Understanding the Costs of Business Process Management Technology

Bela Mutschler and Manfred Reichert

Abstract Providing effective IT support for business processes has become crucial for enterprises to stay competitive in their market. Business processes must be defined, configured, implemented, enacted, monitored and continuously adapted to changing situations. Process life cycle support and continuous process improvement have therefore become critical success factors in enterprise computing. In response to this need, a variety of process support paradigms, process specification standards, process management tools, and supporting methods have emerged. Summarized under the term Business Process Management (BPM), they have become a success-critical instrument for improving overall business performance. However, introducing BPM approaches in enterprises is associated with significant costs. Though existing economic-driven IT evaluation and software cost estimation approaches have received considerable attention during the last decades, it is difficult to apply them to BPM projects. In particular, they are unable to take into account the dynamic evolution of BPM projects caused by the numerous technological, organizational and project-specific factors influencing them. The latter, in turn, often lead to complex and unexpected cost effects in BPM projects making even rough cost estimations a challenge. What is needed is a comprehensive approach enabling BPM professionals to systematically investigate the costs of BPM projects. This chapter takes a look at both known and often unknown cost factors in BPM projects, shortly discusses existing IT evaluation and software cost estimation approaches with respect to their suitability for BPM projects, and finally introduces the EcoPOST framework. EcoPOST utilizes evaluation models to describe the interplay of technological, organizational, and project-specific BPM cost factors as well as simulation concepts to unfold the dynamic behavior and costs of BPM projects.

Bela Mutschler
Business Informatics Group, University of Applied Sciences Ravensburg-Weingarten, Germany,
e-mail: bela.mutschler@hs-weingarten.de

Manfred Reichert
Institute of Databases and Information Systems, University of Ulm, Germany,
e-mail: manfred.reichert@uni-ulm.de

1 Introduction

1.1 Motivation

While the benefits of *Process-Aware Information Systems* (PAISs) and BPM technology are usually justified by improved process performance [53, 67, 69], there exist no approaches for systematically analyzing related cost factors and their dependencies. Though software cost estimation [4] has received considerable attention during the last decades and has become an essential task in information systems engineering, it is difficult to apply existing approaches to BPM projects. This difficulty stems from the inability of these approaches to cope with the numerous technological, organizational and project-driven cost factors to be considered in the context of BPM projects [40]. As example consider the significant costs for redesigning business processes. Another challenge deals with the many dependencies existing between the different cost factors. Activities for *business process redesign*, for example, can be influenced by intangible impact factors like available *process knowledge* or *end user fears*. These dependencies, in turn, result in dynamic effects which influence the overall costs of BPM projects. Existing evaluation techniques [45] are usually unable to deal with such dynamic effects as they rely on rather static models based upon snapshots of the considered project context.

What is needed is an approach that enables organizations to investigate the complex interplay between the many cost and impact factors that arise in the context of PAIS introduction and BPM projects [52]. This chapter presents the EcoPOST methodology, a sophisticated and practically validated, model-based methodology to better understand and systematically investigate the complex cost structures of such BPM projects.

1.2 IT Evaluation - Challenges and Approaches

Generally, the adoption of *information technology* (IT) can be described by means of an *S curve* (cf. Fig. 1A) [10, 11, 62]. When new IT emerges at the market, it is unproven, expensive and difficult to use. Standards have not been established yet and best practices still have to established. At this point, only ”first movers” start projects based on the emerging IT. They assume that the high costs and risks for being an innovator will be later compensated by gaining competitive advantage [7].

Picking up an emerging IT at a later stage, 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. The S curve is then complete. 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 also erode the ability of IT as enabler 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 difficult to achieve further strategic benefits (though rapid technological innovation often continues). This can be illustrated by a *Z curve* (cf. Fig. 1B).

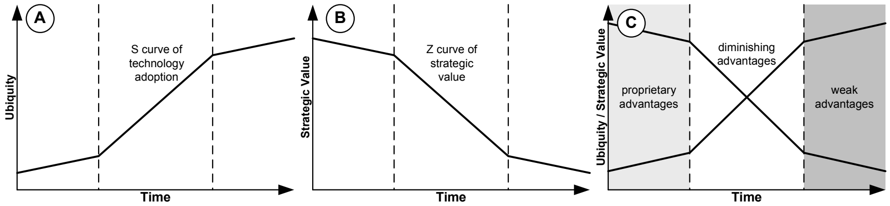
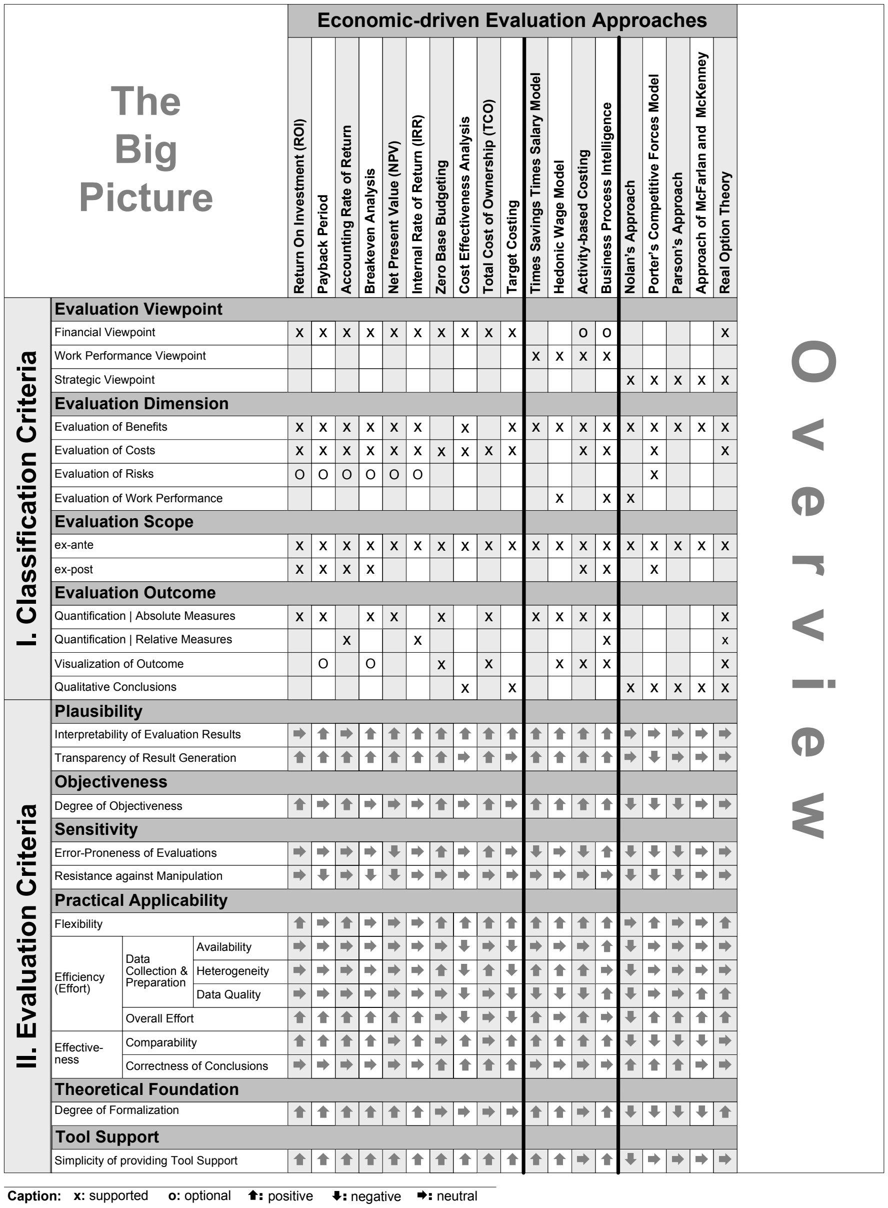


Fig. 1 The Curves of Technology Adoption.

Considering the different curves of IT adoption, decisions about IT investments (and the right point in time to realize them) constitute a difficult task [5, 18] (cf. Fig. 1C). Respective decisions are influenced by numerous factors [30, 36, 39, 37]. Hence, policy makers often demand for a business case [51] summarizing the key parameters of an IT investment. Thereby, different evaluation dimensions are taken into account [31]. As examples consider the costs of an investment, its assumed profit, its impact on work performance, business process performance, and the achievement of enterprise goals.

To cope with different evaluation goals, numerous evaluation approaches have been introduced (e.g. [48, 57, 58]). Fig. 2 shows results of an evaluation of 19 IT-evaluation approaches we presented in [45]. Today's policy makers usually rely on simple and static decision models as well as on intuition and experiences rather than on a profound analysis of an IT investment decision. Further, rules of thumb such as ”invest to keep up with the technology” or ”invest if the competitors have been successful” are often applied. In many cases, there is an asymmetric consideration of costs and benefits. For example, many financial calculations (cf. Section 4) overestimate benefits in the first years in order to realize a positive ROI. Besides, many standard evaluation approaches are often not suitable to be used at early planning stages of IT investments. In fact, many projects (especially those utilizing innovative information technology) - despite their potential strategic importance - have a negative economic valuation result at an early stage. This situation results in a high risk of false rejection. This means that enterprises with independently operating business units, targeting at maximizing the equity of a company in short term, have to overcome the problem to not routinely reject truly important IT investments due to the use of insufficient evaluation techniques.



The Big Picture				Economic-driven Evaluation Approaches																			Overview
				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	
I. Classification Criteria	**Evaluation Viewpoint**																						
	Financial Viewpoint			X	X	X	X	X	X	X	X	X	X			O	O					X	
	Work Performance Viewpoint													X	X	X	X						
	Strategic Viewpoint																	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	
	Evaluation of Costs			X	X	X	X	X	X	X	X	X	X			X	X		X			X	
	Evaluation of Risks			O	O	O	O	O	O										X				
	Evaluation of Work Performance														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	
	ex-post			X	X	X	X									X	X		X				
	Evaluation Outcome																						
	Quantification | Absolute Measures			X	X		X	X		X		X		X	X	X	X					X	
	Quantification | Relative Measures					X			X								X					x	
	Visualization of Outcome				O		O			X		X			X	X	X					X	
	Qualitative Conclusions										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	→	→	→	→	→	→	→	↓	→	↓	↓	↓	↓	↑	↓	→	→	↑	↑	
		Overall Effort		↑	↑	↑	↑	↑	↑	→	↓	→	↓	↑	→	↑	→	↓	↑	↑	↑	↑	
	Effective-ness	Comparability		↑	↑	↑	↑	→	↑	→	↑	→	↑	↑	↑	↑	↑	↓	↓	↓	↓	→	
		Correctness of Conclusions		→	→	→	→	→	→	↑	↑	↑	↑	→	→	→	→	↑	↑	↑	→	→	
	Theoretical Foundation																						
	Degree of Formalization			↑	↑	↑	↑	↑	↑	→	→	→	→	↑	↑	→	↑	↓	↓	↓	↓	↑	
	Tool Support																						
	Simplicity of providing Tool Support			↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	→	↑	↓	→	→	→	→	

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

Fig. 2 IT Evaluation Approaches: The Big Picture.

1.3 Chapter Overview

Section 2 summarizes the EcoPOST cost analysis method. Section 3 introduces evaluation patterns as an important means to simplify the application of the EcoPOST cost analysis approach. Section 4 introduces rules for designing EcoPOST evaluation models, while Section 5 describes general modeling guidelines. Section 6 introduces a case study to illustrate the use of EcoPOST and its practical benefits. Finally, Sections 7 and 8 conclude with a discussion of our approach and a summary.

2 The EcoPOST Cost Analysis Methodology

Our EcoPOST methodology was designed to support the introduction of Process-aware Information Systems (PAISs) [14, 50, 49] and BPM technology [13, 22, 70, 55] in the automotive industry (and was, consequently, also validated and piloted in several BPM projects in this domain). The EcoPOST methodology comprises seven steps (cf. Fig. 3). *Step 1* concerns the comprehension of an evaluation scenario. This is crucial for developing problem-specific evaluation models. The following two steps (Steps 2 and 3) deal with the identification of two different kinds of *Cost Factors* representing costs that can be quantified in terms of money (cf. Table 1): *Static Cost Factors* (SCFs) and *Dynamic Cost Factors* (DCFs).

	Description
SCF	*Static Cost Factors* (SCFs) represent costs whose values do not change during an BPM project (except for their time value, which is not further considered in the following). Typical examples: software license costs, hardware costs and costs for external consultants.
DCF	*Dynamic Cost Factors* (DCFs), in turn, represent costs that are determined by activities related to an BPM 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*. The 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 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 the EcoPOST framework, intangible factors like the one described are represented by *impact factors*.

Table 1 Cost Factors.

Step 4 deals with the identification of *Impact Factors* (ImFs), i.e., intangible factors that influence DCFs and other ImFs. We distinguish between organizational, project-specific and technological ImFs. ImFs cause the value of DCFs (and other ImFs) to change, making their evaluation a difficult task to accomplish. As examples consider factors such as ”End User Fears”, ”Availability of Process Knowledge” and ”Ability to (Re-)design Business Processes”. Finally, ImFs can be *static* or *dynamic* (cf. Table 2).

	Description
Static ImF	Static ImFs do not change, i.e., they are assumed to be constant during an BPM project; e.g., when there is a fixed degree of user fears, process complexity, or work profile change.
Dynamic ImF	Dynamic ImFs may change during an BPM project, e.g., due to interference with other ImFs. As examples consider process and domain knowledge which is typically varying during an BPM project (or a subsidiary activity).

Table 2 Impact Factors.

Unlike SCFs and DCFs the values of ImFs are not quantified in monetary terms. Instead, they are ”quantified” by experts[1] using qualitative scales describing the degree of an ImF. As known from software cost estimation models, such as COCOMO [4], the qualitative scales we use comprise different ”values” (typically ranging from ”very low” to ”very high”). These values are used to express the strength of an ImF on a given cost factor (just like in COCOMO).

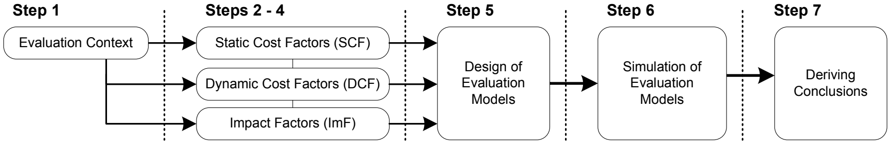

Step 1
Steps 2 - 4
Step 5
Step 6
Step 7
Evaluation Context
Static Cost Factors (SCF)
Dynamic Cost Factors (DCF)
Impact Factors (ImF)
Design of Evaluation Models
Simulation of Evaluation Models
Deriving Conclusions

Fig. 3 Main Steps of the EcoPOST Methodology.

Generally, dynamic evaluation factors (i.e., DCFs and dynamic ImFs) are difficult to comprehend. In particular, intangible ImFs (i.e., their appearance and impact in BPM projects) are not easy to follow. When evaluating the costs of BPM projects, therefore, DCFs and dynamic ImFs constitute a major source of misinterpretation and ambiguity. 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 (*Step 5*; cf. Section 3). These evaluation models can be simulated (*Step 6*) to gain insights into the dynamic behavior (i.e., evolution) of DCFs and dynamic ImFs (*Step 7*). This is important to effectively control the design and implementation of PAIS as well as the costs of respective BPM projects.

2.1 Evaluation Models

In EcoPOST, dynamic cost/impact factors are captured and analyzed by evaluation models which are specified using the System Dynamics [54] notation (cf. Fig. 4). An evaluation model comprises 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 Re-

[1] The efforts of these experts for making that quantification is not explicitly taken into account in EcoPOST, though this effort also increases information system development costs.

design Costs”; cf. Fig. 4A) and dynamic ImFs (e.g., ”Process Knowledge”). A state variable is graphically denoted as rectangle (cf. Fig. 4A), 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 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 variables connected to other ones) (cf. Fig. 4A). 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. 4A). 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. 4A, 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. 4A). In particular, a rate variable controls the inflow/outflow it is assigned to based on those SCFs, DCFs and ImFs which influence it. It can be considered as an interface which is able to merge SCFs, DCFs and ImFs.

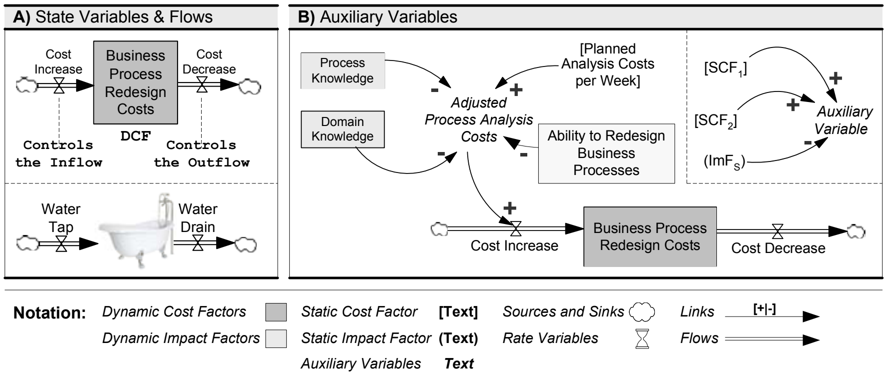


Fig. 4 Evaluation Model Notation and initial Examples.

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 ImFs. As example consider the auxiliary variable ”Adjusted Process Analysis Costs” in Fig. 4B, which merges 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 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 define:

Definition 2.1 (Evaluation Model) *A graph EM = (V, F, L) is denotes as 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 representing 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 representing links with* $Lab := \{+, -\}$ *being the set of link labels:*
  - $(q_i, q_j, +) \in L$ *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.*

Generally, the evolution of DCFs and dynamic ImFs is difficult to comprehend. Thus, EcoPOST additionally provides a simulation component for capturing and analyzing this evolution (cf. Step 6 in Fig. 3).

2.2 Understanding Model Dynamics through Simulation

To enable simulation of an evaluation model we need to formally specify its behavior. EcoPOST accomplishes this by means of a *simulation model* based on *mathematical equations*. Thereby, the behavior of each model variable is specified by one equation (cf. Fig. 5), which describes how a variable is changing over time during simulation.

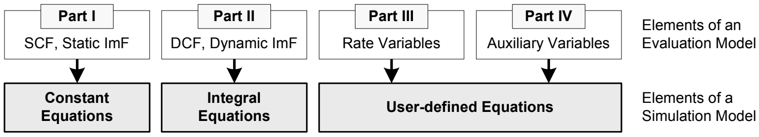


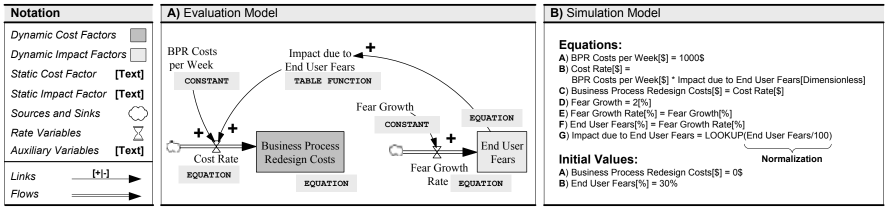
Fig. 5 Elements of a Simulation Model.

Fig. 6A shows a simple evaluation model.[2] Assume that the evolution of the DCF ”Business Process Redesign Costs” (triggered by the dynamic ImF ”End User Fears”) shall be analyzed. End user fears can lead to emotional resistance of users and, in turn, to a lack of user support when redesigning business processes (e.g., during an interview-based process analysis). For model variables representing an SCF or static ImF the equation specifies a constant value for the model variable; i.e., SCFs and static ImFs are specified by single numerical values in *constant equations*. As example consider EQUATION A in Fig. 6B. For model variables representing DCFs, dynamic ImFs or rate/auxiliary variables, the corresponding equation describes how the value of the model variable evolves over time (i.e., during simulation). Thereby, the evolution of DCFs and dynamic ImFs is characterized by *integral equations* [16]. This allows us to capture the accumulation of DCFs and dynamic ImFs from the start of a simulation run (t_0) to its end (t):

Definition 2.2 (Integral Equation) *Let EM be an evaluation model (cf. Definition 2.1) and S be the set of all DCFs and dynamic ImFs defined by EM. An integral equation for a dynamic factor $v \in S$ 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.*



Equations:
A) BPR Costs per Week[$] = 1000$
B) Cost Rate[$] =
BPR Costs per Week[$] * Impact due to End User Fears[Dimensionless]
C) Business Process Redesign Costs[$] = Cost Rate[$]
D) Fear Growth = 2[%]
E) Fear Growth Rate[%] = Fear Growth[%]
F) End User Fears[%] = Fear Growth Rate[%]
G) Impact due to End User Fears = LOOKUP$\underbrace{\text{(End User Fears/100)}}_{\textbf{Normalization}}$

Initial Values:
A) Business Process Redesign Costs[$] = 0$
B) End User Fears[%] = 30%

Fig. 6 Dealing with the Impact of End User Fears.

As example consider EQUATION C in Fig. 6B which specifies the increase of the DCF ”Business Process Redesign Costs” (based on only one inflow). Note that in Fig. 6B the equations of the DCF ”Business Process Redesign Costs” and the dynamic ImF ”End User Fears” are presented in the way they are specified in Vensim [65], the simulation tool used by EcoPOST, and not as real integral equations.

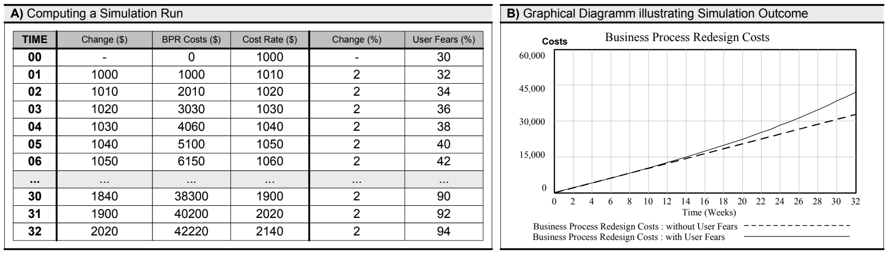
Rate and auxiliary variables are both specified in the same way, i.e., as user-defined functions defined over the variables preceding them in the evaluation model. In

[2] It is the basic goal of this toy example to illustrate how evaluation models are simulated. Generally, evaluation models are much more complex. Due to lack of space we do not provide a more extensive example here.

other words, rate as well as auxiliary variables are used to merge static and dynamic cost/impact factors. During simulation, values of rate and auxiliary 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 and auxiliary variables is specified in the same way:

Definition 2.3 (User-defined Equation) *Let EM be an evaluation model (cf. Def. 2.1) and X be the set of rate/auxiliary variables defined by EM. An equation for* $v \in X$ *is a user-defined function* $f(v_1, ..., v_n)$ *with* $v_1, ..., v_n$ *being the predecessors of v in EM.*

As example consider EQUATION B in Fig. 6B. The equation for rate variable ”Cost Rate” merges the SCF ”BPR Costs per Week” with the auxiliary variable ”Impact due to End User Fears”. Assuming that activities for business process redesign are scheduled for 32 weeks, Fig. 7A shows the values of all dynamic evaluation factors of the evaluation model over time when performing a simulation. Fig. 7B shows the outcome of the simulation. As can be seen there is a significant negative impact of end user fears on the costs of business process redesign.



A) Computing a Simulation Run

TIME	Change ($)	BPR Costs ($)	Cost Rate ($)	Change (%)	User Fears (%)
00	-	0	1000	-	30
01	1000	1000	1010	2	32
02	1010	2010	1020	2	34
03	1020	3030	1030	2	36
04	1030	4060	1040	2	38
05	1040	5100	1050	2	40
06	1050	6150	1060	2	42
...	...	...	...	...	...
30	1840	38300	1900	2	90
31	1900	40200	2020	2	92
32	2020	42220	2140	2	94

B) Graphical Diagramm illustrating Simulation Outcome

Fig. 7 Dealing with the Impact of End User Fears.

2.3 Sensitivity Analysis and Reuse of Evaluation Information

Generally, results of a simulation enable PAIS engineers to gain insights into causal dependencies between organizational, technological and project-specific factors. This helps them to better understand resulting effects and to develop a concrete ”feeling” for the dynamic implications of EcoPOST evaluation models. To investigate how a given evaluation model ”works” and what might change its behavior, we simulate the dynamic implications described by it – a task which is typically too complex for the human mind. 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 its initial condition is changed. This procedure is also known as *sensitivity analysis*. Simulation outcomes can be further analyzed using graphical charts.

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 as well as the many causal dependencies that exist between them. Taking the approach described so far, each evaluation and simulation model has to be designed from scratch. Besides additional efforts, this results in an exclusion of existing modeling experience and prevents the reuse of both evaluation and simulation models. In response to this problem, Section 3 introduces a set of reusable *evaluation patterns* (EP). EPs do not only ease the design and simulation of evaluation models, but also enable the reuse of evaluation information. This is crucial to foster the practical applicability of the EcoPOST framework.

2.4 Other Approaches

To enable cost evaluation of BPM technology, numerous evaluation approaches exist – the most relevant ones are depicted in Fig. 2. All these approaches can be used in the context of BPM projects. However, only few of them address the specific challenges tackled by EcoPOST.

Activity-based costing (ABC) belongs to this category. It does not constitute an approach to unfold the dynamic effects triggered by causal dependencies in BPM projects. Instead, ABC provides a method for 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. Doing so, the scope of the business activities to be analyzed has to be identified in a first step (e.g., based on activity decomposition). Identified activities are then 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 its output 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. 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. Afterwards, 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. Following this, 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 be 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. Finally, the calculated activity unit costs and bills of activity are used for identifying candidates for business process improvement. In total, ABC is an approach to make costs related to business activities (e.g., business process steps) transparent. Therefore, aplying the method can be an accompanying step of EcoPOST in order to make certain cost and impact factors 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.

Besides, there are formalisms (no full evaluation approaches!) that can be applied to unfold the dynamic effects triggered by causal dependencies in BPM projects.

Causal *Bayesian Networks* (BN) [23], for example, promise to be a useful approach. 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. This allows to investigate the interplay of the components of a system and the effects resulting from this. BN do not allow to model feedback loops since cycles in BN would allow infinite feedbacks and oscillations that prevent stable parameters of the probability distribution.

Agent-based modeling provides another interesting approach. Resulting models comprise a set of reactive, intentional, or social agents encapsulating the behavior of the various variables that make up a system [6]. During simulation, the behavior of these agents is emulated according to defined rules [59]. System-level information (e.g., about intangible factors being effective in a BPM 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.

Another approach is provided by *fuzzy cognitive maps* (FCM) [17]. An FCM is a cognitive map appyling relationships between the objects of a ”mental landscape” (e.g., concepts, factors, or other resources) in order to compute the ”strength of impact” of these objects. To accomplish the latter task fuzzy logic is used. Most important, an FCM can be used to support different kinds of planning activities. As example consider (BPM) projects; for them, an FCM alows to analyze the mutual dependencies between respective project resources.

3 EcoPOST Evaluation Patterns

BPM projects often exhibit similarities, e.g., regarding the appearance of certain cost and impact factors. We pick up these similarities by introducing customizable patterns. This shall increase model reuse and facilitate practical use of the EcoPOST framework. *Evaluation patterns* (EPs) do not only ease the design and simulation of

evaluation models, but also enable the reuse of evaluation information [42, 35, 38]. This is crucial to foster practical applicability of the EcoPOST framework.
Specifically, we introduce an *evaluation pattern* (EP) as a predefined, but customizable EcoPOST model, i.e., an EP can be built based on the elements introduced in Section 2. An EP consists of an *evaluation model* and a corresponding *simulation model*. More precisely, each EP constitutes a template for a specific DCF or ImF as it typically exists in many BPM projects. Moreover, we distinguish between *primary* EPs (cf. Section 3.2) and *secondary* ones (cf. Section 3.3). A primary EP describes a DCF whereas a secondary EP represents an ImF. We denote an EP representing an ImF as secondary as it has a supporting role regarding the design of EcoPOST cost models based on primary EPs.

The decision whether to represent cost/impact factors as static or dynamic factors in EPs also depends on the model designer. Many cost and impact factors can be modeled both as static or dynamic factors. Consequently, EPs can be modeled in alternative ways. This is valid for all EPs discussed in the following.

Patterns were first introduced to describe best practices in architecture [1]. However, they have also a long tradition in computer science, e.g., 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 [66, 68, 56, 26] or inter-organizational control [24]. Generally, patterns describe solutions to recurring problems. They aim at supporting others in learning from available solutions and allow for the application of these solutions to similar situations. Often, patterns have a generative character. Generative patterns (like the ones we introduce) tell us how to create something and can be observed in the environments they helped to shape. Non-generative patterns, in turn, describe recurring phenomena without saying how to reproduce them.

Reusing System Dynamics models has been discussed before as well. Senge [61], Eberlein and Hines [15], Liehr [29], and Myrtveit [46] introduce generic structures (with slightly different semantics) satisfying the capability of defining ”components”. Winch [72], in turn, proposes a more restrictive approach 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 and customization.

3.1 Research Methodology and Pattern Identification

As sources of our patterns (cf. Tables 3 and 4) we consider results from surveys [41], case studies [35, 43], software experiments [44], and profound experiences we gathered in BPM projects. These projects addressed a variety of typical settings in enterprise computing which allows us to generalize our experiences.

To ground our patterns on a solid basis we first create a list of candidate patterns. For generating this initial list we conduct a detailed literature review and rely on

Pattern Name	Discussed in chapter	Survey	Case Study	Literature	Experiment	Experiences
Business Process Redesign Costs	**yes**	**x**	**x**	**x**	-	**x**
Process Modeling Costs	**yes**	-	-	**x**	**x**	**x**
Requirements Definition Costs	**yes**	-	**x**	**x**	-	**x**
Process Implementation Costs	**yes**	**x**	**x**	**x**	**x**	**x**
Process Adaptation Costs	**no**	**x**	**x**	**x**	**x**	**x**

Table 3 Overview of primary Evaluation Patterns and their Data Sources.

our experience with PAIS-enabling technologies, mainly in the automotive industry (e.g., [3, 34, 22]. Next we thoroughly analyze the above mentioned material to find empirical evidence for our candidate patterns. We then map the identified evaluation data to our candidate patterns and – if necessary – extend the list of candidate patterns.

Pattern Name	Discussed in chapter	Survey	Case Study	Literature	Experiment	Experiences
Process Knowledge	**yes**	**x**	-	**x**	**x**	**x**
Domain Knowledge	**yes**	**x**	-	**x**	**x**	**x**
Process Evolution	**yes**	**x**	-	**x**	-	**x**
Process Complexity	**yes**	-	-	**x**	-	-
Process Maturity	**no**	-	-	**x**	-	**x**
Work Profile Change	**no**	**x**	-	**x**	**x**	**x**
End User Fears	**no**	**x**	**x**	**x**	-	**x**

Table 4 Overview of Secondary Evaluation Patterns and their Data Sources.

A pattern is defined as reusable solution to a commonly occurring problem. We require each evaluation pattern to be observed at least three times in literature and our empirical research. Only those patterns, for which such empirical evidence exists, are included in the final list of patterns presented in the following.

3.2 Primary Evaluation Patterns

Business Process Redesign Costs. The EP shown in Fig. 8 deals with the costs of business process redesign activities. Prior to PAIS development such activities become necessary for several reasons. As examples consider the need to optimize business process performance or the goal of realizing a higher degree of process automation. This EP is based on our experiences (from 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 process information into a new process design. Process redesign costs are thereby assumed to be varying, i.e., they are represented as DCF.
This EP reflects our experience that six additional ImFs are of particular importance when investigating the costs of process redesign activities: ”Process Complexity”,

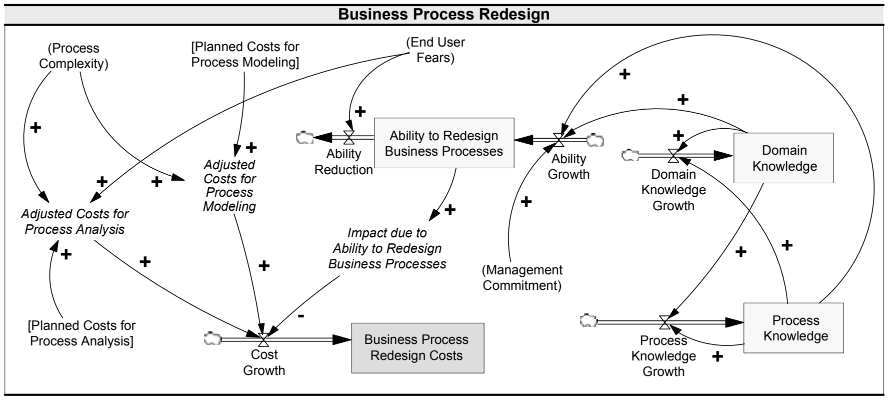

Business Process Redesign
(Process Complexity)
[Planned Costs for Process Modeling]
(End User Fears)
Ability to Redesign Business Processes
Ability Reduction
Ability Growth
Domain Knowledge
Domain Knowledge Growth
Adjusted Costs for Process Modeling
Adjusted Costs for Process Analysis
Impact due to Ability to Redesign Business Processes
(Management Commitment)
[Planned Costs for Process Analysis]
Cost Growth
Business Process Redesign Costs
Process Knowledge
Process Knowledge Growth
+
-

Fig. 8 Primary Evaluation Pattern: Business Process Redesign Costs.

”Management Commitment”, ”End User Fears”, ”Process Knowledge”, ”Domain Knowledge”, and ”Ability to Redesign Business Processes” (cf. Fig. 8). The importance of these factors is confirmed by results from one of our surveys. 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 Process Analysis”), process modeling costs are only influenced by ”Process Complexity” (as end users are typically not participating in the modeling process). Business process redesign costs are further influenced by a dynamic ImF ”Ability to Redesign Business Processes”, which, in turn, is influenced – according to our practical experiences – by four ImFs (causing the ImF ”Ability to Redesign Business Processes” to change): ”Management Commitment”, ”End User Fears”, ”Process Knowledge”, and ”Domain Knowledge”. Note that – if desired – the effects of the latter three ImFs can be further detailed based on available secondary EPs.

Process Modeling Costs. The EP depicted in Fig. 9 deals with the costs of process modeling activities in BPM projects. Such activities are typically accomplished to prepare the information gathered during process analysis, to assist software developers in implementing the PAIS, and to serve as guideline for implementing the new process design (in the organization). Generally, there exist many notations that can be used to specify process models. Our EP, for example, assumes that process models are expressed as *event-driven process chains* (EPC).

Basically, this EP (cf. Fig. 9) 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 EcoPOST impact rating [35]) and the dynamic ImF ”Process Knowledge” (which has been also confirmed by our survey described in [35]). The ImF ”Process Complexity” is not further discussed here. Instead, we refer to [35] where this ImF has been introduced in detail. The ImF ”Process Size”, in turn, is characterized based on (estimated) attributes of the

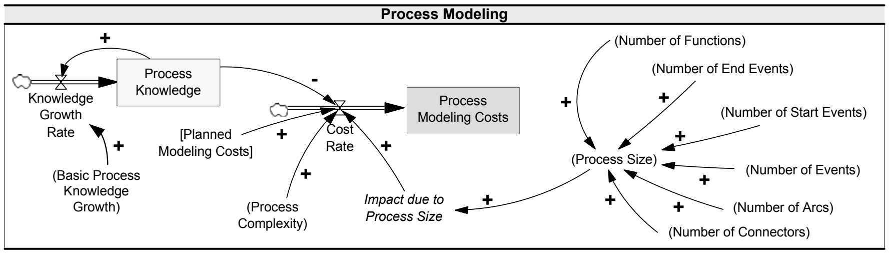


Fig. 9 Primary Evaluation Pattern: Process Modeling Costs.

process model to be developed. These attributes depend on the used modeling formalism. As aforementioned, the EP from Fig. 9 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 an increasing amount of process knowledge results in decreasing modeling costs).

Requirements Definition Costs. The EP from Fig. 10 deals with costs for defining and eliciting requirements [35]. It is based on the two DCFs ”Requirement Definition Costs” and ”Requirement Test Costs” as well as on the ImF ”Requirements to be Documented”. This EP reflects the observation we made in practice that the DCF ”Requirements Definition Costs” is determined by three main cost factors: costs of a requirements management tool, process analysis costs, and requirements documentation costs. Costs of 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”). ”Requirements Test Costs” are determined by two SCFs (”Costs for Test Tool” and ”Test Costs per Requirement”) and the dynamic ImF ”Requirements to be documented” (as only documented requirements need to be tested). Costs for a test tool and test costs per requirement are assumed to be constant (and are represented as SCFs).

Process Implementation Costs. The EP depicted in Fig. 11 deals with costs for implementing a process and their interference through impact factors [35]. An additional EP (not shown here) deals with the costs caused by adapting the process(es)

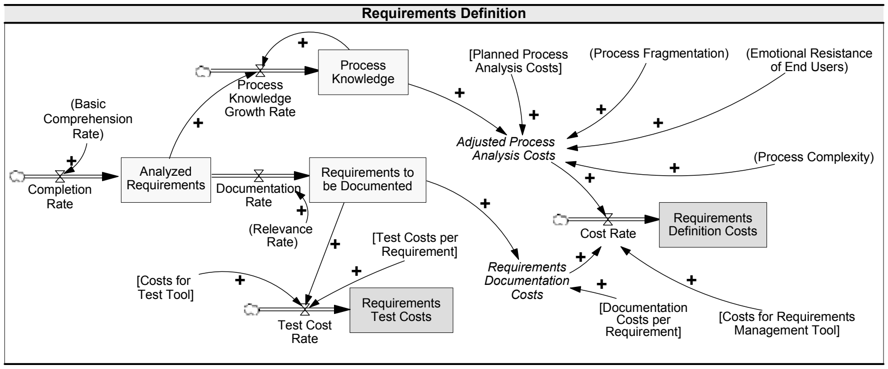

Requirements Definition
Process Knowledge
Process Knowledge Growth Rate
(Basic Comprehension Rate)
Completion Rate
Analyzed Requirements
Documentation Rate
Requirements to be Documented
(Relevance Rate)
[Planned Process Analysis Costs]
(Process Fragmentation)
(Emotional Resistance of End Users)
Adjusted Process Analysis Costs
(Process Complexity)
Cost Rate
Requirements Definition Costs
Requirements Documentation Costs
[Test Costs per Requirement]
[Costs for Test Tool]
Test Cost Rate
Requirements Test Costs
[Documentation Costs per Requirement]
[Costs for Requirements Management Tool]

Fig. 10 Primary Evaluation Pattern: Requirements Definition Costs.

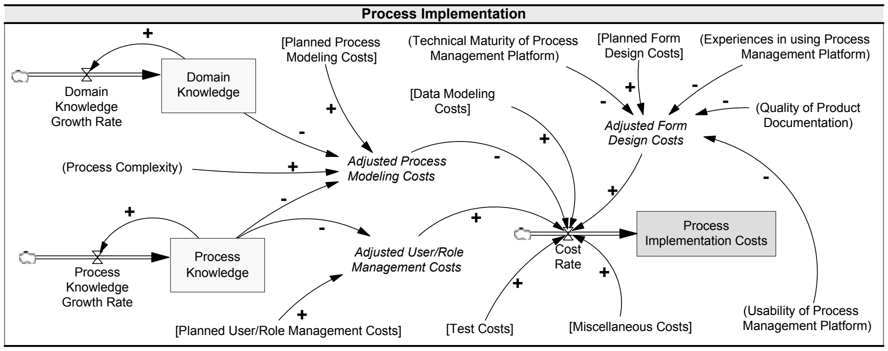

Process Implementation
Domain Knowledge Growth Rate
Domain Knowledge
[Planned Process Modeling Costs]
(Technical Maturity of Process Management Platform)
[Planned Form Design Costs]
(Experiences in using Process Management Platform)
[Data Modeling Costs]
(Quality of Product Documentation)
Adjusted Form Design Costs
(Process Complexity)
Adjusted Process Modeling Costs
Process Knowledge Growth Rate
Process Knowledge
Adjusted User/Role Management Costs
Cost Rate
Process Implementation Costs
[Planned User/Role Management Costs]
[Test Costs]
[Miscellaneous Costs]
(Usability of Process Management Platform)

Fig. 11 Primary Evaluation Pattern: Process Implementation Costs.

supported by an PAIS. This additional EP is identical to the previous EP ”Process Implementation Costs” – except for the additional ImF ”Process Evolution”.

Basic to this EP is our observation that the implementation of a process is determined by six main cost factors (cf. Fig. 11): ”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 ”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 this is considered as necessary, additional SCFs or DCFs can be introduced in order to make specific cost factors more explicit. Note that we have analyzed this EP in more detail in a controlled software experiment [44].

3.3 Secondary Evaluation Patterns

We now summarize secondary EPs. Unlike a primary EP describing a particular DCF, a secondary EP represents an ImF. Again, as pattern sources we consider results from two surveys [41], several case studies [35, 43], a controlled software experiment [44], and profound practical experiences gathered in BPM projects in the automotive and clinical domain [34, 28].

Process Knowledge. Fig. 12 shows an EP which specifies the ImF ”Process Knowledge”, i.e., causal dependencies on knowledge about the process(es) to be supported.

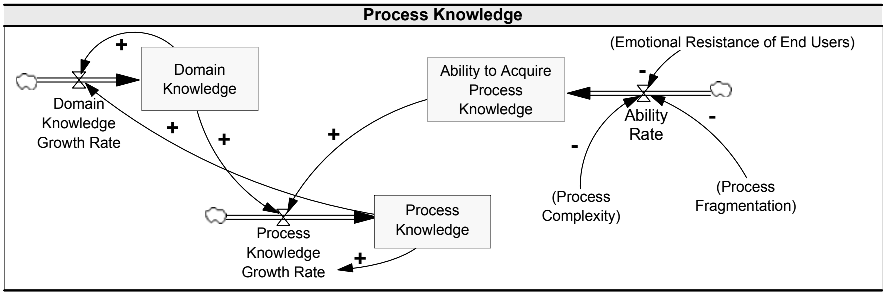

Process Knowledge
Domain Knowledge
Domain Knowledge Growth Rate
Ability to Acquire Process Knowledge
(Emotional Resistance of End Users)
Ability Rate
(Process Complexity)
(Process Fragmentation)
Process Knowledge
Process Knowledge Growth Rate
+
+
+
+
+
-
-
-

Fig. 12 Secondary Evaluation Pattern: Process Knowledge.

Process knowledge includes, for example, knowledge about process participants and their roles as well as knowledge about the flow of data. Acquiring process knowledge necessitates the ability to acquire process knowledge. 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”.

Domain Knowledge. The EP from Fig. 13 deals with the evolution of domain knowledge along the course of an BPM project. Our practical experiences allow for the conclusion that ”Domain Knowledge” is a dynamic ImF influenced by three other ImFs: the period an PAIS engineer is working in a specific domain (captured by the dynamic ImF ”Experience”), the dynamic ImF ”Process Knowledge” 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 BPM project (or a subsidiary activity).

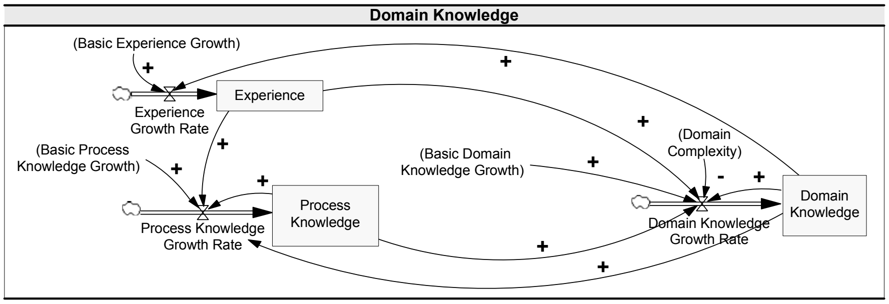

Domain Knowledge
(Basic Experience Growth)
Experience
Experience Growth Rate
(Basic Process Knowledge Growth)
Process Knowledge
Process Knowledge Growth Rate
(Basic Domain Knowledge Growth)
(Domain Complexity)
Domain Knowledge
Domain Knowledge Growth Rate
+
-

Fig. 13 Secondary Evaluation Pattern: Domain Knowledge.

Process Evolution. The EP depicted in Fig. 14 covers the static ImF ”Process Evolution”. Specifically, it describes drivers of process evolution. Basically, this EP reflects the assumption that business process evolution is caused by various drivers. Note that arbitrary drivers of evolution may be included in the EP.

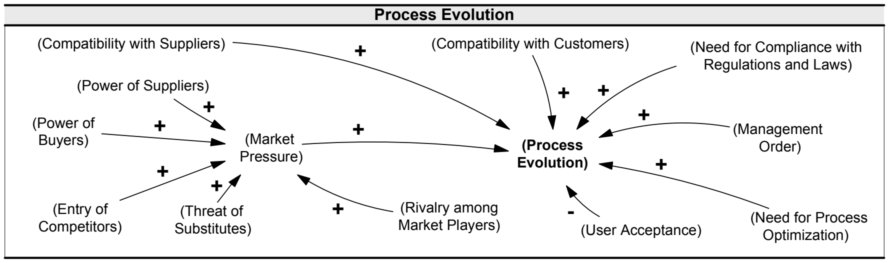


Fig. 14 Secondary Evaluation Pattern: Business Process Evolution.

Process Complexity. The EP from Fig. 15 deals with 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 [8]. Thus, we extend process complexity to ”Process Complexity / Process Model Complexity”. The EP from Fig. 15 further aligns with the assumption that respective process models are formulated using EPC notation. According to the depicted EP, 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”).

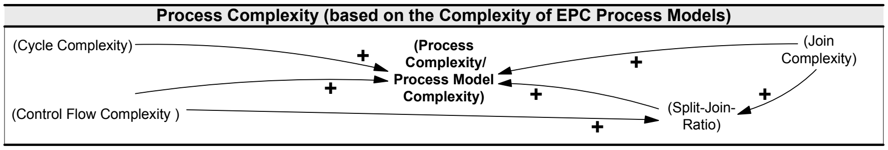

Process Complexity (based on the Complexity of EPC Process Models)
(Cycle Complexity)
(Process Complexity/ Process Model Complexity)
(Join Complexity)
(Control Flow Complexity)
(Split-Join-Ratio)
+
+
+
+
+
+

Fig. 15 Secondary Evaluation Pattern: Process Complexity.

The complexity driver ”Cycle Complexity” has been motivated in [9, 27]. Arbitrary cycles, for example, can lead to EPC models without clear semantics (cf. [25] for examples). The ImF ”Control-Flow Complexity” is characterized by [8]. 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” [19]. 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 [32, 33]. 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 is expressed by the static ImF ”Split-Join-Ratio” (= JC/CFC) [32, 33]. Based on these four static ImFs (or drivers of complexity), we derive the EP from Fig. 15. Thereby, an increasing cycle complexity results in higher process complexity. Also, both increasing CFC and increasing JC result in increasing process complexity. A JSR value different from 1 increases error probability and thus process complexity. It is noteworthy that – if desired – other drivers of process complexity may be considered as well. Examples can be found in [27, 33].

Process Maturity. The EP from Fig. 16 specifies the static ImF ”Process Maturity”. This EP is based on the assumption that increasing process maturity results in lower costs. This static ImF builds upon the 22 process areas of the *capability maturity model integration* (CMMI) [12], a process improvement approach providing organizations with elements of effective processes. The overall ImF ”Process Maturity” is determined by the maturity of the four categories of the CMMI continuous representation: ”Process Management”, ”Engineering”, ”Project Management”, and ”Support”. Each category is further detailed by CMMI process areas.

Work Profile Change. This EP (not shown here, but discussed in [35]) deals with the change of end user work profiles (and the effects of work profile changes). More specifically, it relates the perceived work profile change to changes emerging in the five job dimensions of Hackman’s *job characteristics model* [20, 21]: (1) *skill variety*, (2) *task identity*, (3) *task significance*, (4) *autonomy*, and (5) *feedback from the job*. For each of these five core job dimensions, the emerging change is designated based on the level before and after PAIS introduction.

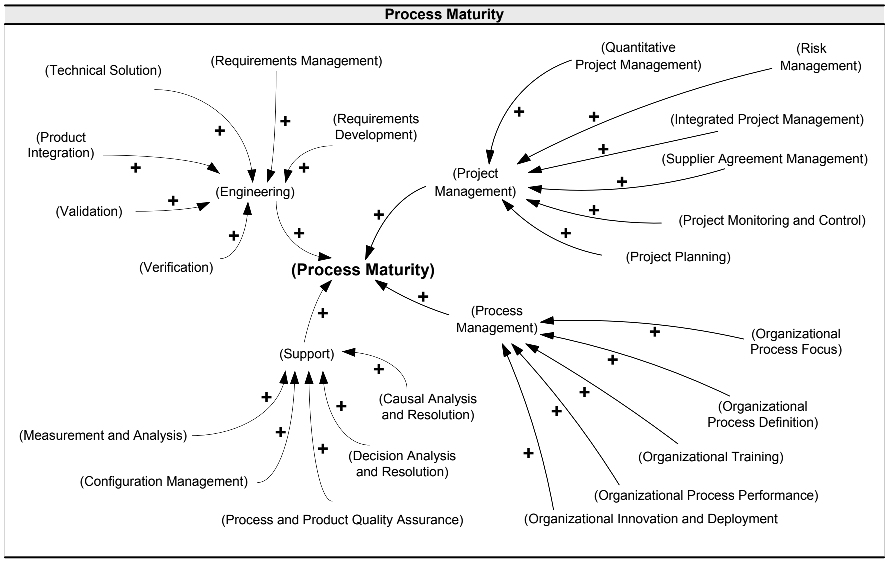

Process Maturity
(Technical Solution)
(Requirements Management)
(Quantitative Project Management)
(Risk Management)
(Requirements Development)
(Integrated Project Management)
(Product Integration)
(Supplier Agreement Management)
(Project Management)
(Engineering)
(Validation)
(Project Monitoring and Control)
(Project Planning)
(Verification)
(Process Maturity)
(Process Management)
(Organizational Process Focus)
(Support)
(Causal Analysis and Resolution)
(Organizational Process Definition)
(Measurement and Analysis)
(Decision Analysis and Resolution)
(Organizational Training)
(Configuration Management)
(Organizational Process Performance)
(Process and Product Quality Assurance)
(Organizational Innovation and Deployment
+

Fig. 16 Secondary Evaluation Pattern: Process Maturity (Continuous Representation).

End User Fears. This EP (not shown here, but discussed in [35] and [42]) is based on experiences which allow to conclude that the introduction of an PAIS may cause end user fears, e.g., due to work profile change (i.e., job redesign) or changed social clues. Such fears often lead, for example, to emotional resistance of end users. This, in turn, can make it difficult to get the needed support from end users, e.g., during process analysis.

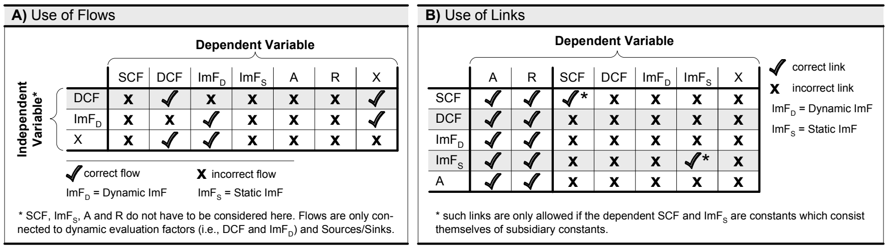
All primary and secondary EPs discussed can be considered as suggestions making similarities in BPM projects explicit. Thus they serve as a baseline and starting point for building more complex evaluation and simulation models.

4 Model Design Rules

Overall benefit of EcoPOST evaluation models depends on their quality. The latter, in turn, is determined by the syntactical as well as the semantical correctness of the evaluation model. Maintaining correctness of an evaluation model, however, can be a difficult task to accomplish.

4.1 Modeling Constraints for Evaluation Models

Rules for the correct use of flows and links are shown in Fig. 17A and Fig. 17B. By contrast, Fig. 18A – Fig. 18F show examples of *incorrect* models.



A) Use of Flows

Dependent Variable

Independent Variable*	SCF	DCF	ImF_D	ImF_S	A	R	X
DCF	x	✓	x	x	x	x	✓
ImF_D	x	x	✓	x	x	x	✓
X	x	✓	✓	x	x	x	x

✓ correct flow x incorrect flow

ImF_D = Dynamic ImF ImF_S = Static ImF

* SCF, ImF_S, A and R do not have to be considered here. Flows are only connected to dynamic evaluation factors (i.e., DCF and ImF_D) and Sources/Sinks.

B) Use of Links

Dependent Variable

	A	R	SCF	DCF	ImF_D	ImF_S	X
SCF	✓	✓	✓*	x	x	x	x
DCF	✓	✓	x	x	x	x	x
ImF_D	✓	✓	x	x	x	x	x
ImF_S	✓	✓	x	x	x	✓*	x
A	✓	✓	x	x	x	x	x

✓ correct link

x incorrect link

ImF_D = Dynamic ImF

ImF_S = Static ImF

* such links are only allowed if the dependent SCF and ImF_S are constants which consist themselves of subsidiary constants.

Fig. 17 Using Flows and Links in our Evaluation Models.

Dynamic evaluation factors, for example, may be only influenced by flows and not by links as shown in Fig. 18A. Likewise, flows must be not connected to auxiliary variables or constants (cf. Fig. 18B). Links pointing from DCFs (or auxiliary variables) to SCFs or static ImFs (cf. Fig. 18C and Fig. 18D) are also not valid as SCFs as well as static ImFs have constant values which cannot be influenced. Finally, flows and links connecting DCFs with dynamic ImFs (and vice versa) are also not considered as correct (cf. Fig. 18E and Fig. 18F).

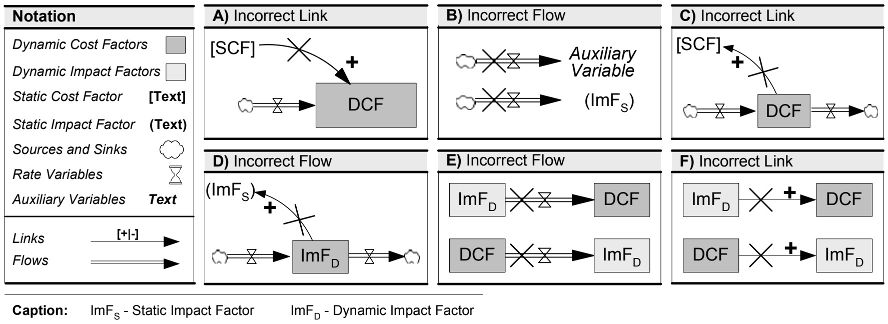


Fig. 18 Examples of Incorrect Modeling.

Several other constraints have to be taken into account as well when designing evaluation models. In the following let EM = (V, F, L) be an evaluation model (cf. Definition 2.1). Then:

Design Rule 1 **(Binary Relations)** Every model variable must be used in at least one binary relation. Otherwise, it is not part of the analyzed evaluation context and can be omitted:

$$\forall v \in (S \mathbin{\dot{\cup}} X) : \exists q \in (S \mathbin{\dot{\cup}} X) \wedge ((v,q) \in F \vee (v,q) \in F) \quad (1)$$

$$\forall v \in (A \mathbin{\dot{\cup}} C) : \exists q \in (A \mathbin{\dot{\cup}} R) \wedge \exists\ (q,v,[+|-]) \in L \quad (2)$$

Design Rule 2 **(Sources and Sinks)** Every state variable must be connected to at least one source, sink or other state variable. Otherwise it cannot change its value and therefore would be useless:

$$\forall v \in S : \exists q \in (S \dot\cup X) \wedge ((v,q) \in F \vee (q,v) \in F) \tag{3}$$

Design Rule 3 **(Rate Variables)** Every rate variable is influenced by at least one link; otherwise the variable cannot change and therefore is useless:

$$\forall v \in R : \exists q \in (S \dot\cup A \dot\cup C) \wedge \exists\ (q,v,[+|-]) \in L \tag{4}$$

Design Rule 4 **(Feedback Loops)** There are no cycles consisting only of auxiliary variables, i.e., cyclic feedback loops must at least contain one state variable (cycles of auxiliary variables cannot be evaluated if an evaluation model is simulated):

$$\begin{gathered}\neg\exists < q_0, q_1, ..., q_r > \in A^{r+1}\ with\ (q_i, q_{i+1}, [+|-]) \in L\ for \\ i = 0,\dots,r-1 \wedge q_0 = q_r \wedge\ q_k \neq q_l\ for\ \ k,l = 1,\dots,r; k \neq l\end{gathered} \tag{5}$$

Design Rule 5 **(Auxiliary Variables)** An auxiliary variable has to be influenced by at least two other static or dynamic evaluation factors or auxiliary variables (except for auxiliary variables used to represent table functions [35]):

$$\forall v \in A : \exists p,q \in (A \dot\cup S \dot\cup C) \wedge ((q,v,[+|-]) \in L \wedge (p,v,[+|-]) \in L) \tag{6}$$

These modeling constraints provide basic rules for EcoPOST users to construct syntactically correct evaluation models.

4.2 Semantical Correctness of Evaluation Models

While syntactical model correctness can be ensured, this is not always possible for the semantical correctness of evaluation models. Yet, we can provide additional model design rules increasing the meaningfulness of our evaluation models.

Design Rule 6 **(Transitive Dependencies)** Transitive link dependencies (i.e., indirect effects described by chains of links) are restricted. As example consider Fig. 19. Fig. 19A reflects the assumption that increasing end user fears result in increasing emotional resistance. This, in turn, leads to increasing business process costs. Consequently, the modeled transitive dependency between ”End User Fears” and ”Business Process Redesign Costs” is not correct, as increasing end user fears do not result in decreasing business process (re)design costs. The correct transitive dependency is shown in Fig. 19B. Fig. 19C illustrates the assumption that increasing process knowledge results in an increasing ability to (re)design business processes. An increasing ability to (re)design business processes, in turn, leads to decreasing process definition costs. The modeled transitive dependency between ”Process

Knowledge” and ”Process Definition Costs”, however, is not correct, as increasing process knowledge does not result in increasing process definition costs (assuming that the first 2 links are correct). See Fig. 19D for the correct transitive dependency.

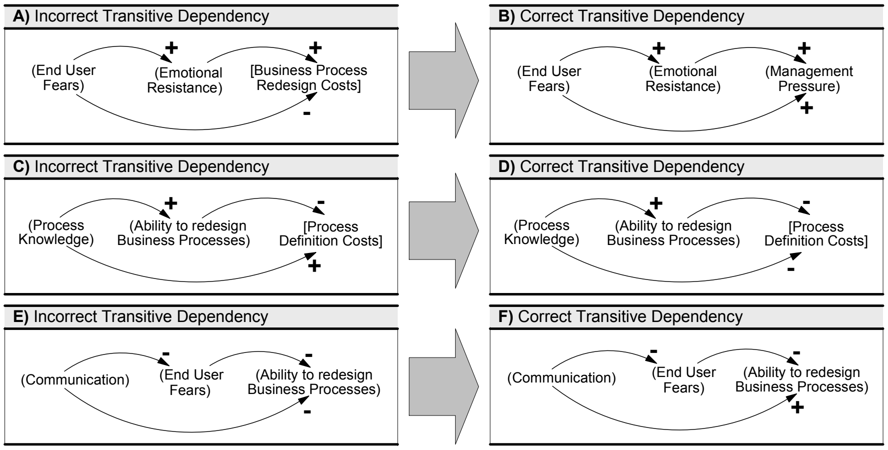

A) Incorrect Transitive Dependency
(End User Fears)
(Emotional Resistance)
[Business Process Redesign Costs]
+
+
-
B) Correct Transitive Dependency
(End User Fears)
(Emotional Resistance)
(Management Pressure)
+
+
+
C) Incorrect Transitive Dependency
(Process Knowledge)
(Ability to redesign Business Processes)
[Process Definition Costs]
+
-
+
D) Correct Transitive Dependency
(Process Knowledge)
(Ability to redesign Business Processes)
[Process Definition Costs]
+
-
-
E) Incorrect Transitive Dependency
(Communication)
(End User Fears)
(Ability to redesign Business Processes)
-
-
-
F) Correct Transitive Dependency
(Communication)
(End User Fears)
(Ability to redesign Business Processes)
-
-
+

Fig. 19 Transitive Dependencies (Simplified Evaluation Models).

Finally, Fig. 19E deals with the impact of communication (e.g., the goals of an PAIS project) on the ability to redesign business processes. Yet, the transitive dependency shown in Fig. 19E is not correct. The correct one is shown in Fig. 19F.

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*. More precisely, transitive dependencies have to comply with the following three *multiplication laws* for transitive dependencies (for any $x, y \in \{+,-\}$):

$$+ * y = y \tag{7}$$

$$- * - = + \tag{8}$$

$$x * y = y * x \tag{9}$$

The evaluation models shown in Fig. 19A and Fig. 19C violate Law 1, whereas the model shown in Fig. 19E violates the second one. Law 3 states that 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:

$$\forall v \in C, \forall q \in A : \neg\exists\ l_1, l_2 \in L\ withl_1 = (v,q,-) \wedge l_2 = (v,q,+) \tag{10}$$

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

$$\forall v \in S, \forall q \in A : \neg\exists\, l_1, l_2 \in L\ with l_1 = (v,q,-) \wedge l_2 = (v,q,+) \qquad (11)$$

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

$$\forall v \in A, \forall q \in A : \neg\exists\, l_1, l_2 \in L\ with l_1 = (v,q,-) \wedge l_2 = (v,q,+) \qquad (12)$$

Finally, there exist two additional simple constraints:

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

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

Without providing model design rules, incorrect evaluation models can be quickly modeled. This, in turn, does not only aggravate the derivation of plausible evaluations, but also hampers the use of the modeling and simulation tools [42] which have been developed as part of the EcoPOST framework.

5 Modeling Guidelines

To facilitate the use of our methodology, governing guidelines and best practices are provided. This section summarizes two categories of EcoPOST governing guidelines: (1) guidelines for evaluation models and (2) guidelines for simulation models.

In general, EcoPOST evaluation models can become large, e.g., due to a potentially high number of evaluation factors to be considered or due to the large number of causal dependencies existing between them. To cope with this complexity, we introduce guidelines for designing evaluation models (cf. Table 5). Their derivation is based on experiences we gathered during the development of our approach, its initial use in practice, and our study of general System Dynamics (SD) guidelines [63]. As example consider guideline EM-1 from Table 5. The distinction between SCFs and DCFs is a fundamental principle in the EcoPOST framework. Yet, it can be difficult for the user to decide whether a cost factor shall be considered as static or dynamic. As example take an evaluation scenario which deals with the introduction of a new PAIS ”CreditLoan” to support the granting of loans at a bank. Based on the new PAIS, the entire loan offer process shall be supported. For this purpose, the PAIS has to leverage internal (i.e., within the bank) and external (e.g., a 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 SCFs as they can be clearly quantified based on a contract or service agreement. If integration is done in-house, however, integration costs should be represented as DCFs as costs might be influenced by additional ImFs in this case. Other guidelines are depicted in Table 5.

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

Table 5 Guidelines for Designing Evaluation Models.

To simulate EcoPOST evaluation models constitutes another complex task. The guidelines from Table 6 are useful to deal with it. Guideline SM-7, for example, claims to assess the usefulness of an evaluation model and related simulation results always in comparison with mental or descriptive models needed or used otherwise. In our experience, there often exists controversy on the question whether an evaluation model meets reality. However, such controversies miss the first purpose of a model, namely to provide insights that can be easily communicated.

GL	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 for model variables.
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 6 Guidelines for Developing Simulation Models.

The governing guidelines and best practices represent a basic set of clues and recommendations for users of the EcoPOST framework. They support the modeler in designing evaluation models, in building related simulation models, and in handling dynamic evaluation factors. 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 guidelines increases the probability of developing meaningful models.

6 Practical Impact

We applied the EcoPOST framework to a complex BPM project from the automotive domain. We investigate cost overruns observed during the introduction of a large information system for supporting the development of *electrical and electronic (E/E) systems* (e.g., a multimedia unit in the car). Based on real project data, interviews with project members (e.g., requirements engineers, software architects, software developers), online surveys among end users, and practical experiences gathered in the respective BPM project, we developed a set of EcoPOST evaluation models and analyzed these models using simulation. Due to space limitations we cannot describe the complete case study in detail (for details see [35]).

6.1 The Case

An initial business case for the considered BPM project was developed prior to project start in order to convince senior management to fund the project. This business case was based on data about similar projects provided by competitors (evaluation by analogy) as well as on rough estimates on planned costs and assumed benefits of the project. The business case comprised six main cost categories: (1) *project management*, (2) *process management*, (3) *IT system realization*, (4) *specification and test*, (5) *roll-out and migration*, and (6) *implementation of interfaces*.

	Description
1	***Process Management Costs***: This category deals with costs related to the (re)design of the business processes to be supported. This includes both the definition of new and the redesign of existing processes. As example of a process to be newly designed consider an E/E data provision process to obtain needed product data. As example of an existing process to be 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.
2	***IT System Realization Costs***: This category deals with costs for implementing the new PAIS on top of process management technology. In our case study, we focus on the analysis of costs related to the use of the process management system, e.g., 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.
3	***III. Online Surveys among End Users***: We conduct two online surveys among two user groups of the new PAIS (altogether 80 survey participants). 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. Goal of the survey is to confirm the significance of selected ImFs like ”End User Fears” and ”Emotional Resistance of End Users”.
IV	***Specification and Test Costs***: This cost category sums up costs for specifying the functionality of the PAIS 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 on whether requirements are met by the PAIS.

Table 7 Analyzed Cost Categories.

In a first project review (i.e., measurement of results), it turned out that 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 analyzed cost overruns in three cost categories using the EcoPOST methodology (cf. Table 7).

To be able to build evaluation and simulation models for the three analyzed cost categories, we collected data. This data is based on four information sources (cf. Table 8), which allowed 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 enables us to spot important causal dependencies between cost and impact factors and to derive evaluation models.

	Description
I	***Project Data***: A first data source is available project data; e.g., estimates about planned costs from the initial business case. Note that we did not participate in the generation of this business case.
II	***Interviews***: We interview 10 project members (2 software architects, 4 software developers, 2 usability engineers, and 2 consultants participating in 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.
III	***Online Surveys among End Users***: We conduct two online surveys among two user groups of the new PAIS system (altogether 80 survey participants). 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. Goal of the survey is to confirm the significance of selected ImFs like ”End User Fears” and ”Emotional Resistance of End Users”.
IV	***Practical Experiences***: Finally, our evaluation and simulation models also build upon practical experiences we gathered when participating in the investigated BPM 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.

Table 8 Data Collection.

6.2 Lessons Learned

Based on the derived evaluation models and simulation outcomes, we were able to show that costs as estimated in the initial business case were not realistic. The simulated costs for each analyzed cost category exceeded the originally estimated ones. Moreover, our evaluation models provided valuable insights into the reasons for the occurred cost overruns, particularly into causal dependencies and resulting effects on the costs of the analyzed BPM project.

LL	Description
LL-1	Our case study confirms that the EcoPOST framework enables PAIS engineers to gain valuable insights into causal dependencies and resulting cost effects in BPM projects.
LL-2	EcoPOST evaluation models are useful for domain experts and can support IT managers and policy makers in understanding an BPM project and decision-making.
LL-3	BPM projects are complex socio-technical feedback systems which are characterized by a strong nexus of organizational, technological, and project-specific parts. Hence, all evaluation models include feedback loops.
LL-4	Our case study confirms that evaluation models can become complex due to the large number of potential SCFs, DCFs and ImFs as well as the many causal dependencies existing between them. Governing guidelines (cf. Section 5) help to avoid too complex evaluation models.
LL-5	Though our simulation models have been build upon data derived from four different data sources, it has turned out that it is inevitable to rely on hypotheses to build simulation models.

Table 9 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 five lessons learned (cf. Table 9).

6.3 Critical Success Factors

We applied the EcoPOST framework in several case studies in the automotive domain. This has made us aware of a number of *critical success factors* which foster the transfer of the EcoPOST framework into practice.

First, it is important that EcoPOST users get enough time to become familiar with the provided evaluation concepts. Note that EcoPOST exhibits a comparatively large number of different concepts and tools, such that it will need some time to effectively apply them. In practice, this can be a 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 not only enables their later reuse, it also allows to reflect on past evaluations and lessons learned as well as to reuse evaluation data. For that purpose, the *EcoPOST Cost Benefit Analyzer* can be used, which is a tool we developed to support the use of EcoPOST [35]. For example, it enables storage of complete evaluation scenarios, i.e., evaluation models and their related simulation models.

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 evolution process.

Finally, the use of EcoPOST has shown that 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 as well as the many causal dependencies that exist between them. Evaluation models we developed to analyze a large BPM project in the automotive domain, for example, comprise more than ten DCFs and ImFs and more than 25 causal dependencies [35]. Besides additional efforts, this results in an exclusion of existing modeling experience, and prevents the reuse of both evaluation and simulation models. In response to this problem, we introduced reusable *evaluation patterns*.

7 Discussion

Evaluating IT investments regarding their costs, benefits, and risks is a complex task to accomplish. Even more complex are respective evaluations of BPM projects. In the previous chapters, we have shown that this difficulty stems from the interplay of the numerous technology-, organization-, and project-specific factors which arise in the context of BPM projects. In order to deal with this challenge, we have 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 BPM projects.

But are the evaluation concepts underlying the EcoPOST framework really suitable to capture causal dependencies and related effects? 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 section 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 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 BPM 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 BPM 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 BPM projects. They are also unable to cope with the causal dependencies and interactions that exist between these factors and the resulting effects. Even process-oriented approaches such as the TSTS approach or the hedonic wage model (see [45] for details) do not allow to address these issues, i.e., they can only be applied to evaluate the impact of a PAIS and BPM technology 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 BPM 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 BPM 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 BPM projects. Our EcoPOST framework 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 guiding the development of the EcoPOST framework.

Requirements Discussion. More generally, we have identified various requirements for the design of an economic-driven evaluation approach for BPM projects: both requirements for performing economic-driven IT evaluation in general and more specific requirements for evaluating BPM projects. 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 BPM 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, strongly depends on the availability of adequate[1] 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*” [63]). As examples for respective actions consider the compliance of evaluation models with defined model design rules 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 BPM 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 BPM 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). 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 been identified as relevant requirement (R-7). Considering our clearly specified

[1] We will denote evaluation data as ”adequate”, if the date 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.

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. Finally, we considered *tool-support* as crucial (R-11). In response to this, we 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 discussed, we use *System Dynamics* (SD) for specifying our evaluation models. SD is a formalism for studying and modeling complex feedback systems, e.g. biological, environmental, industrial, business, and social systems [54, 47]. 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 parts[2].

In BPM 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 BPM 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).

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 BPM 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,

[2] System Dynamics can be easily confounded with *Systems Thinking* [71]. 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 both approaches in the context of the evolution of the systems sciences is given in [60].

making it very difficult to derive conclusions regarding the costs of BPM projects. More specifically, the benefits of our approach can be summarized as follows:

- ***Feedback***: 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 BPM projects. Thus, evaluation models can be easily communicated to decision makers [3].
- ***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***: BPM 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 BPM 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 significant 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 BPM projects.

8 Summary

Though existing economic-driven IT evaluation and software cost estimation approaches have received considerable attention during the last decades, it is difficult to apply existing approaches to BPM projects. Reason is that existing approaches are unable to take into account the dynamic evolution of BPM projects caused by

[3] In [2] and [64], this benefit is confirmed for the management of software projects.

the numerous technological, organizational and project-specific facets influencing BPM projects. It is this evolution which often leads to complex and unexpected cost effects in BPM projects making even rough cost estimations a challenge. What is therefore needed is a comprehensive approach enabling BPM professionals to systematically investigate the costs of BPM projects. In response, this chapter takes a look at both well-known and often unknown cost factors in BPM projects, shortly discusses existing IT evaluation and software cost estimation approaches with respect to their suitability to be used in BPM projects, and finally introduces the EcoPOST framework. This new framework utilizes evaluation models to describe the interplay of technological, organizational, and project-specific BPM cost factors, and simulation concepts to unfold the dynamic behavior and costs of BPM projects.

Finally, there is a number of *critical success factors* (CSF) which foster the transfer of the EcoPOST framework into practice. *First*, it is important that 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, BPM architects, software developers, and consultants have the opportunity to contribute to the model development process.

References

1. C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language*. Oxford University Press, 1979.
2. 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.
3. 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.
4. 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.
5. D. F. Boyd and H. S. Krasnow. *Economic Evaluations of Management Information Systems*. IBM Systems Journal, 2(1), pp.2-23, 1963.
6. 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.
7. E. Brynjolfsson. *The Productivity Paradox of Information Technology*. Comm. of the ACM, 36(12), pp.66-77, 1993.
8. J. Cardoso. *Control-flow Complexity Measurement of Processes and Weyuker's Properties*. Proc. Int'l. Enformatika Conference, Volume 8, pp.213-218, 2005.

9. 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), pp.115-126, 2006.
10. N. G. Carr. *IT doesn't Matter*. Harvard Business Review (HBR), 5, 2003.
11. N. G. Carr. *Does IT matter?* Harvard Business Press, 2004.
12. CMMi. *Capability Maturity Model Integration*. Software Engineering Institute, 2006.
13. P. Dadam and M. Reichert. The ADEPT project: A decade of research and development for robust and flexible process support - challenges and achievements. *Computer Science - Research and Development*, 23(2):81–97, 2009.
14. 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.
15. R. J. Eberlein and J. H. Hines. *Molecules for Modelers*. Proc. 14th SD Conference, 1996.
16. J. W. Forrester. *Industrial Dynamics*. Productivity Press, 1961.
17. M. Glykas. *Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools and Applications (Studies in Fuzziness and Soft Computing)*. Springer, 2010.
18. 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), Big Island, Hawaii, 2005.
19. V. Gruhn and R. Laue. *Complexity Metrics for Business Process Models*. Proc. 9th Int'l. Conf. on Business Information Systems (BIS '06), 2006.
20. R. J. Hackman and G. R. Oldham. *Development of the Job Diagnostic Survey*. Journal of Applied Psychology, 60(2), pp.159-170, 1975.
21. 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.
22. A. Hallerbach, T. Bauer, and M. Reichert. Capturing variability in business process models: The Provop approach. *Journal of Software Maintenance and Evolution: Research and Practice*, 22(6-7):519–546, 2010.
23. F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2002.
24. V. Kartseva, J. Hulstijn, Y.-H. Tan, and J. Gordijn. *Towards Value-based Design Patterns for Inter-Organizational Control*. Proc. 19th Bled E-Commerce Conference, 2006.
25. E. Kindler. *On the Semantics of EPCs: Resolving the Vicious Circle*. Data Knowledge Engineering, 56(1), pp.23-40, 2006.
26. A. Lanz, B. Weber, and M. Reichert. Workflow time patterns for process-aware information systems. In *Proceedings Enterprise, Business-Process, and Information Systems Modelling: 11th Int'l Workshop BPMDS and 15th Int'l Conf, EMMSAD at CAiSE 2010*, volume 50 of *LNBIP*, pages 94–107, 2010.
27. A. Latva-Koivisto. *Finding a Complexity Measure for Business Process Models*. Research Report, Helsinki University of Technology, 2001.
28. R. Lenz and M. Reichert. IT support for healthcare processes - premises, challenges, perspectives. *Data and Knowledge Engineering*, 61(1):39–58, 2007.
29. M. Liehr. *A Platform for System Dynamics Modeling - Methodologies for the Use of Predefined Model Components*. Proc. 20th Int'l. System Dynamics Conference, 2002.
30. J. H. Lim, V. J. Richardson, and T. L. Roberts. *Information Technology Investment and Firm Performance: A Meta-Analysis*. Proc. 37th Hawaii Int'l. Conf. on System Sciences (HICSS '04), Big Island, Hawaii, 2004.
31. L. Meineke. *Wirtschaftlichkeitsanalysen - Basis zur Durchfuehrung von IT-Projekten*. IT Management, Volume 12, pp.12-17, 2003.
32. 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.
33. 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.
34. D. Müller, J. Herbst, M. Hammori, and M. Reichert. It support for release management processes in the automotive industry. In *Proc. 4th Int'l Conf. on Business Process Management (BPM'06)*, volume 4102 of *LNCS*, pages 368–377, 2006.

35. B. Mutschler. *Analyzing Causal Dependencies on Process-aware Information Systems from a Cost Perspective*. PhD Thesis, University of Twente, 2008.
36. 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, 2006.
37. B. Mutschler and M. Reichert. On modeling and analyzing cost factors in information systems engineering. In *Proc. 20th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'08)*, volume 5074 of *LNCS*, pages 510–524, 2008.
38. B. Mutschler and M. Reichert. Evaluation patterns for analyzing the costs of enterprise information systems. In *Proc. 21st Int'l Conf. on Advanced Information Systems (CAiSE'09)*, volume 5565 of *LNCS*, pages 379–394, 2009.
39. B. Mutschler, M. Reichert, and J. Bumiller. *Towards an Evaluation Framework for Business Process Integration and Management*. 2nd Int'l. Workshop on Interoperability Research for Networked Enterprises Applications and Software (INTEROP), Enschede, The Netherlands, 2005.
40. B. Mutschler, M. Reichert, and J. Bumiller. Designing an economic-driven evaluation framework for process-oriented software technologies. In *Proc. 28th Int'l Conf. on Software Engineering (ICSE'06)*, pages 885–888, 2006.
41. 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*, 38(3):280–291, 2008.
42. B. Mutschler, M. Reichert, and S. Rinderle. Analyzing the dynamic cost factors of process-aware information systems: A model-based approach. In *Proc. 19th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'07)*, volume 4495 of *LNCS*, pages 589–603, 2007.
43. 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. In *Proc. 8th Int'l. BPMDS Workshop at CAiSE 2007*, pages 379–384, 2007.
44. B. Mutschler, B. Weber, and M. Reichert. Workflow management versus case handling: Results from a controlled software experiment. In *Proc. ACM SAC 2008*, pages 82–89, 2008.
45. B. Mutschler, N. Zarvic, and M. Reichert. *A Survey on Economic-driven Evaluations of Information Technology*. Technical Report, TR-CTIT-07, University of Twente, 2007.
46. M. Myrtveit. *Object-oriented Extensions to System Dynamics*. Proc. 18th Int'l. System Dynamics Conference, 2000.
47. K. Ogata. *System Dynamics*. Prentice Hall, 2003.
48. T. Pisello. *IT Value Chain Management - Maximizing the ROI from IT Investments: Performance Metrics and Management Methodologies Every IT Stakeholder Should Know*. Alinean, LLC, available at: www.alinean.com, 2003.
49. M. Reichert, S. Rinderle, U. Kreher, and P. Dadam. Adaptive process management with ADEPT2. In *Proc. Int'l Conf. on Data Engineering (ICDE'05)*, 2005.
50. M. Reichert, S. Rinderle-Ma, and P. Dadam. Flexibility in process-aware information systems. *LNCS Transactions on Petri Nets and Other Models of Concurrency (ToPNoC), Special Issue on Concurrency in Process-aware Information Systems*, 2:115–135, 2011.
51. D. J. Reifer. *Making the Software Business Case - Improvement by the Numbers*. Addison-Wesley, 2001.
52. H. Reijers, S. van Wijk, B. Mutschler, and M. Leurs. *BPM in Practice: Who is doing what?* Proc. 8th Int'l. Conf. on Business Process Management (BPM 2010), LNCS 6336, pp. 45-60, Hoboken, USA, 2010.
53. H. A. Reijers and W. M. P. van der Aalst. *The Effectiveness of Workflow Management Systems - Predictions and Lessons Learned*. Int'l. J. of Inf. Mgmt., 25(5), pp.457-471, 2005.
54. G. P. Richardson and A. L. Pugh. *System Dynamics - Modeling with DYNAMO*. Productivity Press, 1981.
55. S. Rinderle, B. Weber, M. Reichert, and W. Wild. Integrating process learning and process evolution - a semantics based approach. In *Proc. Int'l Conf. on Business Process Management (BPM'05)*, volume 3649 of *LNCS*, pages 252–267, 2005.

56. S. Rinderle-Ma, M. Reichert, and B. Weber. On the formal semantics of change patterns in process-aware information systems. In *Proc. 27th Int'l Conference on Conceptual Modeling (ER'08)*, volume 5231 of *LNCS*, pages 279–293. Springer, 2008.
57. P. G. Sassone. *Cost-Benefit Methodology for Office Systems*. ACM Transactions on Office Information Systems, 5(3), pp.273-289, 1987.
58. 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, Palo Alto, 1988.
59. 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.
60. M. Schwaninger. *System Dynamics and the Evolution of Systems Movement - A Historical Perspective*. Discussion Paper, Nr. 52, University of St. Gallen, Switzerland, 2005.
61. P. M. Senge. *The 5th Discipline - The Art and Practice of the Learning Organization (1st Edition)*. Currency Publications, 1990.
62. H Smith and P. Fingar. *IT doesn't matter - Business Processes Do*. Meghan Kiffer Press, 2003.
63. J. D. Sterman. *Business Dynamics - Systems Thinking and Modeling for a Complex World*. McGraw-Hill, 2000.
64. 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.
65. Ventana Systems. *Vensim:*. http://www.vensim.com/, 2006.
66. W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. *Advanced Workflow Patterns*. Proc. 7th CoopIS, LNCS 1901, pp.18-29, 2000.
67. B. Weber, B. Mutschler, and M. Reichert. Investigating the effort of using business process management technology: Results from a controlled experiment. *Science of Computer Programming*, 75(5):292–310, 2010.
68. B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features - enhancing flexibility in process-aware information systems. *Data and Knowledge Engineering*, 66(3):438–466, 2008.
69. B. Weber, M. Reichert, W. Wild, and S. Rinderle-Ma. Providing integrated life cycle support in process-aware information systems. *Int'l Journal of Cooperative Information Systems*, 18(1):115–165, 2009.
70. B. Weber, S. Sadiq, and M. Reichert. Beyond rigidity - dynamic process lifecycle support: A survey on dynamic changes in process-aware information systems. *Computer Science - Research and Development*, 23(2):47–65, 2009.
71. G. M. Weinberg. *An Introduction to General Systems Thinking*. Dorset House Publishing, 2001.
72. 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.