Accepted Manuscript

Identification and Analysis of the Elements Required to Manage
Technical Debt by Means of a Systematic Mapping Study

Carlos Fernández-Sánchez, Juan Garbajosa, Agustín Yagüe,
Jennifer Perez

PII: S0164-1212(16)30213-8
DOI: [10.1016/j.jss.2016.10.018](https://doi.org/10.1016/j.jss.2016.10.018)
Reference: JSS 9869

To appear in: *The Journal of Systems & Software*

Received date: 8 March 2016
Revised date: 30 September 2016
Accepted date: 19 October 2016

Please cite this article as: Carlos Fernández-Sánchez, Juan Garbajosa, Agustín Yagüe, Jennifer Perez, Identification and Analysis of the Elements Required to Manage Technical Debt by Means of a Systematic Mapping Study, *The Journal of Systems & Software* (2016), doi: [10.1016/j.jss.2016.10.018](https://doi.org/10.1016/j.jss.2016.10.018)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© <year>. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Highlights

- Identification and classification of the elements to manage technical debt (TD)
- The classification resulted in a framework including the stakeholders interests
- The industrial relevance of current support to elements was assessed
- The framework is a basis for building TD decision-making models
- It was found out that TD management decisions are context-dependent

Identification and Analysis of the Elements Required to Manage Technical Debt by Means of a Systematic Mapping Study

Carlos Fernández-Sánchez^a, Juan Garbajosa^{a,b}, Agustín Yagüe^{a,b}, Jennifer Perez^{a,b}

^a*Technical University of Madrid (UPM) - CITSEM, ctra. de Valencia Km7, E-28031 Madrid, Spain*

^b*Technical University of Madrid (UPM) - ETSISI, ctra. de Valencia Km7, E-28031 Madrid, Spain*

Abstract

Technical debt, a metaphor for the long-term consequences of weak software development, must be managed to keep it under control. The main goal of this article is to identify and analyze the elements required to manage technical debt. The research method used to identify the elements is a systematic mapping, including a synthesis step to synthesize the elements definitions. Our perspective differs from previous literature reviews because it focused on the elements required to manage technical debt and not on the phenomenon of technical debt or the activities used in performing technical debt management. Additionally, the rigor and relevance for industry of the current techniques used to manage technical debt are studied. The elements were classified into three groups (basic decision-making factors, cost estimation techniques, practices and techniques for decision-making) and mapped according three stakeholders' points of view (engineering, engineering management, and business-organizational management). The definitions, classification, and analysis of the elements provide a framework that can be deployed to help in the development of models that are adapted to the specific stakeholders' interests to assist the decision-making required in

Email addresses: carlos.fernandez@upm.es (Carlos Fernández-Sánchez), jgs@etsisi.upm.es (Juan Garbajosa), agustin.yague@upm.es (Agustín Yagüe), jenifer.perez@upm.es (Jennifer Perez)

technical debt management and to assess existing models and methods. The analysis indicated that technical debt management is context dependent.

Keywords: Technical debt, Technical debt management, Systematic mapping, Decision making, Basic decision-making factors, Cost estimation techniques, Practices and techniques for decision-making, Stakeholders' points of view, Engineering, Engineering management, Business-organizational management, Framework

1. Introduction

The term technical debt is a metaphor that refers to the consequences of weak software development. Technical debt can grow because of the development process [1], or because of circumstances that are beyond the developers' control [S1]. Some technical debt is inevitable in a world with finite resources [1]. Several studies have highlighted the negative effects of uncontrolled technical debt on software development [S3], [1, 2]. The problem is that the benefits of refactoring software to remove technical debt are largely invisible, sometimes intangible, and usually long term, whereas the costs of refactoring activities are significant and immediate [S4]. Technical debt management consists, precisely, of identifying the sources of the extra costs of software maintenance and determining whether it is profitable to invest efforts into improving a software system [1]. If technical debt is not managed, the company might have to be compelled to invest all its efforts into keeping the system running, rather than increasing the value of the system by adding new capabilities [1], and, thus, damaging the company's profitability and risk the inability to fulfill its strategic goals.

Hence, decision-making is essential in technical debt management. Considerations can include issues such as the acceptable amount of technical debt and the right moment to reduce technical debt. Convincing arguments are required regarding when and why to remove technical debt. To identify the extent to which current technical debt management models and approaches can answer these and similar questions the literature was examined. It turned out that current technical debt management approaches usually address ad-hoc cases or concrete types of technical debt, such as [S4, S5, S6], but few of them analyze

technical debt management from an overall holistic perspective. Using a model that focused on only one issue is too narrow to support decision-making. Therefore, the objective evolved towards identifying the entire set of elements that have been used to manage technical debt. **Elements** are understood as the concepts used to implement technical debt management, regardless of their nature. Without a minimum knowledge of these *elements* it would not be possible to define the models or techniques required for general technical debt management decision making. Furthermore, it is also necessary to know about the support of current methods for the elements. In this way, it could be possible to discover potential gaps in the current methods. Rigor and relevance for industry of the current methods and techniques used to manage technical debt are also relevant.

A study focused on the identification and classification of the entire set of *elements* that have been used to manage technical debt, i.e., *Elements*, as defined within this paper, has not been addressed before in literature. However, some of the systematic studies found are related to this one and have also identified technical debt or technical debt management concepts; it is necessary to clarify how they differ from *Elements* in objective or scope. Tom et al. [1] described concepts related to the technical debt phenomenon: dimensions, which are the different types of technical debt; attributes, which are characteristics of technical debt; precedents, which are causes of technical debt; and outcomes, which are consequences of technical debt. However, they do not describe how to manage technical debt. Li et al. [3] identified types of technical debt. They also identified some notions related to technical debt, used to describe or explain the technical debt phenomenon, and, additionally, technical debt management activities, that represent the steps that have to be performed to manage technical debt. The elements, as defined within this paper, conceptually different from activities, are used during the execution of activities as inputs, outputs, or mechanisms. Finally, they identified already implemented tools for technical debt management activities. Some of the types of technical debt were identified in Alves et al. [4] and Alves et al. [5]. In [5] some technical debt indicators of the existence of technical debt, and some strategies for technical debt management

are also identified. Finally, Ampatzoglou et al. [6] identified financial terms related technical debt management, that is, metaphors used in technical debt management that facilitate the understanding of technical debt and its management. They also identified specific financial approaches used in technical debt management and software engineering terms used in these approaches, that is, the software engineering terms (e.g., design or refactoring) used in the financial approaches for technical debt management.

A final aspect is that decisions are always made according to stakeholders' best interests. Therefore, it is not enough to identify the entire set of elements, but it is also necessary to address the different stakeholders' points of view about technical debt management.

Considering all the above issues, this study takes into account the current knowledge of the technical debt management community and concern the following: 1) the elements that have to be considered in managing technical debt when making decisions in a software project; 2) which elements should be considered from the various stakeholders' points of view; and 3) the current support in industrial environments for the identified elements in the form of the methods and techniques used in technical debt management decision-making. The research method applied by this study is the systematic literature mapping of the available literature. The findings are then synthesized to define and classify the elements.

The contributions of this study are as follows:

- The identification, definition, classification, and analysis of the elements that have been considered in the literature to manage technical debt in making decisions about software projects. Because of the objective of classification, a taxonomy of the elements according to their use in technical debt management is provided.
- The elements are classified from the different stakeholders' points of view.
- The assessment of the current support for the elements in industrial environments by means of methods and techniques.

The definition and classification of the elements, as well as the results of the analysis could be deployed to help develop models that are adapted to the specific stakeholder's interests. The purpose of these models will be to assist in the decision-making involved in technical debt management.

The structure of the paper is as follows: Section 2 will discuss previous related studies. Section 3 will present the methodology and the processes used in this study. Section 4 will describe the identified elements. Section 5 will present the mapping of the elements according to the stakeholders' points of view. Section 6 will present the results of evaluating the rigor and relevance of the analyzed papers. Section 7 will present a retrospective of the findings presented in this study. Finally, threats to validity, conclusions and recommendations for future work will be presented in Section 8 and Section 9, respectively.

This study is an extension of a previous study that was published in the Seventh International Workshop on Managing Technical Debt [7]. In the current study new content suggested by the reviewers of the workshop has been included, whereas previously it was excluded because of space limitations. The analysis of the elements has been extended by introducing the industrial perspective and considering rigor and relevance. This perspective has helped identify the current use in industry. Consequently, new findings and conclusions have been obtained.

2. Related Work

As mentioned in the introduction, there are several previous reviews of the literature on technical debt. Tom et al. [1] performed a multivocal literature review to create a taxonomy of the phenomenon of technical debt. Alves et al. [4] performed a systematic mapping to create an ontology of technical debt. Li et al. [3] performed a systematic mapping identifying activities to perform in technical debt management. Alves et al. [5] performed a systematic mapping to identify types of technical debt and methods used for technical debt management. Finally, Ampatzoglou et al. [6] performed a systematic literature review

to study the financial aspects of technical debt. In the literature, several studies have addressed the phenomenon of technical debt [1, 4, 5, 6], the methods used for technical debt management [5], and the activities used in performing technical debt management [3]. Our perspective is different from these previous studies since we are specifically interested in the elements required to manage technical debt. Therefore, these other studies have been used as sources for identifying these elements. Some specific differences between the findings of our study and what can be found in literature are addressed in Section 7. Additionally, in this work, the rigor and relevance for industry of the current techniques for technical debt management have been studied.

Finally, Falessi et al. [S7] identified the requirements for the tools utilized to manage technical debt. We have extended their work by incorporating the ways in which other authors have considered these requirements as well as their current use in industry.

3. Methodology and Research Process

To identify the elements of technical debt management that have been addressed in the current literature, this study utilized systematic mapping. The systematic mapping was performed by following Petersen et al.'s guide [8]. Systematic mapping studies are designed to provide a broad overview of a research area and are consequently appropriate for the goal of this research, which is, to identify the elements that must be taken into account to manage technical debt efficiently. All the steps in the process are shown in Figure 1 and described in the following subsections.

3.1. Research Questions

RQ1: What elements have to be considered when making decisions concerning technical debt management in software projects?

RQ1.1: In the literature, what elements have been suggested to manage technical debt?

Figure 1: Systematic mapping process (adapted from [8])

RQ1.2: What elements have been used in the methods and techniques proposed to manage technical debt?

RQ2: What elements are considered from the various stakeholders' points of view?

RQ3: What is the current support in industrial environments for elements required in decision making in the form of methods and techniques?

To answer these research questions, this study focuses on publications about technical debt management.

3.2. Conduct Search

The reported study focused on methods, techniques, and suggestions for technical debt management. After some trials with the strings “debt” and “technical debt”, the second one was select because it returned all the previously known papers, and the string “debt” led to too many false positives. However, to reduce the risk of leaving out relevant papers, the search was complemented with the snowballing technique (see Section 3.4). Therefore, an automatic search method using the term “*technical debt*” was used to search for papers about technical debt in the following digital repositories: IEEE Xplore, ACM, Scopus, ScienceDirect, Web of Science, and SpringerLink. *Full text* search was used

when this option was available (IEEE Xplore and SpringerLink), as well as for searching in metadata in other cases (ACM, Scopus, ScienceDirect, and Web of Science). The search included all the papers published until and including 2015. The total number of articles obtained (including duplicates) was 971.

3.3. Screening of Papers

The selection process consisted of two levels. In the first one, papers that provided enough information in their title and abstract to be excluded were eliminated. In the second one, the full text of the papers was analyzed. To select the articles, the following inclusion and exclusion criteria were used:

- To be included, the paper had to describe parts, activities, tasks, elements, or considerations for technical debt management.
- To be included, the paper had to be published in a journal, conference proceedings, or workshop proceedings. Only book chapters referenced by another included study were included.
- Papers that failed to address technical debt management in detail were excluded.
- Papers published as abstracts, call for workshops, tutorials, talks, or seminars were excluded.

The selection process was performed separately by two of the authors and discrepancies were resolved through agreement among all the authors. At the end of this step, the number of selected relevant papers was 61.

3.4. Snowballing

In order to include all relevant papers, the bibliography of each included paper was screened by using the snowballing technique. Snowballing refers to using the reference list of a paper to identify additional papers [9]. This technique is commonly used in systematic mapping studies [9, 3, 10]. Each new identified relevant paper starts a new iteration in the snowballing process. When no more relevant papers are found, the process ends.

At the end of this step the number of selected relevant papers was 63, that is, two papers were added. The two papers were identified in the first iteration of the snowballing process. The second iteration did not lead to more articles, and consequently the snowballing process ended. The list of the selected papers is included in the bibliography section at the end of this paper.

3.5. Comparison Between the Number of Papers Selected in this and Previous Studies

Due to the existence of the mentioned previous literature review studies (see Section 2), it is interesting to highlight and analyze the difference in the number of selected papers that exists between this and the other studies. In the present study 63 papers have been selected. Tom et al. [1] selected 19 papers and 35 web blogs and articles. This difference can be explained because Tom et al.'s study checked publications before 2011 whereas the present study included studies until the end of 2015. Even when Tom et al. selected all the papers about any aspect of technical debt, and the present study only selected papers about technical debt management, technical debt management has focused a lot of attention in the recent years, what helps explain numbers. Li et al. [3] selected 94 papers until the end of 2013, a higher number of papers than in the present study. This is likely due to the different criteria used to select the papers, and how the the community is focusing ever more on technical debt and on technical debt management. In fact, whereas Li et al. selected papers about any aspect of technical debt, in the present study only papers that explicitly address with technical debt management (see the first selection criteria in Section 3.3) were selected. Alves et al. [5] selected 100 papers in a study that checked publications until the end of 2015 as the present study does. Again, as in Li et al.'s work [3], the main difference is the selection criteria. Ampatzoglou et al. [6] selected 69 papers. In this case the numbers of selected papers are quite similar. The main difference is that Ampatzoglou et al.'s work checked until the end of 2013. Therefore, in relative terms, they selected more studies than the present study. This is due to some papers that are relevant from a financial point of view, for example, papers explaining the similarities of technical debt and financial debt

are not focused on technical debt management and hence, they were excluded in the present study.

3.6. Keywording

The process used to identify the elements required for technical debt management was adapted from the “keywording” described by Petersen et al. [8]. Figure 2 shows the steps used in this process. The main difference from [8] is that in the present study, the process started using the *full text* of the selected papers instead of the abstracts. The reason for this decision was that the entire description of the technical debt management activity was not usually present in the abstracts of the papers. The process was iterative. After the first classification was obtained, it was refined by contrasting the various elements found. The classification scheme then was updated by binding similar elements. This additional step was necessary because it was found that the same concept (or very similar concepts) was used in different contexts or with different names. The criteria used to extract the elements corresponded to the identification of requirements, estimations, analysis, and activities (including inputs and outputs) in technical debt management that were used or suggested in the selected papers.

3.7. Synthesis

After the keywording step, a synthesis step was performed to refine the classification scheme previously obtained. In this synthesis, the reasons for and intentions to use each element were also extracted from the original papers. A method analogous to the constant comparison used in qualitative data analysis [11] was used to find commonalities in the intentions of using each element. Following this process, the definitions of the elements were created. Additionally, three types of elements were identified. The types of elements defined are explained in Section 4.2. As discussed in Section 7, this synthesis provided a schema that can be considered a framework for the elements for technical debt decision making in practice.

Figure 2: Keywording process (adapted from [8])

3.8. Rigor and Relevance Assessment

The scientific rigor and the industrial relevance of the analyzed papers were assessed. The goal of this step was to know how the identified elements were supported in the currently available methods and techniques used in technical debt management.

The quality assessment of the reviewed papers is not covered by the systematic mapping approach [8]. We used the method proposed by Ivarsson and Gorschek [12] which was previously used with systematic mapping studies in the software engineering domain [10]. The model provides a set of rubrics to measure rigor and relevance for industry. Rigor refers to the precision or exactness of the research method used and how the study is presented. The model in [12] defines three aspects used to measure rigor: context described, study design described, and validity discussed. Each aspect is scored by 0, 0.5, or 1. Consequently, the total rigor score of a paper will be between 0 and 3: 0 is the worst score, and 3 the best score. A detailed explanation about the criteria used to assign each score is provided in [12].

Relevance refers to the realism of the environment in which the results are obtained and the degree to which the research method facilitates the transference of results to practitioners. The model defines four aspects that are scored by 0 or 1. Therefore, the total relevance score of a paper is between 0 and 4: 0 is the worst score, and 4 is the best score. A detailed explanation about the criteria used to assign each score is provided in [12].

3.9. Mapping Process

The papers included in the current review were classified using the various elements found (see Section 4) and the points of view (see Section 5). Also identified were papers showing the use of elements in methods or techniques for technical debt management or in suggestions for using them. It was found that each paper included several elements and/or points of view.

4. Elements of Technical Debt Management

This section addresses the first research question *What elements have to be considered when making decisions concerning technical debt management in software projects?* Table 1 shows the sources that either used or suggested the usage of specific elements for technical debt management. The elements were identified following the steps described in Section 3.6, and they were defined by following the steps defined in Section 3.7. Section 4.1 includes a detailed explanation of the identified elements and Section 4.2 shows a categorization of the elements.

4.1. Elements

In this section the elements are described.

4.1.1. E1 Technical debt items

To manage technical debt properly, it is necessary to know the sources that originated technical debt in the system. This element is basic because if it is not known that a problem exists, it is not possible to manage it. In fact, to identify technical debt items is usually the first step in managing technical debt [S8, S9].

The identification of technical debt items can include establishing a list of the bad practices that create debt [S5, S10] and identifying the potential kinds of technical debt from the sources of technical debt [S7], as well as determining the part of the system that must be refactored [S4, S11]. Therefore, there are many potential sources of technical debt at any time in any system [S1].

4.1.2. *E2 Principal*

In decision-making, it is essential to know the cost that is required to remove a technical debt item by changing the software. Many authors use the term principal to refer to this cost. The principal of a technical debt item is the cost to be paid to eliminate the item [1]. Estimating the principal was a basis of technical debt management according to most authors (see Table 1). It is possible to estimate the principal as a function of three variables: the “should-fix” items with violations, the hours needed to fix each violation, and the cost of labor [S8].

An important consideration is that technical debt is *context* dependent [S12]. Therefore, solving a weakness can cost more or less depending on the project or even the subsystem within a project [S12]. However, because estimating the principal in terms of single values is difficult, practitioners seem to think in ranges of values or best-case, worst-case, and most probable scenarios, rather than single values [S7, S4].

4.1.3. *E3 Interest*

In the decision-making process, to have complete information about technical debt, it is necessary to know not only the principal of the technical debt items but also the cost of not removing them. This cost is usually termed the interest. The interest is the cost to be paid over time if a technical debt item is not eliminated. This cost can include the extra cost of modifying a component that needs refactoring compared to the cost of modifying it after refactoring [S1]. Similar to the principal, estimating the interest was deemed fundamental by most authors (see Table 1). The interest can be seen in various ways. It is possible that depending on the project and its context, the interest could be

non-linear, or it could have limits, maximums, or minimums [S7]. Similar to the principal, an important consideration is that technical debt is *context dependent* [S12]. Hence, the same detected weakness can imply more or fewer future costs depending on the project or even the subsystem within the project [S12].

4.1.4. E4 Interest probability

The interest has a probability of being paid [S9, S7, S1]. That is, it is necessary to know such probability in order to have a real estimation of the interest. This uncertainty exists because the interest must only be paid under some scenarios [S13]. The probability of paying the interest will depend on the probability of the occurrence of future events [S7, S14, S15]. For example, if a technical debt item implies that more effort is required in maintenance activities but that the item will not be changed over time, the interest does not need to be paid. Hence, in this example, the probability of changing this item will parallel the probability of interest. Therefore, the interest is determined by a set of relevant change scenarios [S14], that is, the probable changes that will occur in the future. These scenarios will include, for example, the addition of new functionality, changes in the non-functional requirements, the solutions to problems or bugs in the system.

4.1.5. E5 Technical debt impact

The complete technical debt estimation consists of principal and interest estimations, including the interest probability. To manage technical debt, it is necessary to use models in which items are prioritized based on their interests and principals [S9]. The aim of ranking technical debt items is to identify which items should be resolved first, depending on the business's goals and preferences [S14]. Decisions about technical debt should be made in terms of cost-benefit analysis [S9, S7]. Thus, cost-benefit estimations are necessary to make decisions about removing technical debt [S4]. The *business value* of conducting any activity lies in the difference between the cash flow stream of performing the activity and the cash flow stream of not performing it [13]. That is, a technical debt item must be removed when doing so is profitable. The

basis of cost-benefit analysis is to identify the items with the highest amount of technical debt and consider the cost of fixing them [S9]. Cost-benefit analysis usually includes several variables. The most obvious are the principal (the cost paid to remove a technical debt item) and the interest (the cost to be avoided by removing a technical debt item). Additional costs items could be considered, such as time-to-market penalizations and quality issues. Benefits, such as quality improvements and customer satisfaction could also be considered. Additional information that could be used includes the number of features that a software release contains, the time required to deliver the release, and the technical debt that is generated because of forced quick development [S16].

The outcome of technical debt management should be reviewed in terms of its economic consequences [S7] by taking into account business considerations [S2]. That is, technical debt must be quantified [S3]. It is not always easy to express technical debt using economic data. However, without expressing the economic consequences, it is difficult to quantify the effects of executing or not executing a decision. Tom et al. [1] classified technical debt costs into four types: morale, productivity, quality, and risk [1]. However, one problem is that the financial effects of technical debt are not always direct [1]. For example, technical debt can cause low morale in the development team resulting in systemic problems, such as developer turnover [1, 2]. Another issue is that it is necessary to balance rigor with the usability of the estimation method. A very complex and rigorous mathematical estimation model could be highly precise in its prediction and yield highly reliable results, but it could be unusable in practice because it is overly complex when it is adapted to real, large projects [S7]. Finally, some special situations must be considered in cost-benefit analysis. For example, when a system is “retired”, its technical debt is removed [1]. Another issue is that organizations probably do not have adequate resources to fix all the identified technical debt items [S5]. In summary, a complete cost-benefit analysis must take into account all factors. That is, it must consider not only the principal and interest but also the project constraints, including the deadline, budget, and effects [S5, S14]. Because of this complexity, techniques that help visualize the

estimated effects of the technical debt (not technical debt itself) could greatly help decision-making.

4.1.6. *E6 Automated means*

Big projects can generate high volumes of data. Therefore, because of feasibility, it is necessary to obtain estimates automatically to manage technical debt which avoids the negative effect of intrusiveness in the normal development process [S3]. However, the collection of measures is an extra step for developers, who are already overloaded, and might compromise the success of technical debt management [S12]. Hence, the source code, a project's revision history, a project's bug history, and other similar data sources can be mined using automatic tools to obtain the information required to estimate technical debt automatically [S9, S7] and to propose potential items for refactoring [S4].

4.1.7. *E7 Expert opinion*

Expert opinions about the system are required to manage technical debt because they provide knowledge that cannot be obtained from available software information. Together with automated estimates to manage technical debt, the opinions of the people that know the system deeply are required [S9, S7]. This information can include new contracts to be signed, expected changes to be made, or new technologies to be adopted [S17], helping the manager to provide and apply information regarding issues such as uncertainty about the measures and judgments, the system's external context, and the knowledge of experts (project managers, architects, etc.) [S4]. Finally, the goal of technical debt management methods and tools is to provide the necessary information to human decision-makers [S12].

4.1.8. *E8 Scenario analysis*

In decision-making, it is necessary to estimate the consequences of the decisions made about the system. Therefore, managing technical debt includes defining and analyzing multiple potential scenarios [S7]. By analyzing scenarios, managers can acquire information about the effects of the technical debt if certain events occur in the future, which is discussed in Section 4.1.4. Various

possible implementations can be analyzed to determine which one is the best. In technical debt management, the goals of scenario analysis are as follows.

- To set targets for debt and specify the level that is acceptable for the project or organization [S5].
- To identify the effects of non-fixed technical debt issues on multiple releases. This is performed by using change scenarios [S18, S14, S17]. Change scenarios represent the probable future changes that the system will accommodate, and they are used to determine the amount of interest to be paid because of technical debt [S15].
- To identify when it is profitable either to implement new functionality early (taking more technical debt because of the quick release or when it is profitable to release functionality slowly but with less technical debt [S16]. The scenarios to be analyzed can include the various paths followed to deliver features in the release planning [S19], that is, to analyze how much effort to invest in either refactoring, architecture design, or the addition of new features [S1] by researching various release scenarios [S20].
- With this information, the manager will be able to choose which decisions to implement with the highest probability, and he or she will know the other possible alternatives and the system's technical debt evolutions. The output of the technical debt management must be in the form of possible scenarios as well as the probability of their economic consequences [S7].
- It is necessary to sketch and assess potential alternatives to the benefits and costs to support choosing the most appropriate for handling the technical debt [S2]. This process includes testing various possible scenarios to analyze the effects of removing some technical debt items, that is, performing a “what-if” analysis [S4, S12].

4.1.9. E9 Time-to-market

One important constraint to consider is the time-to-market [S7]. This element includes the resources and constraints involved in achieving a project goal

on time. The solutions to be implemented could be useless if they cannot be implemented within a certain time. In some situations, it could be necessary to release a product by the deadline without all its expected characteristics. In certain environments, being the first to market is vital to obtain customers. In this situation, long-term software problems are not important because they are not visible to the product's customers, and these problems are not very important to the software company. In the long term, the problems will be relevant only if the product or the sponsors obtain customers in the short term [1]. To consider fully the costs and benefits of incurring technical debt, it is necessary to take into account the *release planning* of the product under analysis [S19]. In summary, a tradeoff between release characteristics and technical debt must be made to manage technical debt [S16].

4.1.10. E10 When to implement decisions

Managing technical debt requires making the decision either to pay the principal of technical debt items or to continue to pay the interest on such items. This element is greatly influenced by the constraints on and the availability of the development team's resources. It is necessary to know when to implement this decision [S9, S7]. Some decisions may include when to implement a feature of a product, or when to refactor some part of the system to improve some of its qualities [S19]. Hence, such decisions affect the release dates and planning [S20].

4.1.11. E11 Technical debt evolution

To know how technical debt affects a system it is not enough to have a snapshot of the technical debt in the system. In order to contrast the technical debt of the system with changes in the context, it is necessary to know how the technical debt evolves in the system. Hence, it is necessary to track technical debt over time [S3]. This tracking requires methods to determine the level of technical debt and its evolution [S1]. Monitoring technical debt consists of keeping track of changes in the costs and benefits of unresolved technical debt items [S14]. By monitoring technical debt frequently, it is possible to react quickly [S5]. Thus, monitoring the evolution of the economic consequences of

technical debt [S7] is important. To determine how the system will perform in the future, it is necessary to consider the time-frame required for such an analysis [S17]. Because technical debt implies a cost over time, the time-frame will structure the analysis and enable a cost-benefit analysis. One threshold for the time-frame could be the date estimated for retiring the software [S16]. Other potential time-frames could be obtained from the project release plan [S21] or from the project roadmap.

4.1.12. E12 Technical debt visualization

Managing technical debt without the visibility of the technical debt items, and the software artifacts (files, modules, packages, etc.) in which the items are accumulated is not possible. Therefore, having the means to see how technical debt affects the system or the development process is highly recommended [S22]. Defining a visual language for the entire organization allows for fast and transparent communication between people and entities [S5]. Hence, it is possible to determine the relative effects of technical debt with regard to other activities [S3]. However, the visualization technique must have the ability to summarize the information required for high-level analysis, and it must include the possibility of analyzing technical debt at lower levels, such as at the subsystem or component level [S12]. The visualization of technical debt is especially important in architectural technical debt [S23]. This kind of technical debt usually implies several source code artifacts, such as classes and configuration files. Using mechanisms to determine how these artifacts are grouped helps to clarify the distribution of technical debt throughout the system.

4.2. Grouping of elements according to their use in technical debt management

Figure 3 shows the grouping of the identified elements according to their use in technical debt management, which were found in the literature after applying the synthesis step describe in Section 3.7. In practice, a **taxonomy** of the elements has been obtained. The analysis showed that the elements could be classified into three main groups: basic decision-making factors, cost estimation techniques, and practices and techniques for decision-making. These

Figure 3: Taxonomy of the identified elements for technical debt management. Elements are grouped according to its use in technical debt management

are explained in the following paragraphs. *T1 basic decision-making factors*, are elements that represent the necessary information about the system technical debt. That is, these elements are information that it is needed to make decisions about managing technical debt. Therefore, these elements are mainly focused on the identification and measurement of technical debt. *T2 cost estimation techniques*, are elements focused on how a technical debt management model should be implemented. The main difference from the basic decision-making factors is based on the degree of human intervention. Therefore, models can be implemented using automatic tools, manual processes or a mix of both. Finally, *T3 practices and techniques for decision-making*, are elements focused on the considerations that must be taken into account, in addition to the technical debt estimations, to manage technical debt. Elements of this type indicate requirements to be taken into account in addition to the basic decision-making factors. These elements draw attention to the fact that it is not enough to simply identify and measure technical debt. It is also necessary to integrate technical debt management into the project management process as one of its activities.

	Elements	Sources	Count
T1 Basic decision-making factors	E1 Technical debt items	S1 S4 S5 S7 S24 S8 S25 S21 S26 S27 S13 S28 S14 S29 S30 S10 S31 S32 S33 S34 S35 S15 S36 S37 S9 S12 S38 S39 S40 S41 S42 S43 S11 S18 S44 S45 S46 S47 S48 S49 S50 S51 S52 S53 S54 S55	46
	E2 Principal	S1 S4 S7 S56 S24 S8 S17 S57 S25 S21 S26 S27 S14 S58 S31 S32 S15 S36 S9 S12 S59 S42 S43 S11 S18 S44 S45 S46 S47 S49 S50 S51 S52 S53 S54	35
	E3 Interest	S1 S4 S7 S56 S24 S8 S17 S57 S25 S21 S26 S27 S14 S31 S32 S15 S36 S9 S12 S59 S42 S43 S11 S18 S44 S45 S46 S47 S49 S50 S51 S52 S53 S54	34
	E4 Interest probability	S1 S4 S7 S17 S25 S21 S26 S27 S14 S15 S36 S9 S12 S46 S48 S49 S50 S52	18
	E5 Technical debt impact	S1 S2 S3 S4 S5 S6 S7 S60 S56 S24 S8 S17 S57 S25 S21 S26 S27 S28 S14 S30 S10 S61 S31 S19 S32 S16 S33 S34 S35 S36 S9 S12 S39 S42 S11 S62 S18 S44 S46 S47 S49 S50 S51 S52 S53	45
T2 Cost estimation techniques	E6 Auto-mated means	S3 S4 S5 S7 S8 S28 S10 S58 S33 S35 S9 S12 S38 S39 S40 S11 S62 S44 S45 S46 S48 S50 S51 S53 S54 S55	26
	E7 Expert opinion	S4 S7 S17 S26 S35 S36 S9 S12 S11 S18 S45 S48 S49 S50 S51 S52	16
T3 Practices and techniques for decision-making	E8 Scenario analysis	S1 S2 S4 S5 S7 S56 S17 S20 S27 S28 S14 S19 S16 S33 S15 S36 S12 S62 S18 S50	20
	E9 Time-to-market	S7 S21 S61 S31 S19 S16 S15 S36 S41	9
	E10 When to implement decisions	S4 S7 S56 S63 S25 S26 S20 S13 S19 S33 S9 S18 S44 S47 S50 S51 S52	17

E11	Technical debt evolution	S1 S3 S5 S7 S56 S21 S26 S28 S14 S10 S16 S33 S38 S45 S47 S49 S51 S55	18
E12	Technical debt visualization	S3 S4 S5 S22 S23 S28 S37 S12 S39 S11 S45 S47 S48 S49 S50 S51 S53 S55	18

Table 1: Elements of technical debt management identified in the literature. Every element identified is included in the column *Elements*. The references in which it was identified are shown in the column *Sources*. The number of references in which it was identified is shown in the column *Count*. The column on the left classifies the elements according to their use in technical debt management (see Section 4.2).

5. Technical Debt Management Elements from the Stakeholders' Points of View

In this section, the second research question is addressed: *What elements are considered from the various stakeholders' points of view?* Although the findings showed that various stakeholders are involved in technical debt management, the most appropriate classification system to use was not evident. Clements et al. proposed, in [14], three stakeholders' points of view, namely (software) engineering, technical management, and organizational management, which were used in the first version of the work. Recently, in SWEBOK V3 [15], the terms engineering management, and business organizational were used. This usage reflects an evolution in the understanding of how software product development takes place. Therefore, in this study, the final list of stakeholders' points of view was as follows: (software) *P1 engineering*, *P2 engineering management*, and *P3 business organizational management*. *P1 engineering* includes concerns about processes such as software design and software construction, as well as, software architecture for some members of the community. *P2 engineering management*

Figure 4: Mapping of the elements in the stakeholders' points of view. All the papers identified in this study are considered in the mapping. Mapping of the elements to support decision making in managing technical debt versus the engineering, engineering management, and business organizational management points of view. Each of the selected papers can include several elements and can be mapped onto more than one point of view. In each cell, the number in the center is the number of papers that identify an element from a specific stakeholders' point of view. The upper left percentage is the percentage of papers that identify the element (column) as specific to a point of view of (row), and the lower right percentage is the percentage of papers with the specific point of view (row) that identify the element (column). The first column shows the summary of the papers per stakeholder point of view, while in the bottom of the figure, there are summaries of papers per element and per type of element.

is focused on process planning and monitoring, including measurement. Finally, *P3 business organizational management* is focused on organizational goals, business strategy, time horizons, risk factors, financial constraints, and tax considerations. These points of view are aligned with the stakeholders identified by Yli-Huumo et al. [S47] in technical debt management: *development team, software architect (P1 engineering)*; *team manager (P2 engineering management)*; and *business stakeholder (P3 business organizational management)*.

To analyze the elements of technical debt management from the stakeholders' points of view mapping was conducted. Figure 4 shows the mapping of the identified elements of technical debt management (see Section 4) and the stakeholders' points of view, which are described above.

While performing the mapping, the authors noticed that whereas some papers introduced a technical debt management element, and proposed a method or technique for technical debt management, others simply introduced an element. Therefore, two different mappings were performed, which are presented in the following sections.

5.1. Technical Debt Management Elements from the Stakeholders' Points of View: Papers that Simply Introduced Elements

In the case of the papers that simply introduced elements, the analysis revealed (see Figure 4) that most were focused on the *T1 basic decision-making factors*, that is, on obtaining the necessary information about the system's technical debt. In this case, the preponderant points of view were *P1 engineering* (e.g., the creation of source code, designing, architecting, or testing) and *P2 engineering management* (e.g., planning, or product quality management).

The papers included in *P1 engineering* and *P2 engineering management* strongly highlighted all the identified technical debt management elements. Conversely, this was not the case for *P3 business organizational management*, apart from *E9 time-to-market*, *E11 Technical debt evolution*, and *E6 automated means* elements. This finding indicated a business requirement for technical debt management, which could include the following: to quantify the impact of technical debt; to estimate technical debt automatically to avoid extra effort in

the developments process; to know the effects of technical debt in the delivery capacity of the team putting at risk the ability to meet the projects' deadlines; and to control the evolution of technical debt over time. This finding, however, should be assessed in future studies.

Concerning the *E9 time-to-market* element, an interesting finding was that it is hardly ever taken into account by the current technical debt management tools and methods as discussed in [16]. This is an interesting paradox and a serious issue: whereas *E9 time-to-market* is one of the least referenced, used, and suggested *T3 practices and techniques in decision-making*, it is probably the most referenced cause of technical debt and according to [1] one of its most relevant antecedents. Therefore, managing technical debt without considering *E9 time-to-market* could lead to wrong decisions that could affect important deadlines in a project.

Most of the studies analyzed focused on *T1 basic decision-making factors*. This finding may indicate that technical debt management is still in an initial phase. It also means that data estimation techniques, and metrics for the technical information normally extracted from source code, repositories, or tracking systems have not yet been developed, or, at least, they have not yet been made available to the software engineering community. Whereas *T1 basic decision-making factors* seem to be conceptually close to the project *P1 engineering* activities, *T3 practices and techniques for decision-making* are not far from the decision making process and are therefore closer to *P2 engineering management* and *P3 business organizational management* than to *P1 engineering*. Whereas *P1 engineering* is focused on finding project's technical problems, *P2 engineering management* and *P3 business organizational management* extract information from sources such as strategic decisions about the architecture, product release dates, new contract signatures, budgets, or technologies provided by partners, in the areas of management and strategy. It is worthwhile to highlight that *E5 Technical debt impact* was the only highly referred element, regardless of the point of view that was considered. Therefore, *E5 Technical debt impact* was found to link the three points of view in addressing technical debt manage-

ment concerns. This finding suggests that quantifying the effects of technical debt could form an excellent *communication channel* among the different stakeholders in a project. Moreover, this finding supports the previous conclusion about the importance of the *E5 Technical debt impact* element (Section 4.1.5).

Finally, the elements of *E6 automated means* and *E7 expert opinion* fit the type *T2 cost estimation techniques*. These two elements are complementary: *E6 automated means* element refers to extracting the required data without disturbing the normal activity of developers, whereas *E7 expert opinion* refers to using experts to provide the information that cannot be automatically extracted and to make decisions based on estimations.

5.2. Technical Debt Management Elements from the Stakeholders' Points of View: Papers that Introduce Elements and Propose Methods

The same analysis described in section 5.1 was performed on papers that introduced one or several elements of technical debt management, and proposed methods or techniques to manage technical debt. Papers that simply introduced elements of technical debt management were excluded. Figure 5 shows this second mapping. The comparison of Figure 4 and Figure 5, shows an obvious difference in the amount of papers on each element and point of view. As long as the set of papers used in Figure 5 was a subset of the set of papers used in Figure 4 the obtained difference in the number of papers could be expected. Nonetheless, the percentage of elements introduced with respect to the three points of view were proportional to those obtained in the entire selection of papers (see subsection 5.1).

This finding allowed us to conclude that the identified elements were not simply suggestions made by the authors but a set of elements used by those authors to define concrete techniques for technical debt management. The second mapping was also useful to identify the coverage of the elements. It therefore was possible to determine some gaps in the current methods regarding elements that were not addressed to manage technical debt. The main shortcoming was the insufficient support shown for *E4 Interest probability*, *E7 Expert opinion*,

P3 Business organizational management	15	73.33%	46.67%	46.67%	26.67%	93.33%	66.67%	20.00%	53.33%	13.33%	20.00%	46.67%	40.00%
		11	7	7	0	14	10	9	8	2	3	7	6
P2 Engineering Management	23	39.29%	35.00%	33.33%	36.36%	50.00%	58.82%	27.27%	57.14%	50.00%	27.27%	58.33%	50.00%
		16	15	15	9	21	7	7	12	9	9	8	5
P1 Engineering	33	57.14%	75.00%	71.43%	81.82%	75.00%	41.18%	63.64%	85.71%	100%	81.82%	66.67%	41.67%
		26	19	20	10	23	15	10	10	9	11	11	11
		28	20	21	11	28	17	11	14	4	11	12	12
		E1 Technical debt items	E2 Principal	E3 Interest	E4 Interest probability	E5 Technical debt impact	E6 Automated means	E7 Expert opinion	E8 Scenario analysis	E9 Time-to-market	E10 When to implement decisions	E11 Technical debt evolution	E12 Technical debt visualization
		36					23		30				
		T1 Basic decision-making factors					T2 Cost estimation techniques		T3 Practices and techniques for decision-making				

Figure 5: Mapping of the elements in the stakeholders' points of view. Exclusively the papers that defined methods for technical debt management were considered for this mapping. Mapping of elements to support decision making in managing technical debt versus the engineering, engineering management, and business organizational management points of view. Each selected paper can include several elements and can be mapped onto more than one point of view. In each cell, the number in the center is the number of papers that identify an element from a specific stakeholder's point of view. The upper left percentage is the percentage of papers that identify the element (column) as specific to a point of view of (row), and the lower right percentage is the percentage of papers with the specific point of view (row) that identify the element (column). The first column shows the summary of the papers per stakeholder point of view, while in the bottom of the figure, there are summaries of papers per element and per type of element.

E9 time-to-market, E10 When to implement decisions, E11 Technical debt evolution, and E12 Technical debt visualization. Based on this finding, it can be concluded that additional methods or improvements to the currently available methods are required to support these elements.

6. Technical Debt Management in the Industrial Environment

The method described in Section 3.8 was applied to analyze the extent to which the current techniques used for technical debt management are relevant in an industrial environment. This method provides two scores to quantify the rigor and the relevance of the studies from an industrial perspective. The method used to score rigor and relevance (see Section 3.8) focused on measuring

Figure 6: Mapping of selected papers with respect to relevance and rigor scores as defined in Section 3.8.

whether the techniques and methods have been validated. Therefore, papers that did not provide data about the validation of technical debt management methods obtained low scores.

The results from the **first analysis** are presented in Figure 6, which shows the scores of the analyzed papers. Most papers did not provide enough details about the methods or techniques to be relevant to the industry. That is, they did not provide enough details to allow an independent company to use the method proposed. However, as shown in Figure 7, in recent years, the trend has been that the average rigor and relevance of the research on technical debt management has increased. Therefore, the technical debt research community is aware of this lack of rigor and relevance in the industry, and it is working to improve the situation.

A **second analysis** was performed. The results are presented in Figure 8, which shows the mapping of the elements and points of view in which papers that presented methods or techniques for technical debt management that had

Figure 7: Temporal evolution of the number of papers with respect to the average relevance and the average rigor scores.

rigor and relevance greater than a threshold were considered. In the present study, the threshold for rigor is equal to 2, that is, only papers with a rigor score of 2 or more were considered. This value was chosen so that only papers with a score at least equal to the half-maximum possible score were included. Following the same criteria, the threshold for relevance was set at 3. Many of the elements counted on having only two or three methods with enough rigor and relevance for industry. The elements *E1 Technical debt items* and *E5 Technical debt impact* were outstanding because they had consistent support that was independent of the point of view.

Therefore, based on these findings, it can be concluded that more methods, or more detailed reports of the current methods, are necessary to support their use by software companies in managing technical debt.

P3 Business organizational management	5	100.00%	60.00%	60.00%	20.00%	80.00%	80.00%	20.00%	20.00%	20.00%	20.00%	20.00%	30.00%	20.00%											
		5	3	3	1	4	4	1	1	1	3	1													
P2 Engineering Management	7	85.71%	41.67%	85.71%	42.86%	85.71%	42.86%	33.33%	100.00%	40.00%	57.14%	42.86%	20.00%	42.86%	33.33%	14.29%	100%	42.86%	33.33%	42.86%	60.00%	0.00%	50.00%		
		6	6	6	3	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		
P1 Engineering	13	92.31%	50.00%	53.85%	85.71%	53.85%	85.71%	23.08%	100%	76.92%	70.00%	53.85%	42.86%	38.46%	60.00%	23.08%	100%	7.69%	100%	23.08%	100%	38.46%	60.00%	15.38%	0.00%
		12	7	7	3	10	7	5	3	1	3	5	2												
		E1 Technical debt items	E2 Principal	E3 Interest	E4 Interest probability	E5 Technical debt impact	E6 Automated means	E7 Expert opinion	E8 Scenario analysis	E9 Time-to-market	E10 When to implement decisions	E11 Technical debt evolution	E12 Technical debt visualization												
		10					7		7																
		T1 Basic decision-making factors					T2 Cost estimation techniques		T3 Practices and techniques for decision-making																

Figure 8: Mapping of the elements to stakeholders' points of view to assess industrial rigor and relevance. Only papers that scored rigor equal to or more than 2, and relevance equal to 3 or more were considered. Mapping of elements to support decision making in managing technical debt versus the engineering, engineering management, and business organizational management points of view. Each selected paper can include several elements and can be mapped onto more than one point of view. In each cell, the number in the center is the number of papers that identify an element from a specific stakeholder's point of view. The upper left percentage is the percentage of papers that identify the element (column) as specific to a point of view of (row), and the lower right percentage is the percentage of papers with the specific point of view (row) that identify the element (column). The first column shows the summary of the papers per stakeholder point of view, while in the bottom of the figure, there are summaries of papers per element and per type of element.

7. Retrospective and Discussion

7.1. Identification and Definition of the Elements

This review study performed a systematic mapping of the current literature on technical debt management. Sixty-three papers were analyzed and 12 elements of managing technical debt were identified. The elements were classified into three types: *T1 basic decision-making factors*; *T2 cost estimation techniques*; and *T3 practices and techniques for decision-making*. This classification allowed us to use a top-down hierarchical approach. Previous contributions to the literature addressed topics that were considered necessary to manage technical debt, but it was not clear how these contributions could fit an overall view of the research. The present review study provided a taxonomy of the elements. As Figure 5 shows, not all the elements were used by the same number of methods used to manage technical debt; therefore, not all the elements had the same level of support.

The elements and types of elements, even when they were related, differed from the activities identified by Li et al. [3]. Although the activities mentioned in [3] represented the steps that have to be performed to manage technical debt, elements are used during activities as inputs, outputs, or mechanisms.

Alves et al. [5] identified several management strategies. These strategies referred to concrete methods or techniques used to manage technical debt. These methods included some the sources in which the elements of the present study were identified. Therefore, the goals of the studies are different. Alves et al. identified the current methods for technical debt management, whereas the present study identified what these methods took into account in order to manage technical debt.

These relationships are interesting for studying how to integrate technical debt management into a software product roadmap similar to that described in [17]. Part of this roadmap is the software development process or the strategic planning. Therefore, a more detailed study of such relationships should be addressed in a future work.

7.2. Stakeholders' Points of View with Regard to the Elements

The identified elements were mapped to three different stakeholders' points of view (see Figure 4). These points of view comprise the activities involved in the software product development enterprise: *P1 engineering*, *P2 engineering management*, and *P3 business organizational management*.

This mapping allowed us to determine how different stakeholders considered the same elements. Other stakeholders sometimes considered different elements but in all cases from a different point of view. Two approaches were used in the mapping. In the first approach, all papers that introduced elements were considered. The second approach considered only studies that introduced elements and used them in developing methods for managing technical debt.

The first finding showed that the business organizational perspective was neglected in the literature, which was a serious obstacle from the point of view of enterprise management. More papers were focused on *P1 engineering* and *P2 engineering management* than on *P3 business organizational management*. Companies make products either to sell to customers or to consume internally. In all cases, products must make sense from a business point of view as well as from a technical point of view.

The second finding was related to an important issue: communication among stakeholders. As described in Section 5.1, the element of *E5 Technical debt impact* was the only element that was highly referred to, regardless of the point of view considered. This finding suggests that quantifying the effects of estimating technical debt could be an excellent *communication channel* among different stakeholders. This opens the issue of how estimating the effects of technical debt could be represented so that stakeholders with different technical backgrounds could understand estimation of the effects and could discuss them effectively with each other.

Another remarkable finding was that *E9 time-to-market* was hardly ever taken into account in the methods used to manage technical debt. However, from the point of view of *P3 business organizational management*, only *E9 time-to-market*, *E11 Technical debt evolution*, and *E6 automated means* elements were

highly considered. This finding is consistent with the results reported in previous studies [5] [16], which present an interesting paradox: whereas *E9 time-to-market* was one of the least referenced elements of technical debt management, it was one of the most relevant antecedents of technical debt [1]. Therefore, the lack of support for time-to-market in the current technical debt management literature could lead to wrong decisions.

Some elements also required more support in the methods used for technical debt management: *E4 interest probability*, *E7 expert opinion*, *E9 time-to-market*, *E10 When to implement decisions*, *E11 Technical debt evolution*, and *E12 Technical debt visualization*.

7.3. Support of Methods for Elements in Industrial Environments

To determine the relevance for industry of the studies with regard to technical debt management each paper was analyzed using the method described in Section 3.8. The results showed that most papers did not introduce methods or techniques in enough detail to be relevant to the industry. Nevertheless, in recent years, the relevance and rigor of the papers about technical debt management has increased. By analyzing the mapping of papers with high rigor and relevance (see Figure 8), it was possible to determine that more methods or more detailed reports of the current methods, are necessary to support software companies in using them to manage technical debt. This finding was particularly apparent in *E12 Technical debt visualization*, which is consistent with the results in Alves et al. [5].

7.4. Baseline for a Framework

The analysis described in sections 4,5, and 6 provides a baseline for defining what could be considered a framework for technical debt management. This framework would consist of a bi-dimensional schema at a high level of granularity. The dimensions would be groups of elements and stakeholders' points of view. At a second level of granularity, the schema could be considered to have six dimensions that corresponded to the three groups of elements (basic decision-making factors, cost estimation techniques, and the practices and techniques

used in decision making) and the three Stakeholders' points of view (engineering, engineering management, and business organizational management). This framework would represent technical debt management as an integral job of the enterprise.

The application of this framework could be twofold: first, it would be possible to define purpose-oriented models for technical debt management (which elements are required for managing technical debt according to different objectives), when dimensions and groups were decided according to specific goals. A second application would be to determine how specific methods (or models) that were built outside framework guidelines could be applied according to their characteristics when they were analyzed according to the framework. The assessment of the types and elements with regard to their relevance for the industry and their evaluation for their use in current methods to support industry (see Section 6) could help to achieve the desired goals. Although the assessment was produced to obtain a view of the current situation, it could be applied in the future research on methods.

The use of this framework could show that some elements have special relevance whereas not always support of methods can be found in literature. This is the case of *E9 time-to-market* (see Section 4.1.9 and Section 5). Scenario analysis is important because technical debt management depends on the context (see Section 4.1.2 and Section 4.1.3).

7.5. *Technical Debt Management Decision Making*

In the present review study, the main finding concerning technical debt management decision making was that it is context dependent (see Section 4.1.2 and Section 4.1.3, and the considerations below in the current section). The consequence of such context dependence is that without a clear solid definition of context and precise estimations of technical debt, estimations of effects are of little use.

From the point of view of decision making in technical debt management, the findings showed that the elements of type *T3 Practices and techniques for*

decision-making were the most relevant. These elements are required to make informed decisions. It is necessary to identify and analyze different possible decisions (*E8 Scenario analysis*) in order to realize the possible consequences of such decisions. Therefore, in making decisions about technical debt it is not enough to have an overall picture of the system's technical debt. Without data about the evolution and trend in the amount of technical debt (*E11 Technical debt evolution*), decisions will be made without enough information. By analyzing the trend in the amount of technical debt, it would be possible to estimate when to invest in removing technical debt before the debt becomes too high to be managed. Both *E8 Scenario analysis* and *E11 Technical debt evolution* imply that a future time-frame is required to perform the analysis.

In practice, the complete picture is more complex because time is also a constraint. This was particularly highlighted by *E9 Time-to-market*, *E10 When to implement decisions*, and *E11 Technical debt evolution*. These elements imply that based on the constraint of time, more or less effort should be made to remove technical debt. For example, a scenario in which a startup is in a race to be the first to release a product in a market is different from a scenario in which a consolidated company has a product that leads the market, and therefore, it is possible to delay a new release to remove technical debt. Therefore, any decision made in technical debt management implies a trade-off between software release characteristics and technical debt removal.

Finally, even with effective tools and practices that identify, measure, and estimate the effects of technical debt, without the means to make such debt visible (*E12 Technical debt visualization*) it would be difficult for companies to understand the real situation of their software products. Visualization techniques are a means of providing fast and transparent communication. If decision-makers do not obtain technical debt information in a format that they understand, they could make wrong decisions.

7.6. Implications for Research

The results of the present study have several implications for the research on the management of technical debt, including the following:

- Further research is needed to integrate automatic data extraction with expert knowledge for technical debt management.
- Advancements are required to make trade-offs between new product characteristics and technical debt removal in new releases.
- Because a time frame is required to perform some analyses (*E8 Scenario analysis* and *E11 Technical debt evolution*), it is necessary to determine the appropriate time frame that considers all the related issues.
- There is a lack of research on the business perspective of technical debt. Further studies in this direction would make valuable contributions to the literature.
- There is a need to determine how to define contexts, and to use them in estimations.
- Visualization techniques are required to estimate technical debt and its effects. These techniques should be designed and built specially to take into account the different backgrounds of stakeholder.
- Research is required on the integration of technical debt management into the roadmap of the software product.

7.7. Implication for Practitioners

The results of the present study have several implications for practitioners, such as the following:

- A guide to study specific cases of technical debt management in relation to the identified elements should be produced. This issue pertains to technology transfer or standardization rather than research.

- Using the defined framework, organizations could create models of technical debt management by using the elements identified in this study.
- Practitioners could identify factors that have to be taken into account in making decisions about technical debt management.
- Organizations could compare their current practices in technical debt management with the elements of the described framework to identify gaps in their technical debt management process.
- Practitioners should consider that technical debt management is relevant to their work whenever they are working within a software product roadmap.

8. Threats to Validity

This section addresses potential biases and the actions taken to minimize their effects. To analyze potential biases in a more systematic way, in this study all the potential biases in systematic reviews were analyzed following the definitions given in [18]. According to these definitions there are three main groups of biases: bias in identifying articles, bias in choosing studies, and bias in obtaining accurate data. In the following subsections, the biases in each group are analyzed.

8.1. *Bias in identifying articles*

Several factors can affect the identification of articles: the criteria of the reviewers and editors of journals or conferences, industry-sponsored research in some areas, place of publication, biased indexing studies in literature databases, inadequate or incomplete searches, articles that are cited more often than others are, and studies that generate multiple publications. In the present study, several different literature databases were used to include the maximum number of sources and to minimize the impact of the above-mentioned biases. Because of the complications involved in identifying that several publications are in fact results of the same study, no action was taken to merge publications. This is

a minor risk because few papers included in this study were authored by the same researcher.

8.2. *Choosing study biases*

The process used to select the papers was conducted following the steps provided by Petersen et al. [8]. To reduce possible bias, the inclusion and exclusion step was completed by two of the authors of the present study. Discrepancies were resolved by agreements brokered among all the authors. The inclusion and exclusion criteria could also be influenced by the personal biases of the authors. The *a priori* definition of the criteria helped to minimize this potential bias.

8.3. *Obtaining accurate data bias*

In a literature review, the poor quality of sources can lead to inaccurate conclusions. To mitigate this potential bias, the selected papers were published in journals, conference proceedings, or workshop proceedings according to a peer-review process. To include all relevant studies about technical debt management, some book-chapters were included. Because these were few in number compared to the other selected papers, the effect on the quality of the results was low. To mitigate the risk of the personal bias of the researcher who analyzed the papers, every paper was analyzed by two researcher. Discrepancies were resolved through agreements among all the authors. The authors had to interpret the papers in order to classify them as being about engineering, engineering management, or business organizational management. This classification could be influenced by personal bias or by the information given in the papers, and this classification should be confirmed in subsequent research, such as by conducting interviews with practitioners.

9. **Conclusions and Future Work**

This paper reports the findings, and conclusions of an analysis of the current literature on technical debt management. The focus of the research was to identify the elements of technical debt management and the methods that support the elements in the industrial environment (rigor and relevance). The

research method used systematic mapping [8] and some synthesis activities. This study analyzed how current approaches supported the identified elements. The study was broad in scope because it considered the perspectives of not only engineering and engineering management but also business organization. This study also showed that the rigor and relevance of the research on technical debt management have increased in recent years.

Based on the mapping conducted in this review study, one conclusion is that it was possible to define a framework. This framework could be used to produce specific decision-making models and methods or to assess existing ones (see Section 7.4). In contrast to the majority of the current approaches to technical debt management, the framework was not constrained by a concrete type of technical debt.

Another important conclusion is that technical debt is *context dependent* (see Section 7.5). This means that the context, which is difficult to define, must be part of the estimation model, includes issues such as the history of the product development, prospects, or time to market.

Introducing the business organizational perspective allowed us to identify that most of the previous studies focused on the elements with engineering and engineering management points of view whereas the business organizational perspective was neglected. Within an industrial or government environment, this is a serious issue. It is important to highlight that while *E9 time-to-market* was one of the least suggested elements, it was probably the most referenced cause of technical debt. According to [1], it was one of its most relevant antecedents. This situation could lead to wrong decisions that could affect important deadlines in a project.

There were some indications that *E5 Technical debt impact* could be effective in allowing communication between the different stakeholders in a project (see Section 7.2). This would require both quantifying and visualizing the effects of technical debt.

In future research, the work will be extended to introduce the relationships among the elements in order to establish how one element is affected by other

elements. Another issue to be addressed in future research is to prioritize the elements that are more relevant than others are with regard to the context of the analyzed project.

The elements identified in the present review study could be used to construct a model for assessing the technical debt of large cyber physical systems, such as those in the smart cities domain. The elements could be used to define models for technical debt management for specific systems with two objectives: to demonstrate how the elements work in practice; to implement specific technical debt management models based on the integration of the tools that are currently available for the management of technical debt.

Acknowledgment

This work was partially sponsored by MESC DPI2013-47450-C2-2-R (Spain).

Bibliography

Selected Papers

- [S1] N. Brown, Y. Cai, Y. Guo, R. Kazman, M. Kim, P. Kruchten, E. Lim, A. MacCormack, R. Nord, I. Ozkaya, R. Sangwan, C. Seaman, K. Sullivan, N. Zazworka, Managing technical debt in software-reliant systems, in: FSE/SDP Workshop, 2010.
- [S2] F. Buschmann, To pay or not to pay technical debt, *Software*, IEEE 28 (6) (2011) 29–31.
- [S3] K. Power, Understanding the impact of technical debt on the capacity and velocity of teams and organizations: Viewing team and organization capacity as a portfolio of real options, in: International Workshop on Managing Technical Debt (MTD), 2013.
- [S4] Y. Cai, R. Kazman, C. V. Silva, L. Xiao, H.-M. Chen, Chapter 6 - a decision-support system approach to economics-driven modularity evaluation, in: *Economics-Driven Software Architecture*, 2014.

- [S5] J. Letouzey, M. Ilkiewicz, Managing technical debt with the sqale method, *Software, IEEE* 29 (6) (2012) 44–51.
- [S6] Z. Codabux, B. Williams, Managing technical debt: An industrial case study, in: *International Workshop on Managing Technical Debt (MTD)*, 2013.
- [S7] D. Falessi, M. Shaw, F. Shull, K. Mullen, M. Keymind, Practical considerations, challenges, and requirements of tool-support for managing technical debt, in: *International Workshop on Managing Technical Debt (MTD)*, 2013.
- [S8] B. Curtis, J. Sappidi, A. Szynkarski, Estimating the principal of an application's technical debt, *Software, IEEE* 29 (6) (2012) 34–42.
- [S9] C. Seaman, Y. Guo, N. Zazworka, F. Shull, C. Izurieta, Y. Cai, A. Vetro, Using technical debt data in decision making: Potential decision approaches, in: *MTD Workshop*, 2012.
- [S10] R. Marinescu, Assessing technical debt by identifying design flaws in software systems, *IBM Journal of Research and Development* 56 (5) (2012) 9:1–9:13.
- [S11] R. Kazman, Y. Cai, R. Mo, Q. Feng, L. Xiao, S. Haziye, V. Fedak, A. Shapochka, A case study in locating the architectural roots of technical debt, in: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2, 2015, pp. 179–188.
- [S12] F. Shull, D. Falessi, C. Seaman, M. Diep, L. Layman, Technical debt: Showing the way for better transfer of empirical results, in: J. Münch, K. Schmid (Eds.), *Perspectives on the Future of Software Engineering*, Springer Berlin Heidelberg, 2013, pp. 179–190.
- [S13] P. Kruchten, R. Nord, I. Ozkaya, Technical debt: From metaphor to theory and practice, *Software, IEEE* 29 (6) (2012) 18–21.

- [S14] Z. Li, P. Liang, P. Avgeriou, Chapter 9 - architectural debt management in value-oriented architecting, in: I. Mistrik, R. Bahsoon, R. Kazman, Y. Zhang (Eds.), *Economics-Driven Software Architecture*, Morgan Kaufmann, Boston, 2014, pp. 183 – 204.
- [S15] K. Schmid, On the limits of the technical debt metaphor some guidance on going beyond, in: *MTD workshop*, 2013.
- [S16] N. Ramasubbu, C. Kemerer, Towards a model for optimizing technical debt in software products, in: *International Workshop on Managing Technical Debt (MTD)*, 2013.
- [S17] C. Fernández-Sánchez, J. Díaz, J. Pérez, J. Garbajosa, Guiding flexibility investment in agile architecting, in: *HICSS*, 2014.
- [S18] Z. Li, P. Liang, P. Avgeriou, Architectural technical debt identification based on architecture decisions and change scenarios, in: *Software Architecture (WICSA)*, 2015 12th Working IEEE/IFIP Conference on, 2015, pp. 65–74.
- [S19] R. Nord, I. Ozkaya, P. Kruchten, M. Gonzalez-Rojas, In search of a metric for managing architectural technical debt, in: *WICSA/ECSA*, 2012.
- [S20] J. Ho, G. Ruhe, When-to-release decisions in consideration of technical debt, in: *MTD workshop*, 2014.
- [S21] Y. Guo, C. Seaman, R. Gomes, A. Cavalcanti, G. Tonin, F. da Silva, A. Santos, C. Siebra, Tracking technical debt: An exploratory case study, in: *ICSM*, 2011.
- [S22] J. Bohnet, J. Döllner, Monitoring code quality and development activity by software maps, in: *International Workshop on Managing Technical Debt (MTD)*, 2011.
- [S23] J. Brondum, L. Zhu, Visualising architectural dependencies, in: *International Workshop on Managing Technical Debt (MTD)*, 2012.

- [S24] Z. Codabux, B. Williams, N. Niu, A quality assurance approach to technical debt, in: SERP, 2014.
- [S25] Y. Guo, C. Seaman, A portfolio approach to technical debt management, in: International Workshop on Managing Technical Debt (MTD), 2011.
- [S26] Y. Guo, R. Spínola, C. Seaman, Exploring the costs of technical debt management a case study, Empirical Software Engineering.
- [S27] C. Izurieta, G. Rojas, I. Griffith, Preemptive management of model driven technical debt for improving software quality, in: QoSA 2015, 2015.
- [S28] J.-L. Letouzey, The sqale method for evaluating technical debt, in: International Workshop on Managing Technical Debt (MTD), 2012, pp. 31–36.
- [S29] Z. Li, P. Liang, P. Avgeriou, N. Guelfi, A. Ampatzoglou, An empirical investigation of modularity metrics for indicating architectural technical debt, in: QoSA 2014, 2014.
- [S30] E. Lim, N. Taksande, C. Seaman, A balancing act: What software practitioners have to say about technical debt, Software, IEEE 29 (6) (2012) 22–27.
- [S31] M. Naedele, R. Kazman, Y. Cai, Making the case for a "manufacturing execution system" for software development, Commun. ACM.
- [S32] A. Nugroho, J. Visser, T. Kuipers, An empirical model of technical debt and interest, in: International Workshop on Managing Technical Debt (MTD), 2011.
- [S33] N. Ramasubbu, C. Kemerer, Managing technical debt in enterprise software packages, Software Engineering, IEEE Transactions on 40 (8) (2014) 758–772.
- [S34] N. Ramasubbu, C. Kemerer, C. Woodard, Managing technical debt: Insights from recent empirical evidence, Software, IEEE 32 (2) (2015) 22–25.

- [S35] D. Reimanis, C. Izurieta, R. Luhr, L. Xiao, Y. Cai, G. Rudy, A replication case study to measure the architectural quality of a commercial system, in: ESEM 2014, 2014.
- [S36] K. Schmid, A formal approach to technical debt decision making, in: QoSA, 2013.
- [S37] R. Schwanke, L. Xiao, Y. Cai, Measuring architecture quality by structure plus history analysis, in: ICSE, 2013.
- [S38] C. Siebra, A. Cavalcanti, F. Silva, A. Santos, T. Gouveia, Applying metrics to identify and monitor technical debt items during software evolution, in: ISSREW, 2014, 2014, pp. 92–95.
- [S39] G. Skourletopoulos, R. Bahsoon, C. Mavromoustakis, G. Mastorakis, E. Pallis, Predicting and quantifying the technical debt in cloud software engineering, in: CAMAD, 2014, 2014, pp. 36–40.
- [S40] H. Sneed, Dealing with technical debt in agile development projects, Lecture Notes in Business Information Processing.
- [S41] J. Yli-Huumo, A. Maglyas, K. Smolander, The sources and approaches to management of technical debt: A case study of two product lines in a middle-size finnish software company, Lecture Notes in Computer Science.
- [S42] N. Zazworka, C. Seaman, F. Shull, Prioritizing design debt investment opportunities, in: MTD workshop, 2011.
- [S43] N. Zazworka, A. Vetro, C. Izurieta, S. Wong, Y. Cai, C. Seaman, F. Shull, Comparing four approaches for technical debt identification, Software Quality Journal 22 (3) (2014) 403–426.
- [S44] A. Chatzigeorgiou, A. Ampatzoglou, A. Ampatzoglou, T. Amanatidis, Estimating the breaking point for technical debt, in: Managing Technical Debt (MTD), 2015 IEEE 7th International Workshop on, 2015, pp. 53–56.

- [S45] J. Holvitie, V. Leppnen, Examining technical debt accumulation in software implementations, *International Journal of Software Engineering and its Applications* 9 (6) (2015) 109–124.
- [S46] D. Falessi, P. Kruchten, Five reasons for including technical debt in the software engineering curriculum, in: *Proceedings of the 2015 European Conference on Software Architecture Workshops, ECSAW '15*, ACM, New York, NY, USA, 2015, pp. 28:1–28:4.
- [S47] J. Yli-Huumo, A. Maglyas, K. Smolander, How do software development teams manage technical debt? - an empirical study, *Journal of Systems and Software*.
- [S48] U. Eliasson, A. Martini, R. Kaufmann, S. Odeh, Identifying and visualizing architectural debt and its efficiency interest in the automotive domain: A case study, in: *Managing Technical Debt (MTD)*, 2015 IEEE 7th International Workshop on, 2015, pp. 33–40.
- [S49] F. Oliveira, A. Goldman, V. Santos, Managing technical debt in software projects using scrum: An action research, in: *Agile Conference (AGILE)*, 2015, 2015, pp. 50–59.
- [S50] M. Naedele, H.-M. Chen, R. Kazman, Y. Cai, L. Xiao, C. V. Silva, Manufacturing execution systems: A vision for managing software development, *Journal of Systems and Software* 101 (2015) 59 – 68.
- [S51] N. A. Ernst, S. Bellomo, I. Ozkaya, R. L. Nord, I. Gorton, Measure it? manage it? ignore it? software practitioners and technical debt, in: *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015*, ACM, New York, NY, USA, 2015, pp. 50–60.
- [S52] S. Akbarinasaji, Toward measuring defect debt and developing a recommender system for their prioritization, Vol. 1469, 2015, pp. 15–20.

- [S53] D. Falessi, A. Reichel, Towards an open-source tool for measuring and visualizing the interest of technical debt, in: *Managing Technical Debt (MTD)*, 2015 IEEE 7th International Workshop on, 2015, pp. 1–8.
- [S54] D. Falessi, A. Voegelé, Validating and prioritizing quality rules for managing technical debt: An industrial case study, in: *Managing Technical Debt (MTD)*, 2015 IEEE 7th International Workshop on, 2015, pp. 41–48.
- [S55] T. Mendes, D. Almeida, N. Alves, R. Spnola, R. Novais, M. Mendona, Visminertd: An open source tool to support the monitoring of the technical debt evolution using software visualization, Vol. 2, 2015, pp. 457–462.
- [S56] E. Alzaghou, R. Bahsoon, Evaluating technical debt in cloud-based architectures using real options, in: *ASWEC 2014*, 2014.
- [S57] J. de Groot, A. Nugroho, T. Back, J. Visser, What is the value of your software?, in: *International Workshop on Managing Technical Debt (MTD)*, 2012.
- [S58] A. Mayr, R. Plosch, C. Korner, A benchmarking-based model for technical debt calculation, in: *QSIC 2014*, 2014.
- [S59] V. Singh, W. Snipes, N. A. Kraft, A framework for estimating interest on technical debt by monitoring developer activity related to code comprehension, in: *International Workshop on Managing Technical Debt (MTD)*, 2014.
- [S60] E. Alzaghou, R. Bahsoon, Cloudmtd: Using real options to manage technical debt in cloud-based service selection, in: *International Workshop on Managing Technical Debt (MTD)*, 2013.
- [S61] A. Martini, J. Bosch, M. Chaudron, Architecture technical debt: Understanding causes and a qualitative model, in: *SEAA 2014*, 2014.
- [S62] G. Skourletopoulos, C. X. Mavromoustakis, G. Mastorakis, J. J. P. C. Rodrigues, P. Chatzimisios, J. M. Batalla, A fluctuation-based modelling

approach to quantification of the technical debt on mobile cloud-based service level, in: 2015 IEEE Globecom Workshops (GC Wkshps), 2015, pp. 1–6.

- [S63] I. Griffith, H. Taffahi, C. Izurieta, D. Claudio, A simulation study of practical methods for technical debt management in agile software development, in: WSC 2014, 2014.

References

- [1] E. Tom, A. Aurum, R. Vidgen, An exploration of technical debt, *Journal of Systems and Software* 86 (6) (2013) 1498–1516.
- [2] L. Peters, Technical debt: The ultimate antipattern, in: *International Workshop on Managing Technical Debt (MTD)*, 2014.
- [3] Z. Li, P. Avgeriou, P. Liang, A systematic mapping study on technical debt and its management, *J. Syst. Softw.* 101 (C) (2015) 193–220.
- [4] N. S. R. Alves, L. F. Ribeiro, V. Caires, T. S. Mendes, R. O. Spínola, Towards an ontology of terms on technical debt, in: *International Workshop on Managing Technical Debt (MTD)*, 2014.
- [5] N. S. Alves, T. S. Mendes, M. G. de Mendona, R. O. Spnola, F. Shull, C. Seaman, Identification and management of technical debt: A systematic mapping study, *Information and Software Technology* 70 (2016) 100 – 121.
- [6] A. Ampatzoglou, A. Ampatzoglou, A. Chatzigeorgiou, P. Avgeriou, The financial aspect of managing technical debt, *Inf. Softw. Technol.*
- [7] C. Fernández-Sánchez, J. Garbajosa, A. Yague, A framework to aid in decision making for technical debt management, in: *Managing Technical Debt (MTD)*, 2015 IEEE 7th International Workshop on, 2015, pp. 69–76.
- [8] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: *EASE*, 2008.

- [9] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, ACM, New York, NY, USA, 2014, pp. 38:1–38:10.
- [10] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, P. Abrahamsson, Software development in startup companies: A systematic mapping study, *Information and Software Technology* 56 (10) (2014) 1200 – 1218.
- [11] A. Strauss, J. Corbin, et al., Basics of qualitative research, Vol. 15, Newbury Park, CA: Sage, 1990.
- [12] M. Ivarsson, T. Gorschek, A method for evaluating rigor and industrial relevance of technology evaluations, *Empirical Softw. Engg.* 16 (3) (2011) 365–395.
- [13] S. Tockey, Chapter 3 - aspects of software valuation, in: I. Mistrik, R. Bahsoon, R. Kazman, Y. Zhang (Eds.), *Economics-Driven Software Architecture*, Morgan Kaufmann, Boston, 2014, pp. 37 – 58.
- [14] P. Clements, L. Northrop, *Software Product Lines: Practices and Patterns*, Addison-Wesley Professional, 2001.
- [15] P. Bourque, R. E. Fairley, *Guide to the Software Engineering Body of Knowledge - SWEBOK v3.0*, 2014th Edition, IEEE CS, 2014.
- [16] C. Fernández-Sánchez, J. Garbajosa, C. Vidal, A. Yagüe, An analysis of techniques and methods for technical debt management: a reflection from the architecture perspective, in: *SAM 2015*, 2015.
- [17] T. Suomalainen, O. Salo, P. Abrahamsson, J. Simil, Software product roadmapping in a volatile business environment, *Journal of Systems and Software* 84 (6) (2011) 958 – 975.

- [18] A. C. Tricco, J. Tetzlaff, M. Sampson, D. Fergusson, E. Cogo, T. Horsley, D. Moher, Few systematic reviews exist documenting the extent of bias: a systematic review, *Journal of Clinical Epidemiology* 61 (5) (2008) 422 – 434.

Bibliography

Carlos Fernández-Sánchez is a PhD candidate at the Technical University of Madrid, Spain. He is working in the research fields of technical debt management and software architecture. Before becoming a PhD student, he worked for six years as a software engineer developing data acquisition systems. He holds a bachelors degree in software engineering and a master degree in systems and software from Technical University of Madrid.

Juan Garbajosa is Professor at the Computer Systems School of the Technical University of Madrid (UPM), Spain. He leads the SYST research group, currently working on agile software development, software/systems architecture, and Smart Cities and Smart Grids, and software innovation. He has led several European projects, currently the CPSE-Labs project. His publications are in leading refereed conferences and journals. ISO/IEC-JTC1-SC7-WG20 (2002-2015) Convenor, and co-editor of SWEBOK V3. General Programme co-Chair in XP2011 conference. Before joining UPM he spent over fifteen years in industry. He is Senior Member of IEEE, and a member of ACM.

Agustin Yage is an Associate Professor of the Department of Informatics Systems, Technical University of Madrid. He received his PhD degree in Computer Science from UPM. His research interests include software processes, software processes improvement, project management, agile development approaches, system and software testing processes. He participated in several international and national research projects in Smart grids, agile methodologies and global software development. He is member of AENOR, the Spanish association for standardization. He is working in the technical committee about Information Technology AEN/CNT71. Subcommittee SC-07. He is enrolled as program committee on several International Conferences on Information Technology.

Dr. Jennifer Pérez is associate professor at the E.T.S. Ingeniera de Sistemas Informáticos of the Technical University of Madrid (UPM). She received her PhD degree in Computer Science from the Polytechnic University of Valencia in 2006, and is recipient of the best thesis Award in Computer Science and Telecommu-

nications of the UPV 2006-2007. Since 2001, she is participating in several European and national projects related to Software Engineering and ICT on Cyber-Physical Systems (CPS). Her research interests are focused on Software Architectures, Cloud Computing, Software Product Lines, Model-Driven Development, Software Evolution, Agile methodologies and Aspect-Oriented Software Development.