ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 5, No.17 , September 2015
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Variability management challenges in small and medium sized enterprises (SMEs)

Tuomas Ihme[1], Päivi Parviainen[1], Susanna Teppola[1]

[1]VTT Technical research centre of Finland
P.O.BOX 1100, 02044 VTT, Finland
Tuomas.ihme@vtt.fi, paivi.parviainen@vtt.fi, susanna.teppola@vtt.fi

Abstract

Variability has become important means to provide customer specific products fast to the market. This paper presents findings from a literature study and questionnaire designed to find out the current variability management practices, challenges and needs in small and medium sized companies (SMEs). The literature study was carried out first, and based on its findings questionnaire study was designed and carried out. Transparency of variability and design decisions between R&D and business and customer interfaces and transparency of variability and design decisions within R&D were the most important and challenging variability areas in the studied organisations. The most frequently used architecture design issue was the use of known architectural styles and patterns to handle variability. The findings of the most challenging and important variability management topics, as well as the important improvement goals can be used to direct future research, and efforts to support companies in variability management.

Keywords: *Variability, variability management, software engineering, challenges*

1. Introduction

Nowadays, a trend is that companies must remain efficient and effective while, at the same time, offer a much richer product variety to their customers than even before [1, 2]. According to Svahnberg et al [3], software variability is the ability of a software system or artefact to be efficiently extended, changed, customized or configured for use in a particular context. Variability management is different depending on the context of product development [2, 4, 5, 6, 7, 8]. For example, engineering for safety-critical systems is a complex, difficult and laborious task involving many technical fields and big challenges [9]. Thus, before adoption, the suitability of a variability management approach for the specific domain and needs of a company needs to be evaluated.

Currently main research focus relating to variability has been on product lines, that is a more common practice in large companies, and less in small and medium sized companies. According to Knauber et al [10] new software companies usually start with one idea and if it is successful, they often use the same idea to develop variations of the product. Thus, there is little research on variability management in small and medium sized companies (SMEs), although the variability concept is becoming more and more important also for SME's. The study discussed in this paper was directed to find out what is the current practice of variability management in SME's, what are the most commonly used variability techniques, challenges the companies are facing, and their most important improvement goals with respect to variability.

This paper provides empirical material about variability management practices, challenges and needs in small and medium sized development organisations and the differences between actual industrial needs and practices and the variability management research. The paper is organized as follows: Next the related work is discussed in section 2. Section 3 presents the materials and methods used in the study. Section 4 focuses on presenting the results of the questionnaire of three main topics: technical variability management, designing and implementing variability and variability improvement goals. Section 5 discusses these results, specifically with respect to other research. Finally, conclusions are discussed in section 6.

2. Related work

Understanding commonality and variability among products in a products line's scope plays a central role in product line development [11]. Identifying and understanding commonality aims to form the basis for reuse [12]. Furthermore, software product line research has reported dozens of variability management approaches and solutions to variability related challenges in companies [13, 14, 15, 16]. However, a big mismatch between the output of the research and industrial needs has been reported: more than 70 per cent of the proposed approaches have not been evaluated in an industrial setting [16, 17, 18]. In practice, an industrial product line can include thousands of variable features making variability management and product derivation tasks extremely difficult [5, 6]. Bosch [17] has found a big difference between the academic view and the industrial practice of variability. The academic view was that assets have few and explicitly defined variation points and variants are configured during instantiation by other black-box components. The industrial practice usually implemented variations via configuration and

specialization or replacement of entities internal to the asset.

Jacobson et al. [19] present five ways to implement variability: inheritance, extensions, parameterization, configuration and generation. Many of the techniques can be found in design patterns [20]. The use of simple and known software architecture design patterns compensates some lacking variability implementation features in object-oriented languages and may help manage a multiplicity of variable features [3, 5]. Domain or reference architectures [21] and industrial standards [22] have been used to manage variability in product lines and control systems. Furthermore, Svahnberg et al., [3] have discovered thirteen variability realisation techniques via case studies.

Variability modelling and documentation was the most addressed issue in 33 variability management approaches analysed by Chen et al. [23]. However, academic approaches for variability modelling and documentation seem to have many limitations that hinder industrial practitioners from fully utilizing them [18]. Product derivation was motivating the development of the second largest number of the variability management approaches. Four of the approaches were developed for variability management at the architecture level. Both the evolution of variability and the identification of commonalty and variability were addressed by three approaches. Only a few approaches addressed systematic process support, customizability, tool support, binding time and scalability.

According to literature, small organisations tend to have special challenges and practices [4, 7, 8, 24, 25, 26, 27, 28, 29, 30, 31]. For example, the formality level of documentation and managerial practices tend to increase as the development organization becomes bigger [4, 7, 8, 28]. In the study of Thörn [8], the level of formalism of variability management structures in SMEs was generally rather low and varied from sophisticated to ad hoc approaches. For example, all organizations with 50 or more employees of the study were using architecture and component descriptions.

3. Questionnaire

Literature findings were used as the basis of the questionnaire design. Some findings of previous empirical research emphasise the challenges of product variability in companies. The challenges are outlined in Table 1. Initial categories of the variability dimensions were derived from the publications of real life, industrial situations. The created challenge categories (common themes) are used as a basis for the inductive data analysis described in Section 0. The survey findings are discussed with respect to the challenges presented in Table 1 in Section 0.

Table 1. Technical variability challenges in software companies

Category	*Chen & Ali Babar [18]*	*Bosch et al. [32]*	*Thörn [8}*
Requirements and knowled-ge mana-gement	Knowledge harvest and management Handling variability in different development phases Evolution of variability	Implicit dependencies between architectural elements and features. Visibility to the variability at the requirements and realization level Ambiguous domain concepts	Difficulties in managing the increased size of product variants and several concurrent software versions Problems in versioning and internal communication due to the lack of product structure documentation
Reuse	Extracting variability and commonality from existing technical artefacts		Duplicated work Little reuse in the problem space Variant reuse
Design	Componentizing the existing code and building variability inside and around them The existing architecture does not support the required variability in new requirements Difficulties to compare code bases Difficulties in managing architectural design decisions	Trust in the variants that are added by the 3th party after construction of the product Difficulties to find all code that belongs to one feature Design is often oversimplified by architects A lacking awareness of variability mechanisms Wrong decisions about binding times	Lack of product structure documentation
Testing	Large amounts of efforts are spent on software testing in industry Testing variability in SPL	A large amount of variability may make it virtually impossible to test all combinations during developmen.	-
Methods and tools	Lack of sufficient tool support for managing variability Variability modelling approaches are not very user friendly An integrated and end-to-end tool support for variability	Lack of sufficient tool support to automate variability management	Lack of tools to do systematic reuse

The questionnaire of this survey was sent to the partners of the VARIES project's during autumn 2012. The questionnaire was filled by all industrial partners (i.e., companies). The responses from small and medium sized

(under 250 employees) companies (12 companies, 14 responses) are considered in this study.

The questionnaire was created by combining work from several VARIES project partners, projects and sources, including a prestudy via interviews of several companies and a study of the state of art in variability practices and research (see Table 1). The state of the art study included recent results of software research that has provided a multiplicity of variability management approaches, discussions and solutions to variability related challenges in companies.
The questionnaire combined several viewpoints, such as business, managerial, and technical in order to get a comprehensive view on variability in the companies. As variability is relevant to the whole development organization and its' products, it was also reasonable that the questionnaire addressed variability comprehensively. The questionnaire high level topics included:

- the respondents' organisation (e.g., domain, organisation size, product development model),
- the respondent (e.g., role, experience, variability related responsibilities),
- the product (e.g., maturity, software intensity, safety criticality, life expectancy, differentiation, number of customers),
- general variability aspects (e.g., definition, variability requirements, variability importance),
- current variability practices (e.g., technical variability areas, variability factors impacts, well working variability areas, challenges, current usage of architecture and variability design techniques and variability realisation techniques), and
- variability goals (i.e., how the respondent would like to improve variability handling).

The questions were semi-structured, there were some completely open questions, and some with pre-defined options, including an option to give respondents own option (i.e., other, please fill). Also, in some questions the respondents were asked to rank pre-defined options.
The draft versions of the questionnaire were tested with three industrial partners of the VARIES project. The questionnaire was completed during autumn 2012. Altogether 16 responses from all 14 VARIES industrial partners were received (some industrial partners gave more than one response). 14 responses of 12 companies were from SME's and are thus included in the results presented in this paper. The studied product development organisations were active in many application domains (Table 2), also, many of the companies were working on several domains.

Table 2. The domains of the studied small and large organisations

Domain	***Number***
Telecommunications	3
Consumer electronics	1
Safety equipment	2
Industry automation	5
Automotive and transportation	4
IT systems	5
Research in above domains	1

The respondents from the studied organisations had a long experience (8 over 20 years, 4 over 10 years and 2 under 10 years) of developing products. Most of the respondents were project managers or team leaders (10 responses), the rest were product managers, architects, and developers. Some respondents had more than one role.

Knight's (2002) definition of safety-critical systems was used in this study's questionnaire: "Safety-critical systems are those systems whose failure could result in loss of life, significant property damage, or damage to the environment." The questionnaire included a question and options for the respondents to determine the safety-criticality of their products:

1. No potential for injury, pollution, fire or effect on safety systems
2. Some products or product parts where failure can cause injury, pollution, fire or effect on safety systems
3. All units / complete product is safety-critical
4. Product may have indirect safety-critical effects (e.g., via another system)
5. Other, please specify

We classified the selections of the options 2 and 3 in the category of safety critical products and the selections of the options 1 and 4 in the category of non-safety-critical products. For category 5 the classification depended on the specification given by the respondent. In this case, two respondents selected the option 5. These were classified as non-safety critical, as the specifications were "indirect effect", and "usually our products are not safety critical, but some can be". Altogether six organisations answered from the viewpoint of safety-critical products and eight from the viewpoint of non-safety-critical products.

4. Results

The results are presented from technical variability management, designing and implementing variability and variability improvement goals viewpoints. The questions which responses these sections are based on, are presented in appendix 1.

4.1 Technical variability management

This section presents the respondents view on technical variability management topics, specific focus was to analyse which technical variability management areas work well and which do not work well and also to find out the importance of these topics for companies. The technical variability management topics included in the questionnaire were derived from and classified according to literature study findings summarized in Table 1. The technical variability management includes topics related to practical management of variability, i.e., how the existing variability in the product is managed, relating to realisation, modelling, testing, traceability, documentation, and reuse for example.

The following Fig 1 shows the respondents evaluation of technical variability management topics from the viewpoint of how much they have faced challenges relating to the topic, and how important they saw the topic for their organisation. The position of the bubble in the figure indicates the challenge level, and the importance level of each topic. The more top right corner the topic is placed, the more important and challenging it was seen.

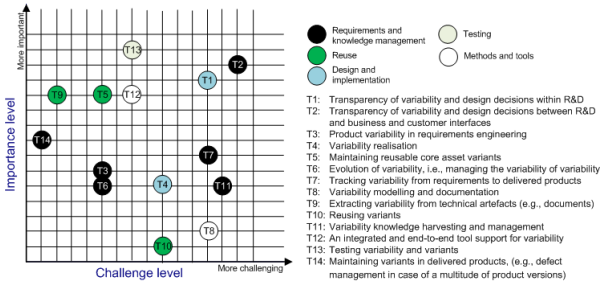


Fig 1. The challenge level and importance of technical variability management topics

The figure shows that the topics 'Transparency of variability and design decisions between R&D and business and customer interfaces' (T2) and 'Transparency of variability and design decisions within R&D' (T1) were the most important and challenging variability areas in the studied organisations. 'Testing variability and variants' (T13) topic was rated important but not very challenging. 'Variability knowledge harvesting and management' (T11) is among the most challenging issues in the organisations but not seen important in the organisations. Topics T9 'Extracting variability from technical artefacts' and T14 'Maintaining variants in delivered products' were seen relatively important, but they seem to be working well in practice, as they were not ranked challenging. The least important but still challenging topics were T10 'Reusing variants' and T8 'Variability modelling and documentation' in the respondents opinion.

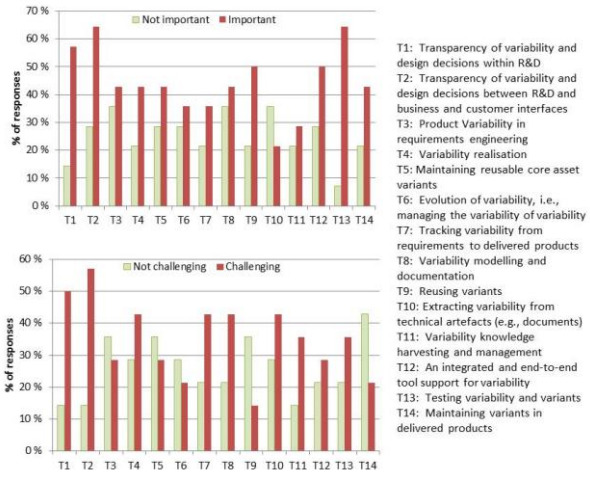


Fig 2. The ratings of importance and challenge level of technical variability management topics

The topics T1, T2, T9, T12 and T13 were seen important by more than 50% of the respondents and topics T1 and T2 as challenging. Topics T9 and T14 were seen important by many respondents but not challenging by most respondents.

One of the respondents commented that they are using a configuration tool on the hardware level to make products different for different customers. Another one of the organisations needed tool support for making decisions whether a product line development approach would be more beneficial than exploiting a configurable product or vice versa. The product development strategy of another organisation was to develop common features to maximize the reuse of their software platform. Their main product delivery strategy was to make customisations by setting parameters without a need for changes to the basic product. In their case, customisations for some very important customers could sometimes conflict with the strategy and the needs of the basic product.

4.2 Designing and implementing variability

Fig 3 shows which architecture design issues were frequently used to handle product variability in all studied organisations (1= we don't use, 5 = we use frequently). Size of the bubble indicates number of responses in that category.

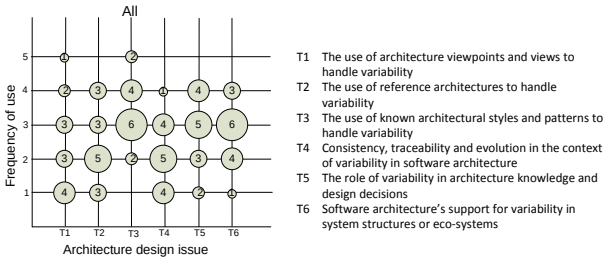


Fig 3. Frequency of use of architecture design issues (frequency of use 1= we don't use, 5 = we use frequently)

The most frequently used architecture design issue was T3 'The use of known architectural styles and patterns to handle variability'. Other architectural design issues usage was quite even. Most of the architectural design issues found during the literature study were used at least at some frequency in the respondents' organisations. The least used issues were T1 'The use of architecture viewpoints and views to handle variability' and T4 'Consistency, traceability and evolution in the context of variability in software architecture.

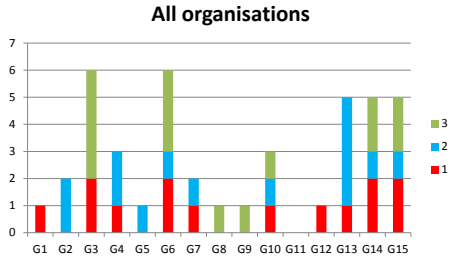
The main difference between safety-critical and non-safety-critical organisations with respect to the use of architectural design issues was that 'The use of reference architectures to handle variability' (T2) was more common in safety-critical organisations than in non-safety-critical organisations. Also, the architecture design

issues in general were more commonly used in safety-critical organisations.

One of the respondents developing non-safety-critical products commented that they have not yet found practical ways to document the used architectural style. They were using parameterisation as the sole intentional mechanism to implement variability. They have had problems in managing a multiplicity of parameters. In addition, they used some design patterns.

4.3 Variability improvement goals

Fig 4 and Fig 5 show the results of the question: 'Which of the following describes best your main improvement goal w.r.t variability?'. As some respondents ranked only the three most important goals, only the respondents' most important (1), second most important (2) and third most important (3) rankings are included in these figures.



G1 Ensuring management's variability support
G2 Better handling of complex variability
G3 Better variability management via the transparency of variability
G4 Make variability decisions more transparent by making the decisions together
G5 Better visibility of variability to customers
G6 Better product platform's support for variability
G7 Better product architecture's support for variability
G8 Better extraction of variability and commonality from the various artefacts of similar existing systems
G9 Better management of the evolution of variability (i.e., managing the variability of variability in scope, requirements, architecture, reusable assets, design & implementation (design decisions))
G10 Improve documenting the variability of products
G11 Better handling of security variability, e.g., manage commonalities and variability of security requirements artefacts and their traceability and improve the transparency of security variability decisions
G12 Better handling of safety variability, e.g., manage commonalities and variability of safety requirements artefacts and their traceability and improve the transparency of safety variability decisions
G13 Use of user friendly variability modelling approaches
G14 Better tool support for managing variability
G15 Better methods and tools for variability in product development phases (requirements engineering, maintenance etc.)

Fig 4. Three most important variability improvement goals

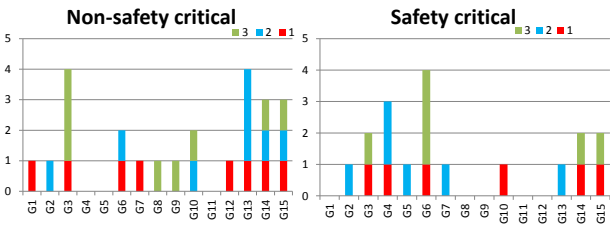
For all organisations, the goals G3 'Better variability management via the transparency of variability' and G6 'Better product platform's support for variability' were ranked the most important. Also goals G13 'Use of user friendly variability modelling approaches', G14 'Better tool support for managing variability', and G15 'Better methods and tools for variability in product development phases' were ranked important. G11 'Better handling of security variabilty' was not ranked within three most important variability improvement goals by any respondent.

Respondents that selected G3 'Better variability management via the transparency of variability' as important goal were asked to select more detailed topics that were important with respect to the goal (respondents could select more than one topic). The topics selected as important by respondents were:
Better transparency of variability between R&D and business and customer interfaces (7 respondents)
Better transparency of variability decisions and architectural design decisions as well as the implications of those decisions (6 respondents)
Better transparency of variability between the core asset team, product teams and maintenance teams (5 respondents)
Making variability more transparent by increasing the visibility of product variability related information (5 respondents)
Making variability more transparent by using shared ownership, knowledge and skills (2 respondents)
Better transparency of variability between different departments who own different yet similar products (2 respondents)

Also, respondents that selected G15 'Better methods and tools for variability in product development phases' as important goal were asked to select more detailed topics that were important with respect to the goal (respondents could select more than one topic). The topics selected as important by respondents were:

- Better methods and tools for variability realisation (6)
- Better methods and tools for variability-related requirements engineering (4)
- Better methods and tools for variability-related quality assurance techniques (4)
- Better methods and tools for maintaining and upgrading delivered products (4)

Fig 5 presents the importance of variability improvement goals in safety-critical and non-safety-critical organisations.



G1 Ensuring management's variability support
G2 Better handling of complex variability
G3 Better variability management via the transparency of variability
G4 Make variability decisions more transparent by making the decisions together
G5 Better visibility of variability to customers
G6 Better product platform's support for variability
G7 Better product architecture's support for variability
G8 Better extraction of variability and commonality from the various artefacts of similar existing systems
G9 Better management of the evolution of variability (i.e., managing the variability of variability in scope, requirements, architecture, reusable assets, design & implementation (design decisions))
G10 Improve documenting the variability of products
G11 Better handling of security variability, e.g., manage commonalities and variability of security requirements artefacts and their traceability and improve the transparency of security variability decisions
G12 Better handling of safety variability, e.g., manage commonalities and variability of safety requirements artefacts and their traceability and improve the transparency of safety variability decisions
G13 Use of user friendly variability modelling approaches
G14 Better tool support for managing variability
G15 Better methods and tools for variability in product development phases (requirements engineering, maintenance etc.)

Fig 5. Three most important variability improvement goals in safety-critical and non-safety-critical organisations

There are clear differences in the importance of variability improvement goals for safety-critical and non-safety-critical organisations. For safety-critical organisations the G6 'Better product platform's support for variability' was clearly the most important variability goal, and the G4 'Make variability decisions more transparent by making the decisions together' the second

most important goal. The G4 was not at all important for non-safety-critical organisations. G6 was relatively important for all non-safety-critical organisations. For non-safety-critical organisations the most important improvement goals were G3 'Better variability management via the transparency of variability', G13 'Use of user friendly variability modelling approaches'. These were only moderately important for safety-critical organisations. Also G13 'Use of user friendly variability modelling approaches', G14 'Better tool support for managing variability' and G15 'Better methods and tools for variability in product development phases' were important for organisations with non-safety critical products and not important for organisations with safety critical products.

5. Discussion

In this section the results presented in section 0 are compared to the findings of the literature study. In general, findings from the questionnaire results are in line with the literature study, e.g., Chen and Ali Babar [18] who have found a big difference between the output of variability management research and industrial needs. The findings are also line with Bosch [17] who argues that there is a difference between the academic and industrial views of product variability.

5.1 Technical variability management challenges

The topic 'Transparency of variability and design decisions between R&D and business and customer interfaces' in the Requirements and knowledge management category was the second most important and the most challenging variability area in the studied organisations. This is in line with the findings of Bosch et al. [32]. The topic 'Transparency of variability and design decisions within R&D' was the third most important and the third most challenging variability area in the studied organisations. This is in line with the findings from several other researchers' work [8, 18, 32]. The topic 'Testing variability and variants' were the most important and the sixth most challenging variability area in the studied organisations as has also been pointed out by Chen & Ali Babar [18] and Bosch et al. [32].

The topic 'An integrated and end-to-end tool support for variability' was highly important (4th) and moderately challenging. This result is slightly different with the finding of Chen and Ali Babar [18]. These differences may be caused by that in Chen and Ali Babar [18] the majority of participants were from large companies. The topic 'Reusing variants' was the least important and the sixth most challenging variability area in the studied organisations. This is not in line with the findings of Thörn [8].

5.2 Designing and implementing variability

The findings of this study show a significant difference between the studied organisations and the organisations that systematically use software product line practices: the variability realisation technologies [3] were not commonly used in the studied organisations.

Some studied organisations used parameterisation as a mechanism to implement variability. Problems were encountered in managing a multiplicity of parameters. Known architectural styles and patterns were the most used architectural issue for handling variability. On the other hand, it was reported that no practical ways were found to document solutions based on some new architectural styles such as the Representational State Transfer (REST) architecture [33].

Some of the studied small and large organisations were using configuration tools to make safety-critical products different for different customers. This is in line with Bosch [17] who has found that industrial companies often implement variation via configuration and use own tools.

The answers also show that hardening product requirements (i.e., safety-critical requirements in this study) tend to increase much more the formality level in designing and implementing variability than the growing size of the development organisation.

5.3 Variability improvement goals

The studied organisations ranked the topics 'Better methods and tools for variability in product development phases' and 'Use of user friendly variability modelling approaches' very important. This is in line with the findings from previous work [18, 32]. The evolution of variability was not among the important goals in the studied organisations. This is not in line with the finding of Chen and Ali Babar [18]. Some studied organisations develop common features to improve the reuse of their software platform. However, identifying and analysing commonalities and variants from the various artefacts [12, 18] was not an important improvement goal in the studied organisations.

6. Conclusions, limitations and future work

This paper has presented findings from a questionnaire responses designed to find out how variability management practices, challenges and needs are different between small and large development organisations, and between non-safety-critical and safety-critical products. Transparency of variability and design decisions between R&D and business and customer interfaces and transparency of variability and design decisions within R&D were the most important and challenging variability areas in the studied organisations. The most frequently used architecture design issue was the use of known architectural styles and patterns to handle variability. Better variability management via the transparency of

variability and better product platform's support for variability were ranked the most important goals for improving variability aspects. These findings can be used to direct future research, and efforts to support companies in variability management.

Findings from this study show that safety-critical requirements were a significant variability factor in small organisations. Testing variability and variants was one of the most important and challenging topics in the studied small organisations with safety-critical products but less important and challenging in small organisations with non-safety-critical products. Transparency of variability and design decisions within R&D was one of the most important topics in safety-critical products but less important in non-safety-critical products. The topic was moderately challenging in both product types.

Although the number of responses in general, and especially from small organisations developing safety-critical products, was not very large, the results of the study are well in line with the results published by other researcher, which make the findings more reliable. For example, the formality level of designing and implementing variability tends to increase as the development organisation becomes bigger.

In the future our goal is to focus the research more on the variability from the business and product transparency perspective. Another aim is to use results of this study to make a survey that validates the organizational challenges and used industrial practices in a larger set of companies and larger amount of company representatives.

Appendix

Appendix 1. Questionnaire questions

Acknowledgments

The work presented in this paper has been carried out in ARTEMIS VARIES project. The authors would like to thank the VARIES partners for contribution to the questionnaire design and providing the responses to the questionnaire. Furthermore, the authors would like to thank ARTEMISIA JU and Tekes (the Finnish Funding Agency for Innovation) for enabling the research work.

References

[1] Kratochvíl, M. & Carson, C., 2005, Growing modular: mass customization of complex products, services and software, Berlin, Springer, 172 p.,

[2] Codenie, W., Gonzalez-Deleito, N., Deleu, J., Blagojevic, V., Kuvaja, P. & Similä, J., 2010, Managing Flexibility and Variability: A Road to Competitive Advantage, In Kang, K., Sugumaran, V. and Park, S. (Eds.) Applied Software Product Line Engineering,. New York: CRC Press, pp. 269-313.

[3] Svahnberg, M., van Gurp, J. & Bosch, J., 2005, A taxonomy of variability realization techniques, Software, Practice & experience, Volume: 35, Issue: 8, pp. 705-754.

[4] Mtigwe, B., 2005, The entrepreneurial firm internationalization process in the Southern African context: A comparative approach, International Journal of Entrepreneurial Behaviour & Research, Vol. 11 No: 5 pp. 358 – 377.

[5] Maccari, A. & Heie; A., 2005, Managing infinite variability in mobile terminal software, Software-Practice and Experience,.Vol. 35, No. 6, pp. 513–537.

[6] Loesch, F., & Ploedereder, E., 2007, Optimization of Variability in Software Product Lines, Proceedings of the 11th International Software Product Line Conference (SPLC 2007), IEEE CS Press, pp. 151–162.

[7] Laporte, C., Y., Alexandre, S. & O'Connor, R., V., 2008, A software engineering lifecycle standard for very small enterprises, Proceedings of the 15th European Software Process Improvement Conference, EuroSPI 2008, Dublin, Ireland, September 3-5, 2008, EuroSPI 2008, pp. 129-141. DOI: 10.1007/978-3-540-85936-9_12

[8] Thörn, C., 2010, Current State and potential of variability management practices in software-intensive SMEs: results from a regional survey, Information and Software Technology, 52, 411-421

[9] Knight, J.C., 2002, Safety critical systems: challenges and directions. Proceedings of the 24rd International Conference on Software Engineering (ICSE 2002), 19-25 May 2002, Orlando, Florida, USA, pp. 547-550.

[10]Knauber, P., Muthig, D., Schmid, K., Widen, T., 2000, "Applying Product Line Concepts in Small and Medium-Sized Companies," Software, IEEE , vol.17, no.5, pp.88,95, Sept.-Oct. 2000

[11]Clements, P. & Northrop, L., 2002, Software Product Lines, New York: Addison-Wesley.

[12]Coplien, J., Hoffman, D. & Weiss, D., 1998, Commonality and Variability in Software Engineering, IEEE Software, November/December 1998. pp. 37-45.

[13]Reinhartz-Berger, I., Sturm, A., 2009, Utilizing domain models for application design and validation, Information and Software Technology, Vol. 5, No. 8, pp. 1275–1289.

[14]Bosch, J., 2006, The challenges of broadening the scope of software product families, Communications of the ACM, vol. 49, no. 12, pp. 41- 44.

[15]Gurp, J., Bosch, J. & Svahnberg, M., 2001, On the Notion of Variability in Software Product Lines, Proceedings of the Working IEEE/IFIP Conference on Software Architecture (2001), pp. 45-55.

[16]Ali Babar, M., Chen, L., & Shull, F., 2010, Managing Variability in Software Product Lines, IEEE Software, vol. 27, no. 3, pp. 89-94.

[17]Bosch, J., 2000, Design and Use of Software Architectures, Adopting and Evolving a Product Line Approach, Pearson Education Limited, London, England, 354 p.

[18]Chen, L. & Ali Babar, M., 2010, Variability Management in Software Product Lines: An investigation of Contemporary Industrial Challenges, Proceedings of the 14th international conference on Software product lines SPLC'10.

[19]Jacobson, I., Griss, M. & Johnson, P., 1997, Software Reuse: Architecture, Process and Organization for Business Success, Addison-Wesley, Reading, MA, 1997.

[20]Gamma, E., Helm, R., Johnson, R., & Vlissides, J., 1995, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, New York.

[21]Pohl, K., Böckle, G. & van der Linden, F., 2005, Software Product Line Engineering: Foundations, Principles, and Techniques, Springer.

[22] Froschauer, R., Dhungana, D. & Grünbacher, P., 2008, Managing the Life-cycle of Industrial Automation Systems with Product Line Variability Models, The proceedings of the 34th Euromicro Conference Software Engineering and Advanced Applications. Pp. 35-42. DOI 10.1109/SEAA.2008.21
[23]Chen, L., Ali Babar, M. & Ali N., 2009, Variability management in software product lines: A systematic review, Proceedings of the 13th international software product line conference (SPLC'09), pp 81-90.
[24] Kamsties, E., Hörmann, K. & Schlich, M., 1998, Requirements engineering in small and medium enterprises, Requirements Engineering 3 (2), pp. 84–90.
[25]Groves, L., Nickson, R., Reeve, G., Reeves, S. & Utting, M., 2000, A survey of software development practices in New Zeland software industry, Aswec (Australian Software Engineering Conference) 00. Pp. 189-202.
[26]Nikula, U., Sajaniemi, J. & Kälviäinen, H., 2000, A state-of-the-practice survey on requirements engineering in small- and medium-sized enterprises, Research Report 1, Telecom Business Research Center, Lappeenranta University of Technology, Lappeenranta, Finland, 20 p.
[27]Manolova, T., Brush, C., Edelman, L. & Greene, P., 2002, Internationalization of Small Firms: Personal Factors Revisited, International Small Business Journal, Vol. 20 (1), pp. 9-31.
[28]Verlage, M. & Kiesgen, T., 2005, Five years of product engineering in small company, ICSE 2005, pp. 534-543.
[29]Mohagheghi, P. & Conradi, R., 2007, Quality, productivity and economic benefits of software reuse: a review of industrial studies, Empirical Software Engineering 12 (5), pp. 471-516.
[30]Rech, J., Bogner, C. & Haas, V., 2007, Using wikis to tackle reuse in software projects, IEEE Software 24 (6), pp. 99–104.
[31]Ihme, T., Pikkarainen, M., Teppola, S., Kääriäinen, J. & Biot, O., 2013, Challenges and industry practices for managing software variability in small and medium sized enterprises, Empirical Software Engineering. Published online: 24 April 2013.
[32]Bosch, J., Florijn, G., Greefhorst, D., Kuusela, J., Obbink, H., Pohl, K., 2002, Variability Issues in Software Product Lines, In: van der Linden, F. (Ed.) Fourth International Workshop on Product Family Engineering (PFE-4), Bilbao, Spain, 2001, LNCS 2290, pp.13-21.
[33]Fielding, R. & Taylor, R. 2002, Principled Design of the Modern Web Architecture, ACM Transactions on Internet Technology, Vol. 2, No. 2, pp. 115–150.

Tuomas Ihme is a Senior Scientist at VTT, the Technical Research Centre of Finland. His professional experience involves several years in the industry as a project manager as well as more than 20 years in research of embedded software and management of industrial development projects and national joint research projects. His areas of expertise are architectural design methods, modelling tools and software architectures in component software, product lines and agile software development. He has authored more than thirty scientific publications.

Dr. Päivi Parviainen received her M.Sc. in information processing science from the University of Oulu in 1996 and her PhD in information processing science from the University of Oulu in 2013. She is currently working as a Principal Scientist in the digital services in context team at VTT in Espoo, Finland. She has worked at VTT since 1995. Over the years, she has authored over 30 publications. Her research interests include product and digital service development practices.

Susanna Teppola (M.Sc.) has worked as a Research Scientist at VTT Technical Research Centre of Finland since 2000. Susanna has over fifteen years' experience in ICT, her current research interests are laying in the area of continuous software engineering, software product/service management and variability. In these areas Susanna has conducted and participated in many industrial and industry-driven research projects and project preparations both at national and international level.

Appendix 1. Questionnaire questions

Technical variability management

How important are the following technical variability areas for your organisation (rate each topic 1= not at all important – 5 very important)?

E1.1 Transparency of variability and design decisions within R&D
E1.2 Transparency of variability and design decisions between R&D and business and customer interfaces
E1.3 Product Variability in requirements engineering
E1.4 Variability realisation
E1.5 Maintaining reusable core asset variants
E1.6 Evolution of variability, i.e., managing the variability of variability
E1.7 Tracking variability from requirements to delivered products
E1.8 Variability modelling and documentation
E1.9 Reusing variants
E1.10 Extracting variability from technical artefacts (e.g., documents)
E1.11 Variability knowledge harvesting and management
E1.12 An integrated and end-to-end tool support for variability
E1.13 Testing variability and variants
E1.14 Maintaining variants in delivered products, (e.g., defect management in case of a multitude of product versions)
Other, please specify

Architecture design issues

Which of the following architecture design issues do you use currently to handle variability (1= we don't use, 5 = we use frequently)? Please, comment how well they work on the ones you are using.

F1.1 The use of architecture viewpoints and views to handle variability
F1.2 The use of reference architectures to handle variability
F1.3 The use of known architectural styles and patterns to handle variability
F1.4 Consistency, traceability and evolution in the context of variability in software architecture
F1.5 The role of variability in architecture knowledge and design decisions
F1.6 Software architecture's support for variability in system structures or eco-systems
Other, please specify

Improvement goals w.r.t. variability

Which of the following describes best your main improvement goal w.r.t variability? Rank at least the most important (1), second most important (2) and third most important (3) of the following items (1 = most important, 2=2nd most important etc.). You may also rank more items if you like. You may use each ranking only once

G1.1 Ensuring management's variability support
G1.2 Better handling of complex variability
G1.3 Better variability management via the transparency of variability
G1.4 Make variability decisions more transparent by making the decisions together
G1.5 Better visibility of variability to customers
G1.6 Better product platform's support for variability
G1.7 Better product architecture's support for variability
G1.8 Better extraction of variability and commonality from the various artefacts of similar existing systems
G1.9 Better management of the evolution of variability (i.e., managing the variability of variability in scope, requirements, architecture, reusable assets, design & implementation (design decisions))
G1.10 Improve documenting the variability of products
G1.11 Better handling of security variability, e.g., manage commonalities and variability of security requirements artefacts and their traceability and improve the transparency of security variability decisions
G1.12 Better handling of safety variability, e.g., manage commonalities and variability of safety requirements artefacts and their traceability and improve the transparency of safety variability decisions
G1.13 Use of user friendly variability modelling approaches
G1.14 Better tool support for managing variability
G1.15 Better methods and tools for variability in product development phases (requirements engineering, maintenance etc.)
Other, please specify, or comment if you are not able to answer