Requirements Scoping Visualization for Project Management

Krzysztof Wnuk[1] and David Callele[2]

[1] Department of Computer Science, Lund University, Lund, Sweden
Krzysztof.Wnuk@cs.lth.se,
WWW home page: http://www.cs.lth.se
[2] Department of Computer Science, University of Saskatchewan, Saskatchewan, Canada
callele@cs.usask.ca,
WWW home page: http://www.cs.usask.ca/

Abstract. Determining requirements process efficiency, and measuring the corresponding monetary impacts, is a challenging but necessary aspect of project management. In this paper, we perform an independent analysis of scoping decisions from a large industrial project with the goal of providing new visualizations that facilitate investigations of process efficiency, agility, and the effects of scoping decisions. The visualizations proposed in this paper can be used to analyze scoping dynamics and support process management decisions on a quantitative rather than a qualitative basis.

Key words: Requirements visualization, process evaluation, requirements scope, project management

1 Introduction

In Market-Driven Requirements Engineering (MDRE) [17], the time taken to deliver the product to market (and hence the overall release scheduling) is important and may strongly affect market success [4]. These time pressures place hard limits on all aspects of the development effort and force requirements efforts to be efficient and responsive [13]. Feature leapfrogging [20] between companies also imposes hard time constrains, which combined with resource constraints force requirements to be prioritized, with some requirements postponed for later implementation [11]. The process of selecting a subset of requirements for immediate implementation within a given project is called *scoping* and is considered a key activity for achieving economic benefits in product line development [19]. In MDRE, the scope of the project must adapt to competitive pressures and respond to changing market conditions in a timely manner – making appropriate scope decisions is a vital part of developing software systems that meet stakeholders' needs and expectations [11, 16].

Project management in this context has the goal of delivering a quality product within the given resource constraints, with appropriate risk management and acceptable predictability. Making decisions in a timely manner is fundamental for productive software management – unnecessarily delaying decisions can lead to wasted resources and can place other aspects of the projects at risk due to resource constraints [1]. However, as reported by Boehm and Sullivan [2] there is a "disconnect" between the process of making technical software product decisions and the value creation criteria of the organizations that develop these products. Moreover, the growing importance of software in all aspects of successful business demands better understanding of the relationships between technical properties of the decisions and the criteria for business value creation. In other words, software development should be considered an "investment activity" that aims at maximizing value creation for the resources invested [2].

Our prior work [25, 26, 27] showed that in MDRE there is the potential for the project scope to be constantly changing, or at least be under pressure to do so, particularly from unanticipated market forces. For example, the last minute addition of cutting-edge features and technologies to a project can lead to significant investments in requirements definition and feasibility analysis efforts. The associated project management risks may include excessive resource allocations that starve other efforts, destabilizing technologies being added to the product before they are ready, and even outright project failure. This prior work proposed new visualizations to help to assess the dynamics of scope changes in this context using a post-mortem analysis perspective.

In this paper, we analyze scoping from a project management perspective and propose new visualizations that support management decision-making. Using the action research strategy [18] the second author of this paper, who is an industrial Project Manager (PM) interested in Decision Support Systems (DSS), reviewed the previous work with the focus on the financial aspects of requirements engineering and requirements scoping and its implications on efficient project management. The project manager needs tools that can help him to identify opportunities for improvement *in the process itself* and to reduce risk by identifying wasted resources. As a result of this participatory effort, the following research questions are stated:

- What factors influence the process of removing features from the scope? (RQ1)
- What factors influence the late addition of features to the scope of the project? (RQ2)

In the remainder of this paper we review the related work and present a summary of the relevant results from our prior work (Section 2). We discuss the research approach and data used in this study in Section 3. Section 4 presents new visualizations that support financial analysis of the scoping process in projects. We discuss threats to validity for this study in Section 5 and close with our conclusions and identification of future work in Section 6.

2 Related Work

Release planning is an integral part of software product management. Multiple releases are considered in a release plan [23] and the selection of the 'right' requirements for a particular release is normally preceded by requirements prioritization [10, 11] and cost estimation [9]. Release planning is performed within the context of the product and corporate strategies, providing input to the feature selection process [21, 6, 12]. The feature selection process is not a trivial task; dependencies on other features [3] or frequent changes to the scope of the next release of a project [27] are typical complications. As a result, the selection process often becomes an uneasy compromise where the development efforts toward some features may need to be sacrificed (at the expense of wasted effort) in support of other features whose priorities have changed.

Managing this scope complexity is considered one of the core functions of software release planning and a key activity for achieving economic benefits in product line development [19]. While the importance of scoping has already been reported in several studies, most of the research is focused on the domain scoping aspect and the process of scope identification [19, 24]. When looking at product portfolio scoping [19] most techniques focus on the financial benefits associated with reuse across the product line [8].

The quantity of information that must be managed for large software projects is often overwhelming and visualizations can be useful in this context, offering more dimensions to represent than text [22]. Appropriate visualizations can assist with, for example, the requirements comprehension problem of gaining a quick assessment of the state of a set of requirements; a task typically impeded by the need to browse through disjoint textual requirements documentation and accompanying models [7]. The visualizations presented in our previous work [25, 26, 27] give a support assessment of the scoping process for large projects and can also be used for more in-depth analysis of the details of the scoping process. In [25] the Feature Survival Chart was introduced and applied to one large industrial project. In [27] the Feature Survival Chart was complemented with a set of scope tracking measurements while in [26] the Feature Transition Chart technique, designed to cover scope changes across the projects, was proposed and initially validated. These visualizations can be extended via metrics such as the volatility of the feature set and temporal measurements such as the time taken to cancel a feature [27]. In this paper, we extend the previous work, presenting a new set of visualizations that aim at supporting project managers in understanding the scoping dynamics and assess the monetary impact of the visualized scope changes.

3 Research Approach and Data in Context

The analysis performed in this study is based on empirical data from a large company that develops embedded systems for a global market. The company uses the software product line approach [14] based upon a common code base (referred to here as the platform) for the product line. A platform project follows the stage-gate model [5] with several increments; Milestones (MSs) and Tollgates (TGs) are used to control the project progress. In particular, there are four milestones for requirements management and design before the implementation starts: MS1 through MS4. The scope of a given project is based on a unit called a *feature*, a group of requirements that constitute a functional enhancement to the platform. At the beginning of a project, the feature definition typically contains a functional description and estimates of market value and development effort. To cancel or descope a feature in this context means to permanently remove the feature from the project plan. The discussions in this paper focus on activity within a current project, features re-introduced in a later project are outside of the scope of this work.

The elapsed time information for the project used in this work is presented in Table 1. The milestone criteria are presented in Table 2. We use the same metrics as in our prior work [27], summarized in Table 3 for convenience.

Table 1. The Project Timeline

Milestones	Project Analyzed
Elapsed Days	Launched May 2007
MS1	0
MS2	98
MS3	143
MS4	203

Table 2. Milestone Criteria

MS1	Potential features are drawn from the long-term roadmap documents. The initial scope (set of features) is defined and baselined. The scope is then documented and updated after the weekly meeting of the Change Control Board (CCB). The CCB is responsible for adding or removing features from the project plan.
MS2	Features are refined to requirements that are specified, reviewed and approved. Each feature typically contains ten or more requirements from various areas in the products. The feature requirements are forwarded to the design teams who return updated effort estimates.
MS3	Requirements are updated as necessary (based on design team feedback) and the effort estimates are refined.
MS4	The requirements work and design are finished. The final project scope has been negotiated with the development resources and the project is ready to start implementation.

The underlying data set has been transformed, as necessary, for the purposes of this paper to ensure that they are logically correct and consistent. The ana-

Table 3. Metrics

M1 *Number of positive and negative scope changes per time stamp/baseline.* A positive scope change means a feature was added, and a negative scope change indicates a feature was removed. *M1 is not used in the current work and is only included here for completeness.*

M2 *Time to feature removal.* The time from feature introduction until permanent removal.

M3 *Number of state changes per feature.* Number of times that the state of the feature in the scope was changed. In this work, we do not consider the initial inclusion to the scope as a scope change.

M4 *Time to feature addition.* The time from the start of the project until the feature was added.

M5 *Reason for feature removal.* A categorical metric focusing on reasons for removal due to project constraints.

lyzed project has 223 features considered during the analysis period. The project contained features that survived from project inception, features that were added during the project, and features that were canceled during the project.

For investigations involving M2, only features that were removed from the scope were included (120 data points). The values for M3 and M4 were calculated for all features in the analyzed project; features that survived the cancellation process were assigned a value of zero to remove them from consideration.

For investigations involving M5, 120 descoping decisions were analyzed and categorized. The entire set of 120 cancellation decisions was used during the M2-M5 and M4-M5 correlation analyses. However, some categories such as "dependent on supplier" or "inadequate feature description" contained such a small number of data points that we focused our analyses on the five categories with the most data points. The specific number of data points for each category is contained in the axis labels (*e.g.* (7dp) means 7 data points) for Figures 1, 2, 3 and 4.

4 Visualizations

The following sub-sections present new visualizations designed to assist a project manager in analyzing the requirements scoping process. The visualizations portray the relationships between the (normalized) metric data.

The underlying business decisions that generate the data sets are assumed to be rational in context – the decisions may not be perfect but the decisions were acceptable in the situation. Under this assumption, if a feature is going to be canceled, it is less wasteful of resources to cancel the feature sooner than later. While there may be cases where efforts to promote early feature cancellation may impede innovation, in this product line process the innovation decisions are made prior to the studied scenarios and are based on an ROI analysis rather than technical feasibility. This is not greenfield product development and innovation is incremental and often small in scope (*e.g.* a new software feature as compared to a new hardware platform).

4.1 The Relationship between the Time to Remove the Feature (M2) and the Reason for Feature Removal (M5) - addressing research question RQ1

The data set was partitioned based on the Reason for Descope of the feature and the resulting temporal distributions (normalized to MS4) are presented as box-whisker plots in Figure 1. We see that approximately 40% of the features were removed as a result of a stakeholder business decision with the mean time to remove the feature approximately 10% of the way into the final milestone of the requirements management process. These characteristics indicate that the feature list was pruned relatively aggressively and relatively quickly.

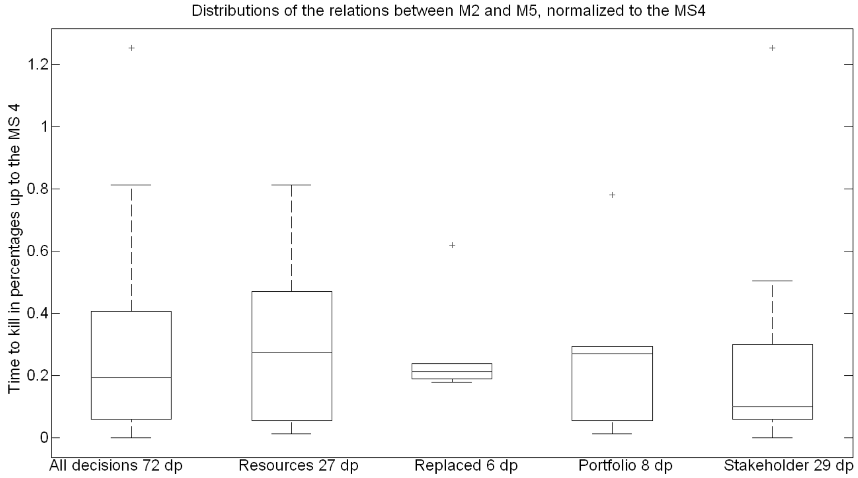

Fig. 1. The Relationship between the Time to Remove the Feature (M2) and the Reason for Feature Removal (M5), normalized to MS4

In contrast, approximately 40% of the features were not descoped due to lack of resources until approximately 30% of the way to MS4, on average, with some features being removed almost at the end of the project due to lack of resources also. This is a matter for concern to a project manager for it may be evidence that one or more of the following statements are true (this list is only exemplary and not exhaustive):

- There are difficulties estimating the scope (effort) required for a feature.

- There are difficulties determining what resources are available.
- There are difficulties estimating the contributions of the available resources.
- There are difficulties matching the available resources to the features (resource mismatch).
- The business process (algorithm) used to make the decisions may need improvement.

Simplistically, from the project manager's perspective, all features that are canceled waste scarce resources. It is imperative to make the keep/cancel decision for a feature as soon as possible to minimize this waste.

Next, the earliest stage of the project was investigated, the time between MS1 and MS2 to see if we could gain any further insight. The data set was partitioned again and only features that have been removed in the time between MS1 and MS2 were kept. The resulting data was normalized to MS2 and the results are shown in Figure 2.

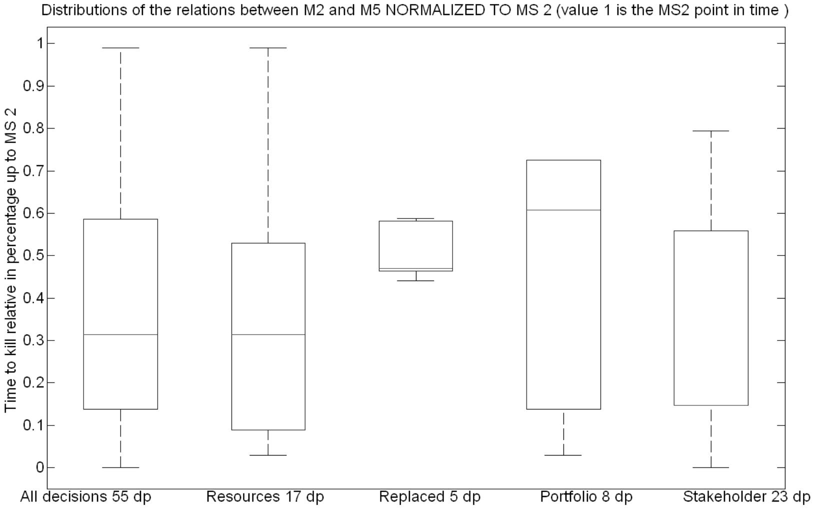

Fig. 2. The Relationship between the Time to Remove the Feature (M2) and the reason for Feature Removal (M5), normalized to MS2

The resulting data represents approximately 76% of the entire data set. We note that all portfolio changes (8/8), almost all replaced or renamed decisions (5/6) and most of the stakeholder business decisions (23/29) were made in this interval. This indicates that the project scope is being reduced in an effective manner, although portfolio changes and reduced or renamed decisions still lag the start of the interval by an average value of approximately 50%.

Only 17 of 27 resource decisions were made in this interval, again with an average delay in excess of 30% of the interval and a tail that extends to the end of the interval. While the situation is not as challenging as it appeared in Figure 1, there does appear to be evidence that improvements could be made.

4.2 The Relationship between the Time to Birth (M4) and the Reason for Feature Removal (M5) - addressing research question RQ2

Figure 3 illustrates the relationship between M4 and M5 for the analyzed project, normalized to MS4 for this project. We note that the medians for each partition show significantly greater diversity than those of Figure 1. The average time to feature introduction was approximately 50% of the timeline. We note that features introduced in the first half of the projects were predominantly cancelled due to lack of resources, replaced or renamed, or portfolio changes. Features introduced after the midpoint were predominantly cancelled as a result of stakeholder business decisons, although there is a much lesser contribution from a lack of resources.

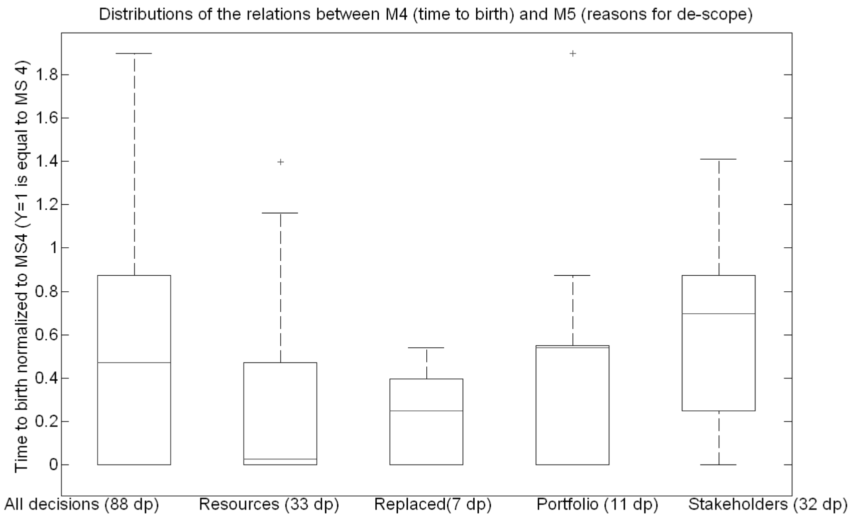

Fig. 3. The Correlation between the Time to Birth and The Reason for Cancellation, normalized to MS4 of the project

Figure 4 illustrates the relationship between M4 and M5 normalized to MS3 for this project. At this point in the project (MS3), the effort estimates are refined from the initial estimates at MS2. Again, we note that there is significant diversity in the median values. However, we are cautious of drawing strong

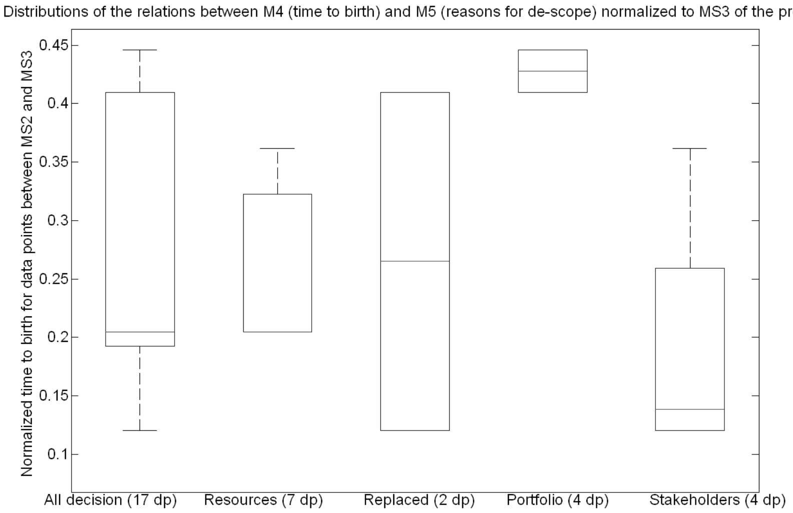

Fig. 4. The Correlation between the Time to Birth and The Reason for Cancellation, normalized to MS3 of the project

conclusions due to the relatively small data set. There is some indication of a relatively binary decision making process: cancel due to lack of resources or defer to a later date (portfolio changes + stakeholder business decision). We present this visualization to draw attention to the challenges faced with analyzing the tails of distributions in support of management decisions.

5 Threats to Validity

In this section, threats to validity are outlined and discussed based on the classification by Wohlin *et al* [28]. The causal conclusions drawn from our analyses were validated in a number of meetings with practitioners who confirmed that frequent (and sometimes late) scope changes are principally caused by specific market forces. However, more research is needed to validate the causal influence of the different reasons for adding and removing features from the scope of the project. The causal influence for the timing of when the features were included or excluded from the scope of the analyzed projects also requires further investigation.

The proposed visualization do not promote particular reasons for including or excluding features from the scope of the project. Therefore, researcher bias toward (or *fishing* for) a specific justification is minimized. Finally, although the example visualizations show four main reasons for removing features from the scope of the project, there is no theoretical limit to the number of attributes that can be visualized using this technique. As a result, the mono-operation threat for under-representation of the construct is minimized.

With respect to external validity threats, we would like to emphasize that while the investigated scenario has been observed in a specific company, using a specific development paradigm and releasing software products to a specific market, the identified issues of estimation challenges and changing the scope of the project are known and reported in the software engineering literature. We are aware that the approach should be validated in one or more independent contexts, yet we believe that the presented visualizations have sufficient support in the current context that they can be applied, with appropriate caution, in other than studied contexts.

6 Conclusions and Future Work

Software engineering, as an engineering discipline, should be guided by the goal of value creation, measured in terms that count for the enterprise that is investing the resources [2]. In order for software systems to best catalyze their potential for novel value generation, management must maintain a broad perspective to ensure that investments is software artifacts deliver an acceptable return. In this context, deciding which requirements to include into the scope of an upcoming project is crucial for the process of value creation. In a rapidly changing situation, such as MDRE, the failure to quickly cancel features or projects that new information shows are unlikely to succeed is a common example of failing to make a value-optimizing decision [2]. Improving the understanding of the connections between technical decisions and enterprise-level value maximization will enable software engineers and managers to make better choices.

In this paper, we propose a new visualizations that emphasize the relationship between the technical and financial aspects of scoping decisions. Utilizing a particular industrial example, with rapidly changing context and a high degree of uncertainty, we demonstrate methods for analyzing the impact of scoping decision on the financial aspects of the project and the company (RQ1). The visualizations help to understand the drivers of late scope exclusions and inclusions and can assist management efforts to control them (RQ2). The results from this paper support both diagnostic and predictive aspects of decision making [15] and have been expressed in a manner that supports strategic decision management for the project.

In future work, we hope to obtain other data sets that will allow us to further generalize this work. We are particularly interested in investigating whether new reasons for removing features from the scope of the project, other than those identified in this work, can be identified as influences on the scoping process. A larger dataset may enable us to propose a method for minimizing waste incurred by late scope removal; we are particularly interested in temporal optimizations based on cost-benefit analysis or return on investment. Formal statistical analyses of the current dataset are in progress; additional datasets are expected to strengthen the results of these analyses.

In the visualization domain, we plan to investigate investigate cost feedback using geometry (such as line thickness proportional to cost) and luminance (proportional to cost). Cost feedback can also be provided using cost as a function of assessed (predicted) risk.

Acknowledgments

This work is supported by VINNOVA (Swedish Agency for Innovation Systems) within the UPITER project.

References

1. *A Guide To The Project Management Body Of Knowledge: 4th edition.* Project Management Institute, 2009.
2. B. Boehm and K. Sullivan. Software economics: a roadmap. In *Proc. of the Conf. on The Future of Soft. Eng.*, ICSE '00, pages 319–343, New York, NY, USA, 2000. ACM.
3. P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag. An industrial survey of requirements interdependencies in software product release planning. In *Proceedings of the Fifth IEEE Int. Symp. on Req. Eng.*, RE '01, pages 84–, Washington, DC, USA, 2001. IEEE Computer Society.
4. J. Chen, R. R. Reilly, and G. S. Lynn. The impacts of speed-to-market on new product success: the moderating effects of uncertainty. *IEEE Transactions on Engineering Management*, 52(2):199–212, 2005.
5. R. G. Cooper. Stage-gate systems: A new tool for managing new products. *Business Horizons*, 33(3):44–54, 1990.
6. T. Gorschek, A. Gomes, A. Pettersson, and R. Torkar. Market-driven requirements engineering process maturity model. *Journal of Software Maintenance*, tba:tba, tba 2010.
7. O. C.Z. Gotel, F. T. Marchese, and S.J. Morris. On requirements visualization. In *Proc. of the Second Int. Workshop on Req. Eng. Visualization (REV 2007)*, pages 80–89, 2007.
8. P. E. Green and A. M. Krieger. Models and heuristics for product line selection. *Marketing Science*, 4(1):pp. 1–19, 1985.
9. M. Jorgensen and M. Shepperd. A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1):33–53, 1992.
10. J. Karlsson. *A Systematic Approach for Prioritizing Software Requirements. Doctorial Dissertation,.* PhD thesis, Linköping University, Sweden, 1998.
11. J. Karlsson and K. Ryan. A cost-value approach for prioritizing requirements. *IEEE Software*, 14(5):67–74, 1997.
12. M. Khurum and T. Gorschek. A systematic review of domain analysis solutions for product lines. *J. Syst. Softw.*, 82:1982–2003, December 2009.
13. C. McPhee and A. Eberlein. Requirements engineering for time-to-market projects. In *Proc. Ninth Annual IEEE Int. Conf. and Workshop on the Eng. of Computer-Based Systems*, pages 17–24, 2002.

14. K Pohl, G. Bockle, and F. J. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques.* SpringerVerlag, 2005.
15. J. C. Pomerol. Scenario Development and Practical Decision Making under Uncertainty: Application to Requirements Engineering. *Requirements Engineering*, 3:3–4, 1998.
16. B. Regnell, P. Beremark, and O. Eklundh. A market–driven requirements engineering process – results from an industrial process improvement programme. *Requirements Engineering Journal*, 3(2):121–129, 1998.
17. B. Regnell and S. Brinkkemper. *Engineering and Managing Software Requirements*, chapter Market–Driven Requirements Engineering for Software Products, pages 287–308. Springer, 2005.
18. C. Robson. *Real World Research.* Blackwell Publishing, 2002.
19. K. Schmid. A comprehensive product line scoping approach and its validation. In *in: 24th Int. Conf. on Soft. Eng. (ICSE 2002)*, pages 593–603, 2002.
20. J. Schumpeter. *Capitalism, Socialism and Democracy.* Harper, 1942.
21. Mikael Svahnberg, Tony Gorschek, Robert Feldt, Richard Torkar, Saad Bin Saleem, and Muhammad Usman Shafique. A systematic review on strategic release planning models. *Inf. Softw. Technol.*, 52:237–248, March 2010.
22. E. Tufte. *Envisioning Information.* Graphics Press LLC, 1990.
23. I. van de Weerd, S. Brinkkemper, R. Nieuwenhui, J. Versendaal, and A. A reference framework for software product management. Technical Report Technical report UU-CS, Volume: 2006, Issue: 014 (2006), Utrecht, 2006.
24. D. M. Wiess and C. T. R. Lai. *Software Product Line Engineering.* Addison-Wesley, 1999.
25. K. Wnuk, B. Regnell, and L. Karlsson. Visualization of feature survival in platform-based embedded systems development for improved understanding of scope dynamics. In *in: Third Int. Workshop on Req. Eng. Visualization (REV 2008)*, pages 41–50, 2008.
26. K. Wnuk, B. Regnell, and L. Karlsson. Feature transition charts for visualization of cross-project scope evolution in large-scale requirements engineering for product lines. In *in: Forth Int. Workshop on Req. Eng. Visualization (REV 2009)*, pages 89–98, 2009.
27. K. Wnuk, B. Regnell, and L. Karlsson. What happened to our features? visualization and understanding of scope change dynamics in a large-scale industrial setting. In *in: Proc. of the 17th IEEE Int. Req. Eng. Conference (RE 2009)*, pages 89–98, 2009.
28. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslen. *Experimentation in Software Engineering An Introduction.* Kluwer Academic Publishers, 2000.