

# Requirements engineering in software product line engineering

Emilio Insfran · Gary Chastek · Patrick Donohoe ·  
Julio César Sampaio do Prado Leite

Published online: 6 November 2013  
© Springer-Verlag London 2013

**Abstract** Many attempts have been made to increase the productivity and quality of software products based on software reuse. Software product line practice is one such approach, one that focuses on developing a family of products which have a majority of features in common. Hence, there are numerous requirements that are common across the family, but others are unique to individual products. Traditional requirements engineering methods were conceived to deal with single product requirements and are usually not flexible enough to address the needs arising from reusing requirements for a family of products. There is also the additional burden of correctly identifying and engineering both product-line-wide requirements and product-specific requirements as well as evolving them. Therefore, in this special issue, we want to highlight the importance and the role of requirements engineering for product line development as well as to provide insights into the state of the art in the field.

**Keywords** Requirements engineering · Software product line · Reuse

## 1 Introduction

As our society becomes ever more reliant on digital technology, there is an increasing demand for software that is becoming ever more complex. In order to obtain high-quality products along with higher productivity, it is necessary to carefully elicit, specify, analyze and manage software requirements. This not only simplifies system design and implementation but also reduces the number of defects that are identified later in the implementation stage [3, 5]. Requirements engineering is introduced to address such issues early in the development process.

Software product line practice [1] is an approach based on the systematic creation and reuse of assets (e.g., requirements, architecture, components) in support of product development. Hence, there are many requirements that are common across the family of products, and there are others that are unique to individual products. In software product line development, requirements engineering processes and techniques are used to identify and characterize this dual nature of product line requirements and allocate them to potential member products based upon their commonality and variability.

The major risk associated with requirements engineering, in general, is the failure to capture the needed requirements. In product line development, this problem will be amplified over more than product. Specifying the wrong or inappropriate requirements, failing to keep the requirements up-to-date or failing to elicit the requirements at all puts the architect and the component developers at risk [4]. In a product line context, there is also the

---

E. Insfran (✉)  
Department of Information Systems and Computation,  
Universitat Politècnica de València, Valencia, Spain  
e-mail: einsfran@dsic.upv.es

G. Chastek · P. Donohoe  
Software Engineering Institute, Carnegie-Mellon University,  
Pittsburgh, PA, USA  
e-mail: gjc@sei.cmu.edu

P. Donohoe  
e-mail: pd@sei.cmu.edu

J. C. S. do Prado Leite  
Pontifícia Universidade Católica do Rio de Janeiro,  
Rio de Janeiro, Brazil  
e-mail: julio@inf.puc-rio.br

additional burden of correctly identifying and engineering both product-line-wide requirements and product-specific requirements as well as evolving them.

The main objective of this special issue is to highlight the importance and the role of requirements engineering in software product line development and to provide researchers and practitioners with insights into the state of the art and the state of the practice. We hope that the publication of this issue will encourage more researchers to look at the conjunction of both fields. Our expectation is that the challenges of complex product line software development will benefit from this integrated perspective.

## 2 The content of this issue

The three articles presented in this special issue have been selected following a thorough review process of 20 submissions that responded to the Call for Articles which was distributed. In addition, the best paper of the First International Workshop on Requirements Engineering Practices on Software Product Line Engineering (REPOS 2012), which was held in conjunction with the Software Product Line Conference (SPLC 2012), was invited to submit an extended and enhanced version. Each of these 21 articles was reviewed by at least three reviewers, with a range from three to four reviewers, in two review stages.

The first article “Requirements-Driven Incremental Adoption of Variability Management Techniques and Tools: An Industrial Experience Report” by Mahdi Derakhshanmanesh, Joachim Fox and Jürgen Ebert reports on an industrial experience at TRW Automotive in developing and adopting an approach for variability management and reuse of requirements. It also includes a description of the customization performed on third party tools (pure::variants, DOORS and Rhapsody) to automate the approach. The initial version of this article [2] was selected as the best paper of the REPOS 2012 workshop.

The second article “Evaluating Scenario-Based SPL Requirements Approaches: The Case for Modularity, Stability and Expressiveness” by Mauricio Alférez, Rodrigo Bonifácio, Leopoldo Teixeira, Paola Accioly, Uirá Kulesza, Ana Moreira, João Araújo and Paulo Borba presents an empirical study that compares and analyzes four existing approaches to modeling and managing variability in

software product line requirements specifications. In particular, graphics- versus text-based, and annotation- versus composition-based specification approaches were considered and their strengths and weaknesses noted. Recommendations for the definition of new variability management approaches are also provided.

The final article “A Holistic Approach to Feature Modeling for Product Line Requirements Engineering” by Jaejoon Lee, Kyo C. Kang, Pete Sawyer and Hyesun Lee proposes a comprehensive goal-driven feature modeling approach that explicitly partitions a product line into different viewpoints representing problem space (requirements) and solution space (features) and establishes explicit mappings between them. The proposed approach aims to make requirements engineering and software product line engineering (particularly the modeling of commonality and variability) more complementary activities.

**Acknowledgments** We would like to thank a number of people that contributed to the realization of this special issue. First, we thank to Professor P. Loucopoulos and Professor W. Robinson, Requirements Engineering Editors-in-Chief, for giving us the opportunity to prepare this special issue. In addition, we would like to thank the many reviewers for their professional effort to select the articles to reflect the essence of this special issue and all authors for their contributions.

## References

1. Clements P, Northrop LM (2001) Software product lines: practices and patterns. Addison-Wesley, Boston
2. Derakhshanmanesh M, Fox J, Ebert J (2012) Adopting feature-centric reuse of requirements assets: an industrial experience report. First international workshop on requirements engineering practices on software product line engineering, Salvador, Brazil
3. Kuloor C, Eberlein A (2002) Requirements engineering for software product lines, proceedings of the 15th international conference on software and systems engineering and their applications (ICSSEA'02), Paris, France
4. Northrop LM, Clements P (2013) A framework for software product line practice. Software engineering institute. <http://www.sei.cmu.edu/productlines/tools/framework/index.cfm>. Accessed 22 July 2013
5. Yu Y, Lapouchnian A, Liaskos S, Mylopoulos J, Leite JCSP (2008) From Goals to High-Variability Software Design. Foundations of Intelligent Systems, 17th International Symposium Proceedings. ISMIS 2008. Springer Lecture Notes in Computer Science, 4994: 1–16