Information and Software Technology 52 (2010) 237–248

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# A systematic review on strategic release planning models

Mikael Svahnberg *, Tony Gorschek, Robert Feldt, Richard Torkar, Saad Bin Saleem, Muhammad Usman Shafique

*Blekinge Institute of Technology, PO Box 520, S-372 25 Ronneby, Sweden*

ARTICLE INFO

*Article history:*
Received 22 June 2009
Received in revised form 13 November 2009
Accepted 18 November 2009
Available online 27 November 2009

*Keywords:*
Strategic release planning models
Systematic review
Road-mapping
Requirements selection factors

ABSTRACT

*Context:* Strategic release planning (sometimes referred to as road-mapping) is an important phase of the requirements engineering process performed at product level. It is concerned with selection and assignment of requirements in sequences of releases such that important technical and resource constraints are fulfilled.
*Objectives:* In this study we investigate which strategic release planning models have been proposed, their degree of empirical validation, their factors for requirements selection, and whether they are intended for a bespoke or market-driven requirements engineering context.
*Methods:* In this systematic review a number of article sources are used, including Compendex, Inspec, IEEE Xplore, ACM Digital Library, and Springer Link. Studies are selected after reading titles and abstracts to decide whether the articles are peer reviewed, and relevant to the subject.
*Results:* Twenty four strategic release planning models are found and mapped in relation to each other, and a taxonomy of requirements selection factors is constructed.
*Conclusions:* We conclude that many models are related to each other and use similar techniques to address the release planning problem. We also conclude that several requirement selection factors are covered in the different models, but that many methods fail to address factors such as stakeholder value or internal value. Moreover, we conclude that there is a need for further empirical validation of the models in full scale industry trials.



**Contents**



* Corresponding author. Tel.: +46 457 385811; fax: +46 457 27125.
*E-mail addresses:* Mikael.Svahnberg@bth.se (M. Svahnberg), Tony.Gorschek@bth.se (T. Gorschek), Robert.Feldt@bth.se (R. Feldt), Richard.Torkar@bth.se (R. Torkar), Aveeator@gmail.com (S.B. Saleem), Ushafique786@yahoo.com (M.U. Shafique).

0950-5849/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2009.11.006

## 1. Introduction

The idea of selecting an optimum set of features or requirements to deliver in a release within given constraints is called strategic release planning or road-mapping [1,30]. The purpose of strategic release planning is to balance between competing stakeholders' demands and benefits for the developing organisation according to available resources [30]. Strategic release planning is a complex problem, as appropriate understanding of planning objectives and other technical and non-technical constraints are required for a good release plan [30,32].

A strategic release plan is refined and re-planned after execution of a release as a consequence of updates and feedback from customers, defects in the previous release, market factors, new customer demands and other technical and non-technical requirement selection constraints [1,30]. Strategic release planning is considered important for both bespoke as well as market-driven software products [32,7]. In the context of bespoke products (i.e., where the customers are known and actively involved in the requirements engineering process), strategic release planning is useful for selecting the most valuable requirements of a customer in the first release and of diminishing importance in future releases [4,1]. In the context of market-driven products (i.e., where there is not a direct contact with customers) the importance of strategic planning is vital, as it helps in deciding which customer (among many competing customers) will get what features or requirements and in which release. Thus, strategic release planning relies extensively on selecting the right requirements to guide a product's evolution and to keep the product aligned with company strategies [3,8].

There are different approaches to develop a strategic release plan and update this plan through post release analysis [37,34,1]. Ad-hoc planning and systematic planning are two basic approaches used for strategic release planning. Some models are developed by combining traditional ad-hoc and systematic approaches named as hybrid approaches [32], but most models discuss release planning from different perspectives and consider different technical and non-technical factors of requirements selection [34,1,13,20]. Various models use a systematic (e.g., *Cost-Value Approach for Prioritising*) and some use a hybrid (e.g., *Evolve**) approach for release planning [20]. A few models are appropriate for strategic release planning with a limited planning scope (one or two releases in advance) and others are useful without any planning scope limitation [32]. Some models have appropriate tool support and these are considered useful in industrial settings, but there are also several models that have no tool support and those that are not validated in industry [34]. Among the validated models a few are partially validated in industry and some are being used in industry, such as e.g. Evolve (implemented in the form of the ReleasePlanner tool) (see e.g. [13,20]).

Each of the available strategic release planning models is based on different technical and non-technical factors of requirements selection [33]. Technical factors includes development tools, existing system architecture, technical precedence among requirements, features to include in a release, quality requirements (like security, performance, maintainability), requirements volatility, reusability and interdependencies (functionality, value and implementation oriented interdependency) between requirements [19,20]. Non-technical factors includes product strategy, business strategy, company strategy, product value, stakeholder value, priority of requirements set by stakeholders, maturity of the product, market place, required and available effort to implement requirements, delivery time of release, development cost estimation [4,9,7]. Producing a scalable strategic release planning model that deals with a few of these factors at a time is challenging. Considering technical together with non-technical factors in a holistic manner [7] is even more challenging.

A comparative analysis of existing models/approaches proved that most of the organisations are still using ad-hoc approaches for strategic release planning even for their large products [32,16], and thus the models proposed for release planning are not commonly adopted in industry. Saliu and Ruhe [34], tried to summarise these facts about release planning models, but they only analysed seven models with respect to a specific system and their scope was limited to models presented by academia. The current research aim in the area of strategic release planning models appears to be to improve and validate existing models or approaches [33]. For example models such as Evolve+ is an improved version of Evolve*, where more requirements selection factors are included, and appropriate tool support is also included in this version [1,19,13]. In terms of validity, models are being validated in different industrial cases to analyse the appropriateness of models in different situations [32,13,20].

In order to assist product managers in their choice of which method to use for strategic release planning, and what to consider in their decision, there is a need to know which models are available and their contribution towards strategic release planning. This is the contribution of this article: a systematic review of available strategic release planning models, their state of validation, and what requirements selection factors they propose. From an industry practitioner standpoint the results can be used to assess what factors the models use as input, but also to what level the model has been evaluated in industry. From an academic standpoint, the results can be used to map current state of the art and to contemplate what model input factors that are currently supported or missing, which may be valuable input for future improvement work.

The research questions are thus as follows:

*RQ1. What strategic release planning models have been presented?*

*RQ2. What technical and non-technical requirements selection factors are discussed in models found through RQ1?*

*RQ3. To what extent have the strategic release planning models in RQ1 been validated?*

*RQ4. Are the models from RQ1 intended to be used in a market-driven or a bespoke context?*

In this study, we use an inclusive definition of "strategic release planning". Most release planning methods are developed with the next release (or project) in mind. However, used appropriately they can become a product management tool for long term release planning. Given that few release planning methods have a more holistic view when it comes to product planning (combining company, product, and project views [7])[32] we have chosen to include rather than exclude methods. We thus adopt the definition used by Al-Emran and Pfahl [1]:

**Table 1**
Search terms construction process.

| Step | |
|---|---|
| 1 | Major terms are formed from the research questions by identifying the population, intervention, outcome, context and comparison |
| 2 | By altering the spellings, identifying alternative terms and synonyms of major search terms |
| 3 | By checking the keywords in some papers we already have |
| 4 | Boolean OR is used for incorporating search terms of alternative spellings and synonyms |
| 5 | Boolean AND is used to link the major terms with other terms and for combing different terms |

*Strategic release planning aims at assigning features to subsequent releases such that technical, resource, risk and budget constraints are met. Once a strategic release plan has been generated, i.e., a decision has been made on which features are to be developed in which release, operational release planning focuses on the development of the identified features in a single software release.*

The remainder of this article is organised as follows. In Section 2 we describe the planning phase of this systematic review. Section 3 explains how this research was executed, and the results of the systematic review. This is analysed in relation to the research questions in Section 4. The results are briefly discussed in Section 5, and the paper is concluded in Section 6.

## 2. Planning

In this section we describe the planning of the systematic review. We discuss the search strategy, the data sources used, the inclusion and exclusion criteria, and the overall methodology used to obtain the results.

### 2.1. Search strategy

All search results were documented to make the search process transparent and replicable [14]. For this purpose a systematic review search log was maintained. Similarly we kept track of selected studies and rejected studies.

Search terms were formulated in collaboration with a librarian. For constructing the search terms the steps in Table 1 was followed as suggested in [10], resulting in the set of search terms presented in Table 2.

In this study, we used the databases listed in Table 3. In addition, we also scanned the journal "International Journal of Hybrid Intelligent Systems".

### 2.2. Study selection criteria and procedures

Basic and detailed inclusion/exclusion criteria were defined for including studies and then selecting the most related studies for the purpose of data extraction. The basic inclusion criterion is to identify studies related to strategic software release planning models, a framework or a study with relevance to a strategic release planning model, a framework of post release analysis of strategic release planning, or any study related to a model framework of strategic release planning or post release analysis of a strategic release plan.

Detailed inclusion/exclusion criteria are presented in Table 4. These were applied to the studies identified using the basic inclusion criterion. From question 3, it can be discerned that literature reviews and systematic reviews were also to be included. Our strategy for dealing with these later was to use them to find the original studies that, in turn, should fit the basic and detailed inclusion/exclusion criteria, and not to include them in any further analysis. During the study, however, it became clear that we did not need to apply this strategy since we did not find any literature reviews or systematic reviews.

**Table 2**
Search terms.

| Search term | |
|---|---|
| 1 | Release plan |
| 2 | Release planning |
| 3 | Planning release |
| 4 | Software release plan |
| 5 | Software release planning |
| 6 | Planning software release |
| 7 | Strategic software release plan |
| 8 | Strategic software RP |
| 9 | Planning strategic software release |
| 10 | Retrospective/post release analysis |
| 11 | Requirements selection |
| 12 | Selecting requirements |
| 13 | Analysing software release defects |
| 14 | Managing software release |
| 15 | 1 OR 2 OR 3 OR 4 OR 5 OR 6 OR 7 OR 8 OR 9 |
| 16 | 5 OR 11 |
| 17 | 5 OR 6 OR 11 |
| 18 | 7 OR 8 OR 9 OR 11 |
| 19 | 5 OR 6 OR 7 OR 8 OR 9 OR 12 |
| 20 | 11 OR 12 |
| 21 | 4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 13 OR 14 |
| 22 | {1,2,4,5,7} AND 12 |
| 23 | {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} AND {Models, framework, Methods prototype, criteria, Techniques, Approaches} |
| 24 | {4, 5, 6, 7, 8, 9} AND Industry |
| 25 | {1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14} AND Market-driven |
| 26 | {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} AND Decisions |

**Table 3**
Databases used.

| Database name |
|---|
| IEEE Xplore |
| ACM Digital Library |
| Springer Link |
| Science Direct (Elsevier) |
| Engineering Village (Compendex, Inspec.) |
| Wiley-Inter Science |
| Business source premier |

Studies were selected individually by the researchers by applying the basic and detailed inclusion/exclusion criteria. The included studies were double-checked through discussions among the researchers. The basic and detailed inclusion/exclusion criteria were applied as follows. First, the basic inclusion criterion was applied by reading the titles, keywords and abstracts of all studies. If a study satisfied the conditions of the basic inclusion criterion then the study was included, and otherwise excluded. Second, the detailed inclusion/exclusion criteria were applied on the thus far included studies' abstracts, conclusions, introductions and sources of publication.

### 2.3. Quality assessment

Along with inclusion/exclusion criteria, it is also important to assess the quality of the included studies [14]. The purpose of quality assessment in this research is to further understand the limitations of each individual study during data synthesis. The criteria listed in Table 5 were used to evaluate the quality of selected studies, as recommended in other studies [14,15,10].

The quality criteria were used as a checklist while extracting data from the selected studies, and each question was answered with Yes or No. The quality assessment result of a particular study is summarised in Section 4.3. Moreover, since this was mostly used

**Table 4**
Detailed inclusion and exclusion criteria.

| | Study inclusion criteria |
|---|---|
| 1 | The article is peer reviewed |
| 2 | The article is available in full text |
| 3 | The article can be a literature review, systematic review, case study, an experiment, industrial experience report, survey, action research or comparative study |
| 4 | The article discuss a model/framework of strategic release planning or post release analysis of strategic release planning |
| 5 | The article will be included if it gives an overview of models/frameworks of strategic release planning or post release analysis of strategic release planning |
| 6 | The article will be included if it compares two or more models/frameworks of strategic release planning or post release analysis of strategic release planning with each other |
| 7 | The article will be included if it evaluates or analyse an existing model of strategic release planning or post release analysis |
| 8 | The article will be included if it discuss a validation of existing model of strategic release planning or post release analysis |
| | *Study exclusion criteria* |
| 1 | Articles that do not match the inclusion criteria will be excluded. |
| 2 | Articles related to only operational release planning will be excluded. |
| 3 | Articles related to re-planning of a release on operational level will be excluded. |

**Table 5**
Quality criteria.

| | Quality criteria |
|---|---|
| 1 | Is an appropriate introduction of strategic release planning or post release analysis of strategic release planning provided? |
| 2 | Is the research methodology clearly defined and appropriate for the problem under consideration? |
| 3 | Is the design of the study clearly stated and does it have proper conceptual argumentation based on references? |
| 4 | Does the research methodology map to study design, the study design to research questions, and the research questions to conclusions? |
| 5 | Are validity threats related to study results reported? |
| 6 | Are negative findings related to the model reported? |
| 7 | Are any restrictions or limitations on results of the study reported? |

as an internal quality measure to fully understand and being able to compare the state of validation between different studies (i.e., aid us in answering *RQ4*), these assessments are further explained in the data extraction form for each study.

### 2.4. Data extraction

We used a standardised data collection form to extract relevant information to answer the research questions. In Table 11 we present the Data Extraction Form. Extracted data was double-checked by two of the authors to eliminate uncertainties. A pilot study was performed on the data extraction form to ensure that it worked before conducting the full scale systematic review. Some difficulties were found and resolved through discussion among the authors. In case of multiple publications of the same data, the most recent results were used for data extraction and synthesis.

## 3. Execution and results

The process of searching for studies that match the basic inclusion/exclusion criteria was performed individually, although the inclusion/exclusion decisions were double-checked and discussed at each stage of execution.

Two literature resources (electronic databases and one journal) were scanned in this systematic review, and this was done in two separate phases. In the first phase each electronic database was scanned by applying the search terms. The basic inclusion/exclusion criterion was applied on the found results and related studies were selected. The information about the total number of results found from each electronic database against each search term, selected articles and rejected articles at each stage (by reading the title only and by reading the title and abstract) were logged in the systematic review search log. All search terms were applied on the specified electronic databases and a total of 12,541 results were retrieved.

From the 12,541 studies 3804 studies were excluded by just reading the title. The titles and abstracts of the remaining 8737 studies were read and the basic inclusion and exclusion criteria were applied, leaving 904 studies in the inclusion set. Finally, duplicates were removed, resulting in 124 remaining studies. In the next stage, the detailed inclusion/exclusion criteria were applied on the selected 124 studies. This resulted in the selection of 27 relevant studies. The other 97 studies were excluded.

In the second phase, one journal was manually scanned and one additional relevant study [31] was found.

In summary, 28 relevant studies were found through systematic review. Table 6 presents the number of results retrieved per database (total found and total selected), and Fig. 1 illustrates the systematic review process. Table 7 lists the selected articles.

**Table 6**
Results found per database.

| | DB name | Total found | Total selected |
|---|---|---|---|
| 1 | Engineering Village (Compendex, Inspec) | 3678 | 369 |
| 2 | IEEE Xplore | 636 | 134 |
| 3 | ACM Digital Library | 2711 | 126 |
| 4 | Springer Link | 2123 | 164 |
| 5 | Science Direct | 1370 | 34 |
| 6 | Wiley-Inter Science | 421 | 35 |
| 7 | Business Source premier | 1602 | 42 |
| | Sum | 12,541 | 904 |

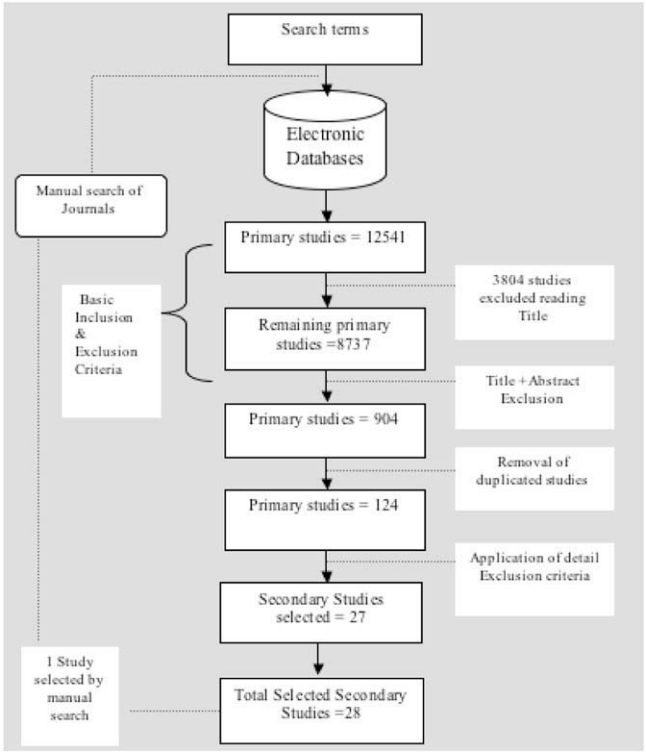


**Fig. 1.** Systematic review process.

**Table 7**
Articles selected from the systematic review.

| Id | Ref | Year | Study name |
|---|---|---|---|
| 1 | [11] | 1997 | A cost-value approach for prioritising requirements |
| 2 | [29] | 2003 | Quantitative studies in software release planning under risk and resource constraints |
| 3 | [28] | 2003 | Trade-off analysis for requirements selection |
| 4 | [27] | 2003 | An analytical model for requirements selection quality evaluation in product software development |
| 5 | [9] | 2004 | Software release planning: an evolutionary and iterative approach |
| 6 | [31] | 2004 | Hybrid intelligence in software release planning |
| 7 | [2] | 2004 | Intelligent support for software release planning |
| 8 | [21] | 2004 | Release planning under fuzzy effort constraints |
| 9 | [34] | 2005 | Supporting software release planning decisions for evolving systems |
| 10 | [35] | 2005 | Determination of the next release of a software product: an approach using integer linear programming |
| 11 | [22] | 2005 | Fuzzy structural dependency constraints in software release planning |
| 12 | [23] | 2005 | Measuring dependency constraint satisfaction in software release planning using dissimilarity of fuzzy graphs |
| 13 | [32] | 2005 | The art and science of software release planning |
| 14 | [30] | 2005 | Strategic release planning and evaluation of operational feasibility |
| 15 | [19] | 2006 | Release planning process improvement – an industrial case study |
| 16 | [17] | 2006 | Decision support for value-based software release planning |
| 17 | [13] | 2006 | Case studies in process improvement through retrospective analysis of release planning decisions |
| 18 | [6] | 2006 | An explanation oriented dialogue approach and its application to wicked planning problems |
| 19 | [12] | 2006 | Introducing tool support for retrospective analysis of release planning decisions |
| 20 | [18] | 2006 | A risk-driven method for extreme programming release planning |
| 21 | [38] | 2007 | An experiment with a release planning method for web application development |
| 22 | [38] | 2007 | A decision modelling approach for analyzing requirements configuration trade-offs in time-constrained web application development |
| 23 | [33] | 2007 | Bi-objective release planning for evolving software systems |
| 24 | [24] | 2007 | A system dynamics simulation model for analyzing the stability of software release plans |
| 25 | [26] | 2007 | A quality performance model for cost-benefit analysis of non-functional requirements applied to the mobile handset domain |
| 26 | [20] | 2008 | A systematic approach for solving the wicked problem of software release planning |
| 27 | [36] | 2008 | Software product release planning through optimization and what-if analysis |
| 28 | [25] | 2008 | Supporting road-mapping of quality requirements |

## 4. Analysis

### 4.1. Available strategic release planning models

*RQ1. What strategic release planning models have been presented?*

Twenty eight studies related to strategic RP models were found through the systematic review. In these studies, 24 models of strategic release planning are presented and the rest of the studies are related to validation of some of the presented models. Specifically, study number 15 is a validation of study 6, study 17 a validation of study 19, study 21 a validation of study 22. Moreover, study 25 presents the same model as study 28. Thus, 24 models are presented.

Of the 24 models found, 10 models are extensions of other models and 14 are original models, as can be seen in Fig. 2. It should, however, be noted that the original models are also often based on existing ideas and techniques. Twenty two of the models are used for strategic release planning or road-mapping and one model (PARSEQ [12]) is related to post release analysis of a strategic release plan. Another model (QIP [2]) is used for strategic release planning process improvement. It is also notable that most of the research on strategic release planning models has been done within the last 10 years.

A further analysis shows that there are three main groups of strategic release planning models; those that are related to the EVOLVE-family and the ReleasePlanner tool [20], those that are created by the SERG research group at Lund University, and those that are related to the Centre for Organization and Information at Utrecht University. In addition, there are a few other publications that are not related to any of these three main categories. Of the three categories, the EVOLVE-family is by far the largest, including 16 of the presented models.

Fig. 2 presents a map of the strategic release planning models. In this figure, we use the name of the model, where it has been named, and the title of the article where the model is not named. The models from Utrecht University are based on a model not related to strategic release planning, which we have included within parentheses in the figure. Each model is also identifiable via a number that corresponds to the id number in Table 7.

### 4.2. Requirement selection factors

*RQ2. What technical and non-technical requirements selection factors are discussed in models found through RQ1?*

All models provide different solutions of strategic release planning and discuss different requirements selection factors. Some of the models categorise requirements selection factors into groups, but most of the models do not discuss any categorisation of factors, but rather give a description and use of factors in the model. There are many common requirements selections factors among the majority of identified models.

In order to assist the analysis, we have created a taxonomy of the factors used by the different strategic release planning models, as presented in Fig. 3. This taxonomy is mainly created using terms and collections of terms used in the included studies. For example, the division into Soft Factors and Hard Constraints is influenced by the requirements selection factors used in Evolutionary Evolve+ [20].

In Fig. 3 we include information (in percent) about how many of the models that address each factor type, and in Table 8 we present which requirements selection factors that are addressed by each study. In this table, we list the factors used in the original study, together with a translation to the constructed taxonomy of requirements selection factors. Below, we briefly describe each of the factors further, and also introduce the acronyms used in Table 8.

#### 4.2.1. Hard constraints

Hard Constraints include those factors that may restrict the order and time when certain features or requirements can be implemented. The hard constraints include technical constraints, budget and cost constraints, resource constraints, effort constraints, and time constraints.

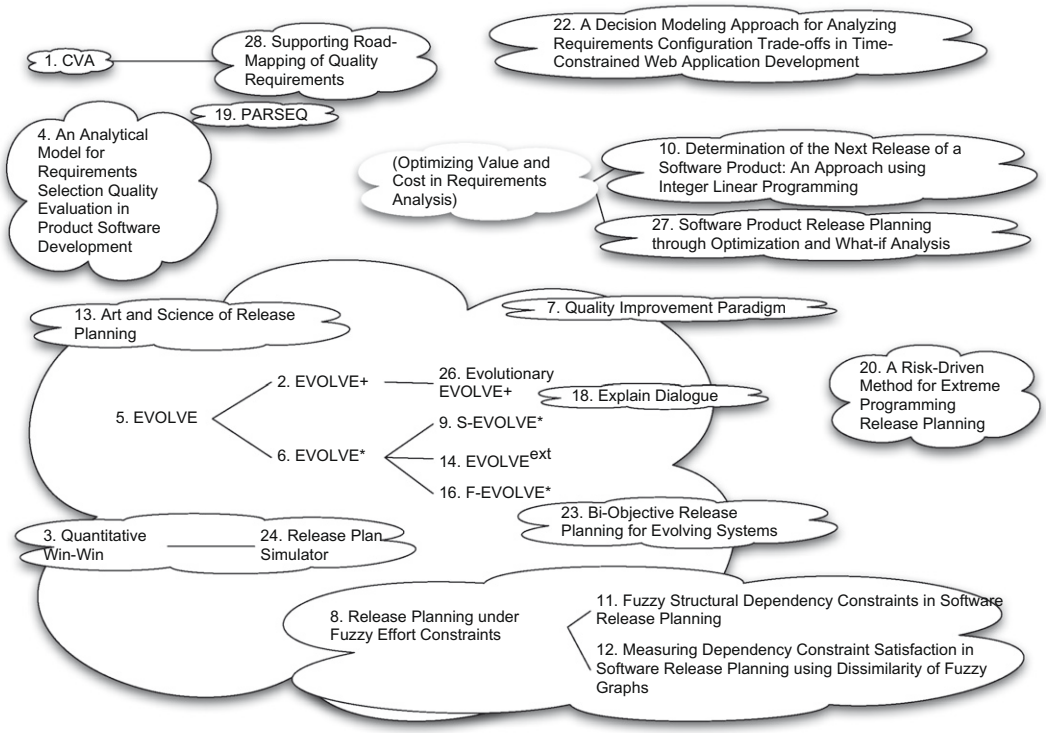


**Fig. 2.** Map of strategic release planning models.

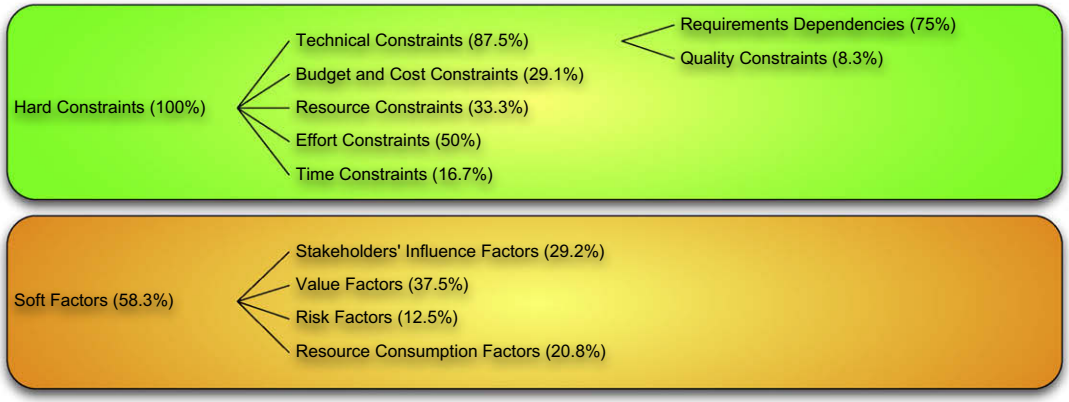


**Fig. 3.** Taxonomy of requirements selection factors.

*4.2.1.1. Technical constraints (TeC).* These constraints deal with constraints in the requirements themselves and the ability to implement them. For example requirements dependencies, which is a sub-category to this category, and the extent to which an existing system needs to be modified to accomodate the requirement.

*4.2.1.2. Requirements dependencies (RD).* Involves all constraints regarding the order in which requirements or features can be implemented, as well as dependencies that may influence the cost or value of requirements.

*4.2.1.3. Quality constraints (QC).* Constraints on quality requirements (or non-functional requirements), legal requirements, etc.

*4.2.1.4. Budget and cost constraints (B and CC).* All constraints that restrict the budget. Typically, these are expressed as cost constraints.

*4.2.1.5. Resource constraints (RC).* Constraints on the amount of resources that may be used during development.

*4.2.1.6. Effort constraints (EC).* Constraints on the amount of effort that can be spent during a development instance.

*4.2.1.7. Time constraints (TiC).* Constraints that mandate that certain requirements are released, resources used, or costs inflicted at certain times.

*4.2.1.8. Soft factors.* Soft factors include those factors that are more difficult to estimate and provide exact numbers on, but may cause certain features or requirements being prioritised higher than others. The soft factors include Stakeholders' influence factors, value factors, risk factors, and resource consumption factors.

*4.2.1.8.1. Stakeholders' influence factors (SIF).* All factors that deals with Stakeholders' ability to influence the release planning.

**Table 8**
Requirements selection factors per article.

| | | Hard constraints | | | | | | | Soft factors | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Technical constraints | | | | | | | | | | | |
| id | Factors in article | RD | QC | TeC | B&CC | RC | EC | TiC | SIF | VF | RF | RCF | Sum |
| 1 | Cost<br>Value<br>Stakeholders' satisfaction | | | | 1 | | | | 1 | 1 | | | 3 |
| 2 | Requirement dependency<br>Requirement effort estimation<br>Risk factors<br>Resource constraints | 1 | | | | 1 | 1 | | | | 1 | | 4 |
| 3 | Stakeholder preferences<br>Effort constraints<br>Time constraints<br>Quality constraints | | 1 | | | | 1 | 1 | 1 | | | | 4 |
| 4 | Requirement dependency<br>Budget restrictions<br>Requirements decomposition | 1 | | | 1 | | | | | | | | 2 |
| 5 | Required effort estimation<br>Requirement dependency<br>Stakeholder evaluation<br>Minimum release penalty<br>Maximum release benefit | 1 | | | | | 1 | | 1 | 1 | 1 | | 5 |
| 6 | Requirement dependency<br>Required effort estimates<br>Resource constraints<br>Budget constraints | 1 | | | 1 | 1 | 1 | | | | | | 4 |
| 7 | Requirement dependency<br>Required effort estimates<br>Resource constraints<br>Bottleneck resource constraints | 1 | | | | 1 | 1 | | | | | | 3 |
| 8 | Fuzzy constraints<br>Requirement dependencies<br>Effort constraints | 1 | | | | | 1 | | | | | | 2 |
| 9 | Stakeholders' value<br>Stakeholders' satisfaction<br>Technological constraints<br>Resource consumptions<br>Capacity bounds on resources<br>System's constraints | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 6 |
| 10 | Requirement dependency<br>Requirements' projected value<br>Requirements resource claim per development team | 1 | | | | | | | | 1 | | 1 | 3 |
| 11 | Structural constraints<br>Effort constraints | 1 | | | | | 1 | | | | | | 2 |
| 12 | Requirement dependency<br>Required effort constraints<br>Resource constraints | 1 | | | | 1 | 1 | | | | | | 3 |
| 13 | Feature dependency<br>Stakeholders' interests<br>Available resources<br>Feature prioritisations | 1 | | | | 1 | | | 1 | | | 1 | 4 |
| 14 | Requirement dependency<br>Stakeholder value<br>Time to market<br>Requirement volatility | 1 | | | | | | 1 | 1 | 1 | | | 4 |
| 16 | Resource capacity constraints<br>Time constraints<br>Feature dependency constraints<br>Implementation cost<br>Annual revenue per requirement | 1 | | | 1 | 1 | | 1 | | 1 | | | 5 |
| 18 | Requirements precedence constraints<br>Requirements coupling constraints<br>Resource constraints<br>Pre-assignment constraints<br>Effort estimation | 1 | | | | 1 | 1 | | | | | | 3 |
| 19 | Cost<br>Value | | | | 1 | | | | | 1 | | | 2 |
| 20 | Requirement dependency<br>Value in terms of cost or revenue<br>Cost of implementation<br>Effort per iteration<br>Business value | 1 | | | 1 | | 1 | | | 1 | | | 4 |
| 22 | Time estimates<br>Requirement dependency<br>Urgency of implementing a requirement | 1 | | | | | | 1 | | 1 | | | 3 |

(*continued on next page*)

**Table 8** (*continued*)

| id | Factors in article | Hard constraints Technical constraints | | | | | | | Soft factors | | | | Sum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RD | QC | TeC | B&CC | RC | EC | TiC | SIF | VF | RF | RCF | |
| 23 | Value from business perspective<br>Risk of implementing a feature<br>Feature dependency | 1 | | | | | | | | 1 | 1 | | 3 |
| 24 | Availability of resources<br>Required effort | | | | | 1 | 1 | | | | | 1 | 3 |
| 26 | Soft constraints<br>Hard constraints | | | | | | | | | | | | 0[1] |
| 27 | Development by one pool of developers<br>Development teams<br>Team transfers<br>External resource or dead-line extension<br>Requirements dependency | 1 | | | | 1 | | | | | | 1 | 3 |
| 28 | Quality of non-functional requirements<br>Cost of non-functional requirements | | 1 | | 1 | | | | | | | | 2 |
| Sum | | 17 | 2 | 1 | 7 | 10 | 12 | 4 | 6 | 10 | 3 | 5 | |

[1] In fact, we interpret this method to be generic enough to cover all categories, especially given that the factors used in the article are the influence for the two major categories in our taxonomy.

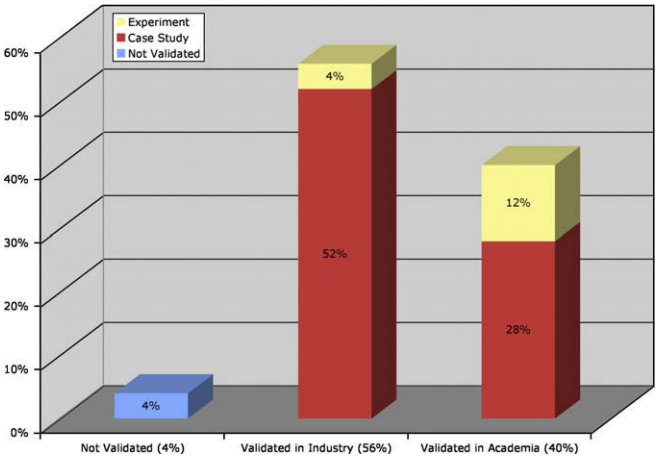


**Fig. 4.** Model validation.

*4.2.1.8.2. Value factors (VF).* Factors that help in assessing or maximising the value of a release.

*4.2.1.8.3. Risk factors (RF).* Factors that help in assessing the risk of requirements or features.

*4.2.1.8.4. Resource consumption factors (RCF).* This factor includes the estimated amount of resources that a requirement will consume. This can then be matched against several of the constraints such as budget and cost, effort, as well as resource constraints.

In summary, we can observe that all models deal with at least some of the hard constraints, whereas 40% do not deal with any of the soft factors. In addition, we can see that most models (87.5%) address technical constraints, and 50% of the models address effort constraints.

Since the EVOLVE-family is the largest group of strategic release planning models, it is interesting to study the requirements selection factors addressed by these models. An analysis reveals that although there is a larger emphasis of the hard constraints, members of the EVOLVE-family also address all of the soft factors. What is interesting, however, is that EVOLVE itself addresses most of the soft factors, whereas its direct derivatives (EVOLVE+ and EVOLVE*) do not address any of them. Only its "grandchildren" reappear among the soft factors, and only four of its "cousins" (i.e. [28,32,33,24]) address any of the soft factors.

### 4.3. State of validation

*RQ3. To what extent have the strategic release planning models in RQ1 been validated?*

In Fig. 4 and Table 9 (using the id's from Table 7 as index) we present details of the model validations. Please note that studies 15, 17, 21, and 25 are not included separately in this list, but are instead combined with their respective original model papers. The quality of each study is further indicated in Table 10, where the quality criteria from Table 5 are applied.

We can thus see that 23 models (96%) are validated and only one model (4%) is not validated. 56% are validated in industry,

**Table 9**
Model validation details.

| id | Validation details |
|---|---|
| 1 | Two industry case studies (interviews with project manager) |
| 2 | Industry case study (20 requirements, 5 stakeholders), academic experiment |
| 3 | Industry simulation (10 requirements) |
| 4 | Industry case study (Survey, 33 respondents) |
| 5 | Industry experiment (20 requirements) |
| 6 | Two industry case studies (first on 30 requirements and 3 stakeholders, second is deployment in a company) |
| 7 | Industry case study (25 requirements, 5 stakeholders) |
| 8 | Academic case study (30 requirements, 5 stakeholders) |
| 9 | Industry case study (49 requirements, 6 stakeholders) |
| 10 | Academic simulation (9, 24, 99 requirements, 3, 17, 17 teams respectively) |
| 11 | Academic simulation (25 requirements) |
| 12 | Academic simulation (10 requirements) |
| 13 | Academic case study (15 features, 2 stakeholders) |
| 14 | Industry case study (interviews) |
| 16 | Industry case study (30 requirements, 2 stakeholders) |
| 18 | Not validated |
| 19 | Two industry case studies (deployment in companies) |
| 20 | Industry case study (deployment on a project) |
| 22 | Academic experiment (63 participants) |
| 23 | Academic case study (33 requirements, 3 stakeholders) |
| 24 | Academic simulation (8 features, 6 stakeholders) |
| 26 | Industry case study (50 requirements, 6 stakeholders) |
| 27 | Academic experiment (2 software packages) |
| 28 | Industry case study (interviews, deployed in company) |

**Table 10**
Quality of selected studies according to quality assessment criteria.

| Id | 1. Is introduction provided? | 2. Is the research methodology defined? | 3. Is the design of the study stated? | 4. Is the study design cohesive? | 5. Are validity threats reported? | 6. Are negative findings reported? | 7. Is there any restrictions or limitations reported? |
|---|---|---|---|---|---|---|---|
| 1 | N | N | N | N | N | Y | Y |
| 2 | Y | Y | Y | Y | Y | N | Y |
| 3 | N | N | N | N | Y | Y | Y |
| 4 | Y | Y | Y | Y | Y | N | Y |
| 5 | Y | N | N | Y | N | N | N |
| 6 | Y | N | N | N | N | Y | Y |
| 7 | Y | Y | Y | N | N | N | N |
| 8 | Y | N | N | N | Y | N | Y |
| 9 | Y | N | N | N | N | Y | N |
| 10 | Y | N | N | N | N | N | Y |
| 11 | Y | N | Y | N | Y | N | Y |
| 12 | Y | N | Y | N | Y | N | Y |
| 13 | Y | Y | N | N | Y | N | Y |
| 14 | Y | Y | Y | N | Y | N | Y |
| 15 | Y | Y | Y | N | Y | Y | N |
| 16 | Y | N | N | N | N | N | N |
| 17 | Y | Y | Y | Y | Y | Y | Y |
| 18 | Y | N | N | N | N | N | Y |
| 19 | Y | Y | N | Y | Y | Y | Y |
| 20 | Y | N | N | N | Y | Y | Y |
| 21 | Y | Y | Y | Y | Y | Y | N |
| 22 | Y | N | N | N | N | N | N |
| 23 | Y | Y | Y | Y | Y | Y | Y |
| 24 | Y | Y | Y | N | Y | Y | Y |
| 25 | Y | Y | Y | N | Y | Y | Y |
| 26 | N | Y | Y | Y | Y | N | Y |
| 27 | Y | N | N | N | Y | Y | Y |
| 28 | N | N | N | N | N | N | Y |
| *Y* | *24* | *13* | *13* | *8* | *18* | *13* | *21* |
| *N* | *4* | *15* | *15* | *20* | *10* | *15* | *7* |

and 40% are validated in academia. Almost every model validated in industry is validated through case studies, except one model that is validated through an experiment. The case studies are carried out by conducting interviews with practitioners and by testing models in industrial contexts on a set of real requirements from industry.

This means that 44% of the models, i.e. those that are not validated or validated in academia, are never validated in any industry setting, not even pilot projects. More alarmingly, most of the models (including those that are validated in industry) are validated in a limited scale, i.e. only a few case studies are performed, or the validation is performed on only a small set of requirements. Reports of actual full scale industrial use (even in pilot projects) are only available for four of the 24 models.

For some models validation details are missing or not provided at all. It is thus difficult for readers to understand and trust the results of the model validations. This may also prevent industrial organisations to adopt a model, as results of model validations can not be generalised and the models are not validated in an industrial setting. Furthermore, only a few of the validation studies report negative findings as well as positive findings.

### 4.4. Intended context

*RQ4. Are the models from RQ1 intended to be used in a market-driven or a bespoke context?*

Most of the presented models provide decision support regarding requirements' selection for market-driven software development. The reason for this is likely to be that delivering a product in releases and developing road-maps (strategic release planning) is an important and common phenomena for market-driven software development. The results show that 83% of the models are considered to be useful for both bespoke and market-driven software development and the remaining 17% are appropriate only for market-driven software development.

## 5. Discussion

Despite the importance of release planning and that the complexity of the task requires support of methods, models, and tools, we find that there are in fact few real choices for practitioners that wish to adapt a release planning model. There are few models to begin with, and there are many family ties between the models that are proposed. Moreover, one of the key drivers for release planning in a market-driven context is to consider value for the customer, value for the company, and value at a given point in time, and decide upon a roadmap or release schedule according to this in order to maximise the profit of the company. However, our study finds that as many as 40% of the models do not focus on these soft factors, but only emphasise the hard constraints.

Furthermore, it is still a challenge to find models that are thoroughly validated; few models are tested in full scale industry trials. One critical issue for strategic release planning models is that of scaleability; a model may work well with a handful of requirements but may be impractical to use when the number of requirements and decisions grow. Moreover, strategic release planning models may be challenged when the requirements are less than perfect, and when stakeholder availability can not be assumed. In addition, studies in requirements engineering (e.g. [5]) indicate that the accuracy of e.g. requirements prioritisation may depend on whether the system in question matters to the subjects. Thus, while carefully planned academic evaluations with large sets of industry grade requirements may test certain aspects of strategic release planning models, the only real test would be to apply the models in real large scale industry cases. As shown in Section 4.3, these kinds of evaluations are rare.

The consequence of this is that it may be difficult to find a release planning model that suits a company's needs and addresses the desired requirements selection factors, and where there is also adequate empirical evidence that the method works as intended in full scale industry trials.

**Table 11**
Data extraction form.

| | | |
|---|---|---|
| *Fundamental information* | | |
| 1 | Data extractor | |
| 2 | Data checker | |
| 3 | Date of data extraction | |
| 4 | Article title | |
| 5 | Authors' Name | |
| 6 | Application domain | |
| 7 | Journal/conference/conference proceedings | |
| 8 | Retrieval search query | |
| 9 | Date of publication | |
| *Specific information* | | |
| 10 | Study context | Academia |
| | | Industry |
| 11 | Research methodology | Literature review |
| | | Systematic review |
| | | Case study |
| | | Experiment |
| | | Survey |
| | | Action research |
| 12 | Study subjects | Professional |
| | | Students |
| 13 | Validity threats | Conclusion validity |
| | | Construct validity |
| | | Internal validity |
| | | External validity |
| *RQ 1 What strategic release planning models have been presented?* | | |
| 14 | Name of presented model/framework | |
| 15 | Model / Framework proposed in Literature or in industry | |
| 16 | Newly presented model/framework or extension of already developed model/framework | |
| 17 | Means of representation (table, diagrammatically, mathematical means, logically) | |
| 18 | Description of presented model | |
| 19 | On what grounds the model/framework is constructed | |
| 20 | Model or framework use in Industry | |
| 21 | Any requirement selection technique used in the model | |
| 22 | Any limitation of the model/framework | |
| 23 | Practical application of model/framework in the form of tool | |
| 24 | Discussion about any other RP model/framework | |
| *RQ 2 What requirements selection factors are discussed?* | | |
| 25 | What technical and non-technical requirement selection factors are discussed | |
| 26 | Any other name of technical and non-technical requirement selection factors | |
| 27 | Common requirements selection factors discussed in two or more than two models/framework. | |
| *RQ 3 To what extent have the models been validated?* | | |
| 28 | Evidence of validation of the proposed model/framework (static validation or dynamic validation) | |
| 29 | Model/framework is validated in academia | |
| 30 | Model/framework is validated in industry | |
| 31 | Model/framework is validated in both academia and industry | |
| 32 | Model/framework is validity threats | |
| 33 | Model/framework is statically validated or implemented in industry | |
| *RQ 4 Which models have been used for bespoke and market-driven software development?* | | |
| 34 | Model/framework is proposed for bespoke only | |
| 35 | Model/framework is proposed for market-driven only | |
| 36 | Model/framework is proposed for both kinds of products | |
| 37 | Model/framework is adopted (in use) for bespoke product(s) | |
| 38 | Model/framework is adopted (in use) for market-driven product(s) | |

### 5.1. Strengths and weaknesses

A large number of articles have been covered in this systematic review in order to extract the articles listed in Table 7. We can thus be fairly sure that the systematic review actually covers the strategic release planning models that have been published to date. A possible weakness is the search terms (Table 2) that can, at least from a logical perspective, be simplified further and also be made more general. To make them as generic as possible, and in order for them to fit the databases used, they were developed in collaboration with a librarian. English is not the native language of any of the involved researchers in this study, so there is – as always – a risk that some of the papers have been misinterpreted during any of the involved stages. To counter this, all decisions and results were double-checked by at least one other person.

## 6. Conclusions

With the increase of market-driven software development, there is also an increasing need for conducting strategic release planning. Several well known models for strategic release planning exist, as well as several not so well known models. When deciding on how to conduct strategic release planning, it is important to be able to make an informed decision, weighing the strengths and weaknesses of several models against each other. To this effect, the systematic review presented in this article identifies the existing models and studies them from three aspects: (1) their coverage of requirements selection factors, (2) their degree of validation, and (3) their applicability for market-driven or bespoke software development. These three aspects are selected since they together give a good understanding of whether a model would be applicable in a given situation or not.

We have analysed the models in the aggregate in order to see overall trends. To this end, we present the found strategic release planning models in an overview map (Fig. 2). Moreover, we have created a taxonomy of requirements selection factors used (Fig. 3).

The principal findings from the systematic review on strategic release planning are:

- Twenty four strategic release planning models have been presented in academic papers. Sixteen of these belong to the EVOLVE-family of release planning models.
- Most methods focus on a limited set of requirements selection factors, with an emphasis on hard constraints. Approximately 58% of the models also include soft factors.
- Most of the presented models are validated, approximately half in industry and half in academia, and a large majority with the help of case studies (80%). Validation on full scale industry projects is scarce.
- All models are intended for market-driven software development. All but two models can also be used for bespoke software development.

Future work includes studying the use of strategic release planning models in industry. This will serve as an additional source of validation for the models found in this systematic review, and may also reveal other factors that need to be addressed. A more in-depth analysis of the identified strategic release planning models, and the process used in each of them is also part of future work.

Our recommendations can be divided into two parts; recommendations for model makers, and recommendations for model users. For model makers, we suggest an increased attention on soft factors and an increased attention to empirically validate the models. For model users, the level of empirical validation of models should be carefully analysed before deciding to use a specific model. Moreover, we suggest that model users define which requirements selection factors are needed, and to use this as a guiding input when selecting a strategic release planning model, since it cannot be expected that any model support all types of requirements selection factors.

## References


[1] A. Al-Emran, D. Pfahl, Operational planning, re-planning and risk analysis for software releases, in: Proceedings of the 8th International Conference on Product-Focused Software Process Improvements (PROFES), Lecture Notes in Computer Science LNCS 4589, Springer Verlag, Berlin, Germany, 2007, pp. 315–329.

[2] A. Amandeep, G. Ruhe, M. Stanford, Intelligent support for software release planning, in: Proceedings of the 5th International Conference on Product Focused Software Process Improvement (PROFES 2004), Lecture Notes in Computer Science LNCS 1840, Springer Verlag, Berlin, Germany, 2004, pp. 248–262.

[3] A. Aurum, C. Wohlin (Eds.), Engineering and Managing Software Requirements, Springer Verlag, Berlin, Germany, 2005.

[4] S. Barney, A. Aurum, C. Wohlin, A product management challenge: creating software product value through requirements selection, Journal of Systems Architecture 54 (6) (2008) 576–593.

[5] P. Berander, Using students as subjects in requirements prioritization, in: Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04), IEEE Computer Society, Washington DC, 2004, pp. 167–176.

[6] G. Du, J. McElroy, G. Ruhe. Ad-hoc versus systematic planning of software releases: a three-staged experiment, in: Proceedings of the 7th International Conference on Product-Focused Software Process Improvement (PROFES2006), Lecture Notes in Computer Science LNCS 4034, Springer Verlag, Berlin, Germany, 2006, pp. 335–340.

[7] T. Gorschek, A.M. Davis, Requirements engineering: in search of the dependent variables, Information and Software Technology 50 (1–2) (2008) 67–75.

[8] T. Gorschek, C. Wohlin, Requirements abstraction model, Requirements Engineering Journal 11 (2006) 79–101.

[9] D. Greer, G. Ruhe, Software release planning: an evolutionary and iterative approach, Information and Software Technology 46 (4) (2004) 243–253.

[10] J. Hannay, D. Sjøberg, T. Dybå, A systematic review of theory use in software engineering experiments, IEEE Transactions on Software Engineering (2007) 87–107.

[11] J. Karlsson, K. Ryan, A cost-value approach for prioritizing requirements, IEEE Software 14 (5) (1997) 67–74.

[12] L. Karlsson, B. Regnell, Introducing tool support for retrospective analysis of release planning decisions, in: Proceedings of the 7th International Conference on Product-Focused Software Process Improvement (PROFES 2006), Lecture Notes in Computer Science LNCS 4034, Springer Verlag, Berlin, Germany, 2006, pp. 19–33.

[13] L. Karlsson, B. Regnell, T. Thelin, Case studies in process improvement through retrospective analysis of release planning decisions, International Journal of Software Engineering and Knowledge Engineering 16 (6) (2006) 885–915.

[14] B. Kitchenham, Procedures for Performing Systematic Reviews, Technical Report TR/SE-0401, Keele University, 2004.

[15] B. Kitchenham, E. Mendes, G. Travassos, Cross versus within-company cost estimation studies: a systematic review, IEEE Transactions on Software Engineering (2007) 316–329.

[16] M. Lindgren, R. Land, C. Norström, A. Wall, Key aspects of software release planning in industry, in: Proceedings of the 19th Australian Conference on Software Engineering, 2008 (ASWEC 2008), IEEE Computer Society Press, Washington DC, 2008, pp. 320–329.

[17] S. Maurice, G. Ruhe, O. Saliu, A. Ngo-The, Decision Support for Value-based Software Release Planning, Springer Verlag, Berlin Germany, 2006. pp. 247–261.

[18] L. Mingshu, H. Meng, S. Fengdi, L. Juan, A risk-driven method for extreme programming release planning, in: Proceedings of the 28th International Conference on Software Engineering (ICSE 2006), IEEE Computer Society Press, Los Alamitos, CA, 2006, pp. 423–430.

[19] J. Momoh, G. Ruhe, Release planning process improvement – an industrial case study, Software Process: Improvement and Practice 11 (3) (2006) 295–307.

[20] A. Ngo-The, G. Ruhe, A systematic approach for solving the wicked problem of software release planning, Soft Computing – A Fusion of Foundations, Methodologies and Applications 12 (1) (2008) 95–108.

[21] A. Ngo-The, G. Ruhe, S. Wei, Release planning under fuzzy effort constraints, in: Proceedings of the Third IEEE International Conference on Cognitive Informatics, IEEE Computing Society, Los Alamitos, CA, 2004, pp. 168–175.

[22] A. Ngo-The, M.O. Saliu, Fuzzy structural dependency constraints in software release planning, in: Proceedings of the 14h IEEE International Conference on Fuzzy Systems, IEEE Computer Society Press, Los Alamitos, CA, 2005, pp. 442–447.

[23] A. Ngo-The, M.O. Saliu, Measuring dependency constraint satisfaction in software release planning using dissimilarity of fuzzy graphs, in: Proceedings of the Fourth IEEE Conference on Cognitive Informatics (ICCI 2005), IEEE Computer Society Press, Los Alamitos, CA, 2005, pp. 301–307.

[24] D. Pfahl, A. Al-Emran, G. Ruhe, A system dynamics simulation model for analyzing the stability of software release plans, Software Process Improvement and Practice 12 (5) (2007) 475–490.

[25] B. Regnell, R. Berntsson-Svensson, T. Olsson, Supporting road-mapping of quality requirements, IEEE Software 25 (2) (2008) 42–47.

[26] B. Regnell, M. Höst, R. Berntsson-Svensson. A quality performance model for cost-benefit analysis of non-functional requirements applied to the mobile handset domain, in: Proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2007), Lecture Notes in Computer Sience LNCS 4542, Springer Verlag, Berlin, Germany, 2007, pp. 277–291.

[27] B. Regnell, L. Karlsson, M. Höst, An analytical model for requirements selection quality evaluation in product software development, in: Proceedings of the 11th International Conference on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, CA, 2003, pp. 254–263.

[28] G. Ruhe, A. Eberlein, D. Pfahl, Trade-off analysis for requirements selection, International Journal of Software Engineering and Knowledge Engineering 13 (4) (2003) 345–366.

[29] G. Ruhe, D.Greer, Quantitative studies in software release planning under risk and resource constraints, in: Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE 2003), IEEE Computing Society, Los Alamitos, CA, 2003, pp. 262–270.

[30] G. Ruhe, J. Momoh, Strategic release planning and evaluation of operational feasibility, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005), IEEE Computer Society Press, Washington, DC, 2005, p. 313b.

[31] G. Ruhe, A. Ngo-The, Hybrid intelligence in software release planning, International Journal of Hybrid Intelligent Systems 1 (1-2) (2004) 99–110.

[32] G. Ruhe, M.O. Saliu, The art and science of software release planning, IEEE Software 22 (6) (2005) 47–53.

[33] M.O. Saliu, G. Ruhe, Bi-objective release planning for evolving software systems, in: Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2007), ACM Press, New York, NY, 2007, pp. 105–114.

[34] O. Saliu, G. Ruhe, Supporting software release planning decisions for evolving systems, in: Proceedings of 29th Annual IEEE/NASA Software Engineering Workshop, 2005, IEEE Computer Society Press, Washington, DC, 2005, pp. 14–26.

[35] J.M.V. vanden Akker, S. Brinkkemper, G. Diepen, J. Versendaal, Determination of the next release of a software product: an approach using integer linear programming, in: Proceeding of the 11th International Workshop on

Requirements Engineering: Foundation for Software Quality (REFSQ 2005), 2005, pp. 119–124.
[36] M. vanden Akker, S. Brinkkemper, G. Diepen, J. Versendaal, Software product release planning through optimization and what-if analysis, Information and Software Technology 50 (1-2) (2008) 101–111.
[37] C. Wohlin, A. Aurum, What is important when deciding to include a software requirement into a project or release, in: Proceedings of the International Symposium on Empirical Software Engineering, Noosa Heads, Australia, 2005.
[38] S. Ziemer, I.C. Calori, An experiment with a release planning method for web application development, in: Proceedings of the 14th European Conference on Software Process Improvement (EuroSPI 2007), Lecture Notes in Computer Science LNCS 4764, Springer Verlag, Berlin, Germany, 2007, pp. 106–117.