Analyzing an Industrial Strategic Release Planning Process – A Case Study at Roche Diagnostics

Gabriele Zorn-Pauli[1], Barbara Paech[1], Tobias Beck[1],
Hannes Karey[2], and Guenther Ruhe[3]

[1] University of Heidelberg, Im Neuenheimer Feld 326, 69120 Heidelberg, Germany
{zorn-pauli,paech}@informatik.uni-heidelberg.de,
tobias.beck@stud.uni-heidelberg.de
[2] Roche Diagnostics GmbH, Sandhofer Strasse 116, 68305 Mannheim, Germany
hannes.karey@roche.com
[3] University of Calgary, AB T2N 1N4, Canada
ruhe@ucalgary.ca

Abstract. **[Context and motivation]** Strategic release planning (SRP) for a globally used information system is a challenging task. Changes to requirements on different abstraction levels are arriving continuously and have an impact on long-term selected features. **[Question/problem]** The major question is how to successfully do SRP to create competitive advantage. **[Principal ideas/results]** An exploratory case study in an industrial context was conducted (1) to get a deeper understanding of the as-is SRP process in practice, (2) to evaluate the suitability of a to-be SRP process, introducing the EVOLVE II method and corresponding ReleasePlanner tool and (3) to gather additional requirements for the to-be SRP process, with respect to feature generation and feature selection. **[Contribution]** In this paper we describe the case study and present lessons learned to improve and customize a SRP process in practice. In particular, we propose the Requirements Abstraction and Solution Model (RASM) to support feature generation.

Keywords: strategic release planning, product roadmapping, long-term feature selection, feature generation, requirement abstraction, decision-support.

1 Introduction

Software release planning focusses on the decision which features to assign to which consecutive future product releases. Strategic release planning (SRP), also called release roadmapping, is used to link business or organizational strategies and solution planning to support long-term product feature selection [11]. For this, SRP aims at long-term feature assignment to subsequent releases fulfilling technical, resource, risk and budget constraints [12]. Compared to SRP, operational release planning focuses only on the development of the next software release, planning the implementation of the identified features [1].

More and more demanding customer needs in a volatile and globally operating business environment require more agility with respect to strategic product planning [19].

J. Doerr and A.L. Opdahl (Eds.): REFSQ 2013, LNCS 7830, pp. 269–284, 2013.

Therefore, to be successful in the future an integrated approach for strategic decision-making, requirements management, and roadmapping processes is required [14]. SRP has to cope with unclear or high-level business requirements and continuously arriving changes to requirements on different abstraction levels that have an impact on existing release plans. Furthermore, competitive customer needs and varying implicit multiple feature selection criteria make SRP difficult. Additionally, the need of considering relevant changes faster increases the complexity of the SRP decision-making process. Since the feature concept is most often used for SRP purpose, the SRP process in practice is characterized by two major decision-makings: (a) *feature generation*, means bottom-up bundling of requirements on a lower abstraction level into features or top-down dividing business or organizational strategies into features, (b) *feature selection*, which means assigning features to subsequent releases based on multiple selection criteria.

In academia several SRP processes exist, where the planning item (e.g. feature) generation is neglected and planning item backlogs and corresponding requirement engineering tasks are taken for granted. Lethola et al. [12] states that the roadmap preparation process for release roadmaps consists of the following four steps: data collection, feature prioritization, release planning, and release roadmap validation, where the feature generation task is not addressed. Van de Weerd et al. [21] provide a reference framework for software product management, where release planning and requirements engineering are identified as separated key process areas. Therefore, product release planning starts with requirement prioritization. Svahnberg et al. [20] conducted a systematic literature review on strategic release planning processes, where all found processes focus on the requirement selection task. However, the feature generation task in practice is an essential part of the SRP process since features are generated in practice top-down and bottom-up. Further, due to the growing number of requirement changes and requirement volatility, which is reflected in the increasing adoption of agile software engineering methods in practice, the problem of overscoping, [4] arises. This requires the integration of strategic release re-planning decision-support to adequately adapt existing plans [9]. In particular, relevant changes to requirements on different abstraction levels have to be identified and aggregated in existing feature structures and validated against multiple business strategies. Currently, little is known about the application of SRP processes in practice [19]. Additionally, there is a need for further empirical validation of existing models in full-scale industry trials.

In this paper we report the results of an exploratory case study in industry and provide the following contributions: (1) a deeper understanding of the as-is SRP process in practice, with focus on the feature generation and feature selection decision-makings, (2) evaluation results on the suitability of a to-be SRP process, introducing the EVOLVE II method [15] and corresponding ReleasePlanner tool and (3) additionally gathered requirements for the to-be SRP process with respect to feature generation and feature selection decision support. The remainder of this paper is structured as follows: Section 2 presents related work. Section 3 describes the case study design and how we proceeded in the case study, while Section 4 presents the case study results. Section 5 discuss the results and presents lessons learned. Section 6 concludes the paper and gives an outlook on future work.

2 Related Work

The ability to successfully do SRP creates competitive advantage. *Selecting a subset of requirements for realisation in a certain release is as complex as it is important for the success of a software product* [5]. Suomalainen et al. [19] provide new empirical results about product roadmapping in volatile business environments, by defining main stakeholders and their roles and by proposing a product roadmapping process framework. The identified most problematic phases of the process are prioritizing features, managing changes and maintaining roadmaps. Through an interview study practitioners were asked about their feature capturing methods and sources. Market trends and standards were stated as the main feature source and the most commonly used method for capturing features was gathering ideas over time.

Bjarnason et al. [4] conducted an empirical interview study about the causes and effects of overscoping, setting a release scope that is too large to deliver in time, in a large-scale industrial setting. They identified six causes for overscoping, where for instance unclear business strategies for software development and continuously incoming requirements flow via multiple channels are stated. Danesh et al. [6] also conducted a qualitative study to increase the understanding of software release planning challenges in several software companies and states that unclear project objectives and frequent change of these objectives are key factors for release planning failures. The difficulties with linking business strategy to solution planning was reported by Komssi et al. [11] investigating the roadmapping process of two Finnish software product companies. An interesting suggestion was a focus shift away from low level software feature prioritization to the analysis of high level customers' business process activities. Komssi et al. see the benefit of discovering new service business opportunities and competitive advantage. Similar investigations according to linking product strategies are conducted by Khurum et al. [10] who developed a method for alignment evaluation of product strategies among stakeholders to ensure that strategies are the basis for planning and development of products.

In literature there are several strategic release planning processes proposed. Svahnberg et al. [20] provides a systematic literature review on 24 strategic release planning processes. The results show that more than 60 % of the presented academic papers belong to the EVOLVE family and most of them could be applied for market-driven and bespoke development. Svahnberg et al. also investigated the state of validation of the SRP processes and concludes that most of the processes are validated in industry with limited scale. An additional industrial proven release planning approach, that was not covered by the systematic literature review, is proposed by Fricker et al. [7]. The major idea is to simplify release planning by utilizing feature trees to structure requirements, instead of using flat requirements lists. The approach was also evaluated in an industrial case study with respect to feasibility.

3 Case Study Design

In this section we provide information about the two case study objects, describing the context of the company under consideration, the case project and the ReleasePlanner tool.

3.1 The Case Company

The case study was conducted in the context of a globally operating company in the health care domain that develops in-house a bespoke and globally used Customer Relationship Management (CRM) system. The case study aims at investigating the SRP process of a CRM subsystem called Global Deal Calculation (GDC), that implements parts of the Contract Life Management cycle. Agile software development methodologies and in particular the *Scrum* framework is used to incrementally develop the GDC subsystem by releasing two minor releases and several patches per year and a major release every three to four years. Due to the adoption of agile development methods the release cycles are partitioned into several iterations. That facilitates communication and negotiation possibilities with the stakeholders after every iteration to adapt existing release plans. An issue tracking system is used to submit requirements, such as bug, change or features requests to the development team. As the number of iterations is varying, the release duration also varies. The company is already in a transition to adapt agile software development practices such as Scrum and not all projects are done in an agile manner. Project management and release roll-outs are still conducted plan-driven, which causes a mix of agile and plan-driven elements. Additionally, not every release version is consumed by all company sites countries, because a roll-out project causes high testing and training effort.

The GDC system is used by different, geographically distributed company sites and corresponding country business units. Primarily, GDC is globally developed, providing standardized functionality, that is used by all countries, but is implemented locally by providing additionally country specific functionality. Since the number of GDC *consumer* countries is growing up to 17 countries in the future, the complexity of linking multiple country strategies to system solution planning requires a systematic SRP method. Some specific challenges of the company, like planning a bespoke and globally used information system considering multiple business strategies, have already been presented in [23].

Several stakeholder boards or teams on different management levels involve IT and business representatives for SRP. The *Change Advisory Board (CAB)* reviews and proposes the project portfolio and major release changes. CABs are also responsible for decisions and prioritization of changes that have been escalated by the Change Management Teams. The *Change Management Team (CMT)* is the global business process owner and prioritizes business requirements, reviews projects and budgets, makes trade-off decisions, discusses strategic and escalated operational issues. The *Iteration Review Group (IRG)* is responsible for the operational release planning by reviewing and approving planned iterations of the current release. The IRG involves also CMT members and the product owner who is primarily responsible for generating and managing release plan proposals. In general, strategic release plan relevant changes on corresponding feature sets are welcome to decrease reaction times on changing business or organizational needs to benefit from IT-enabled competitive advantages.

3.2 ReleasePlanner Tool

The ReleasePlanner[1] is a proprietary, web-based process and decision-support tool. We have chosen the ReleasePlanner, because the tool was proven successfully in about 25 industry and more than 250 academic and student projects (f.i. [16], [3], [18]). It allows prioritizing release objects by multiple stakeholders against multiple criteria and performing subsequent resource optimization to maximize the overall release value for a release period of typically more than just one release. The decision support process is based on an evolutionary problem solving approach called EVOLVE II [15], which is emphasizing the involvement of human experts. The approach comprises 13 different process steps, all of them supported by ReleasePlanner. At each iteration, five optimized and diversified planning alternatives are determined. The final decision is done based on additional aspects such as resource consumption profiles of the proposed alternatives. In addition, implicit concerns not being part of the explicitly formulated model are supposed to be included by the human expert in the selection process.

3.3 Research Methodology

An exploratory case study was planned and conducted based on the guidelines for case study research in Software Engineering by Runeson et al. [17]. The objective of the case study was threefold. (1) Understanding of the as-is SRP process in practice to identify problems and improvement capabilities for a to-be SRP process. (2) Providing a to-be SRP process proposal adopting the EVOLVE II method. (3) Evaluation of the to-be SRP process applying the ReleasePlanner, that implements EVOLVE II. Therefore, the following research questions are investigated with respect to the conducted industrial case study. *(RQ1) How is strategic release planning done in the company? (RQ2) How does the EVOLVE II method and corresponding ReleasePlanner cover/extend the as-is SRP process of the company? (RQ 3) What are additional requirements for the SRP to-be process?* To answer the research questions the case study provides results on qualitative and quantitative data. Table 1 shows the data collection strategy by illustrating which data was collected utilizing the respective data collection method.

3.4 Threats to Validity

Threats to the validity of empirical research have to be examined during all phases of the case study. To evaluate the validity of this case study, the validity perspectives proposed by Wohlin et al. [22] were considered and are analyzed in the following. The threats to *construct validity* are reduced by a cooperation with the industry partner over more than a year and by reviewing the research results by the practitioners in a focus group session and informal discussions to ensure that the studied parameters are relevant to the research questions. Further, the threats to *internal validity* are reduced by triangulation (see Table 1) over multiple empirical data sources and the combination of qualitative and quantitative data. For instance, reflecting the as-is SRP process by a retrospective data analysis helped to validate identified implicit feature selection criteria based on

[1] `https://www.expertdecisions.com`

Table 1. Data source collection strategy

Method	**Data Source**	**Research Question**
Archival Data Analysis (July-Aug 2012)	Feature backlogs; meeting notes and release notes of the last three GDC release versions	RQ 1
Observation of a requirements refinement and prioritization meeting (18.07.2012)	12 Participants: 9 country business representatives, product owner, Observer: first and third author	RQ1 RQ3
Focus Group Session (06.09.2012)	5 Participants: Product Owner, IT Consultant, IT Project Manager, Moderator: first author, Observer: third author	RQ 1 RQ 2 RQ 3
Simulation (Aug-Sep 2012)	Retrospective release planning (GDC 3.6 and 3.7) simulation using the ReleasePlanner	RQ 2 RQ 3

data analysis against feature selection criteria stated by the practitioners. Additionally, the data were collected by two researchers, which reduces the risk of being biased by one person. Finally, the threats to *external validity* are reduced by conducting the case study in a real-world industrial setting. However, the external validity might be still influenced by the studied specific context represented by the mixture of agile and plan-driven methods.

4 Results

In this section, the results of the case study are presented. First, in Subsection 4.1 the as-is SRP process is described thus to provide a deeper understanding of the as-is SRP process in practice (RQ1). Thereafter, Subsection 4.2 provides evaluation results on the suitability of the to-be SRP process, introducing the EVOLVE II and corresponding ReleasePlanner tool, are presented (RQ2). Additionally gathered requirements for the to-be SRP process with respect to feature generation and feature selection are described in Subsection 4.3 to answer RQ3. Finally, the developed Requirements Abstraction and Solution Model (RASM) is introduced in Subsection 4.4.

4.1 Understanding the As-Is Strategic Release Planning Process

To describe how SRP is done by the company (RQ1) in general, Figure 1 outlines the major identified SRP process elements. A *heterogenous requirement pool* comprises requirements on different abstraction levels, where requirements and changes to requirements continuously arrive during the SRP process. This requirement pool is the basis for *feature generation*, where this step also comprises the pre-selection of features to scope the *feature backlog*. After that, the features contained in the feature backlog are assigned to subsequent releases based on multiple *feature selection* criteria. The generated *release roadmap proposals* are basis for stakeholder negotiation and have to be *re-planned* after every release iteration cycle to accommodate intermediately occurring changes.

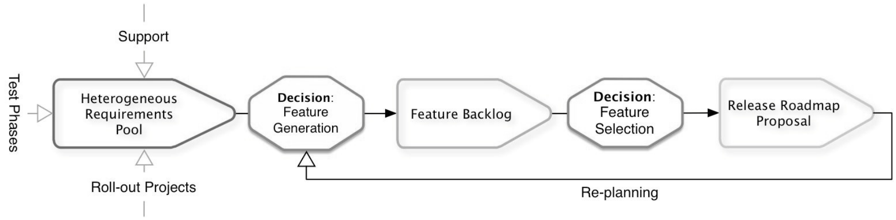
Support
Test Phases
Heterogeneous Requirements Pool
Roll-out Projects
Decision: Feature Generation
Feature Backlog
Decision: Feature Selection
Release Roadmap Proposal
Re-planning

Fig. 1. Identified as-is strategic release planning process

Feature Generation. As mentioned above, the basis for SRP is a heterogeneous requirements pool that comprises all requirements related information used for release planning. We have used the Requirements Abstraction Model (RAM) provided by Gorschek et al. [8] to classify the requirements of the company that are available on different abstraction levels. The RAM was not used so far by the company for requirements engineering or release planning purpose. Table 2 shows sources of requirement relevant information, provides examples and classifies the requirements according to the RAM. The CAB is primary source for business and IT strategy concerns represented in business cases and corresponding IT roadmaps. The CMT assigns *main topics* to future releases, where main topics are used to communicate business strategies within release roadmaps. These main topics could be understood as features and are related to business case initiatives. The analysis of the archival release planning data showed, see Table 3, that release roadmaps for 3.6 comprised only low level requirements, where 3.7 comprised a combination of high level (main topics) and low level business requirements. Starting with 3.8 the feature concept, which groups low level business requirements, was adopted to reduce planning complexity. The requirement abstraction reduces communication and negotiation necessity, because only feature related changes were communicated (e.g. recently identified features). The IRG is responsible to review the release roadmaps after every release iteration duration (7 weeks) to discuss and negotiate changes. There are three different input channels, as shown in Figure 1, for requirements on a lower abstraction level. During (i) roll-out projects, (ii) support and (iii) test phases, requirements are gathered and submitted to the issue tracker system. Features are generated top-down, derived from business case initiatives and bottom-up by grouping low level delta requirements. Therefore, top-down features represent business strategies, whereas bottom-up features, addressing functionality enhancements, are bundled with respect to existing solutions.

Feature Selection. As mentioned above features are strongly connected to business case initiatives, where the initiative priority is based on different criteria, such as reducing costs, efficiency gains or customer impact. Any changes on these priorities directly affect existing release plans. The *pre-selection criteria* for scoping the release backlog are (a) *must* (b) *nice to have* and (c) *must not* considering business case initiative

Table 2. Source and classification of requirement relevant information

Source	**Information Type**	**Examples**	**RAM** [8]
CAB	Business Cases, IT Roadmaps (Business Case Initiatives)	Global Application Standardization	Organizational Strategies
		Provide multi-country system versions	Product Strategies
CMT, IRG	Release Roadmaps (main topics)	Handle several countries in one instance	Product Level (goal)
Roll-out Projects, Support, Test Phases	Business Requirements (Bug/Change/ Functionality Requests)	-	Feature Level (features)
		GDC shall enable multi-currency	Function Level (functions/actions)
		-	Component Level (details-consists of)

rankings and technical feasibility. Based on qualitative and quantitative data the following explicit and implicit SRP selection criteria were identified. Feature *priority* and *implementation risk* were stated by the practitioners as the determining feature selection criteria. The analysis of meeting notes, release notes and requirement documents showed that feature priority comprises additionally the following implicit feature selection criteria.

Requirement Issuer are those countries that rise a requirement, where countries have different priority primarily based on the revenue. In many cases requirements are suitable for all or most of the other countries and are classified as global features/requirements. It is a challenging task to decide which requirements are globally suitable and which of them should be only provided in local implementations.

Release Consumer Order For any new system release a pilot country is chosen to roll-out the new release as a first release consumer. Therefore, raised requirements of the pilot countries are preferred, in particular requirements that aim at assuring the roll-out of the system (e.g. interface or data migration requirements). Additionally, requirements of consumer countries of the next release are also preferred compared to requirements raised by countries that would not consume the current release. The results of the data analysis and observations surfaced the following implicit feature selection criteria:

Effort Estimation There are several stages for effort estimation during release planning. For SRP purpose in some cases it is required to estimate feature effort before solution concepts are developed. This is especially the case for top-down generated features. Assessing the number of touched software areas provides evidence on the expected implementation effort. If solution concepts are clear, the effort estimation is based on comparing the effort of one solution relative to that of another. Therefore, the feature effort includes the sum of all related solution efforts. Additionally, features with high effort estimations are implemented first, except technical constrains require another feature implementation order.

Requirement Volatility This selection criteria was primarily identified through the observation of planning meetings and the analysis of meetings notes. There are several risk factors for requirement volatility such as the involvement of new technology or unclear underlying business processes. Features with high volatility and middle business priority will be postponed, where features with high volatility and high stakeholder priority are assigned to the subsequent releases or release iterations to be able to accommodate intermediate requirement changes.

Table 3. Overview GDC Release Planning Data

	GDC 3.6	GDC 3.7	GDC 3.8
# planning items	89	51	14
# high level req.	0	3	14
# low level req.	89	48	0
# communicated changes	32	8	6

So far, the ad hoc strategic and operational release planning has worked well. However, the growing number of involved countries in the future and the demand to react faster on business change increases the complexity of strategic and operational release planning, which could not adequately be handled ad hoc any more. There are difficulties of utilizing the feature concept for SRP purpose at the company. A feature should be suitable to represent both, high level business requirements and software functionality abstraction at the same time. For operational release planning purpose low level requirements are assigned to 38 different software areas. In many cases a specific requirement is related to several areas and therefore these software areas are not suitable for bottom-up feature generation. Altogether, the GDC development is characterized both through project initiated requirements engineering (GDC roll-out projects) as well as through requirements initiated projects (global GDC development). Moreover, it is noteworthy that all requirements represent delta requirements by specifying enhancement proposals that are only understandable in relation with the existing system, which causes difficulties in relating them to business strategies.

4.2 Strategic Release Planning To-Be Process Proposal and Tool Evaluation

Along with the investigations of the as-is SRP process several issues and requirements for a to-be SRP process are gathered. The major improvement possibilities are seen by the practitioners in a systematic SRP process that integrates decision support for feature generation and feature selection. For that, the ad hoc as-is SRP process (see Figure 1) of the company was aligned to the 13 steps of the EVOLVE II method, as shown in Figure 2, to provide a SRP to-be process proposal (RQ2). To evaluate the suitability of the SRP process solution proposal, the ReleasePlanner was introduced at the company by retrospectively simulating the planning for GDC 3.6 and 3.7. In the following only SRP to-be process proposal gaps (additional requirements RQ3) are considered, which were identified and discussed together with the practitioners. These requirements primarily address decision support needs that are not or not sufficiently supported by the solution proposal.

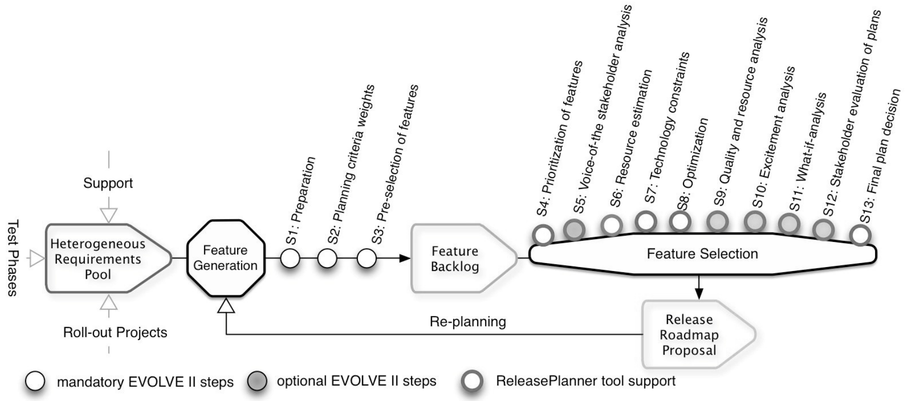
Support
Test Phases
Heterogeneous Requirements Pool
Roll-out Projects
Feature Generation
S1: Preparation
S2: Planning criteria weights
S3: Pre-selection of features
Feature Backlog
S4: Prioritization of features
S5: Voice-of-the stakeholder analysis
S6: Resource estimation
S7: Technology constraints
S8: Optimization
S9: Quality and resource analysis
S10: Excitement analysis
S11: What-if-analysis
S12: Stakeholder evaluation of plans
S13: Final plan decision
Feature Selection
Re-planning
Release Roadmap Proposal
mandatory EVOLVE II steps
optional EVOLVE II steps
ReleasePlanner tool support

Fig. 2. Proposed to-be strategic release planning process

ReleasePlanner Simulation. Applying the tool at the company for release planning simulation creates some integration effort, where the effort depends strongly on the quality and availability of required planning data, the current SRP process and the utilized development tools. The most difficult task was to choose the selection criteria and selection criteria weights. Any adjustments on selection criteria or criteria weights caused significantly plan changes.

Simulation Setup The setup for the retrospective simulation of the release planning process for release 3.6 and 3.7 comprised *135 planning items*. These items probably do not represent the actual requirement backlog, because we cannot ensure that we have replicated the backlog completely. Requirement issuer and requirement volatility are used as selection criteria to represent stakeholder priority and implementation risk. To quantify requirements volatility the discrepancy rate of *best-case* and *worst-case* effort estimations are used. The higher the discrepancy the higher the requirement volatility risk. The selection criteria release consumer was considered by pre-assigning requirements of the pilot countries to the according release. The resource capacities are approximated through story points, where 160 story points for release 3.6 and 80 story points for release 3.7 were assumed.

Simulation Results The tool provides two measures, degree of optimality and stakeholder feature points, to evaluate the quality of the alternative plans. In Table 4 five optimized planning alternatives are compared with a manual baseline plan. Alternative 1 is the best possibility when relying on stakeholder features points, which measures the stakeholder satisfaction related to a specific plan. Compared to the manual plan it can be seen, that the tool computes a plan with which the stakeholder would be more satisfied than with the manual plan. Additionally, alternative 1 provides a plan with a better degree of optimality with respect to available resources, which could be also seen in the different number of assigned features. The discussion of the results with the practitioners yields the following conclusions: (a) the proper requirement selection criteria were identified for release planning, because the calculated plans are very similar to the manual plan. (b) it is difficult to assess, whether the quality of calculated plans is better than the manual plan, it depends strongly on the suitability of utilized planning data such as effort estimations and resource capacities.

Table 4. Comparison of ReleasePlanner computed plans against manual plan

	A 1	A 2	A 3	A 4	A 5	M. Plan
degree of optimality	99.7%	98.2%	97.6%	96.7%	95.7%	99.1%
stakeholder feature points	(566196)	(557325)	(553950)	(549044)	(543505)	(562520)
# features assigned 3.6	62	62	62	62	61	62
# features assigned 3.7	23	23	23	22	24	20

4.3 Additional Identified Strategic Release Planning Process Requirements

Feature Generation. The feature generation task is not addressed by the EVOLVE II method, because available feature sets are taken for granted. As a result the identified requirements (FGx) and corresponding rationals related to the feature generation task are gathered from and discussed with the practitioners.

(FG1) Support top-down and bottom-up feature generation Rationale: Features are used for strategic planning purpose. They are derived top-down from high level business strategies (business case initiatives) or they comprise a bundling of low level requirements that have arrived via different input channels.

(FG2) Support aggregation of relevant changes into existing release plans Rationale: Requirement relevant changes continuously arrive on different abstraction levels and have to be considered during re-planning. These changes for instance comprise priority change, intermediate identified requirements or changing effort estimations. Especially, the adaption of resource capacities or effort estimations are stated by the practitioners as a challenging task.

(FG3) Support delta requirements handling Rationale: Since GDC is developed incrementally over several years, the requirements, that are source for SRP, represent delta requirements. This causes major problems if these delta requirements cannot be linked to planned (to-be) and existing (as-is) system specifications. Due to the strategic (high level) planning purpose it is not clear which abstraction level is necessary to represent the system and how to link it with delta requirements.

FG4) Support feature classification and variability Rationale: Primarily, GDC is developed globally, developing functionality that is used by all countries. However, in some case specific functionality is not necessary (optional) since local business process are different and functionality (features) should be switched on/off for local GDC instances to reduce testing and maintaining costs. Additionally, the local instances of the GDC system require different configuration settings. Therefore, requirements should be classifiable into functional or configuration requirements.

Feature Selection. The feature selection task is well guided by the EVOLVE II method, and according tool, by supporting multiple selection criteria and comprehensive analysis capabilities. However, additional requirements (FSx) related to feature selection are identified and described in in the following.

(FS1) Support pre-selection (Scoping) Rationale: Because of continuously arriving changes overscoping arises. That requires iterative pre-selection and pre- selection support. This requirement is related to the FG2, because pre-selection is necessary after the aggregation of changes.

(FS2) Support multi-view selection criteria Rationale: The results on investigating the feature selection criteria of the as-is SRP have shown that implicitly business/technical and organizational views are reflected in the selection criteria. Providing decision-support for feature selection means to support the identification and solution of conflicts between the three different views, where selection criteria could be assigned to one of the three views.

(FS3) Support strategic analysis capabilities Rationale: A major challenge of developing a globally used software system is to balance multiple country specific business strategies to provide a system that satisfies all stakeholder adequately.

(FS4) Support the modeling of release dependencies Rationale: SRP focusses on long-term feature selection where features are assigned to subsequent releases. This rises the need of considering, besides feature dependencies also release dependencies. (e.g. dependencies to other projects or systems).

4.4 Requirement Abstraction and Solution Model

Based on the insights and requirements on feature generation (FGx) an extension of the RAM proposed by [8] was developed to support the mentioned problems and requirements. The requirement abstraction and solution model (RASM), illustrated in Figure 3, represents a preliminary solution proposal to address the elicited requirements. There are several reasons why the RAM [8] is not sufficient for the purpose of strategic release planning for the company. Most of the requirements are delta requirements, aiming at the change of existing software functionality. RAM provides the possibility of modeling low level requirements on component level, describing how something should be implemented instead of what. However, to handle delta requirements knowledge about the existing structure of software functionality is required. Therefore, the RAM is used to abstract business requirements and is extended by linking product strategies explicitly with software features. The linking on this specific level is necessary, because during SRP only high level requirements are available and the refinement of requirements on functional level happens later on. To overcome the dilemma that a feature should represent business requirements abstraction and software functionality abstraction at the same time, the model distinguishes explicitly between these two types of features to make SRP involved stakeholder also aware of it. We have learned from practice that features could be generated top-down, derived from business strategies, and bottom-up by bundling low level requirements. In some cases low level requirements do not address a specific business strategy, but have a high innovation character. A strictly business strategy oriented selection of requirements, as provided by Khurum et al. [10], could hamper innovation by neglecting requirements that provide innovative solution suggestions.

In the following the RASM elements, illustrated in Figure 3, are explained. A *system release* fulfills business features by implementing new features or by changing one or many existing software features. A *business feature* represents the refinement of business or organizational strategies or the bundling of low level business requirements. A constraint that was mentioned by the practitioners was that a business feature shall be implementable within a specific release and should be explicitly linked to software features. Business features represent the highest available level of business

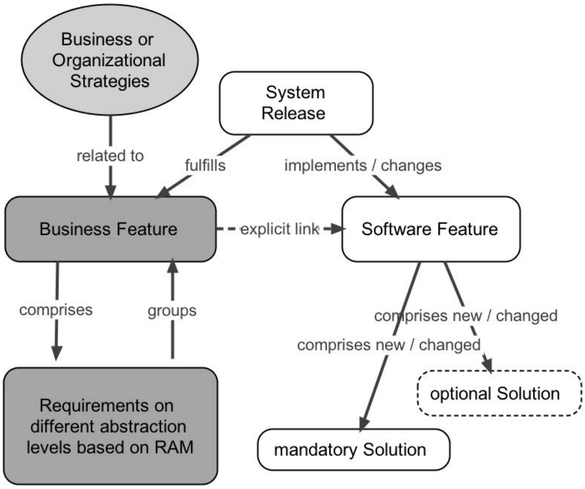

Fig. 3. Requirement and Solution Abstraction Model (RASM)

requirements according to RAM [8]. Referred to Table 2 business features could be mapped onto product strategy or product level and could be refined into one or many requirements on lower level. *Software features* represent the abstraction of planned and existing solutions, where solutions can be classified into optional/mandatory and functional/configurational solutions.

Utilizing RASM for feature generation the above mentioned requirements (FGx) are addressed as follows:

(FG1-Solution) The top-down feature generation decision is supported by the constrain of generating business features top-down only in relation to business strategies. Where as the bottom-up feature generation allows to use business and/or software features to bundle low level requirements.

(FG2-Solution) RASM enables the identification of release relevant change by bottom-up and top-down comparison of requirements change at different abstraction level to business strategies. Therefore, the aggregation of changes to requirements at any requirement abstraction level to business features is also possible.

(FG3-Solution) The linkage between business requirements and software features or corresponding solutions increase the understanding of delta requirements.

(FG4-Solution) The RASM provides a classification of solutions into optional or mandatory solutions that enables the modeling of solution variability.

5 Discussion

This section provides interpretations of the results in relation to existing work and a discussion of limitations. We interpret the SRP process of the company as an important task and the process has to be understood as a continuing activity. The results indicate that the introduction of the feature concept for SRP can provide two benefits. Firstly, through grouping of low level requirements into business features it reduces the complexity caused by huge flattened requirements lists. Secondly, the usage of high-level business features to better link them with business strategies improves the communication between the system stakeholders. Lethola et al. [12] recognized the latter as well in the context of market-driven development. Further, the ReleasePlanner evaluation results indicate that the ad hoc SRP process of the company works reasonably well. However, the systematic process provided by the tool ensures a sufficient degree of optimality and stakeholder satisfaction of release plans also when the complexity of release planning increases due to more involved countries and continuously changing criteria. Benestad et al. [2] also stated that the concern of evolving feature descriptions and design specifications are not well accounted for by release planning models and identified also the lack of handling continuous change. Determining the capabilities of an organizations release planning process was also addressed by Lindgren et al. [13]. They provide a capability model to identify areas for improvement.

As for every study there are limitations that should be discussed. In Section 3.4 we describe how we reduced threats to validity of the case study design, while in this section we discuss other limitations. There are two specific characteristics of the company that may have an impact on the SRP process and also influence the external validity of the results. *Mixture of plan-driven and agile software development elements.* Despite the adoption of agile software development methodologies by the company, there are only a few releases per year that are organized by plan-driven roll-out projects with fixed deadlines. This is necessary as long several depended projects have to be coordinated. In this study we have not analyzed which elements of the SRP process are plan-driven and which one agile or whether these two principles impede each other. To scale agile practices such as Scrum to larger projects and to coordinate several depended agile project releases without fixed deadlines is an open issue. *Mixture of customer- and market-driven development.* The information system is developed bespoke and used globally, where the end user and customers are known and provided functionality is strongly aligned with business processes to enable competitive advantage. However, there are some similarities to market-driven development, where the market is represented by the different globally distributed company sites. It is not always clear during planning and development time, which countries will finally consume which release version. This depends on whether the current system release provides features that are appropriate to satisfy country specific business strategies. The major intent of the company is to standardize the information system functionality to provide a customizable standard software *product* to all countries.

6 Conclusion and Future Work

In this paper, we reported on the results of an industrial case study that aims at the analysis of the SRP process of the company. We have analyzed qualitative and quantitative data to identify the as-is SRP process and developed based on the insights and results a to-be SRP process solution proposal. The major idea of the solution proposal was twofold. (1) The EVOLVE II method was adopted to the as-is process to provide a systematic method for release planning and (2) the RASM was developed to support requirements change on different abstraction level and to handle delta requirements by explicitly linking business strategies with solution planning and solution development. Finally, the SRP process solution proposal was evaluated by introducing the Release-Planner tool to the company. The evaluation results comprise additional requirements with respect to feature generation and feature selection decision-making support needs, that are not or not sufficiently supported by the tool.

Future work includes the implementation of the additional identified FGx and FSx requirements, listed in Section 4.3, where the RASM already addresses FGx requirements. In terms of requirement FG3, it has to be investigated whether a (software) feature-based representation of an existing system is sufficient to handle delta requirements, if solution specifications are rarely available. Moreover, the identified requirements FSx for the to-be SRP process can be used for tool functionality improvements. Finally, there is some effort required to integrate RASM and EVOLVE II to propose a SRP process that combines feature generation and feature selection.

Acknowledgements. The authors would like to thank Roche Diagnostics GmbH (Mannheim) for its financial support of this research. Many thanks are also due to the GDC project team for their participation and valuable discussions of the results.

References

1. Al-Emran, A., Pfahl, D.: Operational planning, re-planning and risk analysis for software releases. In: Münch, J., Abrahamsson, P. (eds.) PROFES 2007. LNCS, vol. 4589, pp. 315–329. Springer, Heidelberg (2007)
2. Benestad, H.C., Hannay, J.E.: A comparison of model-based and judgment-based release planning in incremental software projects. In: Proceeding of the 33rd International Conference on Software Engineering, ICSE 2011, vol. 1325, pp. 766–775. ACM Press, New York (2011)
3. Bhawnani, P., Ruhe, G., Kudorfer, F., Meyer, L.: Intelligent decision support for road mapping a technology transfer case study with seimens corporate technology. In: Proceedings of the 2006 International Workshop on Software Technology Transfer in Software Engineering, TT 2006, pp. 35–40. ACM Press, New York (2006)
4. Bjarnason, E., Wnuk, K., Regnell, B.: Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering. Information and Software Technology 54(10), 1107–1124 (2012)
5. Carlshamre, P.: Release Planning in Market-Driven Software Product Development: Provoking an Understanding. Requirements Engineering 7(3), 139–151 (2002)
6. Danesh, A.S., Ahmad, R.: Software release planning challenges in software development: An empirical study. African Journal of Business Management 6(3), 956–970 (2012)
7. Fricker, S., Schumacher, S.: Release planning with feature trees: Industrial case. In: Regnell, B., Damian, D. (eds.) REFSQ 2011. LNCS, vol. 7195, pp. 288–305. Springer, Heidelberg (2012)
8. Gorschek, T., Wohlin, C.: Requirements Abstraction Model. Requirements Engineering 11(1), 79–101 (2005)
9. Jadallah, A., Galster, M., Moussavi, M., Ruhe, G.: Balancing value and modifiability when planning for the next release. In: 2009 IEEE International Conference on Software Maintenance, pp. 495–498 (September 2009)

10. Khurum, M., Gorschek, T.: A method for alignment evaluation of product strategies among stakeholders (MASS) in software intensive product development. Journal of Software Maintenance and Evolution: Research and Practice 23(7), 494–516 (2011)
11. Komssi, M., Kauppinen, M., Tohonen, H., Lehtola, L., Davis, A.M.: Integrating analysis of customers' processes into roadmapping: The value-creation perspective. In: 2011 IEEE 19th International Requirements Engineering Conference, pp. 57–66. IEEE (August 2011)
12. Lehtola, L., Kauppinen, M., Kujala, S.: Linking the business view to requirements engineering: long-term product planning by roadmapping. In: 13th IEEE International Conference on Requirements Engineering (RE 2005), pp. 439–443. IEEE (2005)
13. Lindgren, M., Land, R., Norström, C., Wall, A.: Towards a capability model for the software release planning process — based on a multiple industrial case study. In: Jedlitschka, A., Salo, O. (eds.) PROFES 2008. LNCS, vol. 5089, pp. 117–132. Springer, Heidelberg (2008)
14. Ngo-The, A., Ruhe, G.: Decision support in requirements engineering. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 267–286. Springer (2005)
15. Ruhe, G.: Product Release Planning: Methods, Tools and Applications. CRC Press, Boca Raton (2010)
16. Amandeep, N.F.N.G., Ruhe, G., Stanford, M.: Intelligent support for software release planning. In: Bomarius, F., Iida, H. (eds.) PROFES 2004. LNCS, vol. 3009, pp. 248–262. Springer, Heidelberg (2004)
17. Runeson, P., Höst, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering. John Wiley & Sons, Inc., Hoboken (2012)
18. Bin Saleem, S., Yu, Y., Nuseibeh, B.: An Empirical Study of Security Requirements in Planning Bug Fixes for an Open Source Software Project. Tech. rep., Dep. of Computing, Faculty of Mathematics, Computing and Technology, The Open University (2012)
19. Suomalainen, T., Salo, O., Abrahamsson, P., Similä, J.: Software product roadmapping in a volatile business environment. Journal of Systems and Software 84(6), 958–975 (2011)
20. Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S.B., Shafique, M.U.: A systematic review on strategic release planning models. Information and Software Technology 52(3), 237–248 (2010)
21. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a Reference Framework for Software Product Management. In: 14th IEEE International Requirements Engineering Conference (RE 2006), pp. 319–322 (September 2006)
22. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell (2010)
23. Zorn-Pauli, G., Paech, B., Wittkopf, J.: Strategic Release Planning Challenges for Global Information Systems A Position Paper. In: International Workshop on Software Product Management (IWSPM), pp. 186–191. ICB-Research Report No. 52 (2012)