Software product line engineering for consumer electronics

Hartmann, Herman

rijksuniversiteit
groningen

Software Product Line Engineering for Consumer Electronics

Keeping up with the speed of innovation

Proefschrift

ter verkrijging de graad van doctor aan de
Rijksuniversiteit Groningen
op gezag van de
rector magnificus prof. dr. E. Sterken
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

vrijdag 9 oktober 2015 om 12.45 uur

door

Johan Herman Hartmann

geboren op 28 september 1966
te Onstwedde

Promotor
Prof. dr. ir. J. Bosch

Beoordelingscommissie
Prof. dr. ir. P. Avgeriou
Prof. dr. ir. I. Crnkovic
Prof. dr. J.C. van Vliet

ISBN: 978-90-367-7991-3

Software Product Line Engineering for Consumer Electronics

Keeping up with the speed of innovation

Herman Hartmann

Abstract

During the last decade consumer electronics products have changed radically. Traditionally, these products were used for a few dedicated tasks, and were implemented through hardware. Nowadays, these products are used for a variety of tasks and most of the functionality is implemented through software. Furthermore there is an increasing amount of variants needed to serve the market and a continuous pressure on development cost, quality, and time-to-market. These trends have caused a significant impact on the way that software product lines for consumer electronics are being developed.

This thesis starts in part I with an exploration of the problem space: namely the challenges that are caused by the changing industry structures. This analysis shows that due to the increasing amount of software and high speed of innovation it is unfeasible for an individual company to develop the software on its own. As a consequence, consumer electronics products are built using components from a large variety of specialized firms. A model of five industry structure types is introduced that describes the transition and the consequences for software architectures. It is shown that software supply chain is the dominant industry structure for developing the software that is embedded in these products. This is because the modularization of the architecture is limited and because resource constrained devices require variants of software that are optimized for different hardware configurations.

To support applications from third parties, many consumer electronics firms offered a stable interface towards the 3rd party applications. In this way these firms transitioned from software product line engineering in an intra-organizational context to ecosystems for these 3rd party applications. Three types of ecosystems are identified that are currently used: vertically integrated hardware/software platforms, closed source software platforms and open source software platforms. A first step towards a decision support method is introduced that can determine which type of ecosystem is most suitable for a specific type of consumer electronics product from a software engineering perspective.

Part II of this thesis presents solutions to the challenges for variability management that arise when the software is developed through a supply chain. When a supplier delivers products to multiple customers, this may lead to multiple product lines. In this thesis a modelling approach is introduced that captures the context as a separate tree of a feature model, which makes it possible to model multiple product lines in software supply chains.

A company that is in the middle of a supply chain, has to integrate components from suppliers and offer (partly configured) products to its customers. To satisfy the variability requirements by each customer it may be necessary to use components from different suppliers, partly offering the same feature set, thus leading to overlapping feature models. This thesis introduces a supplier independent feature model. In this concept, dependency relations between the supplier independent feature model and the feature models of the individual suppliers are used to capture the variability of the combined product line. The use

of alternative components from suppliers partly may also lead to a product line with components which are using different mechanisms for interfacing, binding and variability. A model-driven approach is introduced for automating the integration between components. This approach reduces the development costs and time-to-market.

As a result of the growing number of variation points, feature models often become too complex to deal with. This thesis evaluates a textual variability language which shows that the readability of models using a textual language is easier than with graphical languages, especially when many constraints between features are used, which is common for feature models in software supply chains.

Part III of this thesis presents solutions to the challenge of ensuring a sufficient quality with a high pressure on time-to-market, as identified in the part I. This last part focusses on testing, the last phase in a development process. The consequence for testing is that due to the increasing amount of faults, of which many are caused by a mismatch between components from different suppliers, and the limited time for testing, it is not feasible to remove all faults before market introduction.

In this thesis a statistical model is introduced that determines the benefits of using operation profiles, which is a method to increase the test efficiency based on the quantitative usage by the end users. The analysis shows that using operational profiles improves the test efficiency, however not when a high reliability is required. The analysis of risk based testing, which is a method to focus testing on those areas that have the biggest impact on end user and business, shows that when the usage frequency is treated as a separate dimension of the risk matrix, the efficiency can be further improved.

To reduce the test effort caused by the high amount of variability, this thesis introduces risk based testing for software product line engineering, which extends risk based for single system engineering with a dimension that captures the percentage of product variants that use a particular development artefact. This method provides the means to guide the test effort for different test levels and set priorities during domain and application engineering.

Because operational profiles cannot be directly applied to testing of highly innovative consumer electronics products, as it focuses primarily on technical reliability risks, this thesis proposes an enhanced framework to analyze unexpected user-product interaction. In this framework, product, user as well as environmental conditions are used to analyze user-product interaction which will help reducing non-technical failures.

This thesis has the following key contributions:

- A classification of different industry structures and ecosystems for consumer electronics is introduced along with their architectural challenges.
- Solutions for managing the variability in software supply chains have been created and for managing large and complex feature models.
- Methods are introduced to improve the test efficiency for software product lines and for highly innovative consumer electronics products.

Acknowledgements

Doing a PhD is a journey that one cannot successfully bring to an end without the help of many. This is especially the case in which a PhD is pursued during a regular job. The list of persons that have helped me during the PhD is long, whether it is in the research itself, creating the right environment or giving personal support. Most likely I will have forgotten to mention many.

First of all, I would like to thank Jan Bosch. Jan was an inspiring promotor and he understands what it is like to do a PhD while working in the industry. His encouragements and optimism were essential to keep me motivated. I learned from Jan how to write papers that were business and strategy oriented. With a few questions he helped me to focus and structure my papers. Some of those questions and remarks, I always keep in mind when writing a paper.

Secondly, I would like to thank Tim Trew. Tim actively supported my ambition to do a PhD during my time at NXP Research and he taught me how to be thorough in writing scientific papers. His reviewing capacity is unparalleled and every time I review a paper, I try to do this as precise as Tim does.

My ambition to do research and a PhD started when I was working at Philips CFT. During that period I worked closely together with the Technical University of Eindhoven and I was collaborating on research projects and making my first steps in doing research and writing my own papers. In particular, I want to thank: Elke den Ouden, Vincent Ronteltap, Aarnout Brombacher, Lu Yuan, Guillaume Stollman, Peter Sonnemans, and Wim Geudens.

I joined NXP Research and pursuing a PhD was an important reason for that. Ad ten Berg was a very supportive department leader, despite the general business trends that were not in favor of scientific research. I also want to thank Clara Otero-Perez who continued this support as a department leader.

Doing research, and especially writing papers, is most of the time a solitary activity. When you can write papers together with others this makes the work much more pleasurable, as well as allows you to exchange experience. Therefore, I want to thank the co-authors of my papers, i.e. the persons that have not been mentioned above: Aart Matsinger, Arnaud Hubaux, Aylin Koca, Frank van der Linden, Jarmila Bökkerink, Julia Rubin, Mila Keren, Patrick Heymans, Quentin Boucher, Raphaël Michel and Tali Yatzkar-Haham.

Case studies are an essential pre-requisite for research in the engineering sciences, which often requires the help of persons that are taking part in these case studies, provide the required resources or give advice. Therefore, I would like to thank the following persons: Alan Hartman, Arjan den Hartog, Asaf Shaar, Bart Caerts, Gaëtan Delannay, Hans den Boer, Itay Maman, Jacques Flamand, Jean-Marie Trager, Jack Goossen, Jos Hegge, Michiel Mennen, Ofir Brukner, Peter van Loon, Rob van Ommering, Rob van Twist, Rob Wieringa, Shiri Kremer-Davidson, Somasundaram Venkatachalam, Uri Avraham and Wouter Batelaan.

I would like to thank the reading committee for providing me useful feedback on my thesis: Paris Avgeriou, Ivica Crnkovic and Hans van Vliet. I also want to thank the many anonymous reviewers of the journals and conferences. Often I received very useful feedback which helped me to improve my publications.

I would like to thank the many colleagues at Philips CFT, NXP Research and NXP R&D IT, who provided me a pleasant daily working environment without which it wouldn't be possible to work many evenings and weekends on writing the papers.

Some persons were specifically of importance for my PhD and during my career: Jos Gerards, Paul Meeuwissen and Thérèse Schoenmakers.

Over the years, I have got acquainted with many fine people, during my career or during my personal life, some of those have become good friends. I can recall many conversations on my research. While not all of my friends were able to understand the details of my research, the fact that I was able to share my ideas, and explain the objectives, helped me to formulate the contributions more clearly and encouraged me to continue with the research. For these conversations and for being good friends or fine colleagues, I want to thank: Abhijit Kumar Deb, Caro Afman, Carolyne Trew, Liesbeth Steffens, Manvi Agarwal, Peter Meyer, Peter Reuvekamp, Razvan Dinu, Renee van Liempd and Yanja Dajsuren.

I have always had the luck of being surrounded by a heartwarming family. Without their support, doing a PhD wouldn't have been possible. I therefore want to thank: Martha, Toolsi, Bida, Marie, Maureen, Erik, Ben, Liesbeth, Asha, Johan, Annie, Surita, Lauran Heleen, Remke, Soeris, Hans, Monique. Every time they asked me about my work and research, I regained new energy.

Finally I want to thank Amita, my wife, girlfriend and mate. Amita supported me all the way. She knew that doing a PhD was my longtime ambition. Many evenings and weekends I couldn't be the husband that I wanted to be, but she understood why I was doing this and accepted it. While doing the PhD and not being able to spend much time with her, I again realized what the most important thing in life is.

Contents

Abstract .... **v**

Acknowledgements .... **vii**

List of Figures .... **xiii**

List of Tables .... **xiv**

Chapter 1. Introduction **1**

1.1 Market Trends 1

1.2 Problem Statement 3

1.3 Research Objectives 3

1.4 Research Methodology 4

1.5 Research Questions 6

1.6 Thesis Outline and Article Overview 10

1.7 Applicability of the Research Results 14

1.8 Detailed Overview of Research Processes and Methods 14

1.9 Related Publications 18

1.10 Contributions to the Articles 19

Part I: Industry Structures for Software Development **21**

Chapter 2. The Changing Industry Structure of Software Development for Consumer Electronics **23**

2.1 Introduction 23

2.2 Background 25

2.3 Case Studies: Transition in the Development of Digital Televisions and Mobile Phones 29

2.4 Model of Industry Structures for Software Development in Consumer Electronics 31

2.5 Transitions in the Development of Digital Televisions 35

2.6 Transition in the Mobile Phone Industry 44

2.7 Related Work 50

2.8 Conclusions and Future Research 51
Chapter 3. Consumer Electronics Software Ecosystems 55
3.1 Introduction 55
3.2 Background on Consumer Electronics Products 57
3.3 A Classification of Ecosystem Types 58
3.4 Towards a Multi-Criteria Decision Support Method 60
3.5 Case Studies 68
3.6 Comparison with Related Art 78
3.7 Conclusions and Further Research 78
Part II: Variability Management in Software Supply Chains 81
Chapter 4. Using Context Variability to Model Multiple Product Lines 83
4.1 Introduction 83
4.2 Related work 85
4.3 Feature Models for Multiple Product Lines 85
4.4 Context Variability in Software Supply Chains 90
4.5 Merging MPL-Feature models 93
4.6 Comparison with Related Work 96
4.7 Preliminary experimental results 97
4.8 Discussion and Further Research 98
4.9 Conclusions 98
Chapter 5. Supplier Independent Feature Modelling 99
5.1 Introduction 99
5.2 Background 101
5.3 Integrating Feature Models 101
5.4 Supplier Independent Feature modelling 105
5.5 ZigBee Case Study 110
5.6 Tool support 112
5.7 Discussion and Alternatives 113
5.8 Related Work 114
5.9 Conclusions 114

Chapter 6. Using MDA for Integration of Heterogeneous Components in Software Supply Chains ... 117

6.1 Introduction ... 117

6.2 Problem Description ... 119

6.3 ZigBee Case Study ... 123

6.4 MDA for the Integration of Heterogeneous Components ... 127

6.5 Development Roles ... 137

6.6 Discussion and Further Research ... 138

6.7 Evaluation ... 141

6.8 Comparison with Related Art ... 143

6.9 Conclusions ... 144

Chapter 7. Evaluating a Textual Feature Modelling Language ... 145

7.1 Introduction ... 145

7.2 Related Work ... 147

7.3 TVL ... 149

7.4 Research Method ... 150

7.5 Results ... 156

7.6 Findings ... 159

7.7 Threats to Validity ... 163

7.8 Conclusion ... 164

Part III: Software Testing ... 165

Chapter 8. A Statistical Analysis of Operational Profile Driven Testing and Risk Based Testing ... 167

8.1 Introduction ... 167

8.2 Background ... 169

8.3 A Statistical Model to Estimate the Improvement on Test Efficiency ... 172

8.4 Simulations of the Test Efficiency when using Operational Profiles ... 174

8.5 Case Study: Philips HealthCare ... 177

8.6 A Statistical Analysis of Risk Based Testing. ... 180

8.7 Discussion and Further Research ... 184

8.8 Comparison with Related Art 185
8.9 Conclusions 186
Chapter 9. Risk Based Testing for Software Product Line Engineering 187
9.1 Introduction 187
9.2 Background 189
9.3 Applying Risk Based Testing to Product Line Engineering 192
9.4 Case Study: Philips Healthcare 195
9.5 Tool Support 198
9.6 Comparison with Related Art 202
9.7 Conclusions and Further Research 203
9.8 Appendix: Consistency Rules of Quantified Feature Models 204
Chapter 10. Towards a More Systematic Analysis of Uncertain User-Product Interactions 207
10.1 Introduction 207
10.2 Issues when Applying Operational Profiles for CE Products 209
10.3 Modelling Unexpected User-Product Interaction: an Initial Step 210
10.4 First implementation 213
10.5 Conclusion 219
Chapter 11. Conclusions 221
11.1 Research Questions and Answers 221
11.2 Threats to Validity 232
11.3 Generalization 233
11.4 Key Contributions of this Thesis 235
11.5 Further Research 237
Bibliography 241
Samenvatting 261
Curriculum Vitae 263

List of Figures

Figure 1 Research process 5
Figure 2 Article overview and their main relations 11
Figure 3 Changing structure in the computer industry 24
Figure 4 Example of a software supply chain 26
Figure 5 Model of industry structure types, visualized 32
Figure 6 Forces of moving from one industry type to another 34
Figure 7 Reference software architecture of a TV developed in a single company 37
Figure 8 Reference software architecture of a TV developed in a supply chain 39
Figure 9 Middleware and platform with glue layer 42
Figure 10 Split of glue software and platform 42
Figure 11 Anticipated mobile industry transition 46
Figure 12 Software architecture EMP 47
Figure 13 System architecture of consumer electronics products that support 3rd party applications 58
Figure 14 Classification of ecosystem types with examples from smart phones 59
Figure 15 Multi-Criteria Decision Support Method 61
Figure 16 Multiple product line feature diagram 86
Figure 17 Feature diagram car infotainment system 87
Figure 18 Context variability diagram of the car infotainment system 88
Figure 19 Multiple product line-feature diagram of the car infotainment system 89
Figure 20 Generating specialized feature models 91
Figure 21 MPL-feature diagram for CarA budget 92
Figure 22 MPL-feature diagram of CarA with merged infotainment system 95
Figure 23 Car-infotainment suppliers 100
Figure 24 Integrating functional areas 102
Figure 25 Feature models from the suppliers 102
Figure 26 Alternative subtrees 103
Figure 27 Merged models with dependencies 104
Figure 28 Composed SIFM and SSFMs (CSFM) 106
Figure 29 ZigBee reference architecture. 110
Figure 30 Infotainment high level component diagram. 120
Figure 31 Homogeneous and heterogeneous component integration. 120
Figure 32 Example feature diagram with links to development artifacts, together with its configuration space. 121
Figure 33 Layers in the ZigBee protocol with pofiles 124
Figure 34 Integration of Network and MAC layers 126
Figure 35 Model transformations for integration and configuration of components 128

Figure 36 Process overview 129
Figure 37 SVP's of the case study 130
Figure 38 Conceptual model of entities used 130
Figure 39 Meta-model for a glue component 133
Figure 40 Wizards for the interface map configuration process 134
Figure 41 Snippet library 135
Figure 42 Example of the generated glue code for the ZigBee case 136
Figure 43 Component composition and configuration with reachability analysis 139
Figure 44 Feature diagram of the PhoneMeeting voting component 146
Figure 45 Interview protocol 155
Figure 46 Fault tree of software failures 169
Figure 47 Example of a risk matrix 172
Figure 48 Risk matrix with 5 functions 182
Figure 49 Quantified feature model of a digital television 188
Figure 50 Example of a risk matrix 191
Figure 51 Feature model with linked components 192
Figure 52 Risk matrix for reusable components 194
Figure 53 Philips Medical Workspot platform development 196
Figure 54 Risk matrix for Philips Medical Workspot platform 197
Figure 55 Screenshot of a feature a family model with quantification and consistency checking 200
Figure 56 The enhanced user-product interaction model 212
Figure 57 ANOM results of the test data. 217

List of Tables

Table 1 Applicability of the research results 14
Table 2 Overview of types of ecosystems and their strengths and challenges 67
Table 3 Example of a decision support matrix 68
Table 4 Decision support matrix for gaming consoles 69
Table 5 Decision support matrix for digital photo cameras 71
Table 6 Decision support matrix for digital televisions and set-top boxes 72
Table 7 Decision support matrix for smart watches 74
Table 8 Decision support matrix for smart phones 75
Table 9 Decision support matrix for tablets 76
Table 10 Decision support matrix for personal computers 77

Table 11 ZigBee suppliers 111
Table 12 Feature set of selected suppliers 125
Table 13 Differences between the technologies used in the ZigBee case study 126
Table 14 Quality criteria for the evaluation of TVL 151
Table 15 Profiles of the 5 participants 153
Table 16 Results of the evaluation of TVL 157
Table 17 Definitions in software reliability 170
Table 18 Fictitious operational profile 174
Table 19 Comparison of MTBF with uniform versus OP distribution 175
Table 20 Comparison with large amount of test cases 176
Table 21 Operational profile for the orthopedists 179
Table 22 Simulations of allocating test cases to different medium type risks 183
Table 23 Comparison of risk analysis processes 195
Table 24 Collecting information related to user-product interaction 214
Table 25 Data collected from the consumer test. 215
Table 26 Results of ANOVA of the test data. 217