

# Investigating the Effort of Using Business Process Management Technology: Results from a Controlled Experiment

Barbara Weber <sup>a,\*</sup>, Bela Mutschler <sup>b</sup> Manfred Reichert <sup>c</sup>

<sup>a</sup>*Department of Computer Science, University of Innsbruck,  
Technikerstraße 21a, 6020 Innsbruck, Austria*

<sup>b</sup>*Hochschule Ravensburg-Weingarten  
Postfach 1261, 88241 Weingarten, Germany*

<sup>c</sup>*Institute of Databases and Information Systems, Ulm University  
James-Franck-Ring, 89069 Ulm, Germany*

---

## Abstract

Business Process Management (BPM) technology has become an important instrument for supporting complex coordination scenarios and for improving business process performance. When considering its use, however, enterprises typically have to rely on vendor promises or qualitative reports. What is still missing and what is demanded by IT decision makers are quantitative evaluations based on empirical and experimental research. This paper picks up this demand and illustrates how experimental research can be applied to technologies enabling enterprises to coordinate their business processes and to associate them with related artifacts and resources. The conducted experiment compares the effort for implementing and maintaining a sample business process either based on standard workflow technology or on a case handling system. We motivate and describe the experimental design, discuss threats for the validity of our experimental results (as well as risk mitigations), and present the results of our experiment. In general, more experimental research is needed in order to obtain valid data on the various aspects and effects of BPM technology and BPM tools.

*Key words:* Process-aware Information System, Workflow Management, Case Handling, Controlled Experiment, Information Systems Engineering

---

---

\* Corresponding author.

*Email addresses:* [barbara.weber@uibk.ac.at](mailto:barbara.weber@uibk.ac.at) (Barbara Weber), [bela.mutschler@hs-weingarten.de](mailto:bela.mutschler@hs-weingarten.de) (Bela Mutschler), [manfred.reichert@uni-ulm.de](mailto:manfred.reichert@uni-ulm.de) (Manfred Reichert).

## 1. Introduction

Providing effective IT support for business processes has become crucial for enterprises to stay competitive in their market [1,2]. In response to this need, a variety of process support paradigms (e.g., workflow management, case handling, service orchestration), process specification standards (e.g., WS-BPEL, BPMN), and business process management (BPM) tools (e.g., Tibco Staffware, FLOWer, IBM Websphere Process Server) have emerged supporting the realization of *Process-Aware Information Systems* (PAISs) [3]. Specifically, PAISs enable enterprises to implement and execute complex coordination scenarios either within an enterprise or in a cross-organizational setting [4].

Coordination scenarios are typically described by *coordination models*. Such models integrate the interactions of a number of (heterogeneous) components (processes, objects, agents) into a meaningful description. Relevant research areas are, for example, service-oriented architectures (i.e., service coordination, service orchestration, and service choreography), cooperative information systems (e.g., workflow management technology or case handling technology), component-based systems, multi-agent technology, and related middleware platforms.

When evaluating the suitability of existing BPM technology for a particular coordination scenario or when arguing about its strengths and weaknesses, typically, it becomes necessary to rely on qualitative criteria. As one example consider *workflow patterns* [5], which can be used to evaluate the expressiveness of the workflow modeling language provided by a particular BPM tool. As another example consider process change patterns [7], which facilitate the evaluation of BPM tools regarding their ability to deal with process changes. What has been neglected so far are more profound evaluations of BPM technology based on empirical or experimental research. This is surprising as the benefits of these research methods have been demonstrated in the software engineering area for a long time [8] (e.g., in the context of software development processes or code reviews [9,10]). In addition, a recently conducted survey among IT managers and project leaders has clearly shown that quantitative data on costs, benefits and effects of BPM technology becomes increasingly important [11].

Picking up this demand, this paper illustrates how experimental research can be applied in the BPM context. For this purpose we have conducted a controlled software experiment with 48 participants to investigate the effort related to the implementation and change of business processes either using conventional workflow technology [12] or a case handling system [13]. More precisely, we have used Tibco *Staffware* [14] as representative of workflow technology and *FLOWer* [15] as representative of case handling systems. We describe our experimental design, give a mathematical model of the experiment, and discuss potential threats for the validity of experimental results. Following this we describe the major results of our experiment, which contribute to better understand the complex effort caused by using BPM technology, and discuss them in detail.

This paper is a significant extension of the work we presented in [16]. It includes extended analyses of the data we gathered during our experiment and a more in-depth interpretation of the presented results. In particular, learning effects, which have to be considered when investigating the effort related to the implementation of business processes, constitute an additional aspect being addressed. Moreover, the comparison of workflow technology and case handling has been extended.

The remainder of this paper is organized as follows. Section 2 motivates the need for experimentation in BPM and provides background information needed for understanding our experiment. Section 3 describes our experimental framework. Section 4 deals with the performance and results of our experiment. Finally, Section 5 discusses related work and Section 6 concludes with a summary.

## 2. Background

This section presents the background needed for the understanding of this paper. Section 2.1 introduces Process-Aware Information Systems (PAISs). Section 2.2 deals with different paradigms for realizing PAISs, which we compare in our experiment.

### 2.1. Need for Process-Aware Information Systems

Empirical studies have indicated that providing effective business process support by information systems is a difficult task to accomplish [11,17]. In particular, these studies show that current information systems fail to provide business process support as needed in practice. Among the major reasons for this drawback are the hard-wiring of process logic in contemporary information systems and the missing support for coping with evolving business processes. Enterprises crave for approaches that enable them to control, monitor and continuously improve business process performance [18]. What is needed are PAISs, i.e., information systems that support the modeling, enactment and monitoring of business processes in an integrated and efficient way.

In general, PAISs *orchestrate* processes of a particular type (e.g., handling of a customer order) based on a predefined *process model*. Such a model defines the *tasks* to be executed (i.e., activities), their *dependencies* (e.g., control and data flow), the *organizational entities* performing these tasks (i.e., *process users*), and the *business objects* which provide or store activity data. Unlike conventional information systems, PAISs strictly separate process logic from application code [19]; i.e., PAISs are driven by process models rather than program code (cf. Fig. 3). Consequently, PAISs are realized based on process engines which orchestrate processes and their activities during run-time [20]. Typically, a process engine also provides generic functionality for the modeling and monitoring of processes, e.g., for accomplishing process analysis. Earlier empirical work confirms that PAISs enable a fast and cost-effective implementation as well as customization of business processes [21].

Realizing PAISs also implies a significant shift in the field of information systems engineering. Traditional engineering methods and paradigms (e.g., object-oriented design and programming) have to be supplemented with engineering principles and software technologies particularly enhancing the operational support of business processes (e.g., workflow management, case handling, and service orchestration). This is crucial to tie up those requirements neglected by current information systems so far.

### 2.2. Paradigms for Orchestrating Business Processes and their Activities

Assume that a business process for refunding traveling expenses - in the following denoted as *eTravel business process* - is to be supported by a PAIS which is realized using BPM

technology. The *eTravel business process* is used throughout the paper and is part of the material used for our experiment. It distinguishes between four organizational roles (cf. Fig. 1). The *traveler* initiates the refunding of his expenses. For this purpose, he has to summarize the travel data in a *travel expense report*. This report is then forwarded either to a *travel expense responsible* (in case of a national business trip) or to a *verification center* (in case of an international business trip).

Both the *travel expense responsible* and the *verification center* fulfill the same task, i.e., they verify a received travel expense report. “Verification” means that the declared travel data is checked for correctness and plausibility (e.g., regarding accordance with receipts). An incorrect travel expense report is sent back to the traveler (for correction). If it is correct, it will be forwarded to the *travel supervisor* for final approval. The supervisor role may be filled, for example, by the line manager of the traveler. If a travel expense report is approved by the supervisor, the refunding will be initiated. Otherwise, it will be sent back to either the travel expense responsible (national trip) or the verification center (international trip). Note that this is a characteristic (yet simplified) process as it can be found in many organizations.

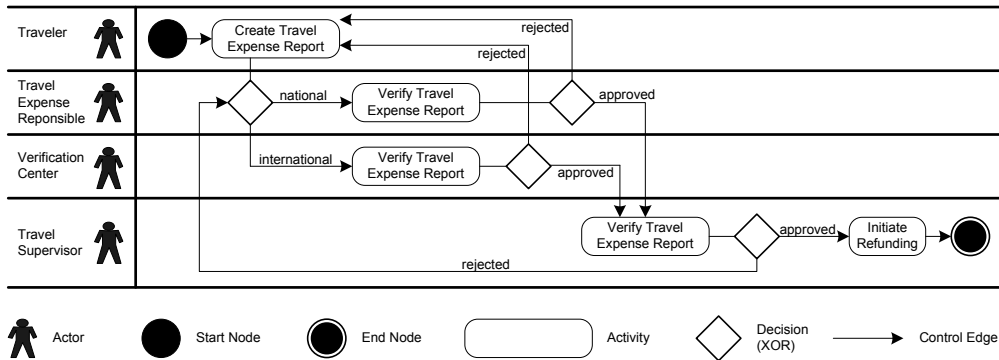


Fig. 1. The eTravel Business Process (modeled as UML Activity Diagram).

When realizing a PAIS which supports this process, one challenge is to select the most suitable BPM technology for this purpose. Currently, there exist various BPM approaches, which can be categorized as shown in Fig. 2. Basically, one distinguishes between *groupware systems*, *workflow management systems*, and *case handling systems*. Groupware systems aim at the support of unstructured processes including a high degree of personal communication. As groupware systems are not suitable for realizing PAISs, they are not further discussed in this paper. Workflow management systems (WfMSs), in turn, are best suited to support business processes which are well structured and have a high degree of repetition (e.g., procurement processes or clearance processes). Often, they are combined with an additional solution to integrate business processes within and across enterprises. Case handling systems (CHSs), in turn, are more flexible than traditional WfMSs [22], but are not suited for integrating heterogeneous application systems. In addition, for fully automated business processes (i.e., processes without the need for human interaction), CHSs are not the proper choice. Both workflow management and case handling are well suited for realizing administrative processes like our eTravel business process.

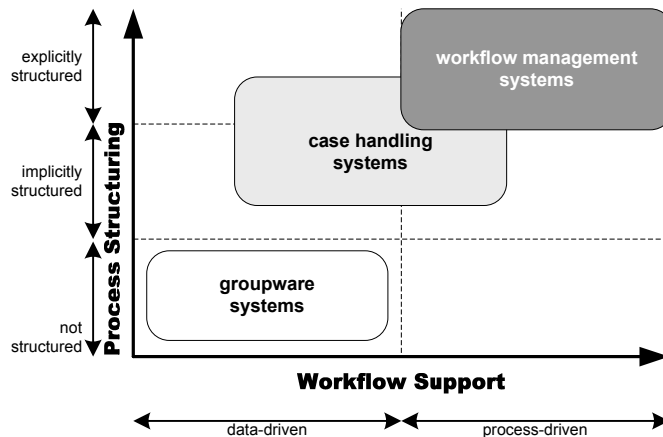


Fig. 2. Process Management Paradigms.

In the following workflow management as well as case handling are briefly introduced (for a detailed qualitative comparison of both paradigms we refer to [22]).

**Workflow Management.** Contemporary *workflow management systems* (WfMSs) enable the modeling, execution, and monitoring of business processes [4]. When working on a particular activity (i.e., process step), typically, in a WfMS-based PAIS only data needed for executing this activity is visible to respective actors, but no other workflow data. This is also known as “context tunneling” [13]. WfMSs coordinate activity execution based on routing rules, which are described by process models and which are strictly separated from processed data (cf. Fig. 3). If an activity is completed, subsequent activities will become active according to the logic defined by the used process model. Accompanying this, the worklists of potential actors are updated accordingly. Finally, for (administrative) processes electronic forms are typically used to implement activities and to present data being processed.

**Case Handling.** An alternative BPM paradigm is provided by case handling [13]. A *case handling system* (CHS) aims at more flexible process execution by avoiding restrictions known from (conventional) workflow technology (cf. Fig. 4). Examples of such restrictions include rigid control flow and the aforementioned context tunneling. The central concepts behind a CHS are the *case* and its data as opposed to the activities and routing rules being characteristic for WfMSs. One should think of a “case” as being the product which is “manufactured” by executing the workflow process. The characteristics of the product should drive the workflow. Typically, the product is information, e.g., a decision based on various data. By focusing on the product characteristics, one can replace push-oriented routing from one worktray to another by pull-oriented mechanisms centered around the data objects relevant for a case.

Usually, CHSs present all data about a case at any time to the user (assuming proper authorization), i.e., context tunneling as known from WfMSs is avoided. Furthermore, CHSs orchestrate the execution of activities based on the data assigned to a case. Thereby, different kinds of data objects are distinguished (cf. Fig. 4). *Free data objects* are not explicitly associated with a particular activity and can be changed at any point in time during a case execution (e.g., Data Object 3 in Fig. 4). *Mandatory* and *restricted data ob-*

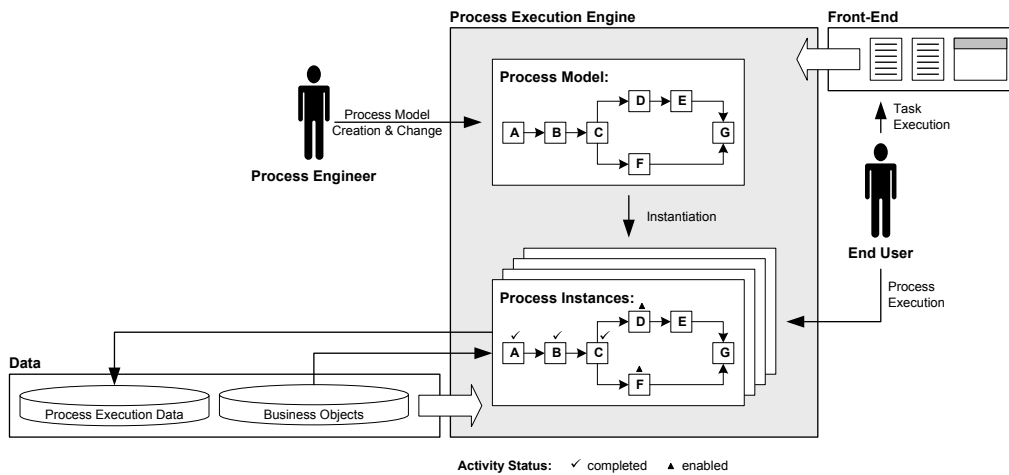


Fig. 3. Architecture of a Workflow Management System.

*jects*, in turn, are explicitly linked to one or more activities. If a data object is mandatory for an activity, a value will have to be assigned to it before the activity can be completed (e.g., Data Object 5 in Fig. 4). If a data object is restricted for an activity, this activity needs to be active in order to assign a value to the data object (e.g., Data Object 6 in Fig. 4). As in WfMSs, forms linked to activities are used to provide context-specific views on case data. Thereby, a CHS does not only allow assigning an *execution role* to activities, but also a *redo role* (to undo an executed activity) and a *skip role* (to omit the execution of activities). User 2 in Fig. 4, for example, may execute Activities 3, 4, 5 and 6, and redo Activities 2 and 3.

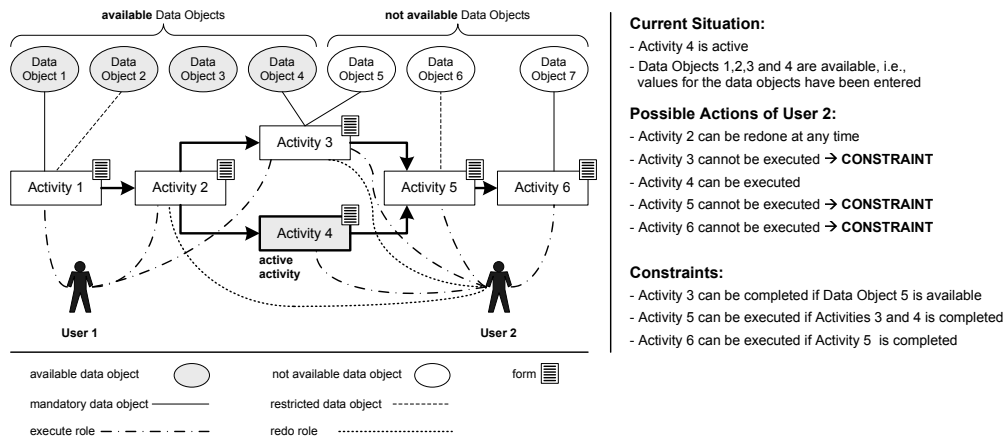


Fig. 4. Data-driven Case Handling.

Fig. 5 summarizes the major conceptual differences between workflow management and case handling, and additionally depicts characteristic representatives of each paradigm.

Despite conceptual differences, both paradigms are suited for implementing administrative business processes as our *eTravel business process*.

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

Fig. 5. Selected Criteria for Comparing Workflow Management and Case Handling.

### 3. Experimental Definition and Planning

This section deals with the definition and planning of our experiment. Section 3.1 explains its context and Section 3.2 describes its setup. Section 3.3 presents considered hypotheses. Section 3.4 explains the specific design of our experiment. Factors threatening the validity of experimental results as well as potential mitigations are discussed in Section 3.5. For setting up and describing our experiment we follow the recommendations given in [23,24]. We strongly believe that the design of our experiment can be applied in similar form to many other BPM related scenarios.

#### 3.1. Context Selection

With workflow management and case handling we have introduced two paradigms for realizing PAISs in Section 2.2. Usually, the selection of “the most suitable” BPM technology for implementing a PAIS depends on project-specific requirements. While some IT managers will consider BPM technology as sufficient if best practices are available, others will take into account more specific selection criteria like the support of a sufficient degree of process flexibility. Likewise, IT managers are interested in value-based considerations as well [17]. In practice, for example, a frequently asked question is as follows:

*Is there a difference in the effort needed for implementing a business process either with BPM technology A or BPM technology B and - if “yes” - how strong is this difference?*

Currently, IT managers typically have to rely on vendor data (e.g., about the return-on-investment of their products), experience reports, and criteria for qualitative comparisons as provided by workflow patterns [5] or process change patterns [7]. What has been not available so far are precise quantitative data regarding the use of workflow management technology and case handling systems respectively (e.g., concerning the effort for implementing processes) [16,25,17]. To generate quantitative data, and thus to complement existing qualitative criteria, controlled software experiments offer promising perspectives. In the following, we pick up this idea and describe an experiment in which we investigate the effort related to the implementation and adaptation of business processes using either a WfMS or a CHS.

The main goal of our experiment is to compare the implementation effort of WfMSs and CHSs. Using the *Goal Quality Metric* (GQM) template for goal definition [26], the goal of our experiment is defined as follows:

<b>Compare</b>	workflow management and case handling technology
<b>for the purpose of</b>	evaluating
<b>with respect to their</b>	implementation effort
<b>from the point of view of</b>	the researchers
<b>in the context of</b>	Bachelor and Master of Computer Science students at the University of Innsbruck

Fig. 6. Goal of our Experiment

### 3.2. Experimental Setup

This section describes the subjects, objects and selected variables of our experiment, and presents the instrumentation and data collection procedure.

**Subjects:** Subjects are 48 students of a combined Bachelor/Master Computer Science course at the University of Innsbruck. All subjects have a similar level of experience. They are taught about workflow management and case-handling in an introductory session preceding the execution of our experiment.

**Object:** The object to be implemented is the *eTravel business process* (cf. Section 2). Its specification comprises two parts: an initial “Base Implementation” (*Part I*) and an additional “Change Implementation” (*Part II*). While the first part deals with the realization of the process support for refunding national business trips, the second one specifies a process change, namely, additional support for refunding international business trips. Both parts describe the elements to be implemented; i.e., the process logic, user roles, and the data to be presented to actors using simple electronic forms. Note that this experimental design does not only enable us to investigate the effort for (initially) implementing a business process, but also to examine the effort for subsequent process changes. In our experiment, with “process change” we mean the adaptation of the implemented business process. After having realized such a process change new process instances can be based on the new process model. We do not investigate the migration of running process instances to the new process schema in this context [27].

**Factor & Factor Levels:** In our experiment, *BPM technology* is the considered factor with factor levels “WfMS” and “CHS”. Thereby, we use Tibco *Staffware* [14] (Version 10.1) as typical and widely applied representative of workflow technology. Its build-time tools include, among other components, a visual process modeling tool and a graphical form editor. The used CHS, in turn, is *FLOWer* [15] (Version 3.1), the most widely used commercial CHS. Like Staffware, FLOWer provides a visual process modeling tool and a form editor.

**Response Variable:** In our experiment the response variable is the *implementation*



effort the subjects (i.e., the students) need for implementing the given object (i.e., the *eTravel* specification) with each of the factor levels (WfMS and CHS). All effort values related to the Staffware implementation are denoted as “WfMS Sample”, while all effort values related to the FLOWer implementation are called “CHS Sample”.

**Instrumentation:** To precisely measure the response variable, we have developed a tool called *TimeCatcher* (cf. Fig. 7). This “stop watch” allows logging time in six typical “effort categories” related to the development of a process-oriented application: (1) *process modeling*, (2) *data modeling*, (3) *form design*, (4) *user/role management*, (5) *testing*, and (6) *miscellaneous effort*. To collect qualitative feedback as well (e.g., concerning the maturity or usability of the applied WfMS and CHS), we use a structured *questionnaire*.

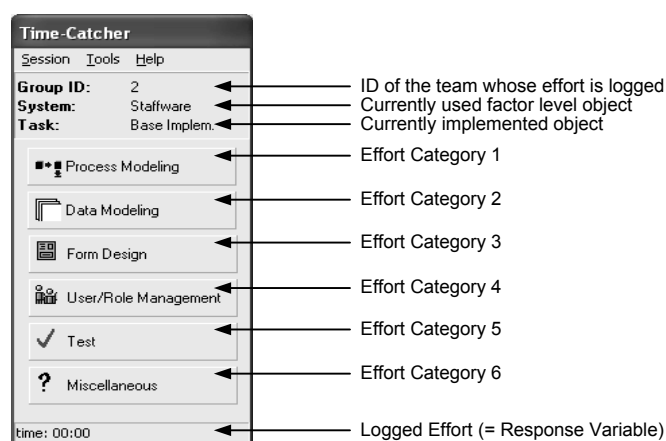


Fig. 7. TimeCatcher Tool.

**Data Collection Procedure:** The TimeCatcher tool is used by the students during the experiment. The aforementioned questionnaire is filled out by them after completing the experiment.

**Data Analysis Procedure:** For data analysis well-established statistical methods and standard metrics are applied (cf. Section 4.3 for details).

### 3.3. Hypothesis Formulation

Based on the goal of our experiment the following hypotheses are derived:

**Differences in Implementation Effort:** In our experiment we investigate whether the used *BPM technology* has an influence on the response variable *implementation effort*.

*Does the used BPM technology have an influence on the response variable “implementation effort”?*

**Null hypothesis  $H_{0,1}$ :** There is no significant difference in the effort values when using workflow technology compared to case handling technology.

**Alternative hypothesis  $H_{1,1}$ :** There is a significant difference in the effort values when using workflow technology compared to case handling technology.

**Learning Effects:** Our experiment further investigates learning effects that might occur when implementing the same business process twice with two different BPM technologies. In particular, we aim at determining the influence of domain knowledge on implementation effort. When implementing the *eTravel business process* with the first BPM technology the process specification is unknown to all subjects. When implementing the respective process with the second BPM technology, however, its specification is already known.

*Does knowledge of the process specification have an influence on the response variable “implementation effort”?*

**Null hypothesis  $H_{0,2}$ :** Domain knowledge does not have a statistically significant impact on the mean effort values for implementing a business process.

**Alternative hypothesis  $H_{1,2}$ :** Domain knowledge has a statistically significant impact on the mean effort values for implementing a business process.

### 3.4. Experimental Design

Literature about software experiments provides various design guidelines for setting up an experiment [28,23,21,8,29]. *First*, the design of an experiment should allow the collection of as much data as possible with respect to the major goals of the experiment. *Second*, collected data should be unambiguous. *Third*, the experiment must be feasible within the given setting (e.g., within the planned time period). Note that meeting these design criteria is not trivial. Often, an experiment cannot be accomplished as planned due to its complex design or due to an insufficient number of participants [8].

Considering these design criteria, we accomplish our experiment as a *balanced single factor experiment with repeated measurement* (cf. Fig. 8). This design is particularly suitable for comparing software development technologies [21]. Our experiment is denoted a single factor experiment since it investigates the effects of one *factor*<sup>1</sup> (i.e., a particular BPM technology) on a common *response variable* (e.g., implementation effort). Our ex-

<sup>1</sup> *Multi-factor experiments*, by contrast, investigate the effects of factor combinations on a common response variable, e.g., effects of a software development technology and a software development process on implementation effort. Even though such experiments can improve the validity of experimental results, they are rarely applied in practice due to their complexity [30].

periment design also allows us to analyze variations of a factor called *factor levels* (i.e., the two BPM tools Staffware and FLOWer). The response variable is determined when the participants of the experiment (i.e. *subjects*) apply the factor or factor levels to an *object* (i.e., the base and the change specification of the *eTravel business process*).

We denote our experiment as *balanced* as all factor levels are used by all participants of the experiment. This enables *repeated measurements* and thus the collection of more precise data since every subject generates data for every treated factor level. Generally, repeated measurements can be realized in different ways. We use a frequently applied variant which is based on two subsequent *runs* (cf. Fig. 8). During the *first run* half of the subjects apply “Staffware” to the treated object, while the other half uses “FLOWer”. After having completed the first run, the *second run* begins. During this second run each subject applies that factor level to the object not treated so far.

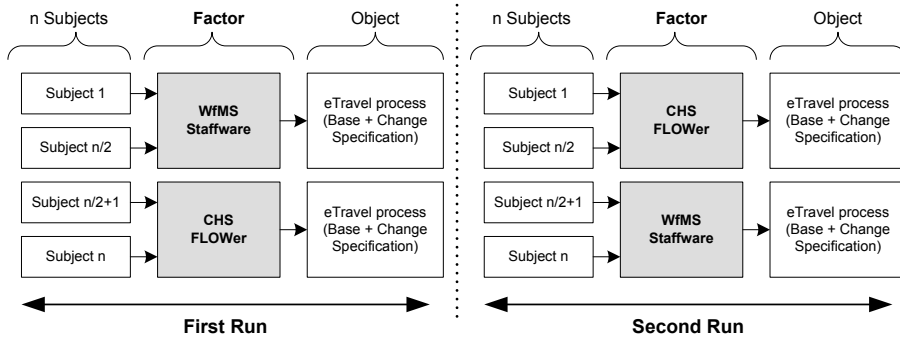


Fig. 8. Design of our Single Factor Experiment.

In our experiment subjects are not working on their own, but are divided into 4 *main groups* each consisting of 4 *teams* with 3 students (cf. Fig. 9). This results in an overall number of 16 teams. The students are randomly assigned to teams prior to the start of the experiment.

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

WfMS = Workflow Management System, CHS = Case Handling System

Fig. 9. Main Groups and Teams.

The mathematical model of our experiment can be summarized as follows:  $n$  subjects  $S_1, \dots, S_n$  ( $n \in \mathbb{N}$ ) divided into  $m$  teams  $T_1, \dots, T_m$  ( $m \in \mathbb{N}, m \geq 2, m$  even) have to implement the *eTravel business process*. The respective specification describes a “Base Implementation”  $O_1$  (corresponding to the “national case” of the *eTravel business process*) and a “Change Implementation”  $O_2$  (additionally introducing the “international case”). During the experiment one half of the teams ( $T_1, \dots, T_{m/2}$ ) implements the complete specification (i.e., base and change implementation) using a WfMS ( $PMS_1$ , Staffware), while the other half ( $T_{m/2+1}, \dots, T_m$ ) accomplishes this implementation using a CHS ( $PMS_2$ ,

FLOWer). After finishing the implementation with the first factor level (i.e., the *first run*), each team has to implement the eTravel process using the second factor level in a *second run* (i.e., the development technologies are switched). The response variable “Effort[Time] of  $T_m$  implementing  $O_i$  using  $PMS_j$ ” is logged with the TimeCatcher tool.

### 3.5. Risk Analysis and Mitigations

When accomplishing experimental research related risks have to be taken into account as well. Generally, there exist factors that threaten both the *internal validity* (“Are the claims we made about our measurements correct?”) and the *external validity* (“Can we justify the claims we made?”) of an experiment.

In our context, threats to internal validity are as follows:

- **People:** The students participating in our experiment differ in their skills and productivity for two reasons: (i) general experience with software development might differ and (ii) experience with BPM technology might not be the same. The first issue can only be balanced by conducting the experiment with a sufficiently large and representative set of students. The number of 48 students promises to achieve such balance. The second issue can be mitigated by using BPM tools unknown to every student. Only three of the participating students had rudimentary workflow knowledge beforehand. As this knowledge might influence experimental results, we have assigned those three students to different teams to minimize potential effects as far as possible. All other students have been randomly assigned to groups.
- **Data collection process:** Data collection is one of the most critical threats. Therefore we have to continuously control data collection during the experiment through close supervision of the students. We further have to ensure that students understand which TimeCatcher categories have to be selected during the experiment.
- **Time for optimizing an implementation:** The specification to be implemented does not include any guideline concerning the number of electronic forms or their layout. This implies the danger that some teams spend more time for implementing a “nice” user interface than others do. To minimize such effects, we explicitly indicate to the students that the development of a “nice” user interface is not a goal of our experiment. To ensure that the implemented solutions are similar across different teams, we accomplish acceptance tests.

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

- **Students instead of professionals:** Involving students instead of IT professionals constitutes a potential threat to the external validity of our experiment. However, the experiment by [32] evaluating the differences of students and IT professionals suggests that results of student experiments are (under certain conditions) transferable and can provide valuable insights into an analyzed problem domain. Furthermore, Runeson [31] identifies a similar improvement trend when comparing freshman, graduate and professional developers. Also note that the use of professional developers is hardly possible in

practice as profit-oriented organizations will not simultaneously implement a business process twice using two different BPM technologies. In addition, using professionals instead of students would also be not entirely free of bias. In particular, it would be very difficult to find professionals which are equally experienced with both systems under investigation.

- **Investigation of tools instead of concepts:** In our experiment, BPM tools are used as representatives for the analyzed concepts (i.e., workflow management and case handling). Investigating the concepts therefore always depends on the quality of the used tools. To mitigate this risk, the used BPM technologies should be representative for state-of-the-art technologies in practice (which is the case as both selected BPM tools are widely used representatives of workflow technology and case handling systems respectively).
- **Choice of object:** To mitigate the risk that the chosen business process is favouring one of the two BPM paradigms (i.e., case handling or workflow management), we have picked a business process that can be found in many organizations; i.e., the *eTravel business process* (cf. Section 2). However, additional experiments are needed to assess how far our results can be generalized to different types of business processes. Furthermore, one may argue that the use of UML activity diagrams can threaten the validity of the experiment as these diagrams are similar to the more explicit, flow-driven notation of Staffware process models, but different from the more implicit, data-driven FLOWer process models. However, in practice, UML activity diagrams (or other activity-centered diagramming techniques like Event-Driven Process Chains or BPMN) are widely used to describe standard business processes [33]. Thus, the use of UML activity diagrams can even improve internal validity as a typical practical scenario is investigated.

## 4. Performing the Experiment

This section deals with the preparation and execution of the experiment (cf. Section 4.1). It further covers the analysis and interpretation of the experimental data (cf. Section 4.2). Finally, it includes a discussion of experimental results (cf. Section 4.3).

### 4.1. Experimental Operation

**Preparation of the Experiment:** In the run-up of the experiment, we prepared a technical specification of the *eTravel business process*. This specification comprised UML activity diagrams, an entity relationship diagram describing the generic data structure of a travel expense report, and tool-specific data models for the considered systems (Staffware, FLOWer). To ensure that the specification is self-explanatory and correct we involved two student assistants in its development.

Before the experiment took place, the same two students implemented the specification with each of the utilized BPM technologies. This allowed us to ensure feasibility of the general setup of our experiment and to identify critical issues with respect to the performance of the experiment. This pre-test also provided us with feedback that helped to further improve comprehensibility of our specification. Finally, we compiled a “starter kit” for each participating team. It included original tool documentation, additional doc-

umentation created by us when preparing the experiment (and which can be considered as a compressed summary from the original documentation), and the technical process specification.

**Experimental Execution:** Due to infrastructure limitations, the experiment was split up in two events. While the first one took place in October 2006, the second one was conducted in January 2007. Each event lasted 5 days, involved 24 participants (i.e., students), and was based on the following procedure: Prior to the start of the experiment, all students had to attend an introductory lecture. We introduced to them basic notions of workflow management and case handling. During this lecture we further inform them about the goals and rules of the experiment. Afterwards, each team received its “starter kit”. Then, the students had to implement the given *eTravel business process* specification with both considered factor levels. Thereby, an implementation will be only considered as completed, if the students successfully pass the acceptance test. This ensured that all developed solutions correspond to the specification and were implemented correctly. After finishing their work on the experiment, students filled out the aforementioned questionnaire.

We further optimized the results of our experiment by applying *Action Research* [35]. Action Research is characterized by an intensive communication between researchers and subjects. At an early stage we optimized the data collection process by assisting and guiding the students in using the TimeCatcher tool properly (which is critical with respect to the quality of the gathered data). In addition, we documented emotional reactions of the students regarding their way of working. This helped us to design the questionnaire. Note that Action Research did not imply any advice for the students on how to implement the *eTravel business process*.

**Data Validation:** After conducting the experiment the data gathered by the teams using the TimeCatcher tool was checked. We discarded the data of two teams as it was flawed. Both teams had made mistakes using the TimeCatcher tool. Finally, the data provided by 14 teams was considered in our data analysis.

## 4.2. Data Analysis and Interpretation

We now deal with the analysis of gathered data and the interpretation of results.

### 4.2.1. Raw Data and Descriptive Analysis

Fig. 10 presents the raw data obtained from our experiment. For both test runs it shows the effort values for the base and the change implementation as well as the overall implementation effort<sup>1</sup> (measured in seconds). In the raw data table the values for the different effort categories (i.e., process modeling, data modeling, form design, user/role management, test, and miscellaneous) are accumulated.

---

<sup>1</sup> The overall implementation effort is calculated as the sum of the base implementation effort and the change implementation effort.

Experiment Data								
Group	First Test Run				Second Test Run			
	System	Base	Change	Overall	System	Base	Change	Overall
1	Staffware	15.303	816	16.119	Flower	23.261	1.521	24.782
2	Staffware	20.297	679	20.976	Flower	21.195	2.367	23.562
3	Flower	33.013	2.497	35.510	Staffware	18.420	1.630	20.050
4	Flower	29.024	2.608	31.632	Staffware	13.627	586	14.213
6	Staffware	13.995	1.609	15.604	Flower	19.487	1.942	21.429
7	Flower	38.814	2.461	41.275	Staffware	11.551	841	12.392
8	Flower	28.026	3.695	31.721	Staffware	12.650	1.126	13.776
10	Flower	35.616	1.215	36.831	Staffware	15.151	1.745	16.896
11	Staffware	13.747	1.617	15.364	Flower	18.407	1.243	19.650
12	Flower	32.049	2.072	34.121	Staffware	10.491	1.116	11.607
13	Staffware	22.440	3.301	25.741	Flower	18.981	2.525	21.506
14	Staffware	25.132	2.060	27.192	Flower	13.132	2.550	15.682
15	Flower	24.765	3.015	27.780	Staffware	6.677	740	7.417
16	Staffware	23.841	2.133	25.974	Flower	14.830	1.226	16.056

Fig. 10. Raw Data Obtained from the Experiment.

Based on this raw data we calculate descriptive statistics for our response variable *implementation effort* (cf. Fig. 11). When analyzing Fig. 11 one can observe the following:

- The mean effort values for FLOWER are higher than those for Staffware. This observation holds for the overall implementation effort as well as for the base and change implementations. Obviously, this means that implementation effort are smaller for the WfMS Staffware when compared to the CHS FLOWER.
- The mean effort values for the base implementation are much higher than those for the change implementation for both Staffware and FLOWER.
- The effort values for implementing the *eTravel business process* using FLOWER in the first test run are higher than those for using FLOWER in the second test run.
- The effort values for implementing the *eTravel business process* using Staffware in the first test run are higher than those for using Staffware in the second test run.

#### 4.2.2. Data Plausibility

We analyze data plausibility based on *box-whisker-plot diagrams*. Such diagrams visualize the distribution of a sample and particularly show outliers. A low number of outliers indicates plausible data distributions of the base implementation effort in our experiment. The diagram takes the form of a box that spans the distance between the 25% quantile and the 75% quantile (the so called *interquartile range*) surrounding the median which splits the box into two parts. The “whiskers” are straight lines extending from the ends of the box. As such, the length of a whisker is at most 1.5 times the interquartile range. All results outside the whiskers can be considered as outliers. As can be seen in Fig. 12A, there are no outliers regarding the *base* implementation effort for the first test run; i.e., all data from these samples lie within the boxed area. However, there exist two outliers (i.e., o4 and o5) in the distribution of the *change* implementation effort for the first test run (cf. Fig. 12B), and no outliers regarding the distribution of the *overall* implementation effort for the first test run (cf. Fig. 12C). For the second test run no outliers can be identified at all (cf. Fig. 12D-F).

Descriptive Statistics								
System	Object	Test Run		N	Minimum	Maximum	Mean	Standard Deviation
Flower	Base	1st	Time [s]	7	24.765	38.814	31.615,29	4.769,575
		2nd	Time [s]	7	13.132	23.261	18.470,43	3.498,155
		<b>Total</b>	<b>Time [s]</b>	<b>14</b>	<b>13132</b>	<b>38814</b>	<b>25042,86</b>	<b>7916,249</b>
	Change	1st	Time [s]	7	1.215	3.695	2.509,00	768,146
		2nd	Time [s]	7	1.226	2.550	1.910,57	586,197
		<b>Total</b>	<b>Time [s]</b>	<b>14</b>	<b>1.215</b>	<b>3.695</b>	<b>2.209,79</b>	<b>726,184</b>
	Overall	1st	Time [s]	7	27.780	41.275	34.124,29	4.332,368
		2nd	Time [s]	7	15.682	24.782	20.381,00	3.492,184
		<b>Total</b>	<b>Time [s]</b>	<b>14</b>	<b>15.682</b>	<b>41.275</b>	<b>27.253</b>	<b>8.071</b>
Staff	Base	1st	Time [s]	7	13.747	25.132	19.250,71	4.837,774
		2nd	Time [s]	7	6.677	18.420	12.652,43	3.697,932
		<b>Total</b>	<b>Time [s]</b>	<b>14</b>	<b>6.677</b>	<b>25.132</b>	<b>15.951,57</b>	<b>5.369,811</b>
	Change	1st	Time [s]	7	679	3.301	1.745,00	885,549
		2nd	Time [s]	7	586	1.745	1.112,00	439,266
		<b>Total</b>	<b>Time [s]</b>	<b>14</b>	<b>586</b>	<b>3.301</b>	<b>1.428,50</b>	<b>747,577</b>
	Overall	1st	Time [s]	7	15.364	27.192	20.995,71	5.327,046
		2nd	Time [s]	7	7.417	20.050	13.764,43	4.007,169
		<b>Total</b>	<b>Time [s]</b>	<b>14</b>	<b>7.417</b>	<b>27.192</b>	<b>17.380,07</b>	<b>5.881,059</b>

Fig. 11. Descriptive Statistics for Response Variable.

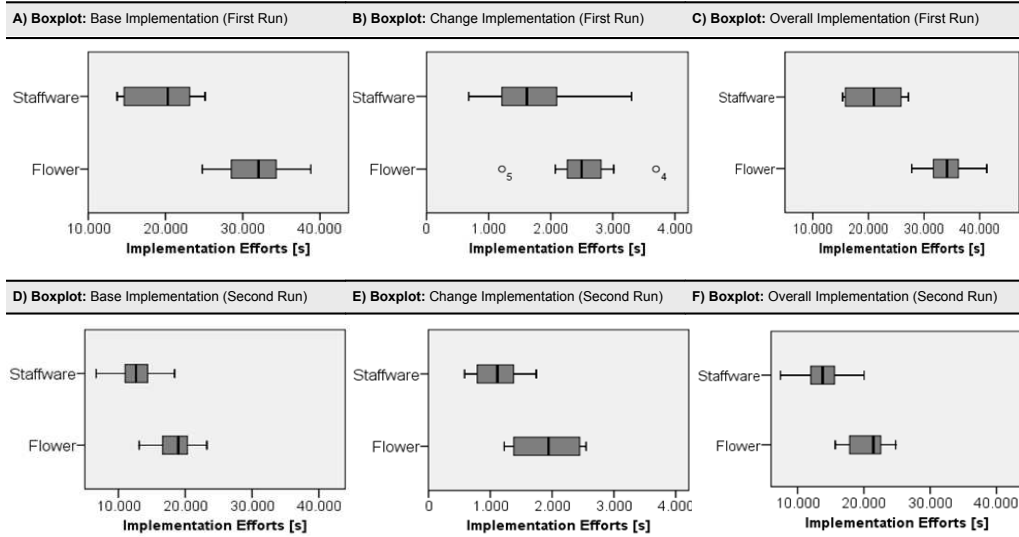


Fig. 12. Data Distribution (Box-Whisker-Plot Diagrams).

#### 4.2.3. Testing for Differences in Implementation Effort

In this section we describe the data analysis for our Hypothesis  $H_{0,1}$ . We analyze this 0-hypothesis based on a *two-sided t-test* [30] (and an additional *sign test* if the t-test fails). Doing so, we are able to assess whether the means of the WfMS sample and the CHS sample are statistically different from each other. A successful t-test (with  $|T| > t_0$  where  $T$  is the observed t-statistic and  $t_0$  is a predefined value depending on the size of sample  $x$  and significance level  $\alpha$ ) rejects our 0-hypothesis. Specifically, the following steps (1) - (4) have to be executed in order to accomplish a t-test (with  $\alpha = 0.05$  as the level of significance):



- (1) **Paired Comparison:** The t-test is combined with a *paired comparison* [30], i.e., we analyze “pairs of effort values”. Each pair comprises one effort value from the WfMS sample and one from the CHS sample. Note that we compose pairs according to the performance of the teams, i.e., effort values of “good” teams are not combined with effort values of “bad” teams (cf. [21]). *Paired Comparison 1* in Fig. 13, for example, combines effort values from Main Groups 1 and 3 with effort values from Main Groups 2 and 4 (precise pairs are shown in Fig. 14).

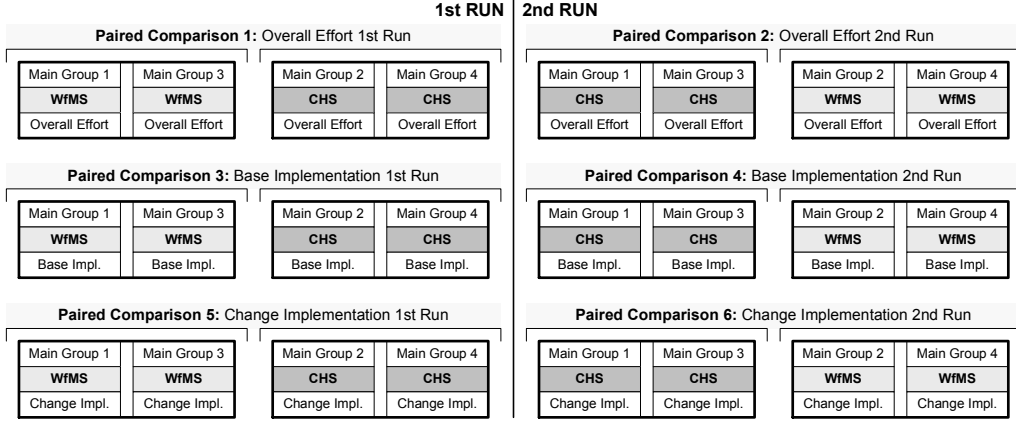


Fig. 13. Paired Comparison.

- (2) **Standardized Comparison Variable:** For each pair, a *standardized comparison variable*  $X_j$  is derived. It is calculated by dividing the difference of the two compared effort values by the first effort value:

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

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

- (3) **Statistical Measures:** For the standardized comparison sample  $x$  we calculate its *median* ( $m$ ), *interquartile range* (IQR), *expected value* ( $\mu$ ), *standard deviation* ( $\sigma$ ), and *skewness* ( $sk$ ).
- (4) **Two-sided t-Test:** Finally, we apply the t-test to  $x$ . Note that the t-test will be only applicable if  $x$  is taken from a *normal distribution* and the WfMS and CHS sample have *same variance*. The first condition can be tested using the *Kolmogorov/Smirnov test* [34]. In particular, the result of this test has to be smaller than  $K_0$  (with  $K_0$  being a predefined value depending on the size of  $x$  and the chosen significance level  $\alpha$ ). The second condition can be tested using the *test for identical variance* [34]. The variance of the WfMS and CHS sample will be identi-

cal, if the result of this test is smaller than  $F_0$  (with  $F_0$  being a predefined value depending on the size of the samples and the chosen significance level  $\alpha$ ).

Fig. 14A shows the results of our analysis regarding *overall implementation effort*. When studying the effort for the workflow implementation, we can see that they are lower than the ones for the case handling implementation. This difference is confirmed by the results of the (successful) t-tests for both the first and the second run, i.e., our 0-hypothesis  $H_{0,1}$  is to be rejected. In the first run, the use of workflow technology results in effort savings of 43.04% (fluctuating between 27.51% and 50.81%) when compared to the effort for using CHS-based technology. In the second run, the use of workflow technology still results in savings of 28.29% (fluctuating between 11.48% and 53.16%).

Fig. 14B and Fig. 14C show results for the *base implementation* as well as the *change implementation*. Again, our results allow the rejection of the 0-hypothesis  $H_{0,1}$  (the failed t-test can be compensated with a successful *sign test*). Using workflow technology results in effort savings of 43.01% for the base implementation in the first run (fluctuating between 32.03% and 50.06%). In the second run, the use of workflow technology results in effort savings of 28.52% when compared to case handling effort. Regarding the change implementation the use of workflow technology results in effort savings of 44.11% in the first run (fluctuating between 16.29% and 56.45%) and 40.46% in the second run.

Fig. 15 illustrates that the obtained effort of the WfMS sample are smaller for all six effort categories when compared to the CHS sample.

The additional analysis of our questionnaire provides possible explanations for these differences. Fig. 16A shows that the concepts underlying workflow technology seem to be easier to understand, i.e., the case handling paradigm is considered as being more difficult. Finally, Fig. 16B deals with the usability of the applied process management systems. The subjective results obtained from the questionnaire show that the students perceive Staffware as being more user-friendly when compared to FLOWer.

Based on the results of the questionnaire we assume that the observed differences in implementation effort between case handling and workflow technology are (1) due to conceptual differences (i.e., the case handling concept was perceived as being more complicated) and (2) due to differences in the tools (i.e., the used case handling system FLOWer is perceived as being less user-friendly). Further experiments with different designs are needed to confirm these assumptions.

#### 4.2.4. Testing for Learning Effects

To investigate learning effects we compare the effort values of the groups using Staffware in the first run with those groups using Staffware in the second test run. In addition, we repeat this procedure for FLOWer.

As all preconditions for the t-test are met (samples are normally distributed and have same variance), we test 0-hypothesis  $H_{0,2}$  regarding the learning effects using the t-test (with  $\alpha = 0.05$  as the level of significance). The t-test reveals that there is a statistically significant difference between effort values for the first test run and for the second one.

Fig. 17A and Fig. 17B show the results for FLOWer and Staffware in respect to overall implementation effort. When comparing the effort values for the two test runs, it can be observed that for both systems the effort for the first run are generally higher than those for the second run. Regarding FLOWer in the second run even the slowest group is

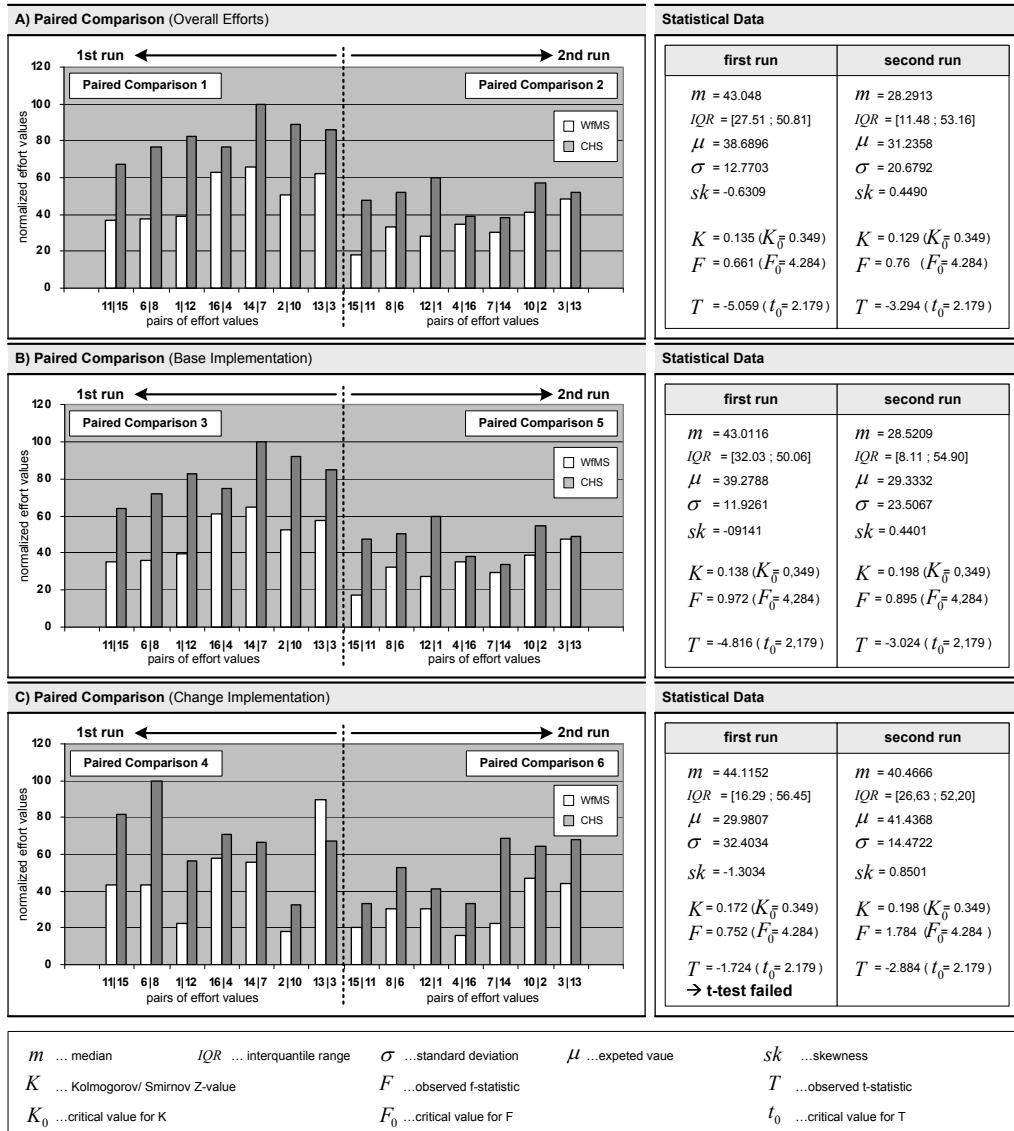


Fig. 14. Experimental Results

faster than the fastest one in the first run. These differences are confirmed by the results of the t-test for both FLOWer and Staffware, i.e., our 0-hypothesis  $H_{0,2}$  is rejected at a significance level of 0.05. For FLOWer the mean difference between first and second test run is 13743 seconds (fluctuating between 9161 and 18326) and for Staffware it is 7231 seconds (fluctuating between 1742 and 12721).

Fig. 17C and 17D show the results for FLOWer and Staffware in respect to base implementation effort. Like for the overall implementation effort the differences in implementation effort between the first and the second test run are statistically significant, which leads to a rejection of 0-hypothesis  $H_{0,2}$ . For FLOWer the mean difference between

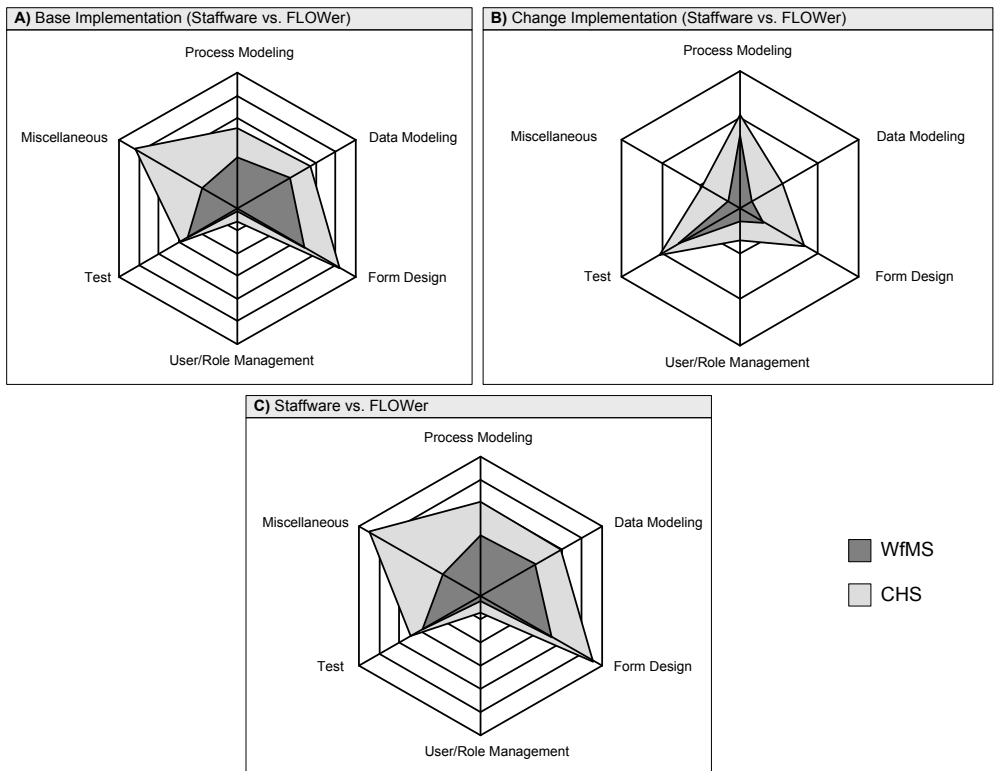


Fig. 15. Understanding Effort Distribution.

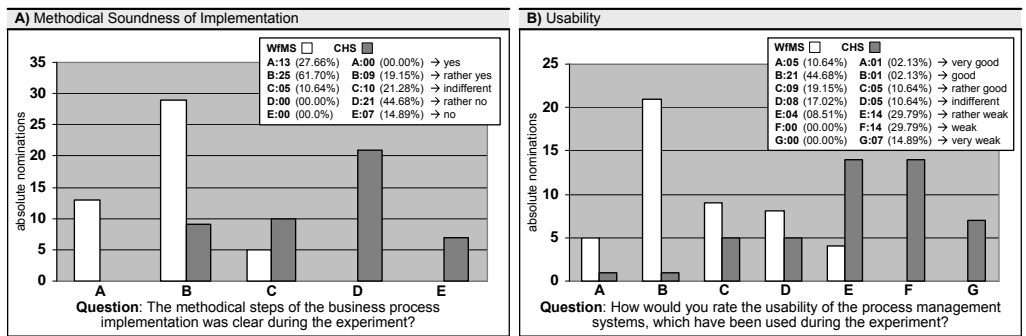


Fig. 16. Selected Questionnaire Results (Part 1).

first and second test run is 13145 seconds and for Staffware it is 6598 seconds.

Fig. 17E and 17F show the results in respect to change implementation effort. Only the effort savings for FLOWer between the first and second test run with a mean difference of 598 seconds are statistically significant. For Staffware, the t-test fails, but can be compensated by a successful Mann Whitney U-test.

The observed differences in implementation effort between the two test runs can be explained either through an increasing process knowledge gathered during the experiment

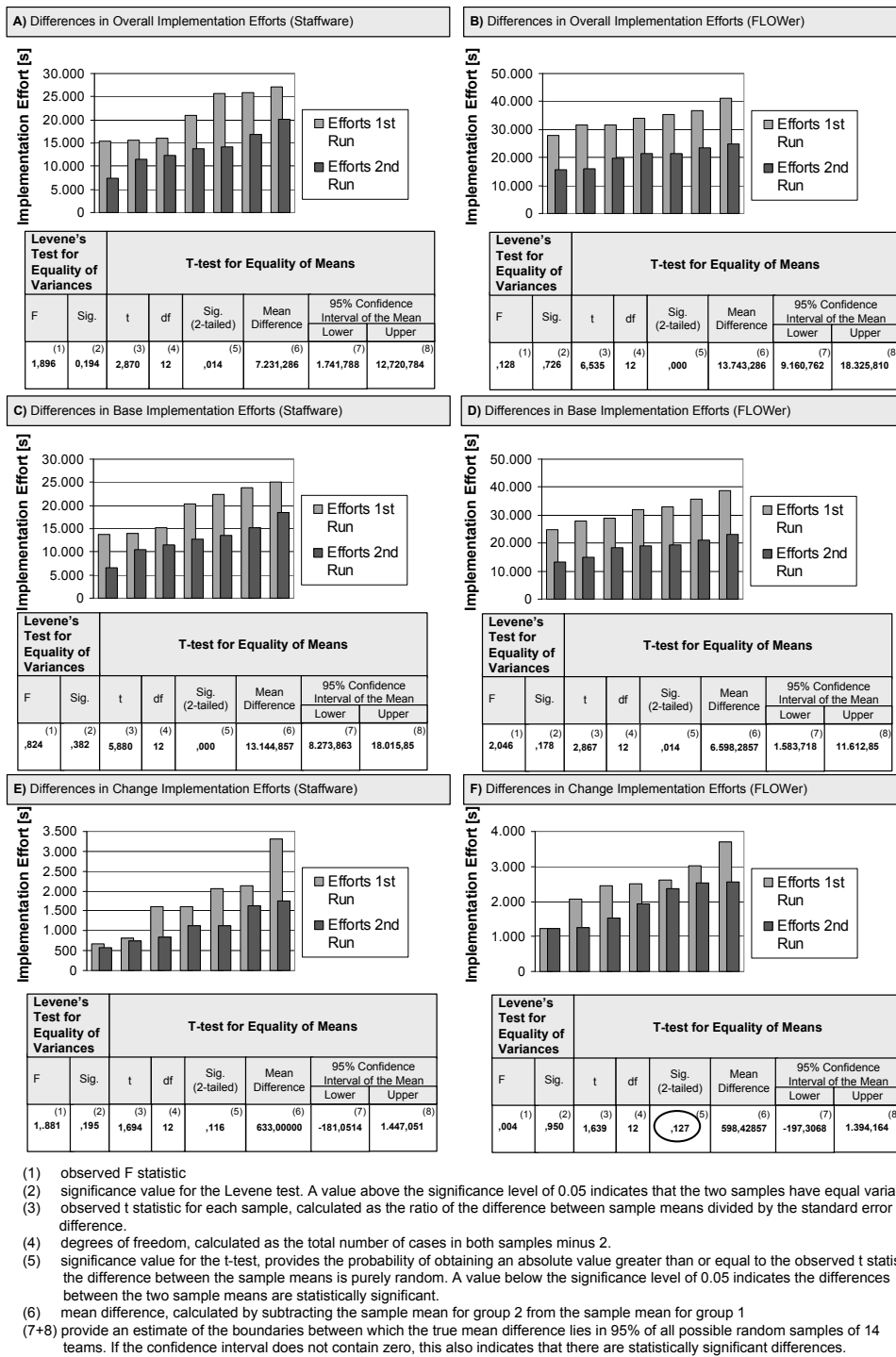


Fig. 17. Implementation Effort for 1st and 2nd Test Run

or an increasing tool knowledge, which is partially transferable when working with other PAISs.

The results of the questionnaire provide possible explanations for the observed learning effects. Fig. 18A illustrates that according to questionnaire results process knowledge gained during the first run significantly simplifies the second run. By contrast, Fig. 18B shows that increased efficiency during the second run can be related only to a much smaller degree to a gained tool knowledge. Consequently, we assume that the observed differences are primarily related to increased process knowledge, and not necessarily to learning effects concerning the used BPM technologies (i.e., tool knowledge).

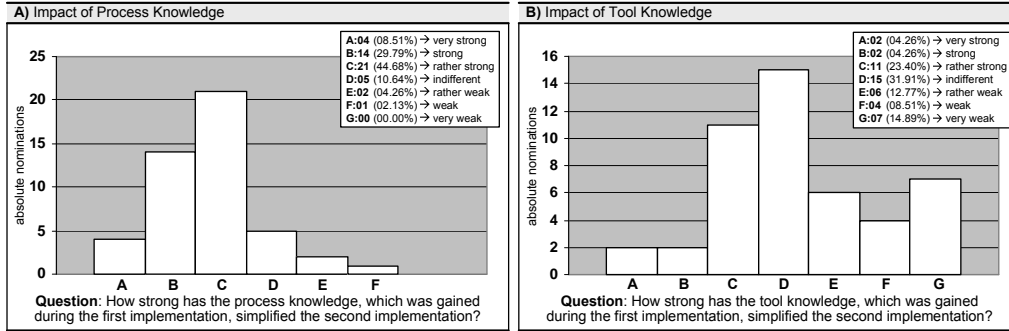


Fig. 18. Selected Questionnaire Results (Part 2).

#### 4.2.5. Additional Observations

Fig. 19 shows that the initial implementation of the *eTravel business process* (i.e., its base implementation) takes significantly more time than subsequent changes. Partially these differences are due to the smaller scope of the change implementation (cf. Fig. 1). While the base implementation requires the implementation of four activities, two decision-points and four forms, the change implementation comprises one additional activity and two additional decision-points. Furthermore, two existing forms need to be slightly adapted.

Besides differences in the scope of base and change implementation it can be assumed that tool-specific learning effects are contributing to these differences. In our experiment design the change implementation immediately follows the base implementation. Consequently, increasing tool knowledge gathered while working on the base implementation might influence change implementation effort.

To investigate whether these differences apply with the same degree to both FLOWer and Staffware we calculate a comparison variable *changeRatio*, which is defined as the ratio of change and base implementation. Fig. 20 shows descriptive statistics for variable *changeRatio*. It can be observed that for FLOWer the mean change ratio in the first test run is 8.3% (fluctuating between 3.41% and 13.18%). For the second test run it is 10.77% (fluctuating between 6.54% and 19.42%). Similarly, for Staffware the mean change ratio for the first test run is 9.11% (fluctuating between 3.35% and 14.71%) and 8.94% for the second run (fluctuating between 4.3% and 11.52%). However, when comparing the effort savings between base and change implementation, there are no statistically significant

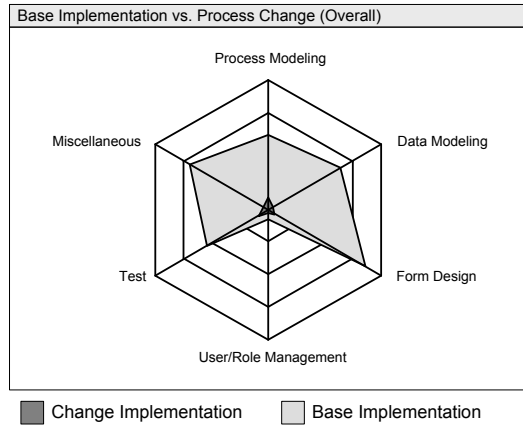


Fig. 19. Base Implementation vs. Change Implementation

differences between Staffware and FLOWer. Interestingly, change ratios are very similar though Staffware and FLOWer are representing different process support paradigms.

Base Implementation versus Change Implementation						
System	Test Run	N	Minimum	Maximum	Mean	Standard Deviation
Flower	1st	7	3,41%	13,18%	8,30%	3,44%
	2nd	7	6,54%	19,42%	10,77%	4,52%
Staffware	1st	7	3,35%	14,71%	9,11%	3,93%
	2nd	7	4,30%	11,52%	8,94%	2,53%

Fig. 20. Ratio Between Change Implementation and Base Implementation

#### 4.3. Discussion

Our results indicate that process implementations based on workflow technology generate lower effort when compared to implementations based on case handling technology (cf. Fig. 14). As illustrated in Fig. 15 these effort savings apply to all six effort categories (i.e., process modeling, user/role management, form design, data modeling, test, and miscellaneous). Moreover, our results show that initial implementations of processes come with a significantly higher effort when compared to subsequent process changes (cf. Fig. 19). Interestingly, the ratio between the effort for subsequent process changes and initial process implementations seems to be tool-independent (cf. Fig. 20). This is particularly important for policy makers, who often focus on short-term costs (e.g., for purchasing BPM technology and initially implementing business processes) rather than on long-term benefits (e.g., lower costs for realizing process changes).

Finally, our data indicates that growing knowledge about the processes to be implemented results in increased productivity of software developers (cf. Fig. 17). Regardless of which BPM technology is used first, all teams reduce their effort in the second run significantly. Questionnaire results further indicate that this effect is not necessarily related

to an increasing knowledge about the used BPM technology, but is fostered by increasing process knowledge. This also emphasizes the need to involve domain experts with high process knowledge when applying BPM technology.

Considering our experimental design, it is inevitable to acknowledge that experiment results are influenced by the quality of the used BPM tools. However, by selecting leading commercial BPM tools as representatives for the analyzed concepts (i.e., workflow management and case handling), we can reduce the impact that tool quality has on the results of our experiment. Yet, based on this single experiment, it cannot be generalized that the effort related to workflow management is generally smaller when compared to case handling. For this purpose, additional experiments with different experimental designs and more specific research questions are needed, e.g., experiments comparing conventional WfMSs, adaptive WfMSs<sup>2</sup>, and CHSs regarding their effectiveness when realizing particular process changes.

We have applied the described experimental results in the EcoPOST project [36]. This project aims to develop an approach to investigate complex causal dependencies and related cost effects in PAIS engineering projects. In particular, our results enable us to quantify causal dependencies in PAIS engineering projects. As an example consider the impact of process knowledge on the productivity of process implementation [36,25]. In addition, our experimental results enable us to specify the effort distribution of the six analyzed effort categories (cf. Fig. 21).

## 5. Related Work

There exist a number of approaches dealing with the evaluation of (economic) effects related to PAIS. So far, focus has been on analyzing the impact of WfMSs on business process performance. The most similar experimental design, when compared to ours, is provided by [21]. This work investigates the impact of workflow technology on software development and software maintenance. Results indicate that the effort for realizing process-oriented information systems can be significantly reduced when using workflow technology (instead of conventional programming techniques). Oba et al. [39], in turn, investigate the introduction of WfMSs to an organization and particularly focus on the identification of factors that influence work efficiency, processing time, and business process standardization. A mathematical model is provided for predicting the rate of reduction of processing times. An extension is the work of Reijers and van der Aalst [40,41]. They use process simulation to compare pre- and post-implementations of information systems that rely on WfMSs. Focus is on analyzing business process performance based on criteria such as lead time, waiting time, service time, and utilization of resources. In most cases, the use of workflow technology has resulted in a significant decrease of lead and service time.

Choenni et al. [42] present a model to measure the added value of WfMSs to business processes. This model builds upon different *performance criteria*, i.e., parameters of a business process that are affected by the introduction of a WfMS (such as speed, quality, flexibility, and reliability).

---

<sup>2</sup> Adaptive WfMSs extend traditional WfMSs with the ability to flexibly deal with process changes during run-time (e.g., to dynamically add, delete or move activities in the flow of control) [37,38]



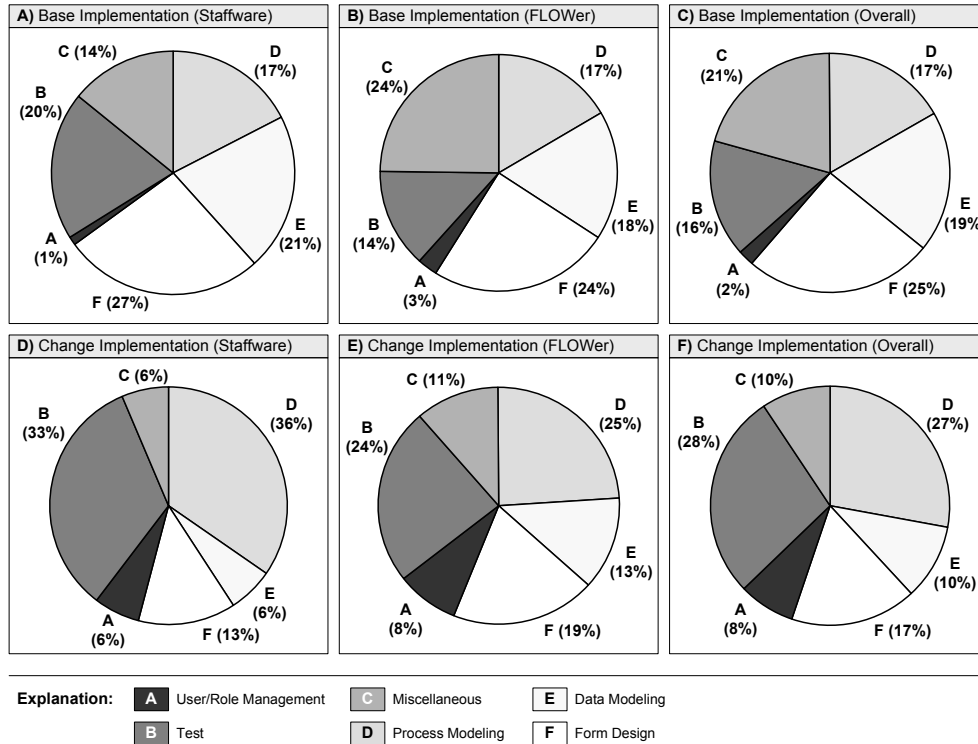


Fig. 21. Distribution of Logged Effort.

Aiello [43] introduces a measurement framework for evaluating workflow performance. The framework is defined in an abstract setting to enable generality and to ensure independence from existing WfMSs.

Becker et. al [44] introduce a framework to identify those processes that can be supported by a WfMS in a “profitable” way. Their framework can serve as a guideline for evaluating processes during the selection and introduction of a WfMS. It contains three groups of criteria: technical, organizational, and economic criteria. Designed as a scoring model, their approach enables users to systematically determine those business processes that can be automated using a WfMS.

A different approach is proposed by Abate et al. [45]. This work introduces a measurement language to evaluate the performance of automated business processes: the “workflow performance query language” (WPQL). This language allows the definition of metrics independently of a specific workflow implementation.

While the approaches described above investigate (economic) effects related to PAISs from a quantitative perspective, existing work on workflow patterns provides qualitative evaluation criteria. Patterns have been introduced for analyzing the expressiveness of process meta models [5,46,47]. In this context, control flow patterns describe different constructs to specify activities and their ordering. In addition, workflow data patterns [48] provide ways for modeling the data aspect in PAISs, and workflow resource patterns [49] describe how resources can be represented in workflows. Moreover, a set of change

patterns and change support features has been proposed by Weber et al.[7,50] to compare PAISs regarding their ability to deal with process change. Furthermore, patterns for describing service interactions and process choreographies [51] as well as exception handling patterns have been proposed [52]. The introduction of workflow patterns has had significant impact on PAIS design as well as on the evaluation of PAISs and process languages like BPEL [46], BPMN [53], EPC [46], and UML [54]. Furthermore, the patterns-based evaluations of both Staffware and FLOWer seem in particular noteworthy in the context of this paper [5,7].

To evaluate the suitability of a BPM technology for a particular scenario patterns are important, but not sufficient. In addition to qualitative criteria quantitative data is needed to support IT decision makers in the selection of suitable technologies. With this paper we want to stimulate more experimental research in the BPM field to achieve this.

## 6. Summary and Outlook

This paper presents the results of a controlled BPM software experiment with 48 students. Our results indicate that business process implementations based on traditional workflow technology generate a lower effort than using case handling technology. Thereby, initial process implementations result in a higher effort than subsequent process changes (independently of whether workflow technology or case handling technology is used). In addition, our results show that the impact of domain knowledge on implementation effort is not negligible. It is important to mention that our results are complementing existing research on workflow patterns [5,7]. While patterns facilitate the selection of appropriate BPM technologies by providing qualitative selection criteria, our experiment contributes to satisfying the increasing demand of enterprises for quantitative data regarding the effort related to BPM technologies [17].

Future work will include additional experiments to investigate the role of domain knowledge and tool knowledge in more detail and to confirm our observations that the observed learning effects are not tool-specific. In addition, we plan more specific experiments to investigate the effort related to process modeling and process change. In particular, we aim at conducting similar experiments to assess whether and - if yes - how far our results can be transferred to different business processes and different types of subjects. In addition, we plan to investigate whether the usage of change patterns [7] leads to a lower effort for process modeling and process change. Moreover, we want to analyze the impact of using business process refactorings [55] on process maintenance effort.

## References

- [1] Y. L. Antonucci, Using Workflow Technologies to improve Organizational Competitiveness, *Int'l. J. of Management*, 14(1), pp. 117-126, 1997.
- [2] R. Lenz and M. Reichert, IT Support for Healthcare Processes - Premises, Challenges, Perspectives, *Data and Knowledge Engineering*, 61(1), pp. 39-58, 2007.
- [3] M. Dumas and W.M.P. van der Aalst and A.H.M. ter Hofstede, *Process-aware Information Systems*, Wiley, 2005.
- [4] M. Weske, *Business Process Management: Concepts, Methods, Technology*, Springer, 2007.

- [5] W.M.P. van der Aalst and A.H.M. ter Hofstede and B. Kiepuszewski and A.P. Barros, Workflow Patterns, Distributed and Parallel Databases, 14(3), pp. 5-51, 2003.
- [6] B. Weber and S. Rinderle-Ma and M. Reichert, Change Patterns and Change Support Features - Enhancing Flexibility in Process-aware Information Systems, Data and Knowledge Engineering 66(3), pp. 438-466, 2008.
- [7] B. Weber and S. Rinderle and M. Reichert: Change Patterns and Change Support Features in Process-Aware Information Systems. In: Proc. CAiSE'07 (2007), LNCS 4495, pp. 574-588.
- [8] D. I. K. Sjöberg and J. E. Hannay and O. Hansen and V. B. Kampenes and A. Karahasanovic and N.-K. Liborg and A. C. Rekdal, A Survey of Controlled Experiments in Software Engineering, IEEE Transactions in Software Engineering, 31(9), pp. 733-753, 2005.
- [9] G. J. Myers, A controlled Experiment in Program Testing and Code Walkthroughs/Inspections, Communications of the ACM, 21(9), pp. 760-768, 1978.
- [10] C. M. Lott and H. D. Rombach, Repeatable Software Engineering Experiments for Comparing Defect-Detection Techniques, Empirical Software Engineering, 1(3), pp. 241-277, 1996.
- [11] B. Mutschler and M. Reichert and J. Bumiller, Unleashing the Effectiveness of Process-oriented Information Systems: Problem Analysis, Critical Success Factors, Implications, IEEE Transactions on Systems, Man, and Cybernetics (Part C), 38(3), pp. 280-291, 2008.
- [12] W.M.P. van der Aalst and K. van Hee, Workflow Management, MIT Press, 2004.
- [13] W.M.P. van der Aalst and M. Weske and D. Grünbauer, Case Handling: A New Paradigm for Business Process Support, Data and Knowledge Engineering, 53(2), pp. 129-162, 2005.
- [14] Tibco, Staffware Process Suite, User Manual, 2005.
- [15] Pallas Athena, Case Handling with FLOWer: Beyond Workflow, 2002.
- [16] B. Mutschler and B. Weber and M. Reichert, Workflow Management versus Case Handling - Results from a Controlled Software Experiment, Proc. 23rd Annual ACM Symposium on Applied Computing (SAC '08), Special Track on Coordination Models, Languages, Architectures, pp. 82-89, 2008.
- [17] B. Mutschler, Modeling and Simulating Causal Dependencies on Process-aware Information Systems from a Cost Perspective, PhD Thesis, University of Twente, 2008.
- [18] J. Chang, Envisioning the Process-Centric Enterprise, EAI Journal, August 2002, pp. 30-33, 2002.
- [19] J. Dehnert and W.M.P. van der Aalst, Bridging the Gap between Business Models and Workflow Specification, Int'l. Journal of Cooperative Information Systems, 13(3), pp. 289-332, 2004.
- [20] M. Reichert and S. Rinderle and U. Kreher and P. Dadam, Adaptive Process Management with ADEPT2, Proc. ICDE '05, pp. 1113-1114, 2005.
- [21] N. Kleiner, Can Business Process Changes Be Cheaper Implemented with Workflow-Management-Systems?, Proc. IRMA 2004, pp. 529-532, 2004.
- [22] C. W. Günther, M. Reichert and W. M. P. van der Aalst, Supporting Flexible Processes with Adaptive Workflow and Case Handling, Proc. WETICE'08, 2008.
- [23] N. Juristo and A. M. Moreno, Basics of Software Engineering Experimentation, 2001.
- [24] C. Wohlin, R. Runeson, M. Halst, M.C. Ohlsson, B. Regnell and A. Wesslen, Experimentation in Software Engineering: an Introduction, Kluwer Academic Publisher, 2000.
- [25] B. Mutschler and M. Reichert, On Modeling and Analyzing Cost Factors in Information Systems Engineering, Proc. CAiSE 2008, LNCS 5074, pp. 510-524, 2008.
- [26] Basili, V.R., Rombach, H.D.: The TAME project: Towards Improvement-oriented Software Environments. IEEE Transactions on Software Engineering 14(6), pp. 758-773, 1998.
- [27] S. Rinderle and M. Reichert and P. Dadam, Flexible Support of Team Processes by Adaptive Workflow Systems., Distributed and Parallel Databases, 16(1), pp. 91-116, 2004.
- [28] V. R. Basili and R. W. Selby and D. H. Hutchens, Experimentation in Software Engineering, IEEE Transactions on Software Engineering, 12(7), pp.733-743, 1986.
- [29] M. V. Zelkowitz and D. R. Wallace, Experimental Models for Validating Technology, IEEE Computer, 31(5), pp. 23-31, 1998.
- [30] L. Prechelt, Controlled Experiments in Software Engineering (in German), Springer, 2001.
- [31] P. Runeson, Using Students as Experiment Subjects - An Analysis on Graduate and Freshmen Student Data, Proc. 7th Int'l. Conf. on Empirical Assessment & Evaluation in Software Engineering (EASE '03), pp.95-102, 2003.
- [32] M. Höst and B. Regnell and C. Wohlin, Using Students as Subjects A Comparative Study of Students and Professionals in Lead-Time Impact Assessment, Empirical Software Engineering, 5(3), 201-214, 2008.

- [33] A. W. Scheer, *Aris-Business Process Frameworks*, Springer, 1999.
- [34] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, 2000.
- [35] P.W. Reason and H. Bradbury, *Handbook of Action Research*, 2001.
- [36] B. Mutschler and M. Reichert and S. Rinderle, Analyzing the Dynamic Cost Factors of Process-aware Information Systems: A Model-based Approach, *Proc. CAiSE 2007*, LNCS 4495, pp. 589-603, 2007.
- [37] M. Reichert and P. Dadam, ADEPT<sub>flex</sub> – Supporting Dynamic Changes of Workflows Without Losing Control., *Journal of Intelligent Information Systems*, 10(2), pp. 93-129, 1998.
- [38] S. Rinderle and M. Reichert and P. Dadam, Correctness Criteria for Dynamic Changes in Workflow Systems – A Survey, *Data and Knowledge Engineering*, 50(1), pp. 9-24, 2004.
- [39] M. Oba and S. Onoda and N. Komoda, Evaluating the Quantitative Effects of Workflow Systems based on Real Cases, *Proc. HICSS 2000*.
- [40] H. A. Reijers, Performance Improvement by Workflow Management Systems: Preliminary Results from an Empirical Study, *Proc. ICEIS '04*, pp.359-366, 2004.
- [41] 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. Manag.*, 25(5), pp. 457-471, 2005.
- [42] S. Choenni and R. Bakker and W. Baets, On the Evaluation of Workflow Systems in Business Processes, *Electronic Journal of Information Systems Evaluation (EJISE)*, 6(2), 2003.
- [43] R. Aiello, *Workflow Performance Evaluation*, PhD Thesis, University of Salerno, Italy, 2004.
- [44] J. Becker and C. von Uthmann and M. zur Muehlen and M. Rosemann, Identifying the Workflow Potential of Business Processes, *Proc. HICSS '99*, 1999.
- [45] A. F. Abate and A. Esposito and N. Grieco and G. Nota, Workflow Performance Evaluation through WPQL, *Proc. SEKE '02*, pp.489-495, 2002.
- [46] N. Russell and A.H.M ter Hofstede and W.M.P van der Aalst and N. Mulyar, Workflow Control-Flow Patterns: A Revised View., *Technical Report BPM-06-22*, BPMcenter.org, 2006.
- [47] F. Puhlmann, M. Weske, Using the Pi-Calculus for Formalizing Workflow Patterns., *Proc. BPM'05*, LNCS 3649, pp. 153-168, 2005.
- [48] N. Russell and A.H.M ter Hofstede and D. Edmond and W.M.P van der Aalst, Workflow Data Patterns, *Technical Report FIT-TR-2004-01*, Queensland Univ. of Techn., 2004.
- [49] N. Russell and A.H.M ter Hofstede and D. Edmond and W.M.P van der Aalst, Workflow Resource Patterns, *Technical Report WP127*, Eindhoven Univ. of Technology, 2004.
- [50] S. Rinderle-Ma and M. Reichert and B. Weber: On the Formal Semantics of Change Patterns in Process-aware Information Systems. In *ER'08*, LNCS 5231, pp. 279-293.
- [51] A. Barros, M. Dumas, A. ter Hofstede, Service Interaction Patterns., *Proc. BPM'05*, LNCS 3649, pp. 302-318, 2005.
- [52] N. Russell and W.M.P van der Aalst and A.H.M ter Hofstede, Exception Handling Patterns in Process-Aware Information Systems, *Proc. CAiSE'06*, LNCS 4001, pp. 288-302, 2006.
- [53] P. Wohed, W.M.P van der Aalst, M. Dumas, A.H.M ter Hofstede, N. Russell, On the Suitability of BPMN for Business Process Modelling, *Proc. BPM'06*, pp. 161-176, 2006.
- [54] N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, P. Wohed, On the Suitability of UML 2.0 Activity Diagrams for BP Modelling, *Proc. APCCM '06*, pp. 95-104, 2006.
- [55] B. Weber and M. Reichert, Refactoring Process Models in Large Process Repositories, *Proc. CAiSE'08*, LNCS 5074, pp. 124-139, 2008.