

REVIEW

Open Access

# Templates for textual use cases of software product lines: results from a systematic mapping study and a controlled experiment

Ismayle S Santos<sup>1\*</sup>, Rossana MC Andrade<sup>1</sup> and Pedro A Santos Neto<sup>2</sup>

\*Correspondence:

ismaylesantos@great.ufc.br

<sup>1</sup> Federal University of Ceara,  
Department of Computer Science,  
Fortaleza-CE, Brazil

Full list of author information is  
available at the end of the article

## Abstract

Use case templates can be used to describe functional requirements of a Software Product Line. However, to the best of our knowledge, no efforts have been made to collect and summarize these existing templates and no empirical evaluation of the use cases' comprehensibility provided by these templates has been addressed yet. The contributions of this paper are twofold. First, we present a systematic mapping study about the SPL variability description using textual use cases. From this mapping, we found twelve SPL use case templates and observed the need not only for the application of these templates in real SPL but also for supporting tools. Secondly, this work presents an evaluation of the comprehensibility of SPL use cases specified in these templates through a controlled experiment with 48 volunteers. The results of this experiment show that the specification of variabilities in the steps' numeric identifiers of the textual use cases is better to the use case understanding than the other approaches identified. We also found evidence that the specification of variabilities at the end of the use cases favors the comprehension of them and the use of questions associated to the variation points in the use cases improves the understanding of use cases. We conclude that each characteristic of the existing templates has an impact on the SPL use case understanding and this should be taken into account when choosing one.

**Keywords:** Use case; Systematic mapping study; Software product line; Controlled experiment

## 1 Introduction

The paradigm of Software Product Line (SPL) has emerged together with large-scale systematic reuse. According to Northrop (2002), an SPL is “a set of software-intensive systems that shares a common managed feature set, satisfying a particular market segment's specific needs or mission and that is developed from a common set of core assets in a prescribed way”.

Because the products of an SPL reuse common artifacts among themselves, the SPL approach maximizes the benefits of systematic and planned reuse. Some benefits of using SPL are (Gallina and Guelfi 2007; Northrop and Clements 2007; Urli et al. 2014): i) improved productivity; ii) better time to market; and iii) higher product quality.

In the SPL paradigm, the requirements engineering activity needs to cope with common and variable requirements for the whole set of products in the family

(Oliveira et al. 2013). The SPL requirements should then be modeled from the reuse perspective by explicitly representing the commonality and variability information (Alves et al. 2010).

In this scenario, one of the requirements artifacts most used in SPL development are use cases (Alves et al. 2010). In the SPL paradigm is important to note that the use cases model needs to be adapted to incorporate the mechanisms of variability (Oliveira et al. 2013). Thus, there are proposals for the description of SPL variability in the use cases model with 'include' and 'extend' relationships (Azevedo et al. 2012; Bragança and Machado 2006; Gomaa 2004).

However, it is not enough to only manage variability among whole use cases; it must also be possible to specify variant behavior within use case descriptions (Erikssona et al. 2004). This variability can be specified through variation points, and optional and alternative use cases (Fant et al. 2013). Therefore, the template used for textual use cases in the SPL paradigm should allow the specification of "small variations" (Gomaa 2004) (fine-grained variation), which can affect just one or two lines in the use case description.

In the literature, there have been different proposals for templates for textual use case descriptions of Software Product Lines, such as (Gallina and Guelfi 2007; Gomaa 2004; Oliveira et al. 2013). Other studies have compared some of these templates (Bonifácio and Borba 2009; Santos et al. 2013). However, to the best of our knowledge, there has not been a systematic effort to collect and summarize the existing templates for textual use case descriptions in the SPL paradigm and there is no empirical assessment of the SPL use cases' comprehensibility provided by these use case templates.

In a previous work (Santos et al. 2014), we identified and classified nine templates for textual use case descriptions of Software Product Lines through a Systematic Mapping (SM) (Kitchenham and Charters 2007). This new paper extends on that work by:

- Updating the systematic mapping in order to consider papers published in 2014;
- Including papers with an aspect-oriented approach for SPL use cases modelling;
- Presenting a controlled experiment, following the guide of Wohlin et al. (2000), conducted to evaluate what kind of template better favors the comprehension of SPL variabilities specified in textual use case descriptions.

By conducting an SM that involves a controlled and formal literature search, we believe that the results of this paper will benefit researchers and practitioners. Researchers will benefit because the SM results indicate existing research gaps that need further investigation. Practitioners will benefit because the results can be used as a reference for choosing an existing template, or in proposing a new use case template for Software Product Lines. The results of the controlled experiment provide an empirical knowledge about the template structures identified, and this knowledge can help in the choice of a use case template for practical use, as well as in the execution of other controlled experiments with use cases.

This paper is divided into six sections. In Section 2, we introduce the background. In Section 3, we discuss the related work. The methodology adopted to conduct the systematic mapping and results are presented in Section 4. In Section 5, we describe the controlled experiment conducted with the SPL use case templates. Finally, in Section 6, we provide conclusions from this work.

## 2 Background

In the SPL paradigm the concepts of feature and feature model are essential. A feature is an attribute, quality or aspect visible to the user (Kang et al. 1990). According to the approach of Kang et al. (1990), the features can be “mandatory”, “optional” or “alternative”. Mandatory features are those available on all systems built within the family. Optional features are those features that may or may not be included in the products. Alternative features represent a selection “exactly-one-of-many” made from a set of features.

A feature model represents the information of all possible products of an SPL in terms of features and the relationships among them (Benavides et al. 2010). Several feature model languages have been proposed since the introduction of Feature-Oriented Domain Analysis (FODA) by Kang et al. (1990). According to Benavides et al. (2010) the feature models can be grouped into three categories: (i) Basic feature models, offering mandatory, optional, alternative and ‘or’ features, as well as constraints between features like ‘requires’ and ‘excludes’; (ii) Cardinality-based feature models, offering, in addition, UML-like multiplicities for feature selection; and (iii) Extended feature models, where additional information is included in terms of feature attributes.

Thus, in an SPL, the requirements define applications and their features. The requirements engineering process of an SPL should include strategic and effective techniques for analyzing domains, finding similarities and variabilities, and dealing with a community of stakeholders probably larger than those for single-system requirements elicitation (Cheng and Atlee 2007; Northrop and Clements 2007).

The process of SPL development also influences the engineering requirements. In this process, unlike the development process of traditional applications, there are three essential activities (Northrop and Clements 2007): core assets development, product development, and management. Core assets development, also known as Domain Engineering, aims to define commonalities and variability and to develop the artifacts for reuse. Product development, also known as Application Engineering, includes the development of final products with reuse. Finally, management is related directly to the control of the processes and activities, in order to allow the previous two activities to work together.

Requirements engineering plays a key role in both domain engineering and application engineering. In domain engineering, the requirements of the domain must be defined as common and variable requirements. In application engineering, the requirements for individual products of the SPL are defined by reusing the domain requirements.

In this scenario, textual requirements and use cases are often used for describing SPL functional requirements (Alves et al. 2010). For this, works in literature (e.g. (Niu and Easterbrook 2008; Oliveira et al. 2014)) adapted these artifacts in order to incorporate the SPL variability. Finally, with regards to the textual description of use cases, the use case template from Cockburn (2000) has inspired the creation of SPL use case templates (e.g. the templates of Bertolino et al. (2003) and Gallina et al. (2007)), with specific elements to deal with the variabilities and commonalities of an SPL.

## 3 Related work

In a search for secondary studies on variability modelling and requirements engineering, we found two Systematic Literature Reviews (SLR) (Kitchenham and Charters 2007)

on Variability (Chen et al. 2009; Galster et al. 2014) and two SLRs on requirements engineering within SPL Engineering (Alves et al. 2010; Neiva 2009).

The SLR of Galster et al. (2014) summarized existing research related to variability handling in software systems and proposed the classification for variability in different dimensions. Chen et al. (2009) identified approaches for the Variability Management in SPL and classified them with respect to historical background, the issues that motivated their creation, variability models used, and their support for the different SPL phases. Our Systematic Mapping also addresses the variability modelling, but with focus on the specification of the variability in the user's perspective through the textual use cases.

Neiva (2009) and Alves et al. (2010) presented a Systematic Review of Requirements Engineering (RE) within Software Product Line Engineering (SPLE). Neiva (2009) investigated which RE activities are adopted, which models and techniques are used, and how the approaches for RE in SPL deal with variability. In the same scenario, Alves et al. (2010) classified different SPL requirements engineering approaches in terms of tool support, RE activities, and adoption strategies. The results of both these SLRs show that use cases is one of the most used artifacts for describing SPL functional requirements, but they do not explore the variability description in such artifacts. On the contrary, this systematic mapping focuses on the variability description within textual use cases, highlighting the differences among the existing SPL use case templates.

We also looked for work that compares requirements engineering approaches for SPL, and, addressing this issue, we found four studies (Alferez et al. 2014; Asadi et al. 2012; Blanes and Insfrán 2012; Kuloor and Eberlein 2002). The difference among these studies is in the applied evaluation criteria and in the type of approach selected.

Blanes and Insfran (2012) analyzed requirement engineering approaches that use Model-Driven Development (MDD) techniques for SPL development. For this comparison, they used five criteria: SPL activity support, RE covered tasks, MDD strategy support, the degree of automatic support with a given tool, and the type of validation of the proposals. Asadi et al. (2012) presented feature-oriented requirements engineering methods for an SPL and assessed them based on an evaluation framework with evaluation criteria related to requirements engineering principles and processes, variability and commonality analysis, and tooling support. Kuloor and Eberlein (2002) described and compared requirements engineering techniques used in existing SPL practices. This comparison was made based on the SPL requirements modeling, specification, verification and management. Alferez et al. (2014) presented a metric-based comparative study of existing scenario-based SPL requirements approaches in order to understand how they address modularity, stability and expressiveness. The main contribution of our work, compared to the previous four studies, is in the assessment of the comprehensibility issues provided by the use case templates and the focus on an artifact (textual use cases) instead of the requirements engineering approach.

We also identified some studies addressing the assessment of the understanding of the requirements provided by the use cases in software development (Dahan et al. 2014; Hadar et al. 2010; Jeyaraj and Sauter 2007; Mustafa 2010; Reinhartz-Berger and Sturm 2014). All of them assess the comprehensibility provided by a use cases model.

Hadar et al. (2010) presented an empirical study that compares the understanding of the requirements model expressed in two requirements modeling methods (Use Cases

and Tropos). The goal of this study was also to estimate the time required to perform simple analysis tasks using both methods. Dahan et al. (2014) conducted a controlled experiment that compared the comprehension of two models of a system and the quality of models created for a certain system with both OO-DFD and the Use Case. Jeyayaj and Sauter (2007) also compared these two modeling methods, but only from the point of view of comprehension of diagrams. Mustafa 2010 presented an experiment that has evaluated the impact of the use cases format on the understanding of the requirements. This work has used subjects with different knowledge in use cases technique in order to investigate whether subjects' experiences play any role in the comprehension of use cases models. Tiwari and Gupta (2013) conducted a controlled experiment to assess the usefulness of eight use case templates against a set of five judging criteria, namely completeness, consistency, understandability, redundancy, and fault proneness. In spite of being interesting empirical studies, none of them dealt with the SPL paradigm, addressed by the experiment described in this paper.

Finally, Reinhartz-Berger and Sturm (2014) presented a controlled experiment where they examined the comprehensibility of domain models specified in a UML-based SPL Engineering method. In this experiment the volunteers were required to answer comprehension questions regarding a domain model specified in use cases, class, and sequence diagrams. This is the only work found that, like the experiment described in this paper, dealt with the comprehensibility of a use cases model in the SPL paradigm. However, in contrast to their work, we do not focus on use cases diagrams. This work focuses on the assessment of different textual use case templates and their impact on the comprehension of the SPL use cases.

## 4 Review

We report the systematic mapping study in this section. First, we describe the methodology adopted, presenting the details of each phase. Then, we present the results and discussion.

### 4.1 Methodology

The Systematic Mapping study is a type of secondary study that can complement Systematic Literature Reviews. While a Systematic Review (SR) is a means of identifying, evaluating and interpreting all available research relevant to a particular question (Kitchenham and Charters 2007), a Systematic Mapping intends to provide an overview of a research area (Petersen et al. 2008).

Other differences between the SR and the SM are (Kitchenham and Charters 2007; Neto et al. 2011; Petersen et al. 2008): i) Mapping studies generally have broader research questions driving them and often ask multiple research questions; ii) The search terms for mapping studies will be less highly focused than for systematic reviews and are likely to return a very large number of studies; and iii) SR aims at establishing the state of evidence while the main focus of an SM is on classification and identification of publication fora. Then, in spite of the fact that this study has a specific focus (textual use case templates) we chose the Systematic Mapping approach because the main goal of this study is to identify and classify the different SPL use case templates.

We used the process defined by Neto et al. (2011), which is divided into three main phases: Research Directives, Data Collection, and Results. In the first phase, Research

Directives, the protocol and the research questions are established. The protocol is a plan that describes the conduct of a proposed SM study (Kitchenham and Charters 2007). The second phase, Data Collection, comprises the execution of the Systematic Mapping, during which the search for the primary studies is performed and the inclusion/exclusion criteria are used in order to select relevant studies according to the research questions. Finally, the third phase, Results, is responsible for reporting the study outcomes based on a classification scheme. The final outcome of this process is a systematic map.

## 4.2 Research directives

In this section we present the mapping protocol. For additional details please refer to our previous work (Santos et al. 2014).

### 4.2.1 Research questions and search string

The main purpose of this study is to map out existing templates for SPL use cases in the textual description form. Then, the research questions used were:

- **RQ1:** Which are the templates for the textual use cases of an SPL?
- **RQ2:** How could SPL variability be modeled in textual use cases?
- **RQ3:** Which variability types can be modeled in textual use cases of an SPL?

In order to answer these research questions, we defined the following search string: (“use case” or “use cases”) and (“product line” or “product lines” or “product family” or “product families” or SPL). We highlight that we removed the keywords “software”, “textual” and “template” from the search string because in preliminary searches with these keywords in the search string we lost important papers for our research goal.

### 4.2.2 Inclusion and exclusion criteria

For this mapping study, we defined just one inclusion criterion:

- **(IC1)** the study presents a template for textual use cases description of an SPL.

On the other hand, we defined four exclusion criteria:

- **(EC1)** the study is not written in English;
- **(EC2)** the study is just published as an abstract;
- **(EC3)** the study does not contain a template for textual use cases description with focus on an SPL; and
- **(EC4)** Duplicate use case template. When the same template is presented in different papers, only the paper that proposed the template was included. If this paper was not found, we included the newest paper.

### 4.2.3 Sources

The search was applied to seven databases. These databases are listed in Table 1 and were selected because we agree with (Morelli and Nakagawa 2011; Souza et al. 2013) that they index the most important publications in Software Engineering.

## 4.3 Data collection

This mapping study aims to identify studies describing textual use case templates for Software Product Lines. It was firstly conducted between January and March 2014 (Santos et al. 2014) and then updated on February 2015 in order to include papers published in 2014.

**Table 1** Electronic databases used in this SM study

Database	URL Address
ACM Digital Library	<a href="http://dl.acm.org">http://dl.acm.org</a> ,
Compendex	<a href="http://www.engineeringvillage2.org">http://www.engineeringvillage2.org</a>
IEEE Explorer	<a href="http://ieeexplore.ieee.org">http://ieeexplore.ieee.org</a>
ISI Web of Knowledge	<a href="http://www.isiknowledge.com">http://www.isiknowledge.com</a>
Science Direct	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>
Scopus	<a href="http://www.scopus.com">http://www.scopus.com</a>
SpringerLink	<a href="http://springerlink.com">http://springerlink.com</a>

In the search process, we considered the studies published until December 2014. Table 2 shows the number of papers, after performing each step of the study selection process. This mapping study started with 2394 primary studies returned from the following databases: ACM DL (711), IEEE Xplorer (31), ISI of Knowledge (51), Compendex (112), Science Direct (491), Scopus (128) and SpringerLink (870).

From Step 1 we found a large number by our search criteria. We believe that this can be attributed to the fact that we have used broad search terms. Furthermore, we do not excluded duplicate papers in this step.

After removing duplicate studies, and reading title and abstract of all the 2394 papers according to Step 2, the result was a set of 101 potentially relevant studies. In the third step, we read the full paper and selected those which have a textual use case template for an SPL with a focus on describing SPL variabilities within the use cases description. The result was a set of 23 papers.

Finally, in the fourth step we removed the duplicate templates. Thus, from the four steps of the study selection process, 12 studies were considered relevant and thus make up the final set of included papers.

#### 4.4 Results

In this section we describe the results of the systematic mapping study.

##### 4.4.1 Classification scheme

In order to classify the included papers, two categories were defined according to the research questions and the keywords identified in the papers:

**Category 1: variability description** This category shows how the use case template describes the SPL variabilities. The keywords in this category are:

- **Tags:** The use case template uses tags (e.g. [Vo], [ALT]) to indicate the variation points within the use cases. With the tags, use cases can also have a section where the variations are defined;

**Table 2** Study selection process

Step	Action	Papers
1	Search in the databases with the search string	2394
2	Reading title and abstract, removing of duplicate studies obtained by different databases and applying the inclusion and exclusion criteria (EC1, EC3)	101
3	Reading full paper and applying the inclusion and exclusion criteria (EC1, EC2, EC3)	23
4	Removing duplicate templates based on the EC4	12

- **Alternative scenarios:** The use case template describes the variations through alternative scenarios within the use cases description;
- **Specific section:** The use case template describes all information about the variation points in a specific section. In this section, the variation type, a brief description and the use case steps that are affected by the variation are specified;
- **Step identifier:** The use case template uses the step identifier of the use case to describe variants in use case scenarios;
- **Advice use case:** The use case template describes the variabilities as advice use cases. The advice use cases capture crosscutting requirements and are defined in the same form as normal use cases, but they may only have some of the use case sections. The linking of advice use cases with affected base use cases is based on syntactical matching of *joinpoints* and *pointcut expressions*.

**Category 2: variability type supported** With this category we wanted to know which variability types the use case template is able to describe. The keywords in this category are:

- **Optional:** The use case template supports the specification of optional steps;
- **Mandatory alternative exactly 1:** The use case template supports the specification of mutually exclusive alternatives for one mandatory step;
- **Mandatory alternative at least 1:** The use case template supports the specification of alternatives for one mandatory step out of which at least one must be selected;
- **Optional alternative exactly 1:** The use case template supports the specification of mutually exclusive alternatives for one optional step;
- **Optional alternative at least 1:** The use case template supports the specification of alternatives for one optional step out of which at least one must be selected.

Besides the two defined categories, we decided to use the category for classification of research defined by Wieringa et al. (2006), named *Research Type*. This category reflects the research approach used in the papers and is independent of a specific topic (Petersen et al. 2008).

#### 4.4.2 Outcomes

In this section we present the analysis and synthesis of the primary studies in consideration of each research question.

**Results related to RQ1** Table 3 presents the papers found at Step 03 and the included papers based on the EC4.

With this mapping study, we found twelve use case templates for an SPL. As shown in Table 4, the oldest template found by the mapping is dated 2003 and the newest template is dated 2013. Thus, observing that we have found templates that are recent, the specification of variability in textual use cases is still an interesting research topic. Moreover, most of the templates were published in international conferences and one of the templates was found in a book.

Bertolino and Gnesi (2003) propose a template called PLUC (Product Line Use Cases) that allow variations to be described, by explicitly enclosing within the sections of the use cases some tags that indicate the variable parts. In this template, there are three types of



**Table 3** Papers from Step 03 and final included papers based on the EC4

Id	Papers with the same UC template	Included paper
1	Bertolino and Gnesi (2003)	Bertolino and Gnesi (2003)
	Colanzi et al. (2013)	
	Assuncao et al. (2011)	
	Bertolino and Gnesi (2004)	
	Bertolino et al. (2006)	
	Fantechi et al. (2004)	
2	Bonifacio et al. (2008)	Kamsties et al. (2003)
	Fantechi et al. (2004)	
3	Kamsties et al. (2003)	Gomaa (2004)
4	Gomaa (2004)	
5	Nakanishi et al. (2007)	Bragança and Machado (2005)
	Bragança and Machado (2005)	
6	Eriksson et al. (2005)	Eriksson et al. (2005)
	Eriksson et al. (2005)	
	Alferez et al. (2014)	
7	Bonifacio et al. (2008)	Gallina and Guelfi (2007)
	Gallina and Guelfi (2007)	
8	Choi et al. (2008)	Choi et al. (2008)
9	Anthonyesamy and Some (2008)	Anthonyesamy and Some (2008)
10	Nguyen (2009)	Nguyen (2009)
11	Jirapanthong (2009)	Jirapanthong (2009)
12	Bonifacio and Borba (2009)	Bonifacio and Borba (2009)
	Alferez et al. (2014)	
13	Oliveira et al. (2013)	Oliveira et al. (2013)
	Oliveira et al. (2014)	

**Table 4** Included papers

Reference	Publication Fora	Year
Bertolino and Gnesi (2003)	European Software Engineering Conference	2003
Kamsties et al. (2003)	International Workshop on Software Product-Family Engineering	2003
Gomaa (2004)	Book	2004
Bragança and Machado (2005)	International Workshop on Model-Based Methodologies for Pervasive and Embedded Software	2005
Eriksson et al. (2005)	International Conference on Software Product Lines	2005
Gallina and Guelfi (2007)	International Working Conference on Requirements Engineering: Foundation for Software Quality	2007
Choi et al. (2008)	International Conference on Computer and Information Technology	2008
Anthonyesamy and Somé (2008)	AOSD Workshop on Early Aspects	2008
Nguyen (2009)	International Workshop on Modeling in Software Engineering	2009
Jirapanthong (2009)	International Conference on Advances in Information Technology	2009
Bonifacio and Borba (2009)	International Conference on Aspect-oriented Software Development	2009
Oliveira et al. (2013)	Brazilian Symposium on Software Components, Architectures and Reuse	2013

tags: alternative, parametric, and optional. Furthermore, the final product is also identified within the use cases. This template was found in 8 papers from the Step 03 (see Table 3). From these, the newest papers are focused on evaluating strategies for testing software product lines (Colanzi et al. 2013) and on the test reuse of an SPL (Assuncao et al. 2011). Both present a use case example with the template proposed by Bertolino and Gnesi (2003).

Similar to the template of Bertolino and Gnesi (2003), Kamsties et al. (2003) present an SPL use case template that uses tags to indicate a place in the use case where the variation occurs. However, they do not describe in the textual description the kind of variation (optional or alternative).

According to Gomaa (2004), fine-grained variation could be specified in the SPL use cases with the following elements: name, type, line of the use case (the target of the variation), and description. Then, Gomaa (2004) proposed that these variations should be described at the end of the SPL use cases. We also found this template in the work of Nakanishi et al. (2007), which reports an experimental case study constructing an SPL with the PLUS method (Gomaa 2004).

The use case template identified in the Bragança and Machado's paper (Bragança and Machado 2005) specifies the variation points through OPT and ALT tags. Using these tags, any text fragment of the textual use case description may be variant and this is explicitly marked by pairs of the XML-like tags <variant> and </variant>. An interesting feature of this template is the presence of questions related to the variation points that are used to guide the instantiation of the product use cases.

Eriksson et al. (2005) propose that use case scenarios of an SPL can be described using an extended version of the tabular RUP SE Black Box flow of events notation (IBM 2002). In this extended one, the step identifier of the flow is used to specify variant behavior. Another distinguishing feature of this template is the use of local (with \$) and global (with @) variables. This template was found in 4 papers from the Step 03 (see Table 3). From these, the newest paper (Alferez et al. (2014)) presents a metric-based evaluation aimed at assessing quality attributes such as modularity, stability and expressiveness of SPL requirements approaches like the PLUSS (Eriksson et al. 2005).

Gallina and Guelfi (2007) propose a Use Case Elicitation Template (UCET) that provides special fields to collect information concerning variabilities: a) selection category, which specifies whether the use case is mandatory, optional or alternative; b) a description of variation points, and c) fault variation description, used to describe the faults strongly related to the variation points. This template was inspired from templates of Bertolino and Gnesi (2003), Cockburn (2000) and Gomaa (2004).

Choi et al. (2008) propose a simpler tag notation than Bertolino and Gnesi (2003). In the template of Choi et al., the tags are used only for marking variation points in use case scenarios of SPL. Each tag is expanded in a section called "Variations" and is mapped to the Orthogonal Variability Model (OVM).

Anthonyamy and Somé (2008) propose an aspect-oriented use cases modelling approach to model product line systems that is supported by a tool called Use Case Editor (UCed). In this approach, the variabilities can be modeled by advice use cases that extend the behavior of base use cases. These advice use cases are defined in the same way as normal use cases, but they may only have some of the use case sections.

Nguyen (2009) extends Gomaa's template (Gomaa 2004) to specify non-functional requirements. In this template, additional sections are used to specify performance, usability, and security requirements. Each of them can have variation points just like other functional steps of the use cases.

The template proposed by Jirapanthong (2009) describes the variations with tags. Furthermore, in this template there is an attribute for specifying which domain of the product line it is and another one that designates which product member the use case is specified for.

Bonifacio and Borba (2009) propose an approach, Modelling Scenario Variability as Crosscutting Mechanisms (MSVCM), to deal with the variability of scenarios as a composition of different artifacts: use cases model, feature model, product configuration, and configuration knowledge. In this approach, the use cases model is composed of use cases and aspectual use cases. An aspectual use case has a name and a list of advices that can extend the behavior of existing scenarios. We also found this template in the work of Alferez et al. (2014) which presents an assessment of the MSVCM approach in regards to modularity, stability, and expressiveness.

Finally, the template presented by Oliveira et al. (2013) is an adaptation of Eriksson et al.'s template (Eriksson et al. 2005). However, unlike Eriksson et al.'s template, Oliveira et al. suggest that the features names should be clearly specified in the use case through the element named "Associated feature" and that the variations can be related to alternative scenarios within the use case. In a recent paper, Oliveira et al. (2014) present a case study with their approach.

**Results related to RQ2** Table 5 presents the results for the category *Variability Description*, which is related to how the SPL variabilities could be modeled in textual use cases. As described in Section 4.4.1, we identified five different template structures for SPL use cases: Tags, Specific Section, Alternative Scenarios, Step Identifier and Advice use case. In the following paragraphs we describe these structures and show examples of them in the specification of the use case "Withdraw Money" (Erikssona et al. 2004). This use case is composed by one optional variant, related to the use of the PIN for the user's identification, and two optional alternative variants, related with other two types of identification (through fingerprint or voice sample).

We can see in Table 5 that the **Tags** structure has the largest number of templates. When tags (e.g. V1) are used, often in the template there is a section where the SPL variability is described, as in the papers (Bertolino and Gnesi 2003; Kamsties et al. 2003; Gallina and Guelfi 2007; Choi et al. 2008; Jirapanthong 2009). However, in the template found in (Bragança and Machado 2005), the variability indicated by the tags is described in the main scenario.

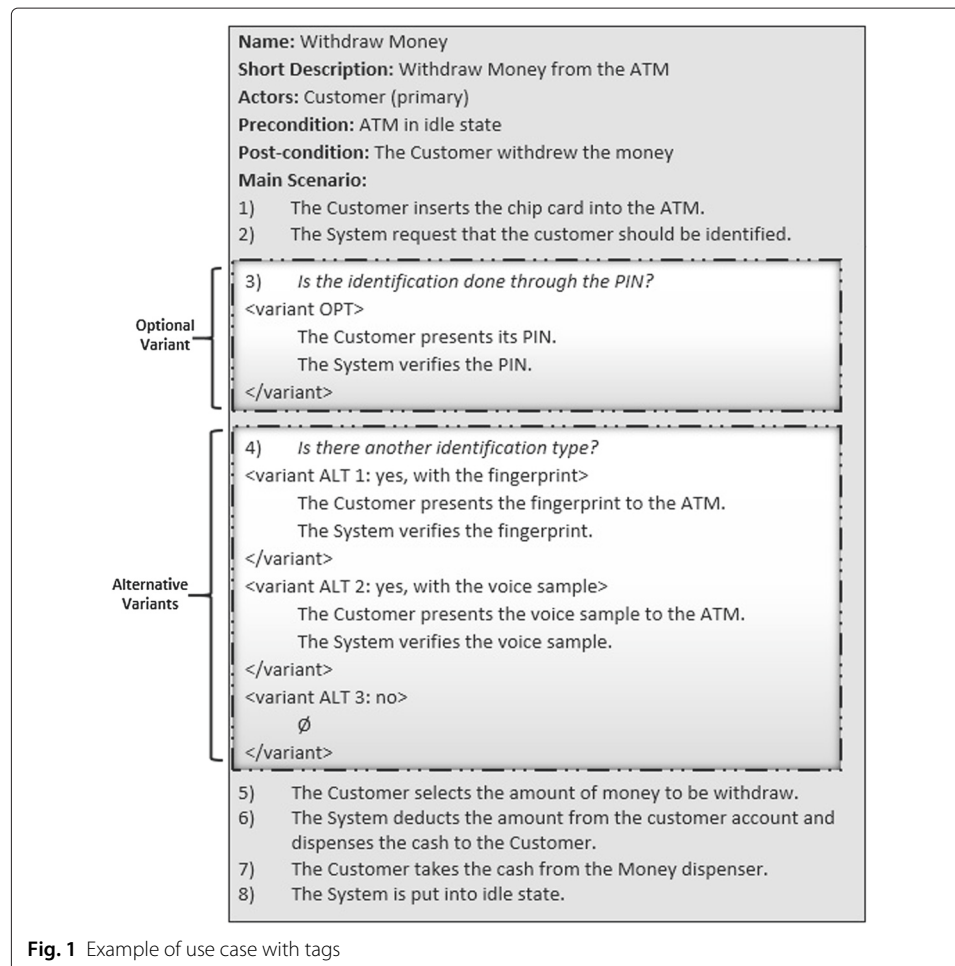
**Table 5** Templates per variability description

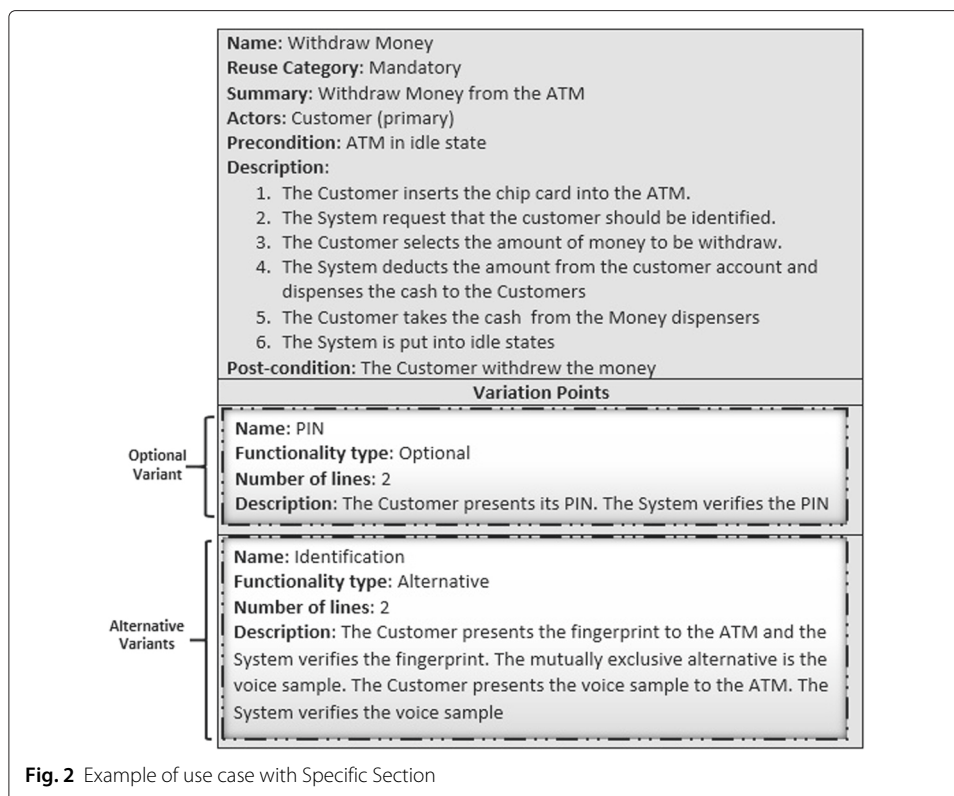
Variability description	Papers
Step Identifier	(Eriksson et al. 2005)
Specific Section	(Gomaa 2004; Nguyen 2009),
Alternative Scenarios	(Oliveira et al. 2013)
Advice Use Case	(Anthonysamy and Somé 2008; Bonifácio and Borba 2009),
Tags	(Bertolino and Gnesi 2003; Bragança and Machado 2005; Choi et al. 2008; Gallina and Guelfi 2007; Jirapanthong 2009; Kamsties et al. 2003)

Figure 1 presents the use case “Withdraw Money” in the template found in the Bragança and Machado work (Bragança and Machado 2005). The interesting about this template is the use of questions (e.g. *Is the identification done through the PIN?* and *Is there another identification type?*) related with the variabilities aiming to guide the instantiation of the product use cases. In this example, the optional variant is described between the tags <variant OPT> and </variant>. On the other hand, the alternative variants are described through the tags <variant ALT> and </variant>.

The Specific Section is present in two templates (Gomaa 2004; Nguyen 2009) while the Alternative Scenarios is used in one template (Oliveira et al. 2013). Both these template structures describe the SPL variability in a use case without affecting the description of the main use case scenario.

Figure 2 presents the use case “Withdraw Money” specified in the template proposed by Gomaa (2004). In this template, we can observe the description of the variabilities at the end of the use case. We highlight that Nguyen’s template (Nguyen 2009) extends Gomaa’s template (Gomaa 2004) to specify non-functional requirements. Then, in both templates the specification of the variabilities is made with the name of the variability, type of requirement (optional or alternative), line number of the use case affected by the variability, and the variability description.





The template of Oliveira et al. (2013) describes the variabilities as alternative scenarios. Figure 3 shows the use case “Withdraw Money” described in this template, where the variabilities were specified with the following associated features: “PIN”, “Fingerprint” and “Voice”.

The Step Identifier structure used by Eriksson et al. (2005) is interesting because it uses the step identifier to specify the alternative and optional steps. In this template, for example, several steps identified with the same number identify a number of alternatives for one mandatory step (see step 3 in Fig. 4) while a number step identifier within parenthesis identifies an optional step in the scenario (see steps 2 and 3 in Fig. 4).

Finally, the Advice Use Case structure is present in two templates (Anthonysamy and Somé 2008; Bonifácio and Borba 2009). Figure 5 shows the “Withdraw Money” specified in this template. In this case, we have one base use case named “Withdraw Money from the ATM” specifying the common behavior and three advice use cases that extends the behavior of the base use case: “Use PIN for user identification” that introduces an optional behavior before the step P2; “Use fingerprint for user identification” and “Use voice sample for user identification” that introduce an optional alternative behavior also before the step P2.

**Results related to RQ3** Table 6 presents the results for the category *Variability Type Supported*, regarding which variability types can be modeled in textual use cases of an SPL.

Five of the templates were not classified in this category because they do not specify explicitly the type of variability in the use cases description. They are: i) Kamties et al.’s

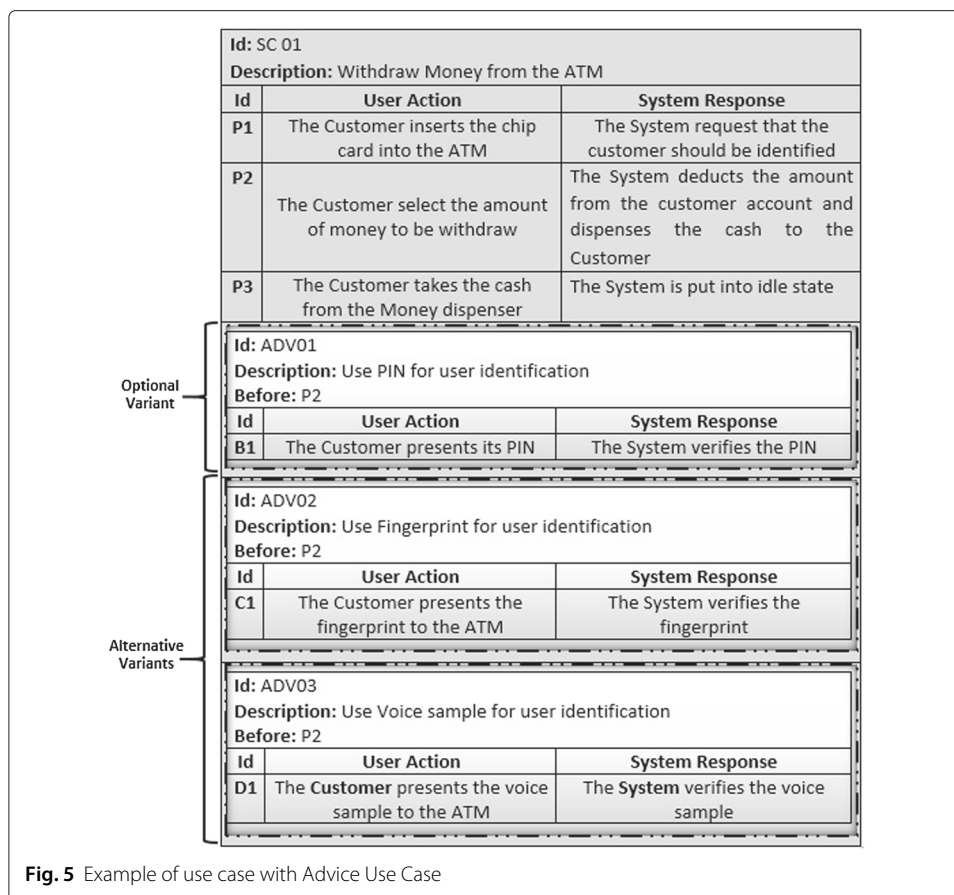
	*Name:		Withdraw Money		
	*Associated feature:		Withdraw	Actor(s) [0..*]: Customer	
	*Pre-condition:		ATM in idle state	*Post-condition: The Customer withdrew the money	
	*Main Success Scenario				
	Step	Actor Action	Black box System Response		
	1	The Customer inserts the chip card into the ATM	The System request that the customer should be identified		
	2	The Customer selects the amount of money to be withdraw	The System deducts the amount from the customer account and dispenses the cash to the Customer		
	3	The Customer takes the cash from the Money dispenser	The System is put into idle state		
	Optional Variant	Alternative Scenario Name:		PIN	
		Associated Feature [0..1]:		PIN	
Step		Actor Action	Black box System Response		
1.1		The Customer presents its PIN	The System verifies the PIN		
Alternative Variants		Alternative Scenario Name:		Fingerprint	
	Associated Feature [0..1]:		Fingerprint		
	Step	Actor Action	Black box System Response		
	1.2	The Customer presents the fingerprint to the ATM	The System verifies the fingerprint		
	Alternative Scenario Name:		Voice Sample		
	Associated Feature [0..1]:		Voice Sample		
	Step	Actor Action	Black box System Response		
	1.2	The Customer presents the voice sample to the ATM	The System verifies the voice sample		

**Fig. 3** Example of use case with Alternative Scenarios

template (Kamsties et al. 2003), which does not explicitly specify the type of variability; ii) Choi et al.'s template (Choi et al. 2008), which describes the variability type with the Orthogonal Variability Model; and iii) the templates of Oliveira et al. (2013), Bonifacio and Borba (2009), and Anthonysamy and Somé (2008), where the variability type is specified by the feature model.

<b>Step</b>		<b>Actor Action</b>	<b>Black Box System Action</b>
1		The use case begins when The Customer inserts the chip card into the ATM	The System request that the customer should be identified
<b>Optional Variant</b>		(2) The Customer presents its @PIN_SIZE character long PIN	The System verifies the PIN
<b>Alternative Variants</b>		(3) The Customer presents the fingerprint to the ATM	The System verifies the fingerprint
		(3) The Customer presents the voice sample to the ATM	The System verifies the voice sample
4		The Customer selects the amount of money to be withdraw	The System deducts the amount from the customer account and dispenses the cash to the Customer
5		The Customer takes the cash from the Money dispenser	The System is put into idle state

**Fig. 4** Example of use case with Step Identifier (Adapted from (Erikssona et al. 2004))



**Fig. 5** Example of use case with Advice Use Case

Regarding the seven other templates, the specification of Optional and Mandatory Alternative Exactly 1 is taken into account in all of them. The templates of Gomaa (2004) and Nyugun (2009) also support the specification of Optional Alternative Exactly 1, but only the template of Eriksson et al. (2005) supports all of the five variability types.

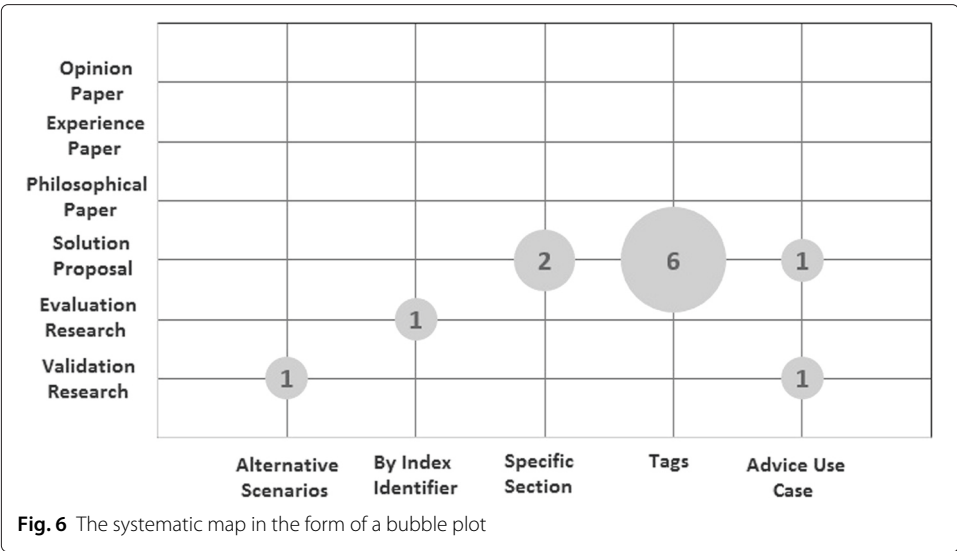
#### 4.4.3 Systematic map

We merged the categories *Research Type* and *Variability Description* in a systematic map (Fig. 6) aiming to generate a quick overview of the evidence gathered from this SM. The category *Research Type* was proposed by Wieringa et al. and has six keywords (Wieringa et al. 2006): Validation Research, Evaluation Research, Solution Proposal, Opinion Paper, Experience Paper and Philosophical Paper.

**Table 6** Templates per variability type supported

Variability type	Papers
Optional	(Bertolino and Gnesi 2003; Bragança and Machado 2005; Eriksson et al. 2005; Gallina and Guelfi 2007; Gomaa 2004; Jirapanthong 2009; Nguyen 2009)
Optional alternative at least 1	(Eriksson et al. 2005)
Optional alternative exactly 1	(Eriksson et al. 2005; Gomaa 2004; Nguyen 2009)
Mandatory alternative at least 1	(Eriksson et al. 2005)
Mandatory alternative exactly 1	(Bertolino and Gnesi 2003; Bragança and Machado 2005; Eriksson et al. 2005; Gallina and Guelfi 2007; Gomaa 2004; Jirapanthong 2009; Nguyen 2009)





In regards to the twelve included papers, we realized that Validation and Evaluation Research are weakly addressed, because we found only one paper (9%) in the Evaluation Research category and two papers (18%) in the Validation Research category. The paper of Eriksson et al. (2005) was classified as Evaluation Research, because it presents an industrial case study. The paper of Oliveira et al. (2013) was classified as Validation Research, because it presents hypotheses and statistical tests. The work of Bonifacio and Borba (2009) was also classified as Validation Research, because they present four case studies comparing their proposal with the PLUS approach (Eriksson et al. 2005).

On the other hand, the Solution Proposal, which is a non-empirical research category, is the topic with more entries (9 papers - 73%). Within this category, six studies (Bertolino and Gnesi 2003; Bragança and Machado 2005; Choi et al. 2008; Gallina and Guelfi 2007; Jirapanthong 2009; Kamsties et al. 2003) present the Tags structure to deal with the SPL variabilities, while only two papers (Nguyen 2009; Gomaa 2004) have used the Specific Section structure and one (Anthonysamy and Somé 2008) has used Advice Use Case.

No papers were classified as Opinion Paper, Experience Paper or Philosophical Paper. However, this was expected since we were looking for works with a concrete specification artifact.

Through this analysis we note that regarding the proposal of a use case template, most of the papers found propose a use case template only with examples from an academic or fictitious SPL. In the meantime, it is possible to find empirical work with the templates identified in this SM, such as the ones from Alferez et al. (2014), Nakanishi et al. (2007) and Oliveira et al. (2014).

4.5 Discussion

At first, it is important to highlight that this mapping study focused on how to describe SPL variability within the use case description. Therefore, approaches to describe variability based on include/extend relationships (e.g. Bragança and Machado (2006)) or activity diagrams (e.g. Yu et al. (2014)), for example, were not taken into account in this study. Moreover, works not using the use cases for the capture of SPL variabilities were not considered (e.g. Zhou et al. (2014)).



We also highlight that, in an SM, the articles are not evaluated regarding their quality, as the main goal is not to establish the state of evidence (Petersen et al. 2008). Therefore, no quality criteria were defined for this mapping study.

Regarding the mapping results, six studies (Bertolino and Gnesi 2003; Bragança and Machado 2005; Choi et al. 2008; Gallina and Guelfi 2007; Jirapanthong 2009; Kamsties et al. 2003) report a template that uses Tags to specify variability in the textual use case. However, when we compared with the first tag-based template (Bertolino and Gnesi 2003), the newest proposals (Choi et al. 2008; Jirapanthong 2009) have a simpler tag system. In the Choi et al. template (Choi et al. 2008), for example, the tagged use case scenario is mapped to the OVM and so, the tag does not have to specify the variability type. In this scenario, Bonifacio and Borba (2008) show that the use of the PLUC (Bertolino and Gnesi 2003) could result in maintainability issues because introducing a new product variant might require changes in several artifacts. Then, the maintainability could be a problem with the use of tags for specifying variabilities within the use cases descriptions.

Oliveira et al. (2013) use the Alternative Scenarios structure in the use case template to describe the variabilities. In a recent work, Oliveira et al. (2014) evaluated their requirements engineering approach, the Feature-Driven Requirements Engineering approach (FeDRE), through a case study for developing an SPL of mobile applications for emergency notifications. The results showed that the approach is easy to use and useful for specifying the functional requirements in this particular SPL.

The use of the Specific Section structure to deal with the variabilities is useful for describing variabilities on functional requirements, like in the template of Gomma (2004), and non-functional requirements, as used in the template of Nguyen (2009). However, the maintainability problems could be present in these templates because, in the variability description, there is the identification of the line number of the main scenario (the target of the variation). In that way, even a small change in the common use case scenario (e.g. a new mandatory step) could result in changes in the variability description.

In the template of Eriksson et al. (2005), the Step Identifier structure is used to describe the variabilities. This template seems to allow the quick identification of the variabilities while reading the use case scenarios, but no evidence related to this was found in the SM.

Bonifacio and Borba (2009) and Anthonysamy and Somé (2008) present the use of the Advice Use Cases structure for the separation between variability management and use case specifications. Furthermore, Bonifacio and Borba (2009) applied their approach in different case studies where they achieved a better feature modularity and scenario cohesion. Then, the use of aspect-oriented use cases modelling has good benefits when we have homogeneous crosscutting features and several variants for a scenario (Bonifácio and Borba 2009; Bonifácio et al. 2008).

Although some templates (Choi et al. 2008; Gallina and Guelfi 2007; Nguyen 2009; Oliveira et al. 2013) have been proposed based on previous ones, these studies do not empirically compare their proposed templates with previously defined ones. An exception is the template of Bonifacio and Borba. These authors first report the benefits of the separation of concerns by comparing their approach with other techniques for handling scenario variability management (Bonifácio et al. 2008), and after that, they describe an approach for use case scenario variability management (Bonifácio and Borba 2009).

On the other hand, there is some empirical work with the templates identified in this SM (Alferez et al. 2014; Nakanishi et al. 2007; Oliveira et al. 2014). The work of Alferez

et al. (2014), for example, evaluated the support of MSVCM (Bonifácio and Borba 2009), PLUSS (Eriksson et al. 2005) and two other approaches to modularity, expressiveness, and stability through the specification and evolution of a car crash crisis management system SPL. However, considering the included studies, we did not find any controlled experiment comparing the use case template structures in terms of perceived ease of use or comprehensibility.

Therefore, based on the mapping study results, we identify the following directions for future research: (i) Controlled experiments to compare the SPL use case templates; (ii) Application of the templates with real SPL, since most of the papers found propose a use case template only with examples from an academic or fictitious SPL; (iii) Development of other support tools, because just a few of the found works present a support tool; and (iv) Proposal of an SPL use case template to support different variability types and even more complex variations (e.g. alternative steps with cardinality).

Finally, two main threats to the validity of this study are: i) potential bias in the papers' selection, as this was performed by one reviewer, who is the primary author; and ii) some relevant use case templates may not be included.

In order to mitigate the first threat, the other two authors carefully reviewed the protocol and monitored the SM process and the analysis of the results. Furthermore, these authors performed data extraction on a sample of the primary studies and their results were cross-checked with those of the first author. Thus, we believe that the SM process was rigorously followed and that the results obtained are valid.

The main reasons for a study which contains a use case template for SPL has not been selected are: i) the publisher sources of the study are not indexed by the databases used in this mapping; ii) the study was not hit by the search string; and iii) the study was written up in a language other than English.

In order to mitigate the second threat, we used seven important electronic databases in Software Engineering according to (Morelli and Nakagawa 2011; Souza et al. 2013), which are frequently used in many systematic reviews, and the search string considered synonyms and acronyms. Regarding the study language, the choice of the English language is justified to make this mapping study replicable and feasible.

## 5 Comparison of the SPL use case template structures

In this section we present the controlled experiment conducted with the template structures found in the literature for textual specification of SPL use cases. The goal of this experimental study was to evaluate the effect of the template structure on the comprehensibility of SPL use cases. This evaluation was made with respect to the effort from the researcher's viewpoint.

### 5.1 Selected templates for the experiment

The execution of an experimental study with all twelve templates found would be costly, since the use of each template requires a lot of training and is time consuming. Thus, we first pick one template per each keyword of the *Variability Description* category.

With the Tags structure, we have found six templates (Bertolino and Gnesi 2003; Bragança and Machado 2005; Choi et al. 2008; Gallina and Guelfi 2007; Jirapanthong 2009; Kamsties et al. 2003) described in Section 4.4.2. Then, two selection criteria were established to define which template would be used in the experiment: a) the template

should not model the final product in the specification, since this reduces the maintainability; and b) the template should describe explicitly the variability type, because this allows the understanding of the use case without other models. As a result, the templates of Bertolino et al. (2003) and Jirapanthong (2009) were excluded due to the first criterion; and the templates proposed by Kamsties et al. (2003) and Choi et al. (2008) were excluded due to the second criterion. Then, from the templates of Gallina and Guelfi (2007), and Bragança and Machado (2005) we chose the last one, because it has questions associated with the variations points and we wanted to verify their impact on the SPL use cases' compressibility.

Using the Specific Section structure, we have found only two templates (Gomaa 2004; Nguyen 2009). As Nguyen's template (Nguyen 2009) is based on Gomaa's template (Gomaa 2004), we chose the template from Gomaa. Furthermore, the templates from Oliveira et al. (2013) and Eriksson et al. (2005) were selected to the experiment because they are unique with their structures (Alternative Scenario and Step Identifier).

From the templates with the Advice Use Case structure, we selected the template from Bonifacio and Borba (2009), because this is the newest with this kind of structure.

Finally, we decided to exclude the template of Oliveira et al. (2013) from this controlled experiment, because the use of the Alternative Scenarios structure is similar to common use case structures, since the variations are described just like alternative scenarios. Therefore, the controlled experiment was conducted with four use case templates representing four different structures to describe SPL variabilities in a use case.

### 5.1.1 Hypotheses, variables, and metrics

The overall goal of this investigation was to evaluate, with respect to effort, from the point of view of the researcher, in the context of Computer Science students and developers, the effect of the use case templates structures on the SPL use cases' comprehensibility. The SPL use cases' comprehensibility was measured based on the subjects' efficiency in understanding the SPL use cases used and performing the comprehension tasks. Therefore, aiming to achieve the goal, the experimental study was designed to answer the following research questions:

- **ERQ1:** Which of the evaluated template structures favors the SPL use cases' comprehensibility?
- **ERQ2:** Which of the evaluated template structures requires less time to understand a use case?

These research questions were then translated into the following hypotheses:

- $H_0$  *accuracy*: there is no statistically significant difference in the SPL use cases' comprehensibility using the evaluated template structures.  $H_1$  *accuracy*: there is a statistically significant difference in the SPL use cases' comprehensibility using the evaluated template structures;
- $H_0$  *time*: there is no statistically significant difference in terms of time required to understand the use cases' behavior using the evaluated template structures.  $H_1$  *time*: there is a statistically significant difference in terms of time required to understand the use cases' behavior using the evaluated template structures;

The independent variables (representing the inputs or causes) of the experimental study were the template of textual use case description, the use cases' examples, and the questionnaires. The use case templates used in this experiment were: i) Bragança and Machado (Tags structure), ii) Eriksson et al. (Step Identifier structure), iii) Gomaa (Specific Section structure), and iv) Bonifácio and Borba (Advice Use Case structure).

The dependent variables (representing the output or effect) were the accuracy and time spent with respect to comprehension tests used to measure the comprehensibility provided by the use case template structures. The accuracy was chosen as the measure of comprehension because we believe that the volunteers could only answer the questions of the comprehension test correctly if they understand the SPL use cases being evaluated. On the other hand, the time taken was collected in order to assess the effort related to the use cases' understanding.

The comprehension tests were made in order to evaluate the comprehension related to each use case considered in the experimental study (in the same way as was done in (Mustafa 2010)). Thus, for each use case of the experimental study, there was a comprehension test that consisted of two questions about the variations and one question related to the use case behavior. The goal of this last question (the final question) was to check if the participant had understood the use case. An example of this question is below:

"Before selecting the amount of money, the user must:

- a) at least enter the PIN
- b) at least insert the chip card in the ATM
- c) at least enter the fingerprint OR a sample voice
- d) at least enter the PIN, and fingerprint or a voice sample
- e) all of the above are incorrect"

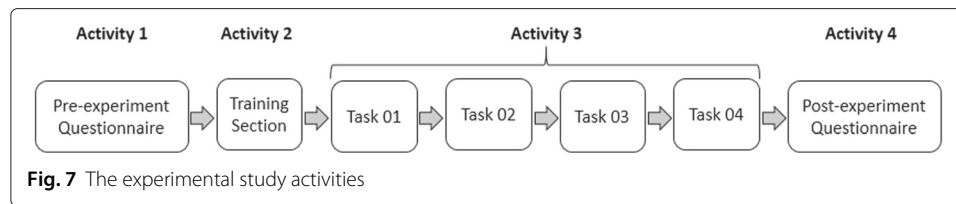
This question presented above is related to the use case "Withdraw Money" shown in Fig. 4. As we can see in this figure, steps 2 and 3 are optional ones. In this way, these steps can be executed or not. As a result, the correct answer to the question is the letter b, which refers to a step mandatory of the use case (see step 1 in the Fig. 4).

The final question of the comprehension test was used to validate the subject's task. Then, in order to evaluate the accuracy, we analyzed the participants' answers for the two first questions of the comprehension test, which asks the volunteer to describe the alternative and optional steps of the task use case.

The variable time spent was measured by collecting the time spent in minutes to answer the comprehension test in each task. For this, the participants registered the initial and final time for each comprehension task.

## 5.2 Experimental study design and subjects

Forty eight (48) volunteers participated in the study. Twenty one were undergraduate Computer Science students, 20 were graduate students (16 MSc and 4 PhD) in Computer Science, and seven were developers working at GREat - Group of Networking, Software and Systems Engineering (GREat 2015). The students were associated to four universities in Brazil.



The experiment was conducted through of the following activities (see Fig. 7):

1. The first activity in the study was related to answering a pre-experiment questionnaire. This questionnaire was applied to characterize all the participants as to their previous knowledge in the treatments of the study. Only three of the 20 undergraduate students had never studied textual use cases descriptions. Nine undergraduate students had never studied SPL. Only two graduate students had never worked with textual use case descriptions while eight graduate students had never studied SPL. All the developers had used textual use cases descriptions professionally, but three of them had never studied SPL. Therefore, we did not find a significant difference among levels of previous knowledge related to the participants that justified a special grouping. However, we performed the analysis of the collected data by splitting the groups according to their academic background in order to discover additional behavior related to their experience in software development.
2. In order to minimize the effects of the lack of knowledge on the experimental study factors, a training session was conducted with all the participants about basic concepts related to SPL and use cases, as well as about the use case template structures used in the experiment. In this training session, we applied a task similar to the comprehension tasks used in the experiment, but we did not assess their results.
3. After the training, the subjects executed the comprehension tasks (Task 01, 02, 03 and 04). With regards to the use cases used in the experiment, we selected the following: i) Keep Velocity (John and Muthig 2002); ii) Cook Food (Gomaa 2004); iii) Withdraw Money (Erikssona et al. 2004); and iv) Proceed to purchase (Bonifácio and Borba 2009)). The order of these use cases in the tasks was fixed, as shown in Table 7. Furthermore, the choice of which structure each volunteer would use in each task was made randomly. For this, the authors have described the four use cases in each template selected for the experiment. During the execution of each task, the subjects were asked to answer the comprehension test about the use case of the task.
4. Finally, the subjects were asked to answer a post-experiment questionnaire, containing, among other things, the most useful structure from their point of view.

**Table 7** Use cases used in the experiment's tasks

Task	Name	Source
Task 1	Cook Food	(Gomaa 2004)
Task 2	Withdraw Money	(Eriksson et al. 2005)
Task 3	Keep Velocity	(John and Muthig 2002)
Task 4	Proceed to purchase	(Bonifácio and Borba 2009)

All the experiment instruments (questionnaires, comprehension tests, and use cases) are available at (Santos IS 2015).

### 5.3 Results and data analysis

Table 8 presents a summary of the collected data. The study started with 48 subjects; however, some of the subjects' tasks were not approved, since the final question of the comprehension test was not answered correctly.

The final question was associated with an important question about the use case understanding. A wrong answer in this question signaled a serious error in the use of the templates. Due to this fact, there are only 134 executions able to be evaluated, since 58 executions were not approved. The result analysis was then made from all cases with a correct answer to the final question.

In order to analyze the results, we have applied several tests using the SPSS tool (IBM 2015). We applied the Kolmogorov-Smirnov test (Hollander and Wolfe 1999; Wohlin et al. 2000) to assess if it is reasonable to assume that both data sets come from a normal population. We have split the data set into two groups, since there are several participants who did not answer correctly the final question, avoiding the use of their data in these analyses. The results for the time and accuracy variables are given in the Table 9 (for the group with the final question correct). Note that the p-values for the Kolmogorov-Smirnov tests are near 0.000 (in the row "Asymp. Sig."). This implies that the data set is not normal because the p-value was smaller than the significance level (alpha) 0.05. Large significance values ( $>0.05$ ) indicate that the observed distribution corresponds to the theoretical distribution. In this case, the significance value for time and accuracy do not exceed 0.05.

**Table 8** Summary of the experimental study

Level	Structure	Samples (#)	Avg Time (m)	Accuracy (%)
Undergraduate student	Advice Use Case	10	8.20	50.00
	Step Identifier	9	5.55	88.88
	Specific Section	21	6.04	73.80
	Tags	15	7.13	53.33
	SubTotal	55	6.65	66.36
Graduate student	Advice Use Case	12	8.41	50.00
	Step Identifier	22	4.50	95.45
	Specific Section	15	5.40	96.67
	Tags	7	7.71	85.71
	SubTotal	56	5.98	84.82
Developers	Advice Use Case	1	6.00	50.00
	Step Identifier	8	4.50	100.00
	Specific Section	5	5.20	70.00
	Tags	9	5.00	100.00
	SubTotal	23	4.91	91.30
Total	Advice Use Case	23	8.21	50.00
	Step Identifier	39	4.74	94.87
	Specific Section	41	5.70	81.70
	Tags	31	6.64	74.19
	SubTotal	134	6.07	78.35

**Table 9** Data set normality test

Description	Time	Accuracy
Kolmogorov-Smirnov Z	1.833	5.004
Asymp. Sig (2 tailed)	0.002	0.000

The test indicated that the data set did not follow a normal distribution, so, it was necessary to use non-parametric tests for hypothesis testing. We have used the Kruskal Wallis test (Hollander and Wolfe 1999; Wohlin et al. 2000), a non-parametric method, for testing whether samples originate from the same distribution (an alternative for one-way ANOVA). The test indicated that the time spent and the accuracy of the four groups (related to the four templates) had statistically significant differences. Thus, with this result the hypothesis  $H_{0 \text{ accuracy}}$  and  $H_{0 \text{ time}}$  can be rejected.

Once it was detected that there were differences in the time and accuracy associated with the templates' use, an analysis was made of the data crossing template by template, trying to identify the differences. In this case, we have used the MannWhitney test (Wohlin et al. 2000) for hypothesis testing. It is a non-parametric test of the null hypothesis and has greater efficiency than the *t-test* on non-normal distributions.

Table 10 presents the results of the hypothesis testing performed in order to compare the results related to time to complete the experiment tasks and the accuracy of the task execution. The values present in this table are the asymptotical significance of the comparison between templates, using the Mann-Whitney test. Values below 0.05 indicate a statistically significant difference between the results.

The results related to the use of the Advice Use Case structure have shown the worst results for both accuracy and time to perform the tasks. The time spent using this structure is statistically higher than using other structures. The accuracy is statistically lower than the accuracy related to the use of other structures (values smaller than 0.05 in Table 10). Besides that, this structure was selected as preferred by only 3% of the volunteers according to the post-experiment questionnaire (considering only the 134 valid executions with the final question correct).

The use of both Specific Section and Tags structures have shown results statistically equal to each other, both in time to perform tasks, and the accuracy of the result. The accuracy from the use of these structures indicated a statistically significant difference (better) when compared to the Advice Use Case structure. Considering only the valid executions, the Specific Section structure had a preference of 29.1% of the experiment volunteers and the Tags structure had a preference of 21.6%.

The use of the Step Identifier structure has shown to have the best results in the experiment. The time to perform the tasks was lower and it was still possible to register a statistically greater accuracy result than by using the other structures (values smaller than

**Table 10** Comparison among the use case template structures

Template	Advice use case		Step Identifier		Specific section		Tags	
	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy
Advice Use Case	-	-	0.000	0.000	0.001	0.001	0.046	0.015
Step Identifier	0.000	0.000	-	-	0.019	0.012	0.024	0.008
Specific Section	0.001	0.001	0.019	0.012	-	-	0.532	0.618
Tags	0.046	0.015	0.024	0.008	0.532	0.618	-	-

0.05 in Table 10). In terms of participants' preference, it was also the favorite, as indicated by 46.3% of volunteers. Thus, the Step Identifier structure, besides being the one with the best results in terms of time and accuracy, was also the favorite among the volunteers.

### 5.3.1 Answering the experiment research questions

After the study execution it was possible to answer the proposed experiment research questions.

- **ERQ1:** Which of the evaluated template structures favors the SPL use cases' comprehensibility? Answer: The Tags structure, represented by the template of Eriksson et al., favors the understanding of the use cases. The subjects using this template structure had better results in terms of accuracy;
- **ERQ2:** Which of the evaluated template structures requires less time to understand a use case? Answer: The Tags structure, represented by the template of Eriksson et al., requires less time to understand a use case. The subjects using this template structure had better results in terms of time spent.

## 5.4 Discussion

With the performed experiment we observed which characteristics of each use case template structure impact on the SPL use cases' comprehensibility. Subjects who selected the Step Identifier structure as the best structure, for example, reported that it has a simple description and an objective, clean, organized, and compact structure. These characteristics make it easy and intuitive to identify whether the use case steps were mandatory, optional, or alternative.

On the other hand, the disapproval related to the Advice Use Cases structure may be justified due to the separation between the main flow and its variation without an explicit definition of the variation type, making it difficult to understand if the variation is optional or alternative. However, it is important to note that, for this experiment, we did not use the feature model in the tasks. So, it is possible that, with the feature model, this structure had better results, because the approach of Bonifacio and Borba (2009) is defined as a composition of different artifacts: use cases model, feature model, product configuration, and configuration knowledge. Furthermore, the examples used in the experiment have a low level of crosscutting features and this may have affected the results related to this structure, since the main advantage of this structure is in specifying crosscutting features.

With respect to the Specific Section structure, some volunteers said that this structure eased the identification of alternative and optional features by presenting all the variations described at the end of the use case.

Finally, concerning the Tags structure, some volunteers enjoyed its sequential structure placing the variations with tags within the main flow of the use case. In general, the questions in the use cases associated to the variation points also improved the use cases' understanding.

In regards to the validity of the results, we discuss in the next paragraphs the internal and external validity.

**Internal validity** The experiment design was planned to minimize the effects of the threats to internal validity. We planned to minimize the effect of instrumentation by performing measures by a single person during the experiment. The variable time spent was



measured based on the initial and final time registered by the participants, which could be a threat to the validity. However, we highlight that we monitored the volunteers during all the experiment in order to guarantee the concreteness of the registered time. The participants were also monitored through log-sheets of their daily performances and they were allowed to use self-reporting. This gave us a chance to observe any effects due to History and Maturation.

As mentioned before, we used questions to evaluate the correct understanding of the use case. The final question was used to cut off the executions that had low quality. In order to avoid threats related to the use of a single group, the four treatments (use case template structures) were used for all the subjects. The order of template used was totally random. We did this to avoid the compensatory equalization of treatments, compensatory rivalry, and resentful demoralization (Wohlin et al. 2000).

**External validity** With respect to the external validity, the use of students as subjects is a threat. The reason for using students in our study was the availability sampling technique. However, we agree with Ferrari et al. (2010) and Dahan et al. (2014) that the use of students should not diminish the results of a controlled experiment, as important results have been found in other Software Engineering studies when student-based studies have been conducted. Furthermore, the study was executed with the presence of 41 students (undergraduate and graduate) and 7 developers. Therefore, we believe that our sample was representative.

The four use cases used in the study are similar in terms of complexity and size. All the use cases are small, since a larger example would demand effort incompatible with the time available for the study. Thus, the four use cases of this experiment may not be representative in terms of size and complexity, but we agree with Dahan et al. (2014) that this limitation is true for almost all controlled experiments conducted in the area of software engineering. In addition, we expect the same results for bigger use cases. Maybe, the difference among the two approaches that got statistically equals results could change, showing that one approach is better than other in this context.

## 6 Conclusions

In Software Product Line development, the requirements engineering activity needs to cope with common and variable requirements for the whole set of products in the family. For this purpose, there are several use case templates available in the literature to describe the functional requirements of an SPL. However, to the best of our knowledge, no efforts have been made to collect and summarize these existing templates. Furthermore, the work proposing an SPL use case template often does not empirically compare the proposed template with other templates.

In this scenario, the contributions of this paper are: i) offer a summary of existing use case templates as a result of a systematic mapping study with a focus on how to describe SPL variability in textual use cases; and ii) describe the results of a controlled experiment with four SPL use case template structures.

By means of this mapping it was possible to identify twelve textual use case templates that describe SPL variability and five different template structures for describing SPL variability in textual use cases: i) using tags; ii) with the step identifier of the use cases;

iii) describing the variability in a specific section of the textual use case; iv) through alternative scenarios; and v) with advice use cases.

We also identified with this SM, the following interesting directions for future research: (i) controlled experiments to compare the SPL use case templates; (ii) application of the templates with real SPLs; (iii) development of more support tools; and (iv) proposal of SPL use case templates to support more complex variations (e.g. alternative steps with cardinality).

The goal of the controlled experiment was to evaluate the SPL use cases' comprehensibility provided by the template structures. As a result, evidence collected shows that the description of commonalities together with variabilities makes the understanding of use cases more clear and the description of all variations at the end of the use case promotes a quick identification of variations.

With regard to the template structures evaluated in this experiment, the Step Identifier structure, where the variability is described in the step identifiers of the use case through some conventions (e.g. two steps identified with the same number are alternative steps), had the best results. With this structure the volunteers spent less time on the tasks and achieved a greater accuracy than by using other structures. This was also the preferred structure according to the results of the post-experiment questionnaire.

Moreover, the description of the variabilities at the end of the use case in the Specific Section structure and the use of questions in the structure presented by Bragança and Machado (Tags structure) were also identified as important characteristics to the understanding of the SPL use case.

Therefore, the results of the systematic mapping study could be interesting to researchers and practitioners who would like to propose a new SPL use case template or to investigate which template is better. On the other hand, the experiment results presented in this paper can provide a basis for other efforts to compare use case templates or to propose new use case template structures based on a controlled experiment.

Since one limitation of the conducted controlled experiment was the size of the use case descriptions, which has just two or three variations, we believe that an interesting future work is the replication of this experiment with more complex use cases. Also, it would be interesting to apply the other SPL use case templates and to use them in the industry.

#### **Competing interests**

The authors declare that they have no competing interests.

#### **Authors' contributions**

ISS carried out the systematic mapping study, and identified the use case templates in the existing work. All authors analyzed the results of the systematic mapping study. PASN and ISS defined the experiment design. ISS conducted the experiment with the volunteers. PASN conducted the result analysis and the hypothesis testing of the experiment. All authors read and approved the final manuscript.

#### **Acknowledgements**

This work is a partial result of the UbiStructure project supported by CNPq (MCT/CNPq 14/2011 - Universal) under grant number 481417/2011-7 and the Maximum project supported by FUNCAP (FAPs/INRIA/INS2i-CNRS 11/2011).

#### **Author details**

<sup>1</sup>Federal University of Ceara, Department of Computer Science, Fortaleza-CE, Brazil. <sup>2</sup>Federal University of Piaui, Department of Computer Science, Teresina-PI, Brazil.

Received: 6 December 2014 Accepted: 11 May 2015

Published online: 28 May 2015

## References

- Alves V, Niu N, Alves C, Valença G (2010) Requirements engineering for software product lines: A systematic literature review. *Inf Softw Technol* 52:806–820
- Alferez M, Bonifácio R, Teixeira L, Accioly P, Kulesza U, Moreira A, Araújo J, Borba P (2014) Evaluating scenario-based spl requirements approaches: the case for modularity, stability and expressiveness. *Requir Eng J* 19:355–376
- Anthony P, Somé S (2008) Aspect-oriented use case modeling for software product lines. In: *Proceedings of the 2008 AOSD Workshop on Early Aspects*. ACM, New York, NY, USA. pp 5–158
- Asadi M, Bagheri E, Mohabbati B, Gasevic D (2012) Requirements engineering in feature oriented software product lines: an initial analytical study. In: *Proceedings of the 16th International Software Product Line Conference*. ACM, New York, NY, USA. pp 36–44
- Assuncao WKG, Trindade DFG, Colanzi TE, Vergilio SR (2011) Evaluating test reuse of a software product line oriented strategy. In: *Proceedings of the 12th Latin American Test Workshop (LATW)*. IEEE Computer Society, Washington, DC, USA
- Azevedo S, Machado RJ, Bragança A, Ribeiro H (2012) On the refinement of use case models with variability support. *Innov Syst Softw Eng* 8:51–64
- Bertolino A, Gnesi S (2003) Use case-based testing of product lines. In: *Proceedings of the 9th European Software Engineering Conference*. ACM, New York, NY, USA. pp 355–358
- Bertolino A, Gnesi S (2004) Pluto: A test methodology for product families. *Lect Notes Comput Sci* Volume 3014:181–197
- Bertolino A, Fantechi A, Gnesi S, Lami G (2006). In: Käkölä T, Duenas JC (eds). *Software Product Lines Research Issues in Engineering and Management*. Springer, Brazil. Chap. 11 - Product Line Use Cases: Scenario - Based Specification and Testing of Requirements
- Benavides D, Segura S, Ruiz-Cortés A (2010) Automated analysis of feature models 20 years later: A literature review. *Inf Syst* 35(6):615–636
- Blanes D, Insfrán E (2012) A comparative study on model-driven requirements engineering for software product lines. *Revista de Sistemas e Computação (RSC Journal)* 2:3–13
- Bonifácio R, Borba P (2009) Modeling scenario variability as crosscutting mechanisms. In: *Proceedings of the 8th ACM International Conference on Aspect-oriented Software Development*. ACM, New York, NY, USA. pp 125–136
- Bragança A, Machado RJ (2005) Deriving software product line's architectural requirements from use cases: an experimental approach. In: *Proceedings of the 2nd International Workshop on Model-Based Methodologies for Pervasive and Embedded Software*. Turku Centre for Computer Science, Turku, Finland
- Bragança A, Machado RJ (2006) Extending uml 2.0 metamodel for complementary usages of the <<extend>> relationship within use case variability specification. In: *Proceedings of the 10th International on Software Product Line Conference*. IEEE Computer Society, Washington, USA. pp 123–130
- Bonifácio R, Borba P, Soares S (2008) On the benefits of scenario variability as crosscutting. In: *Proceedings of the 2008 AOSD Workshop on Early Aspects*. ACM, New York, NY, USA. pp 6–168
- Cheng BHC, Atlee JM (2007) Research directions in requirements engineering. In: *Proceedings of the Future of Software Engineering (FOSE '07)*. IEEE Computer Society, Washington, DC, USA
- Chen L, Babar MA, Ali N (2009) Variability management in software product lines: a systematic review. In: *Proceedings of the 13th International Software Product Line Conference*. Carnegie Mellon University, Pittsburgh, PA, USA. pp 81–90
- Cockburn A (2000) *Writing Effective Use Cases*. 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
- Choi W, Kang S, Choi H, Baik J (2008) Automated generation of product use case scenarios in product line development. In: *Proceedings of the International Conference on Computer and Information Technology*. IEEE Computer Society, Washington, DC, USA
- Colanzi TE, Assunção WKG, Trindade DFG, Zorzo CA, Vergilio SR (2013) Evaluating different strategies for testing software product lines. *J Electronic Testing* 29:9–24
- Dahan M, Shoval P, Sturm A (2014) Comparing the impact of the oo-dfd and the use case methods for modeling functional requirements on comprehension and quality of models. *Requir Eng J* 19:27–43
- Erikssona M, Borstler J, Borg K (2004) Marrying features and use cases for product line requirements modeling of embedded systems. In: *Proceedings of the Fourth Conference on Software Engineering Research and Practice in Sweden*. Institute of Technology, Unitryck, Linköping University, Linköping, Sweden. pp 73–82
- Eriksson M, Börstler J, Borg K (2005) The pluss approach: domain modeling with features, use cases and use case realizations. In: *Proceedings of the 9th International Conference on Software Product Lines (SPLC'05)*. Springer-Verlag, Berlin, Heidelberg. pp 33–44
- Eriksson M, Morast H, Börstler J, Borg K (2005) The pluss toolkit?: Extending telelogic doors and ibm-rational rose to support product line use case modeling. In: *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*. ACM, New York, NY, USA. pp 300–304
- Fantechi A, Gnesi S, Lami G, Nesti E (2004) A methodology for the derivation and verification of use cases for product lines. In: *Proceedings of the International Software Product Line Conference*. Springer-Verlag, Berlin, Heidelberg
- Fantechi A, Gnesi S, John I, Lami G, Dörr J (2004) Elicitation of use cases for product lines. In: *Proceedings of International Workshop on Software Product-Family Engineering*. Springer-Verlag, Berlin, Heidelberg
- Fant JS, Gomaa H, Pettit RG (2013) A pattern-based modeling approach for software product line engineering. In: *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)*. IEEE Computer Society, Washington, DC, USA. pp 4985–4994
- Ferrari R, Miller JA, Madhavji NH (2010) A controlled experiment to assess the impact of system architectures on new system requirements. *Requir Eng J* 15:215–233
- Gallina B, Guelfi N (2007) A template for requirement elicitation of dependable product lines. In: *Proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ'07*. Springer, Berlin, Heidelberg. pp 63–77
- Galster M, Weyns D, Tofan D, Michalik B, Avgeriou P (2014) Variability in software systems - a systematic literature review. *IEEE Trans Softw Eng* 40:282–306

- Gomaa H (2004) Designing Software Product Lines with UML: from Use Cases to Pattern-based Software Architectures. Addison Wesley, Redwood City, CA, USA
- GREat (2015) Group of Networking, Software and Systems Engineering. <http://www.great.ufc.br/index.php/en.html>
- Hadar I, Kuflik T, Perini A, Reinhartz-Berger I, Ricca F, Susi A (2010) An empirical study of requirements model understanding: Use case vs. tropos models. In: Proceedings of the 2010 ACM Symposium on Applied Computing. ACM, New York, NY, USA, pp 2324–2329
- Hollander M, Wolfe DA (1999) Nonparametric Statistical Methods. John Wiley & Sons, USA
- IBM (2002) The Rational Unified Process for Systems Engineering Whitepaper. <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/TP165.pdf>
- IBM (2015) Predictive Analytics Software and Solutions. <http://www-01.ibm.com/software/analytics/spss/>
- Jeyaraj A, Sauter VL (2007) An empirical investigation of the effectiveness of systems modeling and verification tools. *Commun ACM* 50(6):62–67
- Jirapanthong W (2009) Analysis on relationships among software models through traceability activity. In: Proceedings of the 3rd International Conference on Advances in Information Technology (IAIT). Springer-Verlag, Berlin, Heidelberg
- John I, Muthig D (2002) Product line modeling with generic use cases. In: Proceedings of the Workshop on Techniques for Exploiting Commonality Through Variability. Springer-Verlag, Berlin, Heidelberg
- Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS (1990) Feature-oriented domain analysis (foda) - feasibility study. Technical report, Software Engineering Institute, Carnegie Mellon University
- Kamsties E, Pohl K, Reis S, Reuys A (2003) Testing variabilities in use case models. In: Proceedings of the 5th International Workshop Software Product-Family Engineering. Springer-Verlag, Berlin, Heidelberg. pp 6–18
- Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical report, EBSE Technical Report
- Kuloor C, Eberlein A (2002) Requirements engineering for software product lines. In: Proceedings of the International Conference Software Systems Engineering and Their Applications. CNAM - Conservatoire National des Arts et Métiers, Paris, France
- Mustafa BA (2010) An experimental comparison of use case models understanding by novice and high knowledge users. In: Proceedings of the 2010 Conference on New Trends in Software Methodologies, Tools and Techniques. IOS Press, Amsterdam. pp 182–199
- Morelli LB, Nakagawa EY (2011) A panorama of software architectures in game development. In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE). Knowledge Systems Institute Graduate School, Skokie, USA. pp 752–757
- Nakanishi T, Fujita M, Yamazaki S, Yamashita N, Ashihara S (2007) Tailoring the domain engineering process of the plus method. In: Proceedings of the 14th Asia-Pacific Software Engineering Conference. IEEE Computer Society, Washington, DC, USA. pp 486–493
- Neiva D (2009) Riple-re: A requirements engineering process for software product lines. Master's thesis, Federal University of Pernambuco
- Neto PAMS, Machado IC, McGregor JD, Almeida ES, Meira SRL (2011) A systematic mapping study of software product lines testing. *Inf Softw Technol* 53:407–423
- Nguyen QL (2009) Non-functional requirements analysis modeling for software product lines. In: Proceedings of the ICSE Workshop on Modeling in Software Engineering (MISE '09). IEEE Computer Society, Washington, DC, USA
- Niu N, Easterbrook S (2008) Extracting and modeling product line functional requirements. In: Proceedings of the 16th IEEE International Requirements Engineering Conference. IEEE Computer Society, Washington, DC, USA
- Northrop LM (2002) Sei's software product line tenets. *IEEE Softw.* 19:32–40
- Northrop LM, Clements PC (2007) A Framework for Software Product Line Practice, Version 5.0. [http://www.sei.cmu.edu/productlines/frame\\_report/](http://www.sei.cmu.edu/productlines/frame_report/)
- Oliveira RP, Insfran E, Abrahão S, Gonzalez-Huerta J, Blanes D, Cohen D, Almeida ES (2013) A feature-driven requirements engineering approach for software product lines. In: Proceedings of the VII Brazilian Symposium on Software Components, Architectures and Reuse. IEEE Computer Society, Washington, DC, USA
- Oliveira RP, Blanes D, Gonzalez-Huerta J, Insfran E, Abrahão S, Cohen S, Almeida ES (2014) Defining and validating a feature-driven requirements engineering approach. *J Universal Comput Sci* 20(5):666–691
- Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. British Computer Society, Swinton, UK, UK. pp 68–77
- Reinhartz-Berger I, Sturm A (2014) Comprehensibility of uml-based software product line specifications. *Empirical Softw Eng* 19(3):678–713
- Santos IS, Neto PAS, Andrade RMC (2013) A use case textual description for context aware spl based on a controlled experiment. In: Proceedings of the CAISE'13 Fórum. CEUR-WS.org, Valencia, Spain
- Santos IS, Andrade RMC, Neto PAS (2014) How to describe spl variabilities in textual use cases: A systematic mapping study. In: Proceedings of the Eighth Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS). IEEE Computer Society, Washington, DC, USA
- Santos IS (2015) Experiment Data. <https://sites.google.com/site/ismaylesantos/spl-use-case-experiment>
- Souza EF, Falbo RA, Vijaykumar NL (2013) Knowledge management applied to software testing: a systematic mapping. In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE). Knowledge Systems Institute Graduate School, Skokie, USA
- Tiwari S, Gupta A (2013) A controlled experiment to assess the effectiveness of eight use case templates. In: 20th Asia-Pacific Software Engineering Conference, vol. 1. IEEE Computer Society, Washington, DC, USA. pp 207–214
- Urli S, Blay-Fornarino M, Collet P (2014) Handling complex configurations in software product lines: A toolled approach. In: Proceedings of the 18th International Software Product Line Conference - Volume 1. ACM, New York, NY, USA. pp 112–121. <http://doi.acm.org/10.1145/2648511.2648523>
- Wieringa R, Maiden N, Mead N, Rolland C (2006) Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requir Eng J* 11:102–107

- Wohlin C, Runeson P, Host M, Ohlsson M, Regnell B, Wesslen A (2000) Experimentation in Software Engineering: An Introduction. Kluwer Publishers, Norwell, MA, USA
- Yu W, Zhang W, Zhao H, Jin Z (2014) Tdl: a transformation description language from feature model to use case for automated use case derivation. In: Proceedings of the 18th International Software Product Line Conference. ACM, New York, NY, USA
- Zhou J, Lu Y, Lundqvist K, Lonn H, Karlsson D, Liwang B (2014) Towards feature-oriented requirements validation for automotive systems. In: Proceedings of the IEEE 22nd International Requirements Engineering Conference. IEEE Computer Society, Washington, DC, USA. pp 428–436

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---