

# Who Cares About Software Process Modelling? A First Investigation About the Perceived Value of Process Engineering and Process Consumption

Marco Kuhrmann<sup>1</sup>, Daniel Méndez Fernández<sup>1</sup>, and Alexander Knapp<sup>2</sup>

<sup>1</sup> Technische Universität München, Faculty of Informatics, 85748 Garching, Germany  
{kuhrmann|mendezfe}@in.tum.de

<sup>2</sup> Universität Augsburg, Institute of Informatics, 86159 Augsburg, Germany  
knapp@informatik.uni-augsburg.de

**Abstract.** When it comes to designing a software process, we have experienced two major strategies. Process engineers can either opt for the strategy in which they focus on designing a process using an artefact model as backbone or, on the other hand, they can design it around activities and methods. So far, we have first studies that directly analyse benefits and shortcomings of both approaches in direct comparison to each other, without addressing the questions relevant to process engineers and which implications the selection of a particular design strategy has on the process consumption. We contribute a first controlled investigation on the perceived value of both strategies from the perspectives of process engineers and process consumers. While our results underpin the artefact-oriented design strategy to be an advantageous instrument for process engineers, process consumers do not evidently care about the selected design strategy. Furthermore, our first investigation performed in an academic environment provides a suitable empirical basis, which we can use to steer further replications and investigations in practical environments.

**Keywords:** Software processes, Software process modelling, Action research

## 1 Introduction

Software process models are the glue that holds organisations, projects, and people together. They provide a blueprint of all relevant artefacts, activities, roles, and supporting entities necessary to implement a company-specific software process. The way of operating projects of different complexities, including its distribution, people, different cultures and application domains, is reflected in the complexity of today's software processes. This increases the demand to structure and implement software processes in a systematic manner, and to make the resulting process descriptions easily accessible and understandable to project participants. Consequently, one has to consider two basic perspectives on software processes: their design and their consumption.

To design a software process, process engineers face the choice between two major strategies: the *activity-oriented* strategy and the *artefact-oriented* one. The activity-oriented strategy concentrates on analysing the activities and methods applied in projects, and defining a software process on basis of the behaviour of project teams [17].

The basic idea of artefact orientation, in turn, consists of concentrating on what has to be done rather than on how something has to be done. A software process is designed on the basis of an artefact model that defines a blueprint of the results to be created and, thus, it abstracts from the activities for creating the results by using particular methods and modelling notations.

For the last decade, we are investigating the shortcomings and the benefits of the two introduced strategies in direct comparison to each other [11]. Our findings indicate to the benefits of artefact orientation in order to support flexibility in the process, precision in the results, a standardised terminology among different project participants, or a systematic quality assurance. However, the industrial context of those case studies was always characterised by a given process defined and probably lived for years. People had certain expectations and needs regarding the instantiation of a process to efficiently organise and run a project, and to give guidance on how to create precise result structures during operations.

The still unsolved question remains whether the initial choice of a certain strategy matters for the process design and the process consumption if no expectations are present and which implications the choice has.

**Problem statement.** Although we have deep knowledge on specific benefits and shortcomings of different software process models and the underlying paradigms when applying them at selected socio-economic contexts, we have still no knowledge which implications the choice of a particular strategy has for process engineers and process consumers independently of given expectations and experiences in same or similar contexts. The decision for one strategy or the other is, however, of critical importance as the maintenance and a re-design of a process model as part of a software process improvement initiative is a cost- and time-intensive endeavour.

**Research objective.** We aim at investigating which implications the choice for a particular software process design strategy has for process engineers and process consumers. Our expectation is that the selection of a strategy does not affect the consumers' perceived value. To lay the foundation for a systematic investigation of this expectation, we describe a first controlled experiment to be further replicated in different contexts. In this experiment proposal, we compare the design strategies from the perspective of a process engineer and furthermore we investigate the users understanding of the particular outcomes. In summary, we define the research objective as follows:

---

<i>We analyse</i>	the selection of software process design strategies
<i>for the purpose of</i>	evaluation
<i>with respect to</i>	the perceived values
<i>from the point of view of the</i>	process engineers and the process consumers
<i>in the context of</i>	the process life cycle

---

**Contribution.** We contribute a controlled student experiment in which we analyse the perceived value of a given software process w.r.t. the strategy followed when designing

and implementing it, as well as when consuming its deployed descriptions from the perspective of process consumers. The context of the study is the lecture “Software Engineering Processes”, given at the Technische Universität München.

Our experiment compares one process and two representative software process frameworks (incl. software process meta models, authoring tools, and deployment infrastructures). We use both frameworks to analyse, conceptualise, and implement the given process in a tool-supported manner. We finally evaluate the deployed process documentations according to a fixed set of criteria by simulating an appraisal. We opt for the Eclipse Process Framework [4] as a representative of the activity-oriented paradigm and for the V-Modell XT framework [6] as a representative of the artefact-oriented paradigm. Inherited from the nature of a controlled experiment, we have to avoid any side effects to distort the aforementioned procedure, such as organisational cultures, the integration of the constructed process into an organisational structure and further aspects of process maintenance.

**Outline.** The paper is organised as follows. Section 2 introduces the fundamentals necessary for the study environment. We furthermore discuss related work, which gaps are left open, and how our contribution intends to close the gaps. In Sect. 3, we introduce the study design, and we present the results in Sect. 4. In Sect. 5, finally, we give a summary of conclusions, discuss the relation to existing evidence, the impact and limitations of the study, and planned future work.

## 2 Fundamentals and Related Work

In the following, we briefly discuss the fundamentals in terms of engineering software processes, and related work in the context of our investigation.

*Software Process Design Strategies.* For the context of this paper, we distinguish between two design strategies: activity orientation and artefact orientation. The activity-oriented strategy relies on the idea of defining a concrete process by a set of methods to be performed in a particular order by a specific set of roles. Each of the methods provides a construction procedure to combine description techniques [13]. The activity-oriented strategy is especially favoured in the area of (Situational) Method Engineering. For instance, Ralyté and Rolland [15] discuss an assembly-based approach in which pre-fabricated method chunks are combined. Method chunks shall comprise activities as well as artefacts. Since method engineering is still in the phase of discussing and consolidating the basic concepts, first proposals are made to introduce systematic modelling approaches, e.g. Brinkkemper and Saeki [2]. However, method engineering in general pays little attention to artefacts, which is reflected by the summary of method engineering concepts by Henderson-Sellers and Ralyté [7]. Beyond method engineering, current software process metamodels lay the foundation to develop arbitrary software processes. Prominent examples are software processes, which are based on the Software & Systems Process Engineering Metamodel (SPEM; [14]), such as the Rational Unified Process (RUP; [8]) or Hermes [3]. Still, although the importance of a well-defined

artefact model is recognised in this area [5], artefacts are not in scope in available approaches. Braun et al. [1] mention that only 50% of the analysed approaches include an artefact description at all, while approaches that include an artefact description often reduce the artefacts to an optional outcome of the methods.

The activity-oriented strategy thereby does not provide assistance in the creation of precise result structures. In response to this situation, a new design strategy arose baptised as artefact orientation. The idea of the artefact-oriented strategy consists of defining a blueprint of all artefacts that are an (intermediate) result of a project. At project level, the actual process is then defined by agreeing on a set of artefacts to be created by particular roles and to be delivered when reaching particular milestones. The diversity in the process definitions is thereby reduced to the dependencies among the artefacts themselves without having to take into account the complexity of differing processes and detailed workflows. Artefact-oriented approaches are thereby meant to guide in the creation of precise results while offering the necessary flexibility during their creation by avoiding to dictate concrete methods. A detailed discussion on both strategies can be taken from [12].

*Comparative evaluations and studies.* There exist, to the best of our knowledge, barely studies that empirically evaluate both previously introduced strategies to design and implement a software process in direct comparison. We conducted such a study on the application of an artefact-based requirements engineering approach in direct comparison to an activity-based approach previously used in an industrial environment [11]. We could show, for example, an increase in the flexibility of the process and the syntactic quality of the created artefacts when relying on artefact orientation. However, as stated in Sect. 1, our focus lied on an industrial context with practitioners' expectations and experiences shaped for years in that context. So far, we could gain data that shows advantages of the artefact-oriented design strategy when narrowing down the context to a particular discipline, but at the cost of analysing the overall software process life cycle.

To investigate the perceived value of a particular strategy taking into account the overall life cycle of a process while controlling potential side-effects given by aforementioned expectations and experiences, a controlled experiment is yet necessary. The paper at hand reduces this gap by contributing the design for such a controlled experiment as well as first results.

### **3 Research Design**

We organise the experiment's design according to [16]. After defining the goal and the research questions, we describe the case and subject selection. Finally, we describe how we collect and analyse the data, before discussing the validity procedures.

#### **3.1 Research Questions**

As stated in Sect. 1, we investigate which implications the choice of a particular software process design strategy has for process engineers and for consumers. To this end, we formulate a working hypothesis in accordance with our expectation that the selection of a particular design strategy has no impact on the consumers' perceived value:

**WH:** *The selection of a design strategy for establishing an effective process management does not affect its actual consumption.*

Since we initiate experimentation on this hypothesis, we opt for a case-study-like action research approach to capture the domain of interest and to get initial data. To this end, we formulate two research questions that shall investigate the perceived value of the artefact-oriented and the activity-oriented strategy taking first the perspective of a process engineer into account and then taking the perspective of a process consumer.

**RQ 1:** *How suitable is the used design strategy to cover the needs of process engineers when analysing, designing, and implementing a process?*

The first research question considers the analysis of the strengths of one strategy to another when constructing a process. This construction covers the analysis of a process, its conceptual design, and its tool-supported implementation.

**RQ 2:** *To what extent does the used strategy matter for process consumers when interpreting the resulting process documentation?*

Once we constructed a process (and rated the strategies for their support during the construction), we want to know which strategy better supports the application of the generated process documentation from a consumer's perspective.

### 3.2 Case and Subject Selection

The choice of the software process frameworks is opportunistic. We need, however, to ensure that the following prerequisites are fulfilled:

1. The software process framework has to be based on a metamodel representing either the activity-based strategy or the artefact-based one.
2. The software process framework has to offer a tool infrastructure that allows for the implementation and export of a process.

As example process to be implemented with the framework, we opt for a process that reflects the complexity of a "real life" process and that offers enough information for the process engineers to realise the process.

Regarding the subjects involved in the study, we define two groups. The *process engineer* is responsible for realising the process and to rate the realisation. The *process consumer* is exclusively responsible for evaluating the created process documentation without any insights into the construction of the process.

### 3.3 Data Collection Procedure

Figure 1 depicts an overview of the data collection procedure. We set up the study as part of our lecture "Software Engineering Processes" [9] in the winter term 2011/2012. Two groups of students of roughly the same size were assigned to the particular strategy (artefact or activity orientation). Each student group then conducted three workshops with fixed time frames for analysing, for conceptualising, and for implementing

the process in a tool-supported manner. The last session of the lecture considered the evaluation of the results where the students changed their role to the one of a process consumer for evaluating also the process documentation created by the other team. In the following, we explain each step of the data collection in detail.

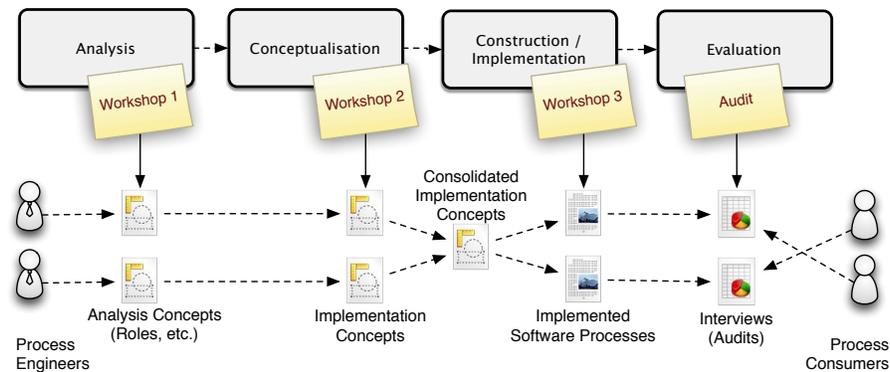


Fig. 1: Overview of data collection procedure.

**Analysis workshop.** The analysis comprehends two sessions in which a given scenario is analysed. The scenario is a real-life process model of a special interest group of the German Computer Society<sup>3</sup>. In the first session, two student groups analyse the particular sources for stakeholders to infer the basic roles. The roles are collected, structured, and prepared for discussion. In the second session, the students analyse either the process elements or the artefacts, depending on the group assignments.

**Conceptualisation workshop.** In the second workshop, the two student groups discuss the analysis results; one group gets the material for the artefacts as input, the other group gets the overall process, including milestone structures and an assignment to a set of activities, as input. The students then switch to the conceptualisation and create first platform-independent artefact and process models.

The outcome of the analysis and conceptualisation workshops are synthesised into an integrated concept. This step is done by the supervisors to get consistent conditions during the construction workshop supporting for comparable outcomes during the evaluation. To this end, the students are provided with a set of consolidated spread sheets and a requirements document for the realisation. Furthermore, the supervisors prepare the technical infrastructure in terms of creating (Subversion) repositories and bare process templates for the students to start with.

**Construction workshop.** In the construction workshop, the students are initially introduced into the consolidated concept. Two implementation groups are formed and in

<sup>3</sup> Available at: <http://www.vorgehensmodelle.de> (in German)

each group one student takes the lead. The first group uses the Eclipse Process Framework to implement the process and the second group uses the V-Modell XT framework, respectively. During the construction, the supervisors assist the students with discussion w.r.t. modelling techniques, in general, and concrete design decisions, in particular. Presentations of the groups' outcomes conclude the construction workshop.

Table 1: Questionnaire with closed questions for process engineers (RQ 1).

No.	Question (condensed)	Aspect
Q1-2	The used tool and the approach were intuitively applicable.	Tool
Q1-3	I got familiar with the tool and could implement all process elements.	Tool
Q1-5	The tool was suitable to implement the process with minimal overhead.	Tool
Q1-6	I was aware of every decision and its consequences.	Tool
Q1-7	The process documentation could be created appropriately.	Tool
Q1-8	I could implement all process elements.	Methodology
Q1-9	I could implement all designed roles.	Methodology
Q1-10	I could implement all designed artefacts.	Methodology
Q1-11	I could implement all designed relationships among roles and artefacts.	Methodology
Q1-12	I could implement all designed activities.	Methodology
Q1-13	I could implement the designed overall process.	Methodology
Q1-14	I could always create an export of the model (process documentation).	Tool
Q1-15	I could perform all required tasks for the implementation.	Tool
Q1-16	I was always able to identify and check the model consistency.	Tool

Table 2: Questionnaire with closed questions for process consumers (RQ 2). Considered aspect is the process documentation only.

No.	Question (condensed)
Q2-1	The information in the export is easy to access.
Q2-2	The information in the export is easy to access for non-familiar users.
Q2-3	The overall process is clearly presented and provides a good overview of all elements.
Q2-4	The export supports the analysis of consistency.
Q2-5	All expectations were met and all designed concepts were appropriately implemented.
Q2-6	All designed roles were implemented.
Q2-7	All designed artefacts were implemented.
Q2-8	All designed relationships between roles and artefacts were implemented.
Q2-9	All designed activities were implemented.
Q2-10	The designed overall process was adequately implemented.
Q2-11	The process model is consistent.

**Evaluation session.** The final step of the experiment considers the evaluation of the designed and implemented processes. The evaluation consists of two steps, both using a given questionnaire:

1. Self-assessment of the produced process documentation taking the perspective of a process engineer (Table 1)

2. Audit of the process documentation of the other group taking the perspective of a process consumer (Table 2)

In the self-assessment, the students rate their own work with a particular focus on the used tool and followed methodology. For the assessment and the audit, questionnaires with closed and open questions are made available (see Table 1 with the condensed, closed questions for the assessment and Table 2 for the audit). We are especially interested in the closed questions to rate and compare given strategies. Each of the closed questions considers the rating of one particular criterion against a given statement in a scale ranging from 0 (“I strongly disagree”) to 7 (“I strongly agree”).

### 3.4 Analysis Procedure

Due to the low number of samples (8 students), statistical (hypothesis) testing is not suitable. After collecting the data, we thus plot radar charts with the rankings given in the closed questions to qualitatively analyse the results and directly compare the results of the groups with each other.

## 4 Study Results

We first give a description of the case and the subjects and summarise, as a second step, the results w.r.t. the single research questions.

### 4.1 Case and Subject Description

*Case – Process frameworks.* Since one part of the study is to implement a selected process, we select two software process frameworks and the corresponding authoring tools. The first chosen framework is the *Eclipse Process Framework* (EPF; [4]), the second framework is the German V-Modell XT [6]. Both frameworks are based on a metamodel supporting process modelling and structuring, as well as providing support for publishing processes to make the process accessible for the process users, and are of a maturity [10] that allows for using these frameworks in an experimentation setting. EPF is based on SPEM [14], an OMG specification, which is basically an UML profile for process engineering. The V-Modell XT is based on the *V-Modell XT Metamodel* [18], a metamodel that was specifically developed to create artefact-oriented processes.

*Case – Process model.* The selected case is a process model given by the special interest group “Software Development Processes” of the German Computer Society. The process under consideration describes how the annual special interest group workshops are organised (including roles, activities, and milestones). Since the process was originally defined in the late 1990’s, it evolved and got into an inconsistent state. Summarised, the process consists of a 20-page word document, an (incomplete) project plan of about 200 (hierarchical) activities, a set of templates, and a couple of Excel spreadsheets that contain checklists and further detailed information for certain tasks.

**Subjects.** The subjects are 8 students of the lecture “Software Engineering Processes” (winter term 2011/2012). As shown in Fig. 1, for each workshop the students are divided into two groups at the beginning of the particular workshops. For the analysis and conceptualisation workshops, the groups are formed at random. At the beginning of the construction workshop, the students are allowed to select a process framework themselves, which, in consequence, also defines the groups for the self-assessments and the audits.

The goals of the selected grouping procedure are that (1) each student should have the same basic knowledge about the considered process model, and that (2) each student should act in different roles—in the role *Process Engineer* during the analysis, conceptualisation, construction and self-assessment, and in the role *Process Consumer* during the audit.

## 4.2 Workshops

Three workshops are performed to analyse, conceptualise, and construct the process models (see Fig. 1). In this section, we show the results of each particular workshop.

**Analysis workshop.** In the analysis sessions, the students analyse the given material to gather information about roles, artefacts, activities, and the basic process structure (i.e. artefact-role assignments). For those sessions, the students use creativity techniques for the elaboration (see Fig. 2) and are provided with pre-defined templates by the supervisors to structure the information. The outcomes of the analysis workshop are:

- An unordered set of 37 roles and stakeholders
- An unordered set of 32 artefacts
- A list of more than 50 activities (partially inferred from the artefacts)



Fig. 2: Photo of the analysis workshop (elaboration of artefacts).



Fig. 3: Photos of the conceptualisation workshop (artefact and process structures).

**Conceptualisation workshop.** The last analysis workshop session fades into the conceptualisation workshop in which the students need to aggregate all information. One group concentrates on artefacts and roles, structures and dependencies (Fig. 3, upper part), and the number of artefacts of a certain type being created. The second group

focusses on the overall process structure that needs to be inferred from the activity structure, and the milestones (see Fig. 3, lower part). Since both groups exchange information, the second group also includes core artefacts to prepare a first mapping between the process elements leading to an overall process and its core workflows.

Since the outcomes are complex and heterogeneous, the supervisors boil down the outcomes to an integrated realisation concept that is the input for the construction workshop. The realisation concept is a 20-slide Powerpoint presentation that contains:

- The overall process, including the phases and top-level work packages
- A breakdown structure for each of the top-level work packages, including the core artefacts, fine-grained activities/workflows, and possible execution orders of the activities/workflows
- Information w.r.t. the dependency structure (between activities and roles, artefacts and roles, and so on)
- A consolidated set of 33 roles and stakeholders
- Consolidated artefact and activity sets
- Refined requirements for the construction

**Construction workshop.** The construction workshop is a full-day event in which the students are at first introduced to the realisation concept. Afterwards, the two implementation groups are formed. To ensure that all groups have the same starting conditions, the supervisors prepare the coarse process template the students can start with. As outcomes, the students create:

- An EPF-based implementation of the process (5 students)
- A V-Modell-XT-based implementation of the process (3 students)

### 4.3 Evaluation (Audit)

The evaluation consists of two parts, according to our research questions. Figures 4 and 5 show the results of the self-assessment w.r.t. RQ 1 considering the two aspects *methodology* and *tool*. Figure 6 shows the results of the audit covering RQ 2. For the second evaluation, a questionnaire is used that requires the students to change their perspectives from the *Process Engineer* view to the *Process Consumer* view.

**RQ 1:** Research question 1 considers the analysis of the strengths and limitations of one strategy compared with the other when constructing a process. This construction covers the analysis of a process, its conceptual design, and its tool-supported implementation. For the evaluation, we distinguish between the aspects *methodology* and *tool*. The first aspect thereby covers the approach to transfer an analysed process and to implement an analysed process in a given process framework. The ratings for the first aspect are shown in Tab. 3 and Fig. 4.

The ratings show that the artefact-oriented strategy behind the V-Modell XT provides better support for the transfer and implementation of its proclaimed basic process elements, i.e. roles, artefacts (work products), and relations/dependencies between the process elements. The questions Q1-9 (completeness of roles), Q1-10 (completeness of artefacts), and Q1-11 (completeness of relationships) have the maximum rating of 7,

while the EPF framework especially suffers in the area of relations/dependencies (Q1-11: 6.2). Regarding the activity parts and the overall process, both frameworks provide comparable support for process engineers. Since EPF is basically built on the activity-oriented strategy, the framework better supports the implementation of activities and workflows and, in consequence, shows a higher rating for Q1-12 (completeness of activities), while the V-Modell XT only allows the implementation of one activity per work product, but does not support the design of workflows.

Regarding the overall process design, both frameworks do not allow for a straightforward implementation, but require the process engineer to select one of the possible approaches to model the process (e.g., copied vs. referred capability pattern in EPF, or non-/hierarchical procedure modules in the V-Modell XT). The question whether the designed process could be easily implemented in general, is undecided with a slightly better rating for EPF (Q1-8, overall completeness: 6.2, to 6.0 for the V-Modell XT).

Table 3: Results for closed questions for process engineers (RQ 1, methodology).

No.	Rating EPF	Rating V-Modell XT
Q1-8	6.2	6.0
Q1-9	6.8	7.0
Q1-10	6.7	7.0
Q1-11	6.2	7.0
Q1-12	6.6	6.5
Q1-13	6.4	6.5

Table 4: Results for closed questions for process engineers (RQ 1, tool).

No.	Rating EPF	Rating V-Modell XT
Q1-2	4.4	5.5
Q1-3	4.8	4.0
Q1-5	4.8	3.5
Q1-6	4.6	6.0
Q1-7	6.6	4.0
Q1-14	5.4	6.5
Q1-15	5.8	6.0
Q1-16	5.0	2.5

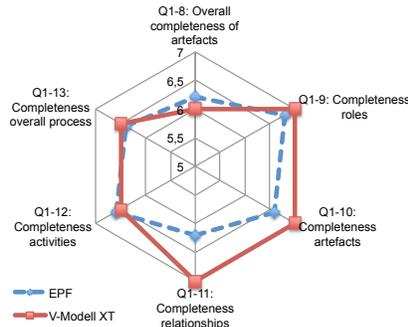


Fig. 4: Radar chart with the results for research question 1 (aspect: methodology).

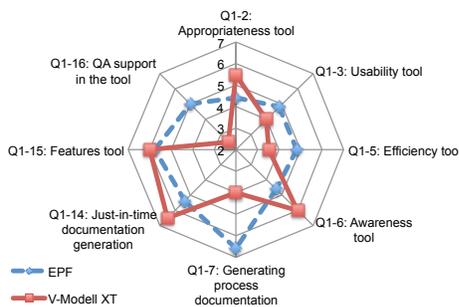


Fig. 5: Radar chart with the results for research question 1 (aspect: tool).

The second aspect covers the appropriateness of the tool environment used for the realisation. The results are shown in Tab. 4 and Fig. 5. The results for the second aspect show that the EPF tool was easier to use for the process design in terms of usability (Q1-3), efficiency (Q1-5), generating process documentation (configurability, Q1-7), and

quality assurance (Q1-16). While most of the subjects were familiar with the Eclipse platform and, thus, had a steeper learning curve w.r.t. technical details, the editor for the V-Modell XT is an expert tool on a fairly basic (technical) level. In consequence, the usability of the EPF tool is better, but the appropriateness (Q1-2) and the awareness (Q1-6) are, due to the direct implementation, better. The data also shows that the just-in-time export of a process (Q1-14) was better rated for the V-Modell XT, while the flexibility of the export (Q1-7) was better rated for EPF.

**RQ 2:** Once the process is constructed, we want to know to which extent the design strategies support the usage of the generated process (documentation) from the consumers' perspective. The ratings of the second questionnaire are shown in Tab. 5 and Fig. 6.

Table 5: Results for closed questions for process user (RQ 2).

Question	Q2-1	Q2-2	Q2-2	Q2-4	Q2-5	Q2-6	Q2-7	Q2-8	Q2-9	Q2-10	Q2-11
<b>Rating EPF</b>	4.67	5.00	6.00	6.33	7.00	7.00	6.67	5.67	6.33	6.67	6.00
<b>Rating V-Modell XT</b>	4.20	3.80	3.00	4.00	5.20	6.00	6.20	4.40	3.80	4.75	4.67

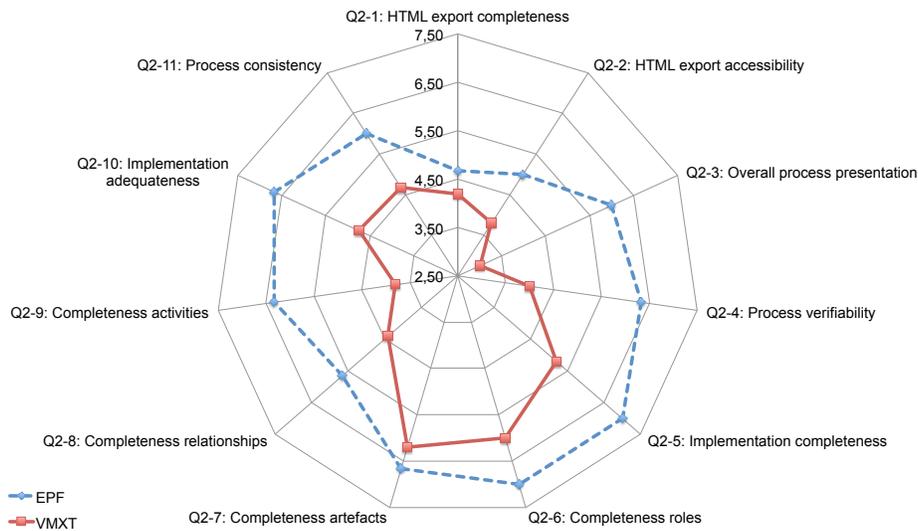


Fig. 6: Radar chart with the results for research question 2.

The data show that for all questions the EPF-based process is better rated than the V-Modell-XT-based process. Significant questions are Q2-3 (overall process presentation – EPF: 6.0, V-Modell XT: 3.0) and Q2-9 (completeness of activities – EPF: 6.33, V-Modell XT: 3.80). Even regarding the completeness of the artefacts (Q2-7), the EPF-based process is better rated than the V-Modell-XT-based process (EPF: 6.67, V-Modell XT: 6.20). Referring to RQ 1, where the implementation capabilities w.r.t.

artefacts and dependencies were rated better for the V-Modell XT (Q1-9 – Q1-11), the data does not underpin similar effects for the users of the process.

#### 4.4 Evaluation of Validity

In this section, we evaluate our findings and critically review our study w.r.t. threats to validity.

*Construct validity.* Regarding the construct validity, we cannot guarantee that the assessment criteria chosen in the questionnaire are complete enough to answer our research questions. The chosen criteria, however, rely on our experiences in the analysed design strategies (see, e.g. [12]) and were narrowly defined, whereby we consider this threat as a minor one. Regarding the process chosen as a case, we believe that it sufficiently covers the complexity of real-life processes.

*Internal validity.* There are different threats to the internal validity we are aware of. For instance, the authors themselves have deep knowledge about the technical infrastructure of the V-Modell XT as they directly participated in its development. We minimised this threat by giving introductory talks about both infrastructures and serving as coaches during the implementation workshops for both tools, but without direct interference during the process implementation. Furthermore, the students were allowed to choose the framework on basis of their own experiences and preferences potentially leading to a bias towards the frameworks and the exported documentation. We minimised this threat by opting for a research triangulation during the evaluation. Another threat to be considered is that technical mistakes during the export of a process documentation could have distorted the results in the audit. In fact, the V-Modell XT export did not follow, for example, the standard stylesheet. However, all information were accessible while non-functional aspects in the design of the tools and exported documentation were not rated.

*External validity.* Regarding the external validity, one major concern is that we have differing numbers of students in the different groups (EPF: 5 students, V-Modell XT: 3 students) and a generally low number of analysed samples. With our experiment, we cannot allow for generalisation. However, our intention was to get first insights into a field, which is, by now, barely investigated.

## 5 Conclusions

We conducted a controlled experiment in which a group of students analysed, conceptualised, implemented, and assessed a software process in a comparative manner. To investigate the perceived value of process modelling and process consumption, we selected two process frameworks covering the artefact-oriented and the activity-oriented strategy to design and implement the given process. Both process frameworks support the deployment of a designed process into the html-format, which we used to investigate the value from the perspective of process consumers. Our results indicate that the

artefact-oriented strategy, implemented with the V-Modell XT framework, seems to be perceived of high value to serve process engineers. For example, the tool-supported realisation of an artefact model, role model, and explicitly defined relationships among those models is better supported by the artefact-oriented strategy. Still, our results do not show similar effects from the perspective of the process consumers. Even the artefacts' completeness and especially the responsibilities, e.g. relationships to roles, were rated to be better accessible using the activity-oriented framework.

Our findings imply that the assessment of a process design strategy in process crafting needs not to coincide with the perceived value of the resulting process (documentation) from the viewpoint of a process consumer. It may well be that the artefact-oriented approach accommodates better the request of process engineers for declarative goal-driven design—to “have something to talk about”—whereas process consumers prefer clear cut procedures.

**Impact/Implications.** We opted for a controlled experiment to explicitly simulate a context where, on the one hand, a pre-defined process was given, and, on the other hand, this given process was not implemented and maintained at all. That is, no particular strategy existed as it is often the case in industrial environments. This gave us the possibility to investigate the perceived value of a strategy without any distortion by given expectations and experiences. Inherited from the nature of a controlled experiment with a limited scope is the need to further replicate the study in other contexts.

The replication should further proof the current working hypothesis that the selection of a strategy for establishing an effective process management does not affect its actual consumption.

This will put us in the situation to empirically infer improvement goals, not only for software process meta models, but also for the process development environments, even already established ones such as EPF (see, e.g. Q1-15 and Q1-16).

**Limitations.** The experiment has several limitations that arise from the particular setting in which we performed the experiment without any side effects. The setting was characterised by limited time, a small number of participants and the choice of two representative process frameworks. Also, we did not cover all features provided by the frameworks that could have impacted the consumers' opinion, e.g., tailoring capabilities and project set-up. Hence, this first controlled experiment does not yet allow for generalisations of our results, but reflects our experiences in practical software process improvement endeavours and is therefore a first step into this direction.

**Future Work.** As future work, we plan to extend our experiment with further empirical studies to support for a long-term generalisation of the results. To this end, we (1) will conduct a survey to determine a taxonomy of characteristics important to process engineers and users, and (2) plan to perform further experiments with student groups involving several meta model-based frameworks to ground our data, before (3) extending our experiments to industrially hosted environments. Besides covering the engineering level, we need further investigation at the application level.

## References

1. C. Braun, F. Wortmann, M. Hafner, and R. Winter. Method Construction — A Core Approach to Organizational Engineering. In *Proc. 20<sup>th</sup> ACM Symp. Applied Computing (SAC'05)*, pages 1295–1299. ACM, 2005.
2. Brinkkemper, S. and Saeki, M. Meta-Modelling Based Assembly Techniques for Situational Method Engineering. *Inf. Syst.*, 24(3):209–228, 1999.
3. Confédération Suisse. The HERMES Method. Online: <http://www.hermes.admin.ch>, 2011.
4. Eclipse Foundation. Eclipse Process Framework (EPF). Online, <http://www.eclipse.org/epf>, 2010.
5. R. Foorhuis, S. Brinkkemper, and R. Bos. An Artifact Model for Projects Conforming to Enterprise Architecture. In *Proc. 1<sup>st</sup> IFIP WG 8.1 Working Conf. Practice of Enterprise Modeling (POEM'08)*, volume 15 of *Lect. Notes Business Inf. Proc.*, pages 30–46. Springer, 2009.
6. German Federal Ministry of the Interior. V-Modell XT. Definition and Documentation on the Web. Online, <http://www.v-modell-xt.de>.
7. B. Henderson-Sellers and J. Ralyte. Situational method Engineering: State-of-the-Art Review. *J. Univ. Comp. Sci.*, 16(3):424–478, 2010.
8. P. Kroll and P. B. Kruchten. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley, 2003.
9. M. Kuhrmann, D. Méndez Fernández, and J. Münch. Teaching Software Process Modeling. In *Proc. 35<sup>th</sup> Int. Conf. Software Engineering (ICSE'13)*, pages 1138–1147. IEEE Computer Society, 2013.
10. M. Kuhrmann, D. Méndez Fernández, and R. Steenweg. Systematic Software Process Development: Where Do We Stand Today? In *Proc. Int. Conf. Software and Systems Process (ICSSP'13)*. ACM Press, 2013. In print.
11. D. Méndez Fernández, K. Lochmann, B. Penzenstadler, and S. Wagner. A Case Study on the Application of an Artefact-Based Requirements Engineering Approach. In *Proc. 15<sup>th</sup> Ann. Conf. Evaluation and Assessment in Software Engineering (EASE'11)*, pages 104–113. IET, 2011.
12. D. Méndez Fernández, B. Penzenstadler, M. Kuhrmann, and M. Broy. A Meta Model for Artefact-Oriented: Fundamentals and Lessons Learned in Requirements Engineering. In *Proc. 13<sup>th</sup> Int. Conf. Model Driven Engineering Languages and Systems (MoDELS'10)*, volume 6395, pages 183–197. Springer, 2010.
13. B. Nuseibeh and S. Easterbrook. Requirements Engineering: A Roadmap. In *Proc. 22<sup>nd</sup> Int. Conf. Software Engineering (ICSE'00) — Future of Software Engineering Track*, pages 35–46. ACM, 2000.
14. Object Management Group. Software and Systems Process Engineering Metamodel (SPEM) Specification v2.0. Technical Standard formal/2008-04-01, OMG, 2008.
15. J. Ralyte and C. Rolland. An Assembly Process Model for Method Engineering. In *Proc. 13<sup>th</sup> Int. Conf. Advanced Information Systems Engineering (CAiSE'01)*, volume 2068 of *Lect. Notes Comp. Sci.*, pages 267–283. Springer, 2001.
16. P. Runeson and M. Höst. Guidelines for Conducting and Reporting Case Study Research in Software Engineering. *Emp. Softw. Eng.*, 14(2):131–164, 2009.
17. P. Tell and M. Babar. Activity Theory Applied to Global Software Engineering: Theoretical Foundations and Implications for Tool Builders. In *Proc. 7<sup>th</sup> Int. Conf. Global Software Engineering (ICGSE'12)*, pages 21–30. IEEE Computer Society, 2012.
18. T. Ternité and M. Kuhrmann. Das V-Modell XT 1.3. Metamodell. Technical Report TUM-I0905, Technische Universität München, 2009. In German.