-suited to its purpose and soundly constructed*". Thereby, "*the model should do the same things as the real system and for the same reasons*".

Thus, the validity of our evaluation models (as well as of SD models in general) is difficult to ensure. In response to this problem, we rely on empirical and experimental research to generate data for validation purpose. Such research is crucial where analytic solutions are neither possible nor practical and where emerging technologies (instead of mature ones) are considered [124]. According to Briand et al. [35] "*each discipline needs to develop its own body of experience and strategies to answer its most pressing research questions*".

In this thesis, the "body of experience" corresponds to two online surveys, three case studies, and one controlled software experiment. Some results of these activities (e.g., of the online surveys) have been already discussed, other will be discussed in later chapters (cf. Part III of this thesis).

One additional though often costly opportunity to increase confidence in evaluation models is to perform model tests⁴. However, as model testing is not the focus of this thesis and as model testing has been intensively discussed in SD literature, we do not discuss it in detail here. Instead, we shortly summarize existing work.

Forrester [64], for example, recommends to strictly ensure model consistency. Model variables and their dimensions (e.g., costs for DCFs) must be clearly defined. Forrester additionally remarks that model behavior should be carefully checked, as "*serious model defects will usually expose themselves through some failure of the model to perform as would be expected of the real system*". In [63], Forrester and Senge describe 17 tests for validating SD models. These 17 tests were grouped into three categories: (1) *model structure tests*, (2) *model behavior tests*, and (3) *policy implication tests*. In [189], Sterman aggregates these 17 tests into 12 ones. In [224], Zagonel and Corbet reorganize Forrester's tests into five categories: (1) *systems mapping*, (2) *quantitative modeling*, (3) *hypothesis testing*, (4) *uncertainty analysis*, and (5) *forecasting and optimization*. Model tests are also considered by Coyle [49, 50] who – like Forrester – discusses principles of dimensional analysis.

Generally, it is often unclear whether the costs of performing model tests really exceed the gained benefits. In [212], Wakeland and Hoarfrost apply model tests as described by Sterman [189] to a complex SD model, measure the time required to carry out each type of test, and finally relate these costs to the gained (and subjectively measured) benefits resulting from each test. Their results indicate that rather than focusing primarily on sensitivity testing, modelers should consider other types of model tests such as "extreme condition tests" and "family member tests". Most important, their study indicates that all proposed

⁴Note that model testing as described in this section also includes tests for analyzing simulation models underlying our evaluation models (cf. Section 4.3). Though we have not yet discussed simulation, we already discuss the issue of model testing at this point and refer to this section in later chapters.

tests were either moderately useful or even very useful.

4.10 Simulating Evaluation Models

Evaluation models like the one discussed in Section 4.9.2 are very useful for PAIS engineers. Yet, the evolution of DCFs and dynamic ImFs, their interference through ImFs, and the dynamic interplay of the different elements of an evaluation model are difficult to comprehend. For this reason, we add components for analyzing the dynamic implications of our evaluation models to our EcoPOST methodology (Step 6 of the EcoPOST methodology).

In order to investigate how a given evaluation model "works" and what policies might change its behavior we simulate the dynamic implications described by it – a task which is typically too complex for the human mind [64, 76]. In particular, we conduct "behavioral experiments" based on series of simulation runs. During these simulation runs selected simulation parameters are manipulated in a controlled manner to systematically investigate the effects of these manipulations, i.e., to investigate how the output of a simulation will vary if the initial condition of a simulation is changed. This procedure is also known as *sensitivity analysis*.

Outcomes of simulation runs can then be further analyzed and compared using graphical charts such as depicted in Fig. 4.18A and Fig. 4.18B.

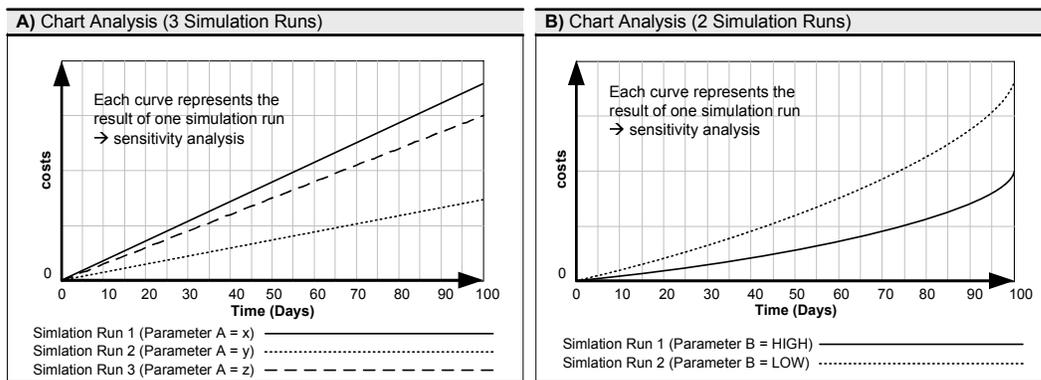


Figure 4.18: Interpreting Simulation Outcomes.

Simulation is enabled by the definition and execution of a simulation model. In our context, the *simulation model* consists of a number of *mathematical equations* – one for each model variable (i.e., dynamic and static evaluation factors as well as rate variables and auxiliary variables). Though we consider the simulation model for an evaluation model as a separate artifact in the following, it becomes also possible to consider the simulation model just as another, executable representation of the evaluation model (which is a mere graph). However, we omit further details here and refer to Chapter 5 which discusses the simulation of evaluation models and related issues (e.g., the definition of simulation models) in detail.

4.11 Deriving Conclusions

Based on EcoPOST evaluation models and results of their simulation PAIS engineers can get insights into causal dependencies and resulting (cost) effects arising in PAIS engineering projects (cf. Requirement R-2 in Chapter 3). More specifically, our approach allows PAIS engineers to gain insights into the following issues:

- ***Investigating Feedback Structures:*** In Chapter 3, we have denoted the modeling of feedback structures and causal dependencies between cost and impact factors as one major requirement (cf. Requirement R-9). Our evaluation models pick up this requirement and enable not only the modeling of causal dependencies, but also the investigation of cyclic feedback structures.
- ***Visualizing Causal Dependencies:*** Our evaluation models offer a simple way of visualizing both the structure and the behavior of interacting cost and impact factors in PAIS engineering projects (cf. Requirement R-9 in Chapter 3). Thus, evaluation models can be easily communicated to decision makers and support decision-making.
- ***Investigating Soft Impact Factors:*** Our evaluation models enable PAIS engineers to investigate the effects intangible impact factors (such as end user fears, management commitment, and process complexity) both in a qualitative (cf. Requirement R-5 in Chapter 3) and a quantitative manner (cf. Requirement R-3 in Chapter 3).
- ***Investigating Delays:*** PAIS engineering projects are typically faced with many delays (e.g., in the evolution of evaluation factors). These delays often develop over time due to internal or external influences. End user fears, for example, may be low at the beginning of a PAIS engineering project, but may quickly increase later. Simulating our evaluation models enables PAIS engineers to investigate such delays (based on the analysis of DCFs and dynamic ImFs).
- ***Performing Sensitivity Analysis:*** Every variable modification in an evaluation model results in various consequences. Some of these consequences can be anticipated and intended. Many others, however, are typically unanticipated and unintended. The opportunity to analyze such effects based on the simulation of evaluation models enables sensitivity analysis and the investigation of (both expected and unexpected) effects (cf. Requirement R-3 in Chapter 3).

4.12 Summary

This chapter has described the main steps of the EcoPOST methodology. This methodology comprises seven steps. Step 1 deals with the comprehension of an evaluation scenario. Step 2 addresses the identification of SCFs, Step 3 the identification of DCFs, and Step 4 the identification of ImFs. Step 5 concerns the design of evaluation models. In this context, we have introduced the notation of our evaluation models, have given an illustrating example, and have also discussed model design rules, model tests, and the problem of model

validation. Step 6 addresses the further analysis of evaluation models through simulation. Simulation becomes necessary in order to unfold the dynamic behavior described by our evaluation models. Our evaluation models as well as their simulation finally allow us to gain insights into the evolution of dynamic evaluation factors and to better understand the dynamic costs of PAIS engineering projects (Step 7). The following chapter picks up Step 6 of this methodology and describes the simulation of evaluation models in detail.

In Chapter 3, we have identified eleven requirements for the design of an economic-driven evaluation approach for PAIS engineering projects: five requirements for economic-driven IT evaluation in general (cf. Section 3.2), and six more specific requirements for evaluating PAIS engineering projects (cf. Section 3.3). For example, we have identified the performance of cost evaluations as an important requirement (R-1). The EcoPOST methodology addresses this requirement based on its evaluation models comprising both static and dynamic cost factors. We have also considered assistance for decision support as relevant requirement (R-2). However, this requirement is only partly fulfilled by the EcoPOST methodology. Focus of our approach is the analysis of dynamic cost and impact factors along the course of time. Decision support is thereby only implicitly supported by raising awareness about the complex cost effects and causal dependencies in PAIS engineering projects (i.e., explicit criteria enhancing decision making are not provided). Modeling causal dependencies has been another important requirement in order to investigate the interplay of organization-, project-, and technology-specific cost and impact factors (R-9) and is also a fundamental notion underlying the EcoPOST methodology. Thus, we consider this requirement as being fulfilled.

We have also identified the need for deriving plausible conclusions as a major requirement (R-4). Generally, the development of plausible evaluation models is a difficult task to accomplish. Notwithstanding, it is possible to scrutinize the overall suitability of evaluation models (though "*validation and verification of models is impossible*" [189]). As an example for respective actions consider the compliance of evaluation models with defined model design rules (cf. Chapter 4). Another example is a sufficient theoretical foundation of evaluation models (R-10). In this context, it is important to mention that EcoPOST evaluation models – like conventional SD models – have a sound theoretical foundation. Finally, we have considered the support of quantitative (R-3) qualitative (R-5) conclusions as important. Both requirements are fulfilled by the EcoPOST framework. While qualitative conclusions can be derived based on the causal dependencies described by EcoPOST evaluation models, quantitative conclusions become possible based on the simulation of evaluation models.

Note that the requirements which are not covered so far (R-6, R-7, R-8, R-11) will be discussed in Chapter 12 after having introduced additional solution concepts included in the EcoPOST framework in the following chapters.

Chapter 5

Simulating Evaluation Models

5.1 Motivation

In the previous chapter, we have described the main steps of the EcoPOST methodology (cf. Fig. 5.1). While EcoPOST evaluation models are easy to understand, the evolution of DCFs, their interference through ImFs, and the dynamic interplay of the different elements of an evaluation model are more difficult to comprehend. For this reason, we have also introduced the notion of simulation (cf. Fig. 5.1).

What we have not provided so far is a detailed description of the concepts underlying EcoPOST simulations. This chapter catches up this breach and takes a closer look at the simulation of evaluation models [130, 131].

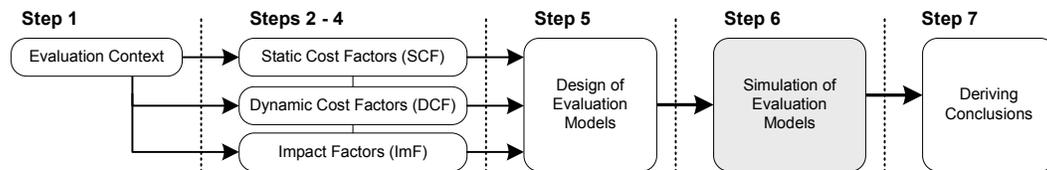


Figure 5.1: Step 6 of the EcoPOST Methodology.

We discuss simulation along four research questions:

- **Research Question 5-1:** Why do we need to simulate evaluation models?
- **Research Question 5-2:** What are the constituting elements of a simulation model?
- **Research Question 5-3:** How is a simulation model specified?
- **Research Question 5-4:** How is a simulation model executed?

Section 5.2 discusses in detail why simulation is needed in the EcoPOST framework (*Research Question 5-1*). Based on this Section 5.3 deals with the specification of EcoPOST simulation models (*Research Questions 5-2 and 5-3*). Section 5.4 illustrates the general computation of an EcoPOST simulation (*Research Question 5-4*). Section 5.5 gives two

illustrating examples. Finally, Section 5.6 discusses the advantages and disadvantages of simulation in the given context, and Section 5.7 concludes with a summary.

5.2 Need for Simulation

The evolution of DCFs, their interference through ImFs, and the dynamic interplay of the different elements of an evaluation model are difficult to comprehend. In this context, feedback loops are of particular importance.

A feedback loop is a *closed cycle* of causes and effects. Within this cycle, past events (e.g., the change of a DCF or dynamic ImF) are utilized to control future actions (e.g., another change of the same DCF or dynamic ImF). If end user fears increase, for example, this can result in a changing communication policy (e.g., an increased communication with end users). A changing communication policy, in turn, influences end user fears and causes them to change again. In other words, when a *change* occurs in a model variable, which is part of a feedback loop, this change is propagated around the loop [189].

As an example consider the feedback loop depicted in Fig. 4.14 (cf. Section 4.9.2, pp. 80) which deals with the impact of end user fears on the ability to redesign business processes. Basic to this model is a cyclic structure connecting the four dynamic ImFs "End User Fears", "Emotional Resistance", "Ability to Acquire Process Knowledge", and "Ability to Redesign Business Processes". As aforementioned, this evaluation model reflects the assumption that the introduction of a PAIS may cause end user fears, e.g., due to a high degree of job redesign. Such end user fears lead to increased emotional resistance. This, in turn, decreases the ability to get support from end users during process redesign and thus decreases the ability to effectively redesign business processes. Finally, a lower ability to redesign business processes can result in decreased end user fears. Reason is that end users will be less afraid of change if the ability to redesign processes decreases.

To understand the dynamic implications of a feedback loop, its polarity needs to be determined. Basically, two kinds of *loop polarities* are distinguished. *Positive* loops generate growth of DCFs and dynamic ImFs (cf. Fig. 5.2A). *Negative* loops, in turn, counteract and oppose growth (cf. Fig. 5.2B). If evaluation models contain both positive and negative feedback loops, more complex effects will result (cf. Fig. 5.2 C-E).

The polarity of a feedback loop can be determined by calculating the sign of a feedback loop [189]. For this purpose, the feedback loop needs to be opened at some point. As an example consider Fig. 5.2F which shows a feedback loop consisting of four variables x_1, \dots, x_4 . Assume that we open the loop at x_1 . This splits x_1 into an *input variable* (x_1^I) and an *output variable* (x_1^O). The feedback effect of a change in a variable as it is propagated around a loop is then defined as the (partial) derivative of x_1^O with respect to x_1^I . The polarity of a feedback loop is calculated as $\text{SGN}(\delta x_1^O / \delta x_1^I)$, where $\text{SGN}()$ is the *sign function*. This function returns +1 in case of positive loop polarity and -1 otherwise; if the sign function returns 0, there will be no loop.

Even if one knows the polarity of a feedback loop, related effects (e.g., the evolution of those DCFs and dynamic ImFs which are part of it) will not be easy to understand [64, 147]. Simultaneously tracing the effects of several feedback loops in the same evaluation model

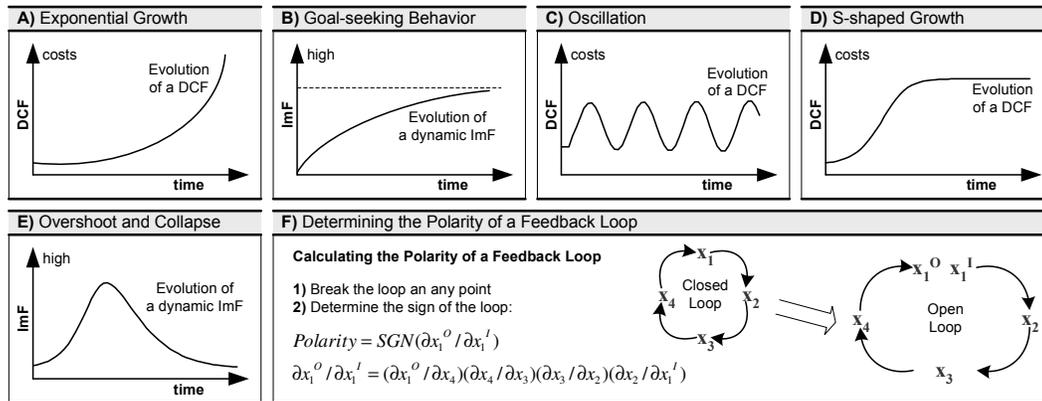


Figure 5.2: Feedback in Evaluation Models – Overview of Potential Dynamic Effects.

is even more difficult. Generally, the dynamic interplay of the different elements of an evaluation model and the forces and actions that act upon these models (e.g., fears, delays, resistance to change) are complex. To investigate this dynamic interplay and to unfold potential effects, the application of simulation is a suitable aid.

5.3 Specifying a Simulation Model

This section introduces the constituting elements of an EcoPOST simulation model (cf. *Research Questions 5-2 and 5-3*). In particular, this section describes how to formally specify simulation models.

5.3.1 Constituting Elements

To enable simulations it becomes necessary to formally specify the behavior of evaluation models by means of a *simulation model*. A simulation model defines the behavior of an evaluation model based on *mathematical equations* (cf. *Research Question 5-2*). Thereby, the behavior of each model variable is specified by exactly one equation. The equation describes how the variable is changing over time from simulation start.

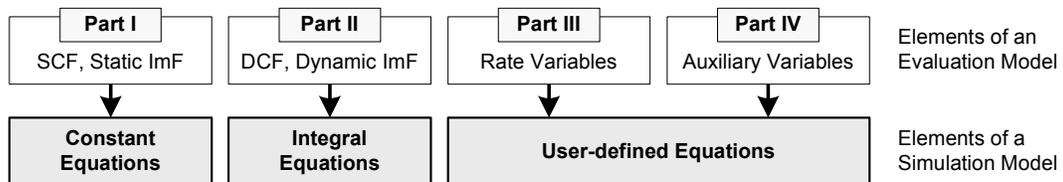


Figure 5.3: Elements of a Simulation Model.

If a model variable represents an SCF or static ImF, the equation will specify a constant value for the model variable. If a model variable represents a DCF, a dynamic ImF, or a

rate or auxiliary variable, the equation will specify how the values of the respective model variable change during simulation. To make this more clear, we characterize the elements of a simulation model in detail (cf. *Research Question 5-3*).

5.3.2 Specifying SCFs and static ImFs

SCFs and static ImFs are specified by single numerical values in *constant equations*. In Fig. 5.4A, for example, two SCFs "Costs for Modeling Tool" and "Process Analysis Costs" are depicted. The SCF "Costs for Modeling Tool", for instance, is specified by the constant equation $\text{Costs for Modeling Tool} = 1000\$$. Both SCFs are then used to calculate another SCF¹ "Process Modeling Costs".

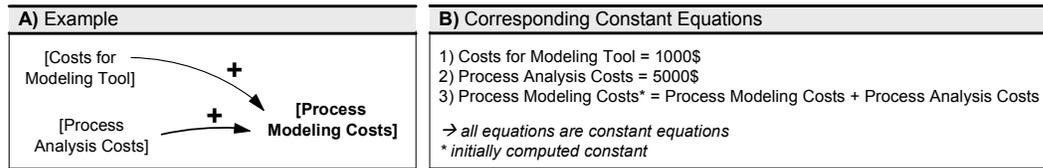


Figure 5.4: Constant Equations.

Note that the SCF "Process Modeling Costs" is an *initially computed constant*, i.e., it is specified in terms of other constants (cf. Fig. 5.4B). In other words, it depends on the two SCFs "Costs for Modeling Tool" and "Process Analysis Costs". Its value will change if a new value is assigned to one of the preceding SCFs. Constant equations are computed only once at the beginning of a simulation.

5.3.3 Specifying DCFs and dynamic ImFs

The evolution of DCFs and dynamic ImFs is characterized by *integral equations* (cf. Fig. 5.5) [64]. Integral equations enable us to capture the accumulation of DCFs and dynamic ImFs from the start of a simulation run (t_0) to its end (t).

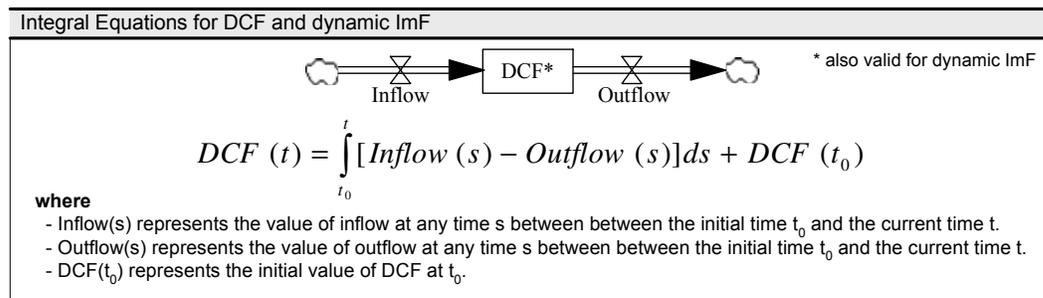


Figure 5.5: Integral Equations.

¹As defined in Section 4.6 SCFs can be composed of subsidiary SCFs.

Definition 5.1 (Integral Equation) Let EM be an evaluation model (cf. Definition 4.3) and let S be the set of all DCFs and dynamic ImFs defined by EM . An integral equation for a DCF or dynamic ImF $v \in S$ is defined as follows:

$$v(t) = \int_{t_0}^t [inflow(s) - outflow(s)]ds + v(t_0) \text{ where}$$

- t_0 denotes the starting time of the simulation run,
- t represents the end time of the simulation run,
- $v(t_0)$ represents the value of v at t_0 ,
- $inflow(s)$ represents the value of the inflow at any time s between t_0 and t ,
- $outflow(s)$ represents the value of the outflow at any time s between t_0 and t .

Integral equations specify how DCFs and dynamic ImFs integrate their *net flow*. During any interval $[t_1, t_2]$ the net flow is the area bounded by the graph of the *net rate*, i.e., the difference of all inflows and outflows between the start and the end of an interval (cf. Fig. 5.6A). Thus, in Fig. 5.6B the value of DCFs and dynamic ImFs at t_2 is calculated as the sum of its value at t_1 (i.e., S_1) and the area bounded by the graph of the net rate between t_1 and t_2 (cf. Fig. 5.6A); the value at t_2 is S_2 .

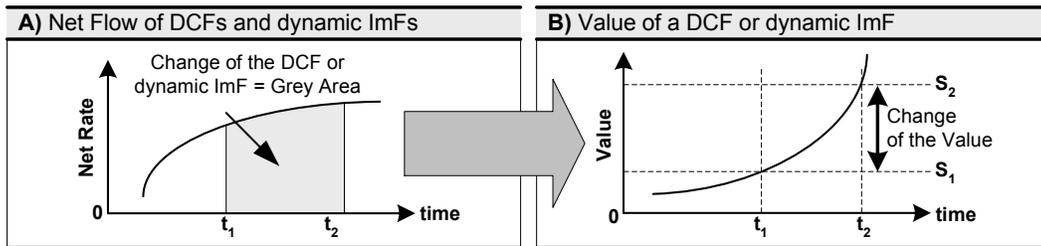


Figure 5.6: Integrating the Net Flow of DCFs and dynamic ImFs.

Concrete examples are given in Fig. 5.7A and Fig. 5.7B. Fig. 5.7A denotes an arbitrary DCF accumulating the net flow generated from additional costs (specified by the rate variable "Cost Increase") and cost savings (specified by the rate variable "Cost Decrease"). Likewise, Fig. 5.7B deals with the accumulation of an arbitrary dynamic ImF.

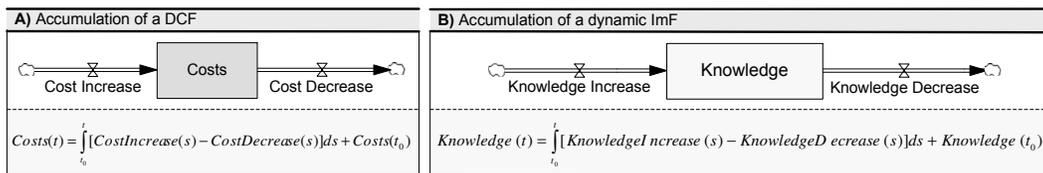


Figure 5.7: Integral Equations (Examples).

So far, we have described how static as well as dynamic cost/impact factors are specified in simulation models. Besides, our evaluation models also comprise rate and auxiliary variables. These model elements have to be considered in simulation models as well.

5.3.4 Specifying Rate and Auxiliary Variables

The common discussion of rate and auxiliary variables becomes possible as both types of variables are specified in the same way, namely as user-defined, arbitrary functions defined over their predecessor variables. In other words, both rate and auxiliary variables are used to merge static and dynamic cost/impact factors of an evaluation model.

Specifying Rate Variables. Rate variables are assigned to inflows and outflows in order to control them (cf. Section 4.9.1). Picking up the metaphor from Section 4.9.1, rate variables assigned to *inflows* can be regarded as "water taps" which specify the increase of DCFs and dynamic ImFs. Rate variables assigned to *outflows*, in turn, can be considered as "drains" which control the decrease of DCFs and dynamic ImFs. During simulation, values of rate variables are dynamic, i.e., they change along the course of time. Reason is that they are not only influenced by SCFs and static ImFs, but also by evolving DCFs and dynamic ImFs. The behavior of rate variables is specified by *rate equations*.

Definition 5.2 (Rate Equation) Let EM be an evaluation model (cf. Definition 4.3) and let R be the set of all rate variables defined by EM . A rate equation for a rate variable $v \in R$ is a user-defined function $f(v_1, \dots, v_n)$ over v_1, \dots, v_n with v_1, \dots, v_n being the predecessors of $v \in EM$ ($i = 1, \dots, n$).

In Fig. 5.8A the equation for the rate variable "Cost Growth Rate" is build upon the two preceding SCFs "Costs for Implementing Business Functions" and "Costs for Implementing Process Logic". The rate equation adds up the two SCFs (i.e., it constitutes an additive equation formulation). In Fig. 5.8B, the rate equation is build from one SCF and from one static ImF and is an example of *multiplicative* equation formulation.

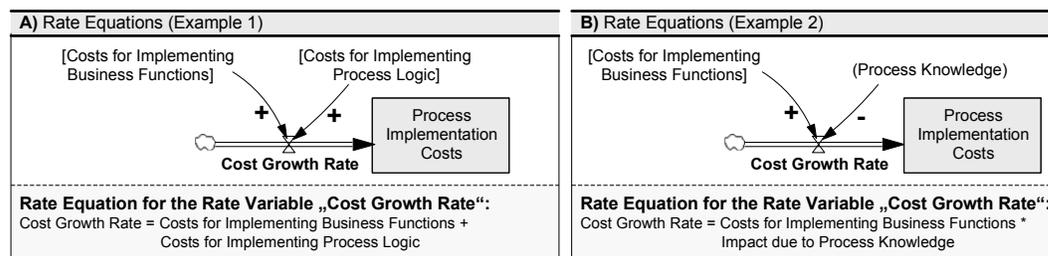


Figure 5.8: Rate Equations.

Specifying Auxiliary Variables. Like rate variables, auxiliary variables are used to merge static and dynamic cost/impact factors. Hence, *auxiliary equations* are constructed in the same way like rate equations.

Definition 5.3 (Auxiliary Equation) Let EM be an evaluation model (cf. Definition 4.3) and let A be the set of all auxiliary variables defined by EM . An auxiliary equation for an auxiliary variable $v \in A$ is a user-defined function $f(v_1, \dots, v_n)$ over v_1, \dots, v_n with v_1, \dots, v_n being the predecessors of $v \in EM$ ($i = 1, \dots, n$).

In Fig. 5.9 the auxiliary variable "Adjusted Process Modeling Costs" is influenced by one SCF "Planned Process Modeling Costs" and by two dynamic ImFs "Process Knowledge" and "End User Fears". The auxiliary equation from Fig. 5.9 is another example of a *multiplicative* equation formulation.

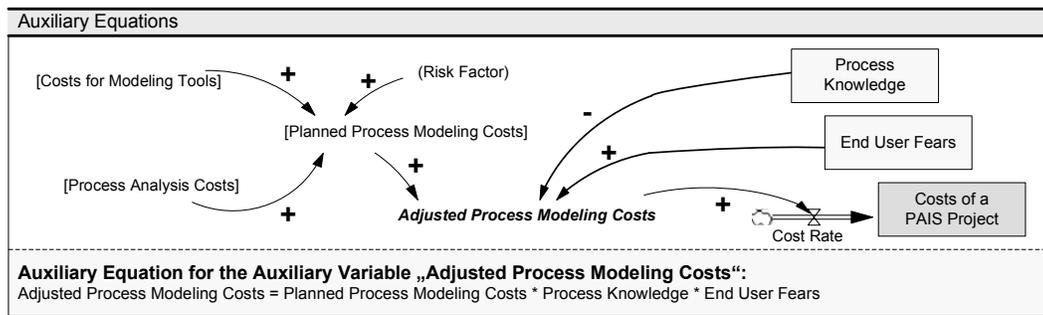


Figure 5.9: Auxiliary Equation.

So far, we have shown how to specify equations in EcoPOST simulation models (cf. *Research Questions 5-2 and 5-3*). What we still have to discuss is a special type of auxiliary equation, namely, table functions to specify those auxiliary variables which realize nonlinear dependencies. In the following, we pick up this issue.

Specifying Table Functions. In Section 4.9.1 we have mentioned that auxiliary variables can be also used to represent nonlinear dependencies between impact and cost factors. Consider Fig. 5.10 which shows both a linear (cf. Fig. 5.10A) and a nonlinear (cf. Fig. 5.10B) dependency between a static ImF "Process Complexity" and a cost factor "Process Analysis Costs". When a nonlinear dependency is assumed, this nonlinearity is represented by auxiliary variables like "Impact due to Process Complexity" in Fig. 5.10C.

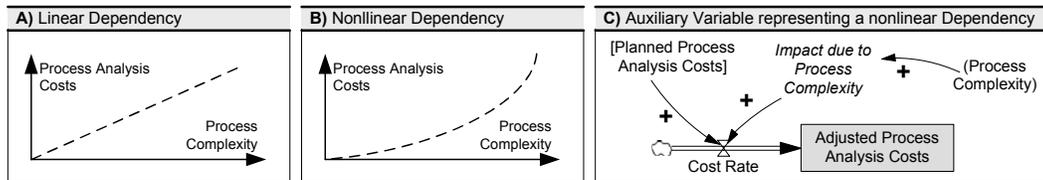


Figure 5.10: Auxiliary Variables for Representing Nonlinear Dependencies.

For specifying such auxiliary variables in simulation models we use simple *table functions*. A table function complies with Definition 5.3, i.e., it is a user-defined function over the

preceding variables of the considered one – although there is only one preceding variable in this case.

Specifically, a table function transfers the *input value* from the ingoing variable into an *output value* (called *impact rating* in our approach). Thereby, the table function is defined based on a set (or table) of predefined *reference values*. Linear interpolation is used for reproducing the values lying between the specified reference values.

Definition 5.4 (Table Function) *Let EM be an evaluation model (cf. Definition 4.3) and let X be the preceding variable of an auxiliary variable Y which represents a nonlinear dependency in EM. The auxiliary equation for Y is a user-defined function $Y = f(X)$ called table function if the following holds:*

- $Y = \text{Effect of } X \text{ on } Y$,
- $\text{Effect of } X \text{ on } Y = \text{Table for Effect of } X \text{ on } Y(X)$,
- $\text{Table for Effect of } X \text{ on } Y = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$,
where (x_i, y_i) represents each pair of reference values defining the relationship.

In other words, the output value Y is calculated based on the input value X through table function f . Fig. 5.11 illustrates the specification of table functions in *Vensim* [208], the modeling and simulation tool EcoPOST uses (cf. Chapter 8).

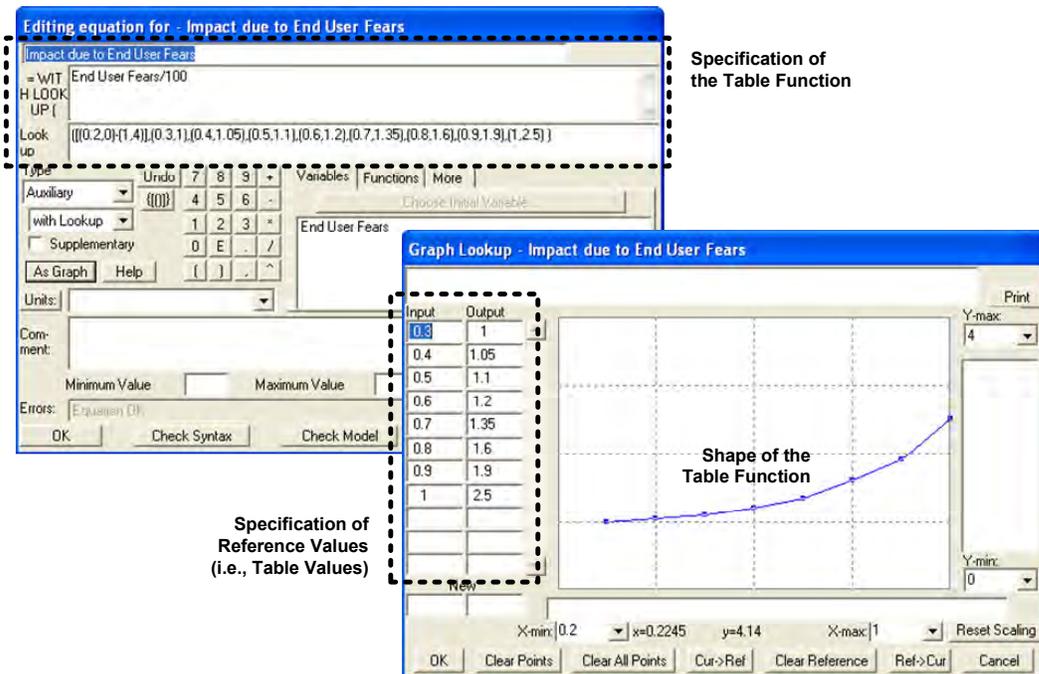


Figure 5.11: Specifying Table Functions and Reference Values in *Vensim* [208].

In our simulation models, the syntax for representing table functions is as follows:

$$\text{VARIABLE} = \text{WITH LOOKUP} (\text{INPUT VARIABLE}, ([\text{point}_{min} - \text{point}_{max}], \text{point}, \dots))$$

Here, VARIABLE denotes the name of the auxiliary variable used to represent the table function. The input value to the table function is INPUT VARIABLE (which can be a SCF, a DCF, a static ImF, a dynamic ImF, or another auxiliary variable). Separated by a comma follows the specification of the table function as a *list of elements*. The first element specifies the range relevant for the table function: point_{min} specifies the lower left corner and point_{max} the upper right corner of the coordinate system in which the table function is defined. The remaining elements specify the table function's curve. The value of VARIABLE results from applying the table function to INPUT VARIABLE.

Illustrating Examples. Fig. 5.12 gives examples of typical table functions. Dependent on the value of an ImF (i.e., on the input value X) a specific *impact rating* is derived (i.e., an output value Y). If used in a rate or auxiliary equation (merging static and dynamic cost/impact factors), an impact rating of less than 1 will decrease costs (cf. Fig. 5.12A), while an impact rating larger than 1 will increase costs (cf. Fig. 5.12B and Fig. 5.12C). An impact rating equal to 1 will neither increase nor decrease costs.

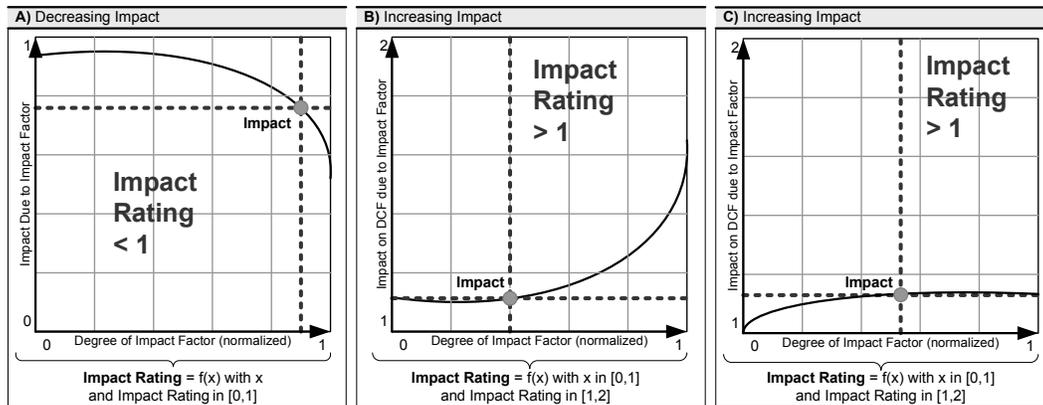


Figure 5.12: Table Functions for Quantifying Impact Factors.

Information needed for specifying table functions can be derived from different sources including practical fieldwork, case study research, and software experiments. Obviously, generating respective data is associated with high efforts. Hence, there exists no standard way for deriving table functions, though a "best practice" guideline for formulating table functions is given in [189].

Picking up our example from Fig. 5.10 we exemplarily describe how the table function for the auxiliary variable "Impact due to Process Complexity" can be defined:

1. **Normalize the impact and the impact rating.** In a first step, the ImF "Process Complexity" (i.e., the input value X) is normalized. This simplifies the definition of reference values of the table function (e.g., of extreme values). The normalized input

value is a dimensionless ratio of the original input value and an additional reference value X' (e.g., 100%). Likewise, the output value (i.e., the calculated impact rating) is a dimensionless value. In our example, the input variable "Process Complexity" shall range from "low" (0%) to "high" (100%), while the potential impact rating shall range from "no impact" (corresponding to an impact rating of "1") to "high impact" (corresponding to an impact rating of "1.4"). The upper value means that process complexity can increase process analysis costs by a maximum of 40%.

2. **Identify reference values.** In a second step, reference values need to be specified, i.e., a set of values which determine the table function by definition. Thereby, it must be ensured that the defined reference values make sense, i.e., they have to conform to existing reference policies and context-specific constraints. Finally, a defined table function needs to be calibrated based on test simulation runs and sensitivity analysis.

Closing Remarks. Unlike constant and integral equations, the specification of rate and auxiliary equations is a difficult task to accomplish. When specifying rate and auxiliary equations, two kinds of formulations are often used in this context (examples have been already given in Fig. 5.8 and Fig. 5.9).

First, *multiplicative formulations* calculate the value of a variable Y ($Y \in R \cup A$) by multiplying the sum of all preceding SCFs and DCFs with all preceding ImFs:

$$Y = (\sum_{i=1}^m SCF_i + \sum_{j=1}^n DCF_j) * \prod_{k=1}^l ImF_k$$

As an example consider the rate variable "Cost Growth Rate" in Fig. 5.13A. The sum of the two SCFs "Planned Process Analysis Costs" and "Tool Costs" is multiplied with the two static ImFs "Process Complexity" and "End User Fears".

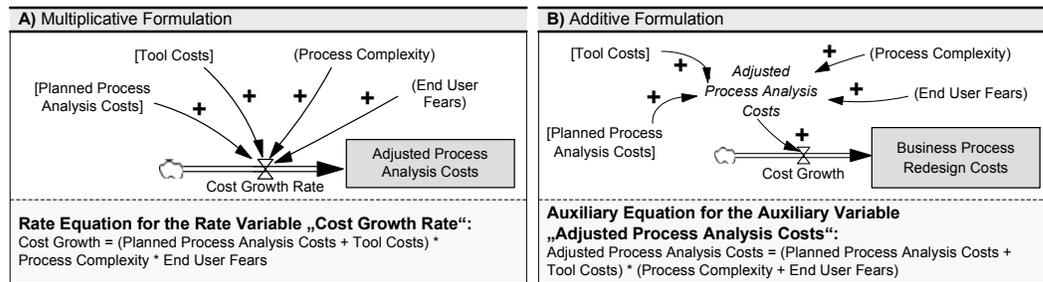


Figure 5.13: Multiplicative and Additive Equation Formulation.

Second, *additive formulations* can be used as well. Then, the value of the variable Y ($Y \in R \cup A$) is determined by multiplying the sum of all preceding SCFs and DCFs with the sum of all preceding ImFs:

$$Y = (\sum_{i=1}^m SCF_i + \sum_{j=1}^n DCF_j) * (\sum_{k=1}^l ImF_k)$$

As example consider the auxiliary variable "Adjusted Process Analysis Costs" in Fig. 5.13B. The sum of the two SCFs "Planned Process Analysis Costs" and "Tool Costs" is multiplied with the sum of the two static ImFs "Process Complexity" and "End User Fears".

It is important to mention that additive formulations are typically used in cases in which the effects of one ImF is supposed to dominate the effects of other ImFs.

The total set of equations of a given evaluation model is denoted as *simulation model*:

Definition 5.5 (Simulation Model) *Let EM be an evaluation model (cf. Definition 4.3) and let T be the simulated time domain. A tuple SM := (EM, func) is called the simulation model of an evaluation model EM, if the following holds:*

– $func : T \times (S \cup R \cup A \cup C) \mapsto \mathbb{R}$ with

$$func(t, v) = \left\{ \begin{array}{l} \int_{t_0}^t [inflow(s) - outflow(s)] ds + v(t_0) \text{ if } v \in S, \\ \text{where} \\ \quad - t_0 \text{ denotes the starting time of the simulation,} \\ \quad - t \text{ represents the end time of the simulation,} \\ \quad - v(t_0) \text{ represents the value of } v \text{ at } t_0, \\ \quad - inflow(s) \text{ represents the value of the inflow at any} \\ \quad \text{time } s \text{ between } t_0 \text{ and } t, \\ \quad - outflow(s) \text{ represents the value of the outflow at any} \\ \quad \text{time } s \text{ between } t_0 \text{ and } t, \\ f(v_1, \dots, v_n) \text{ if } v \in R \text{ and } v \text{ has predecessors } v_1, \dots, v_n \in EM \\ \quad \text{with } v_i \in (S \cup A \cup C), (i = 1, \dots, n) \\ \quad \text{where } f(v_1, \dots, v_n) \text{ is a user-defined function over } v_1, \dots, v_n, \\ f(v_1, \dots, v_n) \text{ if } v \in A \text{ and } v \text{ has predecessors } v_1, \dots, v_n \in EM \\ \quad \text{with } v_i \in (S \cup A \cup C), (i = 1, \dots, n) \\ \quad \text{where } f(v_1, \dots, v_n) \text{ is a user-defined function over } v_1, \dots, v_n, \\ v \text{ if } v \in C, \\ \text{UNDEFINED otherwise.} \end{array} \right.$$

So far, we have addressed *Research Questions 5-2* and *5-3* and have introduced the elements of our simulations models as well as their formal definition.

5.4 Computing a Simulation Model

What we have neglected so far are details on the execution of simulation models. This section picks up this issue and describes what happens when a simulation model is executed (cf. *Research Question 4*).

Performing Simulations. A simulation corresponds to a step-by-step numerical solution of the mathematical equations of the corresponding simulation model. In other words, the equations define how the variables of an evaluation model change over time.

Consider Fig. 5.14 which depicts the simulation of a *DCF* and a dynamic *ImF*. The values at time t_0 have been calculated and the values at time t_1 are now determined. *DT* stands for "Difference in Time" and denotes the length of the time interval between two calculated conditions. $DCF.t_0$ and $ImF.t_0$ designate the two values of *DCF* and *ImF* at time t_0 (cf. Fig. 5.14A). $R1.[t_0, t_1[$ is a rate variable specifying the inflow of $DCF.t_0$ within the time interval $[t_0, t_1[$. Similarly, the rate variables $R2.[t_0, t_1[$ and $R3.[t_0, t_1[$ specify the inflow respectively outflow of $ImF.t_0$ within time interval $[t_0, t_1[$. Therewith, all information needed to compute the new values of *DCF* and *ImF* is available.

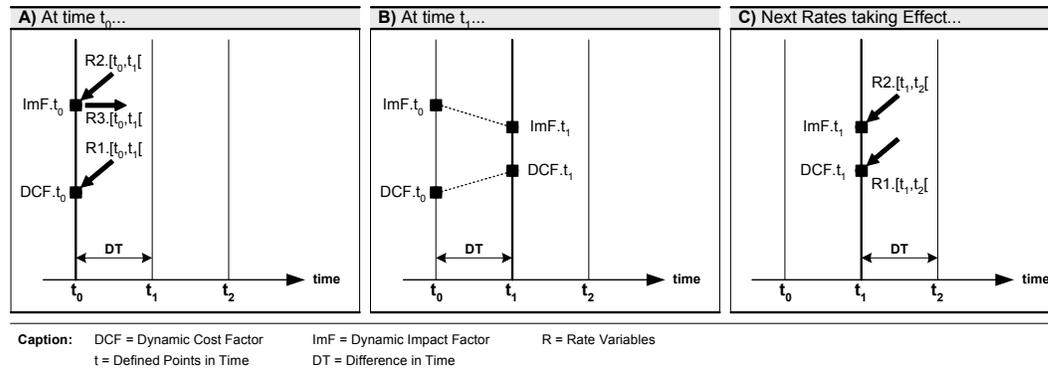


Figure 5.14: Computing a Simulation Model.

Within the time interval $[t_0, t_1[$, the rate variables act on $DCF.t_0$ and $ImF.t_0$ and cause them to change. The new values of $DCF.t_1$ and $ImF.t_1$ at time t_1 are calculated by adding and subtracting the changes represented by these rates (cf. Fig. 5.14B). Finishing the computation creates the situation shown in Fig. 5.14B. In the following, only these values are needed to compute the forthcoming rates for the $[t_1, t_2[$ interval (cf. Fig. 5.14C).

Sensitivity Analysis. Each step-by-step numerical solution corresponds to one simulation run. To perform another simulation run, another step-by-step computation has to be conducted. This enables the accomplishment of "behavioral experiments" based on a series of simulation runs. During these runs the equations of the simulation model are manipulated in a controlled manner to systematically investigate the effects of changed simulation parameters. Therewith, it becomes possible to accomplish sensitivity analysis, i.e., to investigate how the output of a simulation will vary if the initial values of a simulation are changed.

5.5 Illustrating Examples

In this section, we give two examples to illustrate the previously described simulation concepts. The first one deals with the simulation of a *DCF* "Business Process Redesign Costs" and its interference through the *ImF* "End User Fears" (cf. Section 5.5.1). The latter one simulates the costs for maintaining the process logic of a PAIS (cf. Section 5.5.2).

5.5.1 Costs for Business Process Redesign

Fig. 5.15A shows a simple evaluation model. Assume that the evolution of the DCF "Business Process Redesign Costs" (caused by the dynamic ImF "End User Fears") shall be analyzed (ignoring other potential ImFs). The model reflects the assumption that the redesign of business processes (e.g., prior to the introduction of a PAIS) may be influenced by end user fears (caused by a high degree of job redesign or changed social clues). Such end user fears can lead to emotional resistance of users, and, in turn, to a lack of support from users while redesigning business processes, e.g., during an interview-based process analysis.

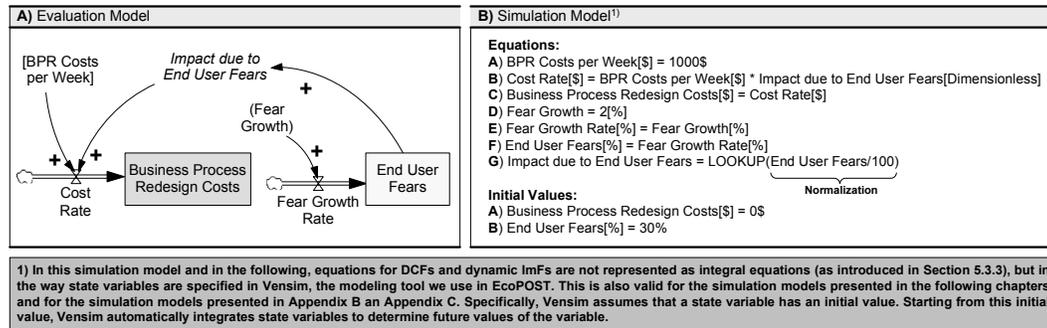


Figure 5.15: Evaluation and Simulation Model.

Assume that the business process redesign activities are scheduled for 32 weeks. To simulate the evolution of the resulting costs along this time frame, we use the simulation model depicted in Fig. 5.15B². Here, the nonlinear impact of end user fears on the costs of business process redesign is represented through a table function. Fig. 5.16A shows selected values of model variables when the simulation model is executed. The underlying principles of this computation have been already described in Section 5.4.

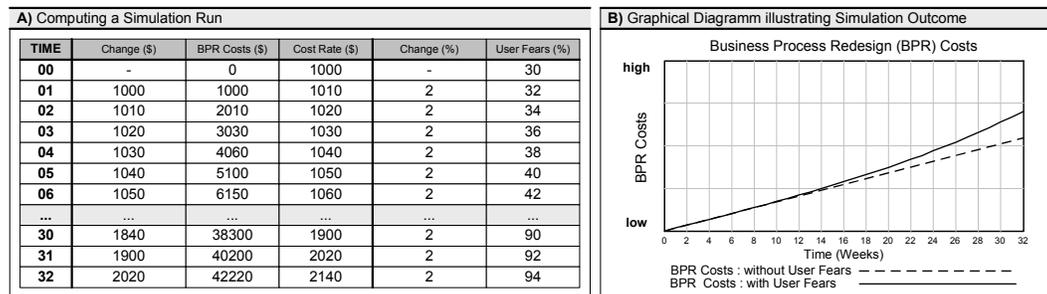


Figure 5.16: Simulation Outcomes.

Finally, Fig. 5.16B shows a graphical diagram which illustrates the outcome of the simulation. As can be seen, there is a significant negative impact of end user fears on the costs of

²Note that equations for DCFs and dynamic ImFs are presented in the way they are specified in Vensim [208], the tool we use in EcoPOST, and not as integral equations. This is also valid for simulation models presented in the following chapters and for the simulation models presented in Appendix A and Appendix B.

business process redesign, i.e., simulation results show that end user fears result in higher business process redesign costs.

5.5.2 Costs for Maintaining Process Logic

Assume that the current implementation of a business process is based on heterogeneous IS, email communication, phone calls, and the exchange of complex excel sheets. This results in problems such as varying process executions, unclear process responsibilities, and difficulties in synchronizing activities. To overcome these problems a new IS shall provide integrated process support.

We distinguish between two implementation scenarios. A first option is to implement a conventional IS with the process logic hard-coded in the application code (*Scenario 1*). The implementation of such an IS would result in development costs of 10000\$. A second option is to realize a PAIS and the separation of process logic from the application code (*Scenario 2*). Implementing a PAIS (realized on top of a WfMS) would result in development costs of 15000\$. Considering these alternatives, it is unclear, which scenario will be advantageous, in particular, if one takes maintenance costs of the candidate solutions into account as well. Thereby, the adaptation of process changes is a particular dominating cost factor as process changes may require to adapt all implementation parts related to the changed process part (which, in turn, can cause significant efforts).

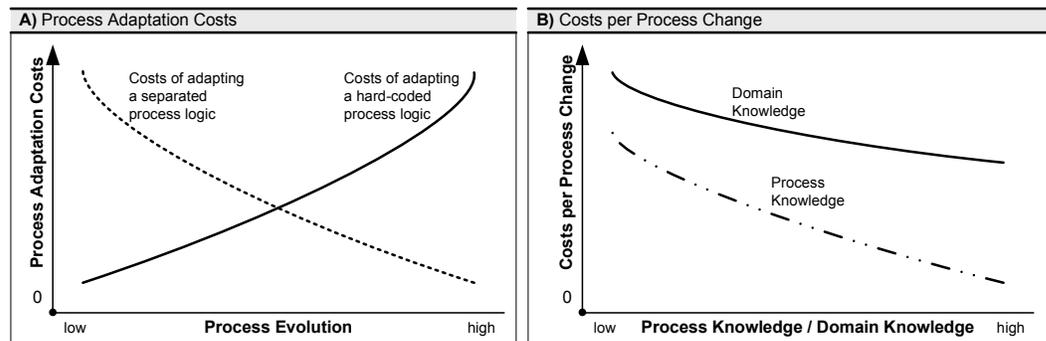


Figure 5.17: Assumed Process Adaptation Costs.

Assumptions. In Scenario 1, each process change requires the adaptation of all business functions including parts of the hard-coded process logic to be changed. This may require considerable programming efforts. In Scenario 2, by contrast, the availability of explicit process models enables process changes at a high semantic level as all affected process fragments can be adapted by manipulating the process model (e.g., using a process modeling tool). Thus, code interventions can be significantly reduced though often not completely avoided (cf. Fig. 5.17A). Despite the need for intense code interventions, it is our working hypothesis that Scenario 1 will be advantageous if the process never or only rarely changes, i.e., costs for realizing process changes do not exceed those 5000\$ saved in case of a PAIS implementation. Likewise, Scenario 2 will be advantageous if there are frequent process changes. In such a case, the higher costs of realizing the PAIS are justified by lower costs

for process changes. In our example, the behavior of the DCF "Process Adaptation Costs" and its interference through process knowledge and domain knowledge (cf. Fig. 5.17B) shall be investigated in detail – for both low and high process evolution.

Evaluation Model. Fig. 5.18 shows the evaluation model we use to analyze the given evaluation context. This evaluation model reflects the assumption that the overall process adaptation costs can be calculated as the sum of all costs for single process adaptations. Specifically, the evaluation model comprises one DCF "Process Adaptation Costs" and two dynamic ImFs "Process Knowledge" and "Domain Knowledge".

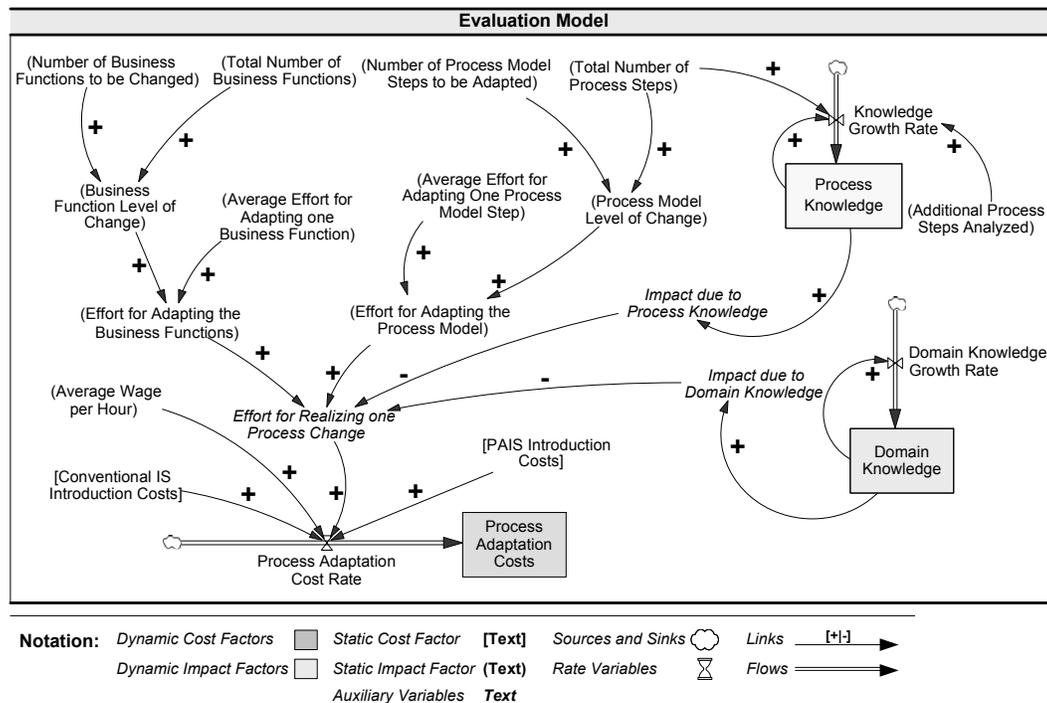


Figure 5.18: Evaluation Model.

The DCF "Process Adaptation Cost" (respectively the "Process Adaptation Cost Rate") is determined by two SCFs ("Conventional IS Introduction Costs" and "PAIS Introduction Costs"), one auxiliary variable ("Effort for Realizing one Process Change"), and one static ImF ("Average Wage per Hour"). The auxiliary variable "Effort for Realizing one Process Change" aggregates the effort for realizing a process change. Thereby, we distinguish between efforts for adapting business functions (with the process logic to be changed hard-coded within these business functions) and efforts for adapting a process model. Note Scenario 1 implies only efforts for changing the business functions, while Scenario 2 necessitates adaptations of both business functions and the process model.

Further, the auxiliary variable "Effort for Realizing one Process Change" is influenced by the two dynamic ImFs "Process Knowledge" and "Domain Knowledge" which both have a nonlinear impact on respective efforts. This nonlinear impact is described by two table

functions (respectively by the two additional auxiliary variables "Impact due to Process Knowledge" and "Impact due to Domain Knowledge"). Process adaptation efforts are finally expressed in costs based on the SCF "Average Wage per Hour".

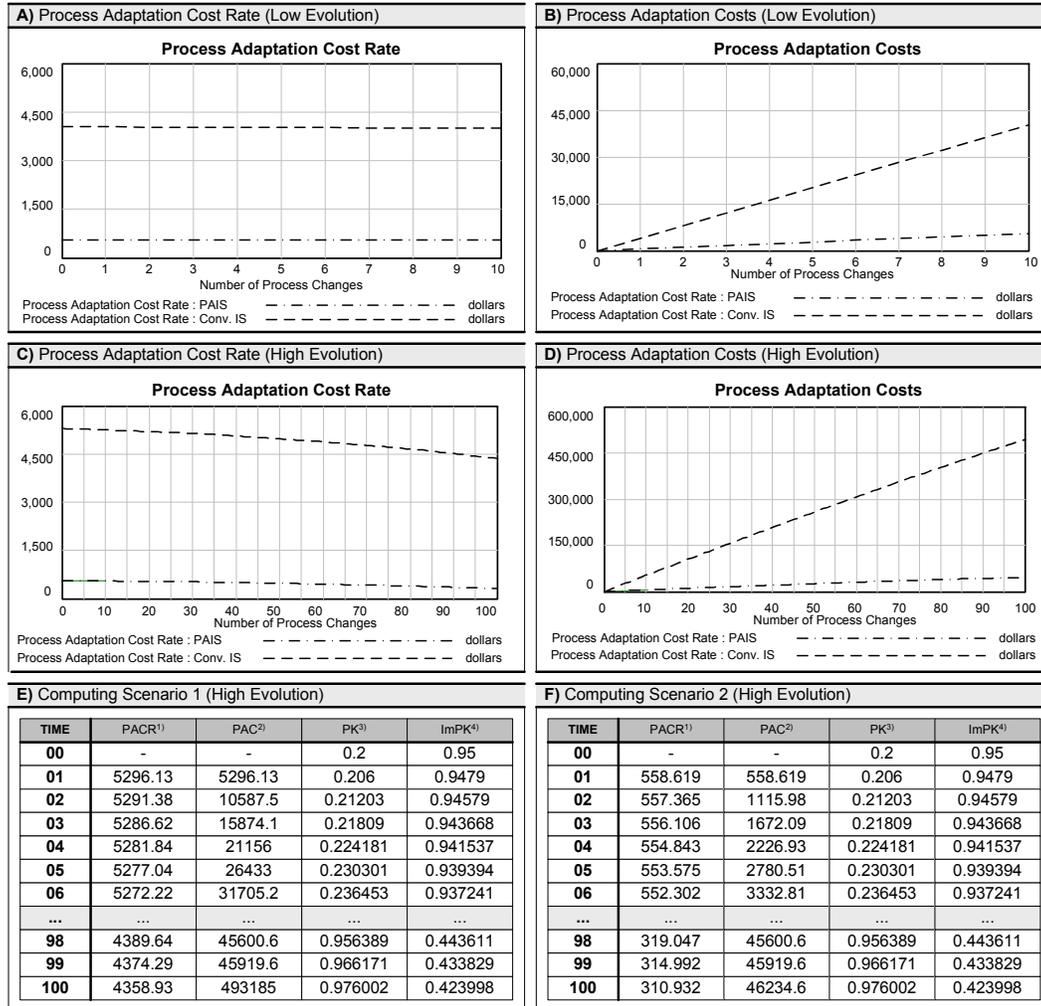
Simulation Model. Fig. 5.19A shows the simulation model for Scenario 1, Fig. 5.19B the one for Scenario 2. For Scenario 1, the average number of business functions to be adapted in case of a process change shall be 35 (from a maximum of 100). For Scenario 2, the average number of business functions to be adapted shall be 3. Additionally, 25 steps of the process model need to be adapted. The adaptation of one business function results in efforts of 2h, whereas the adaptation of a process model step results only in efforts of 0.5h. Note that empirical research described by Kleiner [99, 100] supports these measures.

Simulation Model for Scenario 1 (Conventional IS)	Simulation Model for Scenario 2 (PAIS)
<p>A) Adaptation of Business Functions:</p> <ol style="list-style-type: none"> 1) Average Number of Business Functions to be Changed = 35 2) Total Number of Business Functions = 100 3) Business Function Level of Change [%] = Average Number of Business Functions to be Changed/ Total Number of Business Functions 4) Average Effort for Adapting one Business Function [h] = 2h 5) Effort for Adapting the Business Functions [h] = (Business Function Level of Change*100) * Average Effort for Adapting one Business Function <p>B) Adaptation of Process Model:</p> <ol style="list-style-type: none"> 1) Average Number of Process Model Steps to be Changed = 0 2) Total Number of Process Model Steps = 100 3) Process Model Level of Change [%] = Average Number of Process Model Steps to be Adapted/ Total Number of Process Steps 4) Average Effort for Adapting one Process Model Step [h] = 0.5h 5) Effort for Adapting the Process Model [h] = 0 <p>C) Dynamic Cost Factor:</p> <ol style="list-style-type: none"> 1) Average Wage per Hour [\$] = 25\$ 2) Conventional IS Introduction Costs [\$] = 10000\$ 3) PAIS Introduction Costs [\$] = 0\$ 4) Impact due to Process Knowledge = LOOKUP(Process Knowledge) 5) Impact due to Domain Knowledge = LOOKUP(Domain Knowledge) 6) Process Adaptation Cost Rate [\$] = (Effort for Realizing one Process Change*Average Wage per Hour)+(Conventional IS Introduction Costs*Average Wage per Hour) + PAIS Introduction Costs 7) Process Adaptation Costs [\$] = Process Adaptation Cost Rate <p>D) Impact Factors:</p> <ol style="list-style-type: none"> 1) Process Knowledge = Process Knowledge Growth Rate 2) Process Knowledge Growth Rate = (Additional Process Steps Analyzed/Total Number of Process Steps)+(Process Knowledge*0.005) 3) Additional Process Steps Analyzed 4) Domain Knowledge = Domain Knowledge Growth Rate 5) Domain Knowledge Growth Rate = Domain Knowledge*0.005 <p>E) Initial Values:</p> <ol style="list-style-type: none"> 1) Process Adaptation Costs [\$] = 0\$ 2) Process Knowledge [%] = 20% 2) Domain Knowledge [%] = 40% 	<p>A) Adaptation of Business Functions:</p> <ol style="list-style-type: none"> 1) Average Number of Business Functions to be Changed = 3 2) Total Number of Business Functions = 100 3) Business Function Level of Change [%] = Average Number of Business Functions to be Changed/ Total Number of Business Functions 4) Average Effort for Adapting one Business Function [h] = 2h 5) Effort for Adapting the Business Functions [h] = (Business Function Level of Change*100) * Average Effort for Adapting one Business Function <p>B) Adaptation of Process Model:</p> <ol style="list-style-type: none"> 1) Average Number of Process Model Steps to be Changed = 25 2) Total Number of Process Model Steps = 100 3) Process Model Level of Change [%] = Average Number of Process Model Steps to be Adapted/ Total Number of Process Steps 4) Average Effort for Adapting one Process Model Step [h] = 0.5h 5) Effort for Adapting the Process Model [h] = Average Effort for Adapting One Process Model Step*(Process Model Level of Change*100) <p>C) Dynamic Cost Factor:</p> <ol style="list-style-type: none"> 1) Average Wage per Hour [\$] = 25\$ 2) Conventional IS Introduction Costs [\$] = 0\$ 3) PAIS Introduction Costs [\$] = 15000\$ 4) Impact due to Process Knowledge = LOOKUP(Process Knowledge) 5) Impact due to Domain Knowledge = LOOKUP(Domain Knowledge) 6) Process Adaptation Cost Rate [\$] = (Effort for Realizing one Process Change*Average Wage per Hour)+(Conventional IS Introduction Costs*Average Wage per Hour) + PAIS Introduction Costs 7) Process Adaptation Costs [\$] = Process Adaptation Cost Rate <p>D) Impact Factors:</p> <ol style="list-style-type: none"> 1) Process Knowledge = Process Knowledge Growth Rate 2) Process Knowledge Growth Rate = (Additional Process Steps Analyzed/Total Number of Process Steps)+(Process Knowledge*0.005) 3) Additional Process Steps Analyzed 4) Domain Knowledge = Domain Knowledge Growth Rate 5) Domain Knowledge Growth Rate = Domain Knowledge*0.005 <p>E) Initial Values:</p> <ol style="list-style-type: none"> 1) Process Adaptation Costs [\$] = 0\$ 2) Process Knowledge [%] = 20% 2) Domain Knowledge [%] = 40%

Figure 5.19: Simulation Model.

Process knowledge is assumed to increase by systematically analyzing the constituent activities (or steps) of the process to be supported, e.g., based on interviews with process participants. Hence, we calculate process knowledge as the percentage of process activities

that have been explicitly analyzed. Thereby, existing knowledge can influence the generation of additional knowledge. Domain knowledge, by contrast, is determined by the time a software engineer has worked in a specific domain. Thus, domain knowledge increases with the time a software engineer has worked in the considered evaluation scenario.



1) Process Adaptation Cost Rate
 2) Process Adaptation Costs
 3) Process Knowledge
 4) Impact due to Process Knowledge

Figure 5.20: Simulation Outcomes.

Simulation Outcomes. Fig. 5.20A and Fig. 5.20B illustrate the results for low process evolution (i.e., 10 process changes). As can be seen, process adaptation costs are significantly higher for Scenario 1 than for Scenario 2. Fig. 5.20C and Fig. 5.20D illustrate the results for high process evolution (i.e., 100 process changes). Again, process adaptation costs are higher for Scenario 1 than for Scenario 2. Finally, Fig. 5.20E and Fig. 5.20F show selected values of the evaluation model (of the DCF and one of the dynamic ImF) over time when the simulation model is executed.

Conclusion. Altogether, simulation results show that using a PAIS is recommendable (considering the given evaluation context) in any case. The higher costs of Scenario 2 are clearly justified by the more effective process adaptations. The simulation outcomes also show that the impact of process knowledge and domain knowledge is negligible.

5.6 Discussion

In practice, it is not always obvious which actions or forces cause a DCF or dynamic ImF to change. To investigate the evolution of DCFs and dynamic ImFs, PAIS engineers can simulate evaluation models. In particular, they can perform sensitivity analysis and behavioral experiments by changing selected simulation parameters in a controlled way. So, simulation enables PAIS engineers to gain insights into causal dependencies between organizational, technological, and project-specific factors, to better understand resulting effects, and to develop a concrete "feel" for the dynamic implications of EcoPOST evaluation models.

Despite the fact that one can perform sensitivity analysis by systematically varying some (or all) parameters of a simulation model, the overall number of simulation runs is typically restricted due to time or cost limitations. In fact, simulation tends to be relatively expensive, i.e., a lot of resources (in terms of manpower, money, and time) are needed to develop, validate, and run simulation models.

The expressiveness of simulation results also depends on the plausibility and resilience of the underlying simulation model. In particular, it is difficult to specify adequate simulation models, especially when the scenario under consideration implies the specification of many input parameters and involves complex interactions, as is most often the case with PAIS engineering projects.

To be able to build simulation models, it is often inevitable to rely on hypotheses, sometimes even arguable assumptions. In response to this problem (i.e., to generate needed data), we have accomplished various empirical and experimental research activities in the EcoPOST project (e.g., software experiments, online surveys, case studies) which are discussed in more detail in Chapters 9 to 11 of this thesis. The basic problem, namely the high efforts related to the performance of such activities, cannot be overcome. Hence, simulation outcomes must be carefully interpreted. It is important to keep in mind that any conclusion is based on a simulation rather than a granted real-world situation.

5.7 Summary

This section has described Step 6 of the EcoPOST methodology, i.e., the simulation of evaluation models. First, we have explained why simulation is needed in the context of EcoPOST evaluation models. Based on this, we have introduced the constituting elements of EcoPOST simulation models and have described how these elements can be formally specified. We have also discussed how nonlinear dependencies between impact and cost factors are specified using table functions, and have illustrated the computation of simulations.

To illustrate the described simulation concepts, we have given two examples. The first one has dealt with the simulation of a DCF "Business Process Redesign Costs" and its interfer-

ence through an ImF "End User Fears". The second, more complex one has addressed the simulation of costs for maintaining process logic. Finally, we have discussed strengths and weaknesses of using simulation.

Chapter 6

Evaluation Patterns

6.1 Motivation

Designing evaluation models can be a complicated and time-consuming task. Evaluation models can become complex due to the high number of potential cost and impact factors and due to the many causal dependencies that exist between them. In the approach described (cf. Chapters 4 and 5), each evaluation and simulation model has to be designed from scratch so far. Besides the additional efforts, this situation results in the loss of valuable modeling experiences, and it prevents the reuse of evaluation and simulation models.

In response to these problems this chapter introduces the notion of *evaluation patterns* (EP). EPs do not only ease the design and simulation of evaluation models (cf. Requirements R-6 and R-7 in Chapter 3), but also enable the reuse of evaluation information (cf. Requirement R-8). This is crucial to foster the applicability of the EcoPOST framework.

The remainder of this chapter is organized as follows. Section 6.2 introduces the basic notion of EPs and gives a definition. Sections 6.3 and 6.4 describe EPs of different kind. Section 6.5 deals with the customization of an EP when reusing it. Section 6.6 gives a discussion and Section 6.7 concludes with a summary.

6.2 Primary vs. Secondary Evaluation Patterns

PAIS engineering projects often exhibit similarities, e.g., regarding the appearance of certain cost and impact factors. We pick up these similarities by introducing customizable patterns in order to increase model reuse and to facilitate the practical use of the EcoPOST framework. Specifically, we introduce an *evaluation pattern* (EP) as a predefined, but customizable evaluation model.

Patterns. Generally, patterns describe solutions to recurring problems. They aim at supporting others in learning from available solutions and allow the application of these solutions to similar situations. Typically, a pattern is given a name, which serves as a conceptual handle to facilitate discussions on the information it represents.

Patterns were first used to describe best practices in architecture [5]. However, they have also a long tradition in computer science. At present, there exist patterns, for instance, in the fields of software architecture (*conceptual patterns*), design (*design patterns*), and programming (*XML schema patterns*, *J2EE patterns*, etc.). Recently, the idea of using patterns has been also applied to more specific domains like workflow management [201, 214] or inter-organizational control [88].

Generally, patterns often have a generative character. Alexander [4], for example, notes: "*These patterns in our minds are, more or less, mental images of the patterns in the world: they are abstract representations of the very morphological rules which define the patterns in the world. However, in one respect they are very different. The patterns in the world merely exist. But the same patterns in our minds are dynamic. They have force. They are generative. They tell us what to do; they tell us how we shall, or may, generate them; and they tell us too, that under certain circumstances, we must create them. Each pattern is a rule which describes what you have to do to generate the entity which it defines.*" Generative¹ patterns tell us how to create something and can be observed in the environments they helped to shape (like the patterns we introduce in the following).

Evaluation Patterns. Basically, an *evaluation pattern* (EP) can be built with the same elements as introduced in Chapter 4. An EP consists of an *evaluation model* and an associated *simulation model*. More precisely, each EP is a template for a specific DCF or ImF we encounter in typical PAIS engineering projects. Altogether, we define an EP as follows:

Definition 6.1 (Evaluation Pattern) *A tuple $EP = (id, desc, type, mod)$ is called evaluation pattern, if the following holds:*

- *id is the primary name of the EP,*
- *desc is a textual description of the semantics of the EP,*
- *type specifies the type of the EP, where*
 - *a primary EP represents a DCF denoted with type = "PRIMARY",*
 - *a secondary EP represents an ImF denoted with type = "SECONDARY",*
- *mod is a tuple (EM,SM) with EM an evaluation model and SM a simulation model.*

We distinguish between *primary* and *secondary* EPs. A primary EP describes a particular DCF, and a secondary EP an ImF. Characteristic examples of primary EPs are summarized in Section 6.3. These primary EPs include, for instances, an EP dealing with costs related to business process redesign activities or an EP addressing costs caused by eliciting requirements for adequately supporting business processes. Secondary EPs are summarized in Section 6.4. Examples include EPs dealing with the impact of process or domain knowledge on the costs related to the introduction of a PAIS.

¹Non-generative patterns, in turn, describe recurring phenomena without saying how to reproduce them.

All EPs described in the following² have been derived based on two pillars: (1) results from the survey discussed in Section 4.7.1 and (2) practical experiences gathered in several PAIS engineering projects in the automotive domain.

6.3 Primary Evaluation Patterns

It is important to mention that the representation of cost and impact factors as static or dynamic factors in the EPs discussed in the following may depend on the modeler. In other words, many cost and impact factor can be modeled both as static or dynamic factors, i.e., the depicted EPs (cf. Fig. 6.1) can be modeled in alternative ways as well. Likewise, this is valid for the secondary EPs from Section 6.4.

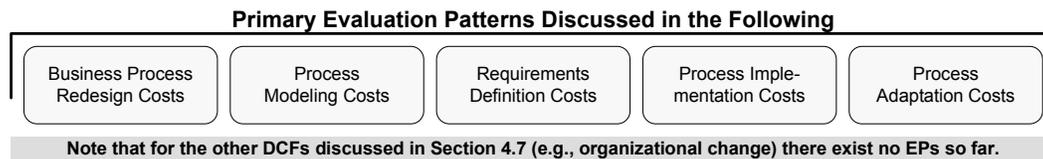


Figure 6.1: Primary Evaluation Patterns.

6.3.1 Business Process Redesign Costs

The EP which is shown in Fig. 6.2 deals with the costs of business process redesign activities [77, 221]. Prior to the development of PAIS such activities can become necessary for several reasons. As examples consider the need to optimize the performance of business processes or the goal of realizing a higher degree of process automation.

Explanation. This EP is based on our experiences in several process redesign projects that business process redesign costs are primarily determined by two SCFs: "Planned Costs for Process Analysis" and "Planned Costs for Process Modeling". While the former SCF represents planned costs for accomplishing interviews with process participants and costs for evaluating existing process documentation, the latter SCF concerns costs for transforming gathered information (about the process to be redesigned) into a new process design. Process redesign costs are thereby assumed to be varying, i.e., they are represented as DCF.

This EP also reflects our experience that six additional ImFs are of particular importance when investigating the costs of process redesign activities: "Process Complexity", "Management Commitment", "End User Fears", "Process Knowledge", "Domain Knowledge", and "Ability to Redesign Business Processes" (cf. Fig. 6.2). The importance of some of these factors is also confirmed by our survey results from Section 4.8 (e.g., of management commitment). While process analysis costs (i.e., the respective SCF) are influenced by "Process Complexity" and "End User Fears" (merged in the auxiliary variable "Adjusted Costs for

²The simulation models of the EPs which are discussed in this chapter are denoted in Appendix B.

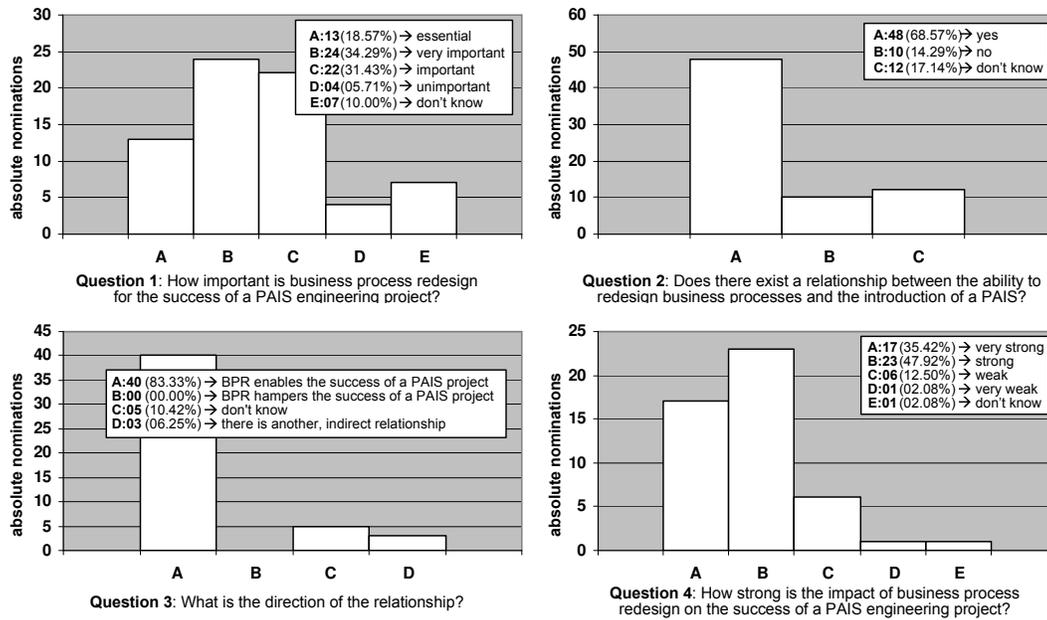


Figure 6.3: Validating the Importance of Business Process Redesign.

other, indirect relationship between the redesign of business processes and the success of a PAIS engineering project. Finally (and again out of these 68.57%), 83.36% of the respondents state (cf. Question 4 in Fig. 6.3) that the impact of business process redesign on the success of a PAIS engineering project either is "very strong" (35.42%) or "strong" (47.92%). This information can be useful when building a table function for specifying the nonlinear impact of "Ability to redesign Business Processes" on "Business Process Redesign Costs".

6.3.2 Process Modeling Costs

The EP shown in Fig. 6.4 deals with the costs of process modeling activities in PAIS engineering projects. Such activities are typically accomplished in order to prepare the information gathered during a process analysis (and requirements elicitation), to assist software developers in implementing the PAIS, and to serve as a guideline when the new process design is implemented in an organization.

Generally, there exist many notations that can be used to specify process models. Our EP, however, assumes that process models are formulated as *event-driven process chains* (EPC), a widely accepted formalism which also used, for example, in the *ARIS toolset*.

Explanation. Basically, this EP reflects our experiences that "Process Modeling Costs" are influenced by three ImFs: the two static ImFs "Process Complexity" and "Process Size" (whereas the impact of process size is specified based on a table function transforming a given process size into an impact rating; cf. Section 5.3.4) and the dynamic ImF "Process Knowledge" (which is also confirmed by our survey from Section 4.8).

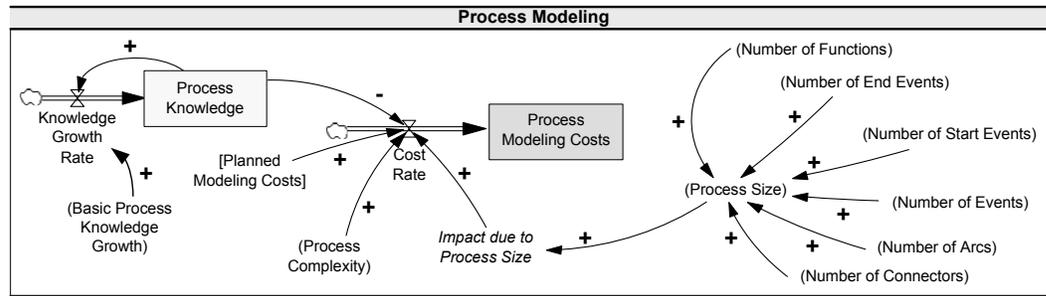


Figure 6.4: Primary Evaluation Pattern: Process Modeling Costs.

The ImF "Process Complexity" is not further discussed here. Instead, we refer to Section 6.4.5 where this ImF is discussed in detail. The ImF "Process Size", in turn, is characterized based on (estimated) attributes of the process model to be developed. These attributes depend on the used modeling formalism. As aforementioned, the EP from Fig. 6.4 builds on the assumption that the EPC formalism is used for process modeling. Taking this formalism, we specify process size based on the "Number of Functions", "Number of Events", "Number of Arcs", "Number of Connectors", "Number of Start Events", and "Number of End Events". Finally, the DCF "Process Modeling Costs" is also influenced by the dynamic ImF "Process Knowledge" (assuming that increasing process knowledge results in decreasing modeling costs). Further, the level of process knowledge increases with the costs of process modeling (as any additionally hour of process modeling typically increases the comprehensiveness of the modeled process).

6.3.3 Requirements Definition Costs

The EP from Fig. 6.5 deals with costs for defining and eliciting requirements. Underlying activities are the elicitation and documentation of requirements. These costs are significantly influenced by attributes of the business process(es) to be supported.

Explanation. This EP reflects our observation from practice that the DCF "Requirements Definition Costs" is determined by three main cost factors: costs for a requirements management tool, process analysis costs, and requirements documentation costs. Costs for a requirements management tool are constant and are therefore represented as SCF. The auxiliary variable "Adjusted Process Analysis Costs", in turn, merges the SCF "Planned Process Analysis Costs" with four process-related ImFs: "Process Complexity", "Process Fragmentation", "Process Knowledge", and "Emotional Resistance of End Users" (whereas only process knowledge is represented as dynamic ImF).

Costs for documenting requirements (represented by the auxiliary variable "Requirements Documentation Costs") are determined by the SCF "Documentation Costs per Requirement" and by the dynamic ImF "Requirements to be Documented". The latter ImF also influences the dynamic ImF "Process Knowledge" (resulting in a positive link from "Analyzed Requirements" to the rate variable "Process Knowledge Growth Rate").

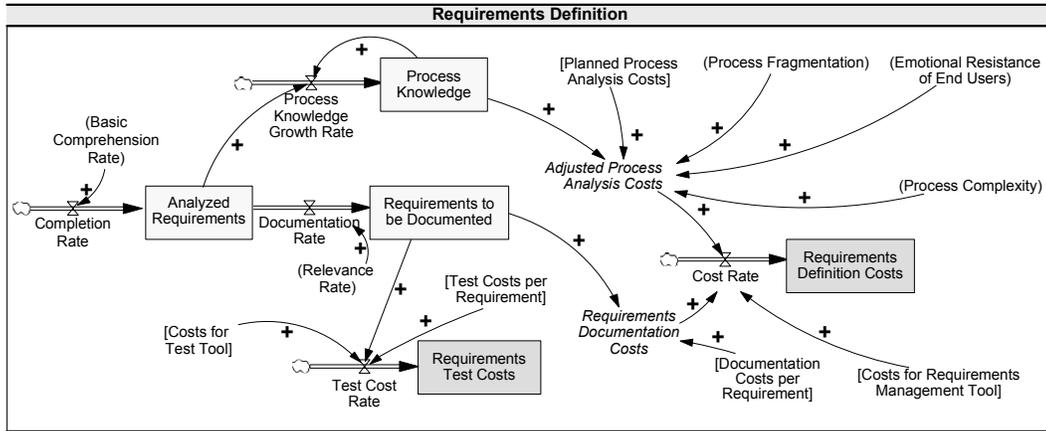


Figure 6.5: Primary Evaluation Pattern: Requirements Definition Costs.

"Requirements Test Costs" are determined by two SCFs ("Costs for Test Tool" and "Test Costs per Requirement") and one dynamic ImF "Requirements to be documented" (as only documented requirements need to be tested). Costs for a test tool and test costs per requirement can be assumed to be constant and are therefore represented as SCFs.

6.3.4 Process Implementation Costs

The EP shown in Fig. 6.6 deals with costs for implementing a process and the interference of these costs through impact factors. Note that we have analyzed this EP in more detail in our controlled software experiment (cf. Chapter 9).

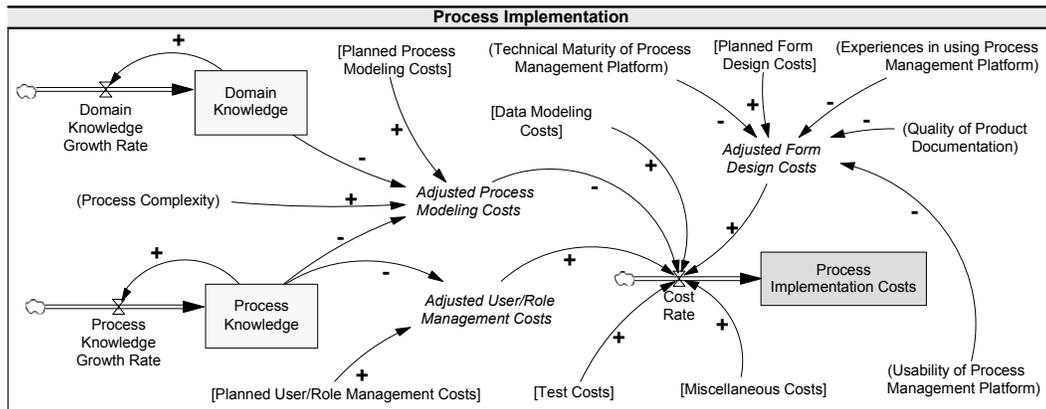


Figure 6.6: Primary Evaluation Pattern: Process Implementation Costs.

Explanation. Basic to this EP is our observation that the implementation of a process is determined by six main cost factors (cf. Fig. 6.6): "Adjusted Process Modeling Costs", "Adjusted User/Role Management Costs", "Adjusted Form Design Costs", "Data Modeling

Costs", "Test Costs", and "Miscellaneous Costs". The first three cost factors are characterized as "adjusted" as they are influenced – according to interviews with software developers and process engineers – by additional process-related ImFs. Therefore, they are represented by auxiliary variables which merge the SCFs with ImFs.

Process modeling costs, for example, are influenced by "Process Knowledge", "Domain Knowledge", and by "Process Complexity". User/role management costs are only biased by "Process Knowledge". Form design costs are influenced by some technology-specific ImFs (cf. Chapter 4.7.4): "Technical Maturity of Process Management Platform", "Experiences in using Process Management Platform", "Usability of Process Management Platform", and "Quality of Product Documentation".

Note that this EP strongly simplifies the issue of process implementation. In particular, we assume that the identified six main cost factors aggregate all other potential cost factors. The SCF "Data Modeling Costs", for example, may include costs for providing database management functionality and for configuring a database management system. However, it is thereby not further distinguished between subsidiary cost factors, i.e., other cost factors are not made explicit. If it is considered as necessary, additional SCFs or DCFs can be introduced in order to make specific cost factors more explicit.

6.3.5 Process Adaptation Costs

This EP (cf. Fig. 6.7) deals with the costs caused by adapting the process(es) supported by a PAIS. Note that this EP is identical to the previous EP "Process Implementation Costs" (cf. Section 6.3.4) – except for one additional ImF "Process Evolution". The inclusion of this ImF becomes necessary as process adaptation costs strongly depend on the number of process changes that need to be realized.

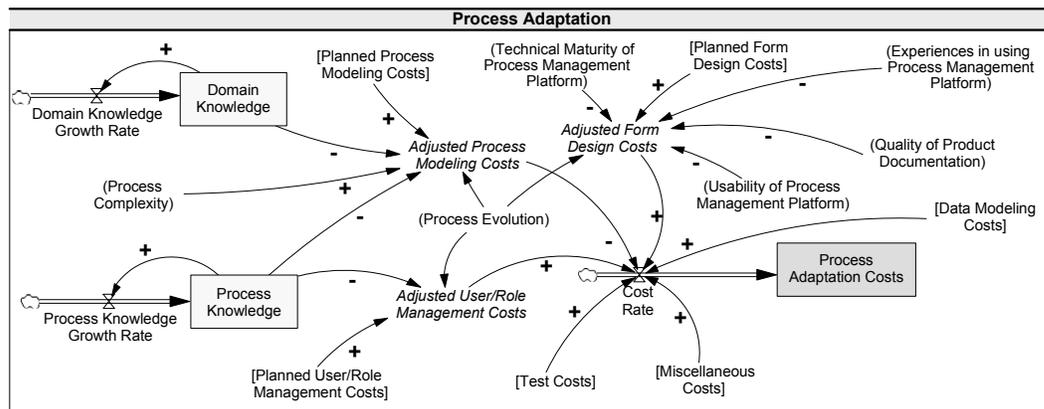


Figure 6.7: Primary Evaluation Pattern: Process Adaptation Costs.

6.4 Secondary Evaluation Patterns

Fig. 6.8 summarizes the secondary EPs we discuss in the following. Like the previously presented EPs, these patterns are also based on survey results and practical experiences.

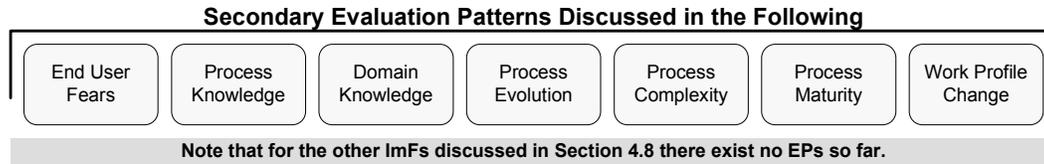


Figure 6.8: Secondary Evaluation Patterns.

6.4.1 End User Fears

Consider Fig. 6.9 which shows an EP dealing with the effects of "End User Fears". More specifically, our experiences as a process engineer and consultant allow for the conclusion that the introduction of a PAIS may cause end user fears, e.g., due to work profile change (i.e., job redesign) or changed social clues. Among other things, such fears can lead to emotional resistance of end users. This, in turn, can make it difficult to get needed support from end users, e.g., during an interview-based process analysis.

Explanation. Basic to this EP (which has been already discussed as introductory example in Section 4.9.2) is a feedback loop connecting the four ImFs "End User Fears", "Emotional Resistance", "Ability to Acquire Process Knowledge", and "Ability to Redesign Business Processes". Their arrangement illustrates the following coherence: increasing end user fears lead to increasing emotional resistance of end users (cf. Fig. 4.14). An increasing emotional resistance of end users may result in a decreasing ability to acquire process knowledge. Reason is that increasing emotional resistance makes profound process analysis (e.g., based on interviews with process participants), a difficult task to accomplish. The inability to acquire process knowledge, in turn, may result in a decreasing ability to redesign business processes (as needed information is missing). Finally, an increasing ability to redesign business processes can even enforce end user fears since end users often consider business process redesign activities as a potential threat for their own job.

The variable "End User Fears" is not only influenced by the "Ability to Redesign Business Processes", but by two other ImFs as well: "Work Profile Change" and "Changes of Social Clues and Interactions". Moreover, "Communication with End Users" can decrease end user fears, e.g., by informing them about the goals of introducing a PAIS.

Empirical Validation. We analyze the issue of end user fears in more detail in our survey (cf. Section 4.7.1). Consider Fig. 6.10. A majority of 74.28% of the participants consider end user fears as "very critical" (28.57%) or "critical" (45.71%) for the overall success of a PAIS engineering projects (cf. Fig. 6.10A). More specifically, 70% of the respondents confirm that there exists a relationship between end user fears and the emotional resistance

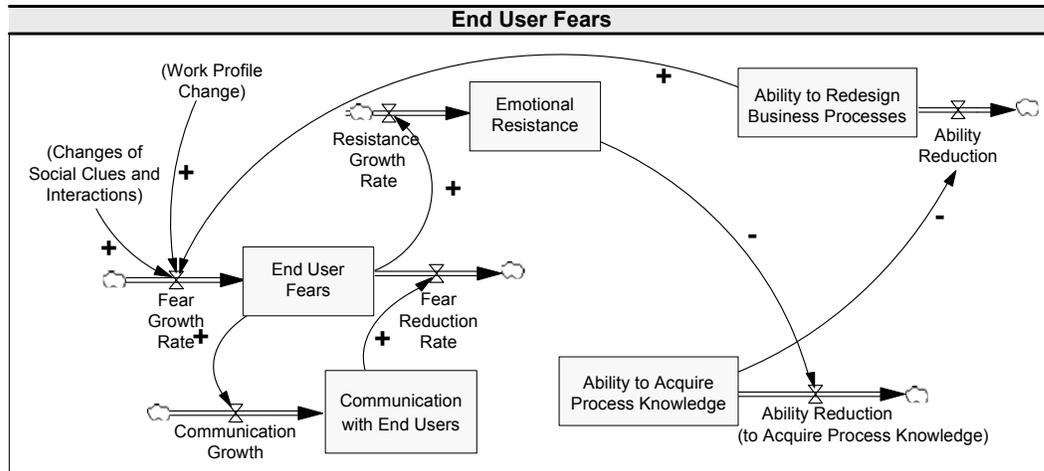


Figure 6.9: Secondary Evaluation Pattern: Dealing with the Impact of End User Fears.

of end users against PAIS engineering projects (cf. Fig. 6.10B). This confirms the link between "End User Fears" and "Emotional Resistance of End Users" in Fig. 6.9.

What is still not clear at this point is the direction of the causal dependency – there are several possibilities in this respect (cf. Fig. 6.9). However, the semantics of the link can be clarified based on the following results. 83.67% share the opinion that increasing end user fears result in increasing emotional resistance (cf. Fig. 6.10C). This confirms that "End User Fears" and "Emotional Resistance" need to be connected with a positive link pointing from the former variable to the latter one. Finally, 89.8% of the respondents state (cf. Fig. 6.10D) that the impact of end user fears on emotional resistance is either "very strong" (42.86%) or "strong" (46.94%). This information can help when developing a simulation model for this EP.

In the evaluation model from Fig. 6.9, we assume that increased communication results in decreasing end user fears. 92.86% of the survey participants consider communication with end users as "essential" (47.14%), "very important" (35.71%) or "important" (10%) for its success. Furthermore, 78.57% of the respondents confirm that there is a causal dependency between communication and end user fears. Out of these, 74.55% are the opinion that an increasing communication results in decreasing end user fears. This confirms that "Communication with End Users" and "End User Fears" can be connected with a negative link pointing from the former variable to the latter one. Finally, 85.45% of the respondents consider the impact of communication on end user fears as either "very strong" (29.09%) or "strong" (56.36%). This information helps us when specifying a corresponding simulation model for this evaluation model.

6.4.2 Process Knowledge

Fig. 6.12 shows an EP which specifies the ImF "Process Knowledge" in more detail, i.e., causal dependencies on knowledge about the process(es) to be supported. Process knowl-

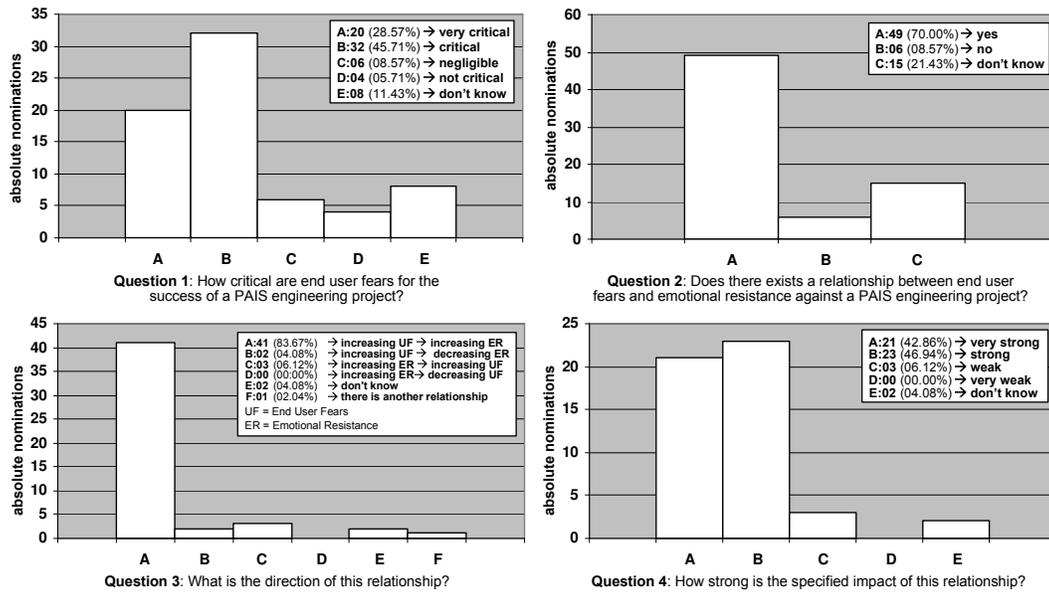


Figure 6.10: Validating the Impact of End User Fears.

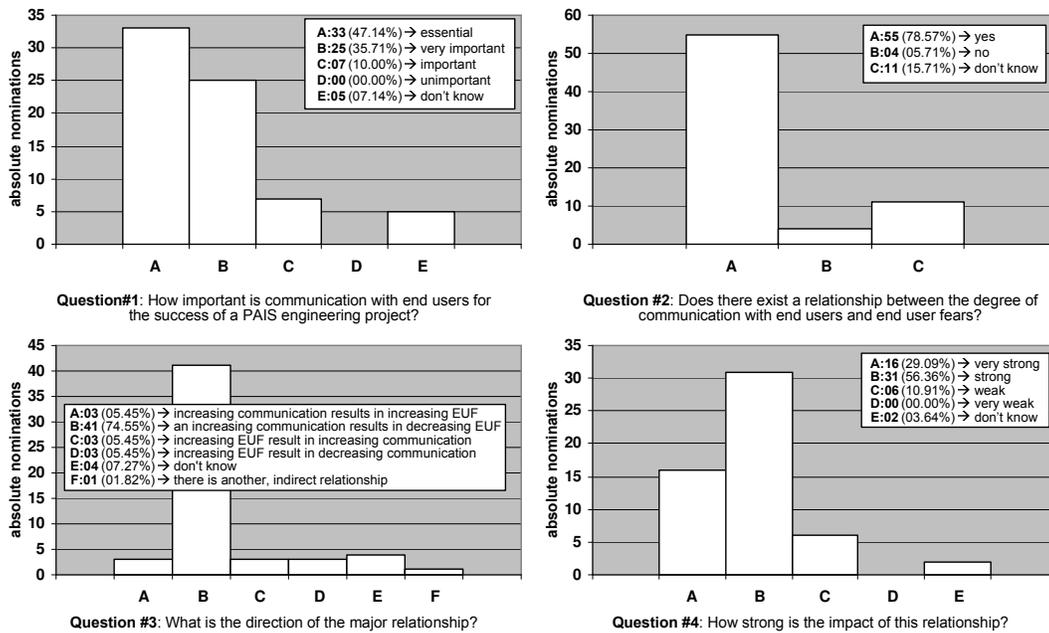


Figure 6.11: Validating the Impact of Communication.

edge includes, for example, knowledge about process participants and their roles as well as knowledge about the flow of data.

Explanation. Acquiring process knowledge necessitates that the ability to acquire process knowledge is given. This ability, however, strongly depends on three ImFs: "Emotional Resistance of End Users", "Process Complexity", and "Process Fragmentation". Besides, process knowledge is also influenced by the dynamic ImF "Domain Knowledge".

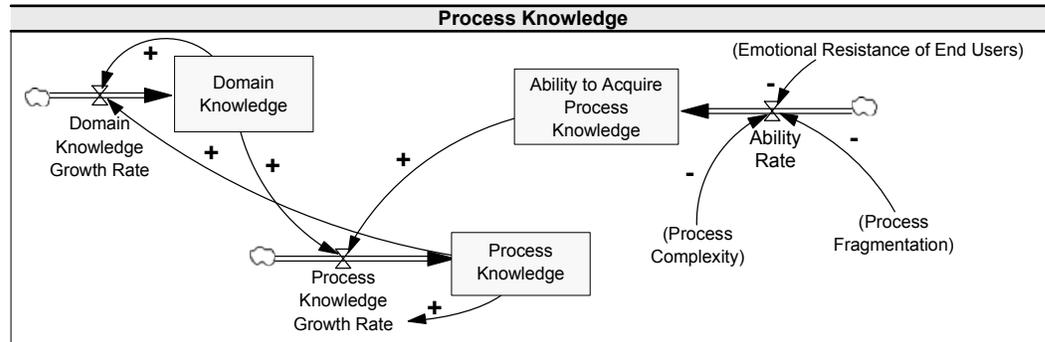


Figure 6.12: Secondary Evaluation Pattern: Process Knowledge.

Empirical Validation. We analyze process knowledge in our survey. 92.86% of the participants consider process knowledge as "essential" (41.43%), "very important" (37.14%) or "important" (14.29%) for PAIS engineering projects (cf. Question 1 in Fig. 6.13). Further, 61.43% of the survey participants confirm that there is a relationship between process knowledge and the costs of a PAIS engineering project (cf. Question 2 in Fig. 6.13).

Out of these respondents, 72.09% share the opinion that a low (high) process knowledge results in increasing (decreasing) costs of a PAIS engineering project (cf. Question 3 in Fig. 6.13). Surprisingly, 18.6% of the survey participants believe that a low (high) process knowledge results in decreasing (increasing) costs of a PAIS engineering project. It can be presumed here whether this figure corresponds to the real opinion of the respective survey participants, or whether some of them did not carefully read and select all possible answers. 6.98% of the respondents state that there is another, indirect relationship between process knowledge and the costs of a PAIS engineering project (without further specifying the kind of indirect relationship). Finally, 50% of these respondents point out (cf. Question 4 in Fig. 6.13) that the impact of process knowledge on the success of a PAIS engineering project either is "very strong" (17.14%) or "strong" (32.86%). These survey results show that profound process knowledge enables more effective process implementations and therewith results in decreasing costs.

6.4.3 Domain Knowledge

The EP from Fig. 6.14 deals with the evolution of domain knowledge along the course of a PAIS engineering project.

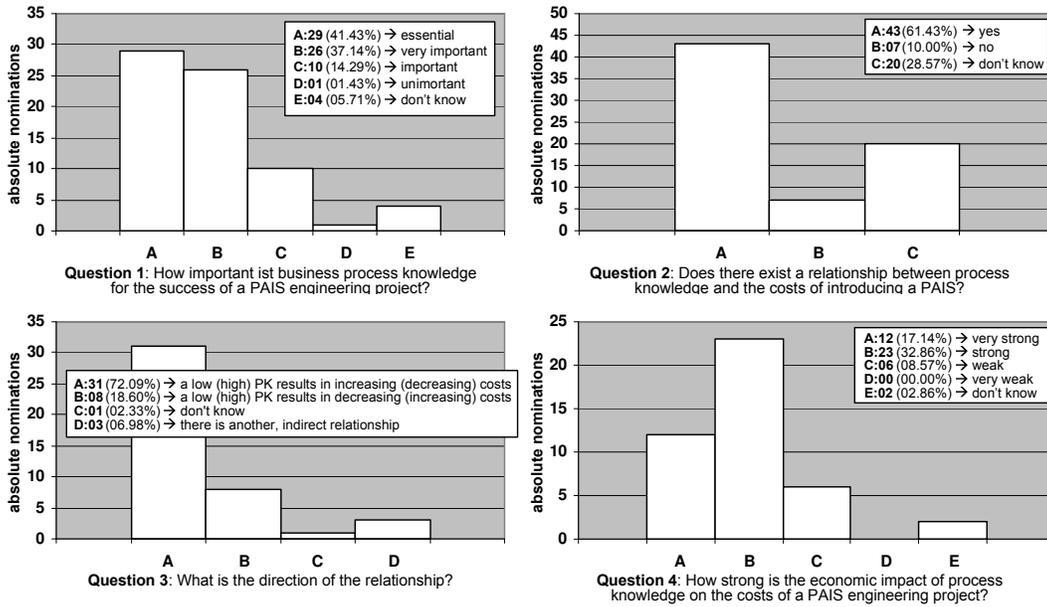


Figure 6.13: Validating the Impact of Process Knowledge.

Explanation. Our practical experiences allow for the conclusion that "Domain Knowledge" is a dynamic ImF which is influenced by three other ImFs: the period a PAIS engineer is working in a specific domain (captured by the dynamic ImF "Experience"), the dynamic ImF "Process Knowledge" (cf. Section 6.4.2), and the complexity of the considered domain (represented by the static ImF "Domain Complexity"). Besides, the dynamic ImF "Domain Knowledge" is additionally influenced by the static ImF "Basic Domain Knowledge Growth". This static ImF reflects the situation that domain knowledge is continuously increasing during a PAIS engineering project (or a subsidiary activity).

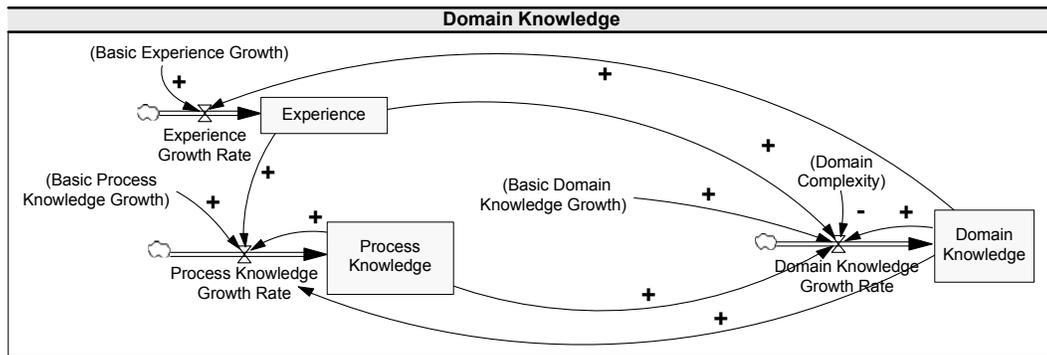


Figure 6.14: Secondary Evaluation Pattern: Domain Knowledge.

Empirical Validation. We investigate the role of domain knowledge in more detail in our survey. Domain knowledge is considered as an "essential" (27.14%), "very important" (38.57%) or "important" (21.43%) factor for PAIS engineering projects by 87.14% of the survey participants (cf. Question 1 in Fig. 6.15). Moreover, 47.14% of the respondents acknowledge that there is a dependency between domain knowledge and the costs of a PAIS engineering project (cf. Question 2 in Fig. 6.15). When compared to process knowledge, this number is rather low and indicates that the impact of domain knowledge on the costs of a PAIS engineering project need to be considered as being of minor relevance. Further, 69.7% of the respondents (out of the 47.14%) share the opinion that a low (high) domain knowledge results in increasing (decreasing) costs of a PAIS engineering project (cf. Question 3 in Fig. 6.15). This confirms that domain knowledge needs to be assigned to DCF with a negative link. Finally – and this important when building a table function specifying the nonlinear impact of domain knowledge on the costs of a PAIS engineering project – 78.13% of the survey participants state (cf. Question 4 in Fig. 6.15) that the overall impact of domain knowledge either is "very strong" (25%) or "strong" (53.13%).

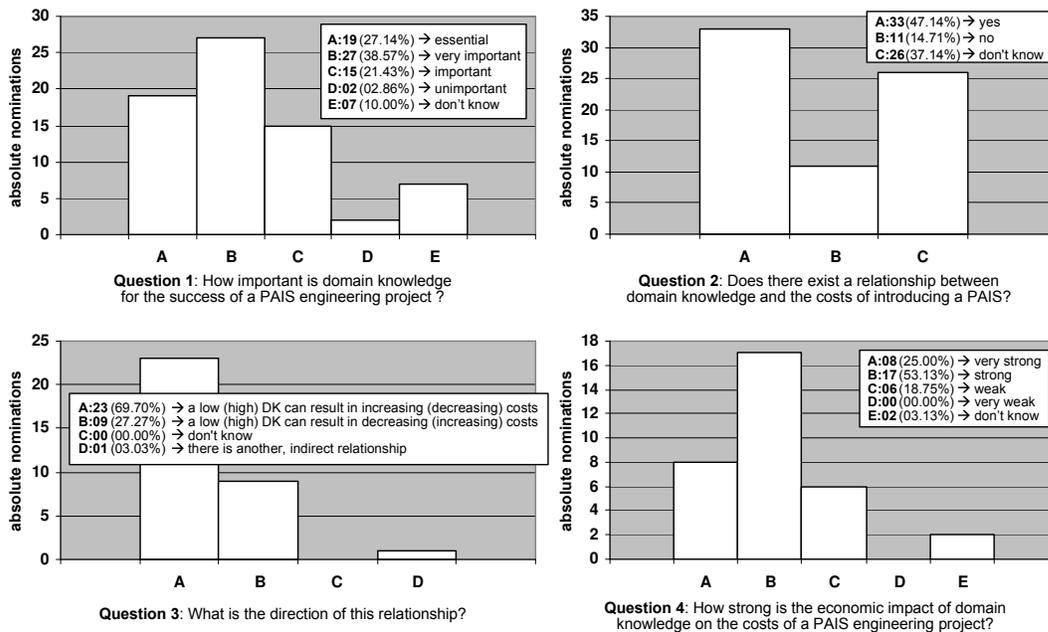


Figure 6.15: Validating the Impact of Domain Knowledge.

Generally, it can be assumed that high domain knowledge results in more effective business process implementations and therewith decreases costs for PAIS engineering project.

6.4.4 Process Evolution

The EP shown in Fig. 6.16 specifies the static ImF "Process Evolution". Specifically, it describes origins of process evolution. Basically, this EP reflects the assumption that process evolution is caused by various drivers of evolution (cf. Section 1.1). As one example

consider the static ImF "Market Pressure". This ImF is further detailed in the EP based on the five competitive forces identified by Porter [154] (cf. Section 2.6.2) – all of them represented as static ImFs. Other included drivers of evolution are "User Acceptance", "Need for Process Optimization", and "Need for Compliance with Regulations and Laws". If desired, additional arbitrary drivers of evolution can be included in the EP depending on the respective evaluation scenario.

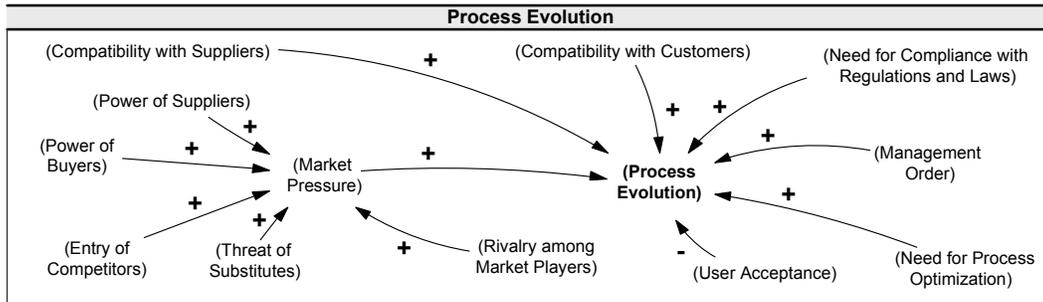


Figure 6.16: Primary Evaluation Pattern: Business Process Evolution.

6.4.5 Process Complexity

The EP from Fig. 6.17 aims at specifying the ImF "Process Complexity". Note that this EP does not specify process complexity itself, but defines it based on an easier manageable replacement factor. In our context, this replacement factor corresponds to the complexity of the process model describing the business process to be supported [37]. Thus, we extend process complexity to "Process Complexity/Process Model Complexity". The EP from Fig. 6.17 (like the one from Section 6.3.2) also aligns with the assumption that respective process models are formulated using the EPC notation.

Explanation. The EP from Fig. 6.17 builds upon the assumption that the static ImF "Process Complexity/Process Model Complexity" is determined by four other static ImFs: "Cycle Complexity", "Join Complexity" (JC), "Control-Flow Complexity" (CFC), and "Split-Join-Ratio" (SJR) (whereas the latter ImF is derived from the SCFs "Join Complexity" and "Control-Flow Complexity").

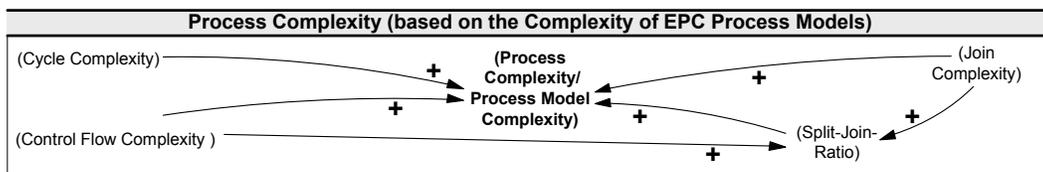


Figure 6.17: Secondary Evaluation Pattern: Process Complexity.

The complexity driver "Cycle Complexity" is confirmed in [38, 105]. Arbitrary cycles, for example, can lead to EPC models without clear semantics (cf. [90] for examples).

The ImF "Control-Flow Complexity" is characterized by Cardoso [37]. It is based on the observation that the three split connector types in EPC models introduce a different degree of complexity. According to the number of potential post-states an AND-split is weighted with 1, an XOR-split is weighted with the number of successors n , and an OR-split is weighted with $2n - 1$. The sum of all connector weights of an EPC model is then denoted as "Control-Flow Complexity" (see also [72]). The ImF "Join Complexity" can be defined as the sum of weighted join connectors based on the number of potential pre-states in EPC models [120, 121]. Finally, the mismatch between potential post-states of splits and pre-states of joins in EPC models is included as another driver of complexity. This mismatch can be expressed by the static ImF "Split-Join-Ratio" ($= JC/CFC$) [120, 121].

Based on these four static ImFs (or drivers of complexity), we derive the EP from Fig. 6.17. Thereby, an increasing cycle complexity results in a higher process complexity. Also, both increasing CFC and increasing JC result in increasing process complexity. Finally, a JSR value different from one increases the error probability and thus process complexity. It is important to mention that – if desired – other drivers of process complexity can be considered as well. Examples can be found in [105, 121].

6.4.6 Process Maturity

The EP from Fig. 6.18 specifies the static ImF "Process Maturity". The exact impact of process maturity is thereby not specified as it is organization-specific, but it can be generally assumed that an increasing process maturity results in lower costs.

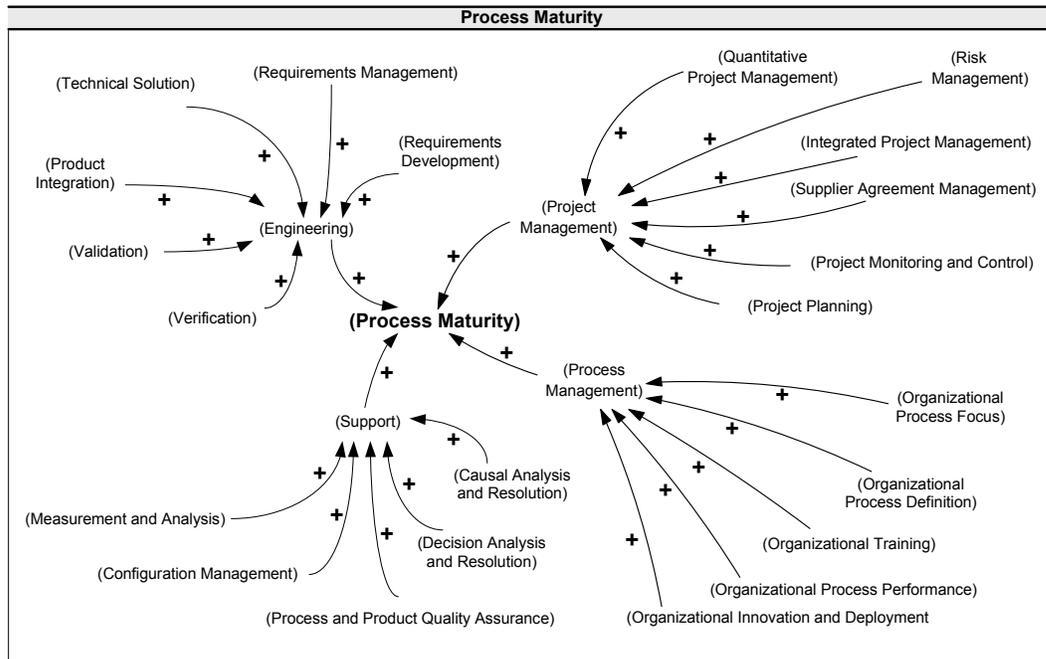


Figure 6.18: Secondary Evaluation Pattern: Process Maturity (Continuous Representation).

Explanation. In particular, this static ImF is characterized along the 22 process areas of the *capability maturity model integration*³ (CMMI) [48]. Thereby, the overall ImF "Process Maturity" is determined by the maturity of the four categories of the continuous representation of CMMI "Process Management", "Engineering", "Project Management", and "Support". Each of these four categories is further detailed by a subset of the 22 CMMI process areas. Note that each process area is associated with a set of specific and generic goals and practices which can be used to evaluate the capability level of each process area.

The quantification of the different process areas (for the simulation model) can be derived either based on the guidelines for calculating the CMMI capability level or based on organization-specific measures.

6.4.7 Work Profile Change

This EP (cf. Fig. 6.19) deals with change of end user work profiles (and the effects of work profile changes) in the context of a PAIS introduction. More specifically, this EP relates the perceived work profile change to changes emerging in the five job dimensions of Hackman's *job characteristics model* [74, 75]: (1) *skill variety*, (2) *task identity*, (3) *task significance*, (4) *autonomy*, and (5) *feedback from the job*.

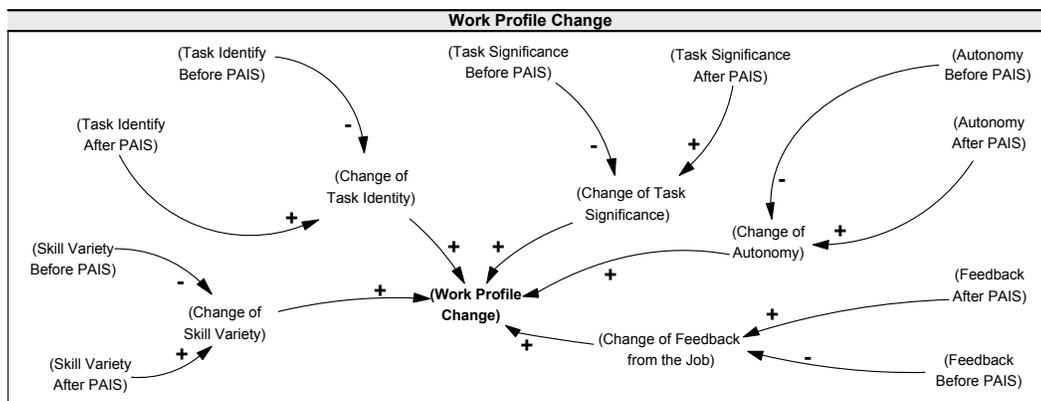


Figure 6.19: Primary Evaluation Pattern: Work Profile Change.

For each of these five core job dimensions, the emerging change is designated based on the level before and after the PAIS introduction. The change of skill variety, for example, is determined based on the ratio of the skill variety before the introduction of the PAIS and the skill variety afterwards. Note that in Chapter 10, we will present the results of a case study in which we investigate this EP and changes in the five job dimensions caused by the

³CMMI is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps to integrate traditionally separate organizational functions, to set process improvement goals and priorities, to provide guidance for quality processes, and to provide a point of reference for appraising current processes. The latest version of CMMI (Version 1.2) was released in August 2006. There are 3 constellations of CMMI in this new version: *CMMI for Development* (underlying our EPs), *CMMI Services* and *CMMI Acquisition*. *CMMI for Development* unifies 22 process areas.

introduction of a PAIS in more detail.

Summary. All primary and secondary EPs which have been discussed in Section 6.3 and Section 6.4 should be considered as suggestions making similarities in PAIS engineering projects explicit. The relevance of selected EPs (e.g., business process redesign costs, end user fears, process knowledge, domain knowledge) has been confirmed based on survey results. This set of EPs can be used as a baseline and starting point for building more complex evaluation and simulation models.

6.5 Customizing Evaluation Patterns

Customization. Customization becomes necessary as EPs are applied in different evaluation context. Thereby, we distinguish between the customization of the *evaluation model* (Step I in Fig. 6.20) and the *simulation model* (Step II in Fig. 6.20) of an EP. The former always requires the subsequent adaptation of the underlying simulation model, while the latter is also possible without adapting the respective evaluation model.

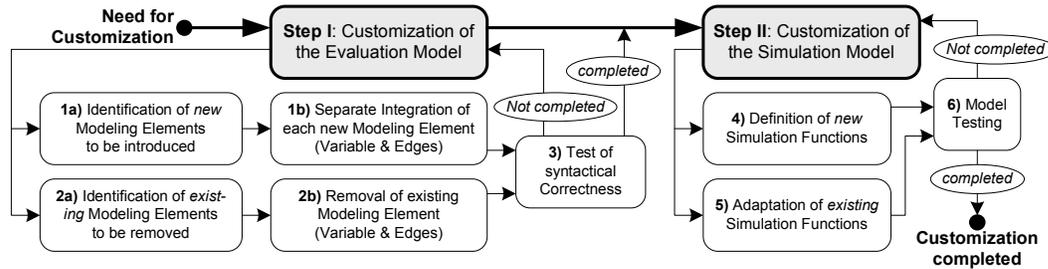


Figure 6.20: Step-by-Step Customization of Evaluation Patterns.

Adapting an evaluation model can be achieved by adding or removing model variables, flows, or links (Step 1a/b and Step 2a/b in Fig. 6.20). The EP "End User Fears" (cf. Fig. 6.9), for example, could be customized by introducing an ImF "Management Commitment" to take into account the impact of this factor on end user fears. Therefore, the new ImF "Management Commitment" is connected to the ImF "End User Fears". In our example this can be achieved with a negative link to denote that increasing management commitment results in decreased user fears. The correctness of a customized EP is ensured through the design rules discussed in Section 4.9.3.

Customizing a simulation model, by contrast, requires adaptations of the equations of the simulation model (Step 4 and Step 5 in Fig. 6.20). As examples of potential customizations consider changes of SCF values or adaptations of rate functions.

Merging Patterns. Customization becomes also necessary when EPs are merged. Assume, for example, that an additional ImF "End User Fears" has to be considered in the context of a DCF "Business Process Redesign Costs". This can be realized by merging a secondary EP (specifying the additional ImF) with a primary EP (specifying the DCF).

#	Merge Algorithm (Pseudo Code Notation)	Explanation
01	Model merge(Model m1, Model m2)	
02	List stateJoints = new List();	list of state joints
03	List rateJoints = new List();	list of rate joints
04	List auxiliaryJoints = new List();	list of auxiliary joints
05	List removal = new List();	list of removals
06	FOR EACH Node n1 IN m1	compare every node in model1
07	FOR EACH Node n2 IN m2	with every node in model2
08	IF n1.name==n2.name&&n1.type==n2.type	if both nodes form potential joint
09	IF n2 INSTANCEOF State	if it's a state joint
10	n1.addConnections(n2.getConnections());	reconnect the connections with n1
11	joint=true;	add joint to removal list
12	ELSE IF n2 INSTANCEOF Rate	if it's a rate joint
13	Boolean joint=false;	remember identified joint
14	IF n1.getSink()==n2.getSink()	if both nodes connect to same sink
15	IF n1.getSource()==n2.getSource()	- and to the same source
16	joint=true;	case 1: joint identified
17	ELSE IF n1.getSource() INSTANCEOF OutOfBounds	- source in model1 out-of-bounds
18	n1.setSource(n2.getSource());	case 2a
19	joint=true;	joint identified
20	ELSE IF n2.getSource() INSTANCEOF OutOfBounds	- source in model2 out-of-bounds
21	joint=true;	case 2a: joint identified
22	ELSE	- and none of the above
23	n1.name = n1.name+"_1";	no merge possible, rename rates
24	n2.name = n2.name+"_2";	
25	ELSE IF n1.getSource()==n2.getSource()	if both connect to same source
26	IF n1.getSink() INSTANCEOF OutOfBounds	- sink in model1 out-of-bounds
27	n1.setSink(n2.getSink());	case 2b
28	joint=true;	joint identified
29	ELSE IF n2.getSink() INSTANCEOF OutOfBounds	- sink in model2 out-of-bounds
30	joint=true;	case 2b: joint identified
31	ELSE	- and none of the above
32	n1.name = n1.name+"_1";	no merge possible, rename rates
33	n2.name = n2.name+"_2";	
34	ELSE IF n1.getSink() INSTANCEOF OutOfBounds&&	if sink in model1 and
35	n2.getSource() INSTANCEOF OutOfBounds	sources in model2 out-of-bounds
36	n1.setSink(n2.getSink());	case 3
37	joint=true;	joint identified
38	ELSE IF n1.getSource() INSTANCEOF OutOfBounds&&	if source in model1 and
39	n2.getSink() INSTANCEOF OutOfBounds	sinks in model1 are out-of-bounds
40	n1.setSource(n2.getSource());	case 3
41	joint=true;	joint identified
42	ELSE	in all other cases
43	n1.name = n1.name+"_1";	no merge possible, rename rates
44	n2.name = n2.name+"_2";	
45	IF joint	joint identified?
46	removals.add(n2);	add joint to removal list
47	n1.addConnections(n2.getConnections());	reconnect connections with n1
48	ELSE IF n2 INSTANCEOF Auxiliary	if it's an auxiliary joint
49	removals.add(a);	add joint to removal list
50	FOR EACH Connection c IN a2.getConnections()	go through all connect. of a2
51	IF c IS NOT IN a1.getConnections()	check if they appear in model1
52	a1.addConnection(c);	if not, add them
53	list2.remove(removal);	remove joint nodes once
54	List l = list1 + list2;	combine models
55	return transformListToMergedModel(l);	transform list to model

Table 6.1: Merge Algorithm for Evaluation Models.

Concerning evaluation models this operation can be automated. As input, a respective algorithm (cf. Table 6.1) needs two evaluation models EM1 and EM2. The merge of EM1 and EM2 is then accomplished through a systematic comparison of all model variables from EM1 with all model variables from EM2. If a model variable, e.g., a DCF, from EM1 has the same name and type as a model variable from EM2, both variables (and their links) can be merged⁴ (though this can be overruled by the model designer). In our example (cf. Fig. 6.21), the ImF "Ability to Redesign Business Processes" has been the mixing point. However, even if there exist no identical variables, evaluation models can be merged manually by the EcoPOST user.

⁴This procedure requires that the two evaluation models to be merged exhibit some overlap, i.e., both evaluation models have to contain at least one identical model variable.

Typically, any merge requires an additional manual post-editing of the newly generated model (regardless whether the merge has been automatically conducted or not). In our example (cf. Fig. 6.21), we introduce, for instance, an additional link between the ImF "Ability to Acquire Process Knowledge" and the ImF "Process Knowledge".

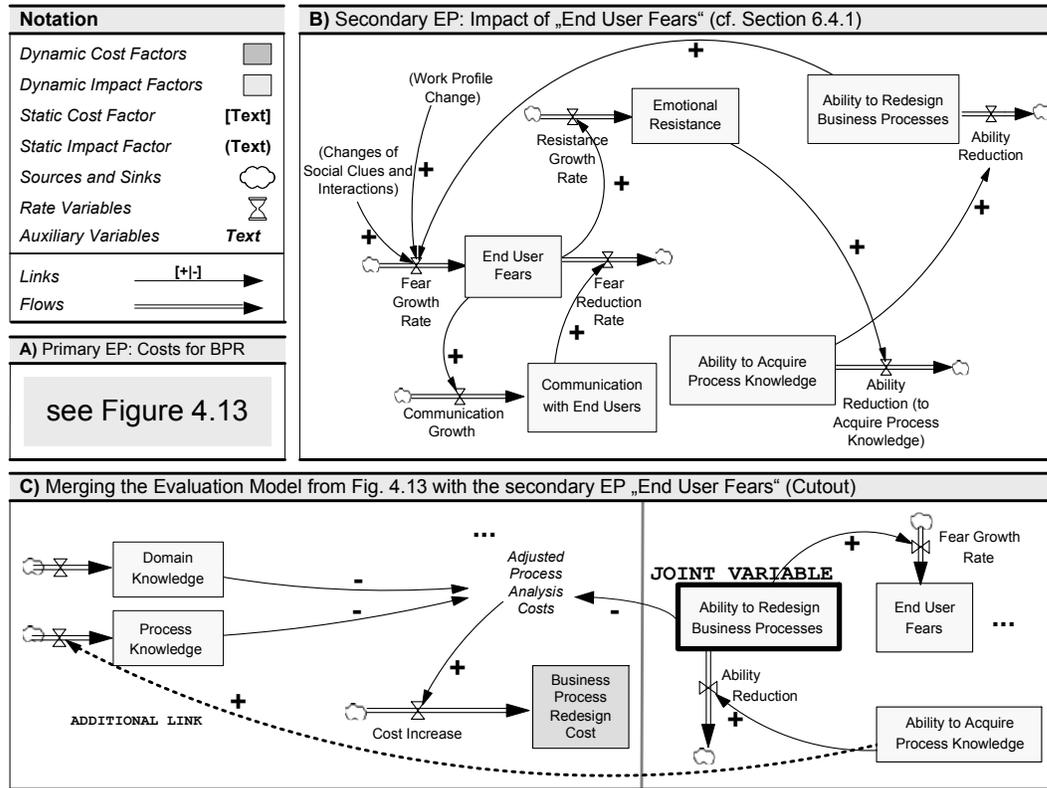


Figure 6.21: Merging Evaluation Patterns.

6.6 Discussion

Using patterns has been widely discussed since the advent of computer science research. Today, patterns are mainly applied to software architectures (*conceptual patterns*), design (*design patterns*), programming (*XML schema patterns*, *J2EE patterns*, etc.), as well as for software development processes and enterprise architecture (cf. Fig. 6.22). Recently, the idea of using patterns has been also applied to more specific domains like workflow management [201, 214] or inter-organizational control [88].

Fig. 6.22 illustrates the basic conceptual pillars underlying our notion of EP. In particular, EPs need to be transferable, i.e., the evaluation context an EP represents must be also applicable in other evaluation context. Moreover, EPs are problem-oriented and they lean against ideas known from object-oriented programming, hierarchical modeling, and reference modeling.

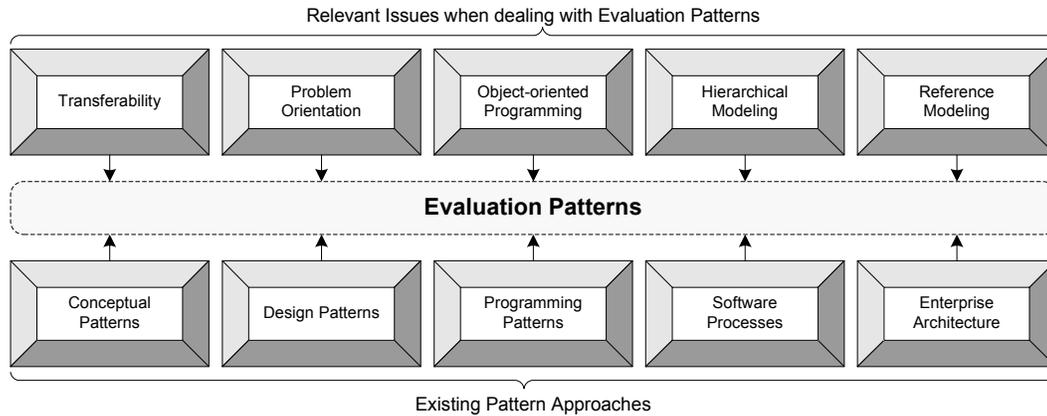


Figure 6.22: Related Issues in the Context of Evaluation Patterns.

The reuse of System Dynamics models has been also discussed before. Thereby, one has to distinguish between two basic directions. On the one hand, authors like Senge [181], Eberlein and Hines [60], Liehr [109], and Myrtveit [143] introduce predefined generic structures (with slightly different semantics). All these approaches satisfy the capability of defining "components". On the other hand, Winch [219] proposes a more restrictive approach which is only based on the parameterization of generic structures (without providing standardized modeling components). Our approach picks up ideas from both directions, i.e. we address both the definition of generic components as well as customization.

In our context, EPs enable the reuse of historical evaluation data. This reduces the need for designing evaluation models from scratch. Moreover, EPs are useful as a means for increasing the awareness for cost effects in PAIS engineering projects.

6.7 Summary

This section has introduced the notion of evaluation patterns. An EP is a predefined, but customizable evaluation model. More precisely, each EP constitutes a template for specific DCFs or ImFs we encounter in typical PAIS engineering projects. Basically, it consists of an evaluation model and an associated simulation model. Thereby, we have distinguished between primary and secondary EPs. A primary EP describes a particular DCF, and a secondary EP describes an ImF. We have described five primary EPs and seven secondary EPs. All these EPs have been derived based on two pillars, i.e., survey results and practical experiences gathered in several PAIS engineering projects in the automotive domain. Selected EPs have been additionally validated based on survey results. Note that we do not claim for completeness here, i.e., we are continuously working on the extension of our pattern collection (as well as their validation).

Chapter 7

Methodology Governing

7.1 Motivation

To facilitate the introduction of the EcoPOST framework in practice, the provision of governing guidelines and best practices becomes necessary. These guidelines and best practices can provide additional support for users of the EcoPOST framework. This chapter picks up this issue and summarizes governing guidelines and best practices we derived during the development of the EcoPOST framework, from its initial use in practice (cf. Chapter 11), and from general System Dynamics (SD) guidelines described in the literature [189].

We distinguish between three classes of governing guidelines focusing on different aspects related to the application of the EcoPOST framework (cf. Fig. 7.1): (1) guidelines for dealing with evaluation and simulation models in general, (2) guidelines focusing on the handling of dynamic evaluation factors (i.e., DCFs and dynamic ImFs), and (3) guidelines for dealing with *evaluation patterns* (EP).

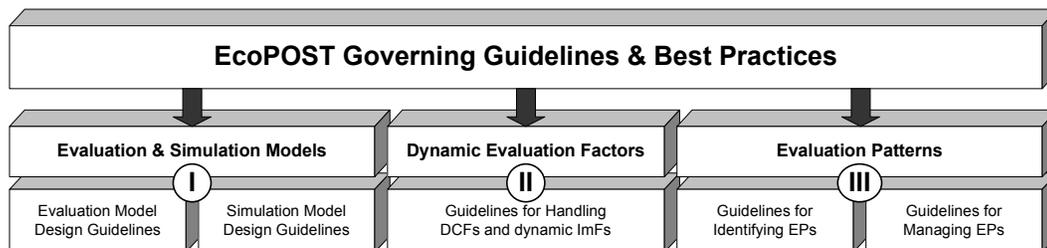


Figure 7.1: EcoPOST Governance Guidelines.

The remainder of this chapter is organized as follows. Section 7.2 presents guidelines for designing evaluation models and Section 7.3 describes guidelines for developing simulation models. Section 7.4 summarizes guidelines for handling DCFs and dynamic ImFs. Section 7.5 deals with the governing of EPs. Section 7.6 discusses why governing guidelines adopt a key role when transferring the EcoPOST framework into practice. Section 7.7 concludes with a summary.

7.2 Guidelines for Designing Evaluation Models

The design of sophisticated evaluation models can constitute a complex and time-consuming task. EcoPOST evaluation models can become very large, e.g., due to the potentially high number of evaluation factors to be considered or due to the large number of causal dependencies existing between those factors. To cope with this complexity, we introduce guidelines for designing evaluation models (cf. Table 7.1).

Guideline	Description
<i>EM-1</i>	Carefully distinguish between SCFs and DCFs.
<i>EM-2</i>	When it is unclear how to represent a given cost factor represent it as SCF.
<i>EM-3</i>	Name feedback loops.
<i>EM-4</i>	Use meaningful names (in a consistent notation) for cost and impact factors.
<i>EM-5</i>	Ensure that all causal links in an evaluation model have unambiguous polarities.
<i>EM-6</i>	Choose an appropriate level of detail when designing evaluation models.
<i>EM-7</i>	Do not put all feedback loops into one large evaluation model.
<i>EM-8</i>	Focus on interaction rather than on isolated events when designing evaluation models.
<i>EM-9</i>	An evaluation model does not contain feedback loops comprising only auxiliary variables.
<i>EM-10</i>	Perform empirical and experimental research to generate needed data.

Table 7.1: Guidelines for Designing Evaluation Models.

In the following, we discuss these guidelines in detail:

Guideline EM-1 (SCF vs. DCF): *Carefully distinguish between SCFs and DCFs.* The distinction between SCFs and DCFs is a fundamental principle in the EcoPOST framework. Yet, it can be difficult for the EcoPOST user to decide whether a cost factor should be treated as static or dynamic. Consider, for example, an evaluation scenario which deals with the introduction of a new PAIS "CreditLoan" to support the granting of loans at a car bank. Based on the new PAIS, the entire loan offer process for financing car purchases shall be supported. For this purpose, the PAIS has to leverage internal (i.e., within the bank) and external (e.g., the car dealer) trading partners as well as other legacy applications for customer information and credit ratings. Among other things, this necessitates the integration of existing legacy applications. In case this integration is done by external suppliers, resulting costs can be represented as SCF as they can be clearly quantified based on a contract or a service level agreement. If integration is done in-house, however, integration costs should be represented as DCF as costs will be influenced by additional ImFs in this case.

Guideline EM-2 (SCF): *When it is unclear how to represent a given cost factor, represent it as SCF.* This can be considered as a direct consequence of the problem addressed by guideline EM-1. An evaluation model will be easier to understand if it comprises more SCFs and less DCFs. Also, SCFs are easier to specify in a simulation model (i.e., by simple constant equations).

Guideline EM-3 (Feedback Loops): *Name feedback loops.* Important enablers of dynamic behavior in our evaluation models are feedback loops (cf. Section 5.2). Particularly if several feedback loops appear in one evaluation model, it will be difficult to keep track of the effects caused by the feedback loops – not only for the modeler himself, but also for other people who may participate in the modeling process (e.g., domain experts, software developers or project managers). Reinforcing (or positive) loops can be additionally marked with an "R" (cf. Fig. 7.2A). Self-correcting (or negative) loops, in turn, can be marked with a "S" (cf. Fig. 7.2B). To simplify the tracking of feedback loops, it is also useful to give each identified feedback loop a unique and meaningful name (cf. Fig. 7.2C). This particularly helps to understand the basic effect of a feedback loop and provides helpful shorthand information during discussions.

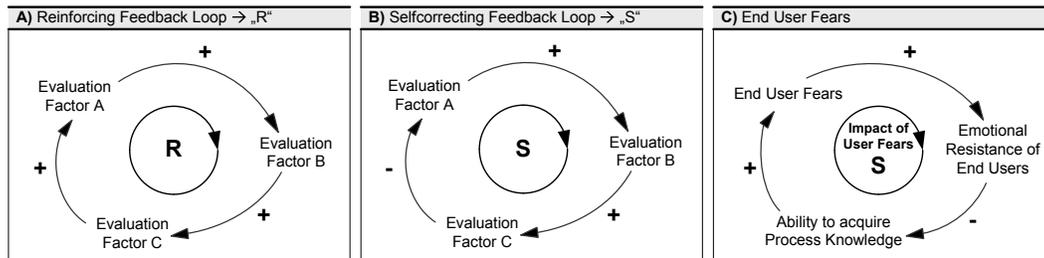


Figure 7.2: Naming Feedback Loops (Simplified Diagrams).

Guideline EM-4 (Naming Convention): *Use meaningful names (in a consistent notation) for cost and impact factors.* In order to improve the understanding of evaluation models, choose precise, non-repetitive terms for model variables. In particular, names of cost and impact factors should be nouns or noun phrases, while variable names including terms like "more", "less", "increases", or "decreases" should be avoided. It can be very hard to interpret a decreasing dynamic ImF "More Process Knowledge" or an increasing dynamic ImF "Less Domain Knowledge".

Guideline EM-5 (Link Polarities): *Ensure that all causal links in an evaluation model have unambiguous polarities.* It can be confusing (and it is disallowed) to ascribe a causal link with both a positive and a negative label. If it seems to be difficult to assign a clear and unambiguous polarity to a link, this will usually mean that there exists more than one causal pathway connecting the two considered model variables. In such a situation, the introduction of additional links with unambiguous polarities can be useful – as long as existing model design rules on transitive link dependencies (cf. Section 4.9.3) are kept in mind.

Guideline EM-6 (Level of Detail): *Choose an appropriate level of detail when designing evaluation models.* Evaluation models can become complex due to the potentially large number of SCFs, DCFs, and ImFs, and due to causal dependencies that exist between them. Complexity will further increase if too much information is incorporated into an evaluation model. Thus, it can be helpful to intentionally reduce the number of considered cost and impact factors and to focus on the analysis of specific effects.

Guideline EM-7 (Number of Feedback Loops): *Do not put all feedback loops into one large evaluation model.* For the same reasons as described in the context of guideline EM-6, it can be useful to separately analyze different feedback loops in different evaluation models. Note that the complexity of evaluation models significantly increases with the number of considered feedback loops.

Guideline EM-8 (Interaction): *Focus on interaction rather than on isolated events when designing evaluation models.* When designing evaluation models, the interaction of cost and impact factors is more important than isolated events like the emergence of a specific ImF. EcoPOST evaluation models aim at the analysis of dynamic effects resulting from interacting cost and impact factors (not from the occurrence of single events). This is generally one fundamental postulate in system sciences [76].

Guideline EM-9 (Cycles of Auxiliary Variables): *An evaluation model does not contain feedback loops comprising only auxiliary variables.* Feedback loops always have to include at least one DCF or dynamic ImF. Feedback loops consisting only of auxiliary variables are disallowed. We have already discussed this issue in the context of model design rules (cf. Section 4.9.3). However, as this is often a source for flaws in evaluation models, we explicitly mention it here again.

Guideline EM-10 (Empirical and Experimental Research): *Perform empirical and experimental research to generate needed data.* For deriving causal dependencies between cost and impact factors and for validating simulation models, results of empirical and experimental research can be helpful. Examples for the former are given in Chapter 4, Chapter 10, and Chapter 11. An example for the latter is given in Chapter 9.

All these guidelines can support the EcoPOST users during the design of evaluation models. The following governing guidelines, in turn, focus on simulation models.

7.3 Guidelines for Developing Simulation Models

Besides the design of evaluation models, their simulation also constitutes a complex task. To deal with this complexity, the guidelines shown in Table 7.2 can be useful.

Guideline	Description
SM-1	Ensure that all equations of a simulation model are dimensionally consistent.
SM-2	Do not use embedded constants in equations.
SM-3	Choose appropriately small time steps for simulation.
SM-4	All dynamic evaluation factors in a simulation model must have initial values.
SM-5	Use appropriate initial values.
SM-6	Initial values for rate variables need not be given.
SM-7	The validity of evaluation models and simulation outcomes is a relative matter.

Table 7.2: Guidelines for Developing Simulation Models.

In the following, we discuss these guidelines in detail:

Guideline SM-1 (Dimensional Consistency): *Ensure that all equations of a simulation model are dimensionally consistent.* In other words, a simulation model will be only executable (i.e., computable), if the left and right hand side of each equation have the same unit of measure. Thereby, units of measure are treated (and simplified) as known from algebraic symbols. As examples of typical units of measure consider "Costs [\$]" (e.g., for integral equations specifying DCFs) or dimensionless impact ratings, e.g., for auxiliary equations combining the effects of several ImFs (cf. Chapter 5).

Guideline SM-2 (Embedded Constants): *Do not use embedded constants in equations.* It is often tempting to simplify equations by embedding numeric constants in them. Consider, for example, the equation for the auxiliary variable "Adjusted Process Analysis Costs" in Fig. 7.3A: *Adjusted Process Analysis Costs = Planned Process Analysis Costs * 1.4*. In this equation, the factor "1.4" is an embedded constant which shall represent an additional risk adjustment factor. A good simulation model, however, makes any constants explicit by means of SCFs or static ImFs. Returning to our example, a better option is thus to introduce an additional static ImF "Risk Adjustment Factor" and to define the simulation model without embedded constant (cf. Fig. 7.3B): *Adjusted Process Analysis Costs = Planned Process Analysis Costs * Risk Adjustment Factor* and *Risk Adjustment Factor = 1.4*. Note that making constants explicit also simplifies the change of constants between different simulation runs as only constant equations of a simulation model need to be adapted.

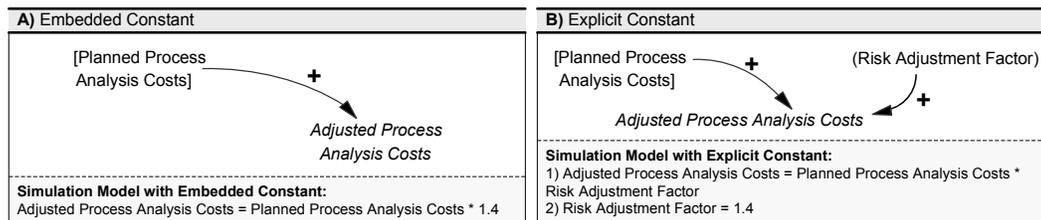


Figure 7.3: Embedded Constants.

Guideline SM-3 (Simulation Time Steps): *Choose appropriately small time steps for simulation.* To a large degree, the expressiveness of simulation results depends on the use of appropriately small time steps when computing a simulation. Suitable time steps for simulations are *days* (for short-lived activities) or *weeks* (for long-lived activities), whereas *months*, *quarters*, or *years* should not be used. As an example consider a business process redesign activity which shall be performed during a period of three months. Using "months" as simulation time steps will be useless. No dynamic behavior will be observable in this case as the time step is far too large. However, if we use "days" as time steps there will be a higher chance to unfold potential dynamic effects.

Guideline SM-4 (Initial Values I): *All dynamic evaluation factors in a simulation model (i.e., DCFs and dynamic ImFs) must have initial values.* This is a major prerequisite for starting the computation of a simulation as the first values of rate variables are determined by the initial values of the model's state variables (cf. Section 5.4). In practice, this does not constitute a problem as this claim is automatically tested by contemporary SD modeling and simulation tools.

Guideline SM-5 (Initial Values II): *Use appropriate initial values.* Besides the definition of initial values (cf. Guideline SM-4), it is likewise important to chose *appropriate* initial values. An initial value will be considered as "appropriate" if it is clearly connected to the current evaluation scenario (cf. Section 4.5). Consider, for example, a dynamic ImF "Process Knowledge" ranging from 0% (no process knowledge) to 100% (high process knowledge). An initial value for this ImF will be always context-specific and can be related, for example, to the assumed level of process knowledge of project staff at simulation start.

Guideline SM-6 (Initial Values III): *Initial values for rate variables need not be given.* Reason is that initial values for rate variables are fully determined by the initial values of the DCF and dynamic ImFs at the beginning of a simulation.

Guideline SM-7 (Model Validity): *The validity of an evaluation model and of simulation outcomes is a relative matter.* The usefulness of an evaluation model and simulation results need always to be judged in comparison with mental and/or descriptive models which would be otherwise used. Often, there exist controversies on the question whether or not reality is exactly presented in an evaluation model. However, such controversies miss the first purpose of a model, namely, to provide concrete statements that can be easily communicated (cf. Section 4.9.5).

7.4 Guidelines for Handling Dynamic Evaluation Factors

This section summarizes governing guidelines for utilizing dynamic evaluation factors. DCFs and dynamic ImFs are typically the variables of interest in our evaluation models, but are also the most complex part to be handled. The guidelines shown in Table 7.3 can support the user in adequately handling dynamic evaluation factors (some of the following issues have been already addressed before).

Guideline	Description
DEF-1	Reduce the number of DCFs and dynamic ImFs as far as possible.
DEF-2	DCFs and dynamic ImFs only change through their assigned rate variables.
DEF-3	Use behavior-over-time-graphs to understand DCFs and dynamic ImFs.
DEF-4	Dynamic ImFs should not be connected among each other based on flows.

Table 7.3: Guidelines for Handling Dynamic Evaluation Factors.

In the following, we discuss these guidelines in detail:

Guideline DEF-1 (Number of Dynamic Factors): *Reduce the number of DCFs and dynamic ImFs as far as possible.* An evaluation model may contain a large number of dynamic evaluation factors relevant for the given evaluation scenario. However, each additional dynamic evaluation factor leads to an increased complexity of the respective evaluation model (cf. Guidelines EM-1 and EM-2 in Section 7.2). In particular, complexity of simulation models significantly increases with the number of dynamic evaluation factors. Thus it can be useful to intentionally reduce the number of dynamic evaluation factors (cf. Guidelines EM-6 and EM-7 in Section 7.2).

Guideline DEF-2 (Rate Variables): *DCFs and dynamic ImFs only change through rate variables.* There exist no causal links directly pointing to DCFs or dynamic ImFs. Dynamic evaluation factors only change through rate variables which alter their values. Note that we have already addressed this issue in the context of model design rules (cf. Section 4.9.3). However, as this important rule is often violated, we explicitly mention it here again.

Guideline DEF-3 (Behavior-Over-Time-Graphs): *Use behavior-over-time-graphs to understand DCFs and dynamic ImFs.* In Section 4.5, we have introduced behavior-over-time-graphs as a useful aid to describe correlations between evaluation factors as well as to make the (assumed) behavior of dynamic evaluation factors explicit. Using such diagrams can be of significant value when investigating DCFs and dynamic ImFs.

Guideline DEF-4 (Dynamic ImFs): *Dynamic ImFs should not be connected among each other based on flows.* Unlike DCFs, whose dimensional consistency is easy to ensure, dynamic ImFs should not be connected using flows (though it is generally allowed). For example, it would be semantically incorrect if a dynamic ImF "Process Knowledge" had been directly connected with a flow to a dynamic ImF "End User Fears".

DCFs and dynamic ImFs are the most important parts of EcoPOST evaluation models as they are typically the variables of interest. The discussed governing guidelines support the modeler in handling these factors.

7.5 Guidelines for Evaluation Patterns

In order to simplify the design of evaluation models and to enhance model reuse, Chapter 6 has introduced the notion of *evaluation patterns* (EP). If appropriately used, EPs will be a powerful means to ease the design of evaluation models and related simulation models. To support their use, this section summarizes guidelines for working with EPs (cf. Table 7.4). Thereby, we distinguish between guidelines for identifying and introducing new EPs (cf. Section 7.5.1) and guidelines for working with existing EPs (cf. Section 7.5.2).

Guideline	Description
EP-1	Document the context-specific constraints related to a new EP.
EP-2	Give each identified EP a unique name.
EP-3	Suitable meta data should be assigned to an identified EP.
EP-4	Focus on proved conclusions.
EP-5	Store EPs in the model repository of the EcoPOST Cost Benefit Analyzer.
EP-6	Carefully distinguish between an EP itself and its adaptation to a specific evaluation context.
EP-7	Customize an EP in a systematic way.

Table 7.4: Guidelines for Identifying and Applying Evaluation Patterns.

7.5.1 Identifying Evaluation Patterns

In Chapter 6, we have described a basic set of primary and secondary EPs. However, the number of EPs is not restricted. To foster the introduction of the EcoPOST framework in practice, it is important to further extend the number of available EPs. Identifying new EPs, however, is difficult. An EP should not only provide facts (like a reference manual or a user's guide), but should also "tell a story" which captures the experience it tries to convey.

Identifying EPs is thus a difficult task to accomplish. In particular, the question to be answered is: How do you know an EP when you come across one? The answer is you do not always know. Often, even if you recognize the beginnings of some things you think are an EP, it will turn out that these things are not an EP at all, or they are only pieces of an EP, or simply good principles or rules of thumb that may form part of the rationale of a particular EP or evaluation model. Yet, we can formulate some guidelines which can be useful for identifying new EPs (cf. Table 7.4):

Guideline EP-1 (Pattern Constraints): *Document the context-specific constraints related to a new EP.* As an example of a respective constraint consider the domain an EP has been derived within (e.g., the automotive domain, health care, etc.). Later, these constraints can be taken into account when deciding whether an EP can be applied to another evaluation scenario or not.

Guideline EP-2 (Pattern Names): *Give each identified EP a unique name.* It is important to give each EP a suitable and unique name.

Guideline EP-3 (Meta Data): *Suitable meta data (e.g., about used SCFs, DCFs, and ImFs) should be assigned to an identified EP.* The author(s) of a new EP should incorporate all comments and insights gained during EP development before presenting it. This will simplify the reuse of the EPs, in particular, if a model repository for managing EPs is used (cf. Section 8.3.3).

Guideline EP-4 (Pattern Validation): *Focus on proved conclusions.* New EPs should describe proved conclusions about causal dependencies and (cost) effects in PAIS engineering projects. To ensure this, it can be useful to discuss an EP candidate (or a set of EP candidates) in an open forum, e.g., in a focus group session.

Guideline EP-5 (Tool Support): *Store EPs in the model repository of the EcoPOST Cost Benefit Analyzer.* It is important to systematically manage EPs. For this purpose, the *EcoPOST Cost Benefit Analyzer* (cf. Chapter 8.3.3) can be used.

7.5.2 Applying Evaluation Patterns

Besides the identification of EPs, their management constitutes another challenging task. Guidelines supporting EcoPOST users in this context are as follows:

Guideline EP-6 (Pattern Customization I): *Carefully distinguish between an EP itself and its adaptation to a specific evaluation context.* An EP is typically the starting point for a context-specific customization. As soon as the EP is changed, do not consider it as an EP anymore, but as a customized instance. The model repository of the *EcoPOST Cost Benefit Analyzer* (cf. Section 8.3.3) also supports this notion and distinguishes between an EP and customized instances.

Guideline EP-7 (Pattern Customization II): *Customize an EP in a systematic way.* Typically, EPs have to be customized when they are applied to a specific evaluation context. This customization should be accomplished in a systematic manner. In Section 6.5, we have already described a generic customization procedure for EPs.

EPs can be a powerful means to optimize the design of evaluation models and the development of simulation models. The described guidelines can support EcoPOST users in unfolding the benefits of this EcoPOST concept.

7.6 Discussion

The governing guidelines and best practices described in this chapter represent a basic set of clues and recommendations for users of the EcoPOST framework. They have been derived based on practical experiences we have gathered during the development of the EcoPOST framework, from its initial use in practice (cf. Chapter 11), and from general SD guidelines described in the literature.

The discussed guidelines support the modeler in designing evaluation models, in building simulation models, in handling dynamic evaluation factors, and in identifying, managing and utilizing EPs. Yet it is important to mention that the consideration of these guidelines does not automatically result in better evaluation and simulation models or in the derivation of more meaningful evaluation results.

Notwithstanding, taking the discussed guidelines into account constitutes an important prerequisite to increase the probability of developing understandable and meaningful evaluation and simulation models. Likewise, a successful application of the EcoPOST framework to practical evaluation scenarios is facilitated. Thus, we strongly recommend to keep the described guidelines in mind when utilizing the EcoPOST framework.

Thereby, it is also important to mention that the described governing guidelines represent an initial, basic set of clues and recommendations rather than a complete collection of

best practices. In particular, these guidelines have been derived based on experiences we gathered in the automotive domain. It cannot be excluded that other governing guidelines and best practices will emerge if our framework is used in other domains.

7.7 Summary

This chapter has presented governing guidelines and best practices for the EcoPOST framework and its underlying evaluation concepts. In particular, the discussed issues shall facilitate the application of the EcoPOST framework in practice. Thereby, we have distinguished between different kinds of governing guidelines: guidelines for designing evaluation models, guidelines for developing simulation models, guidelines for dealing with dynamic evaluation factors, and guidelines for identifying and working with EPs.

Chapter 8

Tool Support

8.1 Motivation

This chapter describes tool support provided for the EcoPOST framework. This constitutes one criterion of our comparison framework described in Chapter 2. Further, tool support has been identified as an important requirement for evaluating PAIS engineering projects (cf. Requirement R-11 in Chapter 3).

This chapter is organized as follows. Section 8.2 summarizes the overall architecture of the EcoPOST tools. Section 8.3 presents the *EcoPOST Cost Benefit Analyzer* in detail, a powerful tool which supports various administrative tasks related to the use of the EcoPOST framework. Section 8.4 concludes with a summary.

8.2 Tool Architecture

Consider Fig. 8.1 which illustrates the EcoPOST tool architecture. Tool support is based on two pillars: (1) a *System Dynamics modeling and simulation tool* and (2) the *EcoPOST Cost Benefit Analyzer*. Both components are discussed in the following.

System Dynamics Tool. In order to design EcoPOST evaluation models as well as to develop and execute simulation models, available *System Dynamics* (SD) modeling and simulation tools can be used. These tools (e.g., *Vensim* [208] or *Powersim* [155]) allow to accomplish systematic "behavioral experiments" based on the temporal change of simulation parameters in a series of simulation runs (i.e., sensitivity analyses). Thereby, the values of all model variables are stored for each simulation run and can be further analyzed using graphical chart tools. Note that the evaluation and simulation models presented in this work have been created with the SD tool *Vensim* [208].

Note that existing SD tools do not support the EcoPOST-specific distinction of DCFs and dynamic ImFs, i.e., they only allow for the modeling of state variables. Likewise, they do not distinguish between SCFs and static ImFs, i.e., they only know constant factors. Yet, it is possible to use these tools to design EcoPOST evaluation and simulation models.

EcoPOST Cost Benefit Analyzer. In order to support administrative tasks related to EcoPOST evaluations we have implemented the *EcoPOST Cost Benefit Analyzer*. This Java-based tool comprises several functional modules (cf. Fig. 8.1): (1) a module for quantifying and visualizing cost evaluations (based on the *JFreeChart Engine* [44]), (2) a module for evaluating the impact of PAIS on work performance based on the *hedonic wage model* (cf. Section 2.5.2), (3) a model repository which enables the storage, management, and reuse of both EPs and entire evaluation scenarios (based on the free Java database *Derby* [57]), and (4) modules for performing *return on investment* (ROI), *net present value* (NPV), and *break even point* calculations. The next section describes these functional modules in detail.

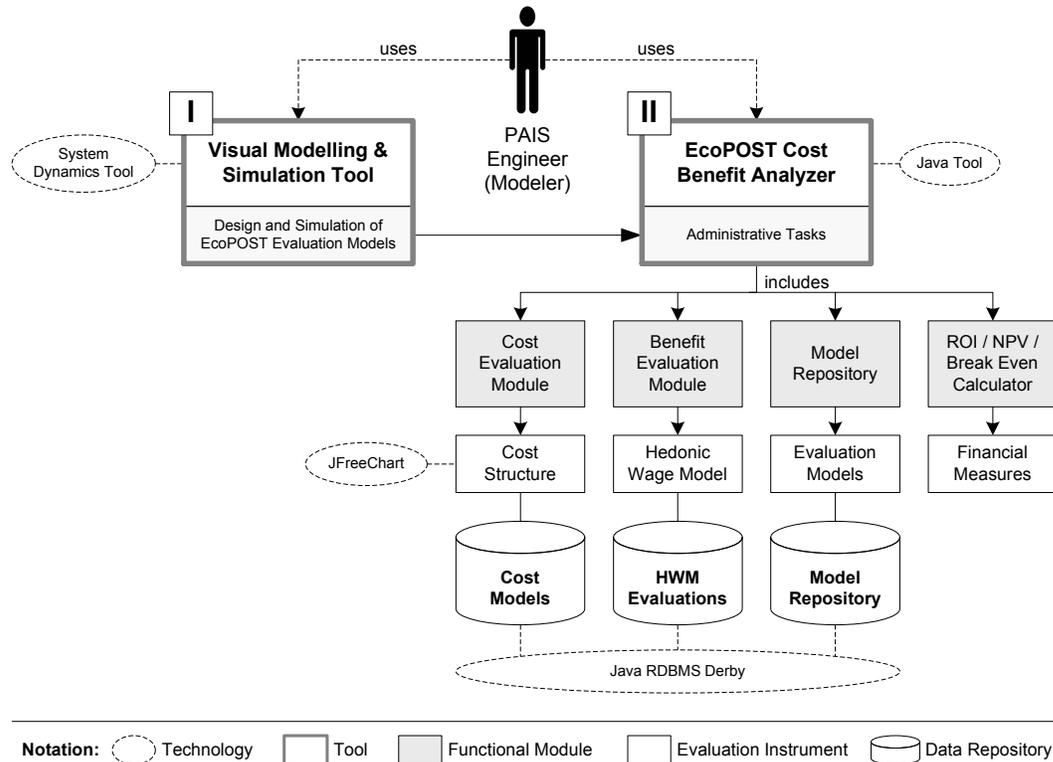


Figure 8.1: EcoPOST Tool Architecture.

Supporting Evaluation Scenarios. The combination of a SD tool and the *EcoPOST Cost Benefit Analyzer* allows to handle entire *evaluation scenarios* (cf. Section 4.5). An evaluation scenario corresponds to the PAIS engineering project to be investigated (or a part of a respective project). The constituting elements of an evaluation scenario, i.e., those elements which are grouped together in the *EcoPOST Cost Benefit Analyzer* and which specify one evaluation scenario, are shown in Fig. 8.2.

One evaluation scenario comprises one *cost evaluation*, one *benefit evaluation*, and a *set of evaluation and simulation models* (cf. Fig. 8.2). A cost evaluation is created with the *cost evaluation module* (cf. Section 8.3.1 for details). The benefit evaluation, in turn, cor-

responds to a work performance analysis created based on the *hedonic wage model module* (cf. Section 8.3.2 for details). The evaluation and simulation models which belong to an evaluation scenario are managed using the *model repository* (cf. Section 8.3.3 for details).

Note that results from ROI, NPV, and break even point calculations (cf. Section 8.3.4 for details) cannot be associated with a specific evaluation scenario, i.e., these calculations are independent of an evaluation scenario. What is possible, however, is to import existing cost or benefit evaluations into the ROI, NPV, and break even point modules.

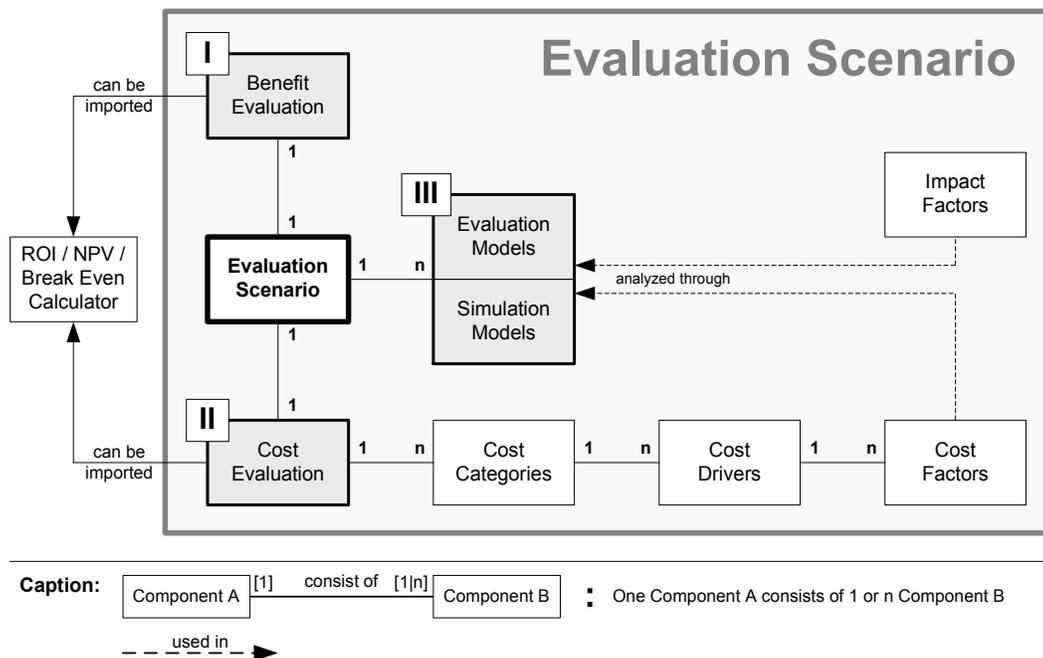


Figure 8.2: Tool Support for Evaluation Scenarios.

8.3 The EcoPOST Cost Benefit Analyzer

8.3.1 Cost Evaluations

The cost evaluation module supports the composition of arbitrary, hierarchical, tree-like *cost models*. One cost model (which is assigned to exactly one evaluation scenario) comprises *n* *cost categories* (cf. Fig. 8.2).

As typical examples of cost categories consider "Software Costs", "Hardware Costs" or "Personnel Costs". Note that a cost category is only a logical node. The costs which are represented by a cost category are aggregated from its subsidiary nodes.

Subsidiary nodes of a cost category are *cost drivers* (cf. Fig. 8.2). As examples of cost drivers consider "Software License Costs" (e.g., as a subsidiary node of the cost category "Software Costs") or "Costs for External Consultants" (e.g., as a subsidiary node of the cost

category "Personnel Costs"). Like a cost category, a cost driver is a logical node, i.e., the costs it represents are aggregated from its subsidiary nodes.

One cost driver, in turn, consists of one or several *cost factors* (cf. Fig. 8.2). A cost factor represents costs that can be quantified in terms of money. As examples consider "Costs for External Consultants from Company A" and "Costs for External Consultants from Company B". Both cost factors can be considered as subsidiary nodes of a cost driver "Costs for External Consultants" (and the cost category "Personnel Costs"). On the cost factor level, we additionally distinguish between *static* and *dynamic* cost factors (i.e., SCFs and DCFs). While the former are easy to quantify, the latter are typically further analyzed based on EcoPOST evaluation models and the outcome of respective simulation runs.

The graphical user interface of the cost evaluation module comprises two parts. On the left hand side, cost models can be created. On the right hand side, existing cost models are visualized based on different graphical tools, e.g., table charts or pie charts.

Note that such cost models pick up the notion of the *work break down approach* (cf. Section 2.7.2). They aim at the preferably detailed fragmentation and characterization of the costs of PAIS engineering projects.

8.3.2 Benefit Evaluations

Though benefit evaluation is not the focus of this thesis, the *EcoPOST Cost Benefit Analyzer* also includes a benefit evaluation module (cf. Fig. 8.3). This module allows to investigate the impact of PAIS on work performance based on the *hedonic wage model* [174]. This evaluation approach has been already described in Section 2.5.2.

Thereby, this module does not only allow to define both the *initial work profile matrix* (specifying the work profile of an organization before an IT investment) and the *second work profile matrix* (specifying the work profile of an organization after an IT investment). It also allows to directly calculate and visualize the impact of an IT investment by comparing the value of considered job levels before and after the IT investment.

8.3.3 Model Repository

The model repository (cf. Fig. 8.4) allows to store, manage and browse EPs as well as entire evaluation scenarios. More specifically, the following features are realized in this module:

- **Database Functionality:** Both EPs and evaluation models (including their simulation models) can be persistently stored in the model repository. Thereby, each EP and each evaluation model can be associated with additional meta data that characterize it in more detail, e.g., lists of used SCFs, ImFs, and DCFs, as well as keywords describing the semantics of a model.
- **Search Functionality:** Based on the meta data associated with EPs and evaluation models, it becomes possible to search for EPs and evaluation models in the model repository. Depending on the selected search scope (e.g., "EP" or "Evaluation Models"), context-specific search fields are provided.

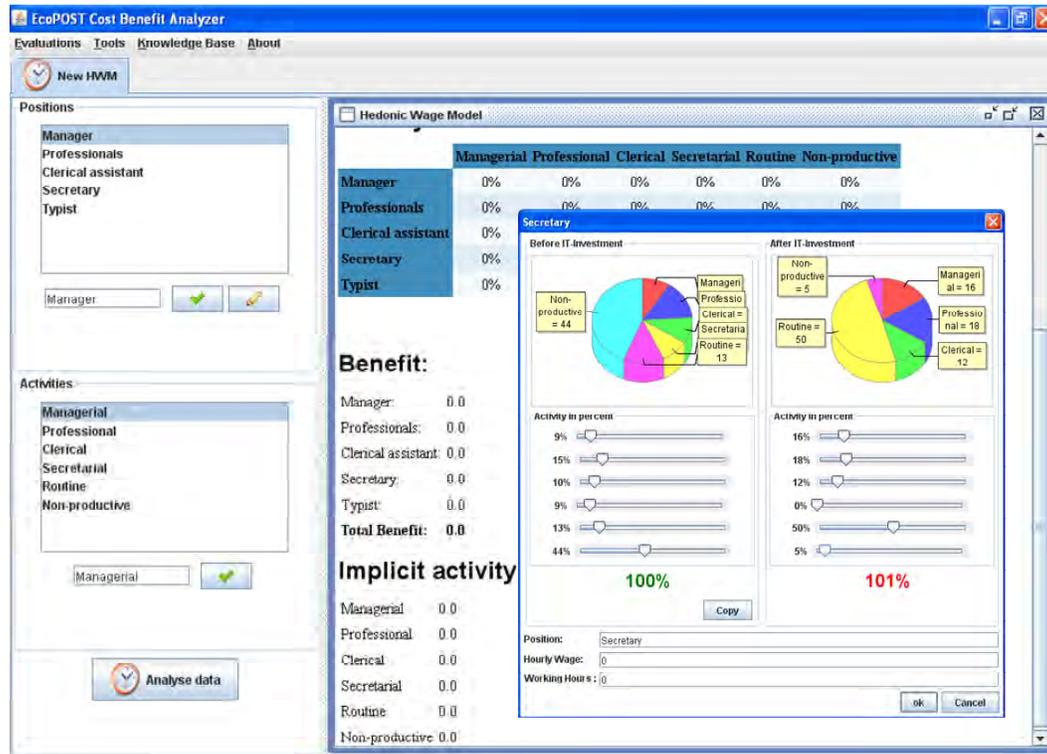


Figure 8.3: Benefit Evaluation Module (Hedonic Wage Model).

- **Preview Functionality:** EPs and evaluation models can be quickly browsed based on a preview function. Depicting previews makes it not necessary to load EPs or evaluation models into *Vensim* in order to take a look at them.
- **Quick Load Functionality:** In combination with the SD tool *Vensim*, it is possible to directly load¹ EPs and evaluation models (i.e., *Vensim* binaries) into the *Vensim* modeling and simulation environment.

The user interface of the model repository comprises three parts (cf. Fig. 8.4). The left column provides mechanisms for navigating within the model repository. The *pattern explorer* allows to browse the available primary and secondary EPs as well as customized instances. The *scenario explorer*, by contrast, allows to browse evaluation scenarios.

If an EP or evaluation model is selected in one of the two browsers in the left column, an additional tab (the *editor tab*) will be shown. This tab allows to manipulate the meta data which are associated with the selected item. Besides, it becomes also possible to explicitly denote information about relationships between EPs and evaluation scenarios in this tab,

¹ If this functionality is used the first time, a dialog will ask for the specific path to the *Vensim* executable in order to activate this functionality. Having denoted this path, *Vensim* can be activated out of the model repository.

e.g., which EP(s) is used in which evaluation scenario. The middle part of the interface (i.e., the *search tab*) provides search features. Finally, the *model viewer* in the right part of the user interface shows a preview of the selected item.

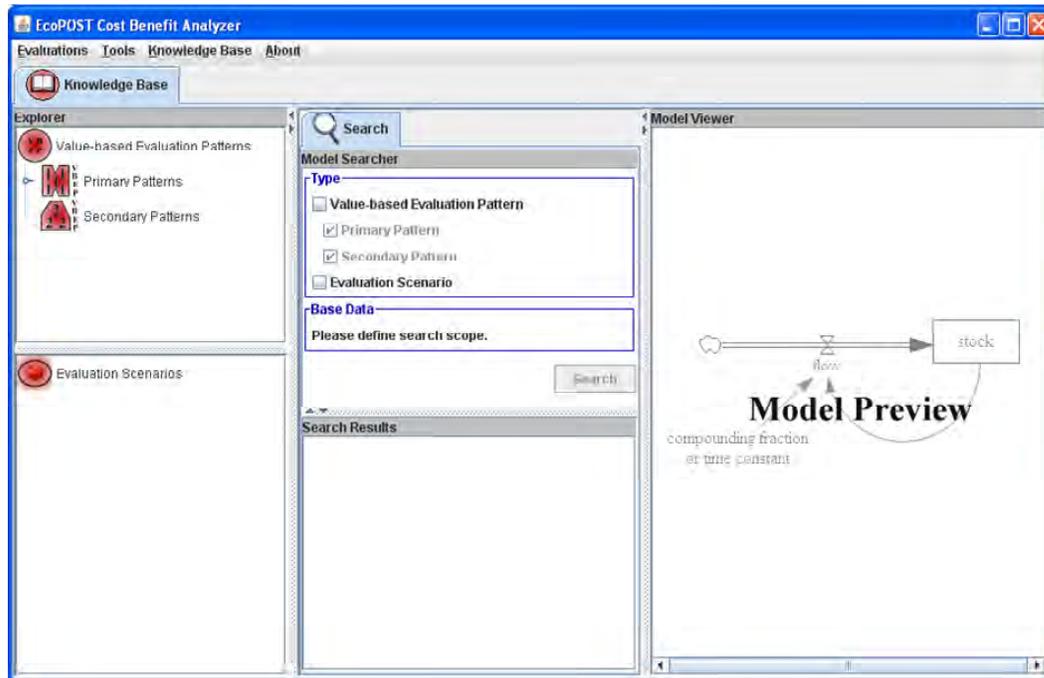


Figure 8.4: Model Repository.

8.3.4 Business Ratios

The *EcoPOST Cost Benefit Analyzer* also includes modules for calculating conventional static (cf. Section 2.4.1) and dynamic (cf. Section 2.4.2) business ratios. Specifically, it includes modules for performing ROI and NPV calculations as well as a module for determining the break even point of an IT investment.

In the ROI module, a calculation can be performed either based on arbitrary, user-defined cost and benefit values, or based on cost and benefit values which can be imported from existing cost or benefit evaluations.

In order to perform a NPV calculation, several parameters have to be specified (cf. Fig. 8.5; left side): (i) the *interest rate* (corresponding to the discount factor), (ii) the *number of periods*, (iii) the *initial cashflow* (i.e., the expected inflow per period), and (iv) the *initial growth rate* (i.e., the expected growth rate per period). See Section 2.4.2 for details.

Having calculated NPV, the break even point of an IT investment can be determined as well. For this purpose, one additional parameter has to be entered: the *initial investment* respectively the costs of the investment (cf. Fig. 8.5; right side).



Figure 8.5: Net Present Value Module and Break Even Point Module.

8.4 Summary

This chapter has stated how tool support in EcoPOST framework looks like. In particular, the EcoPOST tool architecture is based on two pillars: the use of a SD modeling and simulation tool and the *EcoPOST Cost Benefit Analyzer*. The former is applied for designing and simulating EcoPOST evaluation models. The latter is a powerful Java tool to support administrative tasks related to EcoPOST evaluations. It comprises several functional modules: a module for quantifying and visualizing cost evaluations, a module for evaluating the impact of PAIS on work performance based on the *hedonic wage model*, a model repository which enables the storage, management, and reuse of EP and entire evaluation scenarios, and modules for performing ROI, NPV and break even point calculations.

Part III
Validation

Chapter 9

Model Validation through Experimental Research

9.1 Motivation

When building evaluation and simulation models, it is often necessary to rely on hypotheses, sometimes even arguable assumptions. Generally, we face the (common) problem of missing project data. In particular, the quantification of ImFs and their non-linear effects in simulation models has proved to be difficult.

In response to this problem, the accomplishment of experimental and empirical research adopts a key role. Such research activities will become particularly important if analytic solutions are neither possible nor practical and when emerging technologies (instead of mature ones) are considered [124]. However, while empirical research is already accepted in PAIS engineering, experimental research has been neglected so far. This is surprising as the benefits of these research methods have been demonstrated in other software engineering areas for a long time [185]. Examples are software development processes [81] or code reviews [142, 111]. When applying experimental research to the development of PAIS, we expect more valid, quantitative data on related effects. This, in turn, becomes more and more important for IT managers and IT project leaders.

This chapter illustrates how experimental research can be applied to PAIS engineering and how we can use the results of software experiments for validating EcoPOST evaluation and simulation models. For this purpose we have conducted a controlled software experiment with 48 students [140]. Exemplarily, this experiment investigates efforts related to the implementation and change of business processes either using conventional workflow systems [202] or case handling technology [204]. More precisely, we have used Tibco *Staffware* [197] as representative of workflow technology and *FLOWer* [9] as representative of case handling systems. We describe the experiment design, give a mathematical model of the experiment, and discuss potential threats for the validity of experiment results. The results of our experiment help to better understand the efforts for using BPM technology. More important, our experimental framework can serve as a guideline when applying future experimental research in the field of PAIS engineering.

The remainder of this chapter is organized as follows. Section 9.2 gives background information on the performed experiment. Section 9.3 describes its experimental framework (including a mathematical model of the experiment) and discusses potential threats for the validity of experiment results (as well as mitigations). This experimental framework can be taken as a guideline for applying further experimental research in the given context. Section 9.4 describes the performance of the experiment, i.e., its preparation, its execution, its underlying data analysis procedure, and its results. Section 9.5 discusses our results and illustrates how they can be used for quantifying the impact of process knowledge on the efficiency of process implementation in an EcoPOST evaluation model. Section 9.6 discusses related work. Section 9.7 concludes with a summary.

9.2 Background Information

Assume that a business process for refunding traveling expenses shall be supported by a PAIS realized on top of *business process management* (BPM) technology. This *eTravel business process* distinguishes between four roles (cf. Fig. 9.1).

The *traveler* initiates the refunding of his expenses. For this purpose, he has to summarize the travel data in a *travel expense report*. This report is then forwarded either to a *travel expense responsible* (in case of a national business trip) or to a *verification center* (in case of an international business trip). Both the *travel expense responsible* and the *verification center* fulfill the same task, i.e., they verify received travel expense reports. "Verification" means that the declared travel data is checked for correctness and plausibility (e.g., regarding accordance with receipts). Incorrect travel expense reports are sent back to the traveler (for correction). If it is correct, it will be forwarded to the *travel supervisor* for final approval. The supervisor role may be filled, for example, by the line manager of the traveler. If the travel expense report is approved by the supervisor, the refunding will be initiated. Otherwise, it will be sent back to either the travel expense responsible (national trip) or the verification center (international trip). Note that this is a characteristic (yet simplified) process as it can be found in many organizations.

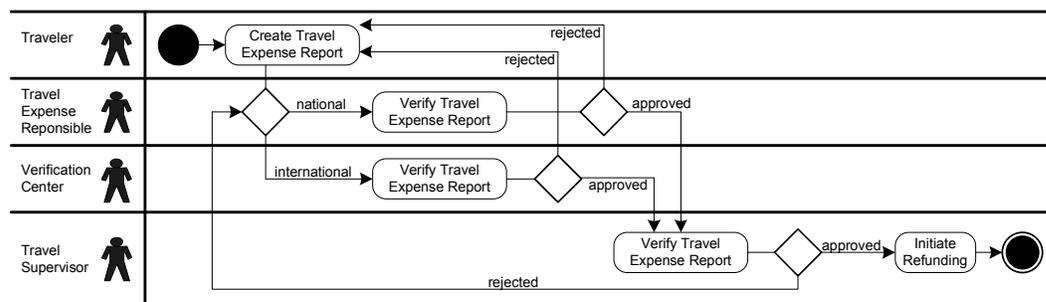


Figure 9.1: The eTravel Business Process.

When developing a PAIS to support this process, one challenge is to select the most adequate BPM technology. Currently, there exist different BPM paradigms, which can be

applied in this context. Among them are *workflow management* and *case handling*.

Workflow Management. Contemporary *workflow management systems* (WfMS) enable the modeling, execution, and monitoring of business processes. Typically, in WfMS-based applications, only the data needed for the execution of an activity is visible to respective actors. This is also known as "context tunneling". WfMS coordinate activity execution based on routing rules, which are described by process definitions and are therefore strictly separated from the processed data. If an activity is completed, the subsequent activities will become active. The worklists of potential actors are updated accordingly. Often, electronic forms are used to implement process activities and to present the data being processed.

Case Handling. An alternative BPM paradigm is case handling [204]. A *case handling system* (CHS) aims at providing more flexibility by avoiding restrictions known from (conventional) workflow technology. Examples of such restrictions include rigid control flow and "context tunneling". The central concept behind a CHS is the case and its data, as opposed to activities and routing rules known from WfMS. CHS always present all data about a case at any time to the user (assuming proper authorization), i.e., context tunneling as known from WfMS is avoided. Furthermore, CHS orchestrate the execution of activities based on the data assigned to a case.

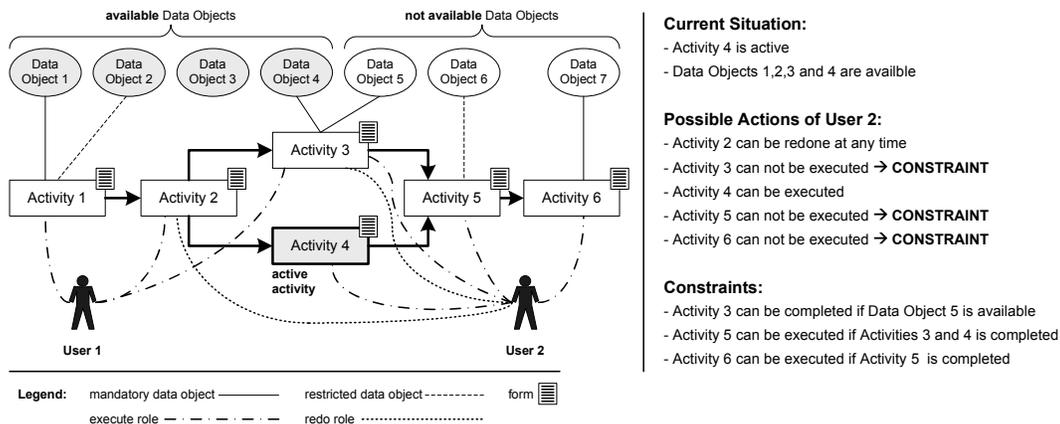


Figure 9.2: Data-driven Case Handling.

Thereby, different kinds of data objects are distinguished (cf. Fig. 9.2). *Free data objects* are not explicitly associated with a particular activity and can be changed at any time while a case is executed (e.g., Data Object 3 in Fig. 9.2). *Mandatory* and *restricted data objects*, in turn, are explicitly linked to one or more activities. If a data object is mandatory for an activity, a value will have to be assigned to it in order to complete the activity (e.g., Data Object 5 in Fig. 9.2). If a data object is restricted for an activity, the activity need to be active for assigning a value to the data object (e.g., Data Object 6 in Fig. 9.2).

Like in WfMS, forms linked to activities are used to provide context-specific views on case data. Thereby, CHS does not only allow to assign an *execution role* to activities, but

also a *redo role* (to undo an executed activity) as well as a *skip role* (to omit the execution of activities). User 2 in Fig. 9.2, for example, may execute Activities 3, 4, 5 and 6, and redo Activities 2 and 3.

Research Question. Fig. 9.3 summarizes the main conceptual differences of workflow management and case handling, and also depicts typical representatives of each paradigm. Despite the conceptual differences, both paradigms can be used for implementing the eTravel process (as well as other many other typical business processes). Thus, the selection of "the most adequate" BPM technology for a given scenario is difficult and depends on context-specific requirements. While some IT managers will consider BPM technology as adequate if experiences are documented, others will take into account more specific decision criteria such as the support of advanced process flexibility features. Likewise, IT managers can be interested in value-based considerations as well. In practice, for example, a frequently asked question is as follows: *Is there a difference in the efforts of implementing a business process with BPM technology A and BPM technology B?* And if "yes": *How strong is this difference?* Dealing with such questions, IT managers typically have to rely on vendor data (e.g., about the return on investment of their products) and qualitative experience reports. What is not available are precise quantitative data about using BPM technology and developing PAIS (e.g., about efforts for implementing business processes).

In order to generate such quantitative data, conducting controlled software experiments offers promising perspectives. In the following, we pick up this issue and exemplarily describe the results of an experiment in which we investigate efforts related to the implementation of business processes using different BPM technologies.

Thereby, we use Tibco *Staffware* [197] (Version 10.1) as typical representative of workflow technology. Its build-time tools include, among other components, a visual process modeling tool and a graphical form editor. The used CHS, in turn, is *FLOWer* [9] (Version 3.1), the most widely used commercial CHS. Like Staffware, FLOWer provides a visual process modeling tool and a form editor.

Criteria for Comparison	WfMS ¹⁾	CHS ²⁾	1) Commercial WfMS	2) Commercial CHS
C1: Basic focus	<i>activity</i>	<i>case</i>	<ul style="list-style-type: none"> - Tibco Staffware, - Microsoft BizTalk Server, - IBM Websphere MQ Workflow, - jBoss jBPM, etc. 	<ul style="list-style-type: none"> - Pallas Athena FLOWer, - Staffware Case Manager, - con:cern (Open Source), etc.
C2: Primary driver for execution of activities	<i>routing rules</i>	<i>case data</i>		
C3: Separation of process control & data	<i>yes</i>	<i>no</i>		
C4: Types of roles associated with tasks	<i>execute</i>	<i>execute, skip, redo</i>		

Figure 9.3: Selected Criteria for Comparing Workflow Management and Case Handling.

9.3 Experimental Framework

Section 9.3.1 discusses basic issues that have to be considered when developing an experiment design. Section 9.3.2 describes the specific design underlying our experiment. Section 9.3.3 discusses factors threatening the validity of experiment results.

9.3.1 Basic Issues

Literature about software experiments [17, 87, 99, 185, 226] provides various design guidelines to set up an experiment. *First*, an experiment design should allow to collect as much data as possible with respect to the major goals of the experiment. *Second*, it should ensure that collected data is unambiguous. *Third*, the experiment must be feasible within the given setting (e.g., within the planned time period). Meeting these design criteria is not trivial. Often, an experiment cannot be accomplished as planned due to a complex experiment design or an insufficient number of participants [185].

Considering these three major criteria, we accomplished our experiment as a *balanced single factor experiment with repeated measurement* (cf. Fig. 9.4). This design is particularly suitable for comparing software development technologies [99]. Specifically, single factor experiments investigate the effects of one *factor*¹ (e.g., a particular software development technology) on a common *response variable* (e.g., implementation efforts). This design also allows to analyze variations of a factor (e.g., two alternative tools for software development). Generally, these variations are called *factor levels*. The response variable is determined when the participants of the experiment (who are also called *subjects*) apply the factor or factor levels to an *object* (e.g., a specification to be implemented, based on a set of requirements).

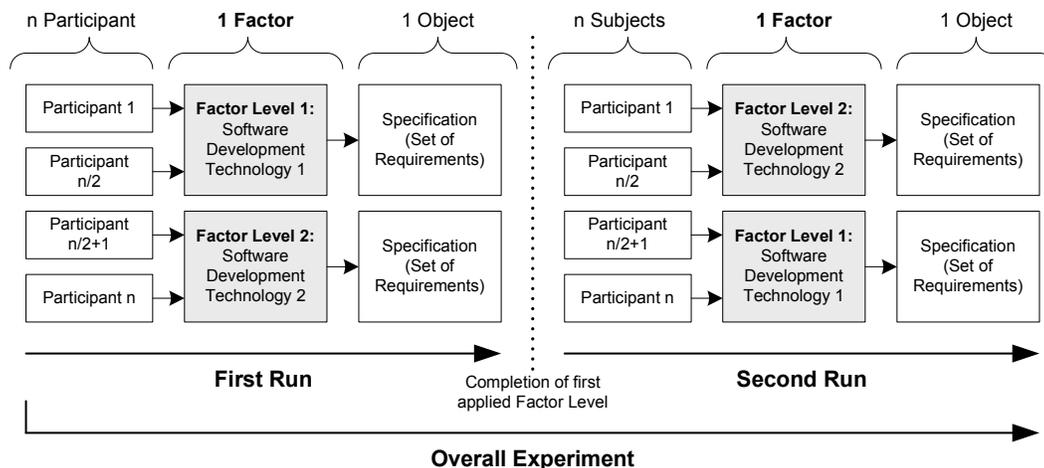


Figure 9.4: Single Factor Experiment.

We will denote a single factor experiment as *balanced* if all factor levels are used by all participants of the experiment. This enables *repeated measurements* and the collection of more precise data as every subject generates data for every treated factor level. Generally, repeated measurements can be realized in different ways. Fig. 9.4 shows a frequently applied variant which is based on two subsequent *runs*. During the *first run* half of the subjects apply "Software Development Technology 1" to the treated object, while the others

¹*Multi factor experiments*, by contrast, investigate the effects of factor combinations on a common response variable, e.g., of a software development technology and a software development process on implementation efforts. However, despite such experiments can improve the validity of experiment results, they are rarely done in practice [156].

use "Software Development Technology 2". After having completed the first run, the *second run* begins. During this run each subject applies that factor level to the object which has not been treated so far.

9.3.2 Experiment Design

Taking the generic experiment design described in Section 9.3.1, our specific BPM experiment design (cf. Fig. 9.6A) comprises different elements:

- **Subjects:** Subjects are 48 students of a combined Bachelor/Master Computer Science course taking place at the University of Innsbruck. These 48 students are divided into 4 *main groups* each consisting of 4 *teams* with 3 students (cf. Fig. 9.5). This results in an overall number of 16 teams. The students are randomly assigned to the teams prior to the start of the experiment.

	Main Group 1	Main Group 2	Main Group 3	Main Group 4
16 Teams	Team 01	Team 03	Team 11	Team 09
	Team 02	Team 04	Team 13	Team 10
	Team 05	Team 07	Team 14	Team 12
	Team 06	Team 08	Team 16	Team 15
	1st Run: WfMS	1st Run: CHS	1st Run: WfMS	1st Run: CHS
	2nd Run: CHS	2nd Run: WfMS	2nd Run: CHS	2nd Run: WfMS

WfMS = Workflow Management System, CHS = Case Handling System

Figure 9.5: Main Groups, Teams, and Students.

- **Object:** The object to be implemented is the *eTravel business process* (cf. Section 9.2). The respective specification comprises two parts: an initial "Base Implementation" (*Part I*) and an additional "Change Implementation" (*Part II*). While the first part deals with the realization of process support for the refunding of national business trips, the second part specifies a process change, namely, additional process support for refunding international business trips. Both parts describe the elements to be implemented: the process logic, user roles, and the data to be presented using simple electronic forms. Note that this specification enables us to not only investigate efforts for initially implementing a business process, but also to examine efforts for subsequent process changes. In our experiment, with "process change" we mean the adaptation of the implemented business process. After having realized such a process change new process instances are based on the new process model. Running process instances are not stopped, i.e., we do not investigate the migration of process instances at runtime [168].
- **Factor & Factor Levels:** *BPM technology* is the considered factor in our experiment with factor levels "WfMS" (Staffware) and "CHS" (FLOWer).
- **Response Variable:** In our experiment the response variable is the *time* the subjects (i.e., the students) need for implementing the given object (i.e., the eTravel specification) with each of the factor levels (WfMS and CHS). All effort values related to the

Staffware implementation are denoted as "WfMS Sample". All effort values related to the FLOWer implementation are denoted as "CHS Sample".

Besides these basic elements, the following issues are important as well:

- **Instrumentation:** To precisely measure the response variable, we have developed an application called *TimeCatcher* (cf. Fig. 9.6D). This "stop watch" allows to log time in six typical "effort categories" related to the development of a process-oriented application: (1) *process modeling*, (2) *user/role management*, (3) *form design*, (4) *data modeling*, (5) *test*, and (6) *miscellaneous efforts*. In order to ensure that developed solutions are correct acceptance test are accomplished by us. Finally, we use a structured *questionnaire* to collect qualitative feedback as well (e.g., on the maturity or usability of the applied WfMS and CHS).
- **Data Collection Procedure:** The TimeCatcher tool is used by the students during the experiment. The aforementioned questionnaire is filled out by the students after the completion of the experiment.
- **Data Analysis Procedure:** For data analysis well-established statistical methods and standard metrics are applied (cf. Fig. 9.6C, cf. Section 9.4.3 for details).

The mathematical model of our experiment can be summarized as follows (cf. Fig. 9.6B): n subjects S_1, \dots, S_n ($n \in \mathbb{N}$) divided into m teams T_1, \dots, T_m ($m \in \mathbb{N}, m \geq 2, m$ even) have to implement the eTravel business process specification. This specification describes a "Base Implementation" O_1 (corresponding to the "national case" of the eTravel process) and a "Change Implementation" O_2 (additionally introducing the "international case"). During the experiment one half of the teams ($T_1, \dots, T_{m/2}$) implements the complete specification (i.e., base and change implementation) using a WfMS (PMS_1 , Staffware), while the other half ($T_{m/2+1}, \dots, T_m$) does this using a CHS (PMS_2 , FLOWer). After finishing the implementation with the first factor level (i.e., the *first run*), each team has to implement the eTravel process using the second factor level in a *second run* (i.e., the development technologies are switched). The response variable "Effort[Time] of T_m implementing O using PMS_j " is logged with the TimeCatcher tool.

9.3.3 Risk Analysis

When accomplishing experimental research, related risks have to be carefully taken into account; in particular when generalizing experiment results. Generally, there exist factors that threaten both the *internal validity* ("Are the claims we made about our measurements correct?") and the *external validity* ("Can we generalize the claims we made?") of experiments. In our context, threats to the internal validity are as follows:

- **People:** Participating students differ in their skills and their productivity for two reasons: (i) general experience with software development and (ii) experience with BPM technology. The first issue can only be balanced by conducting the experiment with

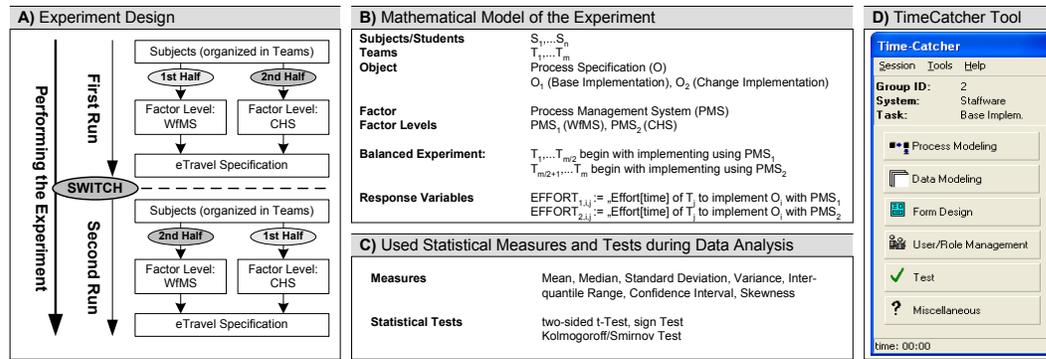


Figure 9.6: Experiment Design, Mathematical Model, and TimeCatcher Tool.

a sufficiently large and representative set of students. The number of 48 students promises to achieve such a balance. The second issue can be mitigated by picking BPM technologies that are unknown to every student. Only three of the participating students have rudimental (and thus negligible) workflow knowledge. However, as we cannot exclude that this knowledge influences our experiment results, we have assigned those three students to different teams in order to minimize potential effects as far as possible.

- **Data collection process:** Data collection is one of the most critical threats. To mitigate this threat, we need to continuously control data collection in the context of the experiment. We further have to ensure that students understand which TimeCatcher categories (cf. Fig. 9.6C) have to be selected during the experiment.
- **Time for optimizing an implementation:** The specification to be implemented does not include any guidelines about the number of electronic forms or the layout. This implies the danger that some teams spend more time for a "nice" user interface than others do. To minimize such effects, we explicitly indicate to the students that the development of a "nice" user interface is not a goal of the experiment. In order to ensure that the implemented solutions are similar across different teams, we accomplish acceptance tests.

Besides, there are threats to the external validity of our results:

- **Students instead of professionals:** Involving students instead of IT professionals may be critical. However, it has been shown before that the results of student experiments are transferable and can provide valuable insights into an analyzed problem domain [81, 171]. Also note that the use of professional software developers is hardly possible in practice as no profit-oriented organization will simultaneously implement a business process twice using two different BPM technologies.
- **Investigation of tools instead of concepts:** Commercial BPM tools (i.e., Staffware and FLOWer) are used as representatives for the analyzed concepts (i.e., workflow

management and case handling). Investigating the concepts therefore always depends on the quality of the used tools. To mitigate this risk, the used BPM technologies should be representative for state-of-the-art technologies in practice. The selected BPM tools (i.e., Staffware and FLOWer) are both market leader in their domain and can thus be considered as suitable.

- **Choice of object:** In order to avoid that the chosen business process setting strongly supports the goals of our experiment, we have picked a business process that can be found in many organizations, i.e., the eTravel business process (cf. Section 9.2).

9.4 Performing the Experiment

This section deals with the preparation, execution and analysis of our experiment (cf. Fig. 9.7). This also includes the presentation of experiment results.

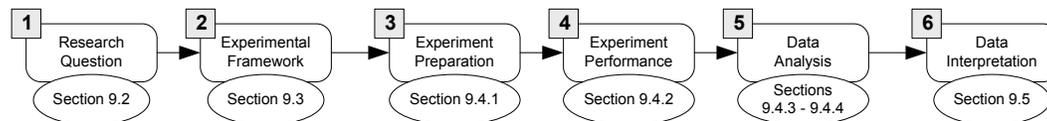


Figure 9.7: Performing the Experiment.

9.4.1 Experiment Preparation

In the run-up of the experiment, we prepare a technical specification of the eTravel process. This specification comprises UML activity diagrams², an entity relationship diagram describing the generic data structure of a travel expense report, and system-specific data models for the considered tools (Staffware, FLOWer). In order to ensure that the specification is self-explanatory and correct, two student assistants are involved in its development.

The same two students also implement the specification with each of the utilized BPM technologies. This allows us to ensure the feasibility of the general setup of the experiment and to identify critical issues with respect to the performance of the experiment. The pre-test also provides us with feedback that helps us to further improve the comprehensibility of our specification. Finally, we compile a "starter kit" for each participating team. It includes original tool documentation, additional documentation created by us when preparing the experiment (and which can be considered as a compressed summary from the original documentation), and the technical specification.

²One may argue that the use of UML activity diagrams can undermine the validity of the experiment as these diagrams are very similar to the explicit, flow-driven notation of Staffware process models, but different from the implicit, more data-driven FLOWer process models. However, in practice, UML activity diagrams are widely used in practice to describe standard business processes. Thus, the use of UML activity diagrams can even improve internal validity as a typical standard scenario from practice is investigated.

9.4.2 Experiment Execution

Due to infrastructure limitations, we split up the experiment in two events. During each event 24 students participated. While the first one took place in October 2006, the second one has been conducted in January 2007. Each event lasted 5 days, and was based on the following procedure:

- **Introductory coaching:** Prior to the start of the experiment, all students have to attend an introductory lecture. We introduce to them basic notions of workflow management and case handling. We further inform them about the goals and rules of the experiment. Afterwards, each team receives its "starter kit".
- **Implementing eTravel process:** The students have to implement the given eTravel business process specification (with both considered factor levels).
- **Accomplishing acceptance test:** After having implemented the eTravel specification with a factor level, an acceptance test is accomplished by us in order to ensure that the developed solution corresponds to the specification. In this acceptance tests, all potential execution paths of the eTravel business processes are tested. If the acceptance fails (e.g., when the developed solution is not working properly), the students have to correct the remaining errors and an additional acceptance test is performed.
- **Filling out questionnaire:** After finishing work on the experiment, every student has to fill out the aforementioned questionnaire.

We further optimize experiment results by applying *Action Research* [159]. Action Research is characterized by an intensive communication between researchers and subjects. At an early stage, we optimize the data collection process by assisting and guiding the students in using the TimeCatcher data collection tool properly (which was critical with respect to the quality of the gathered data). Besides, we document emotional reactions of the students regarding their way of working. This helps us to design the questionnaire. Note that Action Research does not imply any advice for the students on how to implement the eTravel business process.

9.4.3 Data Analysis Procedure

Data analysis comprises three steps: an initial validation of the collected data (*Step 1*), data analysis itself (*Step 2*), and analysis of the questionnaire results (*Step 3*).

Step 1: Data Validation. We validate the collected data regarding its *consistency* ("Is all expected data available?") and *plausibility* ("Is all available data meaningful?"):

- **Data Consistency:** We have to discard the data of two teams as their data is flawed. Both have made mistakes using the TimeCatcher tool. Hence, the data provided by 14 teams is finally included in data analysis.

- Data Plausibility:** We analyze data plausibility based on *box-whisker-plot diagrams* and *scatter-plot diagrams*. Generally, such diagrams visualize the distribution of a sample and particularly show outliers. A low number of outliers indicates plausible data [156]. Fig. 9.8A, for example, shows a box-whisker-plot diagram which illustrates the distributions of the base implementation efforts in our experiment. The diagram takes the form of a box that spans the distance between the 25% quantile and the 75% quantile (the so called *interquartile range*) surrounding the median which splits the box into two parts. The "whiskers" are straight lines extending from the ends of the box to the maximum and minimum values. Outliers are defined as data points beyond the interquartile range, i.e., beyond the edge of the box. As can be seen in Fig. 9.8A, there are no outliers, i.e., all data from these samples lie within the boxed area. Moreover, there exists only one (negligible) outlier in the distribution of the change implementation efforts (cf. Fig. 9.8B), and no outliers regarding the distribution of the overall implementation efforts (cf. Fig. 9.8C).

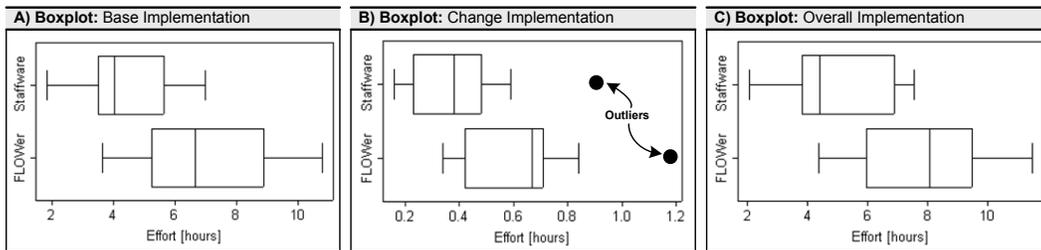


Figure 9.8: Distribution of Samples (Box-Whisker-Plot Diagrams).

Fig. 9.9A shows a two-dimensional scatter-plot diagram illustrating the correlation of WfMS base implementation efforts (horizontal axis) and change implementation efforts (vertical axis). Fig. 9.9B additionally visualizes which data is generated at which run of the experiment. Finally, Fig. 9.9C shows the CHS sample.

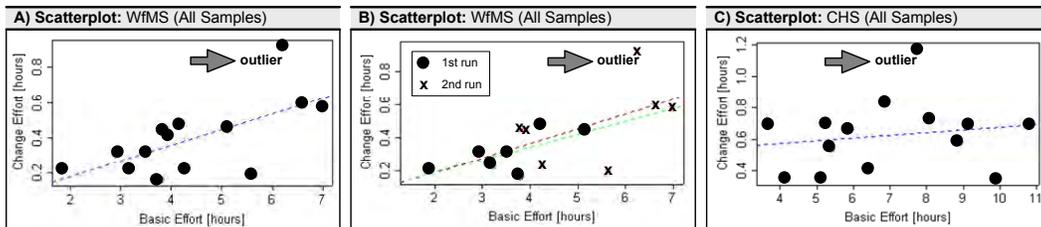


Figure 9.9: Distribution of Samples (Scatter-Plot Diagrams).

Step 2: Data Analysis. The main goal of the experiment is to investigate whether there is a significant difference between the efforts of implementing a business process with workflow technology and the efforts of an implementation using case handling technology. Hence, the 0-hypothesis to be analyzed is as follows: "*Using workflow technology yields no significant difference in implementation efforts when being compared to case handling technology*".

We analyze this 0-hypothesis based on a *two-sided t-test* [156] (respectively an additional sign test if the t-test fails). Doing so, we are able to assess whether the means of the WfMS sample and the CHS sample are statistically different from each other. A successful t-test (with $|T| > t_0$) rejects our 0-hypothesis. Specifically, the following steps have to be executed in order to accomplish a t-test (with $\alpha = 0.05$ as the level of significance):

1. **Paired Comparison:** The t-test is combined with a *paired comparison* [156], i.e., we analyze "pairs of effort values". Each pair comprises one effort value from the WfMS sample and one from the CHS sample. The *Paired Comparison 1* in Fig. 9.10, for example, combines effort values from the Main Groups 1 and 3 with effort values from the Main Groups 2 and 4 (the precise pairs are shown in Fig. 9.11). Note that we compose pairs according to the performance of the teams, i.e., effort values of "good" teams are not combined with effort values of "bad" teams (cf. [100]).

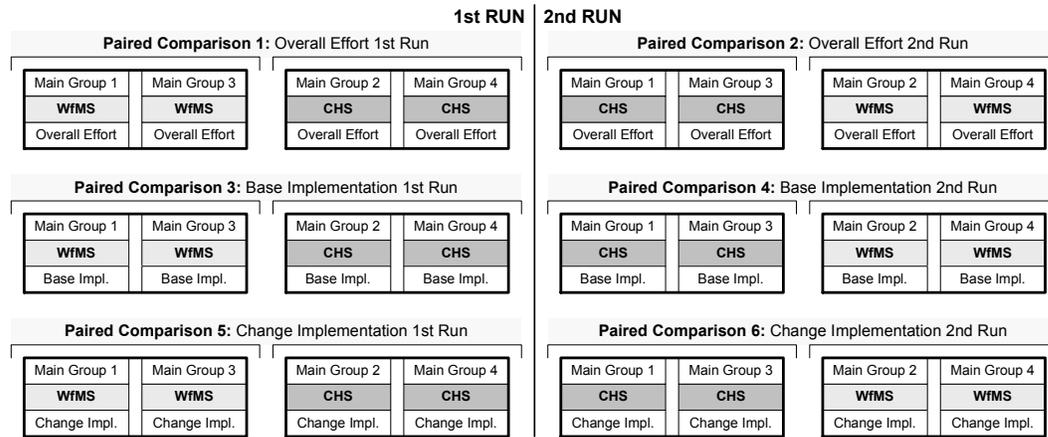


Figure 9.10: Paired Comparisons.

2. **Standardized Comparison Variable:** For each pair, a *standardized comparison variable* X_j is derived [89]. It is calculated by dividing the difference of the two compared effort values by the first one (cf. Fig. 9.6 in Section 9.3.2):

$$X_j := \frac{EFFORT_{j+m/2} - EFFORT_j}{EFFORT_{j+m/2}} \cdot 100\%$$

In other words, X_j denotes how much effort team T_j saves using workflow technology compared to team $T_{j+m/2}$ which uses case handling technology. Together, all X_j constitute a standardized comparison sample $x = (X_{11}, \dots, X_{1m/2})$ which is used as basis when performing the t-test.

3. **Statistical Measures:** For the standardized comparison sample x we calculate the *median* (m), the *interquantile range* (IQR), the *expected value* (μ), the *standard deviation* (σ), and the *skewness* (sk).

4. **Two-sided t-Test:** Finally, we apply the t-test to x . Note that the t-test will be only possible if x emanates a *normal distribution* and if the WfMS and CHS sample have the *same variance*. The first condition can be tested using the *Kolmogoroff/Smirnov test* [184]. In particular, the result of the Kolmogoroff/Smirnov test has to be smaller than K_0 (with K_0 a predefined value depending on the size of x and α). The second condition can be tested based on the *test for identical variance* [184]. The variance of the WfMS and CHS sample will be identical, if the result of this test is smaller than F_0 (with F_0 a predefined value depending on the size of the samples and α). Note that only one violated precondition is sufficient to avoid the accomplishment of the t-test.

Step 3: Questionnaire Analysis. We analyze the data collected based on the questionnaire each student has to fill out. As one of the students became ill at the last day of the January 2007 event, only 47 students participated.

9.4.4 Results

Experiment Results. Fig. 9.11A shows the results for the paired comparison for the *overall implementation efforts* (cf. Paired Comparisons 1 and 2 in Fig. 9.10). When studying the workflow efforts, we can see that efforts are lower than the efforts for the case handling implementation. This difference is confirmed by the results of the (successful) t-tests for both the first and the second run, i.e., our 0-hypothesis is to be rejected.

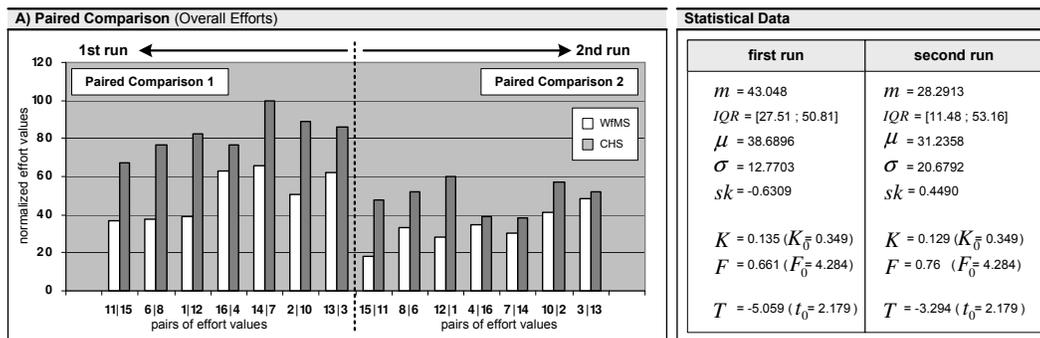


Figure 9.11: Experiment Results (Overall Implementation Efforts).

In the first run, the use of workflow technology has resulted in effort savings of 43.04% (fluctuating between 27.51% and 50.81%) when compared to the efforts for using case handling technology. In the second run, the use of workflow technology has still resulted in savings of 28.29% (fluctuating between 11.48% and 53.16%).

Fig. 9.11A also shows that efforts for the first run are generally higher than those for the second run. Regardless which technology is used first, all teams reduce their efforts in the second run. This can be explained either through learning effects regarding the used BPM technologies or through an increasing process knowledge gathered during the experiment. According to questionnaire results (see below), it can be assumed that this effect is not necessarily related to learning effects concerning the used BPM technologies (i.e.,

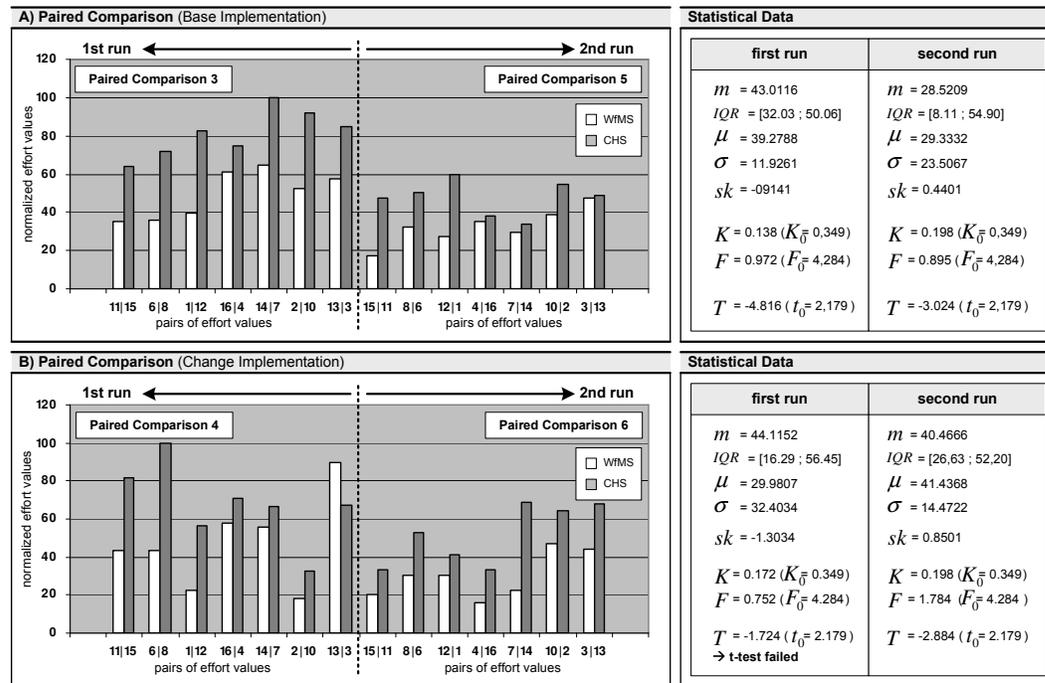


Figure 9.12: Experiment Results (Base and Change Implementation).

tool knowledge), but to increasing process knowledge. This, in turn, mitigates the risk of comparing tools instead of concepts.

Fig. 9.12A and Fig. 9.12B show results for paired comparisons for the *base implementation* (cf. Paired Comparisons 3 and 5 in Fig. 9.10) and the *change implementation* (cf. Paired Comparisons 4 and 6 in Fig. 9.10). Again, our results allow to reject the 0-hypothesis (the failed t-test for Paired Comparison 4 can be compensated with a successful *sign test*). Using workflow technology has resulted in effort savings of 44.11% for the change implementation in the first run (fluctuating between 16.29% and 56.45%). In the second run, the use of workflow technology has resulted in effort savings of 40.46% when compared to the efforts associated with case handling technology. Fig. 9.13 illustrates that workflow efforts are generally much smaller than respective efforts for the case handling implementation.

Questionnaire Results. Fig. 9.14A shows that the methodical soundness of using process management technology is easier to understand in the case of workflow technology, i.e., using case handling technology is considered as being more difficult. Fig. 9.14B illustrates what we have already mentioned above, i.e., process knowledge gained during the first run significantly simplifies the second run. By contrast, Fig. 9.14C shows that the increased efficiency during the second run cannot be related to a gained tool knowledge. Finally, Fig. 9.14D deals with the usability of the applied process management systems. As can be seen, there remains a lot of space for improvement from the students' viewpoint.

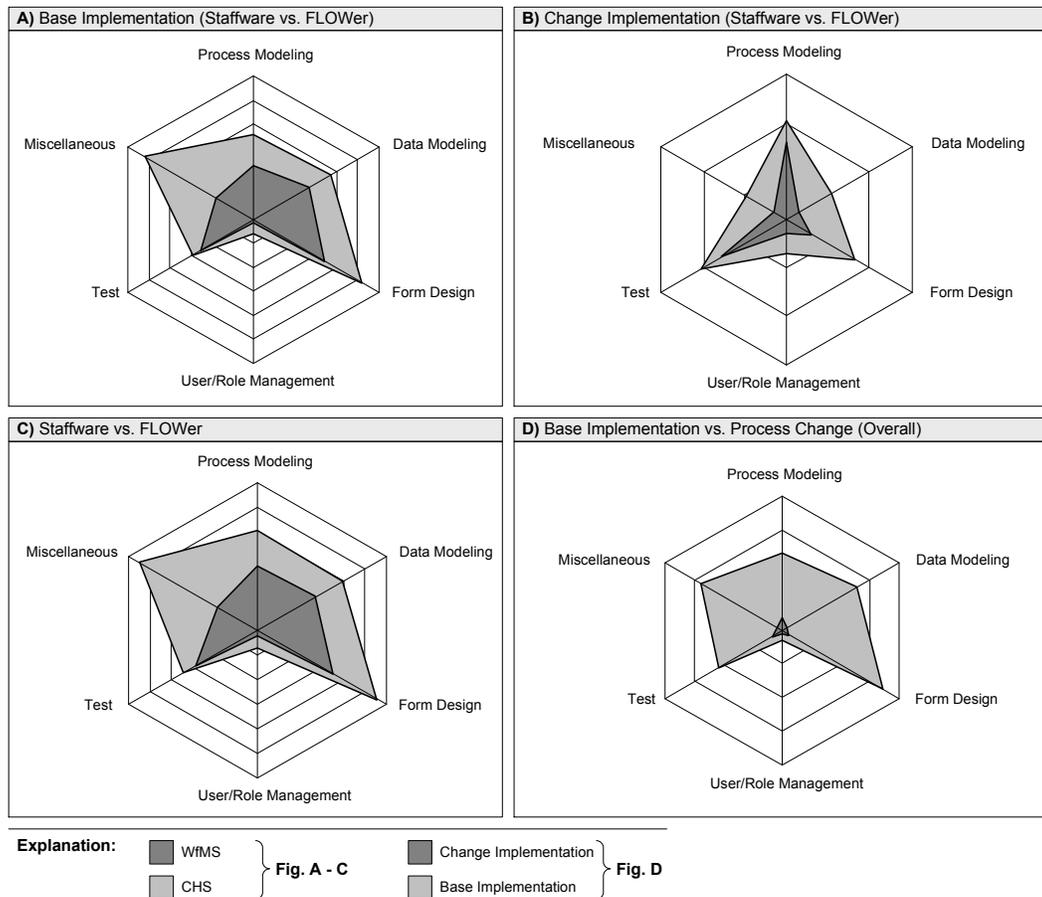


Figure 9.13: Understanding Effort Distribution.

9.5 Discussion

Main Results. Our experiment results indicate that process implementations based on workflow technology generate lower efforts when compared to implementations based on case handling technology. Moreover, our results also show that initial implementations of processes generate significantly higher efforts than subsequent process changes. This is particularly important for policy makers, who often focus on short-term costs (e.g., for purchasing BPM technology and initially implementing business processes) rather than on long-term benefits (e.g., low costs for realizing process changes).

Finally, our data indicates that an increasing knowledge about the processes to be implemented results in increased productivity of software developers. In fact, regardless which BPM technology is used first, all teams reduce their efforts in the second run. Our questionnaire results further indicate that this effect is not necessarily related to an increasing knowledge about the used BPM technologies. This also emphasizes the necessity to involve

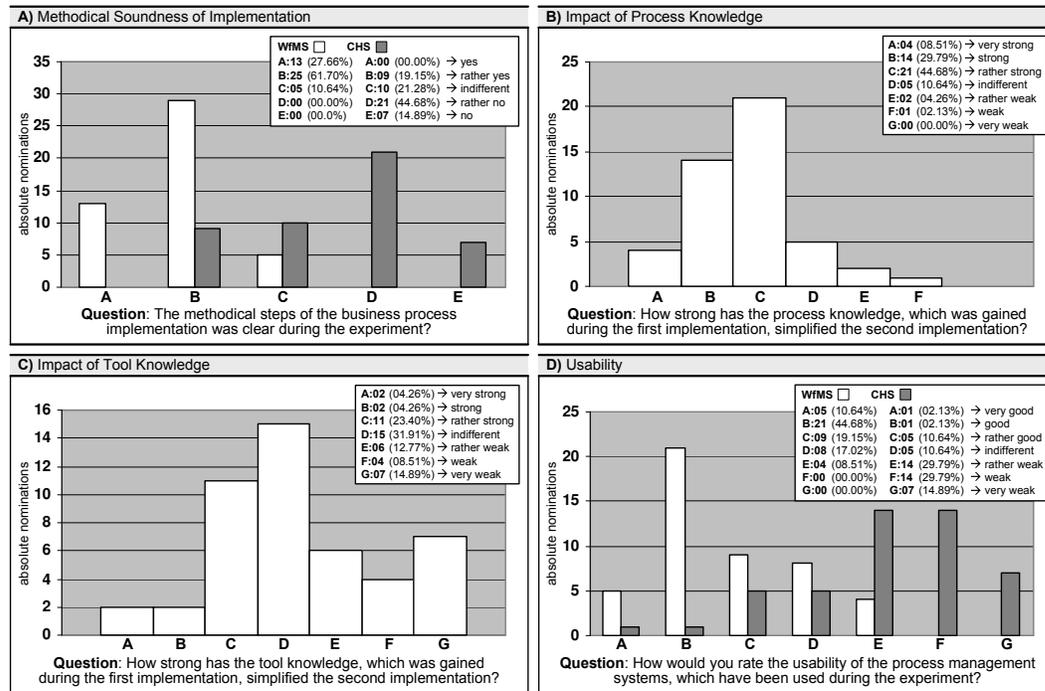


Figure 9.14: Selected Questionnaire Results.

domain experts with high knowledge about the processes to be supported when applying BPM technology and implementing PAIS.

Model Validation. The results of our experiment can be also useful when building Eco-POST evaluation and simulation models. More specifically, results can be utilized to specify the distribution of cost factors as well as for quantifying causal dependencies. As an illustrating example consider the evaluation model shown in Fig. 9.15.

This evaluation model describes cost factors for implementing a business process. More specifically, the model comprises six SCFs ("Planned Process Modeling Costs", "Planned User/Role Management Costs", "Planned Form Design Costs", "Data Modeling Costs", "Test Costs", and "Miscellaneous Costs") which determine the DCF "Process Implementation Costs". These six SCFs are further influenced through various ImFs such as "Process Knowledge", "Technical Maturity of Process Management Platform", and "Quality of Product Documentation".

It is important to mention that this evaluation model is a slightly extended revision of the EP "Process Implementation Costs" which has been already described in Section 6.3.4. Thus, we omit further explanations and focus instead on the utilization of our experiment results to validate this evaluation model.

In particular, our results can be used for two different purposes. *First*, our experiment results enable us to specify the effort distribution of the six SCFs in the evaluation model based on the effort distribution in our experiment (cf. Fig. 9.16). Assume, for example,

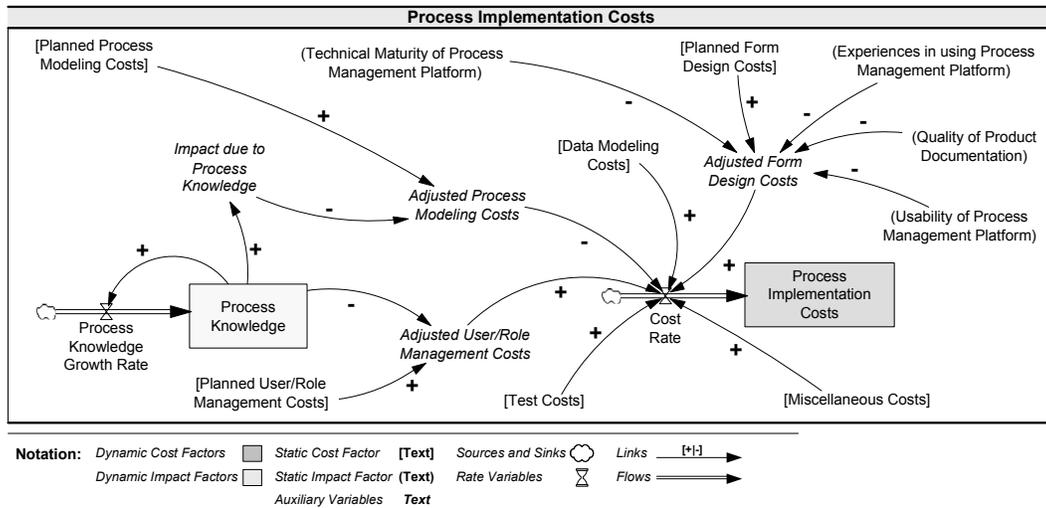


Figure 9.15: Validation Effort Distribution for Process Implementation Costs.

that the effort distribution from Fig. 9.16A shall be used when specifying the underlying simulation model. If we assume that the *planned process implementation costs per time unit* correspond to 1000\$, the effort distribution values for the six SCFs (which determine the overall process implementation costs) can be specified accordingly. Precisely, the constant equations of the simulation model distribution is as follows: (1) "Planned Process Modeling Costs" = 170\$, (2) "Planned User/Role Management Costs" = 10\$, (3) "Planned Form Design Costs" = 270\$, (4) "Data Modeling Costs" = 210\$, (5) "Test Costs" = 200\$, and (6) "Miscellaneous Costs" = 140\$.

Second, experiment results indicate that increasing process knowledge during process implementation can result in effort savings of up to 45%. This information can be useful when building a table function (cf. Section 5.3.4) describing the potentially nonlinear effect of process knowledge on implementation costs. In particular, our experiment data allows to define one extreme value of the table function, namely, the output value of 0.55 (100% - 45%). If process knowledge has its maximum value, process implementation costs will decrease by 45%.

9.6 Related Work

Basic issues on software experiments can be found in [17] and [87]. More specific guidelines on experimental models for validating IT are described by Zelkowitz [226]. Finally, the most similar experiment design when compared to our own is provided by [99, 100] which investigates the impact of workflow technology on software development and software maintenance (see below). Guidelines for reporting on the results of software experiments are given by Jedlitschka [83].

Generally, only few data is available on the effects of workflow technology (regarding case handling, no data is available at all). Oba et al. [146], for example, analyze the intro-

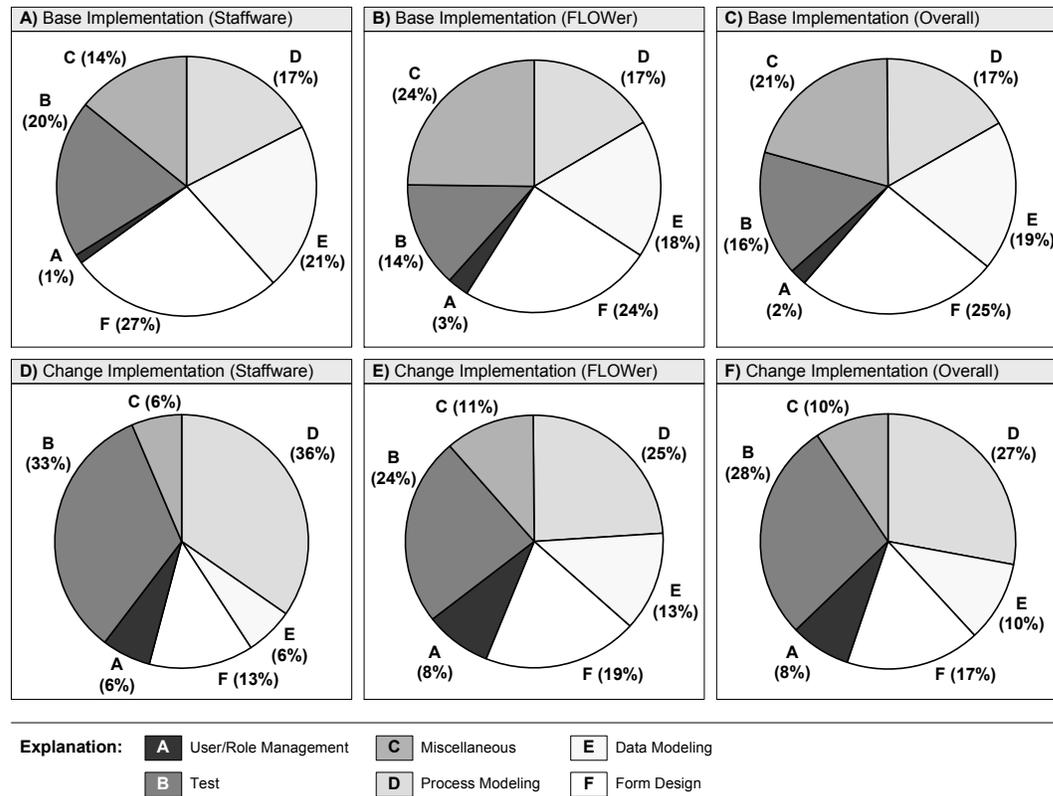


Figure 9.16: Understanding the Distribution of Efforts.

duction of WfMS and particularly focus on the identification of factors that influence work efficiency, processing time, and business process standardization. A mathematical model is provided for predicting the reduction rate of processing times. An extension of this work is [165] where simulation is used to compare pre- and post-implementations of information systems relying on workflow technology. Focus of this work is on analyzing process performance based on criteria such as lead time, waiting time, service time, and utilization of resources. In most cases, the use of workflow technology has resulted in a significant decrease of lead and service time. As a rare example concerning the impact of workflow technology on software development consider the study described in [99]. Results indicate that efforts for realizing process-oriented information systems can be significantly reduced when using workflow technology.

9.7 Summary

This chapter has presented the results of a controlled BPM software experiment with 48 students. Our results indicate that business process implementation based on workflow technology generates lower efforts than using case handling technology. Thereby, initial process implementations result in higher efforts than subsequent process changes. These

data can help enterprises – which crave for quantitative data completing their qualitative decision criteria – to better understand the efforts of using BPM technology. Besides, we have also shown how experiment results can contribute to validate EcoPOST evaluation and simulation models.

More importantly, this chapter illustrates how experimental research can be conducted in the field of BPM. We discuss relevant issues for selecting an experiment design, give a mathematical model of our experiment, and discuss threats for the validity of experimental results (as well as possible mitigations). Therewith, our experiment can be taken as a guideline for applying experimental research to BPM.

Chapter 10

Case Study 1: Process Design and Work Change

10.1 Motivation

The previous chapter has illustrated how experimental research can be used to support the development of evaluation and simulation models. This chapter, in turn, follows another approach to generate respective data, namely the performance of empirical research. In particular, this chapter presents the results of a case study investigating how end users perceive the impact of a newly introduced process design respectively PAIS [139]. Specifically, our goal is to validate the evaluation pattern "Work Profile Change" from Section 6.4.7.

During a PAIS engineering project, business process (re)design is an often performed activity (cf. Section 4.6) [77]. The goal of such an activity is to find an *adequate* process design. "Adequate" means that a business process fulfills its stakeholders' expectations [112]. While expectations of IT managers are carefully considered in most cases (e.g., regarding the efficient execution of a business process), wishes of end users (i.e., of users of the PAIS implementing a new process design) are often neglected.

Consequently, end users will be confronted with significant work profile changes if a new process design is introduced. This can quickly result in end user fears, e.g., due to changing skill variety or decreasing autonomy of end users when process fragments are automated based on workflow technology. End user fears, in turn, can lead to emotional resistance and missing (but highly needed) user support during process (re)design activities, e.g., when conducting interview-based process analysis (cf. Section 4.8.2). Missing user support, in turn, results in increasing business process redesign costs (cf. Fig. 10.1). If end user expectations are taken into account already during business process (re)design¹ it will become possible to avoid unexpected user resistance and to initiate – if necessary – counteractive measures at an early stage, e.g., a better communication policy regarding the goals of the PAIS to be introduced. These causal dependencies are described by the evaluation model shown in Fig. 10.1 (whereas the model part specifying the impact of work

¹Note that end user participation has been considered as "essential" organization-specific impact factor by the majority of survey participants in Section 4.7.2.

profile change corresponds to the evaluation pattern from Section 6.4.7).

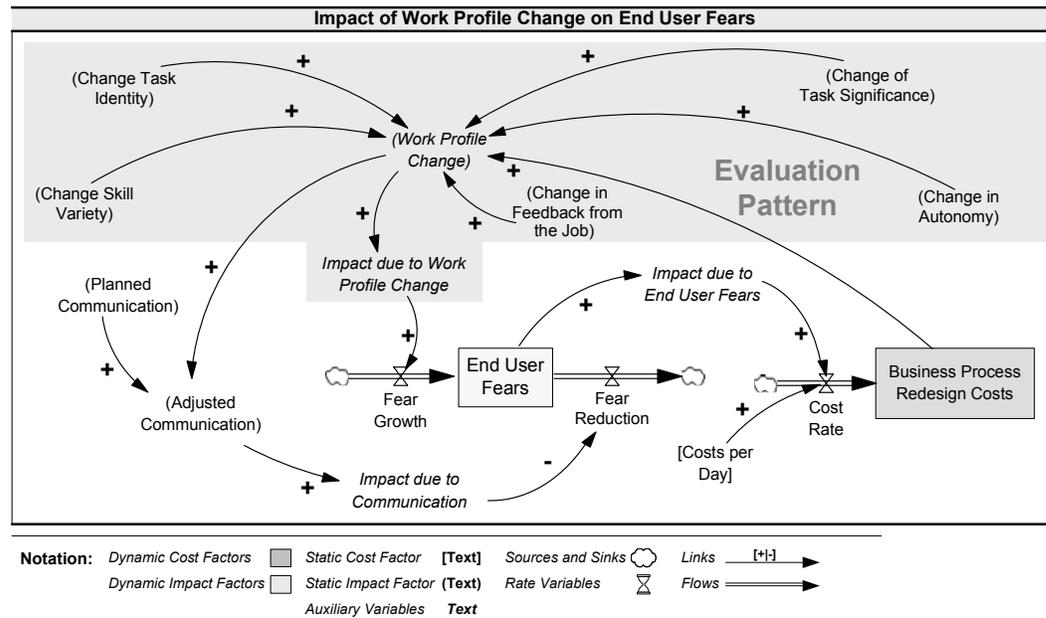


Figure 10.1: Validating the Impact of Work Profile Change.

In our case study, we investigate the potential behavior of those five static ImFs factors (*skill variety*, *task identity*, *task significance*, *autonomy*, and *feedback from the job*) which determine the static ImF "Work Profile Change" (cf. Fig. 10.1).

The remainder of this chapter is organized as follows. Section 10.2 summarizes the design and results of the case study. Section 10.3 discusses in which way the findings of the study can be used to partially validate the evaluation model from Fig. 10.1. Section 10.4 summarizes related work. Section 10.5 concludes with a summary.

10.2 The Case Study

10.2.1 Research Design

Our case study involves two Dutch companies: a facility management organization (*Site 1*) and a housing corporation (*Site 2*). At Site 1, various business processes are supported by the newly introduced PAIS, e.g., financial processes, helpdesk services, and procurement processes. At Site 2, only one business process, namely the company's project management process for building new houses, is supported.

It is the basic goal of this case study to analyze and understand in which way end users perceive the introduction of a PAIS (based on workflow management technology) and the related process (re)design. The major goals and the data collection procedure are explained to survey participants via a newsletter before the case study.

Questionnaire. To collect data, we use a questionnaire and distribute it via a web-based delivery platform. Besides collecting statistical information about the participants, we gather data about the usefulness of the PAIS and data on the impact of the PAIS on job core dimensions in order to specify the work profile change caused by the introduction of PAIS:

- **Questionnaire Part I (General Usefulness):** Addresses the usefulness of the PAIS for the respondents and their organization.
- **Questionnaire Part II (Impact on Job Dimensions):** Addresses the perceived impact of the PAIS (and its underlying process design) on end user work profiles. More specifically, we analyze the perceived impact along the five job dimensions of Hackman's *job characteristics model* [74, 75]:
 - *Job Dimension 1 (Skill Variety):* Deals with the degree to which a job requires different skills for carrying out work.
 - *Job Dimension 2 (Task Identity):* Deals with the degree to which a job requires the treatment of complete tasks, i.e., tasks from beginning to end.
 - *Job Dimension 3 (Task Significance):* Deals with the degree to which a job influences the work of other (internal and external) people.
 - *Job Dimension 4 (Autonomy):* Deals with the degree to which a job provides freedom regarding the scheduling and performance of work.
 - *Job Dimension 5 (Feedback from the Job):* Deals with the degree to which an employee obtains information and feedback about his work performance.

The questions for analyzing these five job dimensions are partially based on the questionnaire used in the *job diagnostic survey* [74, 75] which constitutes one part of the *job characteristics model*. However, we adopt the original questionnaire to our context based on feedback provided by the local supervisors of the case study.

Quantification. In our questionnaire, we use Likert-type scales² for quantifications. Altogether, we have received 25 questionnaires, 10 from Site 1 (equating to a response rate of 71%) and 15 from Site 2 (corresponding to a response rate of 30%). Yet, the number of available questionnaires only allows to identify general trends rather than statistically significant conclusions.

10.2.2 Results

In this section, we discuss results of our case study. *First*, we present data about the general usefulness of the introduced PAIS (cf. Part I of our questionnaire). *Second*, we summarize

²A Likert-type scale consists of a series of declarative statements the survey participant has to make. Survey participants are asked to indicate to which degree he agrees or disagrees with a given statement, e.g., *Do you think skill variety has changed since the introduction of the PAIS?*. Commonly, five options are provided: "strongly agree", "agree", "undecided", "disagree", and "strongly disagree". The individual responses are assigned with numbers in order to quantify the various items. It is thereby necessary that the assigned numbers are consistent with the meaning of the response.

results regarding the perceived impact of the PAIS and its underlying process design on the end users' work profile (cf. Part II of our questionnaire).

Part I - General Usefulness. Fig. 10.2 shows the *mean* and *standard deviation* for data collected at Site 1. We do not only calculate these statistical metrics for the total number of responses, but also for different user groups (technical staff, management, etc.).

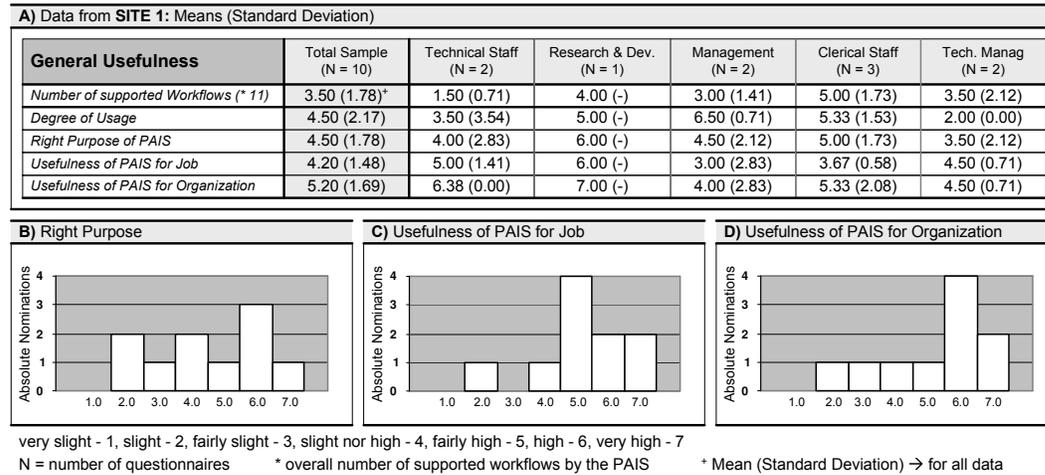


Figure 10.2: General Usefulness (Site 1).

Fig. 10.2A shows the survey results regarding the general usefulness of the PAIS as perceived by respondents. We analyze "general usefulness" based on four questions: (1) *To what degree do you use the PAIS within your organization?* (2) *To what degree does the PAIS contribute to your tasks in a positive way?* (3) *To what degree is the PAIS used for the "right" purpose?* (4) *To what degree does the PAIS contribute to your organization?* For quantifying these four questions, a Likert-type scale with the following predefined answers: *very slight*: 1, *slight*: 2, *fairly slight*: 3, *neither nor*: 4, *fairly high*: 5, *high*: 6, *very high*: 7. Fig. 10.2B, Fig. 10.2C, and Fig. 10.2D show selected results about the usefulness of the PAIS for the job and organization.

Fig. 10.3A shows the data collected at Site 2. Obviously, results are more divergent when compared to the ones from Site 1 (though the general usefulness of the PAIS for the own organization is recognized as well).

Part II - Impact on Job Dimensions. Fig. 10.4A shows the the mean and the standard deviation for the data we collected at Site 1 on the perceived impact of the PAIS on the considered job dimensions (cf. Section 10.2.1). Thereby, we use another Likert-type scale than the one used in Part I. Specifically, participants indicate how each analyzed job dimension has changed since the introduction of the PAIS: *lot less* - 1, *less* - 2, *little less* - 3, *neither less nor more* - 4, *little more* - 5, *more* - 6, *lot more* - 7.

Fig. 10.4B denotes for each job dimension the percentage of respondents perceiving a "decrease", "no change" or "increase" in the respective job dimensions. Fig. 10.4B also

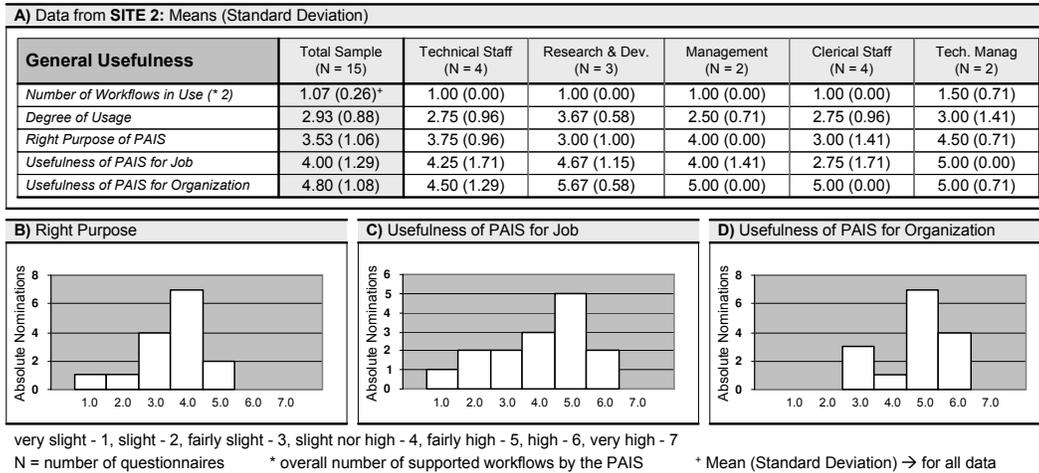


Figure 10.3: General Usefulness (Site 2).

shows the range of answers corresponding to the used Likert-type scale. In every job dimension (except for the "autonomy" dimension), most participants denote an increase, i.e., an increase of skill variety, task identify, task significance, and feedback from the job.

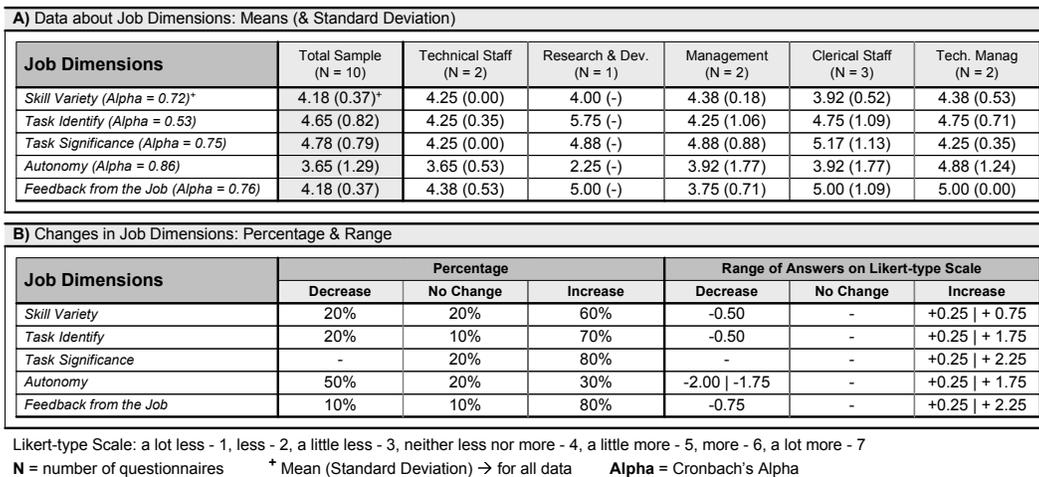


Figure 10.4: Impact on Job Dimensions (Site 1).

Finally, Fig. 10.5 shows the data collected at Site 2 regarding the perceived impact of the PAIS on the considered job dimensions. When compared to Site 1 results are different. For example, in all job dimensions most respondents neither recognize a significant increase nor a significant decrease.

A) Data about Job Dimensions: Means (& Standard Deviation)

Job Dimensions	Total Sample (N = 15)	Technical Staff (N = 4)	Research & Dev. (N = 3)	Management (N = 2)	Clerical Staff (N = 4)	Tech. Manag (N = 2)
<i>Skill Variety</i> (Alpha = 0.29)	4.07 (0.48)*	4.06 (0.92)	4.25 (0.25)	4.00 (0.00)	3.94 (0.13)	4.12 (0.53)
<i>Task Identify</i> (Alpha = 0.72)	4.00 (0.57)	3.63 (0.92)	4.17 (0.52)	4.00 (0.35)	4.19 (0.38)	4.13 (0.18)
<i>Task Significance</i> (Alpha = 0.73)	4.32 (0.39)	4.19 (0.38)	4.17 (0.29)	5.00 (0.00)	4.31 (0.38)	4.13 (0.18)
<i>Autonomy</i> (Alpha = 0.79)	4.00 (0.53)	3.63 (0.75)	4.00 (0.00)	4.75 (0.35)	4.00 (0.41)	4.00 (0.00)
<i>Feedback from the Job</i> (Alpha = 0.77)	4.25 (0.37)	4.06 (0.13)	4.42 (0.52)	4.50 (0.71)	4.13 (0.14)	4.38 (0.53)

B) Changes in Job Characteristics: Percentage & Range

Job Dimensions	Percentage			Range of Answers on Likert-type Scale		
	Decrease	No Change	Increase	Decrease	No Change	Increase
<i>Skill Variety</i>	20%	53.3%	26.7%	-1.00 -0.75	-	+0.25 + 1.25
<i>Task Identify</i>	20%	46.7%	33.3%	-1.75 -0.25	-	+0.25 + 0.75
<i>Task Significance</i>	-	53.3%	46.7%	-	-	+0.25 + 1.00
<i>Autonomy</i>	13.3%	66.7%	20%	-1.50 -0.50	-	+0.50 + 1.00
<i>Feedback from the Job</i>	-	53%	46.7%	-	-	+0.25 + 1.00

Likert-type Scale: a lot less - 1, less - 2, a little less - 3, neither less nor more - 4, a little more - 5, more - 6, a lot more - 7
 N = number of questionnaires * Mean (Standard Deviation) → for all data Alpha = Cronbach's Alpha

Figure 10.5: Impact on Job Dimensions (Site 2).

10.2.3 Methodical Soundness

We discuss the methodological soundness of our case study along four criteria: (1) *data reliability*, (2) *construct validity*, (3) *internal validity*, and (4) *external validity*:

- **Data Reliability:** In order to claim a satisfactory degree of data reliability, Cronbach's *alpha* should be bigger than 0.70 [145]. Fig. 10.4A shows that all alpha values are above 0.70 except for the alpha value of the task identity dimension. In Fig. 10.5A, all alpha values except for the skill variety dimension are bigger than 0.70. Hence, we consider our data as reliable.
- **Construct Validity:** Construct validity refers to the degree to which inferences can be made from an empirical study considering the theoretical constructs underlying the empirical study. Note that by relying on an existing, well-elaborated measurement instrument, i.e., the *job diagnostic survey* questionnaire from the *job characteristics model* [74, 75], we can conclude that construct validity is given. Notwithstanding, there exist threats for construct validity, e.g., the personal (subjective) interpretation of the *job diagnostic survey* questionnaire or the translation of the questionnaire into the Dutch language.
- **Internal Validity:** Internal validity deals with the question whether the independent variable (i.e., the PAIS design) will cause change(s) of the dependent variable (i.e., the perceived impact of the PAIS on the job dimensions). Generally, we can observe that the data collected at Site 1 has a higher degree of internal validity. For this, three reasons exist: (1) the overall number of supported workflows is higher (i.e., the users have more experience using the PAIS), (2) more work is done based on the PAIS (which is a consequence of the first reason), and (3) the employees at Site 1 have worked with the PAIS for a longer time.

- **External Validity:** Given the small number of 25 received questionnaires, we must take into account that single responses have a relatively large impact on our findings. Our results only allow to draw first conclusions on basic trends rather than statistically significant conclusions. We still believe that our results provide valuable insights into effects caused by operationalizing process design.

Considering these criteria, we conclude that results of our case study can be considered as valid. Notwithstanding, it will be necessary to apply our questionnaire to a larger sample in order to derive substantial results.

10.3 Discussion

This section summarizes the main trends that can be derived from the collected data and discusses them towards their suitability for validating the evaluation model from Fig. 10.1. Thereby, we focus on the impact of the PAIS on the analyzed job dimensions (and neglect the self-explanatory results on the usefulness of the PAIS):

- **Skill Variety:** In his study on the impact of workflow systems on organizations [102], Kueng states that "*through the use of a workflow system, jobs become more structured and more routine*". Taking this statement one may conclude that survey respondents perceive a significant decrease of skill variety. However, this is not the case. Our results show (cf. Fig. 10.4 and Fig. 10.5) that the majority of participants perceive no decrease in skill variety (some even perceive a slight increase).
- **Task Identity:** Typically, PAIS provide to the user only that data needed for the execution of an activity (in literature, this is known as "context tunneling" [202]). Hence, one may expect a decrease in the task identity dimension, e.g., due to the feeling of being less involved in a job as end users have no overview of the entire "work case" anymore. However, at both sites, our data do not show significant decrease in task identity (both for the total samples and for each user group).
- **Task Significance:** One might also conclude that working with a PAIS may lead to an increasing task significance. End users might perceive their job as being more important and become aware of the inherent interdependence of their work. Generally, our data confirm this and shows a slightly increasing task significance.
- **Autonomy:** Kueng [102] states that "*individuals are forced to stay within given limits. Since a larger proportion of work is programmed, it becomes harder to exercise and integrate creativity and ingenuity*". Some of our data support this conclusion. Generally, our data only show a minor decrease of perceived autonomy at Site 1 (cf. Fig. 10.4). Only the "Research & Development" and the "Management" user group perceive a stronger drop. This is, however, remarkable as these two groups use the PAIS more intensively when compared to other ones. At Site 2, by contrast, our data show no significant change in the autonomy dimension (cf. Fig. 10.5). Since the PAIS generally adopts a less important role at Site 2 (by means of a smaller number

of supported workflows), this allows for the conclusion that autonomy decreases with an increasing amount of supported workflows.

- **Feedback from the Job:** At both sites, our data show that working with PAIS gives end users more direct feedback on their performance. Only the "Management" user group at Site 1 negates this (cf. Fig. 10.4).

Generally, we conclude that the impact of a PAIS on work profile change is not as strong as we expected it to be. This result, however, is useful for building the table function specifying the auxiliary variable "Impact due to Work Profile Change" (cf. Fig. 10.1). In particular, we can conclude that the shape of the table function is rather flat as the impact of work profile change needs to be considered as low.

Regarding the behavior of the five considered job dimensions (which are also represented in the evaluation model from Fig. 10.1), we conclude that changes in task significance and autonomy are generally perceived as stronger than changes in skill variety, task identify, and feedback from the job. This means that when building the equation for "Work Profile Change" in a simulation model, task significance and autonomy need to be higher weighted than the three other job core dimensions.

10.4 Related Work

There exist only few studies which address organizational changes enabled by PAIS, mainly in the field of workflow technology. Sarmento and Machado [172], for example, propose a framework to investigate the impact of a WfMS on an organization. They identify relevant contingency factors and describe organizational domains and sub-domains which are affected by workflow technology. In [173], they additionally describe changes enabled by WfMS as well as approaches to evaluate such changes. Kueng [102] states that "*through the use of a workflow system, jobs become more structured and more routine*". Hence, workflow design as well as the introduction of workflow technology should be extended towards job design and organizational design.

Recently, Reijers and Poelmans [164] presented a study on workflow implementations that showed that workflow systems are widely applied in modern service industry and are considered as useful by performers to support their way of working.

10.5 Summary

This chapter has summarized the results of a case study at two Dutch companies in which we investigated how end users perceive the impact of a newly introduced process design and its implementation based on a PAIS. More specifically, we have investigated changes in the five job dimensions of Hackman's *job characteristics model*: *skill variety*, *task identity*, *task significance*, *autonomy*, and *feedback from the job*.

Results show that working with a PAIS does not necessarily entail a significant decrease in skill variety. Moreover, end users intensively working with a PAIS perceive a higher

degree of task identity and task significance, i.e., end users experience their work as more meaningful. However, previous statements on the negative impact of PAIS on the autonomy of end users are partly confirmed. Finally, results show that working with a PAIS gives end users more direct feedback on their overall work performance.

Chapter 11

Case Study 2: Using EcoPOST in Practice

11.1 Motivation

In the previous two chapters we have shown how experimental and empirical research contributes to the derivation and validation of EcoPOST evaluation and EcoPOST simulation models. This chapter concludes Part III of this thesis and presents another case study, this time focusing on the overall applicability of the EcoPOST framework.

We apply the EcoPOST framework to a complex PAIS engineering project from the automotive domain. We investigate cost overruns observed during the introduction of a PAIS, which shall support the development of *electrical and electronic (E/E) systems*. Based on real project data, interviews with project members (e.g., requirements engineers, software architects, software developers), two online surveys among the end users of the PAIS, and practical experiences gathered in the analyzed PAIS engineering project, we develop a set of EcoPOST evaluation models and analyze these models using simulation.

The remainder of this chapter is organized as follows. Section 11.2 gives background information. Section 11.3 summarizes the research design, developed evaluation models, and simulation results related to our case study. Section 11.4 discusses these results and Section 11.5 concludes with a summary and a short discussion of lessons learned.

11.2 Background Information

For several years the automotive industry is confronted with technological changes caused by the introduction of complex *electrical and electronic (E/E) systems* [122]. The E/E system of a car consist of *electrical control units (ECUs)*, i.e., embedded systems comprising both hardware and software components. In modern cars, we can find up to 70 ECUs, most of them interconnected by bus systems, realizing complex features like safety, driving assistance, and telematics [22, 101].

To comply with requirements of product liability regulations, laws, and industrial standards (e.g., CMMI, IEC 61508, etc.) during E/E system development, integrated IT support

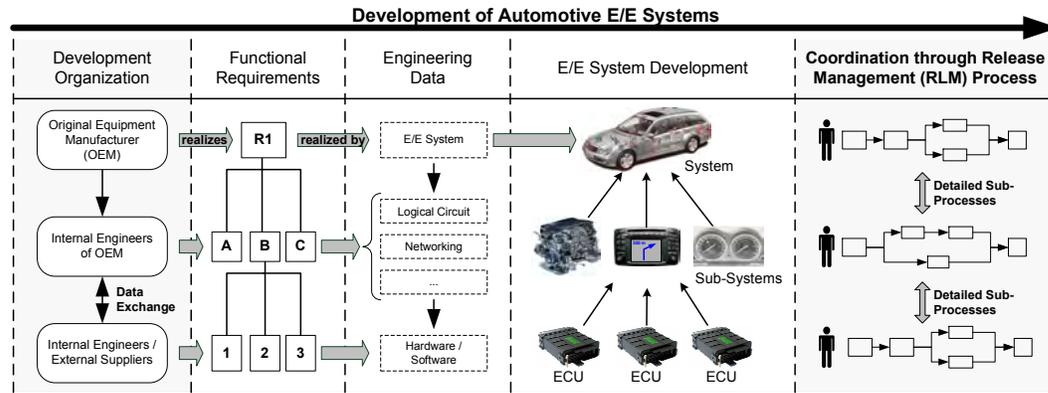


Figure 11.1: Automotive E/E System Development [122].

for the entire E/E system development process becomes necessary. At best, integrated IT support means that a central *product data management* (PDM) backbone is provided which enables engineers (working for the automotive original equipment manufacturer) to effectively control the exchange of relevant product data with both internal colleagues and external suppliers (cf. Fig. 11.1). On the process level, this exchange is typically specified by a *release management* (RLM) process. The RLM process describes the systematical configuration, test, and release of hardware and software components (or complete ECUs) at a specific point in time; e.g., when certain quality gates or milestones are reached.

Current E/E system development and RLM implementation are often based on a variety of heterogeneous IS, email communication, phone calls, and the exchange of complex excel sheets. This results in problems such as varying process executions, unclear process responsibilities, and a difficult synchronization of concurrent development activities.

In the considered project, a new PAIS (and PDM system respectively) is introduced to overcome these problems. The new PAIS system shall provide integrated IT support for E/E system development, deploy development documentation to all involved users (e.g., engineers, suppliers), and synchronize concurrent access to these data.

11.3 The Case Study

11.3.1 Research Design

According to the EcoPOST methodology, the first step in our case study corresponds to a closer look at the evaluation scenario to be investigated (cf. Section 4.5).

Step 1: Understanding the Evaluation Scenario. An initial business case for the considered PAIS engineering project is developed prior to project start in order to convince senior management to fund the project. This business case¹ is based on data about similar projects provided by competitors (evaluation by analogy) as well as on rough estimates on

¹Note that we did not participate in the development of this initial business case.

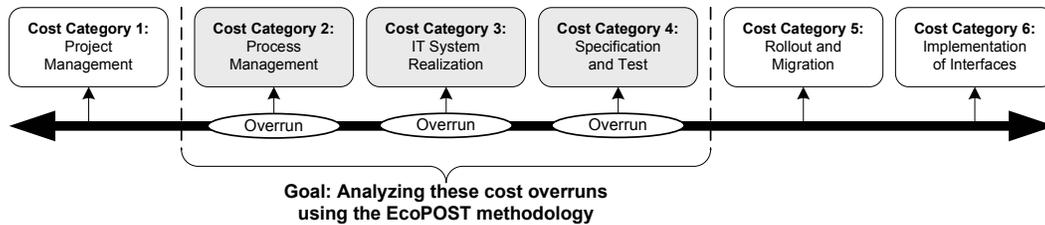


Figure 11.2: The Business Case.

planned costs and assumed benefits of the project. It comprises six main cost categories (cf. Fig. 11.2): (1) *project management*, (2) *process management*, (3) *IT system realization*, (4) *specification and test*, (5) *roll-out and migration*, and (6) *implementation of interfaces*.

At a first project review (i.e., measurement of results), it has turned out that the originally planned project costs are not realistic, i.e., cost overruns were observed – particularly concerning cost categories 2 and 3. In our case study, we analyze these cost overruns based on the evaluation concepts provided by the EcoPOST framework. In particular, we analyze three of the six cost categories in detail (cf. Fig. 11.2):

- **Process Management Costs:** This cost category deals with costs related to the (re-)design of the E/E system development processes to be supported. This includes both the definition of new and the redesign of existing processes. As an example of a new business process to be designed consider an E/E data provision process to provide external suppliers with needed product data. As an example of an existing process which is redesigned consider the basic E/E release management process. Among other things, process management costs include costs for performing interview-based process analysis and costs for developing process models.
- **IT System Realization Costs:** This cost category deals with costs for implementing the new PDM system on top of a process management system. In our case study, we focus on the analysis of costs related to the use of the process management system. This includes costs for specifying and implementing the business functions and workflows to be supported as well as costs for identifying potential user roles and implementing respective access control mechanisms.
- **Specification and Test Costs:** This cost category sums up costs for specifying the functionality of the PDM system as well as costs for testing the coverage of requirements. This includes costs for eliciting and documenting requirements as well as costs for performing tests whether requirements are met by the PAIS.

To be able to build evaluation and simulation models, i.e., to investigate causal dependencies and resulting effects in the three analyzed cost categories, we need to collect data.

Step 2 to Step 4: Data Collection. Steps 2 to 4 of the EcoPOST methodology deal with the identification of SCFs (Step 2), DCFs (Step 3), and ImFs (Step 4). The collection of respective data is based on four pillars (cf. Figure 11.3):

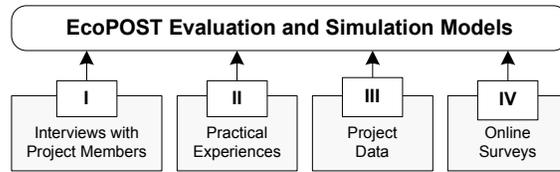


Figure 11.3: Data Collection (Information Sources).

- **I. Project Data:** A first data source is available project data², in particular, estimates about planned costs from the initial business case.
- **II. Qualitative Data based on Interviews:** We interview 10 project members (2 software architects, 4 software developers, 2 usability engineers, and 2 consultants related to the project). Our interviews are based on a predefined, semi-structured protocol. Each interview lasts about 1 hour and is accomplished on a one-to-one basis. Goal of the interviews is to collect data about causal dependencies between cost and impact factors in each analyzed cost category (respectively the underlying activities).
- **III. Online Surveys among End Users:** We conduct two online surveys among two user groups of the new PDM system (which will have more than 4000 users in the end). The questionnaires are distributed via a web-based delivery platform. They slightly vary in order to cope with the different work profiles of both user groups. Most questions are multiple-choice questions, i.e., they provide predefined sets of possible answers. Goal of the two online surveys is to confirm the significance of selected ImFs like "End User Fears" and "Emotional Resistance of End Users". 70 users from the user group of "*component responsables*" (engineers who are in charge for the development of E/E components) participate in the survey. In the second survey, 10 users from the user group "*main data customers*" (engineers who utilize data on E/E components to develop other car components) answer our questions.
- **IV. Practical Experiences:** Finally, our evaluation and simulation models also build upon practical experiences we gathered when participating in the investigated PAIS engineering project. We have worked in this project as requirement engineers for more than one year and have gained deep insights during this time. Besides the conducted interviews, these experiences are the major source of information when designing our evaluation models.

These four information sources allow us to identify relevant cost and impact factors, i.e., evaluation factors that need to be included in the evaluation models to be developed. Likewise, the information sources also enable us to spot important causal dependencies between cost and impact factors and to derive evaluation and simulation models.

Based on practical experiences and interviews, for instance, we have identified a set of relevant ImFs for each of the analyzed cost categories (cf. Fig. 11.4) – out of the ImF

²As this project data is confidential, it has to be made anonymous both in the simulation models and the simulation outcomes.

baseline we have given in Section 4.8. Obviously, there exist ImFs which are important in all three analyzed cost categories (e.g., domain knowledge, knowledge about existing processes/process knowledge, and fears and emotional resistance of end users).

Besides, Fig. 11.4 also shows which ImFs have been identified based on which of the information sources from Fig. 11.3. Note that ImFs we consider as important based on our practical experiences are also confirmed by our interviews. Results from the two online surveys confirm additional ImFs, for example, the role of fear and emotional resistance of end users. However, many ImFs are not that visible from an end user perspective. Thus, only few ImFs can be derived from our survey results. Existing project data cannot contribute to the identification of ImFs; available data only deal with costs.

		Organization-specific ImFs										Project-specific ImFs										Technology-specific ImFs																	
		Experiences in using Proc. Mgmt. Technology	Ability to redesign Business Processes	Reorganization of Information	Domain Knowledge	End User Participation	Access to required Information	Organizational Process Maturity	Availability of Process Documentation	Mature Technology Infrastructure	Ability to adapt IT Governance	Overview of existing Processes	Knowledge about existing Processes	Information about existing Processes	Evolutionary Process Design	Revolutionary Process Design	Degree of Job Redesign	End User Fears	Communication with End Users	Management Commitment	Use of Process Modeling Tools	Adequate Planning	Access to required Skills	Motivation for the Project	Technical Maturity of Proc. Mgmt. Platform	Experiences using the Proc. Mgmt. Platform	Available Support for the Proc. Mgmt. Platform	Supported Degree of Process Flexibility	Availability of Developing Tools	Support of Standards and Norms	Low Licence Costs	Good Documentation	Regular Product Updates	Model-driven Application Development	Usability of the Proc. Mgmt. Platform	Powerful Application Programming Interfaces			
View-points	Process Management Viewpoint	x	x	x							x					x	x	x	x																				
	IT System Realization Viewpoint				x						x							x							x	x							x				x		
	Specification and Test Viewpoint			x							x						x																						
Sources	Interviews with Project Members	x	x	x						x						x	x	x	x					x	x							x				x			
	Practical Experiences	x	x	x						x						x	x	x	x					x	x							x				x			
	Project Data	Project data is only available on the estimated costs of the six considered cost categories - not for ImFs!																																					
	Survey among User Groups			x												x	x																				x		

Figure 11.4: Identification of relevant Impact Factors.

Step 5 and Step 6: Evaluation Models and Simulation. Based on the collected data each of the following sections presents one evaluation model for each of the analyzed cost categories (Step 5 of the EcoPOST methodology). We also present results of simulation runs based on which we investigate the dynamic behavior of the three evaluation models (Step 6 of the EcoPOST methodology).

11.3.2 Analyzing Process Management Costs

Fig. 11.5 shows the derived evaluation model for analyzing process management costs. Basically, this evaluation model builds upon several *evaluation patterns* (EP), namely, the primary EP "Business Process Redesign Costs" (cf. Section 6.3.1) and the four secondary EPs "End User Fears" (cf. Section 6.4.1), "Process Knowledge" (cf. Section 6.4.2), "Domain Knowledge" (cf. Section 6.4.3), and "Work Profile Change" (cf. Section 6.4.7).

These EPs have been selected based on results of our interviews and practical experi-

ences we gathered in the analyzed project.

Note that the simulation model for this evaluation model – like the two other simulation models in the following sections as well – are not depicted in this chapter, but can be found in Appendix C. Reason is that the simulation models comprise a high number of equations whose full presentation would unnecessarily disrupt the entire chapter.

Step 5: Evaluation Model. In the following we describe how we build the evaluation model shown in Fig. 11.5:

- **Step 5-a (Planned Costs):** Based on interview results and practical experiences we conclude that business process redesign costs are primarily determined by two cost factors (cf. Area I in Fig. 11.5), i.e., costs for performing process analysis and costs for process modeling. Estimated process analysis costs are represented as SCF "Planned Costs for Process Analysis". Estimated process modeling costs are represented as SCF "Planned Costs for Process Modeling". Both SCFs are quantified in the simulation model based on data from the initial business case.
- **Step 5-b (Process Analysis Costs and related Impact Factors):** As a result of our interviews we merge the SCF "Planned Costs for Process Analysis" with the two ImFs "End User Fears" and "Process Complexity" in the auxiliary variable "Adjusted Costs for Process Analysis". While the former ImF is a dynamic one (which is itself determined by other ImFs), the latter ImF is a static one.
- **Step 5-c (Process Modeling Costs and related Impact Factors):** Likewise, we merge the SCF "Planned Costs for Process Modeling" with the static ImF "Process Complexity" in the auxiliary variable "Adjusted Costs for Process Modeling". Our experiences indicate that the high complexity of the business processes to be supported is one major factor leading to increased process modeling costs.
- **Step 5-d (Ability to Redesign Business Processes):** Interview results and practical experiences also result in the incorporation of an additional dynamic ImF "Ability to (re)design Business Processes" when considering business process (re)design costs (cf. Area II in Fig. 11.5). Reason is that efficient process (re)design depends on the overall ability to perform such activities. Consequently, the dynamic ImF "Ability to (re)design Business Processes" is connected to the rate variable "Cost Growth" with a negative link.
- **Step 5-e (Additional Impact Factors):** The ability to (re)design business processes, in turn, is determined by two dynamic ImFs "Domain Knowledge" and "Process Knowledge", by the static ImF "Management Commitment", and by the auxiliary variable "Emotional Resistance of End Users". Process and domain knowledge as well as management commitment can increase the ability to (re)design business processes (and are therefore assigned to the rate variable controlling the inflow of the dynamic ImF). Emotional resistance of end users, in turn, can decrease the ability to (re)design

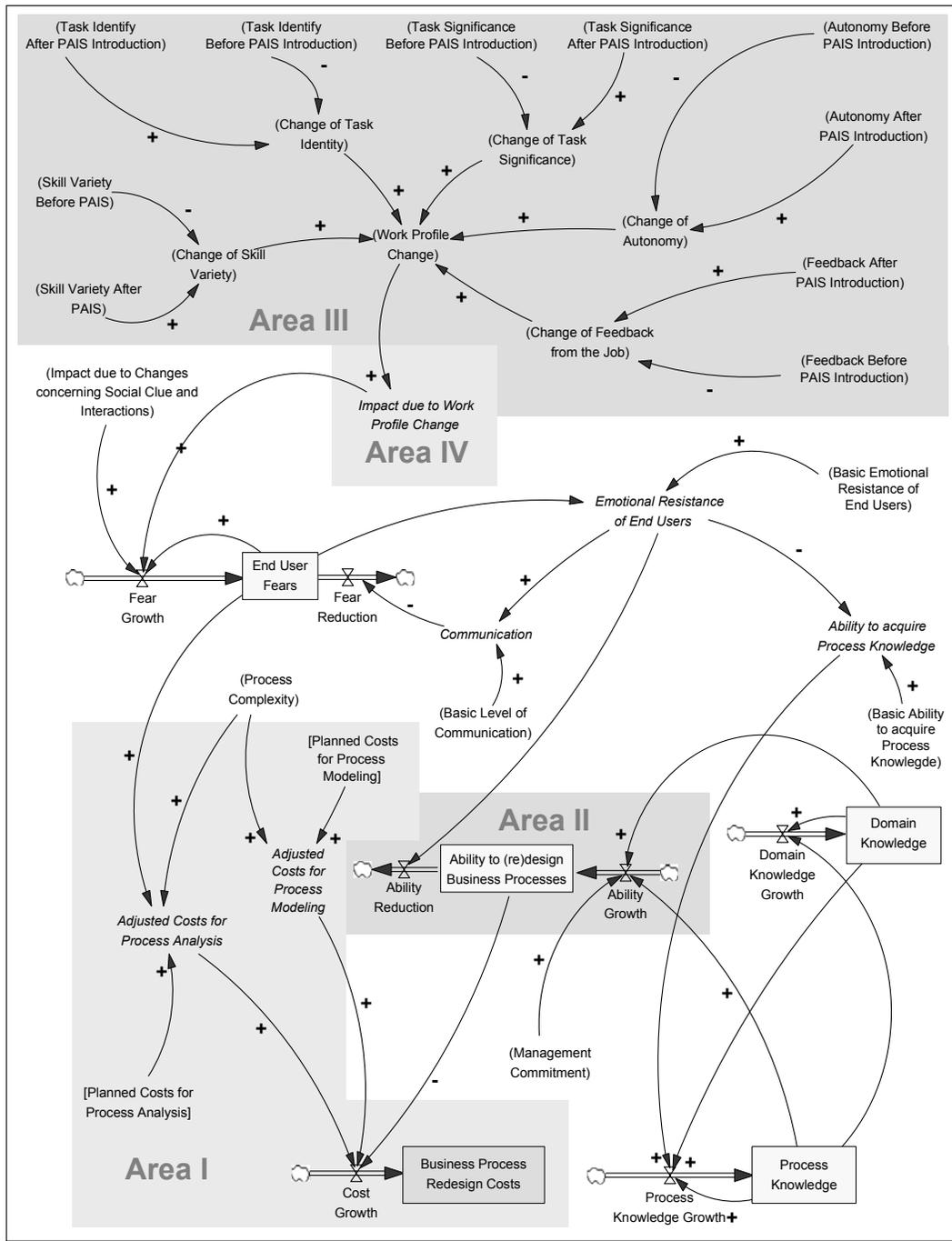


Figure 11.5: Process Management Costs (Evaluation Model).

business processes (and is therefore assigned to the rate variable controlling the out-flow). The relevance of these four ImFs is confirmed by our interview results. The relevance "Emotional Resistance of End User" is additionally confirmed by results of the conducted online surveys.

- **Step 5-f (End User Fears):** The auxiliary variable "Emotional Resistance of End User" is derived from the basic level of user resistance (represented by the static ImF "Basic Emotional Resistance of End Users") and the dynamic ImF "End User Fears". The latter dynamic ImF, in turn, is determined – according to our practical experiences and survey results – by the two static ImFs "Impact due to Changes concerning Social Clue and Interactions" and "Work Profile Change". The model part dealing with work profile change corresponds to the EP from Section 6.4.7 (cf. Area III in Fig. 11.5).
- **Step 5-g (Impact due to Work Profile Change):** As we assume a nonlinear dependency between the ImF "Work Profile Change" and the ImF "End User Fears", we introduce an additional auxiliary variable "Impact due to Work Profile Change" (cf. Area IV in Fig. 11.5). This auxiliary variable is specified based on a table function (cf. Section 5.3.4) which, in turn, is defined based on results from the case study which has been described in Chapter 10.

Step 6: Simulation. Fig. 11.6 shows simulation results for this evaluation model. Fig. 11.6A illustrates the evolution of end user fears along the simulated time domain of 100 days. Obviously, the dynamic ImF "End User Fears" is continuously increasing. Likewise, emotional resistance of end users is increasing (cf. Fig. 11.6B). This result is particularly valuable for the managers of the considered project who often tend to underestimate the impact of end user participation on the costs of PAIS engineering projects.

Fig. 11.6C shows that the ability to redesign business processes is decreasing during business process redesign. This is caused by the increasing emotional resistance of end users making process analysis a difficult task to accomplish. Positive effects of increasing domain knowledge (cf. Fig. 11.6D), increasing process knowledge (cf. Fig. 11.6E), and a strong management commitment cannot compensate the negative effects caused by the increasing emotional resistance of end users.

Altogether, Fig. 11.6E shows that costs for business process redesign exceed originally planned costs. When analyzing one of the roots of end user fears, i.e., work profile change, it can be seen that a higher degree of work profile change results in higher end user fears, and, in turn, in higher (re)design costs (cf. Fig. 11.6E).

11.3.3 Analyzing IT System Realization Costs

Fig. 11.7 shows the derived evaluation model for analyzing IT system realization costs. This evaluation model is based on the primary EP "Process Implementation Costs" (cf. Section 6.3.4) and the two secondary EPs "Process Knowledge" (cf. Section 6.4.2) and "Domain Knowledge" (cf. Section 6.4.3). Again, we have selected these EPs based on interview

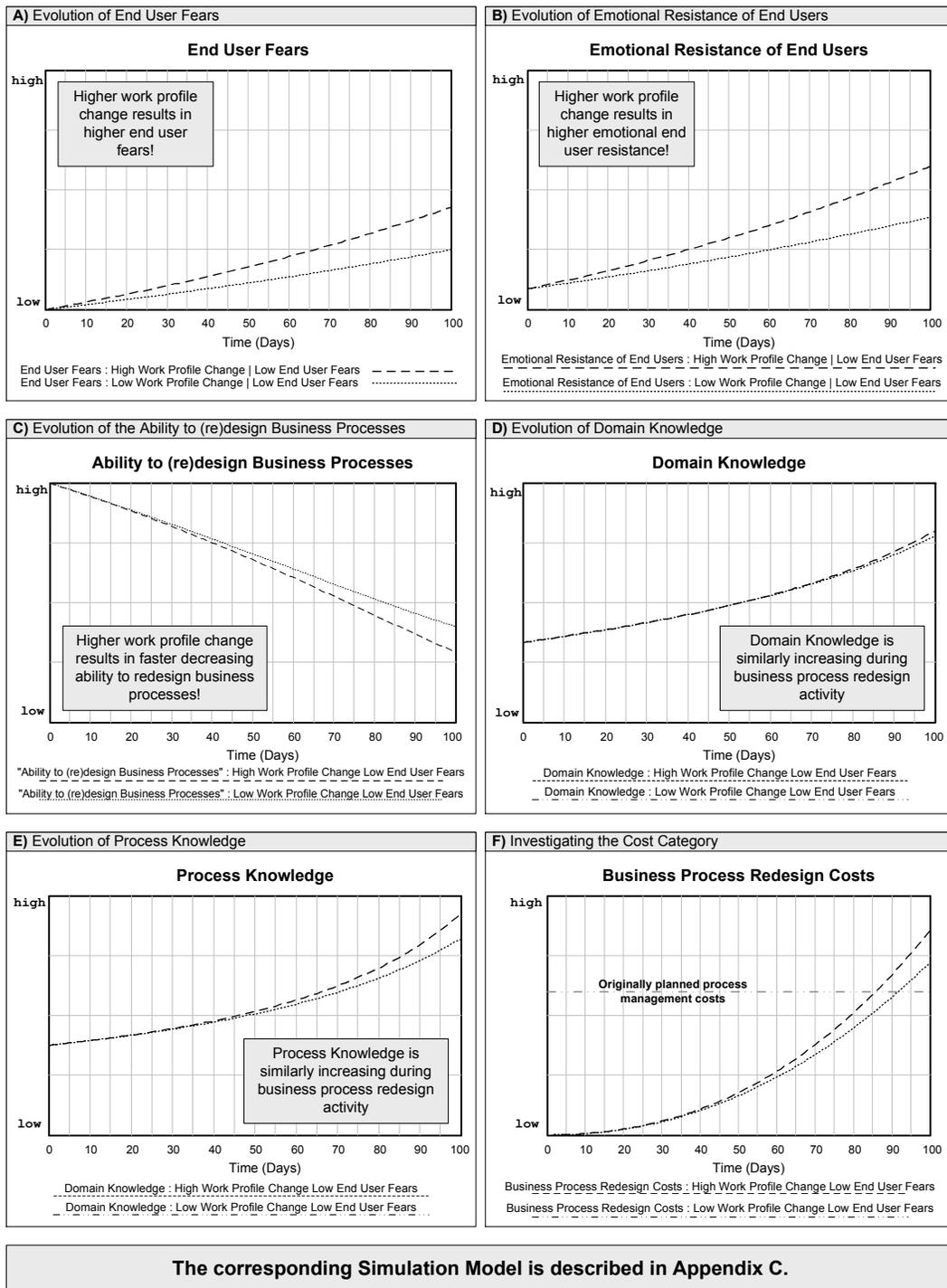
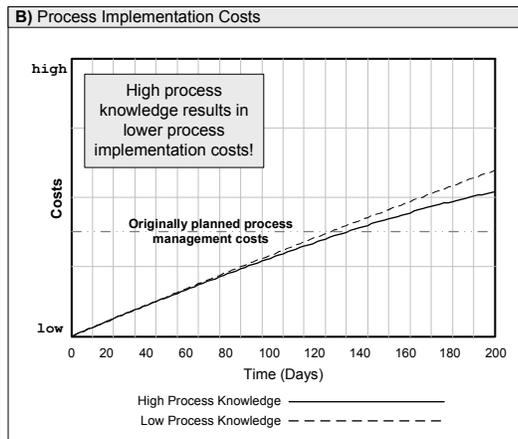
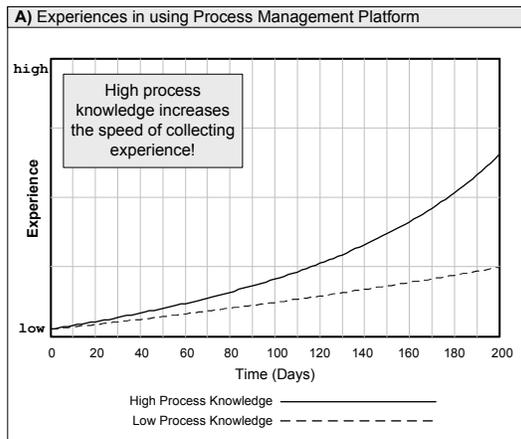
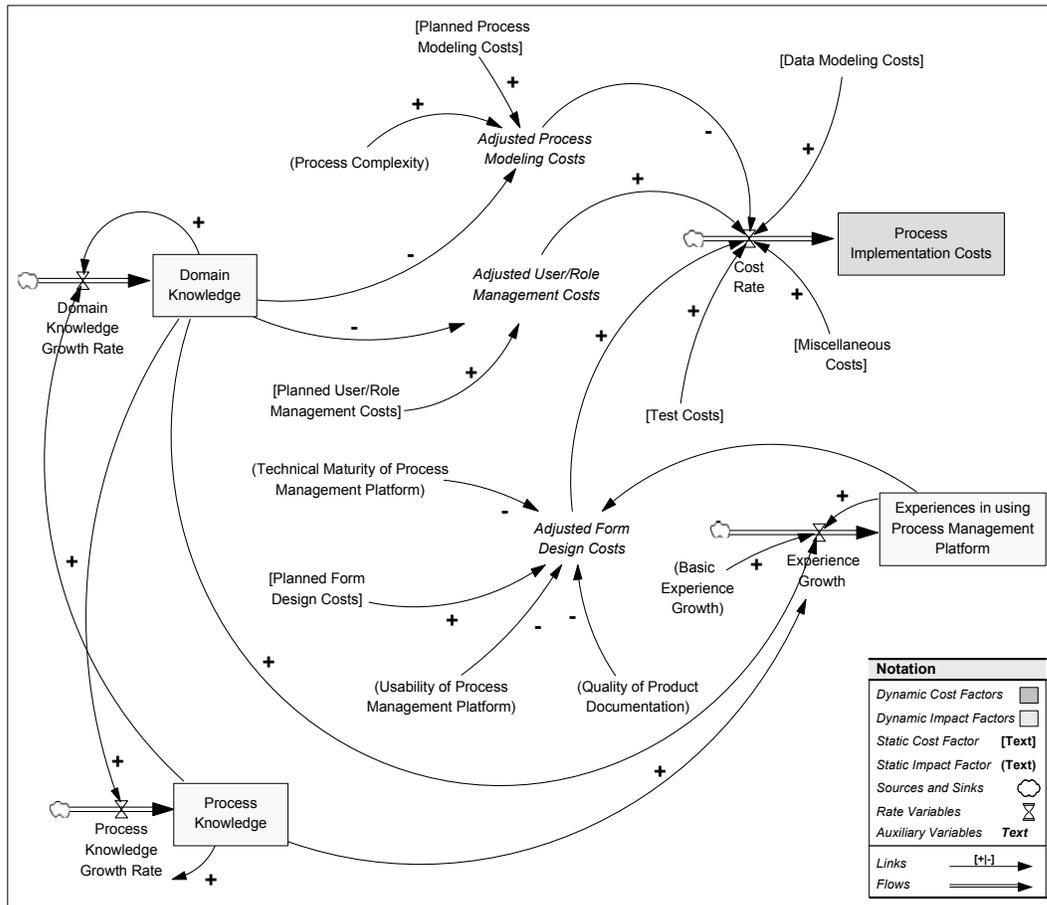


Figure 11.6: Process Management Costs (Simulation Results).



The corresponding Simulation Model is described in Appendix C.

Figure 11.7: IT System Realization Costs (Evaluation Model and Simulation Results).

results and practical experiences.

Step 5: Evaluation Model. As can be seen in Fig. 11.7, the DCF "Process Implementation Costs" is determined by six SCFs: "Data Modeling Costs", "Planned Process Modeling Costs", "Planned User/Role Management Costs", "Planned Form Design Costs", "Test Costs", and "Miscellaneous Costs". Note that these cost factors have been already discussed in detail in Chapter 9 (including possible cost distributions).

Based on interview results and practical experiences we assume that three of these SCFs are further influenced by ImFs. First, the SCF "Planned Process Modeling Costs" is influenced by the static ImF "Process Complexity" as well as by the dynamic ImF "Domain Knowledge". In the evaluation model from Fig. 11.7 this situation is covered by the auxiliary variable "Adjusted Process Modeling Costs" merging the SCF with the ImFs.

Second, the SCF "Planned Form Design Costs" is influenced by the three static ImFs "Technical Maturity of Process Management Platform", "Usability of Process Management Platform", and "Quality of Product Documentation", as well as by the dynamic ImF "Experiences in using Process Management Platform". Accordingly, the SCF and the related ImFs are merged in the auxiliary variable "Adjusted Form Design Costs".

Third, we assume that the SCF "Planned User/Role Management Costs" is influenced by the dynamic ImF "Domain Knowledge". Reason is that implementing user/role management requires detailed organizational knowledge about involved users and user groups (e.g., engineers, suppliers) and their authorizations and responsibilities.

Step 6: Simulation. Fig. 11.7 also shows corresponding simulation results. Fig. 11.7A illustrates the evolution of the dynamic ImF "Experiences in using Process Management Platform" along the simulated time domain of 200 days. In particular, it can be seen that a higher starting level of process knowledge results in a faster buildup of experience. This illustrates the need to carefully analyze the business processes that shall be supported by the new PAIS.

Fig. 11.7B shows that process implementation costs are lower in case of a higher initial process knowledge, but exceed originally planned costs anyway.

11.3.4 Analyzing Specification and Test Costs

Fig. 11.8 depicts the evaluation model we have developed for exploring the overrun of specification and test costs. This evaluation model is based on the primary EP "Requirements Definition Costs" (cf. Section 6.3.3) and the three secondary EPs "End User Fears" (cf. Section 6.4.1), "Process Knowledge" (cf. Section 6.4.2), and "Domain Knowledge" (cf. Section 6.4.3). Like before, these EPs have been selected based on results of our interviews as well as based on practical experiences we gathered in the analyzed project.

Step 5: Evaluation Model. In Fig. 11.8 the DCF "Requirements Definition Costs" is determined by three cost factors (according to our practical experiences and interviews). Costs for using a requirements management tool are represented by the SCF "Requirements Management Tool Costs". Costs for performing interview-based process analysis are rep-

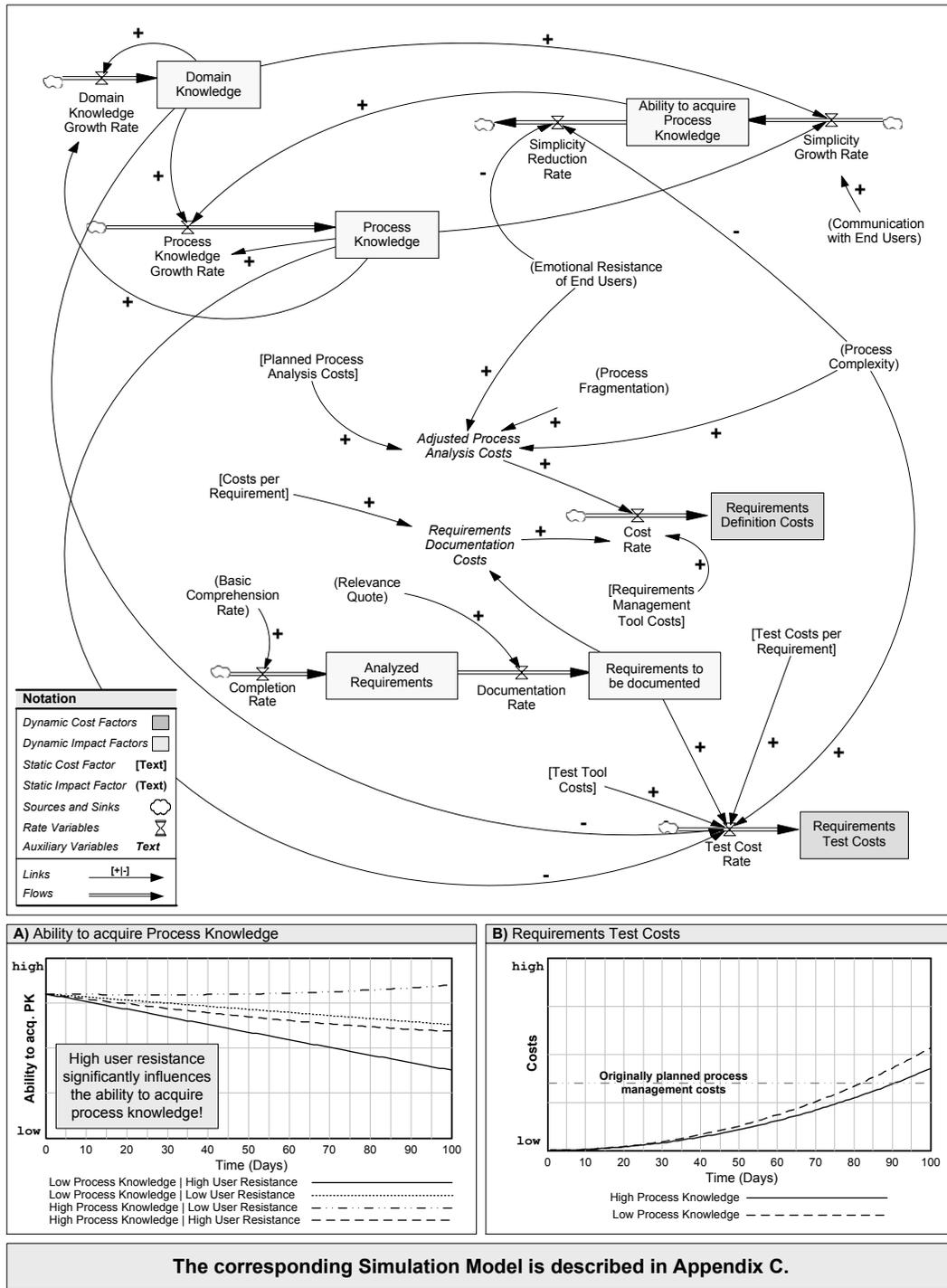


Figure 11.8: Specification and Test Costs (Evaluation Model and Simulation Results).

resented by the SCF "Planned Process Analysis Costs". Performing interviews³ becomes necessary in order to elicit requirements. Costs for documenting identified requirements are represented by the auxiliary variable "Requirements Documentation Costs", which, in turn, is influenced by the SCF "Costs per Requirement" and the dynamic ImF "Requirements to be documented".

Based on interview results, the SCF "Planned Process Analysis Costs" is influenced by three static ImFs: "Process Fragmentation", "Emotional Resistance of End Users", and "Process Complexity". Accordingly, the SCF and the three ImFs are merged in the auxiliary variable "Adjusted Process Analysis Costs".

Finally, the DCF "Requirements Test Costs" is influenced by the SCF "Test Costs per Requirement", the dynamic ImF "Requirements to be documented", the SCF "Test Tool Costs", and by the two dynamic ImFs "Process Knowledge" and "Domain Knowledge". The relevance of two latter ImFs is confirmed by our interviews. All factors are finally merged in the rate variable "Test Cost Rate" (cf. Fig. 11.8).

Step 6: Simulation. Fig. 11.8 also shows simulation results for this evaluation model for the simulated time domain of 100 days. Obviously, the ability to acquire process knowledge strongly depends on the initial levels of process knowledge and user resistance (cf. Fig. 11.8A). In particular, if process knowledge has a high initial value and user resistance has a low initial value, the ability to acquire process knowledge will faster increase (when compared to other combinations). This result is another valuable information for the managers of the considered project. Fig. 11.8B shows that – like requirements definition costs – requirements test costs exceed the originally planned costs.

11.4 Discussion

Step 7: Deriving Conclusions. The detailed results of our simulations have been already discussed in Sections 11.3.2 to 11.3.4 and shall not be repeated here. Generally, simulation results show that the planned costs as included in the initial business case cannot be considered as realistic. The simulated costs for each analyzed cost category exceed the originally estimated costs. Reason obviously is that the original business case does neither consider potential effects of causal dependencies in the analyzed project nor does it analyze possible implications of intangible ImFs.

Thereby, our case study reveals the importance of the five ImFs *end user fears*, *emotional resistance of end users*, *domain knowledge*, *process knowledge*, and *end user participation*.

The key role of emotional resistance of end users can be well illustrated based on our survey results. According to our surveys, emotional resistance of end users is mainly caused by significant efforts for the end users of the new PAIS. 70% of the participants from the first survey among "component responsables" (cf. Fig. 11.9A) consider the efforts resulting from providing support for the considered project (e.g., during process design) as either "very

³Note that these interviews differ from the interviews considered in the previous section. In this case, interviews aim at identifying the business functions that need to be implemented to support the previously identified process activities.

high" (50%) or "high" (20%). Likewise, 59.42% of the participants from the second survey among "main data customers" (cf. Fig. 11.9C) consider efforts as "very high" (24.64%), "high" (14.49%), or "noticeable" (20.29%). In the given context, this unexpected effort can be considered as the main driver of increasing emotional resistance of end users. Emotional user resistance is further increased by the fact that these efforts have not been conceivable for end users (cf. Fig. 11.9B and Fig. 11.9D).

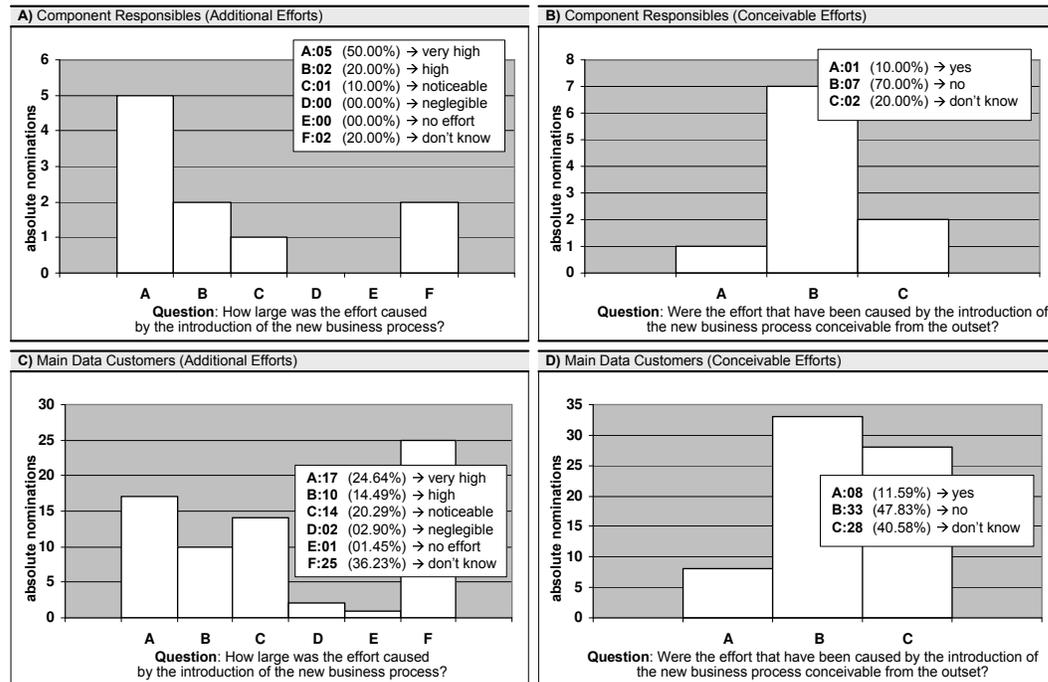


Figure 11.9: Efforts Caused by the new Process Designs.

Moreover, many survey participants – and this is also confirmed by our interviews – consider the current E/E system development process as sufficiently efficient. Survey respondents do not see a need for introducing a new solution (even though the current E/E system development process is based on intensive email communication, phone calls, and the exchange of complex excel sheets). Accordingly, survey results indicate that the newly introduced process designs are considered as not efficient by the majority of survey participants: 60% of the component responsables and 33.34% of the main data customers state that process efficiency has not increased through the new process designs (cf. Fig. 11.10A and Fig. 11.10C). These results also confirm the high level of emotional resistance of end users in the analyzed project.

However, 80% of the component responsables (cf. Fig. 11.10B) look favourably upon future process efficiency ("yes": 10% and "rather yes": 70%). In the case of main data customers, results are more divergent (cf. Fig. 11.10D).

Altogether, our simulation results clearly show the (under-estimated) role of intangible impact factors in PAIS engineering projects. Besides these specific case study findings, we

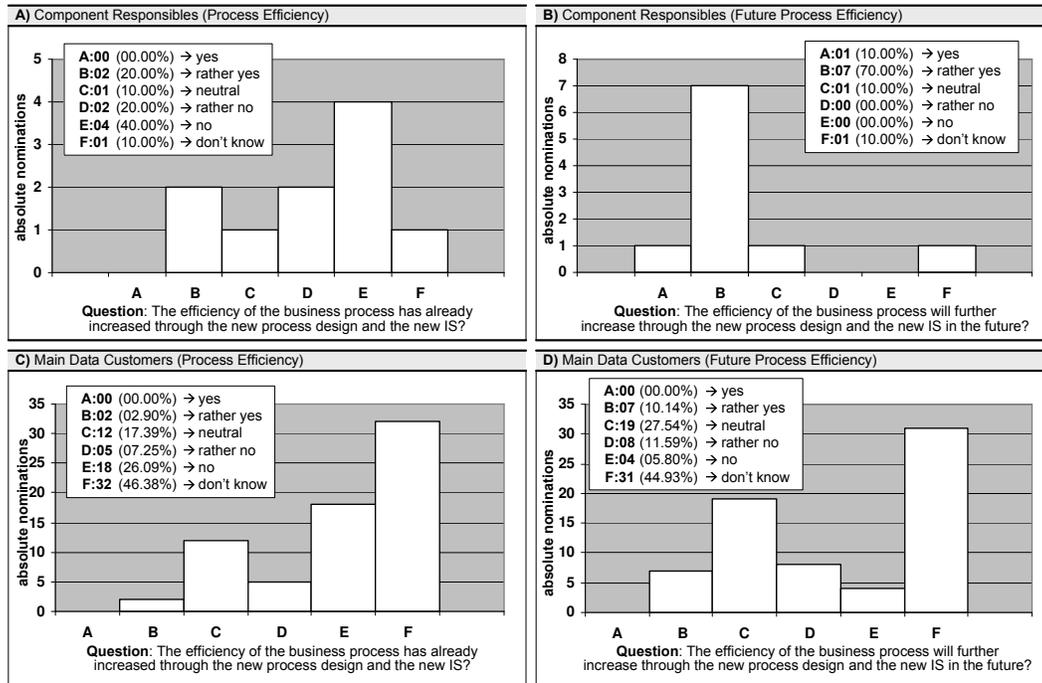


Figure 11.10: Efficiency of the new Process Designs.

can also draw conclusions regarding the overall applicability of the EcoPOST framework. These conclusions are summarized in the following section.

11.5 Summary and Lessons Learned

This chapter has described a case study in which we have applied the EcoPOST framework to investigate cost overruns observed in a large PAIS engineering project from the automotive domain. Based on real project data, interviews with project members, two online surveys among end users of the PAIS, and practical experiences gathered in the analyzed PAIS engineering project, we have developed a set of three evaluation models. To analyze the dynamic implications of these models we have additionally simulated them.

Case Study Conclusions. Based on the derived evaluation models and simulation outcomes, we are able to show that the costs as estimated in the initial business case are not realistic. The simulated costs for each analyzed cost category exceed the originally estimated costs. Moreover, our evaluation models provide valuable insights into the reasons for the occurred cost overruns, particularly into causal dependencies and resulting effects on the costs of the analyzed PAIS engineering project.

Lessons Learned. Regarding the overall goal of the case study, i.e., the investigation of the practical applicability of the EcoPOST framework and its underlying evaluation concepts,

our experiences confirm the expected benefits. More specifically, we can summarize our experiences by means of six lessons learned:

1. Our case study confirms that the EcoPOST framework enables PAIS engineers to gain valuable insights into causal dependencies in PAIS engineering projects as well as into resulting cost effects.
2. EcoPOST evaluation models are easy to understand and can support IT managers and policy makers in understanding a PAIS engineering project and decision-making.
3. PAIS engineering projects constitute complex socio-technical feedback systems which are characterized by a strong nexus of organizational, technological, and project-specific parts. All of our evaluation models include at least one feedback loop.
4. Our case study confirms that evaluation models can become complex due to the large number of potential SCFs, DCFs, ImFs, and causal dependencies that exist between them. However, the governing guidelines from Chapter 7, which we also took into account during this case study, help to avoid too complex evaluation models.
5. The availability of evaluation patterns significantly simplifies the design of evaluation and simulation models.
6. Though our simulation models have been build upon data derived from four different information sources (i.e., project data, interviews, online surveys, and practical experiences), it has turned out that it is inevitable to rely on hypotheses in order to build complete simulation models.

Altogether, our case study shows that the expected benefits of the EcoPOST framework can be realized – though some critical issues emerged as well (e.g., the need to rely on hypotheses when developing simulation models). In the following chapter we discuss the contributions of the EcoPOST framework as well as its benefits and limitations in detail.

Part IV

Discussion and Summary

Chapter 12

Discussion

Evaluating IT investments regarding their costs, benefits, and risks is a complex task to accomplish. Even more complex are respective evaluations of PAIS engineering projects. In the previous chapters, we have not only shown that this difficulty stems from the interplay of the many technology-, organization-, and project-specific factors which arise in the context of PAIS engineering projects. In order to deal with this challenge, we have also introduced the EcoPOST framework, a practically approved, model-based approach which enables PAIS engineers to better understand and investigate causal dependencies and resulting cost effects in PAIS engineering projects.

But are the evaluation concepts underlying the EcoPOST framework really suitable to capture causal dependencies and related effects in PAIS engineering projects? Is the use of simulation applicable in our context? What has been done and what still needs to be addressed in order to increase the expressiveness of conclusions derived using the EcoPOST framework? This chapter picks up these and other issues and discusses them.

Complexity of IT Evaluation. Today, IT evaluation is typically based on simple and static models as well as on intuition and experiences rather than on a profound analysis. Besides, rules of thumb such as "*invest to keep pace with technology*" or "*invest if competitors have been successful*" are often used as an evaluation baseline as well. Moreover, many financial evaluations (as discussed in Sections 2.4.1 and 2.4.2) exhibit an asymmetric consideration of short-term costs and long-term benefits, e.g., in order to increase "sympathy" for a potential IT investment by "proving" an extremely positive ROI.

Generally, existing IT evaluation approaches and software cost estimation techniques will lack satisfactory outcomes, in particular, if they are used at early planning stages of IT investments. Consequently, many IT projects (especially those dealing with innovative IT like PAIS engineering projects) often have – despite their potential strategic importance – a negative economic valuation result at an early stage (even if an asymmetric consideration of short-term costs and long-term benefits is intentionally enforced; see above). In particular, this situation may result in a high risk of false rejection of IT investments, i.e., decision makers need to avoid the problem of not routinely rejecting important IT investments based on results of too simple or inadequate evaluation techniques.

Likewise, these problems are valid in the context of PAIS and PAIS engineering projects as well. In particular, existing IT evaluation techniques are unable to take into account the numerous technology-, organization-, and project-specific evaluation factors arising in PAIS engineering projects. They are also unable to cope with the causal dependencies and interactions that exist between these factors and the resulting effects. Even the process-oriented approaches discussed in Section 2.5 (TSTS approach, hedonic wage model, etc.) do not allow to address these issues, i.e., they can only be applied to evaluate the impact of a PAIS on organizational business process performance and work performance.

Special Case: Process-aware Information Systems. The introduction of a PAIS – like the introduction of any large IT system – is typically associated with high costs. These costs need to be systematically analyzed and monitored during a PAIS engineering project. Yet, there exist no approaches to do so, for existing IT evaluation approaches and software cost estimation techniques are unable to deal with the complex interplay of the many cost and impact factors which arise in the context of such projects. Thus, decision makers thus often elude to less meaningful evaluation criteria (e.g., mere technical feasibility) and often rely on assumptions (e.g., regarding benefits such as improved business process performance) when justifying the costs of PAIS engineering projects. From the decision maker’s viewpoint, this is rather insufficient.

What is needed, by contrast, is a comprehensive approach which enables PAIS engineers to investigate the complex interplay between cost and impact factors in PAIS engineering projects. Our EcoPOST framework (as presented in Part II of this thesis) picks up this challenge and particularly focuses on analyzing the dynamic interplay and causal dependencies of those factors that determine the complex costs of PAIS. Note that this has also been one major requirement that has guided the development of the EcoPOST framework (cf. Requirement 9 in Chapter 3).

Requirements Discussion. More generally, Chapter 3 has identified eleven requirements for the design of an economic-driven evaluation approach for PAIS engineering projects: five requirements for performing economic-driven IT evaluation in general (cf. Section 3.2), and six specific requirements for evaluating PAIS engineering projects (cf. Section 3.3). Fig. 12.1 shows the identified requirements and additionally depicts whether our EcoPOST framework fulfills them (either fully or only partly).

As a first requirement we have identified the performance of *cost-oriented evaluations* (R-1). The EcoPOST framework fully supports this requirement based on its evaluation models comprising both static and dynamic cost factors (SCFs and DCFs). We have also considered assistance for *decision support* as relevant requirement (R-2). However, this requirement is only partly fulfilled. Focus of the EcoPOST framework is on analyzing evolving dynamic cost and impact factors along the course of time. Decision support is only implicitly given by raising awareness about the complex causal dependencies and resulting cost effects emerging in PAIS engineering projects. Explicit criteria enhancing decision making, by contrast, are not provided. As a further requirement, we have identified the *derivation of plausible conclusions* (R-4). Deriving plausible conclusions, however,

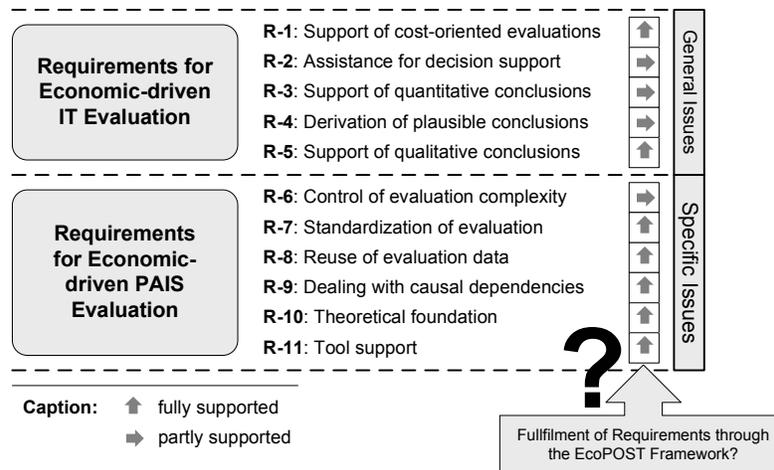


Figure 12.1: Requirements at a Glance.

strongly depends on the availability of adequate¹ evaluation data underlying the developed evaluation and simulation models. Generally, the development of plausible evaluation and simulation models is a difficult task to accomplish. Notwithstanding, it is possible to scrutinize the overall suitability of evaluation models (though "*validation and verification of models is impossible*" [189]). As examples for respective actions consider the compliance of evaluation models with defined model design rules (cf. Section 4.9.3) and the careful consideration of governing guidelines (cf. Chapter 7). Most important, however, is the availability of quantitative data. The experimental and empirical research activities we have described in Part III of this thesis have been important examples in this respect. Notwithstanding, there remain many unclarities about causal dependencies in PAIS engineering projects and we face the (common) problem of missing quantitative data. As final general requirements for economic-driven IT evaluation, we have identified the support of both *quantitative* (R-3) and *qualitative* (R-5) *conclusions*. Both requirements are supported by the EcoPOST framework. While qualitative conclusions can be derived based on our evaluation models and the causal dependencies and feedback loops specified by them, quantitative conclusions are only possible to some degree based on the simulation of evaluation models and the interpretation of respective outcomes.

Besides, we have identified six specific requirements for evaluating PAIS engineering projects. First, we have recognized *control of evaluation complexity* as a relevant requirement (R-6). In this context, the most important step – besides the provision of governing guidelines and the availability of adequate tool support – has been the introduction of *evaluation patterns* (EP) in Chapter 6. These predefined evaluation models can significantly reduce the complexity of building evaluation models as it is not always necessary to develop an evaluation model from scratch. Besides, *standardization of evaluation* has

¹We will denote evaluation data as "adequate", if the data clearly support the applicability of an evaluation approach. While some approaches require the availability of real project data, others additionally settle for data derived from on interviews, surveys, or focus-group sessions.

been identified as relevant requirement (R-7). Considering our clearly specified evaluation methodology (comprising seven consecutive steps), the provided governing guidelines, and the availability of EPs, we consider this requirement as being fulfilled. Further, *reusing historical evaluation data* has been an important requirement as well (R-8). This requirement can be also considered as fulfilled considering both the availability of EPs and the possibility to store evaluation models and EPs in the model repository of the *EcoPOST Cost Benefit Analyzer*. Also, the *modeling of causal dependencies* has been an important requirement (R-9). We consider this requirement as being fulfilled as modeling causal dependencies is one fundamental notion underlying our evaluation models. As another requirement, we have demanded for a sufficient degree of *formalization* (R-10). In this context, it is important to mention that our evaluation and simulation models – like conventional System Dynamics models – have a sound theoretical foundation (cf. Chapter 4 and Chapter 5 for formal definitions). Finally, we have considered *tool-support* as crucial (R-11). In response to this, Chapter 8 has described how the combination of a System Dynamics modeling and simulation tool and the *EcoPOST Cost Benefit Analyzer* supports the enforcement of EcoPOST evaluations. Hence, we consider this requirement as fulfilled.

Using System Dynamics. As aforementioned, we use *System Dynamics* (SD) for specifying our evaluation models. SD is a formalism for studying and modeling complex feedback systems, such as one finds in biological, environmental, industrial, business, and social systems [166, 147]. Its underlying assumption is that the human mind is excellent in observing the elementary forces and actions of which a system is composed (e.g., pressures, fears, delays, resistance to change), but unable to understand the dynamic implications caused by the interaction of a system's parts² [76].

In PAIS engineering projects we have the same situation. Such projects are characterized by a strong nexus of organizational, technological, and project-driven factors. Thereby, the identification of these factors constitutes one main problem. Far more difficult is to understand causal dependencies between factors and resulting effects. Only by considering PAIS engineering projects as feedback systems we can really unfold the dynamic effects caused by these dependencies (i.e., by the interacting organizational, technological, and project-driven system parts).

There are other formalisms that can be applied to this context as well. Causal *Bayesian Networks* (BN) [84], for example, promise to be a useful approach. BN deal with (un)certainly and focus on determining probabilities of events. A BN is a directed acyclic graph which represents interdependencies embodied in a given joint probability distribution over a set of variables. Variables can be measurable or intangible parameters or random variables (which form the "Bayesian" aspect of a BN). In our context, we are interested in the interplay of the parts (components) of a system and the effects resulting from this interplay. BN do not allow to model feedback loops as cycles in BN would allow infinite feedbacks

²System Dynamics can be easily confounded with *Systems Thinking* [215]. Generally, Systems Thinking utilizes the same kind of models to describe causal dependencies, but does not take the additional step of constructing and testing a computer simulation model, and does also not test alternative policies in a model (i.e., sensitivity analysis). A good overview on the role of System Dynamics and Systems Thinking in the context of the evolution of the systems sciences is given in [179].

and oscillations that would prevent stable parameters of the probability distribution.

Agent-based modeling provides another promising approach. Resulting models comprise a set of reactive, intentional, or social agents encapsulating the behavior of the various variables that make up a system [34]. During simulation, the behavior of these agents is emulated according to defined rules [178]. System-level information (e.g., about intangible factors being effective in a PAIS engineering project) is thereby not further considered. However, as system-level information is an important aspect in our approach, we have not further considered the use of agent-based modeling.

Benefits. Based on the use of SD, the EcoPOST framework can unfold its benefits. In particular, the EcoPOST framework is the first available approach to systematically structure knowledge about PAIS engineering projects, to interrelate both hard facts and soft observations on respective projects, and to investigate and better understand causal dependencies and resulting effects emerging in them. Focus is on analyzing the complex interplay of organizational, project-specific, and organizational cost and impact factors. Not using the framework would imply that existing knowledge and experiences remain a mere collection of observations, practices, and conflicting incidents, making it very difficult to derive conclusions regarding the costs of PAIS engineering projects. More specifically, the benefits of our approach can be summarized as follows:

- **Feedback:** In Chapter 3, we have denoted the modeling of feedback structures and causal dependencies between cost and impact factors as one major requirement. Our evaluation models pick up this requirement and enable not only the modeling of causal dependencies, but also the investigation of cyclic feedback structures.
- **Visualization:** Our evaluation models offer a simple way of visualizing both the structure and the behavior of interacting cost and impact factors in PAIS engineering projects. Thus, evaluation models can be easily communicated to decision makers³.
- **Intangible Impacts:** Our evaluation models enable the PAIS engineer to investigate the effects of intangible impact factors (such as end user fears and process knowledge).
- **Delays:** PAIS engineering projects are typically faced with many delays (e.g., related to the evolution of dynamic cost and impact factors). These delays often develop over time due to internal or external influences. End user fears, for example, may be low at the beginning of a PAIS engineering project, but may quickly increase later. Our evaluation models allow to deal with such delays based on the notion of using state variables to represent both DCFs and dynamic ImFs.
- **Sensitivity Analysis:** Every modification of a variable in an evaluation model (respectively simulation model) results in various consequences. Some of these consequences can be anticipated and intended. Many others, however, are typically unanticipated and unintended. The opportunity to perform sensitivity analysis is of signi-

³In [16] and [191], this benefit has been generally confirmed for the management of software projects.

ficant help in this context and can provide valuable insights into the consequences of changing variables.

Yet, our approach has also some limitations. The design of our evaluation models, for example, can constitute a time-consuming task – despite the availability of a library of predefined evaluation patterns. Also, evaluation models can become complex due to the large number of potential evaluation factors and the causal dependencies that exist between them. Further, the quantification of ImFs and their either linear or nonlinear effects (respectively their specification in simulation models) is difficult. When building simulation models, it often cannot be avoided to rely on assumptions. Generally, there remain many unclarities about causal relationships and feedback loops in PAIS engineering projects.

Experimental and Empirical Validation. In response to this problem experimental and empirical research activities have constituted a main pillar when developing the EcoPOST framework. Such research activities are particularly important where analytic solutions are neither possible nor practical and where emerging technologies (instead of mature ones) are considered [124]. According to Briand et al. [35] "*each discipline needs to develop its own body of experience and strategies to answer its most pressing research questions*".

Basic issues on software experimentation, for example, are described by Basili [17], Juristo [87], and Pfleeger [150, 151]. More specific guidelines on applying experimental models to IT validation are described by Zelkowitz [226]. Guidelines for reporting on the results of software experiments are given by Jedlitschka [83]. Tichy [198] promotes quantitative, controlled, statistically-analyzable experimentation and Kitchenham et al. [97] recognize the value of "observational studies" in addition to formal experimentation (but also emphasize industrial-context, quantitative evaluation, and statistics). Seaman [180], in turn, promotes the value of qualitative evaluation. For conducting case study research, Yin [220], Eisenhardt [61] and Kitchenham et. al [91, 92, 93, 98] describe basic research designs and fundamental methods.

In the EcoPOST framework, we have conducted both empirical and experimental research activities at all research phases [217], i.e., during (1) *problem investigation*, (2) *solution validation*, and (3) *implementation evaluation* (cf. Fig. 12.2):

- **Problem Investigation:** During problem investigation, the gap between contemporary solutions and stakeholder goals is analyzed. In the given context, problem investigation analyzes the inability of existing IT evaluation approaches and software cost estimation techniques to evaluate the costs of PAIS engineering projects. Thereby, the following questions are of particular importance: Who are the stakeholders? What are their goals? What is the problem domain? Which phenomena in the domain are problematic, and why? Problem investigation in this thesis has been based on the following empirical research activities (cf. Fig. 12.2):
 - *Exploratory Case Study* (cf. Chapter 1): We conducted an exploratory case study in the automotive domain. Over a period of three months, we analyzed two characteristic business processes and their IS support. We have conducted

26 formal interviews with software developers, domain experts, and users of the considered IS. In particular, this case study has provided us initial insights into many practical problems related to the development and operational use of IS providing business process support.

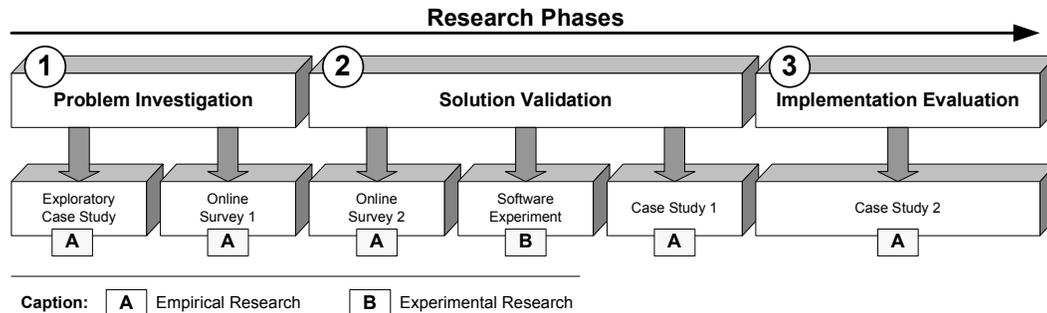


Figure 12.2: Experimental and Empirical Research in the EcoPOST Framework.

- *Online Survey 1* (cf. Chapters 1 & 3): In order to increase the validity of the results from our domain-specific case study, we performed an online survey among IT practitioners from different disciplines and domains. Altogether, 79 IT practitioners participated in this online survey.
- **Solution Validation:** Unlike problem investigation, solution validation investigates the properties of a proposed solution (which has not yet been completely implemented). For this purpose, mathematical analysis, prototypes, simulations, and other ways to provide engineering arguments are used. The basic goal of solution validation is to acquire sufficient knowledge about possible solution details (in order to be able to choose which one to implement). In this case, solution validation has been supported through the following research activities:
 - *Online Survey 2* (cf. Chapters 4 & 5): In order to empirically confirm the relevance of assumed intangible ImFs (e.g., process knowledge, business process redesign, end user fears, and communication) we have conducted an online survey among 70 BPM experts. This second online survey has been particularly important as the identified impact factors (cf. Section 4.8) represent an important baseline when building evaluation models.
 - *Controlled Software Experiment* (cf. Chapter 9): We have presented the results of a controlled software experiment with 48 students. This experiment has compared the implementation of business processes using both workflow technology and case handling technology. In particular, our experiment has illustrated how experimental research can be applied to the validation of evaluation and simulation models.
 - *Case Study 1* (cf. Chapter 10): We have presented the results of a case study at two Dutch companies. In this case study we have investigated in which way

employees perceive the impact of a newly introduced PAIS with respect to five job dimensions: skill variety, task identity, task significance, autonomy, and feedback from the job.

- **Implementation Evaluation:** Implementation evaluation, finally, investigates the overall performance of a developed solution approach. It analyzes whether the selected solution behaves as predicted and fulfills the expected benefits. In the case of the EcoPOST framework, implementation evaluation has been based on a case study in the automotive domain:
 - *Case Study 2* (cf. Chapter 10): In this case study, we used the EcoPOST framework to analyze the cost overruns related to the introduction of a large PDM system. Based on real project data, interviews, two online surveys, and practical experiences we have developed a set of evaluation and simulation models to examine whether the observed cost overruns could have been predicted using our EcoPOST methodology.

Generally, experimental and empirical research activities can contribute to improve the overall validity of evaluation and simulation models. However, there will always be a trade-off between the benefits gained through generating data and the typically significant efforts caused by such activities.

Success Factors. There are a number of *critical success factors* (CSF) which can foster the transfer of the EcoPOST framework into practice. *First*, it is important that the users of the EcoPOST framework get enough time to get familiar with the underlying evaluation concepts and the provided tools. Using the EcoPOST framework will be difficult, particularly if it is initially used. It exhibits a comparatively large number of different evaluation concepts and tools and it will need some time for users to effectively apply them. In practice, this can be a problematic barrier for potential users. However, this complexity quickly decreases through gathered experiences. *Second*, it is crucial that results of EcoPOST evaluations are carefully documented. This allows not only to fall back on these results at a later day. It also allows to reflect past evaluations and problems respectively lessons learned. *Third*, evaluation models should be validated in an open forum where stakeholders such as policy makers, project managers, PAIS architects, software developers, and consultants have the opportunity to contribute to the model development process.

Process-oriented Software Technologies in Practice. Generally, our EcoPOST framework allows for evaluating PAIS and PAIS engineering projects from a systemic viewpoint. The timeliness of our approach is also confirmed by results from our first survey in which we have investigated the dissemination of technologies enabling the realization of PAIS.

63.29% of the respondents confirm the use of *WfMS* (cf. Fig. 12.3A). 36.71% even deploy more comprehensive *BPM systems*. 53.16% of the respondents confirm the (exploratory) use of *web services*. 32.91% already set up complete *service-oriented architectures*. *Enterprise application integration* platforms (e.g., to enable process integration) are applied by 39.24% of the respondents.

These results do not allow for any conclusion regarding the extent to which enterprises use a respective technology. Nevertheless, the increasing importance and use of process-oriented software technologies is indicated.

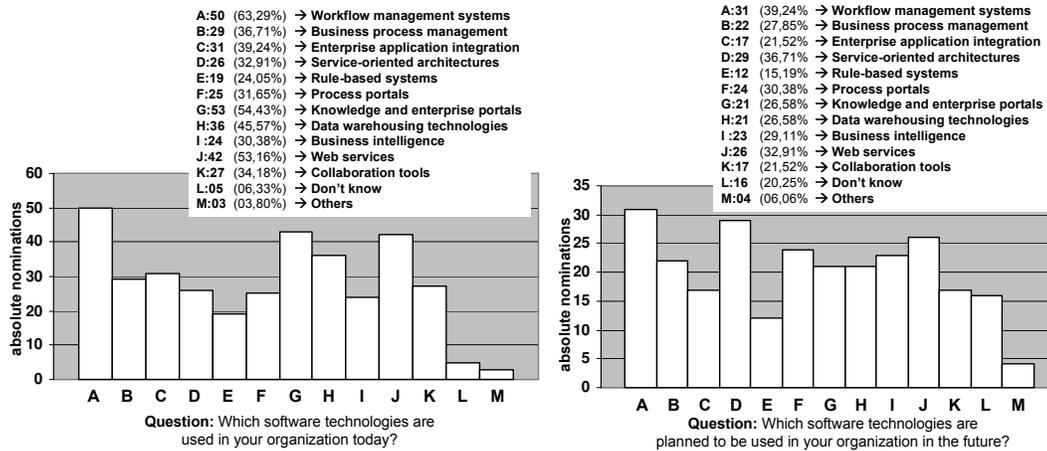


Figure 12.3: Software Technologies used Today and Tomorrow.

In order to investigate the sustainability of this trend, we have asked survey participants about which process-oriented software technologies they will use in future (cf. Fig. 12.3B). The most frequent answers are *WfMS* (39.24%) *services-oriented architectures* (36.71%), *web services* (32.91%), and *process portals* (30.38%).

These results indicate a continuing trend towards an increasing dissemination of process-oriented software technologies and PAIS and also show the necessity to have suitable evaluation approaches such as the EcoPOST framework at hand.

Summary. In Chapter 2, we have introduced a comparison framework to classify and compare economic-driven IT evaluation approaches. Fig. 12.4 applies this framework to our EcoPOST framework. Basically, the EcoPOST framework addresses the financial viewpoint (*Criterion C-1*). Decision support is only implicitly given by raising awareness about complex causal dependencies and resulting cost effects in PAIS engineering projects (*Criterion C-2*). Explicit criteria enhancing decision making are not provided. By contrast, the EcoPOST framework aims at the analysis of costs, but can address benefits and risk factors as well (*Criterion C-3*). Both ex-ante and ex-post (*Criterion C-4*), quantitative and qualitative evaluations (*Criterion C-5*) can be performed.

Plausibility of evaluation outcomes strongly depends on available data underlying EcoPOST evaluation and simulation models and is therefore considered as neither positive nor negative (*Criterion E-1*). Similarly, the degree of objectiveness (*Criterion E-2*) as well as the sensitivity of the EcoPOST framework (*Criterion E-3*) depend on the modeler and are therefore considered as neither positive nor negative. Regarding the practical applicability of the EcoPOST framework it is particularly the supported evaluation flexibility (*Criterion E-4*) which has to be considered as advantageous. The degree of formalization is high (*Criterion E-5*) and providing tool support is simple (*Criterion E-6*).

The EcoPOST Framework				
I. Classification Criteria	C-1: Evaluation Viewpoint			
	Financial Viewpoint		X	
	Work Performance Viewpoint			
	Strategic Viewpoint			
	C-2: Decision Support			
	The EcoPOST framework supports decision-making indirectly by increasing awareness about cost effects.			
	C-3: Evaluation Dimensions			
	Evaluation of Benefits		O	
	Evaluation of Costs		X	
	Evaluation of Risks		O	
	Evaluation of Work Performance			
	C-4: Evaluation Scope			
	ex-ante		X	
	ex-post		X	
C-5: Evaluation Outcome				
Quantification Absolute Measures		O		
Quantification Relative Measures		X		
Visualization of Outcome		X		
Qualitative Conclusions		X		
II. Evaluation Criteria	E-1: Plausibility			
	Interpretability of Evaluation Results		➡	
	Transparency of Result Generation		➡	
	E-2: Objectiveness			
	Degree of Objectiveness		➡	
	E-3: Sensitivity			
	Error-Proneness of Evaluations		⬇	
	Resistance against Manipulation		➡	
	E-4: Practical Applicability			
	Flexibility		⬆	
	Efficiency (Effort)	Data Collection & Preparation	Availability	➡
			Heterogeneity	⬆
			Data Quality	⬇
		Overall Effort		➡
Effective-ness	Comparability		⬆	
	Correctness of Conclusions		➡	
E-5: Theoretical Foundation				
Degree of Formalization			⬆	
E-6: Tool Support				
Simplicity of providing Tool Support			⬆	

Caption: x: supported o: optional ⬆: positive ⬇: negative ➡: neutral

Figure 12.4: Evaluation of the EcoPOST Framework.

Chapter 13

Summary and Outlook

This thesis has introduced the EcoPOST framework. This framework enables PAIS engineers to model, simulate, and investigate causal dependencies and resulting cost effects in PAIS engineering projects. Focus is on analyzing cost factors and their interference through organizational, technological, and project-specific impact factors.

Our research has started with an analysis of a frequent problem in practice, namely, to understand and evaluate the cost effects of complex causal dependencies and feedback structures in PAIS engineering projects. During this initial phase of our research, we have conducted a literature study on economic-driven IT evaluation approaches and software cost estimation techniques, an exploratory case study in the automotive domain, and a first online survey. We have utilized results of these activities to identify major requirements for investigating PAIS engineering projects. Based on these requirements, we have designed our EcoPOST framework comprising evaluation models, simulation, evaluation patterns, governing guidelines, and tool support. Finally, we have validated our solution based on a second online survey, two case studies, and a controlled software experiment. More specifically, the contributions of this thesis can be summarized as follows:

- We have identified organizational, technological, and project-specific impact factors which influence the costs of PAIS engineering projects.
- We have introduced a comprehensive methodology for modeling causal dependencies and simulating resulting (cost) effects in PAIS engineering projects.
- We have described a basic set of generic, customizable evaluation models called *evaluation patterns* (EP). Primary EPs specify dynamic cost factors and their influence through impact factors. Secondary EPs describe effects related to impact factors.
- We have described governing guidelines and best practices which support PAIS engineers when using the EcoPOST framework.
- We have illustrated how the combination of a regular System Dynamics tool and the *EcoPOST Cost Benefit Analyzer* support the application of the EcoPOST framework.

- Each evaluation (and simulation) model can be considered as a specific theory on cost factors. In order to confirm these theories, we have shown how experimental and empirical research activities can be applied to validate evaluation and simulation models.
- We have demonstrated the practical applicability of our solution based on a case study in which we have analyzed cost overruns in a large PAIS engineering project in the automotive industry.

Our framework is particularly suitable to gain insights into the complex causal dependencies and resulting effects which arise in PAIS engineering projects. It is the first solution which enables PAIS engineers to systematically structure knowledge and experiences about PAIS engineering projects and to interrelate facts and observations. Without our framework, knowledge about PAIS engineering projects remains a mere collection of observations, best practices, and conflicting incidents, making it very difficult to learn from experiences.

Yet, our approach has also some limitations. The design of evaluation models, for example, can constitute a time-consuming task – despite the availability of predefined evaluation models (i.e., EPs). Also, evaluation models can become complex due to the potentially large number of organizational, project-specific, and technological impact factors and causal dependencies that exist between them. The biggest challenge, however, is the quantification of impact factors and non-linear relationships in our simulation models. In fact, to be able to build simulation models, it is often inevitable to rely on hypotheses, sometimes even arguable assumptions. We face the problem of missing data for substantiating simulation models (though we have illustrated how empirical and experimental research can contribute to generate needed data). Generally, there still remain unclarities about causal dependencies and feedback loops in PAIS engineering projects.

Future research will contribute to further resolve these limitations. The basic set of primary and secondary EPs described in Chapter 6, for example, needs to be extended in order to further facilitate the practical applicability of the EcoPOST framework. Thereby, it is not only necessary to identify new EPs, but also to effectively manage them. Note that the model repository of the *EcoPOST Cost Benefit Analyzer* can provide support in this context. Besides, it will also be necessary to collect and document best practices, e.g., on the quantification of impact factors, on the use of simulation, and on the effectiveness of available tool support. The described governing guidelines are only a first step in this respect. Finally, it will be also subject of future research to extend the framework itself, for example, regarding its suitability to enable more quantitative conclusions (which is also important from a practical-driven perspective).

Note that the EcoPOST framework has been developed in the automotive domain, i.e., in a domain which is characterized by complex engineering processes, a strong nexus of organizational, project-specific, and technological factors, and typically very large PAIS. It will be a task as well to investigate the suitability of the EcoPOST framework for other domains (e.g., based on additional case studies). Thereby, one has to separately investigate its suitability for other IT domains (e.g., service-oriented architectures, cross-organizational PAIS), other industry domains (e.g., health care domain, financial services), and also for small and medium enterprises.

Appendices

Appendix A

Glossary

Auxiliary Variable: In an evaluation model, auxiliary variables are intermediate variables which bring together – like rate variables – cost and impact factors, i.e., they merge SCFs, DCFs, and static and dynamic ImFs. → *Rate Variable*

Dynamic Cost Factor (DCF): Represent costs that are caused by activities related to a PAIS engineering project. These activities cause measurable efforts, which, in turn, vary due to the influence of *Impact Factors*. The (re)design of business processes prior to the introduction of PAIS, for example, constitutes such an activity. → *Impact Factor*

Dynamic Impact Factor: A dynamic ImF is an ImF whose value may change along the course of time due to the influence of other ImFs. → *Impact Factor*

Evaluation Model: An evaluation model is specified using the *System Dynamics* (SD) notation. It comprises different kinds of model variables (cost factors, impact factors, rate variables, and auxiliary variables) and makes causal dependencies between model variables explicit. → *System Dynamics*

Impact Factor (ImF): Intangible factor which influences DCFs (or more precisely, the activities underlying a DCF). As examples consider factors such as "End User Fears", "Process Knowledge", or "Ability to redesign Business Processes". ImFs cause the value of DCFs and dynamic ImFs to change. → *Static Impact Factor, Dynamic Impact Factor*

Flows: Values of DCFs and dynamic ImFs change through inflows and outflows. An *inflow* increases the value of DCFs and dynamic ImFs. An *outflow*, by contrast, decreases their value.

Links: SCFs, static ImFs, and auxiliary variables are embedded in an evaluation model with *links* (not flows), i.e., labeled arrows. A *positive link* (labeled with "+") between x and y (with y as dependent variable) indicates that y will tend in the same direction if a change occurs in x. A *negative link* (labeled with "-") expresses that the dependent variable y will

tend in the opposite direction.

Process-aware Information System (PAIS): PAIS strictly separate process logic (comprising the activities to be executed) from application code, i.e., PAIS are driven by process models rather than program code. PAIS can be implemented in two ways: (1) by developing an organization-specific process support system, or (2) by configuring a generic process support system. In the former case, the PAIS is build "from scratch" and incorporates organization-specific information about the structures and processes to be supported. In the latter case, the PAIS does not contain any information about the structures and processes of a particular organization. Instead, an organization needs to configure the PAIS by specifying processes, organizational entities, and business objects.

Rate Variable: Rate variables are assigned to flows. A rate variable controls the flow it is assigned to based on those SCFs, DCFs, and static and dynamic ImFs which influence it. Rate variables – like auxiliary variables – are interfaces which merge SCF, DCF, and static and dynamic ImF. → *Link*

Simulation Model: Consists of a number of algebraic equations – one for each model variable (i.e., SCFs, DCFs, static and dynamic ImFs, as well as rate and auxiliary variables).

State Variable: Represents a dynamic evaluation factor, i.e., it captures changing values of DCFs and dynamic ImFs. The value of a state variable at time t is determined by the accumulated changes of this variable from starting point t_0 to present moment t ($t > t_0$). → *DCF, dynamic ImF*

Static Cost Factor (SCF): The value of a static cost factor does not change during a PAIS engineering project (except for its time value, which is not further considered in our context). As typical examples of SCFs consider software license costs, hardware costs, or costs for external consultants.

Static Impact Factor: A static ImF is an ImF whose value does not change along the course of time. → *Impact Factor*

System Dynamics (SD): SD is a formalism for studying and modeling complex feedback systems, as they can be found, for example, in biological, environmental, industrial, business, and social systems. The underlying assumption of SD is that human mind is excellent in observing the elementary forces and actions out of which a system is composed (e.g., fears, delays, resistance to change), but unable to understand dynamic implications resulting from these forces and actions.

Evaluation Pattern (EP): An EP is a template for DCFs or (static or dynamic) ImFs we encounter in typical PAIS engineering projects. An EP consists of an *evaluation model* and an associated *simulation model*. → *Evaluation Model, Simulation Model*

Appendix B

Simulation Models from Chapter 6

Primary Evaluation Patterns

SECTION 6.3.1 - BUSINESS PROCESS REDESIGN COSTS:

Ability Growth

Equation: $(\text{Domain Knowledge}/100) * \text{Management Commitment} * (\text{Process Knowledge}/100)$

Ability Reduction

Equation: *End User Fears*

Ability to Redesign Business Processes

Equation: $\text{Ability Growth} - \text{Ability Reduction}$

Range: [0 to 100]

Initial Value: [CONSTANT]

Business Process Redesign Costs

Equation: *Cost Growth*

Initial Value: [0]

Cost Growth

Equation: $(\text{Adjusted Costs for Process Analysis} + \text{Adjusted Costs for Process Modeling}) * \text{Impact due to Ability to Redesign Business Processes}$

Domain Knowledge

Equation: *Domain Knowledge Growth*

Range: [0 to 100]

Initial Value: [CONSTANT]

Domain Knowledge Growth

Equation: $(\text{Process Knowledge}/100) + (\text{Domain Knowledge}/100)$

End User Fears

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Impact due to Ability to Redesign Business Processes

Equation: *WITH LOOKUP (Ability to Redesign Business Processes [(0, 0)-(100, 10)],(10, 1.1), (20, 1.05), (30, 1),(40, 0.98),(50, 0.95),(60, 0.9),(70, 0.85),(80, 0.8),(90, 0.75),(100, 0.7))*
 Range: [0.5 to 1.5]

Management Commitment

Equation: *CONSTANT*
 Range: [0.5 to 1.5]

Adjusted Costs for Process Analysis

Equation: *Planned Costs for Process Analysis * End User Fears * Process Complexity*

Adjusted Costs for Process Modeling

Equation: *Planned Costs for Process Modeling * Process Complexity*

Planned Costs for Process Analysis

Equation: *CONSTANT*

Planned Costs for Process Modeling

Equation: *CONSTANT*

Process Complexity

Equation: *CONSTANT*
 Range: [0.5 to 1.5]

Process Knowledge

Equation: *Process Knowledge Growth*
 Range: [0 to 100]
 Initial Value: [CONSTANT]

Process Knowledge Growth

Equation: *(Process Knowledge/100) + (Domain Knowledge/100)*

SECTION 6.3.2 - PROCESS MODELING COSTS:**Basic Process Knowledge Growth**

Equation: *CONSTANT*

Cost Rate

Equation: *Planned Modeling Costs * Impact due to Process Size * Process Complexity * Process Knowledge*

Impact due to Process Size

Equation: *WITH LOOKUP (Process Size, [(0, 0)-(1000, 10)], (0, 1),(10, 1.01),(30, 1.03),(50, 1.05), (100, 1.1),(200, 1.15),(300, 1.25),(500, 1.4),(1000, 1.5))*

Knowledge Growth Rate

Equation: *Basic Process Knowledge Growth + (Process Knowledge/100)*

Number of ArcsEquation: *CONSTANT***Number of Connectors**Equation: *CONSTANT***Number of End Events**Equation: *CONSTANT***Number of Events**Equation: *CONSTANT***Number of Functions**Equation: *CONSTANT***Number of Start Events**Equation: *CONSTANT***Planned Modeling Costs**Equation: *CONSTANT***Process Complexity**Equation: *CONSTANT*

Range: [0.5 to 1.5]

Process KnowledgeEquation: *Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [10]

Process Modeling CostsEquation: *Cost Rate*

Initial Value: [0]

Process SizeEquation: *Number of Arcs + Number of Connectors + Number of End Events + Number of Events + Number of Functions + Number of Start Events***SECTION 6.3.3 - REQUIREMENTS DEFINITION COSTS:****Analyzed Requirements**Equation: *Completion Rate - Documentation Rate*

Initial Value: [0]

Basic Comprehension RateEquation: *CONSTANT***Completion Rate**

Equation: *Basic Comprehension Rate*

Costs for Requirements Management Tool

Equation: *CONSTANT*

Costs for Test Tool

Equation: *CONSTANT*

Cost Rate

Equation: *Adjusted Process Analysis Costs + Requirements Documentation Costs + Costs for Requirements Management Tool*

Documentation Costs per Requirement

Equation: *CONSTANT*

Documentation Rate

Equation: *Relevance Quote*

Emotional Resistance of End Users

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Adjusted Process Analysis Costs

Equation: *Planned Process Analysis Efforts * Emotional Resistance of End Users * Process Complexity * Process Fragmentation * (Process Knowledge/100)*

Planned Process Analysis Costs

Equation: *CONSTANT*

Process Complexity

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Process Fragmentation

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Process Knowledge Growth Rate

Equation: *(Process Knowledge/100) * (Analyzed Requirements/100)*

Process Knowledge

Equation: *Process Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [20]

Relevance Rate

Equation: *CONSTANT*

Requirements Definition Costs

Equation: *Cost Rate*

Initial Value: [0]

Requirements Documentation Costs

Equation: *Requirements to be Documented* * *Costs per Requirement*

Requirements Test Costs

Equation: *Test Cost Rate*

Initial Value: [0]

Requirements to be Documented

Equation: *Documentation Rate*

Initial Value: [0]

Test Cost Rate

Equation: (*Requirements to be Documented* * *Test Costs per Requirement*) + *Costs for Test Tool*

Test Costs per Requirement

Equation: *CONSTANT*

SECTION 6.3.4 - PROCESS IMPLEMENTATION COSTS:

Cost Rate

Equation: *Data Modeling Costs* + *Miscellaneous Costs* + *Adjusted Form Design Costs* + *Adjusted Process Modeling Costs* + *Adjusted User/Role Management Costs* + *Test Costs*

Data Modeling Costs

Equation: *CONSTANT*

Domain Knowledge

Equation: *Domain Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [10]

Domain Knowledge Growth Rate

Equation: *Domain Knowledge/100*

Experiences in using Process Management Platform

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Miscellaneous Costs

Equation: *CONSTANT*

Adjusted Form Design Costs

Equation: *Planned Form Design Costs* * *Experiences in using Process Management Platform* * *Quality of Product Documentation* * *Technical Maturity of Process Management Platform* * *Usability of Process Management Platform*

Adjusted Process Modeling Costs

Equation: $\text{Planned Process Modeling Costs} * (\text{Domain Knowledge}/100) * \text{Process Complexity} * (\text{Process Knowledge}/100)$

Adjusted User/Role Management Costs

Equation: $\text{Planned User/Role Management Costs} * (\text{Process Knowledge}/100)$

Planned Form Design Costs

Equation: *CONSTANT*

Planned Process Modeling Costs

Equation: *CONSTANT*

Planned User/Role Management Costs

Equation: *CONSTANT*

Process Complexity

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Process Implementation Costs

Equation: *Cost Rate*

Initial Value: [0]

Process Knowledge

Equation: *Process Knowledge Growth Rate*

Range: [0 to 10]

Initial Value: [10]

Process Knowledge Growth Rate

Equation: $\text{Process Knowledge}/100$

Quality of Product Documentation

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Technical Maturity of Process Management Platform

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Test Costs

Equation: *CONSTANT*

Usability of Process Management Platform

Equation: *CONSTANT*

SECTION 6.3.4 - PROCESS ADAPTATION COSTS:**Cost Rate**

Equation: *Data Modeling Costs + Miscellaneous Costs + Adjusted Form Design Costs + Adjusted Process Modeling Costs + Adjusted User/Role Management Costs + Test Costs*

Data Modeling Costs

Equation: *CONSTANT*

Domain Knowledge

Equation: *Domain Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [10]

Domain Knowledge Growth Rate

Equation: *Domain Knowledge/100*

Experiences in using Process Management Platform

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Miscellaneous Costs

Equation: *CONSTANT*

Adjusted Form Design Costs

Equation: *Planned Form Design Costs * Experiences in using Process Management Platform * Quality of Product Documentation * Technical Maturity of Process Management Platform * Usability of Process Management Platform*

Adjusted Process Modeling Costs

Equation: *Planned Process Modeling Costs * (Domain Knowledge/100) * Process Complexity * (Process Knowledge/100)*

Adjusted User/Role Management Costs

Equation: *Planned User/Role Management Costs * (Process Knowledge/100)*

Planned Form Design Costs

Equation: *CONSTANT*

Planned Process Modeling Costs

Equation: *CONSTANT*

Planned User/Role Management Costs

Equation: *CONSTANT*

Process Complexity

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Process EvolutionEquation: *CONSTANT*

Range: [0.5 to 1.5]

Process Implementation CostsEquation: *Cost Rate*

Initial Value: [0]

Process KnowledgeEquation: *Process Knowledge Growth Rate*

Range: [0 to 10]

Initial Value: [10]

Process Knowledge Growth RateEquation: *Process Knowledge/100***Quality of Product Documentation**Equation: *CONSTANT*

Range: [0.5 to 1.5]

Technical Maturity of Process Management PlatformEquation: *CONSTANT*

Range: [0.5 to 1.5]

Test CostsEquation: *CONSTANT***Usability of Process Management Platform**Equation: *CONSTANT***Secondary Evaluation Patterns****SECTION 6.4.1 - END USER FEARS:****Ability Reduction**Equation: *Ability to Acquire Process Knowledge/100***Ability Reduction (to Acquire Process Knowledge)**Equation: *Emotional Resistance/100***Ability to Acquire Process Knowledge**Equation: *Ability Reduction (to Acquire Process Knowledge)*

Range: [0 to 100]

Initial Value: [100]

Ability to Redesign Business ProcessesEquation: *Ability Reduction*

Range: [0 to 100]

Initial Value: [100]

Changes of Social Clues and Interactions

Equation: *CONSTANT*

Range: [0.5 to 1.5]

Communication Growth

Equation: *End User Fears/100*

Communication with End Users

Equation: *Communication Growth*

Initial Value: [60]

Emotional Resistance

Equation: *Resistance Growth Rate*

Range: [0 to 100]

Initial Value: [0]

End User Fears

Equation: *Fear Growth Rate - Fear Reduction Rate*

Range: [0 to 100]

Initial Value: [80]

Fear Growth Rate

Equation: *(Changes of Social Clues and Interactions * Work Profile Change) * (1-(Ability to Redesign Business Processes/100))*

Fear Reduction Rate

Equation: *Communication with End Users/100*

Resistance Growth Rate

Equation: *End User Fears/100*

Work Profile Change

Equation: *CONSTANT*

Range: [0.5 to 1.5]

SECTION 6.4.2 - PROCESS KNOWLEDGE:

Ability Rate

Equation: *Emotional Resistance of End Users * Process Complexity * Process Fragmentation * (Process Knowledge/100) * (Domain Knowledge/100)*

Ability to Acquire Process Knowledge

Equation: *Ability Rate*

Range: [0 to 100]

Initial Value: [CONSTANT]

Domain KnowledgeEquation: *Domain Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [CONSTANT]

Domain Knowledge Growth RateEquation: $(\text{Domain Knowledge}/100) * (\text{Process Knowledge}/100)$ **Emotional Resistance of End Users**Equation: *CONSTANT*

Range: [0.5 to 1.5]

Process ComplexityEquation: *CONSTANT*

Range: [0.5 to 1.5]

Process FragmentationEquation: *CONSTANT*

Range: [0.5 to 1.5]

Process KnowledgeEquation: *Process Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [CONSTANT]

Process Knowledge Growth RateEquation: $(\text{Domain Knowledge}/100) * (\text{Process Knowledge}/100) * (\text{Ability to acquire Process Knowledge}/100)$ **SECTION 6.4.3 - DOMAIN KNOWLEDGE:****Basic Domain Knowledge Growth**Equation: *CONSTANT***Basic Experience Growth**Equation: *CONSTANT***Basic Process Knowledge Growth**Equation: *CONSTANT***Domain Complexity**Equation: *CONSTANT*

Range: [0.5 to 1.5]

Domain KnowledgeEquation: *Domain Knowledge Growth Rate*

Range: [0 to 100]

Initial Value: [CONSTANT]

Domain Knowledge Growth Rate

Equation: $Basic\ Domain\ Knowledge\ Growth + (Domain\ Complexity * (Domain\ Knowledge/100) * (Experience/100) * (Process\ Knowledge/100))$

Experience

Equation: $Experience\ Growth\ Rate$

Range: [0 to 100]

Initial Value: [CONSTANT]

Experience Growth Rate

Equation: $Basic\ Experience\ Growth + (Domain\ Knowledge/100)$

Process Knowledge

Equation: $Process\ Knowledge\ Growth\ Rate$

Range: [0 to 100]

Initial Value: [CONSTANT]

Process Knowledge Growth Rate

Equation: $Basic\ Process\ Knowledge\ Growth + ((Domain\ Knowledge/100) * (Experience/100) * (Process\ Knowledge/100))$

SECTION 6.4.4 - PROCESS EVOLUTION:**Compatibility with Customers**

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Compatibility with Suppliers

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Entry of Competitors

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Need for Compliance with Regulations and Laws

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Power of Buyers

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Power of Suppliers

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Management Order

Equation: $CONSTANT$

Range: [0.5 to 1.5]

Market Pressure

Equation: *Entry of Competitors * Power of Buyers * Power of Suppliers * Rivalry among Market Players * Threat of Substitutes*

Need for Process Optimization

Equation: *CONSTANT*
Range: [0.5 to 1.5]

Process Evolution

Equation: *Compatibility with Customers * Compatibility with Suppliers * Management Order * Market Pressure * Need for Compliance with Regulations and Laws * Need for Process Optimization * User Acceptance*

Rivalry among Market Players

Equation: *CONSTANT*
Range: [0.5 to 1.5]

Threat of Substitutes

Equation: *CONSTANT*
Range: [0.5 to 1.5]

User Acceptance

Equation: *CONSTANT*
Range: [0.5 to 1.5]

SECTION 6.4.5 - PROCESS COMPLEXITY:

Control-Flow Complexity

Equation: *CONSTANT*
Range: [0.5 to 1.5]

Cycle Complexity

Equation: *CONSTANT*
Range: [0.5 to 1.5]

Join Complexity

Equation: *CONSTANT*
Range: [0.5 to 1.5]

Process Complexity/Process Model Complexity

Equation: $((Join\ Complexity * Control-Flow\ Complexity * Cycle\ Complexity) + Split-Join-Ratio)/2$

Split-Join-Ratio

Equation: $Join\ Complexity/Control-Flow\ Complexity$

SECTION 6.4.6 - PROCESS MATURITY:**Basic Project Management**Equation: *Organization-specific Calculation*

Range: [1 to 5]

Causal Analysis and ResolutionEquation: *CONSTANT*

Range: [1 to 5]

Configuration ManagementEquation: *CONSTANT*

Range: [1 to 5]

Continuous Process ImprovementEquation: *Organization-specific Calculation***Decision Analysis and Resolution**Equation: *CONSTANT*

Range: [1 to 5]

Integrated Product ManagementEquation: *CONSTANT*

Range: [1 to 5]

Integrated TeamingEquation: *CONSTANT*

Range: [1 to 5]

Measurement and AnalysisEquation: *CONSTANT*

Range: [1 to 5]

Organizational Innovation and DevelopmentEquation: *CONSTANT*

Range: [1 to 5]

Organizational Process DefinitionEquation: *CONSTANT*

Range: [0 to 5]

Organizational Process FocusEquation: *CONSTANT*

Range: [1 to 5]

Organizational TrainingEquation: *CONSTANT*

Range: [1 to 5]

Process Maturity

Equation: *Organization-specific Calculation*

Process Standardization

Equation: *Organization-specific Calculation*

Range: [1 to 5]

Project Management

Equation: *CONSTANT*

Range: [1 to 5]

Project Monitoring and Control

Equation: *CONSTANT*

Range: [1 to 5]

Product and Process Quality Assurance

Equation: *CONSTANT*

Range: [1 to 5]

Product Integration

Equation: *CONSTANT*

Range: [1 to 5]

Quantitative Management

Equation: *Organization-specific Calculation*

Range: [1 to 5]

Quantitative Process Management

Equation: *CONSTANT*

Range: [1 to 5]

Requirements Development

Equation: *CONSTANT*

Range: [1 to 5]

Requirements Management

Equation: *CONSTANT*

Range: [1 to 5]

Risk Management

Equation: *CONSTANT*

Range: [1 to 5]

Software Quality Management

Equation: *CONSTANT*

Range: [1 to 5]

Supplier Agreement Management

Equation: *CONSTANT*

Range: [1 to 5]

Technical Solution

Equation: *CONSTANT*

Range: [1 to 5]

Validation

Equation: *CONSTANT*

Range: [1 to 5]

Verification

Equation: *CONSTANT*

Range: [1 to 5]

SECTION 6.4.7 - WORK PROFILE CHANGE:**Autonomy After PAIS**

Equation: *CONSTANT*

Autonomy Before PAIS

Equation: *1*

Change of Autonomy

Equation: *Autonomy After PAIS/Autonomy Before PAIS*

Change of Feedback from the Job

Equation: *Feedback After PAIS/Feedback Before PAIS*

Change of Skill Variety

Equation: *Skill Variety After PAIS/Skill Variety Before PAIS*

Change of Task Identity

Equation: *Task Identify After PAIS/Task Identify Before PAIS*

Change of Task Significance

Equation: *Task Significance After PAIS/Task Significance Before PAIS*

Feedback After PAIS

Equation: *CONSTANT*

Feedback Before PAIS

Equation: *1*

Skill Variety After PAIS

Equation: *CONSTANT*

Range: [0 to 100]

Skill Variety Before PAIS

Equation: *1*

Task Identity After PAISEquation: *CONSTANT*

Range: [0 to 100]

Task Identity Before PAISEquation: *1***Task Significance After PAIS**Equation: *CONSTANT***Task Significance Before PAIS**Equation: *1***Work Profile Change**Equation: $((\text{Change of Autonomy} + \text{Change of Feedback from the Job} + \text{Change of Skill Variety} + \text{Change of Task Identity} + \text{Change of Task Significance})/5)/100$

Range: [0 to 5]

Appendix C

Simulation Models from Chapter 11

Process Management Costs

THE SIMULATION MODEL:

Ability Growth

Equation: $1 + ((\text{Domain Knowledge}/100) * \text{Management Commitment} * (\text{Process Knowledge}/100))$

Range: [1;5]

Ability to acquire Process Knowledge

Equation: $\text{Basic Ability to acquire Process Knowledge} * \text{Emotional Resistance of End Users}$

Range: [0.5;1.5]

Ability Reduction

Equation: $1 + \text{Emotional Resistance of End Users}$

Range: [1;5]

Ability to (re)design Business Processes

Equation: $\text{Ability Growth} - \text{Ability Reduction}$

Range: [0;100]

Initial Value: [100]

Autonomy After PAIS Introduction

Equation: 80

Autonomy Before PAIS Introduction

Equation: 1

Basic Ability to acquire Process Knowledge

Equation: 1.2

Range: [0.5;1.5]

Basic Level of Communication

Equation: 0.7

Range: [0.5;1.5]

Basic Emotional Resistance of End Users

Equation: *1.2*

Range: [0.5;1.5]

Business Process Redesign Costs

Equation: *Cost Growth*

Initial Value: [0]

Change of Autonomy

Equation: *Autonomy After PAIS/Autonomy Before PAIS*

Change of Feedback from the Job

Equation: *Feedback After PAIS/Feedback Before PAIS*

Change of Skill Variety

Equation: *Skill Variety After PAIS/Skill Variety Before PAIS*

Change of Task Identity

Equation: *Task Identify After PAIS/Task Identify Before PAIS*

Change of Task Significance

Equation: *Task Significance After PAIS/Task Significance Before PAIS*

Communication

Equation: *Basic Communication * Emotional Resistance of End Users*

Range: [0.5;1.5]

Cost Growth

Equation: *(Adjusted Costs for Process Analysis + Adjusted Costs for Process Modeling) * (1 - ("Ability to (re)design Business Processes"/100))*

Domain Knowledge

Equation: *Domain Knowledge Growth*

Range: [0;100]

Initial Value: [40]

Domain Knowledge Growth

Equation: *(Process Knowledge/100) * ((Domain Knowledge/100) * 1.2)*

Emotional Resistance of End Users

Equation: *(Basic Emotional Resistance of End Users) * (End User Fears/100)*

Range: [0.5;1.5]

End User Fears

Equation: *Fear Growth - Fear Reduction*

Range: [0;100]

Initial Value: [10]

Fear GrowthEquation: *Impact due to Changes concerning Social Clue and Interactions* **Impact due to Work Profile Change* * (*End User Fears/100*)

Range: [0.5;1.5]

Fear ReductionEquation: *Communication*

Range: [0.5;1.5]

Feedback After PAIS IntroductionEquation: *40***Feedback Before PAIS Introduction**Equation: *1***Impact due to Changes concerning Social Clue and Interactions**Equation: *1.3*

Range: [0.5;1.5]

Impact due to Work Profile ChangeEquation: *WITH LOOKUP (Work Profile Change, [(0, 0)-(1.6, 1.6)], (0, 0.5),(0.1, 0.8),(0.2, 1),(0.3, 1.1),(0.4, 1.15), (0.5, 1.2),(0.6, 1.25),(0.7, 1.3),(0.8, 1.35),(0.9, 1.4),(1, 1.48))*

Range: [0.5;1.5]

Management CommitmentEquation: *1.4*

Range: [0.5;1.5]

Adjusted Costs for Process AnalysisEquation: *Planned Costs for Process Analysis* * *Process Complexity* **((End User Fears/100)+1)***Adjusted Costs for Process Modeling**Equation: *Planned Costs for Process Modeling* * *Process Complexity***Planned Costs for Process Analysis**Equation: *2000***Planned Costs for Process Modeling**Equation: *1000***Process Complexity**Equation: *1.05*

Range: [0.5;1.5]

Process KnowledgeEquation: *Process Knowledge Growth*

Range: [0;100]

Initial Value: [30]

Process Knowledge Growth

Equation: $((\text{Process Knowledge}/100) * 1.4) * (\text{Domain Knowledge}/100) * \text{Ability to acquire Process Knowledge}$
 Range: [0;1]

Skill Variety After PAIS Introduction

Equation: 20

Skill Variety Before PAIS Introduction

Equation: 1

Task Identify After PAIS Introduction

Equation: 30

Task Identify Before PAIS Introduction

Equation: 1

Task Significance After PAIS Introduction

Equation: 60

Task Significance Before PAIS Introduction

Equation: 1

Work Profile Change

Equation: $((\text{Change of Autonomy} + \text{Change of Feedback from the Job} + \text{Change of Skill Variety} + \text{Change of Task Identity} + \text{Change of Task Significance})/100)/5$

IT System Realization Costs

THE SIMULATION MODEL:

Basic Experience Growth

Equation: 5

Cost Rate

Equation: $\text{Data Modeling Costs} + \text{Miscellaneous Costs} + \text{Adjusted Form Design Costs} + \text{Adjusted Process Modeling Costs} + \text{Adjusted User/Role Management Costs} + \text{Test Costs}$

Data Modeling Costs

Equation: 250

Domain Knowledge

Equation: $\text{Domain Knowledge Growth Rate}$
 Range: [0;100]

Initial Value: [30]

Domain Knowledge Growth Rate

Equation: $((\text{Domain Knowledge}/100) * 1.4) * (\text{Process Knowledge}/100)$

Experience Growth

Equation: $(\text{Domain Knowledge}/100) * ((\text{Experiences in using Process Management Platform}/100) * 1.4) * (\text{Process Knowledge}/100) * \text{Basic Experience Growth}$

Experiences in using Process Management Platform

Equation: *Experience Growth*

Range: [0;100]

Initial Value: [30]

Miscellaneous Costs

Equation: 200

Adjusted Form Design Costs

Equation: $\text{Planned Form Design Costs} * (\text{Experiences in using Process Management Platform}/100) * \text{Quality of Product Documentation} * \text{Technical Maturity of Process Management Platform} * \text{Usability of Process Management Platform}$

Adjusted Process Modeling Costs

Equation: $\text{Planned Process Modeling Costs} * (\text{Domain Knowledge}/100) * \text{Process Complexity}$

Adjusted User/Role Management Costs

Equation: $\text{Planned User/Role Management Costs} * \text{Domain Knowledge}$

Planned Form Design Costs

Equation: 500

Planned Process Modeling Costs

Equation: 750

Planned User/Role Management Costs

Equation: 150

Process Complexity

Equation: 1.15

Range: [0.5;1.5]

Process Implementation Costs

Equation: *Cost Rate*

Initial Value: [0]

Process Knowledge

Equation: *Process Knowledge Growth Rate*

Range: [0;100]

Initial Value: [30]

Process Knowledge Growth Rate

Equation: $((\text{Process Knowledge}/100) * 1.4) * (\text{Domain Knowledge}/100)$

Quality of Product Documentation

Equation: 1.05

Range: [0.5;1.5]

Technical Maturity of Process Management Platform

Equation: 1.1

Range: [0.5;1.5]

Test Costs

Equation: 200

Usability of Process Management Platform

Equation: 1.1

Range: [0.5;1.5]

Specification and Test Costs

THE SIMULATION MODEL:**Ability to acquire Process Knowledge**

Equation: $\text{Simplicity Growth Rate} - \text{Simplicity Reduction Rate}$

Range: [0;100]

Initial Value: [90]

Analyzed Requirements

Equation: $\text{Completion Rate} - \text{Documentation Rate}$

Initial Value: [0]

Basic Comprehension Rate

Equation: 3

Communication with End Users

Equation: 1.4

Range: [0.5;1.5]

Completion Rate

Equation: $\text{Basic Comprehension Rate}$

Costs for Requirements Management Tool

Equation: 150

Costs per Requirement

Equation: 50

Cost Rate

Equation: *Adjusted Process Analysis Costs + Requirements Documentation Costs + Costs for Requirements Management Tool*

Documentation Rate

Equation: *Relevance Quote*

Domain Knowledge

Equation: *Domain Knowledge Growth Rate*

Range: [0;100]

Initial Value: [30]

Domain Knowledge Growth Rate

Equation: $(\text{Domain Knowledge}/100) * (\text{Process Knowledge}/100)$

Emotional Resistance of End Users

Equation: *1.3*

Range: [0.5;1.5]

Adjusted Process Analysis Costs

Equation: *Planned Process Analysis Efforts * Emotional Resistance of End Users * Process Complexity * Process Fragmentation*

Planned Process Analysis Costs

Equation: *2000*

Process Complexity

Equation: *1.1*

Range: [0.5;1.5]

Process Fragmentation

Equation: *1*

Range: [0.5;1.5]

Process Knowledge

Equation: $\text{Process Knowledge} + (\text{Process Knowledge Growth Rate})$

Range: [0;100]

Initial Value: [60]

Process Knowledge Growth Rate

Equation: $(\text{Process Knowledge}/100) * (\text{Domain Knowledge}/100) * (\text{Ability to acquire Process Knowledge}/100)$

Relevance Quote

Equation: *2*

Requirements Definition Costs

Equation: *Cost Rate*

Initial Value: [0]

Requirements Documentation Costs

Equation: *Requirements to be documented* * *Costs per Requirement*

Requirements Test Costs

Equation: *Requirements Test Costs* + (*Test Cost Rate*)

Initial Value: [0]

Requirements to be Documented

Equation: *Documentation Rate*

Initial Value: [0]

Simplicity Growth Rate

Equation: *Communication**(*Domain Knowledge*/100)

Simplicity Reduction Rate

Equation: *Emotional Resistance of End Users* * *Process Complexity*

Test Cost Rate

Equation: ((*Requirements to be Documented* * *Test Costs per Requirement*) * *Process Complexity*) + *Test Tool Costs*

Test Costs per Requirement

Equation: 20

Test Tool Costs

Equation: 40

Bibliography

- [1] A. F. Abate, A. Esposito, N. Grieco, and G. Nota. *Workflow Performance Evaluation through WPQL*. Proc. 14th Int'l. Conf. Software Engineering and Knowledge Engineering (SEKE '02), pp.489-495, 2002.
- [2] J. Adams and L. Juleff. *Managerial Economics for Decision Making*. Palgrave Macmillan, 2003.
- [3] R. Aiello. *Workflow Performance Evaluation*. PhD Thesis, University of Salerno, Italy, 2004.
- [4] C. Alexander. *The Timeless Way of Building*. Oxford University Press, 1979.
- [5] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language*. Oxford University Press, 1979.
- [6] M. Amberg and M. Hirschmeier. *Reframing of Economic (e) Valuations for Innovative Technologies*. Proc. 7th Conf. on Synergetics, Cybernetics and Informatics (SCI '03), 2003.
- [7] J. L. Andresen. *A Framework for Selecting an IT Evaluation Method - In the Context of Construction*. PhD Thesis, Technical University of Denmark, Lyngby, Denmark, 2001.
- [8] Y. L. Antonucci. *Using Workflow Technologies to Improve Organizational Competitiveness*. Int'l. Journal of Management, 14(1), pp.117-126, 1997.
- [9] Pallas Athena. *Case Handling with FLOWer: Beyond Workflow*. 2002.
- [10] D. Bachmann, J. Elfrink, and G. Vazzana. *E-mail and Snail Mail Face Off in Rematch*. Journal of Market Research Society, 11(4), pp.11-15, 2000.
- [11] Z. Baida, H. de Bruin, and J. Gordijin. *e-Business Cases Assessment: From Business Value to System Feasibility*. Int'l. Journal of Web Engineering & Technology, 1(1), pp.127-144, 2003.
- [12] J. Y. Bakos and B. R. Nault. *Ownership and Investment in Electronic Networks*. Information Systems Research, 8(4), pp.321-341, 1997.
- [13] Y. Barlas. *Multiple Tests for Validations of System Dynamics Type of Simulation Models*. European Journal of Operational Research, 42(1), pp.59-87, 1989.
- [14] Y. Barlas. *Formal Aspects of Model Validity and Validation in System Dynamics*. System Dynamics Review, 13(3), pp.183-210, 1996.
- [15] J. J. Baroudi, M. H. Olson, and B. Ives. *An Empirical Study of the Impact of User Involvement on System Usage and Information Satisfaction*. Communications of the ACM, 29(3), pp.232-239, 1986.
- [16] M. D. O. Barros, C. M. L. Werner, and G. H. Travassos. *Evaluating the Use of System Dynamics Models in Software Project Management*. Proc. 20th Int'l. System Dynamics Conference, 2002.

- [17] V. R. Basili, R. W. Selby, and D. H. Hutchens. *Experimentation in Software Engineering*. IEEE Transactions in Software Engineering, 12(7), pp.733-743, 1986.
- [18] S. Bassil. *Workflow Technology for Complex Socio-Technical Systems*. PhD Thesis, University of Montreal, Canada, 2005.
- [19] J. Becker, C. v. Uthmann, M. zur Muehlen, and M. Rosemann. *Identifying the Workflow Potential of Business Processes*. Proc. 32nd Hawaii Int'l. Conf. on System Sciences (HICSS '99), 1999.
- [20] M. Benaroch. *Managing Information Technology Investment Risk: A Real Options Perspective*. Journal of Management Information Systems, 19(2), pp.43-84, 2002.
- [21] M. Benaroch and R. J. Kaufman. *A Case Study for Using Real Options Pricing Analysis to Evaluate Information Technology Project Investments*. Information Systems Research, 10(1), pp.70-86, 1999.
- [22] U. Bestfleisch, J. Herbst, and M. Reichert. *Requirements for the Workflow-based Support of Release Management Processes in the Automotive Sector*. Proc. 12th European Concurrent Engineering Conference (ECEC '05), pp.130-134, 2005.
- [23] R. Bobrik, T. Bauer, and M. Reichert. *Proviado - Personalized and Configurable Visualizations of Business Processes*. Proc. 7th Int'l. Conf. on Electronic Commerce and Web Technologies (EC-Web '06), pp.61-71, 2006.
- [24] R. Bobrik, M. Reichert, and T. Bauer. *Requirements for the Visualization of System-Spanning Business Processes*. Proc. 1st Int'l. Workshop on Business Process Monitoring and Performance Management (BPMPM '05), 2005.
- [25] D. Boddy, A. Boonstra, and G. Kennedy. *Managing Information Systems: An Organisational Perspective*. Prentice Hall (2nd Edition), 2005.
- [26] L. Bodenstaff, A. Wombacher, M. Reichert, and R. Wieringa. *Monitoring Collaboration from a Value Perspective*. Proc. 2007 IEEE Int'l. Conf. on Digital Ecosystems and Technologies (DEST '07), pp.134-140, 2007.
- [27] B. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.
- [28] B. Boehm. *Value-Based Software Engineering*. ACM SIGSOFT Software Engineering Notes, 28(2), pp.1-12, 2003.
- [29] B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece. *Software Cost Estimation with Cocomo 2*. Prentice Hall, 2000.
- [30] B. Boehm and L. G. Huang. *Value-Based Software Engineering - A Case Study*. IEEE Computer, 36(3), pp.33-41, 2003.
- [31] B. Boehm and L. G. Huang. *Value-Based Software Engineering: Reinventing "Earned Value" Monitoring and Control*. ACM SIGSOFT Software Engineering Notes, 28(2), pp.29-36, 2003.
- [32] B. Boehm and K. Sullivan. *Software Economics: A Roadmap*. Proc. 22nd Int'l. Conf. on Software Engineering (ICSE '00), pp.319-343, 2000.
- [33] D. F. Boyd and H. S. Krasnow. *Economic Evaluations of Management Information Systems*. IBM Systems Journal, 2(1), pp.2-23, 1963.
- [34] K.-H. Brassel, M. Möhring, E. Schumacher, and K. G. Troitzsch. *Can Agents Cover All the World? Simulating Social Phenomena*, LNEMS 456, pp.55-72, 1997.

- [35] L. Briand. *Empirical Studies of Object-Oriented Artifacts, Methods, and Processes*. Empirical Software Engineering, 4(4), pp.387-404, 1999.
- [36] E. Brynjolfsson. *The Productivity Paradox of Information Technology*. Communications of the ACM, 36(12), pp.66-77, 1993.
- [37] J. Cardoso. *Control-flow Complexity Measurement of Processes and Weyuker's Properties*. Proc. Int'l. Enformatika Conference, Volume 8, pp.213-218, 2005.
- [38] J. Cardoso, J. Mendling, G. Neumann, and H. Reijers. *A Discourse on Complexity of Process Models*. Proc. Int'l. Workshop on Business Process Design (BPI '06), LNCS 4103, pp.115-126, 2006.
- [39] N. G. Carr. *IT doesn't Matter*. Harvard Business Review (HBR), Volume 5, 2003.
- [40] N. G. Carr. *Does IT matter?* Harvard Business Press, 2004.
- [41] J. S. Carson. *Model Verification and Validation*. Proc. 2002 Winter Conference, pp.52-58, 2002.
- [42] F. Casati, U. Dayal, M. Sayal, and M.-Ch. Shan. *Business Process Intelligence*. HP Technical Report, HPL-2002-119, 2002.
- [43] J. Chang. *Envisioning the Process-Centric Enterprise*. eai Journal, August 2002, pp.30-33, 2002.
- [44] JFree Chart. *Object Refinery Limited*. <http://www.jfree.org/jfreechart/>, 2007.
- [45] S. Choenni, R. Bakker, and W. Baets. *On the Evaluation of Workflow Systems in Business Processes*. Electronic Journal of Information Systems Evaluation (EJISE), 6(2), 2003.
- [46] S. Chulani, B. Boehm, and B. Steece. *Bayesian Analysis of Empirical Software Engineering Cost Models*. IEEE Transactions on Software Engineering, 25(4), pp.573-583, 1999.
- [47] E. K. Clemons. *Evaluation of Strategic Investments in Information Technology*. Communications of the ACM, 34(1), pp.22-36, 1991.
- [48] CMMi. *Capability Maturity Model Integration*. Software Engineering Institute: www.sei.cmu.edu/cmmi/, 2006.
- [49] R. G. Coyle. *Management System Dynamics*. John Wiley and Sons, 1977.
- [50] R. G. Coyle. *System Dynamics Modeling: A Practical Approach*. Chapman and Hall, 1996.
- [51] S. Cronholm and G. Goldkuhl. *Strategies for Information Systems Evaluation - Six Generic Types*. Electronic Journal of Information Systems Evaluation (EJISE), 6(2), 2003.
- [52] M. Daneva. *ERP Requirements Engineering Practice: Lessons Learned*. IEEE Software, 21(2), pp.26-33, 2004.
- [53] M. Daneva and R. Wieringa. *A Conceptual Framework for Research in Cross-organizational ERP Cost Estimation*. Proc. Workshop on Requirements Engineering and Project Management in Software Projects (PROMan), 2005.
- [54] T. H. Davenport. *Mission Critical: Realizing the Promise of Enterprise Systems*. Harvard Business School Press, 2000.
- [55] J. Dehnert and W.M.P. van der Aalst. *Bridging the Gap between Business Models and Workflow Specification*. Int'l. Journal of Cooperative Information Systems, 13(3), pp.289-332, 2004.

- [56] Tom DeMarco. *Controlling Software Projects: Management, Measurement and Estimation*. Yourdon Press, 1983.
- [57] Derby. *Apache Foundation*. <http://db.apache.org/derby/>, 2007.
- [58] C. Devnani-Chulani. *Bayesian Analysis of Software Cost and Quality Models*. PhD Thesis, University of Southern California, 2001.
- [59] M. Dumas, W. M. P. van der Aalst, and A. H. ter Hofstede. *Process-aware Information Systems: Bridging People and Software through Process Technology*. Wiley, 2005.
- [60] R. J. Eberlein and J. H. Hines. *Molecules for Modelers*. Proc. 14th Int'l. System Dynamics Conference, 1996.
- [61] K. M. Eisenhardt. *Building Theories from Case Study Research*. Academy of Management Review, 14(4), pp.532-550, 1989.
- [62] D. J. Falconer and R. A. Hodgett. *Why Executives Don't Respond to Your Survey*. Proc. 10th Australasian Conf. on Information Systems, pp.279-285, 1999.
- [63] J. Forrester and P. Senge. *Tests for Building Confidence in S-D Models*. TIMS Studies in the Management Sciences, Volume 14, pp.208-228, 1980.
- [64] J. W. Forrester. *Industrial Dynamics*. Productivity Press, 1961.
- [65] D. Garmus and D. Herron. *Function Point Analysis*. Addison-Wesley, 2000.
- [66] J. Gebauer and F. Schober. *Information System Flexibility and the Cost Efficiency of Business Processes*. Journal of the Association for Information Systems, 7(3), pp.122-147, 2006.
- [67] K. H. Goh and R. J. Kauffman. *Towards a Theory of Value Latency for IT Investments*. Proc. 38th Hawaii Int'l. Conf. on System Sciences (HICSS '05), 2005.
- [68] J. Gordijn. *Value-based Requirements Engineering: Exploring Innovative e-Commerce Ideas*. PhD Thesis, Free University of Amsterdam, 2002.
- [69] J. Gordijn and H. Akkermans. *E3-value: Design and Evaluation of e-Business Models*. IEEE Intelligent Systems, 16(4), pp.11-17, 2001.
- [70] J. Gordijn and H. Akkermans. *Value-based Requirements Engineering: Exploring Innovative E-commerce Ideas*. Requirements Engineering Journal, 8(2), pp.114-134, 2003.
- [71] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M. Shan. *Business Process Intelligence*. Computers in Industry, 53(3), pp.321-343, 2004.
- [72] V. Gruhn and R. Laue. *Complexity Metrics for Business Process Models*. Proc. 9th Int'l. Conf. on Business Information Systems (BIS '06), 2006.
- [73] C. W. Günther, S. Rinderle, M. Reichert, and W. M. P. van der Aalst. *Change Mining in Adaptive Process Management Systems*. Proc. Int'l. Conf. on Cooperative Information Systems (CoopIS '06), pp.309-326, 2006.
- [74] R. J. Hackman and G. R. Oldham. *Development of the Job Diagnostic Survey*. Journal of Applied Psychology, 60(2), pp.159-170, 1975.
- [75] R. J. Hackman and G. R. Oldham. *Motivation through the Design of Work: Test of a Theory*. Organizational Behavior & Human Performance, 16(2), pp.250-279, 1976.
- [76] T. Haeberlein. *Understanding Software Acquisition through Modeling and Simulation*. PhD Thesis, University of Ulm, Germany, 2005.

- [77] M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, 1993.
- [78] J. Hares and D. Royle. *Measuring the Value of Information Technology*. Wiley, 1994.
- [79] P. Hawking, A. Stein, and S. Foster. *Revisiting ERP Systems: Benefit Realization*. Proc. 37nd Hawaii Int'l. Conf. on System Sciences (HICSS '04), 2004.
- [80] S. Horwitz. *The Economic Benefits of BPM*. eai Journal, June 2002, pp.37-39, 2002.
- [81] F. Houdek. *Empirisch basierte Qualitätsverbesserung - Systematischer Einsatz externer Experimente im Software Engineering*. PhD Thesis, University of Ulm, Germany, 1999.
- [82] A. Idri, T. M. Khoshgoftaar, and A. Abran. *Can Neural Networks be easily interpreted in Software Cost Estimation?* Proc. Int'l. Conf. on Fuzzy Systems (FUZZ-IEEE '02), pp.1162-1167, 2002.
- [83] A. Jedlitschka and D. Pfahl. *Reporting Guidelines for Controlled Experiments in Software Engineering*. Technical Report, ISERN-05-01, 2005.
- [84] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2002.
- [85] L. Jin, F. Casati, and A. van Moorsel. *Design of A Business Process Analyzer for Business Process Management*. Proc. Int'l. Symposium on Information Systems and Engineering, 2002.
- [86] G. Judge, W. Griffiths, and C. Hill. *Learning and Practicing Econometrics*. Wiley, 1993.
- [87] N. Juristo and A. M. Moreno. *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers, 2001.
- [88] V. Kartseva, J. Hulstijn, Y.-H. Tan, and J. Gordijn. *Towards Value-based Design Patterns for Inter-Organizational Control*. Proc. 19th Bled Electronic Commerce Conference "eValue", 2006.
- [89] R. Kennet and S. Zacks. *Modern Industrial Statistics: Design and Control of Quality and Reliability*. Brooks/Cole Publishing Company, 1998.
- [90] E. Kindler. *On the Semantics of EPCs: Resolving the Vicious Circle*. Data Knowledge Engineering, 56(1), pp.23-40, 2006.
- [91] B. A. Kitchenham. *Evaluating Software Engineering Methods and Tool: Part 10 - Designing and Running a Quantitative Case Study Methodology*. ACM SIGSOFT Software Engineering Notes, 23(3), pp.20-22, 1998.
- [92] B. A. Kitchenham. *Evaluating Software Engineering Methods and Tool: Part 11 - Analyzing Quantitative Case Studies*. ACM SIGSOFT Software Engineering Notes, 23(4), pp.18-20, 1998.
- [93] B. A. Kitchenham. *Evaluating Software Engineering Methods and Tool: Part 9 - Quantitative Case Study Methodology*. ACM SIGSOFT Software Engineering Notes, 23(1), pp.24-26, 1998.
- [94] B. A. Kitchenham and S. L. Pfleeger. *Principles of Survey Research: Part 2 - Designing a Survey*. ACM SIGSOFT Software Engineering Notes, 27(1), pp.18-20, 2002.
- [95] B. A. Kitchenham and S. L. Pfleeger. *Principles of Survey Research: Part 3 - Constructing a Survey Instrument*. ACM SIGSOFT Software Engineering Notes, 27(2), pp.20-24, 2002.

- [96] B. A. Kitchenham and S. L. Pfleeger. *Principles of Survey Research: Part 6 - Data Analysis*. ACM SIGSOFT Software Engineering Notes, 28(2), pp.24-27, 2002.
- [97] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, D. C. Hoaglin, K. E. Emam, and J. Rosenberg. *Preliminary Guidelines for Empirical Research in Software Engineering*. IEEE Transactions on Software Engineering, 28(8), pp.721-734, 2002.
- [98] B. A. Kitchenham, L. Pickard, and S. L. Pfleeger. *Case Studies for Method and Tool Evaluation*. IEEE Software, 12(4), pp.52-62, 1995.
- [99] N. Kleiner. *Can Business Process Changes Be Cheaper Implemented with Workflow Management Systems*. Proc. 2004 Information Resources Management Association Conference (IRMA '04), 2004.
- [100] N. Kleiner. *Verbesserungen des Workflow-Design-Prozesses*. PhD Thesis, University of Ulm, Germany, 2004.
- [101] E. Knippel and A. Schulz. *Lessons Learned from Implementing Configuration Management within Electrical/Electronic Development of an Automotive OEM*. Proc. 14th Int'l. Symp. Managing Complexity and Change (INCOSE '04), 2004.
- [102] P. Kueng. *The Effects of Workflow Systems on Organization: A Qualitative Study*. Business Process Management, LNCS 1806, pp. 301-316, 2000.
- [103] T. S. Kuhn. *The Structure of Scientific Revolutions*. B & T, 1996.
- [104] G. Lamla. *Investigation, Comparison and Evaluation of Approaches for Measuring Business Process Performance (in German)*. Master Thesis, University of Ulm, 2005.
- [105] A. Latva-Koivisto. *Finding a Complexity Measure for Business Process Models*. Research Report, Helsinki University of Technology, 44421w, Tf-N, 2001.
- [106] R. Lenz and M. Reichert. *IT Support for Healthcare Processes: Premises, Challenges, Perspectives*. Data and Knowledge Engineering, Volume 61, pp.39-58, 2007.
- [107] F. Leymann, D. Roller, and M.-T. Schmidt. *Web Services and Business Process Management*. IBM Systems Journal, 41(2), pp.198-211, 2002.
- [108] X. Li and J. D. Johnson. *Evaluate IT Investment Opportunities Using Real Options Theory*. Information Resources Management Journal, 15(3), pp.32-47, 2002.
- [109] M. Liehr. *A Platform for System Dynamics Modeling - Methodologies for the Use of Predefined Model Components*. Proc. 20th Int'l. System Dynamics Conference, 2002.
- [110] N. Lofts. *Process Visualization*. John Wiley, 2002.
- [111] C. M. Lott and H. D. Rombach. *Repeatable Software Engineering Experiments for Comparing Defect-Detection Techniques*. Empirical Software Engineering, 1(3), pp. 241-277, 1996.
- [112] T. Malone, K. Crowston, J. Lee, B. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C. Osborn, A. Bernstein, G. Herman, M. Klein, and E. O'Donnell. *Towards a Handbook of Organisational Processes*. Management Science, 45(3), pp.425-443, 1999.
- [113] P. Mandal and A. Gunasekaran. *Issues in Implementing ERP: A Case Study*. European Journal of Operational Research, Volume 146, pp.274-283, 2003.
- [114] F. W. McFarlan and I. L. McKenney. *Corporate Information Systems Management*. Richard D. Irwin Inc., 1983.

- [115] R. Mehta and E. Sivadas. *Comparing Response Rates and Response Content in Mail versus Electronic Mail Survey*. Journal of Market Research Society, 37(4), pp. 429-439, 1995.
- [116] L. Meineke. *Wirtschaftlichkeitsanalysen - Basis zur Durchführung von IT-Projekten*. IT Management, Volume 12, pp.12-17, 2003.
- [117] F. Melchert and R. Winter. *The Enabling Role of Information Technology for Business Performance Management: Decision Support in an Uncertain and Complex World*. Proc. 2004 IFIP Int'l. Conf. on Decision Support Systems (DSS '04), pp.535-546, 2004.
- [118] F. Melchert, R. Winter, and M. Klesse. *Aligning Process Automation and Business Intelligence to support Corporate Performance Management*. Proc. 2004 Americas Conf. on Information Systems (AMCIS '04), pp.53-63, 2004.
- [119] M. W. Mende, L. Brecht, and H. Österle. *Evaluating existing Information Systems from a Business Process Perspective*. Proc. 1994 Computer Personnel Research Conference on Reinventing IS, pp.289-296, 1994.
- [120] J. Mendling, M. Moser, G. Neumann, H. M. W. Verbeek, B. F. van Dongen, and W. M. P. van der Aalst. *Faulty EPCs in the SAP Reference Model*. Proc. 4th Int'l. Conf. on Business Process Management (BPM '06), LNCS 4102, pp. 451-457, 2006.
- [121] J. Mendling, M. Moser, G. Neumann, H. M. W. Verbeek, B. F. van Dongen, and W. M. P. van der Aalst. *A Quantitative Analysis of Faulty EPCs in the SAP Reference Model*. BPM Center Report, BPM-06-08, BPMcenter.org, 2006.
- [122] D. Müller, J. Herbst, M. Hammori, and M. Reichert. *IT Support for Release Management Processes in the Automotive Industry*. Proc. 4th Business Process Management (BPM '06), LNCS 4102, pp.368-377, 2006.
- [123] F. Müller. *Konzeption und Implementierung eines Werkzeugs zur Kosten-Nutzen-Analyse von Informationssystemen*. Master Thesis, University of Ulm, Germany, 2006.
- [124] G. Murphy, R. Walker, and E. Baniassad. *Evaluating emerging Software Development Techniques*. IEEE Transactions on Software Engineering, 25(4), pp.438-455, 1999.
- [125] K. E. Murphy and S. J. Simon. *Using Cost Benefit Analysis for Enterprise Resource Planning Project Evaluation: A Case for Including Intangibles*. Proc. 34th Hawaii Int'l. Conf. on System Sciences (HICSS '01), 2001.
- [126] B. Mutschler. *Kosten-Nutzen-Analysen zur Bewertung prozessorientierter Informationssysteme*. Proc. 2. GI Fachtagung IT-Controlling, 2005.
- [127] B. Mutschler and J. Bumiller. *Improving the Return-on-Investment of Product Data Management Systems Using Usability Engineering*. Proc. Product Data Technology Europe (PDT Europe '04), pp.215-222, 2004.
- [128] B. Mutschler and M. Reichert. *Aktuelles Schlagwort: Business Process Intelligence*. EMISA Forum, 26(1), pp.27-31, 2006.
- [129] B. Mutschler and M. Reichert. *A Survey on Evaluation Factors for Business Process Management Technology*. Technical Report TR-CTIT-06-63, Centre for Telematics and Information (CTIT), University of Twente, The Netherlands, 2006.
- [130] B. Mutschler and M. Reichert. *Exploring the Dynamic Costs of Process-aware Information Systems through Simulation*. Proc. Workshops and Doctoral Consortium at CAiSE '07, Volume 1, 12th Int'l. Workshop on Exploring Modeling Methods for Information Systems Analysis and Design (EMMSAD '07), pp.173-182, 2007.

- [131] B. Mutschler and M. Reichert. *Simulation Models for Analyzing the Dynamic Costs of Process-aware Information Systems*. Technical Report TR-CTIT-07-14, Centre for Telematics and Information (CTIT), University of Twente, The Netherlands, 2007.
- [132] B. Mutschler, M. Reichert, and J. Bumiller. *An Approach to Quantify the Costs of Business Process Intelligence*. Proc. Int'l. Workshop on Enterprise Modeling and Information Systems Architectures (EMISA '05), LNI P-75, pp.152-165, 2005.
- [133] B. Mutschler, M. Reichert, and J. Bumiller. *Towards Process-Aware Enterprise Software Environments*. Proc. 7th Int'l. Conf. on Enterprise Information Systems (ICEIS '05), pp.374-378, 2005.
- [134] B. Mutschler, M. Reichert, and J. Bumiller. *An Approach for Evaluating Workflow Management Systems from a Value-Based Perspective*. Proc. 10th IEEE Int'l. Conf. on Enterprise Computing (EDOC '06), pp.477-482, 2006.
- [135] B. Mutschler, M. Reichert, and J. Bumiller. *Designing an Economic-driven Evaluation Framework for Process-oriented Software Technologies*. Proc. 28th Int'l. Conf. on Software Engineering (ICSE '06), pp.885-888, 2006.
- [136] B. Mutschler, M. Reichert, and J. Bumiller. *Why Process-Oriented is Scarce: An Empirical Study of Process-oriented Information Systems in the Automotive Industry*. Proc. 10th IEEE Int'l. Conf. on Enterprise Computing (EDOC '06), pp.433-438, 2006.
- [137] B. Mutschler, M. Reichert, and J. Bumiller. *Unleashing the Effectiveness of Process-oriented Information Systems: Problem Analysis, Critical Success Factors and Implications*. IEEE Transactions on Systems, Man, and Cybernetics (SMC) - Part C: Applications & Reviews, 2008 (accepted for publication).
- [138] B. Mutschler, M. Reichert, and S. Rinderle. *Analyzing the Dynamic Cost Factors of Process-aware Information Systems: A Model-based Approach*. Proc. 19th Int'l. Conf. on Advanced Information Systems Engineering (CAiSE '07), pp.589-603, 2007.
- [139] B. Mutschler, M. Rijkpema, and M. Reichert. *Investigating Implemented Process Design: A Case Study on the Impact of Process-aware Information Systems on Core Job Dimensions*. Proc. Workshops and Doctoral Consortium at CAiSE '07, Volume 1, 8th Int'l. Workshop on Business Process Modeling, Development, and Support (BPMDS '07), pp.379-384, 2007.
- [140] B. Mutschler, B. Weber, and M. Reichert. *Workflow Management versus Case Handling - Results from a Controlled Software Experiment*. Proc. 23rd Annual ACM Symposium on Applied Computing (SAC '08), Special Track on Coordination Models, Languages, Architectures (CM Track), 2008 (accepted for publication).
- [141] B. Mutschler, Novica Zarvic, and M. Reichert. *A Survey on Economic-driven Evaluations of Information Technology*. Technical Report TR-CTIT-07-21, Centre for Telematics and Information (CTIT), University of Twente, The Netherlands, 2007.
- [142] G. J. Myers. *A controlled Experiment in Program Testing and Code Walkthroughs/Inspections*. Communications of the ACM, 21(9), pp. 760-768., 1978.
- [143] M. Myrvtveit. *Object-oriented Extensions to System Dynamics*. Proc. 18th Int'l. System Dynamics Conference, 2000.
- [144] D. P. Norton. *Stage by Stage - Letter to Management*. The Economics of Computing in the Advanced Stages, 1981.
- [145] J. Nunnally. *Psychometric Theory*. McGraw-Hill, 1978.

- [146] M. Oba, S. Onoda, and N. Komoda. *Evaluating the Quantitative Effects of Workflow Systems based on Real Cases*. Proc. 33rd Hawaii Int'l. Conf. on System Sciences (HICSS '00), 2000.
- [147] K. Ogata. *System Dynamics*. Prentice Hall, 2003.
- [148] A. Parkes. *Critical Success Factors in Workflow Implementation*. Proc. 6th Pacific Asia Conf. on Information Systems (PACIS '02), pp.363-380, 2002.
- [149] A. Parkes. *A Case Study of Workflow Implementation Success Factors*. Proc. 15th Australasian Conf. on Information Systems (ACIS '04), 2004.
- [150] S. L. Pfleeger. *Experimental Design and Analysis in Software Engineering*. Annals of Software Engineering, 1(1995), pp.219-253, 1995.
- [151] S. L. Pfleeger. *Experimental Design and Analysis in Software Engineering: Types of Experimental Designs*. ACM SIGSOFT Software Engineering Notes, 20(2), pp.14-16, 1995.
- [152] T. Pietsch. *Bewertung von Informations- und Kommunikationssystemen: Ein Vergleich betriebswirtschaftlicher Verfahren*. Erich Schmidt Verlag, 2003.
- [153] T. Pisello. *IT Value Chain Management - Maximizing the ROI from IT Investments: Performance Metrics and Management Methodologies Every IT Stakeholder Should Know*. Alinean, 2003.
- [154] M. E. Porter. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*. Free Press, 1998.
- [155] Powersim. *Powersim Software*. <http://www.powersim.com/>, 2007.
- [156] L. Prechelt. *Kontrollierte Experimente in der Softwaretechnik*. Springer, 2001.
- [157] L. Putnam and W. Meyers. *Measures for Excellence: Reliable Software on Time, Within Budget*. Yourdon Press, 1991.
- [158] A. Rapoport. *General System Theory: Essential Concepts and Applications*. Abacus Press, 1986.
- [159] P.W. Reason and H. Bradbury. *Handbook of Action Research: Participative Inquiry and Practice*. SAGE Publications, 2001.
- [160] M. Reichert, S. Rinderle, U. Kreher, and P. Dadam. *Adaptive Process Management with ADEPT2*. Proc. 21st Int'l. Conf. on Data Engineering (ICDE '05), pp.1113-1114, Demo Session, 2005.
- [161] D. J. Reifer. *Making the Software Business Case - Improvement by the Numbers*. Addison-Wesley, 2001.
- [162] D. J. Reifer. *Use of Real Options Theory to Value Software Trade Secrets*. Proc. 2nd Int'l. Workshop on Economics-Driven Software Engineering Research (EDSER '03), 2003.
- [163] H. A. Reijers. *Performance Improvement by Workflow Management Systems: Preliminary Results from an Empirical Study*. Proc. 6th Int'l. Conf. on Enterprise Information Systems (ICEIS '04), 3, pp.359-366, 2004.
- [164] H. A. Reijers and S. Poelmans. *Re-Configuring Workflow Systems to Facilitate a 'Smooth Flow of Work'*. Int'l. Journal of Cooperative Information Systems, 16(2), pp.155-175, 2007.
- [165] H. A. Reijers and W. M. P. van der Aalst. *The Effectiveness of Workflow Management Systems - Predictions and Lessons Learned*. Int'l. Journal of Information Management, 25(5), pp.457-471, 2005.

- [166] G. P. Richardson and A. L. Pugh. *System Dynamics - Modeling with DYNAMO*. Productivity Press, 1981.
- [167] S. Rinderle, R. Bobrik, M. Reichert, and T. Bauer. *Business Process Visualization - Use Cases, Challenges, Solutions*. Proc. Int'l. Conf. on Enterprise Information Systems (ICEIS '06), pp.204-211, 2006.
- [168] S. Rinderle, M. Reichert, and P. Dadam. *Flexible Support of Team Processes by Adaptive Workflow Systems*. Distributed and Parallel Databases, 16(1), pp. 91-116, 2004.
- [169] S. Rinderle, B. Weber, M. Reichert, and W. Wild. *Integrating Process Learning and Process Evolution - A Semantics Based Approach*. Proc. Int'l. Conf. on Business Process Management (BPM '05), pp.252-267, 2005.
- [170] S. Roy and P. K. J. Mohapatra. *Methodological Problems in the Formulation and Validation of System Dynamics Models Incorporating Soft Variables*. Proc. 21st Int'l. System Dynamics Conference, 2000.
- [171] P. Runeson. *Using Students as Experiment Subjects - An Analysis on Graduate and Freshmen Student Data*. Proc. 7th Int'l. Conf. on Empirical Assessment & Evaluation in Software Engineering (EASE '03), pp.95-102, 2003.
- [172] A. Sarmiento and A. Machado. *The Adoption of Workflow Systems: Proposal of a Model for a Methodology to Analyse the Impact of Workflow Systems in Organizations*. 2nd Int'l. Conf. on Enterprise Information Systems (ICEIS '00), pp.349-355, 2000.
- [173] A. Sarmiento and A. Machado. *Impact Evaluation of Organisational Changes Enabled by Workflow Systems*. 6th Int'l. Workshop on Groupware (CRIWG '00), pp.134-137, 2000.
- [174] P. G. Sassone. *Cost-Benefit Methodology for Office Systems*. ACM Transactions on Office Information Systems, 5(3), pp.273-289, 1987.
- [175] P. G. Sassone. *Cost Benefit Analysis of Information Systems: A Survey of Methodologies*. Proc. Int'l. Conf. on Supporting Group Work (GROUP '88), pp.73-83, 1988.
- [176] M. Sayal, F. Casati, U. Dayal, and M.-Ch. Shan. *Business Process Cockpit*. Proc. 28th Int'l. Conf. on Very Large Databases (VLDB '02), pp.880-883, 2002.
- [177] A.-W. Scheer, F. Abolhassan, and W. Bosch. *Real-Time Enterprise*. Springer, 2003.
- [178] H. J. Scholl. *Agent-based and System Dynamics Modeling: A Call for Cross Study and Joint Research*. Proc. 34th Hawaii Int'l. Conf. on System Sciences (HICSS '01), 2001.
- [179] M. Schwaninger. *System Dynamics and the Evolution of Systems Movement - A Historical Perspective*. Discussion Paper, Nr. 52, University of St. Gallen, Switzerland, 2005.
- [180] C. Seaman. *Qualitative Methods in Empirical Studies of Software Engineering*. IEEE Transactions on Software Engineering, 25(4), pp.557-572, 1999.
- [181] P. M. Senge. *The 5th Discipline - The Art and Practice of the Learning Organization (1st Edition)*. Currency Publications, 1990.
- [182] J. K. Shank and J. Fisher. *Target Costing as a Strategic Tool*. MIT Sloan Management Review, 41(1), pp.73-82, 1999.
- [183] M. Shapiro, T. Mandel, and M. Schwaninger. *The Primer Toolbox (Version 3.30.96)*. International Society for the Systems Sciences, 1996.

- [184] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman and Hall, 2000.
- [185] D. I. K. Sjoberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal. *A Survey of Controlled Experiments in Software Engineering*. IEEE Transactions on Software Engineering, 31(9), pp.733-753, 2005.
- [186] H. Smith and P. Fingar. *IT Doesn't Matter - Business Processes Do*. Meghan Kiffer Press, 2003.
- [187] R. Smith. *Applying Options Theories to Technology Management Decisions*. CTONet.org, 2004.
- [188] I. Sommerville and P. Sawyer. *Requirements Engineering: A Good Practice Guide*. John Wiley, 1997.
- [189] J. D. Sterman. *Business Dynamics - Systems Thinking and Modeling for a Complex World*. McGraw-Hill, 2000.
- [190] M. Sumner. *Risk Factors in Enterprise-Wide/ERP Projects*. Journal of Information Technology, 15(4), pp.317-327, 2000.
- [191] D. Sycamore and J. S. Collofello. *Using System Dynamics Modeling to Manage Projects*. Proc. 23rd Int'l. Conf. on Computer Software and Applications Conference, 15(4), pp.213-217, 1999.
- [192] N. Tadayon. *Neural Network Approach for Software Cost Estimation*. Proc. Int'l. Conf. on Information Technology: Coding and Computing (ITCC '05), 2005.
- [193] R. Tagg. *Requirements Engineering for Generic Process-aware Information Systems*. Proc. 9th Australian Workshop on Requirements Engineering (AWRE '04), pp.16.1-16.7, 2004.
- [194] P. P. Tallon and K. L. Kraemer. *A Process-oriented Alignment of Information Systems and Business Strategy: Implications for IT Business Value*. Proc. 4th Americas Conf. on Information Systems (AIS), 1998.
- [195] R. G. Santana Tapia. *A Value-based Maturity Model for IT Alignment in Networked Businesses*. Proc. Int'l. Conf. on Advanced Information Systems Engineering (CAiSE '06), pp.1201-1208, 2006.
- [196] M. Tegstam and J. Weiner. *Evaluating Costs and Benefits when Implementing an Information System - A PDM System at Autoliv Inc*. Masters Thesis, School of Economics and Commercial Law, Göteborg University, 1999.
- [197] Tibco. *Staffware Process Suite*. Staffware User Manual, 2005.
- [198] W. Tichy. *Should Computer Scientists Experiment More?* IEEE Computer, 31(5), pp.32-40, 1998.
- [199] M. P. Papazoglou W.J. van den Heuvel. *Business Process Development Life Cycle Methodology*. Communications of the ACM, 50(10), pp. 79-85, 2007.
- [200] W. M. P. van der Aalst. *Business Process Management - A Personal View*. Business Process Management Journal, 10(2), pp.135-139, 2004.
- [201] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. *Advanced Workflow Patterns*. Proc. 7th Int'l. Conf. on Cooperative Information Systems (CoopIS '00), LNCS 1901, pp.18-29, 2000.

- [202] W. M. P. van der Aalst and K. van Hee. *Workflow Management*. MIT Press, 2004.
- [203] W. M. P. van der Aalst and A. J. M. M. Weijters. *Process Mining - A Research Agenda*. *Computers in Industry*, 53(3), pp.231-244, 2004.
- [204] W. M. P. van der Aalst, M. Weske, and D. Grunbauer. *Case Handling: A New Paradigm for Business Process Support*. *Data and Knowledge Engineering*, 53(2), pp.129-162, 2005.
- [205] W.M.P van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. *Workflow Patterns*. *Distributed and Parallel Databases*, 14(3), pp.5-51, 2003.
- [206] P. van Eck, H. Blanken, and R. Wieringa. *Project GRAAL - Towards Operational Architecture Alignment*. *Int'l. Journal of Cooperative Information Systems*, 13(3), pp.235-255, 2004.
- [207] H. Vangheluwe, J. de Lara, and P. J. Mosterman. *An Introduction to Multi-Paradigm and Simulation*. *Proc. on AI, Simulation and Planning in High Autonomy Systems (AIS '2002)*, pp.9-20, 2002.
- [208] Vensim. *Ventana Systems*. <http://www.vensim.com/>, 2007.
- [209] VITAL. *Value-based IT ALignment (VITAL)*, University of Twente & Free University of Amsterdam. Project Homepage: <http://www.vital-project.org/>, 2006.
- [210] C. Vogt. *Intractable ERP: A Comprehensive Analysis of Failed Enterprise-Resource-Planning Projects*. *ACM SIGSOFT Software Engineering Notes*, 27(2), pp.62-68, 2002.
- [211] H. Voordijk, A. van Leuven, and A. Laan. *Enterprise Resource Planning in a Large Construction Firm: Implementation Analysis*. *Construction Management and Economics*, 21(5), pp.511-521, 2003.
- [212] W. Wakeland and M. Hoarfrost. *The Case for Thoroughly Testing Complex System Dynamics Models*. *Proc. 23th Int'l. System Dynamics Conference*, 2000.
- [213] B. Weber. *Aktuelles Schlagwort: Business Process Life Cycle*. EMISA Forum, Volume 1, 2007.
- [214] B. Weber, S. Rinderle, and M. Reichert. *Change Patterns and Change Support Features in Process-Aware Information Systems*. *Proc. 13th Int'l. Conf. on Advanced Information Systems Engineering (CAiSE '07)*, 2007.
- [215] G. M. Weinberg. *An Introduction to General Systems Thinking*. Dorset House Publishing, 2001.
- [216] R. J. Wieringa and J. M. G. Heerkens. *The Methodological Soundness of Requirements Engineering Papers: A Conceptual Framework and Two Case Studies*. *Requirements Engineering*, 11(4), pp.295-307, 2006.
- [217] R. J. Wieringa, N. A. M. Maiden, N. R. Mead, and C. Rolland. *Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion*. *Requirements Engineering*, 11(1), pp.102-107, 2006.
- [218] R. J. Wieringa, P. van Eck, and H. Blanken. *Architecture Alignment in a Large Government Organization: A Case Study*. *Proc. 16th Int'l. Conf. on Advanced Information Systems Engineering (CAiSE '04)*, Forum, Riga, Latvia, 2004.
- [219] G. Winch and D. J. W. Arthur. *User-Parameterised Generic Models: A Solution to the Conundrum of Modelling Access for SMEs?* *System Dynamics Review*, 18(3), pp.339-357, 2003.

- [220] R. K. Yin. *Case Study Research - Design and Methods*. SAGE Publications, 2003.
- [221] E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD Thesis, University of Toronto, 1995.
- [222] Y. Yusufa, A. Gunasekaranb, and M. S. Abthorpe. *Enterprise Information Systems Project Implementation: A Case Study of ERP in Rolls-Royce*. Int'l. Journal of Production Economics, Volume 87, pp.251-266, 2004.
- [223] J. A. Zachman. *A Framework for Information Systems Architecture*. IBM Systems Journal, 26(3), pp. 276-292, 1987.
- [224] A. A. Zagonel and T. F. Corbet. *Levels of Confidence in System Dynamics Modeling: A Pragmatic Approach to Assessment of Dynamic Models*. Proc. 24th Int'l. System Dynamics Conference, 2006.
- [225] N. Zarvic, M. Daneva, and R. Wieringa. *Value-Based Requirements Engineering for Value Webs*. Proc. 13th Working Conf. on Requirements Engineering: Foundation for Software Quality (REFSQ '07), 2007.
- [226] M. V. Zelkowitz and D. R. Wallace. *Experimental Models for Validating Technology*. IEEE Computer, 31(5), pp.23-31, 1998.
- [227] R. W. Zmud. *Information Systems in Organisations*. Scott Foresman, 1983.
- [228] M. zur Muehlen. *Workflow-based Process Controlling - Foundation, Design and Application of Workflow-driven Process Information Systems*. Logos Verlag, 2004.

Samenvatting

Om concurrerend in hun markt te kunnen blijven, is het voor ondernemingen cruciaal geworden te voorzien in effectieve IT ondersteuning van bedrijfsprocessen. Bedrijfsprocessen moeten gedefinieerd, geïmplementeerd, uitgevoerd, gecontroleerd en continue aangepast worden aan veranderende omstandigheden. Ondersteuning van de levenscyclus van processen en continue procesverbetering zijn kritieke succesfactoren geworden in hedendaagse en toekomstige bedrijfsbrede geautomatiseerde informatievoorziening.

Procesbewuste informatiesystemen (PBIS) spelen in deze context een sleutelrol. Daarbinnen zijn organisatiespecifieke en -generieke proces ondersteunende systemen te onderscheiden. In het eerste geval wordt de PBIS vanuit het niets opgebouwd en omvat organisatiespecifieke informatie over de structuur en de te ondersteunen processen. In het tweede geval bevat de PBIS geen enkele informatie over de structuur en de processen van een specifieke organisatie. In plaats daarvan moet een organisatie de PBIS configureren door processen, organisatorische entiteiten, en bedrijfsobjecten te specificeren.

Om de realisatie van PBIS mogelijk te maken, zijn talrijke proces ondersteunende paradigma's, procesmodellering standaarden, en bedrijfsprocesmanagement tools geïntroduceerd. De toepassing van deze methodes in PBIS engineeringprojecten wordt niet alleen beïnvloed door technologische maar ook door organisatorische en projectspecifieke factoren. Tussen deze factoren bestaan talrijke causale afhankelijkheden die op hun beurt vaak leiden tot complexe en onverwachte effecten in BPIS engineeringprojecten. Meer specifiek, de kosten van BPIS engineeringprojecten worden significant beïnvloed door deze causale afhankelijkheden.

Daarom is een allesomvattende aanpak nodig die PBIS ingenieurs in staat stelt om systematisch deze causale afhankelijkheden evenals hun invloed op de kosten van PBIS engineeringprojecten te onderzoeken. Bestaande financieel gemotiveerde IT evaluaties en software kostenramingen zijn echter niet in staat causale afhankelijkheden en resulterende effecten in aanmerking te nemen. Als antwoord hierop introduceert deze thesis het EcoPOST raamwerk. Dit raamwerk maakt gebruik van evaluatiemodellen om de wisselwerking tussen technologische, organisatorische, en projectspecifieke evaluatiefactoren te beschrijven, en maakt gebruik van simulatieconcepten om het dynamische karakter van PBIS engineeringprojecten inzichtelijk te maken. In deze context ondersteunt het EcoPOST raamwerk ook hergebruik van evaluatiemodellen gebaseerd op een bibliotheek van generieke vooraf gedefinieerde evaluatiepatronen en voorziet ook in besturende richtlijnen die de overgang van het EcoPOST raamwerk naar de praktijk bevordert (bv. richtlijnen voor het ontwerpen van een model). Tool support is ook beschikbaar.

Ten slotte presenteren wij de resultaten van twee online surveys, drie case studies, en één gecontroleerd software experiment. Ondersteund door deze empirische en experimentele onderzoeksactiviteiten zijn wij in staat de evaluatieconcepten die aan de basis staan van het EcoPOST raamwerk te valideren. Daarnaast zijn wij door voornoemde onderzoeksresultaten in staat de praktische toepasbaarheid te demonstreren.

SIKS Dissertation Series

1998

[1998-1] J. van den Akker (CWI) *DEGAS - An Active, Temporal Database of Autonomous Objects*

[1998-2] F. Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*

[1998-3] A. Steuten (TUD) *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*

[1998-4] D. Breuker (UM) *Memory versus Search in Games*

[1998-5] E. W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

[1999-1] Mark Sloof (VU) *Physiology of Quality Change Modelling - Automated Modelling of Quality Change of Agricultural Products*

[1999-2] Rob Potharst (EUR) *Classification using Decision Trees and Neural Nets*

[1999-3] Don Beal (UM) *The Nature of Minimax Search*

[1999-4] Jacques Penders (UM) *The Practical Art of Moving Physical Objects*

[1999-5] Aldo de Moor (KUB) *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*

[1999-6] Niek J.E. Wijngaards (VU) *Re-design of Compositional Systems*

[1999-7] David Spelt (UT) *Verification Support for Object Database Design*

[1999-8] Jacques H.J. Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation*

2000

- [2000-1] Frank Niessink (VU) *Perspectives on Improving Software Maintenance*
- [2000-2] Koen Holtman (TUE) *Prototyping of CMS Storage Management*
- [2000-3] Carolien M.T. Metselaar (UVA) *Sociaal-organisatorische gevolgen van kennistechnologie: een procesbenadering en actorperspectief*
- [2000-4] Geert de Haan (VU) *ETAG - A Formal Model of Competence Knowledge for User Interface Design*
- [2000-5] Ruud van der Pol (UM) *Knowledge-based Query Formulation in Information Retrieval*
- [2000-6] Rogier van Eijk (UU) *Programming Languages for Agent Communication*
- [2000-7] Niels Peek (UU) *Decision-Theoretic Planning of Clinical Patient Management*
- [2000-8] Veerle Coup, (EUR) *Sensitivity Analysis of Decision-Theoretic Networks*
- [2000-9] Florian Waas (CWI) *Principles of Probabilistic Query Optimization*
- [2000-10] Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*
- [2000-11] Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

- [2001-1] Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*
- [2001-2] Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*
- [2001-3] Maarten van Someren (UvA) *Learning as problem solving*
- [2001-4] Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
- [2001-5] Jacco van Ossenbruggen (VU) *Processing Structured Hypermedia: A Matter of Style*
- [2001-6] Martijn van Welie (VU) *Task-based User Interface Design*
- [2001-7] Bastiaan Schonhage (VU) *Diva: Architectural Perspectives on Information Visualization*
- [2001-8] Pascal van Eck (VU) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*

[2001-9] Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

[2001-10] Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice BRAHMS: A Multiagent Modeling and Simulation Language for Work Practice Analysis and Design*

[2001-11] Tom M. van Engers (VUA) *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002

[2002-01] Nico Lassing (VU) *Architecture-Level Modifiability Analysis*

[2002-02] Roelof van Zwol (UT) *Modelling and Searching Web-based Document Collections*

[2002-03] Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*

[2002-04] Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*

[2002-05] Radu Serban (VU) *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*

[2002-06] Laurens Mommers (UL) *Applied Legal Epistemology: Building a Knowledge-based Ontology of the Legal Domain*

[2002-07] Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*

[2002-08] Jaap Gordijn (VU) *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

[2002-09] Willem-Jan van den Heuvel(KUB) *Integrating Modern Business Applications with Objectified Legacy Systems*

[2002-10] Brian Sheppard (UM) *Towards Perfect Play of Scrabble*

[2002-11] Wouter C.A. Wijngaards (VU) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

[2002-12] Albrecht Schmidt (Uva) *Processing XML in Database Systems*

[2002-13] Hongjing Wu (TUE) *A Reference Architecture for Adaptive Hypermedia Applications*

[2002-14] Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

[2002-15] Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

[2002-16] Pieter van Langen (VU) *The Anatomy of Design: Foundations, Models and Applications*

[2002-17] Stefan Manegold (UVA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

[2003-01] Heiner Stuckenschmidt (VU) *Ontology-Based Information Sharing in Weakly Structured Environments*

[2003-02] Jan Broersen (VU) *Modal Action Logics for Reasoning About Reactive Systems*

[2003-03] Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

[2003-04] Milan Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*

[2003-05] Jos Lehmann (UVA) *Causation in Artificial Intelligence and Law - A Modelling Approach*

[2003-06] Boris van Schooten (UT) *Development and Specification of Virtual Environments*

[2003-07] Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*

[2003-08] Yongping Ran (UM) *Repair Based Scheduling*

[2003-09] Rens Kortmann (UM) *The Resolution of Visually Guided Behaviour*

[2003-10] Andreas Lincke (UvT) *Electronic Business Negotiation: Some Experimental Studies on the Interaction between Medium, Innovation Context and Culture*

[2003-11] Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

[2003-12] Roeland Ordelman (UT) *Dutch Speech Recognition in Multimedia Information Retrieval*

[2003-13] Jeroen Donkers (UM) *Nosce Hostem - Searching with Opponent Models*

[2003-14] Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

[2003-15] Mathijs de Weerd (TUD) *Plan Merging in Multi-Agent Systems*

[2003-16] Menzo Windhouwer (CWI) *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*

[2003-17] David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

[2003-18] Levente Kocsis (UM) *Learning Search Decisions*

2004

[2004-01] Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

[2004-02] Lai Xu (UvT) *Monitoring Multi-Party Contracts for E-Business*

[2004-03] Perry Groot (VU) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

[2004-04] Chris van Aart (UVA) *Organizational Principles for Multi-Agent Architectures*

[2004-05] Viara Popova (EUR) *Knowledge discovery and monotonicity*

[2004-06] Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*

[2004-07] Elise Boltjes (UM) *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

[2004-08] Joop Verbeek(UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politile gegevensuitwisseling en digitale expertise*

[2004-09] Martin Caminada (VU) *For the Sake of the Argument - Explorations into Argument-based Reasoning*

[2004-10] Suzanne Kabel (UVA) *Knowledge-Rich Indexing of Learning-Objects*

[2004-11] Michel Klein (VU) *Change Management for Distributed Ontologies*

[2004-12] The Duy Bui (UT) *Creating emotions and facial expressions for embodied agents*

[2004-13] Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*

[2004-14] Paul Harrenstein (UU) *Logic in Conflict - Logical Explorations in Strategic Equilibrium*

[2004-15] Arno Knobbe (UU) *Multi-Relational Data Mining*

[2004-16] Federico Divina (VU) *Hybrid Genetic Relational Search for Inductive Learning*

[2004-17] Mark Winands (UM) *Informed Search in Complex Games*

[2004-18] Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*

[2004-19] Thijs Westerveld (UT) *Using Generative Probabilistic Models for Multimedia Retrieval*

[2004-20] Madelon Evers (Nyenrode) *Learning from Design: Facilitating Multidisciplinary Design Teams*

2005

[2005-01] Floor Verdenius (UVA) *Methodological Aspects of Designing Induction-Based Applications*

[2005-02] Erik van der Werf (UM) *AI techniques for the game of Go*

[2005-03] Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*

[2005-04] Nirvana Meratnia (UT) *Towards Database Support for Moving Object Data*

[2005-05] Gabriel Infante-Lopez (UVA) *Two-Level Probabilistic Grammars for Natural Language Parsing*

[2005-06] Pieter Spronck (UM) *Adaptive Game AI*

[2005-07] Flavius Frasincar (TUE) *Hypermedia Presentation Generation for Semantic Web Information Systems*

[2005-08] Richard Vdovjak (TUE) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*

[2005-09] Jeen Broekstra (VU) *Storage, Querying and Inferencing for Semantic Web Languages*

[2005-10] Anders Bouwer (UVA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*

[2005-11] Elth Ogston (VU) *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*

[2005-12] Csaba Boer (EUR) *Distributed Simulation in Industry*

[2005-13] Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*

[2005-14] Borys Omelayenko (VU) *Web-Service Configuration on the Semantic Web - Exploring how Semantics meets Pragmatics*

[2005-15] Tibor Bosse (VU) *Analysis of the Dynamics of Cognitive Processes*

- [2005-16] Joris Graaumans (UU) *Usability of XML Query Languages*
- [2005-17] Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*
- [2005-18] Danielle Sent (UU) *Test-Selection Strategies for Probabilistic Networks*
- [2005-19] Michel van Dartel (UM) *Situated Representation*
- [2005-20] Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*
- [2005-21] Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

- [2006-01] Samuil Angelov (TUE) *Foundations of B2B Electronic Contracting*
- [2006-02] Cristina Chisalita (VU) *Contextual Issues in the Design and Use of Information Technology in Organizations*
- [2006-03] Noor Christoph (UVA) *The role of Metacognitive Skills in Learning to Solve Problems*
- [2006-04] Marta Sabou (VU) *Building Web Service Ontologies*
- [2006-05] Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*
- [2006-06] Ziv Baida (VU) *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*
- [2006-07] Marko Smiljanic (UT) *XML Schema Matching - Balancing Efficiency and Effectiveness by means of Clustering*
- [2006-08] Eelco Herder (UT) *Forward, Back and Home Again - Analyzing User Behavior on the Web*
- [2006-09] Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*
- [2006-10] Ronny Siebes (VU) *Semantic Routing in Peer-to-Peer Systems*
- [2006-11] Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*
- [2006-12] Bert Bongers (VU) *Interactivation - Towards an E-cology of People, our Technological Environment, and the Arts*
- [2006-13] Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*
- [2006-14] Johan Hoorn (VU) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*

- [2006-15] Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*
- [2006-16] Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*
- [2006-17] Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*
- [2006-18] Valentin Zhizhkun (UVA) *Graph Transformation for Natural Language Processing*
- [2006-19] Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*
- [2006-20] Marina Velikova (UvT) *Monotone Models for Prediction in Data Mining*
- [2006-21] Bas van Gils (RUN) *Aptness on the Web*
- [2006-22] Paul de Vrieze (RUN) *Fundamentals of Adaptive Personalisation*
- [2006-23] Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*
- [2006-24] Laura Hollink (VU) *Semantic Annotation for Retrieval of Visual Resources*
- [2006-25] Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*
- [2006-26] Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
- [2006-27] Stefano Bocconi (CWI) *Vox Populi: Generating Video Documentaries from Semantically annotated Media Repositories*
- [2006-28] Borkur Sigurbjornsson (UVA) *Focused Information Access using XML Element Retrieval*

2007

- [2007-01] Kees Leune (UvT) *Access Control and Service-Oriented Architectures*
- [2007-02] Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*
- [2007-03] Peter Mika (VU) *Social Networks and the Semantic Web*
- [2007-04] Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-Agent Systems: a Dialogue-based Approach*
- [2007-05] Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*

- [2007-06] Gilad Mishne (UVA) *Applied Text Analytics for Blogs*
- [2007-07] Natasa Jovanovic' (UT) *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*
- [2007-08] Mark Hoogendoorn (VU) *Modeling of Change in Multi-Agent Organizations*
- [2007-09] David Mobach (VU) *Agent-Based Mediated Service Negotiation*
- [2007-10] Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
- [2007-11] Natalia Stash (TUE) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
- [2007-12] Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
- [2007-13] Rutger Rienks (UT) *Meetings in Smart Environments - Implications of Progressing Technology*
- [2007-14] Niek Bergboer (UM) *Context-Based Image Analysis*
- [2007-15] Joyca Lacroix (UM) *NIM: A Situated Computational Memory Model*
- [2007-16] Davide Grossi (UU) *Designing Invisible Handcuffs - Formal Investigations in Institutions and Organizations for Multi-Agent Systems*
- [2007-17] Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*
- [2007-18] Bart Orriens (UvT) *On the Development and Management of Adaptive Business Collaborations*
- [2007-19] David Levy (UM) *Intimate Relationships with Artificial Partners*
- [2007-20] Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*
- [2007-21] Karianne Vermaas (UU) *Fast Diffusion and Broadening Use: A Research on residential Adoption and Usage of Broadband Internet in the Netherlands between 2001 and 2005*
- [2007-22] Zlatko Zlatev (UT) *Goal-oriented Design of Value and Process Models from Patterns*
- [2007-23] Peter Barna (TUE) *Specification of Application Logic in Web Information Systems*
- [2007-24] Georgina Ramirez Camps (CWI) *Structural Features in XML Retrieval*
- [2007-25] Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*

2008

[2008-01] Katalin Boer-Sorban (EUR) *Agent-Based Simulation of Financial Markets: A Modular, Continuous-time Approach*

[2008-02] Alexei Sharpanskykh (VU) *On Computer-Aided Methods for Modeling and Analysis of Organizations*

[2008-03] Vera Hollink (UVA) *Optimizing Hierarchical Menus: A Usage-based Approach*

[2008-04] Ander de Keijzer (UT) *Management of Uncertain Data - Towards Unattended Integration*

[2008-05] Bela Mutschler (UT) *Modeling and Simulating Causal Dependencies on Process-aware Information Systems from a Cost Perspective*