Software Qual J (2012) 20:229–260
DOI 10.1007/s11219-011-9147-6

Requirements and constructors for tailoring software processes: a systematic literature review

Tomás Martínez-Ruiz · Jürgen Münch · Félix García · Mario Piattini

Published online: 19 June 2011

Abstract Organizations developing software-based systems or services often need to tailor process reference models—including product-oriented and project-oriented processes—to meet both their own characteristics and those of their projects. Existing process reference models, however, are often defined in a generic manner. They typically offer only limited mechanisms for adapting processes to the needs of organizational units, project goals, and project environments. This article presents a systematic literature review of peer-reviewed conference and journal articles published between 1990 and 2009. Our aim was both to identify requirements for process-tailoring notation and to analyze those tailoring mechanisms that are currently in existence and that consistently support process tailoring. The results show that the software engineering community has demonstrated an ever-increasing interest in software process tailoring, ranging from the consideration of theoretical proposals regarding how to tailor processes to the scrutiny of practical experiences in organizations. Existing tailoring mechanisms principally permit the modeling of variations of activities, artifacts, or roles by insertion or deletion. Two types of variations have been proposed: the individual modification of process elements and the simultaneous variation of several process elements. Resolving tailoring primarily refers to selecting or deselecting optional elements or to choosing between alternatives. It is sometimes guided by explicitly defined processes and supported by tools or mechanisms from the field of knowledge engineering. The study results show that tailoring notations are not as mature as

T. Martínez-Ruiz (✉) · F. García · M. Piattini
Alarcos Research Group, Department of Information Technologies and Systems,
Escuela Superior de Informática, University of Castilla-La Mancha, Paseo de la Universidad 4,
13071 Ciudad Real, Spain
e-mail: tomas.martinez@uclm.es

F. García
e-mail: felix.garcia@uclm.es

M. Piattini
e-mail: mario.piattini@uclm.es

J. Münch
University of Helsinki, Gustaf Hällströmin katu 2b, 00014 Helsinki, Finland
e-mail: juergen.muench@cs.Helsinki.fi

the industry requires if they are to provide the kind of support for process tailoring that fulfills the requirements identified, i.e., including security policies for the whole process, or carrying out one activity rather than another. A notation must therefore be built, which takes these requirements into consideration in order to permit variant-rich processes representation and use this variability to consistently support process tailoring.

Keywords Systematic review · Software process · Process tailoring · Variability · Process lines · Variant-rich processes · Project management

1 Introduction

At present, if a software-intensive product is to take its place in the market, then quality is an important characteristic. Since the quality of the software process influences the quality of the final product (Fuggetta 2000), software development organizations develop software by using capable software processes, which are those *satisfying their specified product quality, service quality, and process-performance objectives* (SEI 2004)). Enterprises also know that capable processes help them to improve their maturity, i.e., "the ability of an organization to enact capable processes" (SEI 2004), and image, thus allowing them to stand out over the competition.

But applying processes in an organization is no trivial task. Several software process reference models have been created to help organizations apply process-oriented management, (Sheard and Lake 1998), such as CMMI (SEI 2004) or ISO 12207 (ISO 2002). These process reference models often include the description of generic processes taken from industry. This being so, a software organization needs to tailor its processes (Yoon et al. 2001) before applying them in order to meet the characteristics of each of its enactments (Osterweil 1987). Those characteristics affecting process enactments could be classified into three categories. The first one includes those factors that are related to the project itself, such as time constraints, or risks affecting some parts of the process; the second category lists the factors to do with the organization, such as the number of available resources, their culture; and the last category includes the social, legal, and political contexts in which the organization is situated.

Moreover, once processes have been established, tailoring is then also necessary since, according to assessment models such as SCAMPI (SEI 2001) or ISO 15504 (ISO 2004), in order to continue improving processes, "they must be adapted from a standard set of processes of the organization" (Chrissis et al. 2006). This signifies that process tailoring is an important activity in both the establishment and the improvement of software processes and must therefore be considered in any software organization.

Considering the importance of tailoring when it comes to achieving and executing capable processes, the actions involved must be performed in the best-engineered manner. This implies defining, establishing the scope, and executing changes in a systematic and consistent manner, in order to adapt the process models to the unique characteristics of the project in a given context. This is known as process variability.

However, although several initiatives regarding software process tailoring in software processes exist in the literature, such as Simidchieva et al. (2007), Baldassarre et al. (2002), Schnieders (2006), or Lazovik and Ludwig (2007), there is no united approach or consensus regarding how to perform process tailoring in a controlled and consistent manner nor is there a complete notation that supports it.

In this article, we present a systematic literature review (SLR) dealing with the efforts that have been undertaken to tailor software process models to fit the characteristics of organizations or projects. The objective is to analyze these adaptations to discover how they were performed and to observe the requirements and the constructors that were considered. The main contribution expected is to describe the state-of-the-art in process tailoring, stating a complete set of requirements that need to be fulfilled by a tailoring notation, according to the particular needs identified in relevant tailoring initiatives. The remainder of this article is set out as follows. Section 2 presents related work. Section 3 describes the research question formulation of this SLR, whose planning and execution stages are presented in Sect. 4. Section 5 presents the results obtained and a discussion of them. Section 6 deals with the limitations of this work. Finally, our conclusions and future work are outlined in Sect. 7.

2 Related work

A systematic literature review (SLR) is an exhaustive method that is used to search for all relevant results regarding a research topic (Kitchenham et al. 2002). In addition, an SLR will also provide information concerning the actual interest in this topic by means of considering trends and ensure the high scientific value of the results obtained. SLRs can also be the basis for developing *systematic mapping studies* (Budgen et al. 2008).

To the best of our knowledge, systematic reviews have not yet been widely applied to the analysis of the state-of-the-art in software processes. The only existing related work deals with *gathering and analyzing the tools, techniques, approaches*, *and experiences for the tailoring of software process* (Pedreira et al. 2007). Its results show that tailoring is carried out firstly at an organizational level and later on projects; tailoring initiatives try to include some formality; and tailoring is more widely applied in large organizations. This related work therefore suggests that tailoring is a real need in software processes, but how to perform this is not within its scope.

The results obtained from the aforementioned SLR motivated us to carry out this work in order to obtain the most exact and in-depth description of tailoring initiatives in software processes and to derive from them the requirements of a complete tailoring notation. Since that was the case, the most suitable research method was that of the systematic literature review. This SLR also describes the current trends related to the topic.

3 Research question formulation

The main research question (MRQ) in this SLR was "Which approaches for tailoring and adaptation software process can be found in the literature and which variability requirements do they imply?" In order to obtain the relevant information to answer the main research question, this was divided into four specific research sub-questions:

- SRQ1: Which elements of the software process model are used when adaptation is considered?
- SRQ2: What kinds of tailoring operations are used in existing process models?
- SRQ3: Which process-modeling notations are used as a basis for supporting tailoring in process models?
- SRQ4: How are processes tailored to meet organizational or project characteristics?

A detailed knowledge of how tailoring is carried out in software processes involves studying several different aspects in depth. Each of the research-specific sub-questions therefore deals with these aspects in detail and can provide us with detailed knowledge of how variability (or other ad hoc mechanisms) has been used for tailoring software processes.

The SRQ1 sub-question focuses on obtaining the key process elements that need to be varied in order to tailor the process, i.e., *which* software process elements are used for tailoring software processes. With regard to this specific sub-question, it is important to take into account that processes are the conjunction of three main elements: work elements (activities, steps, etc.), roles, and products. It is of interest to know whether these elements vary individually and/or in groups, along with which of them are the most significant when tailoring a process. This specific sub-question could confirm these aspects.

SRQ2 addresses *how* the process elements vary, signifying the actions (add, modify, replace, delete) that are applied to such elements when tailoring the software process.

SRQ3 allows us to analyze to *which* modeling *notations* and process reference models tailoring mechanisms have been applied.

And finally, the SRQ4 sub-question has to do with in *what way* software processes are tailored. The aim is to discover what steps need to be followed in order to apply the tailoring mechanisms (from sub-question 2) to the process-composing elements (from sub-question 1), along with the techniques or tools used, if any.

The results obtained from these specific sub-questions provide a description of software process-tailoring initiatives from different points of view, which complement each other. They also provide a guide as to how to present the results obtained.

The description of the existing process-tailoring initiatives provides the necessary information with regard to how tailoring must be supported. The requirements to be fulfilled by a tailoring notation that supports, guides, and facilitates process tailoring can then be extracted from these tailoring initiatives.

4 Planning and executing the review regarding tailoring in software processes

The protocol of Kitchenham (2004) has been used to permit a far-reaching discovery of any initiatives regarding how to tailor software processes, covering different aspects related to this topic. The documentation of this systematic review was also guided by the template developed by Biolchini et al. (2005).

4.1 Systematic review protocol

The systematic review protocol followed in this SLR was based on the iterative and incremental procedure defined in (Pino et al. 2008).

The SLR was carried out in four iterations, each one corresponding to the respective source (bearing in mind that the searches in the ACM Digital Library were already supported by the IEEE Digital Library engine). Before starting the iterations, a pilot search was performed, and after checking that it returned diverse relevant studies that were previously known by the authors of this SLR, no refinements to the search string were considered. Each iteration included the following steps: firstly, the search string (Sect. 4.2.1) was adapted to the particular search engine (if necessary), and when results were returned, duplicate papers were removed from the set of studies *found*. The inclusion and exclusion criteria (Sect. 4.2.2) were then applied in order to select the *primary* studies. Once the studies had been selected (Sect. 4.2.3), the data extraction began, which included the quality assessment (Sect. 4.3.2) of the studies and the information extraction (Sect.

4.2.4), along with the analysis of the results obtained (Sect. 4.2.5). The final report was created by means of aggregating the partial results obtained in each iteration.

This SLR was carried out by the first author of this paper; the other authors played the role of experts who reviewed the work. All of them took part in several meetings in each iteration in order to corroborate both the study selection and the data collection and analysis.

4.2 Systematic review execution

4.2.1 Keywords, search string, sources, and selection criteria

The goal of this SLR stipulates that the selected keywords must be oriented toward finding initiatives containing both *software process* and *tailoring* and that these must be joined by the logical AND. The tailoring part of the search string also considered tailoring synonyms (joined by OR). As a result, the string was stated as: "'software process' AND ('variability' OR 'variant' OR 'variation point' OR 'tailoring' OR 'flexibility' OR 'customization' OR 'process instantiation' OR 'adaptation')". In order to apply this question to the different search engines, it was suitably adapted and divided when necessary. The search string was applied on the title, abstract, and keywords of the articles. When this was not possible, the search string was applied on the full paper, in order to ensure that no potentially relevant papers were excluded.

The initial list of sources identified by experts in the area as a basis for conducting the review was as follows:

- Science Direct
- Wiley InterScience
- Springer-Link
- IEEE Digital Library
- ACM Digital Library (in conjunction with searches of IEEE)

These sources included the proceedings of the conferences and journals identified as being especially relevant in the area of software processes, such as the Software *Process Improvement & Practice Journal*, *the International Conference on Software Processes* (ICSP), the *International Software Process Workshop* (ISPW), and the *International Workshop on Software Process Simulation and Modeling* (SPW/ProSim).

The Systematic Review was executed in four iterations in which Science Direct, Wiley InterScience, Springer-Link, and IEEE Digital Library in conjunction with ACM Digital Library were searched (see Sect. 4.2.2).

A study obtained from the search engine was considered to be relevant (i.e., to satisfy the inclusion criterion) if, after analyzing the title, abstract, and key words, it was related to software process tailoring according to its application to one specific (whatever) context, from theoretical to practical perspectives.

Those studies whose titles made it clear that they were not related to the topic involved in this systematic review (exclusion criterion) were excluded from this study. This is especially important with regard to those studies discussing any process in product lines or product tailoring, which were beyond the scope of this work. Duplicated papers of the same work were also removed, and only the most comprehensive version was included in the SLR.

4.2.2 Study selection and data extraction

After executing the procedure for gathering primary studies and applying the inclusion and exclusion criteria, 32 primary studies were obtained. Table 1 describes the articles found in each iteration for each source.

Three aspects concerning the study selection should be highlighted. First of all, a lot of the papers returned discussed software product lines (SPLs) and these were excluded, in accordance with the stated criteria. The reason for obtaining such a large quantity of non relevant papers may be that keywords of this SLR may have matched keywords in the field of SPLs. The second aspect deals with the appropriate use of synonyms of tailoring in the search string. The inclusion of a complete set of tailoring synonyms guaranteed the completeness of the results returned, although various synonyms did not appear in some papers, such as “variation point”.

Finally, the last aspect was that the IEEE Digital Library does not allow us to search only within the abstract, so the search was applied to the full text. Moreover, it returns a fixed number of papers as a result of each search (100 papers), meaning that the total number of papers returned in our searches was 800 (Table 1). After reviewing the title, abstract, and keywords, the number of studies that included the keywords in the abstract (as the inclusion criteria states) was not more than about 25 papers per search.

The information from each selected primary study was extracted by following the protocol. The information was stored in several tables, one for each specific research sub-question and an additional one in which to store the data from each study such as title, authors, year, and publication as is shown later. Appendix 3 includes an excerpt of these tables. In addition to this, a template was designed to store the textual results obtained from each document, which contained information concerning the identification of the studies with relevant objective and subjective results. The *meta-summary* method (Sandelowski et al. 2007) was used to extract subjective information from the studies.

The method of *coding* was used to insert information into the tables. The most suitable codes related to each specific research sub-question were identified according to the knowledge of authors in the field and were then placed in their corresponding tables before starting the data extraction step of the first iteration. Once the data extraction started, each paper was analyzed and its results placed in its corresponding tables; newly discovered relevant codes were added. An excerpt of the most relevant codes used is as follows:

- Identification Table:
    - Identification of the study (full reference)
    - Source

Table 1 Summary of the studies dealt with in this SLR

No. of iteration	Source	Search String	Retrieved	Duplicates	Unique	Primaries
1	Science Direct	1	18	0	18	3
2	Willey Intersc.	1	185	41	144	6
3	Springer-Link	1	169	20	149	8
4	IEEE + ACM DL	1	800	75	725	15
Total	All	1	1,172	136	1,036	32

Search String 1: “software process” and (“variant” or “variability” or “variation point” or “tailoring” or “flexibility” or “customization” or “process instantiation” or “adaptation”)

    - Paper type (T1, T2, or T3)
    - Authors
- SRQ1 Table
    - Elements being used for tailoring (activity, artifacts, roles…)
    - Elements driving the variation (activities, artifacts, roles…)
- SRQ2 Table
    - Types of operations carried out (add, replace, delete, modify)
    - Approaches they were based on (product lines, components…)
- SRQ3 Table
    - Process-modeling notation.
    - Process model (CMMI, ISO 12207…)
- SRQ4 Table
    - Tailoring steps and strategies.
    - Use of supporting approaches (knowledge base, …)
    - Causes motivating the tailoring (to fit a project or organization)

4.2.3 Data analysis

Data analysis focused on providing suitable information to answer the four research sub-questions. In addition, publication trends were derived by tabulating the publications, year, and study type.

The specific research sub-questions were answered by analyzing the frequency of the presence of the codes stated in the studies. Since several similar codes were generated in the information extraction, these were grouped according to their similarities. Different aspects in each specific sub-question were dealt with separately with the objective of making the results more understandable.

Since the objective of this review is to state the requirements for a process-tailoring notation, all the relevant information about it was synthesized. First of all, we considered significant requirements to be those dealt with in several papers, and their relevance was therefore analyzed quantitatively. However, specific relevant ideas appearing in only one or a few papers were also considered when stating the requirements.

4.3 Evaluation of results

4.3.1 Reliability of the study selection process

The reliability of the study selection process was ensured in two ways. Firstly, if no decision could be made as to whether or not to include a study as a result of this analysis, it was included conditionally, meaning that the article was read in its totality and was deleted if it did not actually fit the objective of this review. Secondly, the studies were selected through a process of agreement between the main author and the others, who are experts in the research area. Disagreements were resolved by reevaluating the studies.

According to the protocol, the interrater reliability was verified once the set of studies had been set from each source and before beginning to extract the information. This was done in order to assess the reliability, by considering all the papers selected from the source.

4.3.2 Quality assessment

Each article was assessed by using three quality assessment questions.

QA1 Does the paper describe the tailoring approach used in tailoring the software processes?

QA2 Is the tailoring initiative detailed enough? This quality assessment question is divided into four:

- QA2.1 Does the tailoring initiative include sufficient data to infer that the process-composing elements are used in tailoring software processes?
- QA2.2 Does the tailoring initiative clearly describe the actions used to tailor processes?
- QA2.3 Does the tailoring initiative define the process model to be tailored, as well as the process-modeling notation used?
- QA2.4 Does the tailoring initiative specify how the tailored process was obtained?

QA3 Does the paper illustrate the tailoring application by means of an empirical study?

These questions were scored as follows. 1, if the question was completely satisfied, 0.5 if it was partially satisfied, and 0, if it was not satisfied at all.

Based on the evaluation of the QA3, three types of studies were identified according to their treatment of process tailoring.

- T1: Studies in which adaptation mechanisms are proposed to process models, but where these adaptations are not applied to real processes.
- T2: Studies in which adaptations are proposed for real process models and which show how these are carried out by means of some examples.
- T3: Studies discussing previously performed real process tailoring.

The analysis of these types allowed us to show trends concerning the empirical validation of software process tailoring (see Sect. 5.1).

4.3.3 Reliability of the information extraction process

As the extracted information was classified to satisfy the four specific research sub-questions presented, reliability was first set by ensuring that the studies offered sufficient information in order to be able to answer all the sub-questions. The results obtained were also discussed and agreed upon by all the authors.

5 Results and interpretation

This section summarizes the results obtained after conducting the systematic review. The analysis of the results focuses on answering the main research questions shown in Sect. 3, and their specific research sub-questions. A discussion of the publication trends is presented first, and the following four sub-sections then deal with answering each of these sub-questions.

5.1 Overview of the literature identified

The relevant papers that have been dealt with have been published over the last two decades, but it is worth noting that half the studies were published between 2005 and 2009. This is an indicator to demonstrate that there is a growing interest in software processes variability, as shown in Fig. 1.

The studies were classified according to their treatment of process variability and the year of publication, as displayed in Fig. 2. The results show that theoretical studies (Type 1) were the type published most often in the early 1990s; studies that include theoretical results and some attempts to transfer them into practice (Type 2) began to be created during the 1990s and were the kind published most often in the early twenty-first century; similarly, Type 3 studies reporting experiences related to this tailoring began to be published around the turn of the new century but have not yet become predominant over Type 2 studies.

Publications related to process variability are therefore evolving from the experts' proposal to their application in industry and reports of experiences of these applications.

5.2 Software process elements used in tailoring processes

Software processes are composed of several elements and relationships, and these are affected when processes are tailored. This sub-section therefore deals with the identification of the single elements or relationships that vary, along with their importance in adapting processes. The specific research sub-question tackled is "SRQ1: Which elements of the software process model are used when adaptation is considered?"

The study first focused on the single elements of which a software process is made up and which present some variations in the studies analyzed. Each study proposes varying several and different process elements. In order to establish a common understanding, the definition of process given by Fuggetta (2000) and the main process elements given by CMMI (SEI 2004) were adopted. Process-composing elements were thus classified as *work elements* (which include *activities*, *tasks*, *sub-processes*, or *steps* according their definition in the different studies analyzed), *roles*, *artifacts*, *resources*, and *techniques and practices*. The process composition elements that vary in the studies were matched to these categories, as Fig. 3 summarizes.

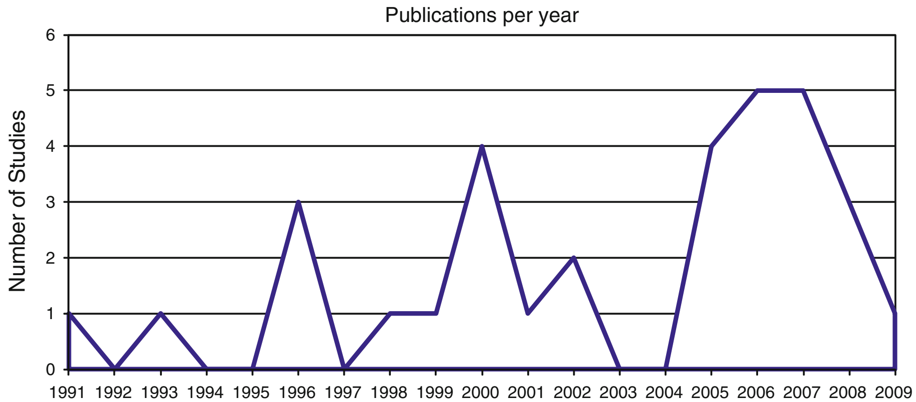


Fig. 1 Number of publications per year

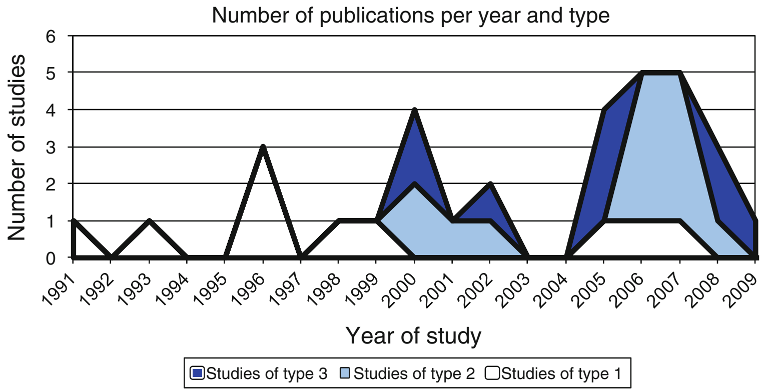


Fig. 2 Publications per type and year

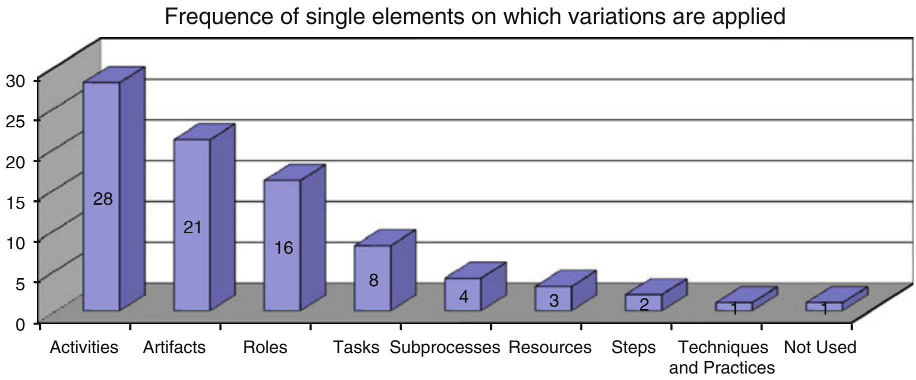


Fig. 3 Single process elements to vary

The results in Fig. 3 show that activities were modeled as variable elements (87.50%), followed by artifacts (65.63%) and roles (50.00%). In short, 31 out of 32 studies propose variations in work elements (including activities, tasks, sub-processes, or steps), which highlights the importance of tailoring these elements if the entire process is to be adapted. It should also be noted that other process elements such as resources or techniques and practices are modified in only a few studies. One reason for this might be that many process models are not on the level of technical activities. The relationships between these process-composing elements when variations are carried out were analyzed, and the following patterns were identified:

1. First, it is interesting to note that roles vary when activities vary, signifying that roles usually vary in the context of activity variation.
2. Tasks vary according to activities except in two cases, one in which variations are treated by means of breaking down work at the task level (Plögert 1996) and another in which tasks are considered as the work unit (Sutton and Osterweil 1996). This implies that task variations are part of activity variations.
3. Steps always vary with tasks.
4. Artifacts always vary in conjunction with activities or tasks.

5. Resources, techniques, and practices always vary according to activities, tasks, artifacts, and roles.
6. Sub-processes always vary according to activities, except in (2006), in which variations are proposed by means of adding process modules to another process module. However, this module can be also considered as a sub-process.

None of the previously identified patterns belong to a specific process reference model, since we did not find any correlation between each pattern and the model to which it was applied.

We further analyzed how many process element types vary together in an approach. The results show that most of the proposals include two to four varying elements, which support the findings presented previously in which three elements (activities, artifacts, and roles) are those that vary the most. Process adaptations therefore imply that more than one element is modified and that these elements should vary in a consistent manner. It is therefore necessary to consider certain variability relationships in order to ensure consistency between these single process elements when variations occur.

Moreover, software process tailoring has not only been dealt with by means of variations in the single elements making up the process. Processes are also composed of structuring elements, which act as the glue between single elements, such as *control* or *data/product flows*, or predefined element groups such as *iterations* or *components*. The process can be also tailored by means of varying these structuring elements, as proposed in: Simidchieva et al. (2007), Hesse and Noack (1999), Giese et al. (2007), Schnieders (2006), Jaufman and Münch (2005), or Lobsitz (1996), which include variability in control or data flows; Chou and Chen (2000), Biffl et al. (2006), Park et al. (2006), Madhavji and Schafer (1991), or Dai and Li (2007) propose grouping element variations. Figure 4 shows these proposals that are classified according to the type of element that is varied, namely control and data flow variations, elements, and others.

From the results shown in Fig. 4, it can be observed that the most frequently used mechanism is variation in control flow (17 out of 32 studies, 54.84%). The second most widely used mechanism is the grouping of single elements, which appears in 9 out of 32 studies (equaling 29.03%). This includes variations in unitary elements, named in the different proposals as blocks, modules, fragments, and so on. Variations in data flow are also considered in 4 out of 32 studies (12.50%). Other proposed changes affecting relationships are variations in interface elements (2000) and constraints (1993). In addition, six studies suggest that variability should not be used in relationships, and their proposed

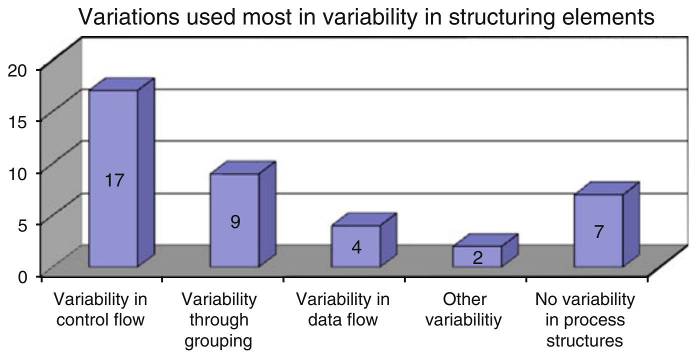


Fig. 4 Structured process elements to vary

variability therefore focuses only on elements that make up the process (signifying 19.35%).

An analysis of Figs. 3 and 4 shows that variability in single elements (applied in 31 out of 32 studies, 96.88%) is more widely used, and therefore more popular, than variability using relationships (applied in 25 out of 32 studies, 78.13%). However, both of these are used together in most of the studies (24 out of 32 studies, 75%), signifying that they complement each other to support consistent process tailoring.

It is also noticeable that in addition to proposing modifying elements, some approaches, such as Armbrust et al. (2008), Cass et al. (2002), Chou and Chen (2000), or Madhavji and Schafer (1991), suggest modifying the structures that the definition of these elements is based on, such as templates or product types. This type of modification can be designated as *crosscutting variation*, which implies the simultaneous variation in several single elements. *Crosscutting* and *single* types of variations are therefore complementary and must be used together to tailor processes when possible.

Finally, we analyzed whether the studies suggested varying any of the process elements in the first place as a main variation and then varying the others as secondary variations. In order to determine whether one element is therefore implicitly (or explicitly) driving the process tailoring, several rules were defined:

- When the tailoring mechanism is explained, and the authors focus on first tailoring one element before the others.
- When the authors place much more emphasis on one element than on the others, and the variation in this element is clearly much more important than that of the others.
- When this is explicitly stated in the study.

Upon considering the rules defined earlier, it will be noted that most of the studies do not specify the tailoring element that carries out the driving. However, in some others, sufficient evidence was obtained by applying these rules to determine the driving elements of the tailoring. Figure 5 shows the number of studies in which the driving element was determined and those in which it was not. There are ten studies in which work elements (activities, tasks, and so on) drive the tailoring and four that are driven by artifacts. One study proposes driving process tailoring by using either the activities (work) or artifacts

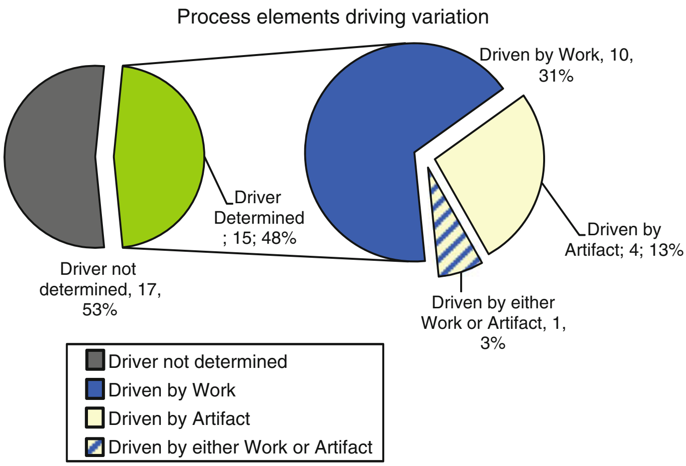


Fig. 5 Single process elements driving the tailoring

(Hausen 1998). In addition, the results show that these studies consider work elements (activities, tasks, and so on) or artifacts as being the most important process element; they tend to set them and then decide which of the other process-variable elements need to be varied. As a consequence, tailoring is supported first by providing main variations in the elements driving it and then by varying the rest of the elements according to the main ones, thus guaranteeing the consistency of the process.

5.2.1 Tailoring requirements derived from SRQ1

Table 2 shows the requirements, which were stated as a result of analyzing the information extracted to answer the SRQ1:

5.3 Tailoring operations used in existing process models

The process elements that tend to vary during process tailoring were identified in the previous section. It is also necessary to know how these process elements vary, which means ascertaining what actions may be carried out on the process elements. This section tackles the specific research sub-question "SRQ2: What kind of tailoring operations are used in existing process models?"

It is first important to underline that in some studies, the semantics of process variability is clearly taken from adaptation mechanisms in software engineering: Some of the studies analyzed take mechanisms from the product line approach (Clements and Northrop 2002), and others apply process fragment modification (Brinkkemper 1996), which is based on the paradigm of component-based software development (CBSD) (Szyperski et al. 1998).

CBSD/fragments based studies, i.e., Chou and Chen (2000), Madhavji and Schafer (1991), and Yoon et al. (2001), were applied in tailoring software processes before 2002, when the product lines approach first began to be developed. From that time on, software process variability initiatives tend to focus on applying product lines, such as in Biffl et al. (2006), Giese et al. (2007), Jaufman and Münch (2005), Schnieders (2006), or Martínez-Ruiz et al. (2008). Researchers therefore now attempt to apply product initiatives to processes rather than CBSD/fragments, as trends show.

After analyzing product line–based studies, we concluded that they are mainly based on variation points and variants to processes. Initiatives of Schnieders (2006) and Martínez-Ruiz et al. (2008) propose including dependencies and constraints in conjunction with variation points and variants to model the process variability notation. These are the main variability elements in product lines, and any approach toward modeling process lines should take them into consideration.

The proposals also define different ways in which to tailor processes by means of varying their process-composing elements. In order to structure the analysis, variations performed directly on these elements—denominated as *direct operations*—were considered first, as is shown in Fig. 6. From these results, it is interesting to highlight that insertion and deletion operations of elements are those most often used for process tailoring (26 out of 32 studies, 81.25%); the second most frequently used operation is that of modification (14 out of 32 studies, 43.75%), which implies changing some properties of the elements, without replacing them with others. Both operations are frequently used together, but two proposals do not use them. The first, Giese et al. (2007), provides an approach toward tailoring by means of inheritance relationships, and the approach of Baldassarre et al. (2002) is based on patterns of tailoring and encapsulation. The widely employed use

Table 2 Requirements answering the first specific research sub-question

Id	Requirement	Description
RQ1.1	Variability in activities/tasks	Tailoring may imply changing the way the work is performed in the process. In order to do this, activities, tasks, or any other unit of work must be variable, which implies that they must define variability
RQ1.2	Variability in artifacts	Processes produce, use and modify artifacts, and so variability throughout the process implies that their artifacts change
RQ1.3	Variability resources	Processes are executed by roles, using several tools, which are resources of the organization. These resources are always different from one project to another, which makes it necessary to consider these variations when tailoring the process
RQ1.4	Variability in control flow	Tailoring in a process does not affect one activity or task in a single way; it may affect the way the work is broken down. So, variations must consider changing the control flow of the process
RQ1.5	Variability in product flow	Tailoring in the process may affect the artifacts, and because they are related to each other, this makes it necessary to introduce variability in the product flow
RQ1.6	Variations consistency	When tailoring is done on an element, it affects the related elements, and sometimes these must vary too. The first one realizes a primary variation, and the others therefore perform a secondary variation This point is important, because large variation changes a lot of roles, artifacts, tasks, etc., and it might be of interest to discover whether any of these elements is changed because of itself, and what is due to other changes outside the particular element From the point of view of the process engineer, abstraction of the secondary variation could be an interesting option, particularly when several primary variations must be carried out to tailor a process For example: Let us imagine that we wish to tailor a process by adding a task, and this task is realized by a role. In order to ensure the consistency of the new process, if the task is added, the role must be added too. This signifies that the role is the secondary variation because of the addition of the task (primary variation)
RQ1.7	Crosscutting variations	Tailoring a process model to certain types of projects may imply making a lot of different changes in the process. Moreover, the evolution of a process may also imply that the process needs to be varied at several points of this structure. Large changes may therefore make it easier to undertake tailoring. The objective must be to tailor a high percentage of process elements, but not the entire process Furthermore, large variations must be compatible with detailed ones
RQ1.8	Single (or on-point) variations	Each project and organization is different, thus signifying that certain characteristics may be different from one process to another. Variability must therefore be able to modify any entity of the process model. Detailed tailoring allows the process engineer to carry out variations, so as to tailor single processes

of direct variation mechanisms thus means that they are powerful mechanisms for varying software processes and their composing elements.

Other operations were found, such as the definition of relationships and constraints between the varying elements (used in 9 studies, 28.13%), or element replacement (used in 4 studies, 12.50%), the latter consisting of a combination of deletion and insertion operations.

Although most initiatives propose carrying out direct variations for tailoring software processes, some studies propose using other operations, called *indirect operations*, which

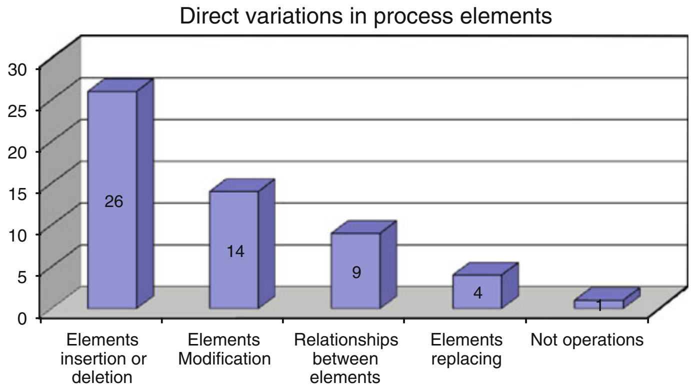


Fig. 6 Use of direct variations for tailoring software processes

are applied to change process-composing elements. Figure 7 displays the use of indirect operations.

From the results shown in Fig. 7, it is possible to assert that the most frequently used indirect operation is that of patterns (8 out of 32 studies, 25.00%). Patterns are composed of well-defined variations to be applied when certain requirements are satisfied, similarly to how patterns are applied in software design. Tailoring by means of parameterization appears in 6 studies (i.e., 18.75%), and this refers to assigning a value to certain previously defined parameters of the process when it is going to be tailored. The use of inheritance appears in four studies (12.50%) and allows us to tailor the parent process by defining child processes that extend the properties of the parent, according to each particular context. These are the second and third operations, respectively, but they are used less widely. Other operations are present in only a few studies, such as the encapsulation of activities (Schnieders 2006), which implies defining groups of activities that are dealt with jointly for the tailoring of the process, or decision nodes composed of conditions, which are used to change the flows between activities (control flows) or to change usage of products (product flow). All these mechanisms therefore have lower percentages of use, and most studies (20 out of 32 studies, 62.50%) do not even consider them.

Processes are composed of several elements; they could therefore be tailored more easily by varying one or more of these elements, as is shown in 31 out of the 32 studies.

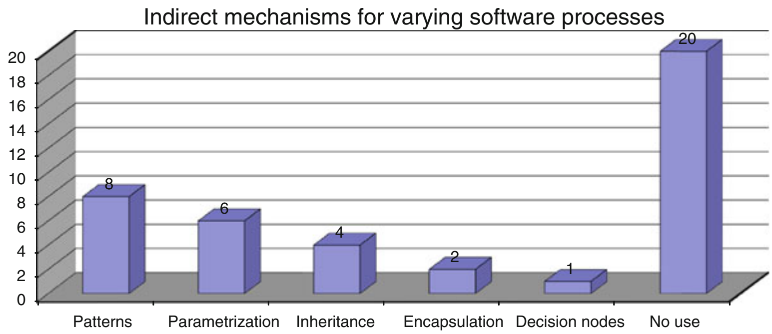


Fig. 7 Use of mechanisms that tailor the process indirectly

However, the use of indirect mechanisms in conjunction with direct mechanisms could make process tailoring easier. Bearing in mind the crosscutting and selective variations described in Sect. 5.2, the direct variation in process elements is suitable for selective variations, and the use of patterns, parameters, encapsulation, and so on (indirect operations) supports tailoring in more than one element at the same time, thus signifying crosscutting variations.

Moreover, current trends show that tailoring software processes is managed by means of variant-rich processes, based on software product lines (Clements and Northrop 2002). This is known as *process lines*, and the approach is equivalent to product lines, in which products are replaced by processes. By following this approach, variability is defined by means of variation points and variants. Three main types of variability can be considered: *optional*, *mandatory*, or *alternative*. Optional refers to elements that may or may not be present in the process, in which case the variation points may be empty; mandatory implies that the element is present in the process, and if not, another element replacing it must be in its place; alternative refers to variation points in which several alternative elements are placed.

These variant-rich process variability mechanisms are also applied to single process-composing elements, and some equivalences with direct operations (insertion, deletion, replacement, and modifiability) can therefore be stated. The matching between direct operations and variability types are shown in Fig. 8.

Furthermore, upon considering the matching described in Fig. 8, it is possible to deduce that most of the proposals include variability in line with the variant-rich processes approach. These results are presented in Fig. 9.

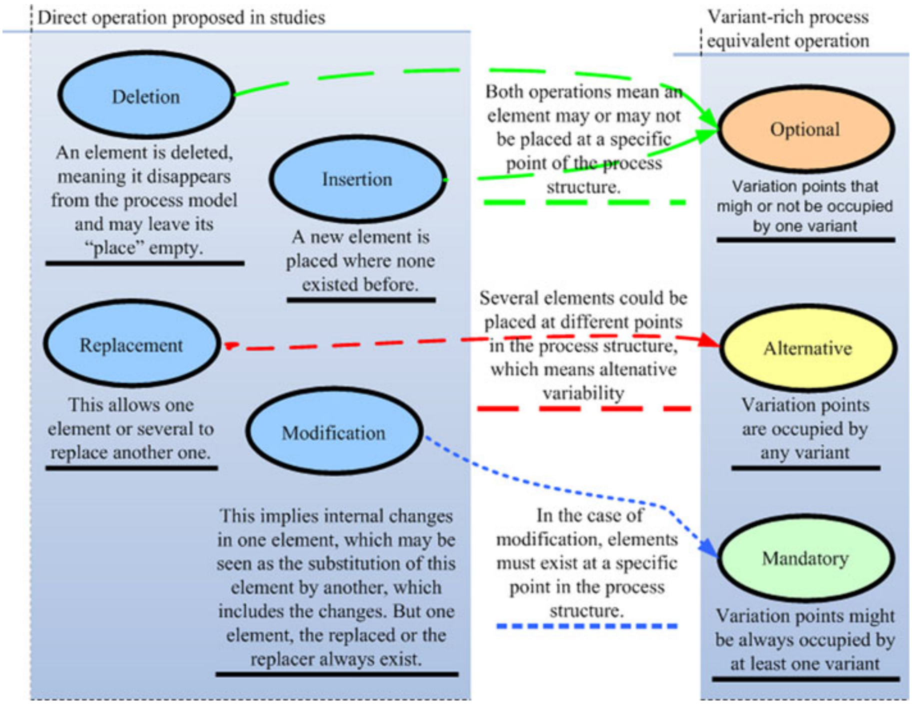


Fig. 8 Matching between process operations and variant-rich process mechanisms

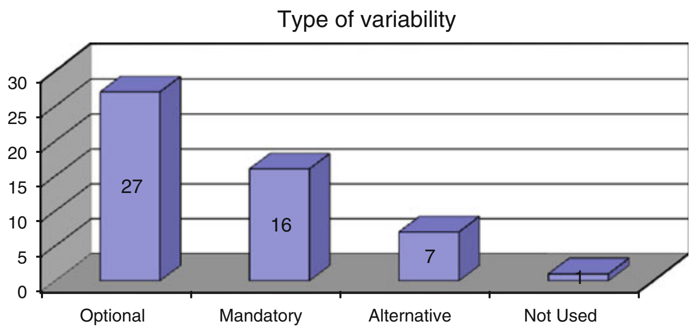


Fig. 9 Types of variability in the studies ordered by use

The aforementioned results show that the most frequently used variability mechanism (27 out of 32 studies, 84.38%) is optional: a process-composing element may or may not appear, depending on the final process desired. In addition to the optional type, the type known as alternative, which allows several elements to be treated in a common way in order to create certain processes, is included in 7 out of 32 studies, and, finally, the mandatory type is also used by 16 out of 32 studies (signifying 21.88 and 50.00%, respectively).

5.3.1 Tailoring requirements obtained after answering the SRQ2

Table 3 shows the requirements extracted from the aforementioned results.

5.4 Notations used for tailoring software processes

This section tackles the third specific research sub-question "SRQ3: Which process-modeling notations are used as a basis for supporting tailoring in process models?" Table 4 summarizes the notations used as a basis for modeling tailoring initiatives in the studies analyzed and the number of studies that include them.

As Table 4 shows, apart from studies that either only indicate tailoring steps but not the notation used to model the process or propose variability mechanisms in a generic manner (11 and 2 out of 32 respectively, 34.38 and 6.25%), a multitude of notations are used. Workflows are the most frequently used notation (4 out of 32 studies, 12.50%) but it is far from being clearly predominant. Other notations, such as code language and formal definition are used in 3 studies, signifying up to 9.38%, while Petri Nets are used in 6.25%. Finally, seven notations are used in one single study out of the 32 (signifying 3.13%). In addition, five studies propose an extension to the notation to represent variability and their characteristics, but there is no agreement between the extension proposals, and each one is based on its own modeling notation. Only SPEM, by means of method variability constructors (package *Method Plugin* and *Variability* element), and Little JIL (indicating multiplicity of sequence steps) incorporate explicit variability modeling support. The tailoring applied to other process-modeling approaches is proposed to be carried out in an ad hoc manner (see Sects. 5.2, 5.3). One conclusion of this is that there is no single modeling notation that is much more widely used than the others. The variability mechanisms must therefore be as general as possible if they are to be applicable to any modeling notation.

Table 5 provides an overview of the reported models that were subject to tailoring. It shows both the typical process models and the SPI initiatives that have been tailored in

Table 3 Requirements derived from answering the specific research sub-question SRQ2

Id	Requirement	Description
RQ2.1	Optional variation	Variability must support some elements that are included to tailor one process and not included to tailor other processes at some point in the process
RQ2.2	Alternative variation at a point in the process	Variability must support the element(s) that could be selected from among several, according to the particular characteristics of the process, at some point in the process
RQ2.3	Alternative points of a variation	Variability must also support an element that may be placed in several places of the final process, in order to tailor it to the characteristics of the project
RQ2.4	Mandatory variation elements	Because of the characteristics of the process, some elements must be included in the final process, which makes it necessary for variability to support these variations
RQ2.5	Mandatory variation places	In other cases, because of the characteristics of the project, elements must or not must be placed at some particular points of their structure
RQ2.6	Variants evolution	Variant-rich processes are composed of several variants; it is possible to define some new ones based on previously existing ones. In this case, new variants refine the characteristics and make them more usable for a given specific context
RQ2.7	Constraints	In order to ensure that the consistency of the process is being tailored from the base process, constraints must ensure a correct use of variants
RQ2.8	Dependences	Processes are composed of several elements, and variations in one of these elements affect others and depend on them
RQ2.9	Variations across elements contained	Several elements, mainly work units, may contain several other elements. The variations in the first must therefore be consistent with the elements contained in it
RQ2.10	Variability propagation	When a variation is carried out, this implies that a new element is included in the process model, and this element may offer us the possibility of tailoring its related elements, i.e., the variability is propagated. For example, variation in an activity can imply variations in the roles that perform it

order for them to be applied to real organizations. According to the aim of this SLR, we analyzed how these reported models were tailored without comparing them according to their nature (SPI framework, methodology, etc.).

These results show that typical process models, such as V-Model (Das V-Model XT v1.3, http://v-modell.iabg.de/) or RUP, and SPI frameworks, such as CMMI (SEI 2004), are of interest in process tailoring, as they appeared in 5 and 2 of 32 studies (signifying 15.63 and 6.25%, respectively), which motivates the need for tailoring as a previous important step to enact such processes. Five process models are used in a single initiative (signifying 3.13%), and 11 out of 32 studies (34.38%) use a generic process model. These results also show that if tailoring is applied to any process model, then any mechanisms supporting it must be generic, and they must differentiate common (unmovable) process parts from variable ones.

5.4.1 Tailoring requirements obtained after answering the SRQ3

Table 6 shows the requirements obtained from SRQ3.

Table 4 Notations used for modeling variability in software processes

Order	Notation	Number of studies	Percentage
1	Work flows	4	12.50
2	Code language	3	9.38
2	Formal definition	3	9.38
3	Petri Nets	2	6.25
4	Standard notations	1	3.13
4	Activity diagrams	1	3.13
4	Spearmint	1	3.13
4	BPEL	1	3.13
4	SPEM	1	3.13
4	UML state machines	1	3.13
4	Little JIL	1	3.13
	Generic	2	6.25
	Not specified	11	34.38

Table 5 Process models tailored in the studies

	Process model	Number of studies	Percentage
1	V-Model (97/XT) (Das V-Model XT v1.3, http://v-modell.iabg.de/)	5	15.63
2	RUP (Krutchen 2000)	2	6.25
3	CMMI (SEI 2004)	1	3.13
3	ISO IEC 12207 (ISO 2008)	1	3.13
3	ISO IEC 15504 (ISO 2006)	1	3.13
3	ISI IEC 12207 simplified (ISO 2008)	1	3.13
3	RUP simplified (Krutchen 2000)	1	3.13
	Generic	11	34.38
	Not specified	9	28.13

5.5 Software process tailoring process

The tailoring of processes allows us to obtain concrete processes from general ones, by means of tailoring actions that may be single or complex. In this respect, all the studies describe their own process for configuring the tailoring process. This section tackles the specific research sub-question "SRQ4: How are processes tailored to meet organizational or project characteristics?"

First, the tailoring initiatives described in the Type 3 studies (studies with experience reports from tailoring actions) focus on tailoring a software process to meet the characteristics of the organization or the project. From these initiatives, the aim of tailoring processes is to first make the process meet the characteristics of an organization (7 out of 32 studies, 21.88%) or meet the project features first (4 out of 32 studies, 12.50%). Based on these results, it would appear that any process tailoring that is carried out according to an organization's characteristics is a step that is performed prior to adapting these processes to the projects.

Table 6 Requirements obtained by answering the third specific sub-question

Id	Requirement	Description
RQ3.1	Differentiate common and variable parts	To carry out variations correctly, the common parts of all the processes of the process line and the parts that may be changed need to be differentiated. In order to do so, the notation must include specific icons for each one
RQ3.2	Variability standardization	Any variability notation must be as generic as possible in order for it to be implemented in any modeling notation (metamodel) to model and give support to variability in any process reference model

Figure 10 shows the strategies for the tailoring processes used in the studies analyzed.

The results show the operational choices used to carry out tailoring, from those most frequently used to those least frequently used. The choice most frequently applied is that of tailoring assistance (19 out of 32 studies, 59.38%), which will be dealt with specifically later. The next is the definition of several tailoring steps (9 out of 32 studies, 28.13%), which can be seen as dividing the tailoring work into more concrete and specific tailoring actions. In addition, 7 out of 32 studies (21.88%) propose the definition of several abstraction levels from a variant-rich process to a concrete process, meaning that a more concrete process is obtained during the tailoring of each of these levels. The number of proposed abstraction levels varies from 2 to 4 depending on the author. Others propose defining as many levels as necessary. Another five different strategies have also been proposed:

- Testing the tailoring process and verifying its compliance with the process reference model, as in the approaches of (Hausen 1998) and (Yoon et al. 2001). The latter also proposes carrying out several steps while checking for compliance.

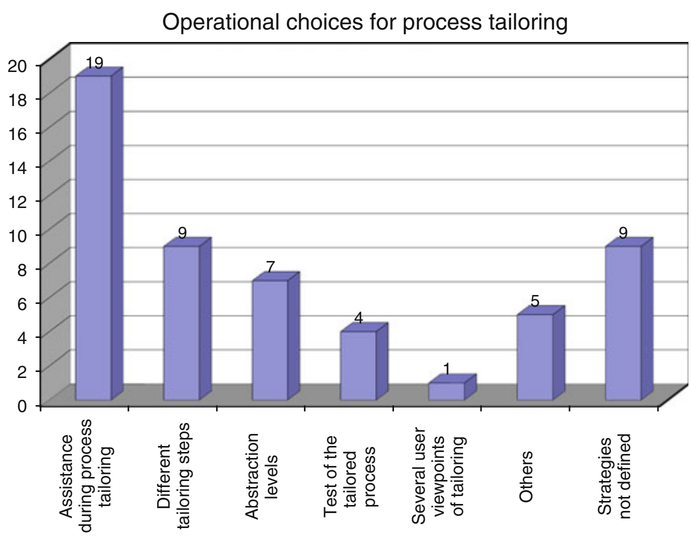


Fig. 10 Operational choices for tailoring processes

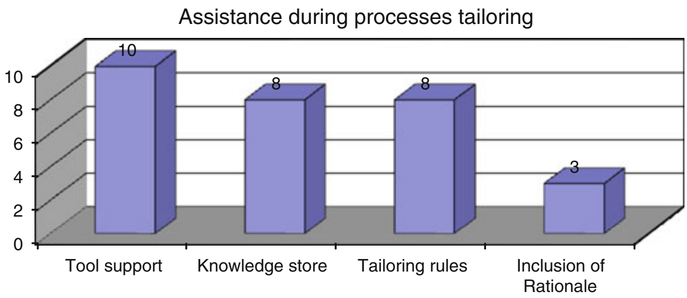


Fig. 11 Assistance during the process tailoring proposed

- Using one point of view for each process element, as (Hausen 1998) proposes, helps process engineers to see the results of their work. This approach also proposes process simulation as another means of verifying process consistency.
- Performing the tailoring first in a bottom-up and then in a top-down manner in order to better meet the process requirements (Jaufman and Münch 2005).
- Making use of process patterns containing all the knowledge about process tailoring and defining a protocol for designing and evolving these patterns (Baldassarre et al. 2002).
- Using brainstorming to tailor a process in an organization (Hanssen et al. 2005).

According to Fig. 10, most of the studies (19 out of 32 studies, 59.38%) propose the use of assistance in order to facilitate tailoring. This means providing techniques to help the process engineer decide how to resolve and decide about each variation in the process. Figure 11 shows the different means provided to assist in process tailoring. The assistance strategy most frequently used is that of tool support (present in 10 out of 32 studies, 31.25%), while knowledge storing and tailoring based on rules are the second most frequently used options, with 8 studies for each (i.e., 25.00%). The former refers to storing information about process tailoring in order for it to be used in tailoring the process again, and the latter signifies that certain defined rules guide the process engineer in carrying out the process tailoring. The inclusion of rationales regarding the variations achieved is only considered in three studies (9.38%).

5.5.1 *Tailoring requirements obtained after answering the SRQ4*

The requirements extracted from the SRQ4 results analysis are presented in Table 7.

6 Limitations

The main threats to the validity of an SLR are publication selection bias, inaccuracy in data extraction, and misclassification (Sjoeberg et al. 2005). We are aware that it is really not possible to analyze all the literature relating to one topic, since we cannot access all the articles. However, we have used five well-known sources, which include the most important conferences and journals in this area; this therefore ensures the completeness of this SLR. In addition, we ensured that all the gray literature we previously knew of was also found in this SLR.

Table 7 Requirements for a process-tailoring notation

Id	Requirement	Description
RQ4.1	Relations between variability and base process	Variant elements in a process modify it and affect the rest of the elements (variant and non-variant) of the process. Variation elements are therefore defined to fit the characteristics of the process, which makes it necessary to establish a relationship between these variation elements and the process they were designed for
RQ4.2	Variability transformation/ process tailoring	Processes must be tailored in order to fit the characteristics of products, projects, and organization Tailored processes can be executed and used in projects, without any variability. All their elements are stable; there are no differences between processes tailored from a process line and processes defined on their own, with no process line
RQ4.3	Default variations	Variations are defined to offer the user a choice with which to tailor the process. However, in some cases, from all the elements in that choice, one of them may be the most suitable for use; in this case, the variable element must be configured as a default variation
RQ4.4	Documentation of elements	For there to be a good use of the variability elements defined, a detailed documentation of these is necessary to specify why they were created, how they behave, and how they tailor the process.
RQ4.5	Rationale of the use of elements	The rationale concerning what is changed to tailor one process, what the new value is, why the change was performed, when it was carried out, what their consequences are … must be elicited This rationale helps users to select correct variations according to the characteristics of the project, process, product, and organization

In order to avoid bias in the selection process, the first author's work was reviewed by the other authors. These authors also reviewed some randomly selected papers to ensure the consistency of the process, along with the correct application of the inclusion and exclusion criteria. Article duplication is another potential threat, which can affect statistics. The selected papers were checked twice in order to detect and remove duplicate papers, including only the last and more complete versions.

Data extraction and analysis were carried out by the main author and checked by the others. All disagreements were resolved through discussion. Owing to the lack of a standard terminology and standards for reporting and presenting studies, some inaccuracies in the data extraction process or misclassification may occur. To mitigate this, a rigorous selection and extraction process was followed based on the guidelines of Kitchenham (2004).

7 Summary and conclusions

In addition to other aspects, high product quality results significantly from high-quality processes. These processes need to be adapted to development contexts. In an effort to discover how processes are adapted, and how their requirements and constructors are used in the literature, this paper presents a systematic review regarding tailoring mechanisms. As a result of this SLR, 32 primary studies were found, which is a relatively low number compared with other areas of software engineering, but it is necessary to bear in mind that tailoring in software processes is a new topic. However, the primary studies found were sufficient to satisfy the aims of this SLR.

The fulfillment of the stated research questions of this SLR was as follows:

7.1 MRQ: Which approaches for tailoring and adaptation software process can be found in the literature?

The results obtained in the SLR show the state-of-the-art in tailoring software processes, from the more theoretical to the more practical tailoring approaches, and as one of its main contributions, it allows the requirements that a process-tailoring notation needs to fulfill to be understood. These results allow certain conclusions to be drawn.

7.2 SRQ1: Which elements of the software process are used when adaptation is considered?

The first conclusion is that the software engineering community has shown an ever-increasing interest in software process tailoring. This interest has been particularly great during the last 3 years. We found that publications increasingly report approaches that have been applied in industrial practice. Of further interest is what has occurred since the product lines paradigm appeared around 2002: those studies employing product-based initiatives now attempt to apply this paradigm to software process tailoring rather than using the fragments approach.

The second important result is that activities are the process element that is most frequently varied, followed by artifacts and roles. Most studies propose varying two or three of these elements when tailoring processes. With respect to variations in relationships between process elements, variations in control and data flows are those most frequently used and grouping of single elements into larger process structures follows them. These variations are used to complement variations in single elements. In half of the studies analyzed, variations in single elements focus first on deciding on the variation of one of the elements making up the process and then concentrate on tailoring the other elements.

7.3 SRQ2: What kind of process-modeling notations are used in existing process models?

Another important result is that operations such as element deletion or insertion, and element (definition) changes are widely used to vary processes throughout the studies. However, they also use other mechanisms such as patterns, parameterization, or inheritance relationships between the elements. After mapping these operations onto the optional, alternative, and mandatory variabilities variant-rich process support based on product lines, the optional and alternative methods are widely employed to tailor software processes.

7.4 SRQ3: Which process-modeling notations are used as a basis for supporting tailoring in process models?

The fourth result is that only a few studies propose a special notation with which to represent variability in the processes, and the development of a variability notation is thus a real need.

7.5 SRQ4: How are processes tailored to meet organizational or project characteristics?

Finally, it is important to highlight that the studies offer information about how to obtain a tailored (specific) process after the tailoring. In order to do this, most of the studies propose the use of tools to assist in tailoring and knowledge storing.

Some approaches show that tailoring must be carried out in several steps, with decisions about certain variations being made in each step. Others propose the use of several abstraction levels to obtain the tailored process. It is vital to recall that process tailoring must ensure consistency between the tailored process and the original standard process model. According to the studies, verification may be carried out directly after tailoring or could take place after each tailoring action.

7.6 Application of the results obtained

However, after carrying out the SLR, we discovered that no notations that include all of these requirements currently exist. To fill this gap, it is necessary to develop a process notation that includes the variability requirements discovered in this SLR. All these requirements and the process line approach (Rombach 2005) have been united, and a notation based on the OMG standard *Software Process Engineering Metamodel* (*SPEM*) has been proposed in Martínez-Ruiz et al. (2008, 2009). This approach, called vSPEM, allows process variability representation and process tailoring through the management of variant-rich processes. The protocol of this systematic review could be used in future iterations to perform secondary searches in other sources in an attempt to search for other new requirements.

Acknowledgments This work is partially supported by the Program FPU of the Spanish Ministerio de Educación and by the PEGASO/MAGO (Ministerio de Ciencia e Innovación and Fondo Europeo de Desarrollo Regional, TIN2009-13718-C02-01), MEDUSAS (Centro para el Desarrollo Tecnológico e Industrial (Ministerio de Ciencia e Innovación), IDI-20090557), ALTAMIRA (Junta de Comunidades de Castilla-la Mancha, Fondo Social Europeo, PII2I09-0106-2463), and INGENIOSO (Junta de Comunidades de Castilla-la Mancha, PEII11-0025-9533) projects. We would also like to thank Sonnhild Namingha from Fraunhofer IESE for reviewing the first version of this paper.

Appendix 1: Requirements summarization and classification

This appendix summarizes all the requirements for building a variability notation that we have found in the systematic literature review. These could be divided into five categories, according to the aspect of the notation. Each one of these categories represents a sub-goal that needs to be achieved in order to define the variability notation. A more exhaustive description of these categories and the requisites they include is provided in the following sections.

1. Variation support for process model elements.
2. Types of supported variation (optional, mandatory…)
3. Consistency assurance among variants by means of restriction models.
4. Usability support to the variation mechanisms.
5. Documentation and reuse of knowledge.

Requirements for process model element variation support

According to Fuggetta (2000), software process models are composed of several elements, and in order to model variations in software processes, the first set of requirements is to define which of their elements acquire the capability to vary. Table 8 lists the requirements associated with the software process elements.

Requirements for types of supported variation

Once the elements that need to vary in a software process are determined, it is necessary to determine how they could vary. In other words, with regard to the software process, we need to know that the behavior of these elements when they vary to tailor the process, meets the specific characteristics of the project, the product, or the organization. Table 9 summarizes the requirements for modeling this behavior.

Consistency assurance among variants by means of restriction models

The behavior of the elements that vary in a process model must be controlled in order to prevent inconsistencies between the tailored processes. This being so, another set of requirements fits this need for consistency assurance in several ways, by relating elements, and, based on this, by modeling constraints. Table 10 includes all the requirements associated with this category.

Table 8 List of requirements for the goal of defining concepts to model variability parts in process models

Id	Requirement
Q1.1	Variability in activities/tasks
Q1.2	Variability in artifacts
Q1.3	Variability resources
Q1.4	Variability in control flow
Q1.5	Variability in product flow

Table 9 List of requirements for defining rules to model the way variability occurs in process models

Id	Requirement
Q2.1	Optional variation
Q2.2	Alternative variation in a point
Q2.3	Alternative points of a variation
Q2.4	Mandatory variation elements
Q2.5	Mandatory variation places
Q2.6	Evolution of variants

Table 10 List of requirements for defining rules to model the way variability occurs in process models

Id	Requirement
Q2.7	Constraints
Q2.8	Dependences
Q2.9	Variations across contained elements
Q1.6	Variations realized because other ones
Q2.10	Variability propagation

Requirements for usability support to the variation mechanisms

Process variability is based on the idea of someone deciding all the variabilities of a variant-rich process model and generating a tailor process model that fits a specific requirement. In order to do this, another set of requirements of a process notation must be designed in an effort to make the tailoring task easier. Table 11 includes all the requirements for supporting easy variant-rich process tailoring.

Requirements for documentation and reuse of knowledge

Tailoring also involves converting characteristics in the context of the process into correct variations; in the variant-rich process, it is important not to start from scratch. Documenting process variations allows reuse of knowledge, which makes deciding process variability an important point. The requirements fitting this category are listed in Table 12.

Appendix 2: Complete list of articles

Table 13 includes all the studies analyzed in this systematic review. It also identifies the type of study according to the classification presented in the "Inclusion and Exclusion criteria" section.

Appendix 3: Excerpts of tables for extraction of information

This appendix presents extracts from the information tables used to organize information during the systematic review. Figures 12 and 13 present an overview of the description and the first information tables, respectively. The other information tables are quite similar to that shown in Fig. 13.

Table 11 List of requirements for defining concepts to abstract the user from variability details

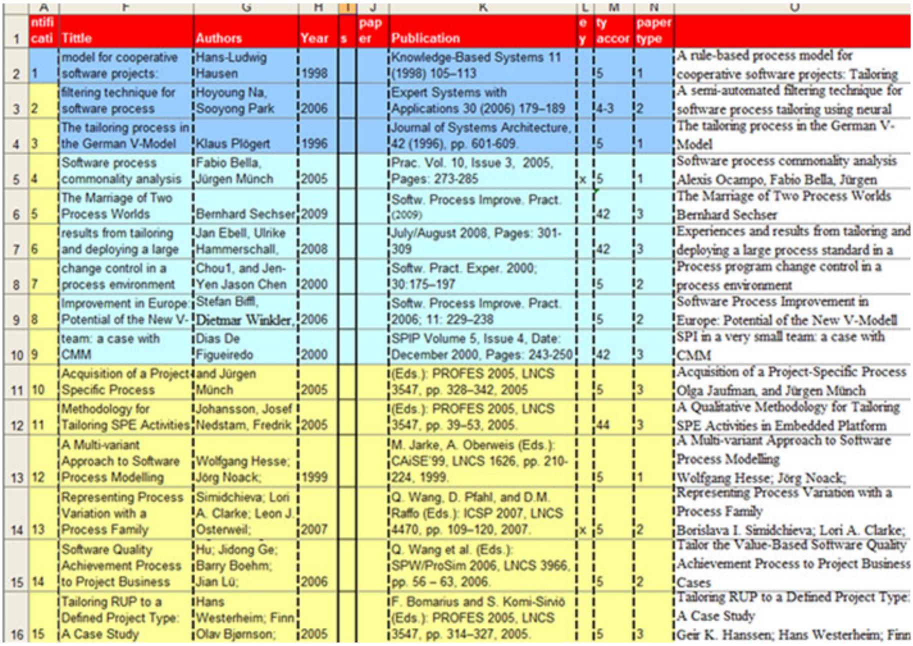
Id	Requirement
Q1.7	Extensive variations
Q1.8	Detailed variations
Q4.1	Relations between variability and base process
Q4.2	Variability transformation/process tailoring
Q4.3	Default variations
Q3.2	Variability standardization

Table 12 List of requirements for defining concepts to allow variability documentation

Id	Requirement
Q4.4	Documentation of elements
Q4.5	Rationale of the use of elements
Q3.1	Differentiate between common and variable parts

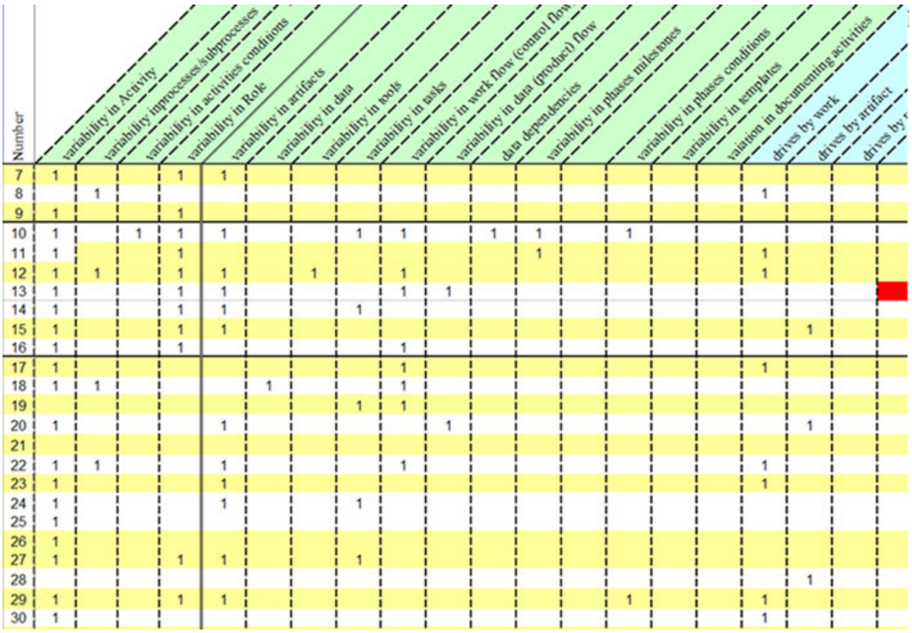
Table 13 Complete list of primary studies

No.	Title	Type
1	A rule-based process model for cooperative software projects: Tailoring and testing a software process to be used on the Web (Hausen 1998)	1
2	A semi-automated filtering technique for software process tailoring using neural network (Park et al. 2006)	2
3	The tailoring process in the German V-Model (Plögert 1996)	1
4	Software process commonality analysis (Ocampo et al. 2005)	1
5	Software Process Improvement in Europe: Potential of the New V-Modell XT and Research Issues (Biffl et al. 2006)	2
6	Experiences and results from tailoring and deploying a large process standard in a company (Armbrust et al. 2008)	3
7	Process program change control in a process environment (Chou and Chen 2000)	2
8	The Marriage of Two Process Worlds (Sechser 2009)	3
9	SPI in a very small team: a case with CMM (Batista 2000)	3
10	Towards a SPEM v2.0 Extension to Define Process Lines Variability Mechanisms (Martínez-Ruiz et al. 2008)	2
11	A Multi-variant Approach to Software Process Modelling (Hesse and Noack 1999)	1
12	Representing Process Variation with a Process Family (Simidchieva et al. 2007)	2
13	A Qualitative Methodology for Tailoring SPE Activities in Embedded Platform Development (Johansson et al. 2005)	3
14	Acquisition of a Project-Specific Process (Jaufman and Münch 2005)	3
15	Tailor the Value-Based Software Quality Achievement Process to Project Business Cases (Huang et al. 2006)	2
16	Tailoring RUP to a Defined Project Type: A Case Study (Hanssen et al. 2005)	3
17	Managing Process Customizability and Customization: Model, Language and Process (Lazovik and Ludwig 2007)	2
18	A Practical Approach for Process Family Engineering of Embedded Control Software (Giese et al. 2007)	2
19	Variability Mechanism Centric Process Family Architectures (Schnieders 2006)	2
20	PDP: Programming A Programmable Design Process (Sutton and Osterweil 1996)	1
21	A Method for Assembling a Project-Specific Software Process Definition (Lobsitz 1996)	1
22	Prism-Methodology and Process-Oriented Environment (Madhavji and Schafer 1991)	1
23	SPiCE in Action—Experiences in Tailoring and Extension (Cass et al. 2002)s	3
24	Tailoring and Verifying Software Process (Yoon et al. 2001)	2
25	Tailoring ISO/IEC 12207 for Instructional Software Development (Demirors et al. 2000)	3
26	Tailoring Software Evolution Process (Dai and Li 2007)	1
27	Tailoring test process by using the component-based development paradigm and the XML technology (Seo and Choi 2000)	2
28	Using Process Tailoring to Manage Software Development Challenges (Xu and Ramesh 2008)	3
29	Flexible Software Development: From Software Architecture to Process (Balasubramaniam et al. 2007)	2
30	A Procedure for Customizing a Software Process (Ibargüengoitia et al. 2006)	1
31	ProMisE: A Framework for Process Models Customization to the Operative Context (Baldassarre et al. 2002)	2
32	Techniques for Process Model Evolution in EPOS (Jaccheri and Conradi 1993)	1



	A	F	G	H	I	J	K	L	M	N	O
1	ntifi cati	Tittle	Authors	Year	s	pap er	Publication	e y	ty accor	paper type	
2	1	model for cooperative software projects:	Hans-Ludwig Hausen	1998			Knowledge-Based Systems 11 (1998) 105–113		5	1	A rule-based process model for cooperative software projects: Tailoring
3	2	filtering technique for software process	Hoyoung Na, Sooyong Park	2006			Expert Systems with Applications 30 (2006) 179–189		4-3	2	A semi-automated filtering technique for software process tailoring using neural
4	3	The tailoring process in the German V-Model	Klaus Plögert	1996			Journal of Systems Architecture, 42 (1996), pp. 601-609.		5	1	The tailoring process in the German V-Model
5	4	Software process commonality analysis	Fabio Bella, Jürgen Münch	2005			Prac. Vol. 10, Issue 3, 2005, Pages: 273-285	x	5	1	Software process commonality analysis Alexis Ocampo, Fabio Bella, Jürgen
6	5	The Marriage of Two Process Worlds	Bernhard Sechser	2009			Softw. Process Improve. Pract. (2009)		42	3	The Marriage of Two Process Worlds Bernhard Sechser
7	6	results from tailoring and deploying a large	Jan Ebell, Ulrike Hammerschall,	2008			July/August 2008, Pages: 301-309		42	3	Experiences and results from tailoring and deploying a large process standard in a
8	7	change control in a process environment	Chou1, and Jen-Yen Jason Chen	2000			Softw. Pract. Exper. 2000; 30:175–197		5	2	Process program change control in a process environment
9	8	Improvement in Europe: Potential of the New V-	Stefan Biffl, Dietmar Winkler,	2006			Softw. Process Improve. Pract. 2006; 11: 229–238		5	2	Software Process Improvement in Europe: Potential of the New V-Modell
10	9	team: a case with CMM	Dias De Figueiredo	2000			SPIP Volume 5, Issue 4, Date: December 2000, Pages: 243-250		42	3	SPI in a very small team: a case with CMM
11	10	Acquisition of a Project-Specific Process	and Jürgen Münch	2005			(Eds.): PROFES 2005, LNCS 3547, pp. 328–342, 2005.		5	3	Acquisition of a Project-Specific Process Olga Jaufman, and Jürgen Münch
12	11	Methodology for Tailoring SPE Activities	Johansson, Josef Nedstam, Fredrik	2005			(Eds.): PROFES 2005, LNCS 3547, pp. 39–53, 2005.		44	3	A Qualitative Methodology for Tailoring SPE Activities in Embedded Platform
13	12	A Multi-variant Approach to Software Process Modelling	Wolfgang Hesse; Jörg Noack;	1999			M. Jarke, A. Oberweis (Eds.): CAiSE'99, LNCS 1626, pp. 210-224, 1999.		5	1	A Multi-variant Approach to Software Process Modelling Wolfgang Hesse; Jörg Noack;
14	13	Representing Process Variation with a Process Family	Simidchieva; Lori A. Clarke; Leon J. Osterweil;	2007			Q. Wang, D. Pfahl, and D.M. Raffo (Eds.): ICSP 2007, LNCS 4470, pp. 109–120, 2007.	x	5	2	Representing Process Variation with a Process Family Borislava I. Simidchieva; Lori A. Clarke;
15	14	Software Quality Achievement Process to Project Business	Hu; Jidong Ge; Barry Boehm; Jian Lü;	2006			Q. Wang et al. (Eds.): SPW/ProSim 2006, LNCS 3966, pp. 56 – 63, 2006.		5	2	Tailor the Value-Based Software Quality Achievement Process to Project Business Cases
16	15	Tailoring RUP to a Defined Project Type: A Case Study	Hans Westerheim; Finn Olav Bjørnson;	2005			F. Bomarius and S. Komi-Sirviö (Eds.): PROFES 2005, LNCS 3547, pp. 314–327, 2005.		5	3	Tailoring RUP to a Defined Project Type: A Case Study Geir K. Hanssen; Hans Westerheim; Finn

Fig. 12 Overview of the description table



Number	variability in Activity	variability inprocesses/subprocesses	variability in activities conditions	variability in Role	variability in artifacts	variability in data	variability in tools	variability in tasks	variability in work flow (control flow)	variability in data (product) flow	data dependencies	variability in phases milestones		variability in phases conditions	variability in templates	vaiation in documenting activities	drives by work	drives by artifact	drives by r
7	1			1	1														
8		1															1		
9	1			1															
10	1		1	1	1			1	1		1	1		1					
11	1			1								1					1		
12	1	1		1	1		1		1								1		
13	1			1	1				1	1									
14	1			1	1			1											
15	1			1	1													1	
16	1			1					1										
17	1								1								1		
18	1	1				1			1										
19								1	1										
20	1				1					1								1	
21																			
22	1	1			1				1								1		
23	1				1												1		
24	1				1			1											
25	1																		
26	1																		
27	1			1	1			1											
28																		1	
29	1			1	1									1			1		
30	1																1		

Fig. 13 Overview of the first information table

References

Armbrust, O., Ebell, J., Hammerschall, U., Münch, J., & Thoma, D. (2008). Experiences and results from tailoring and deploying a large process standard in a company. *Software Process Improvement and Practice, 13*, 301–309. doi:10.1002/spip.

Balasubramaniam, D., Morrison, R., Greenwood, R. M., & Warboys, B. (2007). Flexible software development: From software architecture to process. In *Proceedings of software architecture, 2007. Wicsa '07*. IEEE Computer Science, 14 pp.

Baldassarre, M. T., Caivano, D., Visaggio, C. A., & Visaggio, G. (2002). Promise: A framework for process models customization to the operative context. In *International symposium on empirical software engineering (ISESE '02)*, pp. 103–110.

Batista, J. (2000). SPI in a very small team: A case with CMM. *Software Process Improvement and Practice, 5*(4), 243–250.

Biffl, S., Winkler, D., Höhn, R., & Wetzel, H. (2006). Software process improvement in Europe: Potential of the new v-modell xt and research issues. *Software Process Improvement and Practice, 11*, 229–238.

Biolchini, J., Gomes, P., Cruz, A., & Travassos, G. (2005). *Systematic review in software engineering*. Rio de Janeiro, Brazil: Systems Engineering and Computer Science Department, UFRJ.

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development method and tools. *Information and Software Technology, 38*(4), 275–280.

Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. A. (2008). Using mapping studies in software engineering. In *PPIG 2008*, Lancaster, UK, pp. 195–204.

Cass, A., Volcker, C., Sutter, P., Dorling, A., & Stienen, H. (2002). Spice in action—Experiences in tailoring and extension. In *Euromicro*, pp. 352–360.

Chou, S.-C., & Chen, J.-Y. J. (2000). Process program change control in a process environment. *Software: Practice Experience, 30*, 175–197.

Chrissis, M. B., Konrad, M., & Shrum, S. (2006). *CMMI: Guidelines for process integration and product improvement* (Vol. 1, The SEI series in software engineering). Boston: Pearson.

Clements, P., & Northrop, L. (2002). *Software product lines. Practices and patterns*. Boston: Addison-Wesley.

Dai, F., & Li, T. (2007). Tailoring software evolution process. In *Eighth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing*, Qingdao, pp. 782–787.

Demirors, O., Demirors, E., Tarhan, A., & Yildiz, A. (2000). Tailoring ISO/IEC 12207 for instructional software development. In *Proceedings of the 26th Euromicro conference*, pp. 2300-2307.

Fuggetta, A. (2000). Software process: A roadmap. In A. Finkelstein (Ed.), *The future of software engineering* (pp. 25–34). Limerick, Ireland: ACM.

Giese, C., Schnieders, A., & Weiland, J. (2007). A practical approach for process family engineering of embedded control software. In *Proceedings of the 14th annual IEEE international conference and workshops on the engineering of computer-based systems (ECBS '07)*, pp. 229–239.

Hanssen, G. K., Westerheim, H., & Bjørnson, F. O. (2005). Tailoring RUP to a defined project type: A case study. In F. Bomarius, & S. Komi-Sirviö (Eds.), *PROFES* (Vol. LNCS 3547, pp. 314–327). New York: Springer.

Hausen, H.-L. (1998). A rule-based process model for cooperative software projects: Tailoring and testing a software process to be used on the web. *Knowledge-Based Systems, 11*, 105–113.

Hesse, W., & Noack, J. (1999). A multi-variant approach to software process modelling. In M. Jarke, & A. Oberweis (Eds.), *CAiSE '99, 1999* (Vol. LNCS 1626, pp. 210–224). New York: Springer.

Huang, L., Hu, H., Ge, J., Boehm, B., & Lü, J. (2006). Tailor the value-based software quality achievement process to project business cases. In *Software process change* (Vol. LNCS 3966, pp. 56–63). Heidelberg: Springer.

Ibargüengoitia, G., Salazar, J. A., Sánchez, M. G., & Ramírez, A. Y. (2006). A procedure for customizing a software process. In *Proceeding of ENC '03 proceedings of the 4th Mexican international conference on computer science* (pp. 68–72). IEEE Computer Society.

ISO. (2002). ISO/IEC 12207:2002. Information technology—Software life cycle processes. Geneva: International Organization for Standardization.

ISO. (2004). ISO/IEC 15504-2:2003/cor.1:2004(e). Information technology—Process assessment—Part 2: Performing an assessment. Geneva: International Organization for Standardization.

ISO. (2006). ISO/IEC 15504-5:2006(e). Information technology—Process assessment—Part 5: An exemplar process assessment model. Geneva: International Organization for Standardization.

ISO. (2008). ISO/IEC 12207:2008. Systems and software engineering—Software life cycle processes. Geneva: International Organization for Standardization.

Jaccheri, M. L., & Conradi, R. (1993). Techniques for process model evolution in EPOS. *IEEE Transactions on Software Engineering, 19*(2), 1145–1156.

Jaufman, O., & Münch, J. (2005). Acquisition of a project-specific process. In F. Bomarius, & S. Komi-Sirviö (Eds.), *PROFES 2005* (Vol. LNCS 3547, pp. 328–342). New York: Springer.

Johansson, E., Nedstam, J., Wartenberg, F., & Höst, M. (2005). A qualitative methodology for tailoring spe activities in embedded platform. In *Product focused software process improvement* (Vol. LNCS 3547, pp. 39–53). Heidelberg: Springer.

Kitchenham, B. (2004). *Procedures for performing systematic reviews* (pp. 1–28). Australia: Keele University and ICT Australia Ltd.

Kitchenham, B., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., et al. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering, 28*(8), 721–734.

Krutchen, P. (2000). *The rational unified process: An introduction* (2nd ed.). New York: Addison Wesley.

Lazovik, A., & Ludwig, H. (2007). Managing process customizability and customization: Model, language and process. In B. Benatallah et al. (Ed.), *Wise 2007*, Vol. LNCS 4831, pp. 373–384.

Lobsitz, R. M. (1996). A method for assembling a project-specific software process definition. In *Proceedings of the 1996 Hawaii international conference on system sciences (HICSS-29).*

Madhavji, N. H., & Schafer, W. (1991). Prism-methodology and process-oriented environment. *IEEE Transactions on Software Engineering, 17*(12), 1270–1283.

Martínez-Ruiz, T., García, F., & Piattini, M. (2008). Towards a SPEM v2.0 extension to define process lines variability mechanisms. In R. Lee (Ed.), *SERA* (Vol. SCI 150, pp. 115–130). Praga: Springer.

Martínez-Ruiz, T., García, F., & Piattini, M. (2009). Enhanced variability mechanisms to manage software process lines. In *EUROSPI 2009* (pp. 12.13–12.23). Alcalá de Henares, Madrid: Publizon.

Ocampo, A., Bella, F., & Münch, J. (2005). Software process commonality analysis. *Software Process – Improvement and Practice, 10*(3), 273–285.

Osterweil, L. (1987). *Software processes are software too*. Paper presented at the 9th ICSE, Monterey, CA, March 1987.

Park, S., Na, H., Park, S., & Sugumaran, V. (2006). A semi-automated filtering technique for software process tailoring using neural network. *Expert Systems with Applications, 30*, 179–189.

Pedreira, Ó., Piattini, M., Luaces, M. R., & Brisaboa, N. (2007). A systematic review of software process tailoring. *ACM SIGSOFT Software Engineering Notes, 32*(3), 1–6.

Pino, F., García, F., & Piattini, M. (2008). Software process improvement in small and medium software enterprises: A systematic review. *Software Quality Journal, 16*(2), 237–261.

Plögert, K. (1996). The tailoring process in the german v-model. *Journal of Systems Architecture, 42*, 601–609.

Rombach, D. (2005). Integrated software process and product lines. In M. Li, B. Boehm, & L. Osterweil (Eds.), *ISPW* (Vol. LNCS 3840, pp. 83–90). Beijing, China: Springer.

Sandelowski, M., Barroso, J., & Voils, C. I. (2007). Using quantitative metasummary to synthesize qualitative and quantitative descriptive findings. *Research in Nursing and Health, 30*(1), 99–111.

Schnieders, A. (2006). Variability mechanism centric process family architectures. In I. C. Society (Ed.), *ECBS '06: Proceedings of the 13th annual IEEE international symposium and workshop on engineering of computer based systems, 2006*, pp. 289–298. doi:10.1109/ECBS.2006.72.

Sechser, B. (2009). The marriage of two process worlds. *Software Process Improvement and Practice, 14*(2), 349–354.

SEI. (2001). Standard CMMI® appraisal method for process improvement (SCAMPI), version 1.1: Method definition document (CMU/SEI-2001-HB-001). Pittsburgh: Software Engineering Institute (SEI).

SEI. (2004). Capability maturity model integration (CMMI), version 1.1 CMMI (CMMI-SE/SW/IPPD/SS, v1.1) staged representation. Pittsburgh: Software Engineering Institute (SEI).

Seo, J., & Choi, B. (2000). Tailoring test process by using the component-based development paradigm and the xml technology. In *Proceeding APSEC '00 proceedings of the seventh Asia-Pacific software engineering* (pp. 356–363). IEEE Computer Society.

Sheard, S. A., & Lake, J. (1998). Systems engineering standards and models compared. In *Proceedings of the 8th international symposium on systems engineering (INCOSE)*, pp. 589–605.

Simidchieva, B. I., Clarke, L. A., & Osterweil, L. (2007). Representing process variation with a process family. In Q. Wang, D. Pfahl, & D. M. Raffo (Eds.), *ICSP 2007, Minneapolis, USA, 2007* (Vol. LNCS 4470, pp. 121–133). New York: Springer.

Sjoeberg, D. I. K., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N.-K., et al. (2005). A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering, 31*(9), 733–753.

Sutton, S., & Osterweil, L. J. (1996). PDP: Programming a programmable design process. In *8th International workshop on software specification and design, 1996*, pp. 186–190.
Szyperski, C., Bosch, J., & Weck, W. (1998). Component-oriented programming. In A. Moreira, & S. Demeyer (Eds.), *Object-oriented technology ECOOP '99 workshop reader* (Vol. LNCS 1743, pp. 184–192). Berlin: Springer.
Xu, P., & Ramesh, B. (2008). Using process tailoring to manage software development challenges. *IT Professional, 10*(4), 39–45.
Yoon, I.-C., Min, S.-Y., & Bae, D.-H. (2001). *Tailoring and verifying software process*. Paper presented at the 8th APSEC, Macao, China.

Author Biographies

Tomás Martínez-Ruiz holds the MSc degree in Computer Science from the University of Castilla-La Mancha 2007, and he is currently a PhD student in Computer Science. He Works in Alarcos Research Group at the University of Castilla-La Mancha. His research interests include process improvement, variant-rich software processes, and aspect-oriented software engineering.

Dr. Jürgen Münch is a Professor in the Department of Computer Science at the University of Helsinki. His research in software and systems engineering centers on measurement and quantitative analysis of software processes and systems, cloud-based software engineering, software process modeling and management, global software development, and empirical software engineering. He is the author of more than 110 refereed publications and co-inventor of GQM + Strategies, a method for aligning company's software strategies with long-term business goals. Dr. Münch has been a principal investigator of numerous research and industrial development projects, including the definition of international quality and process standards. He has been awarded the IFIP TC2 Manfred Paul Award for Excellence in Software Theory and Practice, four best paper awards, the technology innovation award from the Lotto foundation Rhineland-Palatinate, and a PhD thesis award from the Fraunhofer-Gesellschaft. He is vice-chairman of the German Association for Software Metrics and Cost Estimation (DASMA).

Félix García received his M.S. (2001) and Ph.D. (2004) degrees in computer science from the University of Castilla-La Mancha (UCLM). He is currently an associate professor in the Department of Information Technologies and Systems at the UCLM. He is member of the Alarcos Research Group, and his research interests include business process management, software processes, software measurement, and agile methods.

Mario Piattini is a full professor at the UCLM. He holds the PhD degree in Computer Science from the Technical University of Madrid and leads the Alarcos Research Group. He is CISA, CISM, CGEIT, and CRISC by ISACA. His research interests include software quality, metrics and maintenance. He is the Director of the Joint UCLM-Indra Software Research and Development Center and the Institute of Information Systems and Technologies.