Release Planning in Industry: Interview Data

MRTC report ISSN 1404-3041
ISRN MDH-MRTC-219/2007-1-SE

Markus Lindgren

November 27, 2007

Contents

Chapter 1

Introduction

Release planning is a company-wide optimization problem involving many stakeholders where the goal is to maximize utilization of the often limited resources, such as budget and developers, of a company and turn them into business benefit [10]. The release planning results in a decision of what to include in future *release(s)* of a product. In making this decision one needs to consider how to make a product profitable both in the short- and long-term. As input to release planning are a set of *needs* that, when realized into a product, provides some business/customer value. Normally the cost of implementing all of the proposed needs is larger than the budget allocated to a release, therefore a decision needs to be made of what to include in a release and what to postpone. Also, the *set of needs* needs to be prioritized in order to maximize business value of the needs included in a release. In addition, there are constraints that need to be considered during release planning [10]. For example, time-to-market, dependencies between different needs, required competencies, competitors' product offerings, new technology, market demand, and quality aspects, as is illustrated in Figure 1.1.

A *need* is a proposal for a change that normally is negotiable to some extent; needs later become project requirements. However, there can also be parts of a need that are non-negotiable, i.e., constraints on the need. Examples of such constraints are legislation, time-to-market, and number of available resources. Usually there is a grey-zone in terms of what is negotiable and non-negotiable for a need.

The release planning problem is a hard problem for a number of reasons. For example, the stakeholders involved in making these decisions usually have diverse backgrounds, e.g., some with software engineering skills, some with hardware skills, some with marketing skills. Reaching consensus on what is important for a release in such a diverse group of people is part of the release planning problem. Another problem is that of rating requirements and/or features in respect to each other. For example, is the *customer value* of feature $f1$ higher than that of feature $f2$? Is the *cost* of implementing feature $f1$ higher than that of feature $f2$.

There are more criterias which are relevant than the ones mentioned above. However, most existing research on release planning make use of subjective judgments of only a few criteria of a feature, usually *value* (or *benefit*) and *cost*. This means that stakeholders must themselves aggregate different criterias into a single one, usually

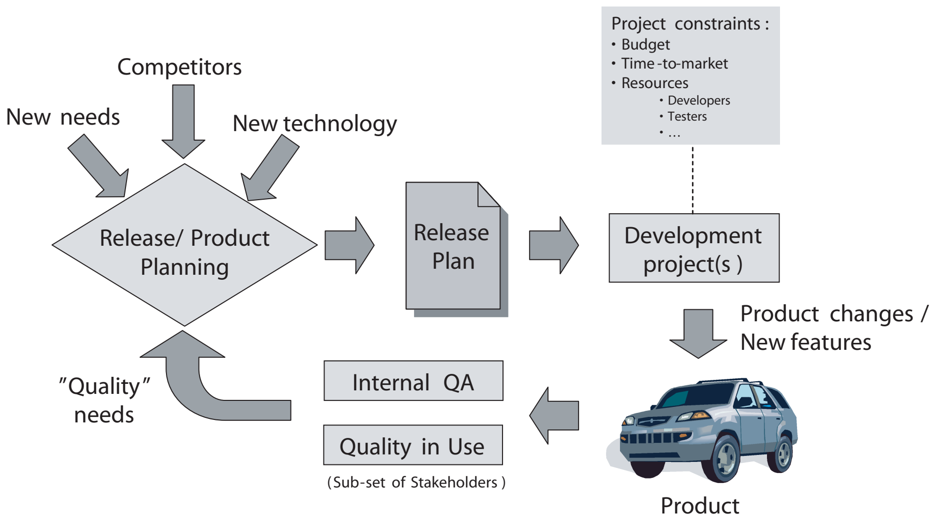

Figure 1.1: A selected set of issues relevant to consider during release planning.

aggregation is mainly relevant for the *value* statement of a feature, which normally is a non-trivial task. Cost estimation is also a non-trivial task which is influenced by many factors.

Today most product development is required to be based on existing systems for economical and time-to-market reasons [9]. This forces existing systems to be evolved by adding features or improving existing features. During such evolution the existing system often has impact on which extensions/improvements are easy/hard/possible to perform. The importance of considering evolution aspects is further motivated by "...the software development community is coming to grips with the fact that roughly 80 percent of a typical software system's cost occurs *after* initial deployment" [1].

For products that are evolved over a number of years a complicating issue can be that a product's existing software architecture can place restrictions on which features are economically feasible to develop (without costly architectural rework). So considerations of an existing product's quality can be an important part of release planning. Furthermore, as time passes customers' expectations can change, this again is an issue with impact on release planning. For example, when a new product is introduced customers can sometimes tolerate lower quality and a relatively high product price as long as they can still use the product sufficiently well, however, as time passes customers usually expect higher quality and a lower product price. This problem is also affected by competitor product offerings.

A common problem companies are faced with during release planning is prioritization between adding new customer features and improving quality of existing features. This problem is rarely addressed in existing research. If it is addressed, the *value* of a quality requirement is judged just as any other feature. Again, this is a problem relevent duing the evolutionary phase of a product.

This report presents the interview data from a case-study investigating some of the challenges of release planning. The case-study covers 7 different cases where most of the data has been collected through semi-structured interviewes. In total 16 people have been interviewed.

The purposes of this study are summarized below:

- Investigate how release planning is performed in industry today and compare it with current research literature. It is rarely the case that what is being used in industry matches the latest research results. Where are the differences?
- Identify areas that require further research, and thereby shape our efforts in the area of improving the release planning process. For example, are there aspects not being considered by current research?
- We believe that release planning in companies today are lacking in their consideration of software architecture and quality aspects, a belief we desire to either strengthen or falsify. Furthermore, we also think that there is need to improve life-cycle cost aspects in release planning, i.e., considerations of how the company considers how both their own and customer costs are impacted by product changes.

The report is organized as follows:

Chapter 1 This chapter, introduces the release planning problem and outlines the remainder of the report.

Chapter 2 Presents some of the related work within release planning. The body of related work is immense therefore only a selected set of relevant areas are presented in this chapter.

Chapter 3 Documents the material obtained through interviews at the companies, and in a few cases, the documents obtained from the companies. The chapter presents, in a sense, the raw material being used as basis for the analysis. Note that there is *one* description for each company, which consists of the *merged* notes taken during the interviews. (Interview questions etc. are documented in the case-study design.)

Chapter 4 Contains a short summary of the report.

Chapter 2

Related Work

The body of related work is immense, since release planning is related to many different areas, for example: *product planning*, *requirements prioritization*, *project execution*, *software architecture*, and *quality*. This chapter only covers a selected set of related work for release planning alone.

2.1 Release planning Key-aspects

Saliu and Ruhe discuss key-aspects of software release planning in [11], which need to be taken into account by a release planning method. These key-aspects are summarized below:

Scope Multiple releases need to be considered during planning, since some stakeholders are likely to become disappointed if their high prioritized requirements aren't included in the next release. For such changes in stakeholders needs.

Time Horizon There are two ways of planning release intervals, 1) release planning with fixed and predetermined release intervals, and 2) release planning with flexible release intervals. Today, it is most common with fixed release intervals.

Objectives There must be some notion of an objective for release planning. "Typically it is a mixture of value, urgency, and risk, satisfaction/dissatisfaction, return on investment, etc."[11].

Stakeholder Involvement A stakeholder is any person or organization has interest in a development project. Typical stakeholders are users, customers, developers, management, etc. To develop products that stakeholders are satisfied with it is required to involve the stakeholders during planning and development. The problem lies in identifying the important stakeholders.

Prioritization Mechanism In typical organizations there is rarely enough resources to, all at once, develop all desired requirements. There needs to be some mechanism for prioritizing requirements important to different stakeholders. Voting is one such method.

Technological Constraints It has been observed by Carlshamre et. al. [2] about 80% of the requirements where in one way or another dependent on each other. Therefore release planning needs to consider dependencies between requirements. An example of a dependency, requirement x cannot be implemented without requirement y.

Resource Constraints In development projects there are different resource constraints, e.g., release schedule, budget, and resources such as programmers, testers, and designers.

System Constraints Usually organizations have an existing product offering for which new releases are being planned, in an evolutionary manner. The existing software architecture, code base, defect history etc. have impact on the effort, risk, and complexity trade-offs when adding new features.

Charachter and Quality of Solutions Offered "This dimension addresses the question of what is considered and accepted to be a solution of the problem. The range is quite large: A single solution versus a set of alternatives? An ad hoc solution that is not necessarily feasible versus a solution that is actually satisfying all (or most) of the constraints? Or a solution with undefined quality versus a solution that achieves a predefined level of optimality?"[11]

Tool Support Release planning is a non-trivial problem involving lots of data, which can be simplified by the use of tools.

In [11] the authors compares different existing release planning methods to investigate how well these consider the key-aspects of release planning. The comparison results are presented in Table 2.1.

2.2 Knapsack problem

Jung [7] presents an approach to release planning, which formulates the problem as a *knapsack problem*. The goal is select a set of requirements that provides maximum value at minimum cost, which is similar to the goal of the knapsack problem. In the original knapsack problem the number of items that can fit into one bag on a trip, e.g., how many boxes with different weight that can fit into one bag. These are the assumptions of the method Jung's approach to release planning:

- Each requirement r_j provides, when implemented, some value v_j.
- Each requirement r_j has an associated development cost c_j.
- The budget for the release is b.

The problem is to assign requirements to a release such that Z_0 is maximized in Equation 2.1.

$$Max\ Z_0 = \sum_{j=0}^{n} v_j x_j \tag{2.1}$$

	PSERM	ENFEM	NRP	OVAC	COVAP	IFM	EVOLVE
Scope	1 release	1 release	1 release	1 release	1 release	Chunks of small releases	2 releases planned ahead
Time horizon	Fixed release	Fixed release	Fixed release	Fixed release	Fixed release	Fixed release	Fixed release
Objectives	Based on benefit of system changes	Based on benefit of features to customers	Based on weight of customers	Based on value of requirements	Based on customer satisfaction	Based on return of investment	Based on value, urgency, stakeholders weights and satisfaction
Stakeholder involvement	Project manager	Developers, project management	Project manager, customer	Involvement of project manager is implied	Project manager, customer, users	All major stakeholders	All major stakeholders
Technological constraints	Not available	Not available	precedence	Not available	Not available	Precedence (precursor)	Coupling and precedence
Resource constraints	Cost, risk	Effort	Cost	Cost	Cost	Cost, time -to-market	Effort, risk, schedule
System constraints	Operational risks	Not available	Not available	Not available	Not available	Not available	Not available
Character and quality of solutions	One solution plan	One solution plan	One solution plan by any chosen search algorithm	One solution Plan generated	One solution Plan provided	One plan spanning many release periods	Several alternative solution plans are provided. These plans are diversified and fulfill some target quality level.
Tool support	Not available	Time -tracking system	Not available	Not available	Not available	Partially available	ReleasePlanner ®

Table 2.1: Comparison of release planning methods from [11].

where x_j is a decision variable which is 1 if requirement j is included in the release, and 0 if requirement j is not included in the release. Equation 2.1 needs to be solved such that the constraint in Equation 2.2 is fulfilled, i.e., the selected requirements can be developed without exceeding the release budget b.

$$\sum_{j=0}^{n} c_j x_j \leq b \tag{2.2}$$

2.3 Art and Science

There are several different formalized approaches to release planning, Section 2.2 presented one such approach. This section presents one yet another formalized approach presented by Ruhe and Saliu in [10], which considers more factors. In their paper they discuss both one *art* approach and one *science* approach. This description only covers the science approach.

The *science* approach formalizes the release planning problem, and solves the problem of deciding the K next releases. The aim is to assign a collection of features $F = \{f(1), (f2), \ldots, f(n)\}$, which represent new functionalities, customer change requests, and defect corrections, to releases. The decision variables $\{x(1), x(2), \ldots, x(n)\}$ specify how features/requirements are mapped to releases, where $x(i) = k$ denotes feature $f(i)$ being assigned to release k.

A certain number of resources, and of different types, are required for realization of a feature $f(i)$; designers, testers, and programmers are examples of resource types. Ruhe and Saliu assume there are T resource types, for which there are $Cap(k, t)$ resources of type t available to release k. Furthermore, every feature $f(i)$ requires $r(i, t)$ resources of type t to be realized.

This results in the following constraint needs to be met for a release plan x, assigning feature $f(i)$ to release k, as expressed by $x(i) = k$.

$$\forall t \in T \ \ \forall k \in K : \sum_{\forall i : x(i) = k} r(i, t) \leq Cap(k, t) \tag{2.3}$$

That is, for all releases k and resource types t the feature set assigned to a release must meet the resource constraints.

Often there can be dependencies between features, for example, some features can be required to be released in a specific order and some features are required to be released at the same time. Coupling dependencies, which express features that must be release jointly are expressed by a set of pairs C, where each pair (a, b) express that feature a and b must be assigned to the same release. Precedence dependencies, which express ordering of features, are expressed by the set of pairs P, where each pair (y, z) express that feature y must be release before feature z.

Stakeholders are important during release planning, but every stakeholder doesn't have the same importance. Ruhe and Saliu assume a set of stakeholders $S = S(1), S(2), \ldots, S(q)$, which are assigned relative importance $\lambda(p) \in \{1, \ldots, 9\}$ where 1 represents lowest priority and 9 highest priority.

Features are prioritized according to *value* and *urgency*. Value is expressed using a nine-point scale, where 1 represents lowest value/urgency and 9 represents highest value/urgency. Each stakeholder p is asked to assign an ordinal value to feature i, which is expressed by $value(p, i)$.

Each stakeholder has nine votes per feature i for expressing its urgency, i.e., how important time-to-market is. These nine votes can be assigned to three possible choices: *assign to release 1*, *assign to release 2*, or *postpone*; these are the available options when planning 2 release, but this can be generalized for planning a higher number of releases. The expression $urgency(p, i) = (9, 0, 0)$ represents feature i being of highest importance for stakeholder $S(p)$ of being assigned to the first release, and for $urgency(p, i) = (0, 9, 0)$ it is of higest importance to assign the feature to the second release.

During release planning the objective is often to maximize some objective function, which usually is a mixture of value, urgency, risk, satisfaction, dissatisfaction, and return-of-investment. Ruhe and Saliue uses the objective function $F(i)$ expressed in Equation 2.4.

$$F(x) = \sum_{k=1}^{K} \sum_{\forall i : x(i) = k} WAS(i, k) \tag{2.4}$$

where

$$WAS(i,k) = \xi(k)\left(\sum_{p=1}^{q} \lambda(p) \times value(p,i) \times urgency(p,i,k)\right) \tag{2.5}$$

where $\xi(k)$ expresses the relative importance of release k.

The objective function expressed in Equation 2.4 must be maximized while fulfilling the dependency and resource constraints. Details of how to solve the equation can be found in [10].

In essence what Ruhe and Saliu does is to solve the same problem as presented in Section 2.2, but where the *value* and *constraint* expressions are more detailed.

2.4 Provoking an Understanding

As Carlshamre puts it in [2]:

> "In market-driven software development, release planning is one of the most critical tasks. 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. Despite this, the literature provides little information on how release planning is done in practice."

To discover more about the nature of the release planning problem Carlshamre has developed and evaluated a tool to aid in release planning. Here we discuss some of his findings, which were discovered during evaluation of the tool when in use by three people at three different companies.

Some of the features of the tool used during the evaluation are as follows:

- Graphical user interface providing overview of the requirements.
- The requirements selection algorithm uses a trade-off between requirement value and cost.
- Dependencies between requirements are considered, and broken/non-broken dependencies are visualized.
- It is possible to override the automatic selection of requirements and instead manually force the inclusion/exclusion of particular requirements.
- The tool presents several alternative release plans, such that the user can determine the best solution.

In the paper Calshamre argues that release planning is a *Wicked problem.* These are the characteristics of Wicked problems, which Carlshamre has referenced from Rittel and Webber:

- There is no stopping rule, solving stops when time, money, or patience runs out. "Good enough" becomes the stop criteria.

- There are better or worse solutions, but no optimal solution. Solutions must be discovered and compared.
- The problems are often unique even though there are many similarities with previous problems. There is always some additional property of strong importance that makes the problem unique.
- There is no clear objective of success.
- Iteration is required to find an acceptable solution.
- “There is no definitive formulation of a wicked problem. To define the problem is the same as defining the solution.”

Carlshamre motivates release planning to be a wicked problem by:

The Value of a Release is Hard to Define

Criteria Cannot be Defined in Advance

Judgements Are Always Relative

Planners Discover Properties as They Plan

Need Support for Alternative Models

One Release is not Enough

New Insights about Interdependencies

Credibility Issues

For which there are additional detail and motivation in [2].

Chapter 3

Cases

During interviews with people at the different companies we have tried to cover all issues in our case-study design. However, still it is not possible to cover all areas for a number of reasons:

- At each company it is not always easy to locate suitable persons to interview. A responsibility/role which at one company is handled by one person, can at another company be handled by either several people or no one at all (can be a non-issue for some products).
- We have limited the interview time to 60 minutes per person, since the people we interview usually have very tight schedules. (In a few cases the interview time has been up to 80 minutes.)
- To limit the time to perform the case-study we have limited the number of companies we investigate, but also the number of people we interview at each company.

Hence, if a certain area is not documented in the below company descriptions it doesn't necessarily mean that the company has no policies/guidelines or similar in the area, it merely means that the area has not been covered during the interviews.

3.1 Case 1

Company 1 develops and produces consumer products sold on a market with intense competition. The use of electronics and software within the company's has been rapidly growing over the last 20 years, and there has been considerable growth in number of employees within this area. At the same time there is quite extreme price pressure on the market due to the vast amount of competitors. A challenge within this line of business is the complexity in the products, which in turn contribute to the problem of verifying product properties, e.g., reliability. Currently the company has a product portfolio consisting of about 10 different products, where a single product can have up to 100 different options.

There are different kinds of projects, where larger projects usually have a duration of 3–5 years and smaller projects are usually run on a yearly basis, where the difference in budget for these projects is more than 10-fold. The development of new products has an even higher budget. Development projects follow a gated development process.

The company has development and production in a number of different countries in the world.

3^{rd} party components

The products produced by the company consists to 80–90% of sub-supplier components, both software, hardware, and mechanics. When a new sub-supplier is contracted there is considerable work to define business contracts, and evaluation of the maturity of the development organization at the sub-supplier. There is a common software package provided by the company to sub-suppliers in order to be able to integrate software developed by the sub-suppliers into the products.

Release/Product Planning

There are yearly updates of the products in the product portfolio, which can include one or more design, mechanical, software, and hardware changes. In addition, there is normally yearly requirements from upper management to reduce, e.g., production cost by a few percent. There are also projects focusing on development of new products.

There is a yearly process that decides what the company should focus on during the next year(s). Examples of input to this process are proposals, and sometimes requirements, on for example (illustrated in Figure 3.1):

- Cost reductions
- Quality goals
- New legislation requirements, which the products must fulfill
- Market needs

Pre-studies are performed before changing/developing any of the company's products. The purpose of a pre-study is to investigate the consequences of the proposed

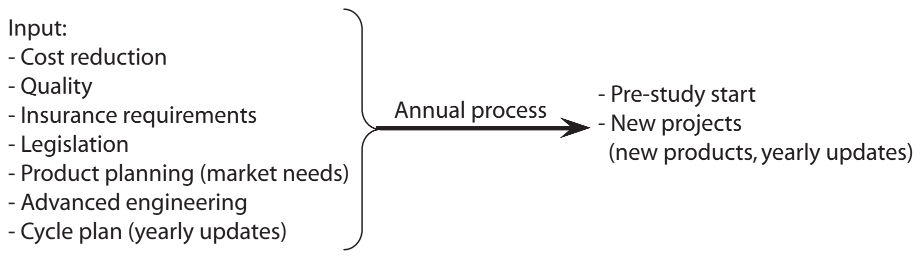

Figure 3.1: Input to the yearly process deciding which projects to start.

change(s). Decision material is prepared in pre-studies, which should contain cost estimations, both development cost and impact on production cost, property descriptions, return-on-investment calculus, budget, a set of change requests (concerning what needs to be changed in the products), and the consequences of making the change(s). This is later stored as a change request in a database. One interviewee stressed the importance of carefully investigating the consequences of proposed changes.

The result of a pre-study is delivered to product management which take decisions concerning future and existing products. Using the more detailed information from the pre-studies product management has possibility of re-prioritizing different alternatives.

Resource owners within the organization are responsible for performing pre-studies and produce the decision material requested by product management. Product management take decisions on which projects to perform, R&D performs the projects.

During early phases of a project, before the first gate, there is an iterative process which defines in detail what the project should deliver. At the first gate the deliverables from a project is "freezed". However, there have been occassions when there have been changes to the set of decided deliverables after the first gate.

The products developed by the company use different platforms/architectures for different areas of the products; about three such areas exist within the products. Changes to the platforms are handled using a prioritization process. Platform changes can, e.g., be the result of advanced engineering projects. Historically there have been periods where 80% of the development resources have been used for development of the base platform.

Within the company there are "three different pipes" responsible for dealing with different aspects of the products. These three areas ("pipes") are *product*, *quality*, and *cost reduction*, where each area is, in essence, operated independently.

It is rare for new projects to be initiated during a budget year, but there are cases when the scope is changed for different projects due to new needs. For these needs it does happen that new sub-projects are created within an existing project.

One interviewee considers the company to have a good process for software development, especially when compared to its competitors. Probably this is the result of intensive investments during 1990–2000.

One interviewee considers the organization to be good at "cross-functional"-aspects, e.g., issues crossing both mechanics and electronics. There is also a large openness within the organization which contributes to reaching good results.

Scope & Time Horizon

Normally the platforms rarely change, typically there are 8–10 years between larger modifications of the base platform, for the other platforms there is a somewhat shorter cycle and also easier to make changes. When a change is made to the base platform there is a long verification and validation phase. One reason for the long verification and validation phase is due to the large amount of components from sub-suppliers that must be integrated in a reliable way.

Objectives

The company uses an *attribute profile* which sets top-level requirements on products developed and produced by the company. The attribute profile describes within which areas the company should be leading and which areas are less important, when compared to the company's competitors. For each product, and within each market segment, an attribute profile is created that each product should aim for. Currently the attribute profile contains more than 20 different attributes. Examples of attributes are customer experienced quality, cost of ownership, and performance.

Requirements/Needs on each product are prioritized using its attribute profile as a basis. The attribute profile is in large associated with the company brand, hence, there are often small changes to the profile over time. Subjective judgements are used for determining how well particular needs meet the attribute profile. However, there are cases when deviations from the attribute profile occurs, e.g., when it is too expensive to realize a need using existing technologies, or when it is too expensive with respect to the current product offering. Cost aspects are also relevant during planning of the product offering.

Prioritization mechanism

The prioritization of what to develop is influenced by several different aspects:

- Yearly goals for customer satisfaction, production cost, guarantee cost.
- All products must meet a base level for quality, production cost etc.

Still, there is no defined method controlling how prioritization is done between different things. Instead this is often a discussion between the involved stakeholders.

As one interviewee puts it, "decisions concerning how different aspects should be prioritized is primarily handled by argumentation". To get your own will through it is important to "lobby for your own case". There is no formalized process for this, but it is rather a discussion between the different stakeholders and their proposed needs. In those cases where there exist objective metrics these can, of course, be used to strengthen your own argumentation, e.g., customer surveys indicating that a particular feature is important can have large impact on these discussions.

Stakeholder Involvement

System constraints

Resource constraints

A challenge for the company is the rapid growth in software and electronic content of the products. For example, today there are 20 times as many people working with this area when compared with 25 years ago. At the same time this line of business is under strong price pressure.

One challenge seen by one interviewee is all the communication required between different parts of the company that need to cooperate in order to produce good products. A complicating factor is simply the amount of people involved in taking decisions concerning the products.

Technological constraints

Tool Support

Information concerning the products are stored in several different in-house developed tools, e.g., quality aspects are stored in 2–3 different tools.

Product Strategy

There are different ways for new “innovative” features to be included in the products. In some cases there are proposals/studies or new technology for innovative features as a result from advanced engineering projects. Sometimes there are larger sub-suppliers which themselves propose new innovative features, but most often the company places orders to sub-suppliers to develop new features to be developed. Concept studies are usually peformed for those cases where sub-suppliers propose new innovative features.

At the same time, competitors product offerings are monitored and sometimes the company follows features introduced by competitors. However, it is hard to tell how large part is new innovation and following competitors.

Innovative features are normally first introduced in “top-of-the-line” products, and with time, when the features become less innovative, introduced in low-end products. The innovation value of a feature normally degrade over time as competitors “catch up”. In a sense the innovative features are moved lower down into the value chain, from top-of-the-line to low-end products. However, sometimes it happens that innovative features are first introduced some other product segment (not top-of-the-line), due to how development projects are paced. One interviewee lacked a clear strategy for how the company should act in relation to this.

The balance between innovation and commodity features is to a large extent controlled by the attribute profile. For example, the attribute profile describes in which area the products should be technology leaders and in which areas one can act as followers. This balance is decided by product management and marketing.

There are other parts of the company that deal with follow up of sales volumes and evaluates suitable pricing strategies. There is a yearly follow up of the pricing strategy.

Decision material, Decision meeting, and Consensus

Documenting release plans

Market needs on new products are documented quite sparsely. There are examples of descriptions of one line of text, but there also exists more novel-like descriptions.

Quality

Estimation on the impact of quality improvements is sometimes performed, and when performed it is often based on, e.g., cost and repurchasing frequency. Follow up is performed of customer satisfaction, guarantee errands, and repairs. For example, the company looks at trends for replacement of components during repairs, how often different parts of the software is updated at customers. When discussing quality with the interviewees they refer to customer experienced quality.

Severe problems in the products detected by customers are corrected as fast as possible, i.e., on need by need basis; these corrections go directly to production. In critical cases it can become necessary to recall earlier sold products to correct problems. Less severe problems are taken care of the next time it becomes necessary to make changes in that part of the product(s). In many of these cases there are sub-suppliers that are involved in making the changes, and there are not only technical aspects related to this. For example, there is need to take care of contracts with sub-suppliers in respect to resolving problem.

Follow up of a number of quality aspects takes place with regular intervals. Some of the investigated quality areas are:

- Each quarter customer satisfaction is investigated, both by own customer surveys but also by independent institutions/companies. Customer satisfaction has high importance during prioritization of different proposed requirements/needs.
- Continious follow up of how the different products meet their target attribute profiles. There is a separate department working with this kind of follow up. How well a product meets its target attribute profile is determined using subjective judgement.
- Each quarter quality and cost is monitored, e.g., production cost.

Approximately 70% of the development cost for software and electronics is spent on verification and validation.

Life-cycle costs

Project execution

3.2 Case 2

The products developed and produced by the company has high configurability, and within the company they usually say that there basically is no product that is identical to another. This doesn't necessarily mean that there is a difference in software and electronics, since there also be choices of color etc.

The area of software and electronics has rapidly been growing for a number of years. Basically the "growth of personell within this area follows Moore's law" (a law predicting a doubling in processor capacity every 18 months). The growth in this area is due to the increasing importance of software and electronics for the functionality in the product offering.

3^{rd} party components

Approximately 50% of the product consists of 3^{rd} party components, however, most "critical" functionality is developed in-house. Sometimes there are sub-suppliers supplying multiple companies with a particular sub-system, which in addition delivers high quality. However, the use of 3^{rd} party components is not entirely simple, since there is need to verify that sub-suppliers delivers what is expected from them. For example, there is thorough testing of components delivered to the company, to make sure that these meet expectations.

Normally there are requirements on 3^{rd} party components being "freezed" earlier than in-house developed components, in terms of deliveries to integration testing, since that test process usually takes longer time. In addition to development/production at 3^{rd} parties there is need for an acquisition process, test, verification, and it is hard for the company to control reprioritization occuring at 3^{rd} party companies.

Earlier they had high expectations on sub-suppliers delivering exactly what was being ordered, however, it has turned out that sub-suppliers aren't always able to deliver what is being requested. This has resulted in the need to also verify what is being delivered by sub-suppliers. For example, today testing of these components starts in earlier phases to avoid project delays.

There is also a group of quality assurance people that visit and inspect the sub-suppliers development processes. It has even occurred that a "special task force" has been sent to a sub-supplier to teach the "thinking" used within the company.

Release/Product Planning

Planning of a release starts approximately 18 months before its release date, and before reaching the release date there are 16 milestones. None of the interviewed people are part of product management therefore we have little information concerning details of product/release planning at the company, but generally decisions concerning which projects to carry out is decided during product management meeting. Their decision material includes: estimation of resource needs, investment needs, market analyzes and return-of-investment estimations. These meetings are held once per month.Part of the decision material is an assessment of how a project has impact on the company's 8 core values defined for the products. There are also 3 prestige values that are prioritized higher than the others. For each project goals are quantified within these areas.

Every 3^{rd} month there are meetings between management and resource owners to discuss releases, e.g., integration aspects. Primarly decisions are taken via consensus in the group, in case of conflicts upper management decide what to do. During these meetings it is primarly three things on the agenda:

1. Follow up of the release decided during the previous meeting.
2. Discussion concerning the features which need to be frozen to the next release.
3. Planning of future releases.

In essence there are 3 releases being discussed during these meetings.

In order for the products to remain competitive the production cost must be monitored closely. There are cost-cut projects with aim of reducing, e.g., production cost. The production cost is impacted by the construction design, choice of methods, and manufacturing methods.

Legislation with this line of business somestimes can make it necessary to develop completely new platforms, perhaps there are 8–10 years between new platforms; here the term platform also include the mechanical construction of the products.

Projects that deviate from set goals are reported and followed up. The most common deviation is for projects not being on time. It also happens that projects deviate more than allowed tolerances from set goals, e.g, for example if the goal is to improve the performance by 1% but during the project it is realized that only an improvement of 0.1% can be reached. In those case it can happen that projects are cancelled.

After it has been decided to run a project it immediately becomes “high priority”, i.e., all running projects are important. In case of being in doubt whether a project should be started or not, it is usually wise to wait until the next decision meeting such that additional facts, profitability assesments or market potential can be evaluated.

One challenge identified by one interviewee is how to handle the large amount of concurrently executing projects, with different target dates. At the same time the company has “continous introduction” of new features into production. Furthermore, today almost everything is run through projects, which results in a lot of adiministration of upstart and closing activities for project and many reports to document progress of projects. Currently there are more than 100 projects in progress. What earlier used to be part of larger projects is today handled as separate projects, which results in a high work load for project managers.

One challenge seen by one interviewee is how to gather all information concerning all changes being made to functionality in the products, information which needs to be included with future releases. Today this information is received via e-mail, telephone calls, or even when bumping into people in the corridor. There is a need to gather this information in a more systematic way.

Scope & Time Horizon

There are 4 releases per year, where 2 of the releases introduce new features while the other 2 releases are mainly for bug corrections. Normally there is no problem to delay the integration of new features since marketing doesn't start until 4 months before the release of a new feature, i.e., the feature is not available in the price list. The reason for having 4 releases per year is that it provides a suitable amount of time for test activities, e.g., integration and system testing. Some amount of time is required for testing a release before it reaches production, as a minimum it is desired to test a new feature both during summer and winter conditions. In addition, 4 releases per year is what they consider the organization be capable of handling.

Time-to-market is not what matters most for this kind of products, customer value and quality has higher importance. As one interviewee puts it, "we don't just release some fancy feature with no reason, it is really important to verify that sufficient and good quality has been reached before releasing a feature to the market".

The release of a new feature is almost always postponed in case development projects are unable to deliver functionality for integration testing on time. Basically there are no other options, since there must be high quality in delivered functionality and this must be verified through different test activities before being released to the market. Futhermore, it is often hard to speed up late projects by adding more resources (if detected in late project phases), since this often results in the project being even later due to overhead with training new resources. In case a delay is detected in early project phases it is possible to strengthen the resources, for example, by using consultants.

Objectives

One person usually becomes responsible for "researching" the customer value of a proposed feature, for example, by interviews with customers. However, normally there are earlier decisions by product management on what contains highest customer value and, hence, the priorities for what needs to be developed.

Prioritization mechanism

In those cases where prioritization needs to be done it is always customer value and quality which are primary concerns. There is a company profile which to a large degree controls how to prioritize.

Stakeholder Involvement

System constraints

Resource constraints

Technological constraints

As far as possible they try to keep projects apart such that there are no dependencies between projects, since there is a higher risk of failing to meet set target dates in case there are project dependencies. There are several different kinds of dependencies that can exist between projects. Also, it is easier to handle dependencies that are known, still it happens that project dependencies are detected later than desired.

Tool Support

They have been using Excel as an aid when planning integration testing etc., which is related to how releases are planned. Today, they use an in-house developed tool; only used by two people in the company. However, they feel that they have needs not being fulfilled by the tool. In the future they plan to look into a tool capable of handling a larger problem scope, e.g., requirements and deviations. Preferably they would like to use a commercial tool.

Product Strategy

The company has no goal of being a technology leader in this line of business, but rather they strive for being a company supplying high quality and reliable products.

Competitors

Future development within the company is to a large degree controlled by R&D. Usually there are many ideas proposed by the engineers.

The company actively monitors its competitors and usually each development department is responsible for monitoring the developments within their area. In case it is observed that competitors introduce some new feature in their product offering, it can happen that the company introduce some feature with similar purpose. However, primarily the development is controlled by internally generated ideas. In addition, a lot of development is controlled by changes in legislation.

Decision material, decision meeting, and consensus

There are templates for describing the features to be developed. Both at product management levels, and on lower levels. On lower levels in the organization the descriptions are focused on technical details, e.g., how a feature impacts other parts of the system. These descriptions are independent of the implementation.

Documenting release plans

Quality

Employees time is reported in four different main types of activities: development, pre-development, maintenance, and administration and training. Under each of these main types of activities there are a number of sub-categories. How time is reported into these main groups provides an indication of the quality of products.

There is a separate part within the organization that monitors how well the products perform at customers. There are also specific configurations of the products that some customers use on request by, and partly financed by, Company 1 from which feedback is collected.

Sometimes it is desired to get rid of dependencies to earlier releases, and in those cases they can remove the requirement of being backward compatible in order to be able to make new developments to the system architecture. Changes to the system architecture occurs about every 2–3 year, but the ambition is every 5 years. How often these changes are required is dependent on the growth rate of software content of the products.

Within the company there are procedures for documentation and follow up of errors/deviations in the products, where also the severity and priority of the errors are assessed. There are no dedicated bug releases/fixes, instead these are included in the normal flow of releases. In a way, bug fixes take a shorter path to integration testing. Upon serious errors at customer locations it can be required with product recalls to upgrade different parts. Such decisions are, however, made by upper management since these are often associated with large costs.

Quality is a primary concern to the company, and "quality is not an area where short-cuts are taken". "A developer doesn't pass things on without being proud of what he/she has achieved, they don't consider quality issues to be someone elses problem."

Normally each department has responsibility for a specific part of the product, and they have large own responsibility of how to solve different issues within their product area.

One challenge with the products produced by the company is the vast configuration possibilities, which practially results in it being impossible to fully test all configurations. Instead it is required to select a number of standard configurations that are tested, e.g., during integration testing. The high configurability makes it necessary for developers to deliver functionality with high quality. Hence, developers need to carefully conduct their component tests. One interviewee comments this as, "the high awareness of quality within the organization makes it possible to use less strict processes than otherwise would be required".

There are a number of tools within the company for follow up of quality etc. in the products. For example, one tool where customer reported problems are stored and classified. Furthermore, there is a separate part within the organization whose purpose is to follow up quality of the products when being in use by customers, and follow up quality in several different ways. For example, errors/problems reported by customers are taken care of by a group that daily go through reported problems. Either this group takes care of resolving the problem directly, or passes the error/problem to R&D, which can result resources from ongoing projects being used to resolve problems.

Life-cycle costs

Project execution

No details were discussed concerning project execution, but every second week there are meetings to "sync" the running projects. During these meetings deviations from set plans are visualized on a large board.

Once a release has been released for production it takes between 5 weeks to 6 months before it becomes available in a product. The first case is possible in case there are only software changes, while the longer time is required when adjustments to production machinery must be done, e.g., due to mechanical construction changes.

When a new product is ready for being launched to the market, managers go through the entire product and have proposals on different improvements. However, it is not always the case that these improvements are made, but rather becomes an input to the meetings aimed at deciding the design and functionality of products.

3.3 Case 3

Company 3 is a management consulting company, and consequently doesn't develop or produce any products of its own. Several of the employees at the company have long experience in leading positions at multi-national companies. This, combined with the nature of being consultants, makes them experienced in the product planning processes used at different companies.

We have interviewed one person from the company mainly for obtaining, in a sense, more general observations concerning the product- and release planning problem. During the interview the person have been requested to answer the questions based on how he/she desires the product planning process should be, i.e., not necessarily how it is at any company. During the interview the interviewee refers to a staged-gate development process [3], today many companies use some form of gated development process, to examplify when different steps should be completed. The description below refers to the responses given by the interviewee.

Requirements

First off, the interviewee differentiates between *constraints* and *requirements*, where constraints are non-negotiable requirements. A challenge seen by interviewee is requirements management up until Gate 2. The interviewee emphasizes the importance of not rushing through these decisions! The requirements need to be well-formulated and well-documented at this stage.

One motivation for carefully investigating requirements and system concepts etc. before Gate 2 is the high costs associated with performing changes in later development stages, which is illustrated in Figure 3.2 . The stair-case in the figure principially illustrates how the cost increases by roughly a factor 10 after each gate passed by a project. The two dotted lines in the figure illustrate the cost of making changes in different steps of a project; the total cost for these changes is obtained by integration below the lines and with consideration of the stair-case expressing the cost for making changes at a particular project stage. It is easily realized that it is much more cost-efficient to make many changes early in a project, e.g., before Gate 2, than in later stages of a project. In later stages of a project there is often many artifacts that are impacted by changes, e.g., manuals and test scripts.

The interviewee emphasizes the importance of not letting development projects pass Gate 2 too early, since it is extremely important that everything has been carefully thought through before commencing; the motivation for this is reflected in Figure 3.2.

Release/Product Planning

Requirements management is closely related to product- and release planning, since requirements are mapped to releases. Requirements management is a very important activity. Having documented, identifiable, verifiable and traceable requirements is a necessity for a release to have the possibility of being delivered in time, and also for development projects to reach their goals in time.

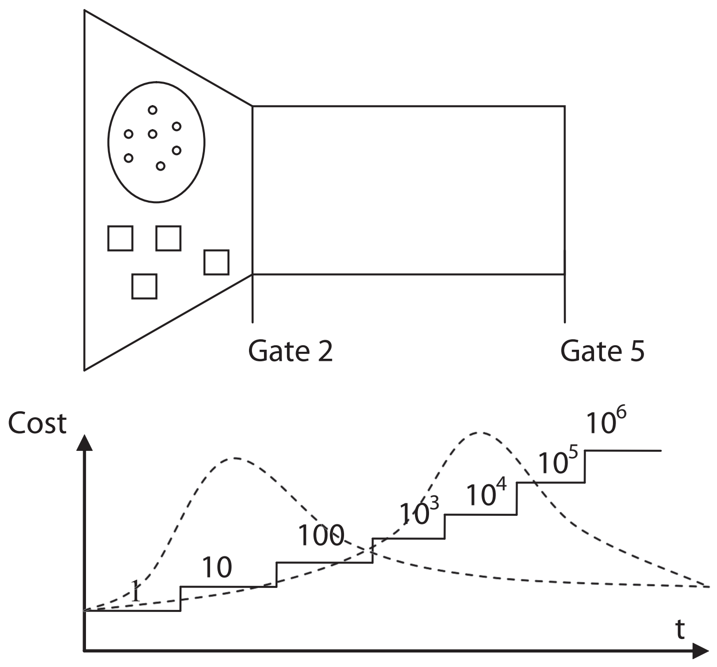

Figure 3.2: Making changes late in a project is more costly.

To start our discussions the interviewee drew the picture in Figure 3.3, which illustrates how different input, such as requirements from different stakeholders, existing technology platforms, new technology, and methods need to be considered in detail before passing Gate 2. The picture is supposed to look like a funnel where the large amount of input is narrowed down before Gate 2 is passed. The interviewee stresses the importance of having proper and strictly defined requirements. Furthermore, the interviewee considers that many companies today are lacking in their requirements management, thereby making this area especially important to consider. As a rule of thumb 25% of the work should be placed before Gate 2, and the remaining 75% after Gate 2.

The interviewee considers the work after Gate 2 to be more or less “straight ahead” if the preparations up until Gate 2 has been performed correctly, therefore one can see the activities after Gate 2 as more or less “factory work”; as long as no changes are made to the requirements or similar. To make this possible it is required that the “right” requirements have been identified, and that potential risks have been eliminated as far as possible before Gate 2. If the preparations are done correctly it can result in the time between Gate 2 and Gate 5 to be relatively short.

In addition, different system concepts should be evaluated before Gate 2 (refer to Figure 3.3). The architect/R&D should be responsible for the technical part of this work. To early look at system concepts is necessary if it is desired to go from Gate 2 to Gate 5 without any larger deviations. The architect/R&D looks at how different system

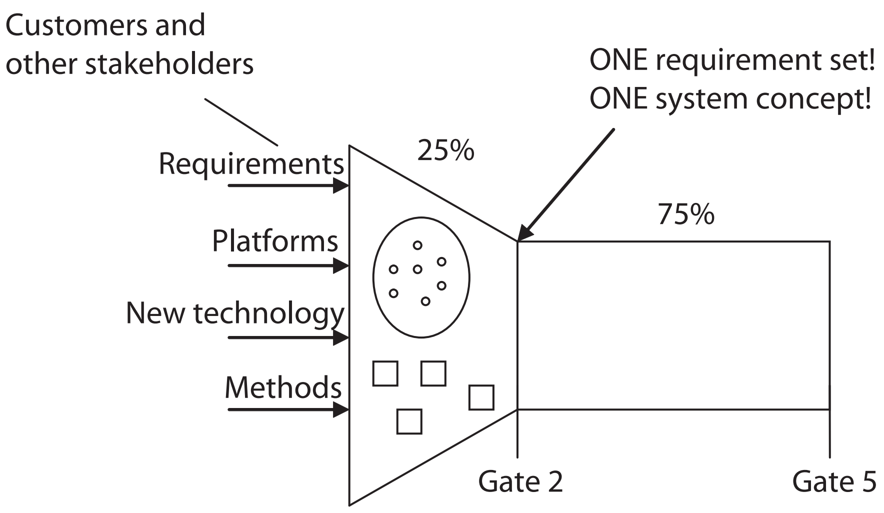

Figure 3.3: The “funnel”.

concepts impact the requirements.

Scope & Time Horizon

The interviewee doesn’t consider there to be any general rules for the time-horizon aspect of software releases. The release cycle is dependent on the product domain and the product life-cycle is also relevant. For example, it is useless to have a release cycle of 24 months for mobile phones, since the product life-cycle of mobile phones is relatively short.

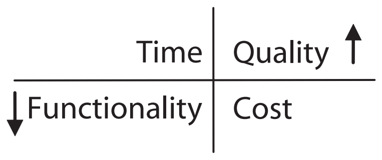

Figure 3.4: Prioritize quality before functionality in case of trouble meeting set plans.

In case a development project has trouble meeting set plans the company should prioritize (sufficient) quality before product functionality, as illustrated in Figure 3.4. Often the project cost is fixed, hence, the project scope needs to be modified to be able to meet any of the project goals.

In case new market demands arise during a development project these should not be considered in the current development project, instead, these new demands should be taken care of in a new project. The motivation for this is that in case new requirements are brought into a project during its execution it will drastically reduce the possibilities of meeting earlier set plans. Today, however, it is not rare for projects being forced to take care of new demands in this manner, perhaps without fully realizing the conseqeuences of such decisions.

In case the changes in market demands are large it can possibly lead to the currently executing projects being cancelled.

Objectives

The interviewee likes the development model/principles described in IEEE 1220 [6], where concepts such as *measure of effectiveness* (MoE) and *measure of performance* (MoP) are defined. MoE is a formula which expresses the customer value of a product. For example, for a mobile phone the battery time can be important for customers. This criteria, and others, are expressed in MoE. MoP expresses the internal efficiency of the organization. For example, an architectural change may improve the cost-efficiency of adding new customer functionality, which will become visible in MoP.

It is not always easy to derive MoE for a product, but it is necessary to express it somehow. This expression shouldn't be too detailed, but should be detailed enough to capture the main customer values.

Prioritization mechanism

The MoE expression forms the basis for how to prioritize different requirements. MoE is a mathematial expressions for what to prioritize. MoE should also be used as a basis for evaluation of different system concepts. However, there are factors for which it is hard to measure the customer value of, e.g., the user friendlIness of a product (including its GUI). The interviewee sees this as another kind of development, which perhaps best is performed in an iterative manner where, e.g., GUI prototypes are evaluated by different stakeholders. These iterations continue until a satisfactory result is achieved.

The evaluation of user friendlyness aspects, e.g., GUI, should be performed before Gate 2. That is, before Gate 2 it should be fully decided how the GUI should look. Without such decisions before Gate 2 it is hard to develop, and later meet, such a project plan.

Stakeholder Involvement

The interviewee considers it to be an absolute necessity that a software architect participates in the work before Gate 2, both to reach consensus within R&D and to properly be able to investigate system concepts before Gate 2. The architect needs to have authorization to take decisions, especially to decide interfaces between different parts of the system. In addition, the architect should be authorized to decide that, e.g., system concept C no longer needs to be investigated in further detail and be removed from the set of feasible alternatives. The architect should be involved in the decisions made concerning which system concept to chose.

The architect/R&D should be authorized to decide which system concept to chose. Project managers and bosses should not interfere with this choice, since this is not of benefit to anyone; it can become hard for everone to strive for the same direction otherwise.

System constraints

Resource constraints

Technological constraints

Tool Support

The interviewee likes Focal Point [5][8] for requirements management, especially the pair-wise comparisons for requirements prioritization. It feels as an easy to use tool and one quickly reaches consensus on what is important to do. The interviewee has briefly looked at DOORS [4] and Collaborative System Engineering from IBM.

Product Strategy & Competitors

Competitors and patent issues need to be considered early in the process (before Gate 2). This is an important aspect, since, for example, it may be necessary to consider patent issues early to avoid competitors locking out the company from a certain kind of solution.

Decision material, decision meeting, and consensus

Performing ROI analysis for requirements is an absolute necessity, but it can be hard to perform for individual requirements. Instead one can group some requirements and analyze ROI for groups of requirements.

Documenting release plans

Requirements need not be documented using long descriptions, but they need to be clear and have a context within which they are valid. Requirements need to be broken-down and documented, and it should also be documented how different software components are impacted by different requirements. This needs to be done before a decision is made in Gate 2. At Gate 2 there should be a decision material that documents the impact of the requirements.

Life-cycle costs

Quality

If faced with a decision to choose between adding new customer functionality to a product or to perform an architectural improvement, e.g., too reduce maintenance cost, one should develop a decision tree that illustrates the consequences of the different alternatives. For example, what will be the business impact of introducing the new customer functionality instead of performing the architectural improvement? The choice above must become a conscious choice, and the consequences of the choice must be understood.

Evaluating earlier product/release planning decisions is rarely made, since it is often extremely hard to see how a particular requirement has impacted a certain market segment. A factor making this even harder is that the traceability required for this rarely exists. The interviewee is of the opinion that today companies need to become better at requirements management, and perhaps later it can become interesting to look evaluation of earlier decisions.

Project execution

The interviewee is of the opinion that method improvements should be performed directly in real development projects, which may seem as a somewhat contriversial opinion. The motivation for testing method improvements in real projects, rather than in e.g. pre-studies, is to avoid the risk of these issues being left out (due to high pressure to perform other more “important" tasks). This approach is of course a risk-taking, where one for each occassion needs to estimate the consequences of the method improvement not turning out as imagined.

Mechanics, hardware, and software

There is great difference in flexibility between, e.g., mechanics and software. Mechanics often follows a (static) waterfall development process, and there may be steps that require relatively long calendar time. Software on the other hand can be developed, e.g, using iterative development methods. The interviewee believes that software development today is better than development of mechanics in terms of keeping track of requirements etc.

3.4 Case 4

The people interviewed at Company 4 develop a *platform* used by two other parts of the company, referred to as *A* and *B*, to develop products based on the platform. The platform itself is developed at 6 different development centers around the world.

The application organizations finance the development of the platform; A funds $x\%$ of the development cost and B $y\%$ funds of the development cost. Still, roughly 80-90% of what is being developed in the platform is used by both A and B.

Currently there is little room for customer specific customization of the company's products, earlier these possibilities were larger. More or less, the company develops standard products with a specific configurability, e.g., it is possible to activate some features by configuration. There is a license management system within the products to prevent customers from using features they haven't paid for.

Release/Product Planning

Company 4 has a large and globally distributed R&D organization, and hence, one important planning aspect for the company is to make it possible to maximize utilization of the internal R&D organization. Failure to plan correctly can result in a relatively large number of employees not being utilized to their full potential, which results in wasted resources, i.e., an undesired cost for the company. To be more concrete, system testing, design, pre-studies, development etc. must be run in parallel to efficiently use all available resources, resources which are used for creating new, or improving existing, business opportunities.

The primary "customers" of the platform are the two internal divisions A and B. Hence, product/release planning for the platform is aimed at serving needs of A and B. A and B, in turn, perform product/release planning based on their individual needs, for which part of the needs are relevant to consider during product/release planning for the platform.

The company strives for the product management for A to "speak with one voice" to the platform product management, hence, A must first internally form consensus on how their needs are to be prioritized. The same applies for B, i.e., "one voice" with prioritized needs from B. In addition to customer needs from A and B, consideration is also taken to needs from system responsibles at A and B, again, "one voice" from A and B respectively. Lastly, product management for the platform must consider needs from the platform system responsibles, e.g., needs concerning improvements of the existing system. The output from product management is "one voice" to R&D. The purpose of using "one voice" from different parts of the organization is that it should be clear what to prioritize. Still, there are plenty of discussions before reaching consensus on the "one voice". These different "voices" to the platform product management are illustrated in Figure 3.5.

Product management for the platform is located in a separate business unit, and works by issuing orders to the project office located in the R&D organization; partly illustrated in Figure 3.5.

One basic understanding concerning release planning at the company is that *there will be changes* during a release project, e.g., there will always be needs which aren't

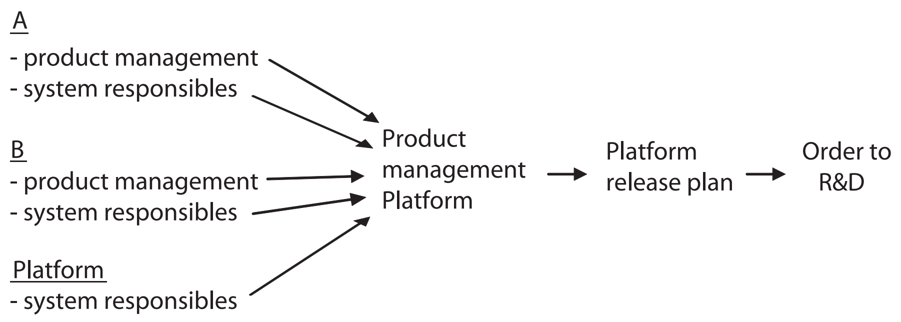

Figure 3.5: Flow of needs to platform product management.

thought of during the initial planning of a release and there will be changes to proposed needs. To cope with this the initial release plan must not assign more than 50% of the release budget, the remaining 50% is planned to be used for needs and changes during the release project. The initial content of the release plan, i.e., 50% of the assigned release budget, is called the *core specification*, which is part of the *main requirement specification* (MRS). During the initial phases of release planning there is usually 4 times as many needs as can fit within the budget of a release. This set of needs need to be reduced by prioritization. In their current release plan they have not yet reached a level where only 50% of the release budget is assigned.

Each candidate need must be documented in a "one-pager". A one-pager describes the need/requirement and shall contain:

- a slogan,
- business benefit,
- estimated cost, and
- impact on other parts of the system, i.e., dependencies.

In case there exists a one-pager for a need, and product management considers it to be relevant to proceed with the need, a pre-study is performed with purpose of refining the need. The result of a pre-study is an *implementation specification* (IS). In case product management still finds the need to be interesting it is possible to proceed with a *feasibility study*, which results in an *implementation proposal*, containing even more details, e.g., cost estimates. If product management still finds the need interesting after the feasibility study it is possible to proceed to *execution*, as illustrated in Figure 3.6. Note that after each step in the process the need is further refined and more details are investigated. Further note that it is possible to stop investigation of a need after each step; it is, of course, always possible to resume with a need at a later time.

One purpose of performing pre-studies and feasibility studies is to locate the needs providing greatest return-of-investment/customer benefit. These investigations explore more needs than can be developed within the release budget by the R&D organization.

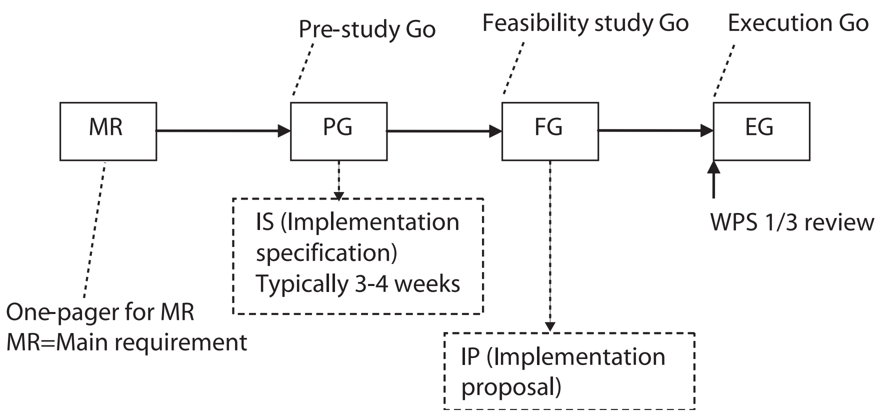

Figure 3.6: Overview of how needs are iteratively refined.

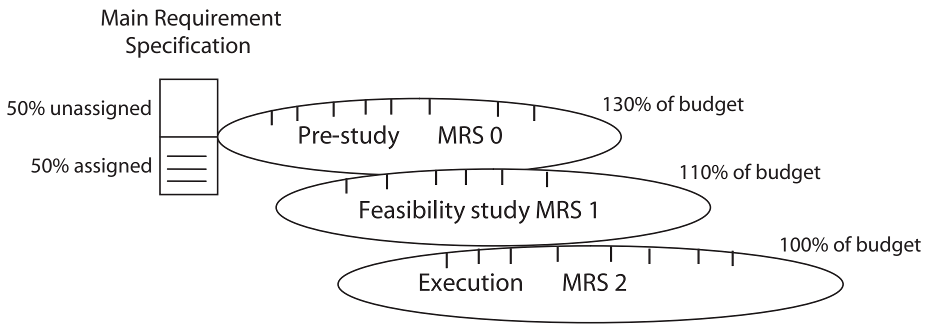

Figure 3.7: Overview of how needs are iteratively refined.

The needs for which pre-studies are performed requires, if developed, roughly 130% of the release budget. Needs are prioritized after the pre-studies such that feasibility studies are performed for needs requiring roughly 110% of the release budget. Prioritization is also performed after the feasibility studies, which in the end result in a release plan requiring 100% of the release budget, illustrated in Figure 3.7.

As mentioned in the beginning, one aim of the company is to maintain high utilization of the different parts of the company, which also applies to pre-studies, feasibility studies, and execution. Usually there are limited resources that are capable of performing pre-studies, since these normally require specific expertise. This resource constraint makes it necessary to start pre-studies at different points in time, which not necessarily is negative. Similar motivation exists for starting feasibility studies and execution of needs at different points in time. This iterative process is illustrated in Figure 3.7, where start of pre-studies, feasibility studies, and execution are illustrated by lines in the upper parts of the ellipses.

The purpose with this iterative planning process is to obtain:

- increased flexibility, e.g., by improved time-to-market due to the possibility of relatively late being able to investigate new needs in a controlled manner,
- high utilization of the R&D organization,
- risk reduction through incremental decisions, and
- improved planning, which is done in a step-wise manner through the process.

Good planning, and to early start planning, is required for success when using this process, and good planning is required to be able to deliver in time.

A challenge seen by one interviewee is how to map different needs to R&D, the "development machinery", how to prioritize what to do, and how to form a release plan which can be realized. The interviewee is satisfied with the amount of parallelism achieved in the organization and the flexibility offered by this process, especially the possibility of relatively late in a release project add new needs/requirements to a release in a controlled manner.

Lack of system architects/designer in the early phases, e.g., pre-studies and feasibility studies, can result in a "sloppy design", which leads to bugs which are expensive to correct. On the other hand a good design enables high utilization of the R&D organization.

A release plan at the company is desired to consider needs within the following areas:

- Business benefit
- Product maintenance
- Process improvements

Types of Releases

Roughly every 6^{th} week there is an internal release, a *shipment*, to A and B, which enable them to use the extended functionality offered by the platform in the development of their products. These internal shipments go through quality assurance before reaching A and B, both within the department delivering the functionality as well as within the system test department. One of these shipments will later become the main release being released to customers. The aim is to have 1 main release of the platform per year, but historically this has not always been the case.

The company also strives for minimizing the time between a main release (to customers) and the first internal version of the next main release being shipped to system testing, which is yet another way of maximizing usage of the R&D organization; this time particularly aimed at maximizing usage of system test resources.

Bug corrections are released to customers in the form of correction packages, but there is often a resistance from customers to use the latest release. Normally, customers only update to the latest release in case they see direct benefits. Typically only every 5^{th} correction package reaches customers and becomes used. Similar resistance exists within other lines of businesses as well.

Scope & Time horizon

The company strives to have one release per year of the platform, since often there is no need to release new releases more often than once per year, due to the “built-in” inertia within this line of business. Furthermore, to fully make use of the products developed by the company there is often need for other complementing products as well. This is one factor contributing to the “built-in” inertia. At the moment, one release per year is considered to result in a reasonable time-to-market.

In addition, there is often a resistance from customers to use the latest releases where “versions numbers to the left have increased”, e.g., major releases, since the customers see the possibility of the new release disturbing existing functionality. It is extremely important for the customers with high availability and high reliability, since failures can be associated with very high costs. As one interviewee noted, it has happened that new customer features have been introduced in a release for which only the version number for critical problems were incremented (due to the problem above). This in turn is a bit of a risk taking, since critical updates normally don’t go through full system tests. However, sometimes there is need to rapidly introduce new customer features in a release and therefore the planning process must be able to cope with these issues.

Within the company there has been high requirements on planning accuracy, plans shouldn’t deviate more than a few percent. There is a strong policy of projects delivering on time. In principle, it is not tolerated for projects, or parts of projects, to be late. In order to deliver on time and with high quality it is required to plan well. As one interviewee puts it, “it is a great challenge in succeeding with all the planning required for delivering on time”.

One interviewee pointed out an important aspect of being able to deliver on time, namely “doing what one should do” and not spending time on things that “shouldn’t be done, e.g., doing things which some think might come in handy later”. In the interviewee’s experience the company is good in doing what they should do, at least when compared with some earlier experiences of the interviewee.

In spite of this it does occur that projects are late, as happens in many companies these days. For late projects there are different possibilities:

- look for alternative solutions within the team,
- reduce scope,
- partial deliveries,
- move resources locally, and
- move resources from other departments, or even other development centers.

Exactly how to proceed with late projects, or parts of projects, is decided by a project’s steering committee.

When a new release of the platform is released it is desired for A and B to as soon as possible start using the new platform, since the sooner the organization is adopting the new release the earlier it is possible to reduce maintenance activities for old platform

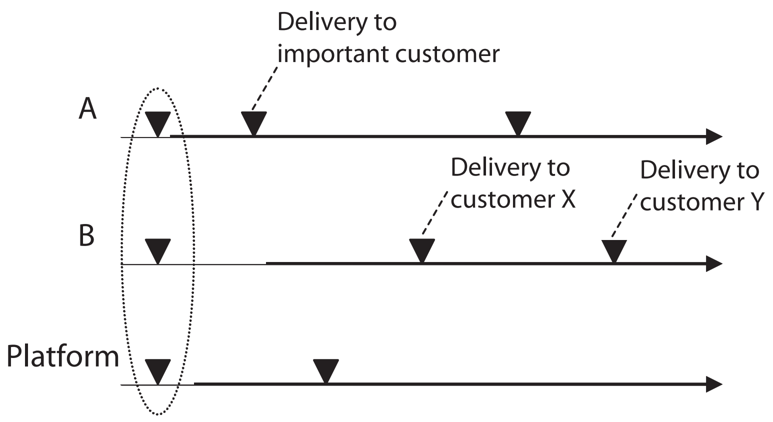

Figure 3.8: Synchronization of releases reduces the need for maintenance of old releases.

releases. This is also one motivation for having synchronized releases of the platform and the products developed by A and B, as is illustrated by the ellipsis in Figure 3.8. However, there are cases when synchronization cannot be maintained, for example, due to some important customer requesting some specific functionality within a product. There are also cases when a release is more or less targeted for a specific customer, which also is a reason for not always maintaining synchronization of releases (see Figure 3.8).

Objectives & Prioritization mechanism

Primarily it is a discussion between different involved parts which decide which needs to prioritize. Normally *business value* has highest importance during prioritization, and second comes the *cost*. If there is room in the budget, needs which provide some business value to a relatively low cost are often included in a release.

One interviewee notes that strong individuals, alternatively people in strong positions, can have easier of getting their proposals through, but this is not considered to be a problem within the company. Partly this problem is alleviated by these decisions being taken in groups.

Stakeholder Involvement

System constraints

Resource constraints

To be able to perform a good release plan it is required that the R&D organization keeps track of their available engineering hours, and within the different competence areas these hours exist, and the time required for competence build-up within other areas. However, this is not always the case; sometimes product management has a better total picture of this than R&D. In practice product/release planning must consider competencies available within different areas.

One fundamental aspect required for being able to deliver functionality in time are *committed resources*. Therefore it is not allowed to start working on a work package, i.e., a need or a set of needs, without committed resources. This is strictly enforced within the company, since this is a basis for delivering on time. The team working with a work package maintains a specification, which, for example, contains a detailed time plan and the committed resources assigned to perform the work.

As mentioned, good planning is one basis for delivering on time. As an aid for time estimations of work packages there is a time estimation guide within the company, with factors for different kinds of development, competences etc.

Technological constraints

The company is large and there are strong dependencies between different parts of the company. For example, the platform is required by A and B, and if a release is delivered late, or with poor quality, it can have significant impact on the success of A and B. As one interviewee stated it, “quality must be there and it must be delivered on time”. We didn’t discuss in detail how these dependencies are dealt with.

Tool Support

Focal Point is used by A and B, and for the early phases of planning platform releases. The tool MARS is used for requirements management.

Product Strategy & Competitors

Competitors were only briefly discussed during the interviews. However, sometimes politics and competitors in different countries have impact on the products developed by the company. For example, when a competitor from, e.g., Asia, adds a new optional function in their product offering it is often required for Company 4 to add that function to their offering to even be able to compete on that market; it can also be the case that such a function becomes mandatory within this market.

Decision material, decision meeting, and consensus

Decision material has already been discussed in the form of: one-pager, implementation specification, and implementation proposal, where an implementation proposal can contain several different alternatives of how to realize a need, and in that case, cost estimations for the different alternatives.

One interviewee pointed out that the complexity of the platform is partly a problem (referring to running development projects), since there is often many people with opinions on what needs to be done. “Reaching decisions within projects is a bit of a problem.” This is perhaps due to the “Swedish” way of working by consensus, which often takes longer time but often results in high quality in the end. To succeed with development projects it is required to have very good communication, and that all are in agreement concerning the current prioritizations and what needs to be done etc.

Documenting release plans

Quality

Approximately 20–25% of the release budget is assigned to product maintenance activities; correcting bugs is not included in the product maintenance activities. One interviewee noted that “there is a relatively large grey zone between what is considered quality improvements/product maintenance and development of customer functionality, since the activities performed within quality improvement/product maintenance usually becomes visible as business value later on”. For example, an architectural improvement may make it possible for new type of customer needs to be developed. There are also product maintenance activities aimed at following the development of new processors and perform upgrades. Currently there are activities aimed at performing architectural improvements to better exploit the possibilities of multi-core processors.

As one interviewee states, even though it is hard to value (in money) how much product maintenance is worth, in terms of the revenues these activities generate, these activites do result in return-of-investment/business value in a longer time frame. In many cases it is more or less obvious that these activities will result in business value, these cases are performed basically without discussion. In other cases it is not as obvious, and then discussions are needed, e.g., “why should this be done?”. One interviewee spoke quite a lot about the grey zone between product maintenance and development of new customer functionality.

There was not sufficient time during the interviews to discuss exactly how product maintenance activities are balanced with the development of new customer functionality, e.g., has the balance varied over time. However, as one interviewee commented, when a new product is released to a new market segment there can be a period of time with a higher number of reported errors/quality problems. Significant problems, e.g., to not detect dependencies between work packages in time, are investigated using *root cause analysis*. The result of the root cause analysis is used for process improvements within the company.

During the interviews quality were only discussed in general terms. Yet, quality is of high importance in the company. There are established processes for testing, e.g., integration tests, regression tests, and system tests. We didn’t go into detail on monitored quality metrics etc. However, before a major release is released there are a number of criteria that must be fulfilled. For example, there are a number of main requirements that must be fulfilled and there is a constraint on the number of allowed trouble reports and the severity of these. Failure to fulfill release criteria results in a delayed release.

Part of the product maintenance is performed at other development centers, which one interviewee has only positive experiences from. “There is some e-mailing back-and-forth to get things done, but these mails almost becomes a form of a contract concerning what needs to be done.”

Life-cycle costs

Project execution

There is continuous follow-up and adjustment of development projects’ scope which is performed by product management. Changes to project scope are handled using change requests, but by using the process with incremental decisions (pre-studies, feasibility studies, and execution) the need for change requests is reduced.

A *work package* consists of one or more needs from the main requirement specification and is assigned to a work package team, which is responsible for realizing the work package. A work package team can consist of as few as 1–2 people, but there are also cases with large work package teams. The work package specification is a refinement of the original need in the main requirement specification.

Before a work package is moved into execution/implementation a $1/3$ review is performed where the work package is reviewed to ascertain that all relevant aspects are taken into account, there is a good design etc. It is important that the *right* people participate at this meeting, which is an important factor in achieving high quality in the end result. Related product areas having dependencies to a work package participate during this review. Known dependencies between work packages can often be handled, however, there are cases when dependencies aren’t detected in time, which can be a cause for project delays.

Implemented work packages are delivered to integration/building of local versions. These in turn are later delivered for building into the latest version of the platform, which, after quality assurance, are delivered to A and B. Hence, the platform functionality is increased over time, and the functionality in the platform is incrementally introduced to A and B. Since these deliveries are used by A and B in developing their products it is important to deliver on time and with high quality, otherwise this can cause problems at other parts of the company.

Mechanics, hardware, and software

The hardware is often a governing factor during planning, since hardware often has longer lead-time to develop. However, sometimes parallel development and testing of software has been performed. For example, there are cases where emulators have been used to achieve parallel development of hardware and software. For those cases it is possible to perform some testing before the hardware is complete. There is minimal mechanic content in the products developed by the company.

3.5 Case 5

Primarily this part of the company delivers products to internal customers, approximately 70–80% are internal customers and the remaining part is external.

On large orders there can be a high degree of development, while smaller orders are mainly based on standard products. For larger orders the company is willing to do quite a lot, since an order is either taken or not, hence, it is vital to have those features which these customers value. In addition, there is strong competition within this line of business, which makes cost-efficiency an important factor in both development and order projects.

Application and development centers are located in different parts of the world, e.g., North America, Europe, and Asia. The products have high mechanical and electrical content.

Release/Product Planning

Product/Release planning is performed at different levels within the company. First there is planning on 5–10 years, which has more details in the short-term and less detail in the long-term. Second, there is planning done by R&D for the next release. This description mainly covers the 5–10 years part of the planning.

Product planning follows a yearly cycle, basically on calendar basis, which in February–March focuses on product strategies where, e.g., different markets are analyzed, trends within markets and market segments are looked into, as well as competitors and their offering. Input to product planning meetings should be available, roughly, in May.

There have been budget variations between development centers over the years, which most often is related to how development projects are assigned to different development centers. However, the yearly budget on division level is basically the same from year to year. The need for development projects is always due to new market needs, hence, in case there is less need for development the development budget will be lower. From a product planning perspective the budget is set from the beginning, hence, the problem turns into deciding how to best use it.

The product strategy work results in, say, 40 different needs. These needs are mapped to different development projects in different ways. For example, if there are many improvements that need to be performed it can result in a large number of small projects, and sometimes it results in a (lower) number of larger development projects.

Work is done during autumn to define the development projects that should be started, e.g., developing new customer features and/or performing cost-reductions. Regular meetings are held to discuss the different alternatives. Normally a rough prioritization of the different needs is performed which usually takes relatively little time; for example, needs that are very visionary are often removed, while needs that are closer in time, where the business impact is easier to estimate, remain. In the early phases of product planning it is not unusual to have 3 times as many needs compared to what can fit within the budget. The tough part is to remove the last 10% of the proposals that cannot fit within the budget, since by this time the proposals which remain are normally all relevant. Sometimes 10% is cut off the budget for all projects, and sometimes are individual projects/needs removed.

For each project being considered there shall be a business-case, an investment calculus (ROI), and a resource plan. It is hard to prioritize the different project proposals based on their business-cases, since these are written by different people at different places across the global organization. There can be several hundreds of business-cases that need to be prioritized, where each development center tries to market and give prominence to their own business-case(s).

One of the interviewees considers the company to have a good balance between market- and technology driven development, and there is good dialog between these areas. The R&D organization thinks about the market and mainly develops market driven needs. There is no exact figure on how the budget is divided between market- and technology driven development, a rough estimate is 50% per area.

Scope & Time horizon

Objectives

There is basically two different kinds of development:

- Development aimed at fulfilling market needs
- Technology driven needs, e.g., improvement proposals.

All these needs are documented using a common template, and are also prioritized according to similar criteria. The documentation on this level is easy to understand and the motivation to why a need exists is also easy to understand, even for people not being aware of technical details. As one interviewee stated it, “if you don’t understand the need (expressed on this level) it is probably not worth spending time on improving it”.

Prioritization mechanism

The method used for selecting which needs to prioritize is not strictly defined, but basically it starts from an economic viewpoint, e.g., how to turn a need into company profit. In doing this they look at the market share for which the proposed need have impact on, ROI, and product gap on different markets, e.g., needs that exist on the market but today aren’t fulfilled by the company’s product offering. Usually product gap is given high priority. In deciding what to prioritize it is mainly a discussion and argumentation that controls what should be developed, which is based on the possibility of turning the need into company profit.

Some other important product areas are customer price, reliability, and product weight. Improvements that don’t provide visible customer value are often disregarded. An improvement where production cost is reduced and there is a large volume has high probability of being included, since price is often the most important customer criteria.

Business needs that are not included for development one year have possibility of being subject to prioritization the next year, or sometimes even later. However, usually the decision material needs to be updated before it can be used again.

In addition, there is a political aspect concerning which needs to prioritize, which has its roots in development projects being assigned to different countries and application/order construction takes place in possibly other countries. As one interviewee stated, “this doesn’t make decision making easier, and one needs to be careful with sensitive subjects”. The political aspect was more sensitive a few years ago when the company went through a merger with another large company. However, this is not only a geographical/”political” problem, but it is also related to the different competencies which exist at the different development centers. Hence, it can partly be seen as a resource problem.

The decision making process is partly influenced by strong individuals, e.g., people which are good at presenting their case. One political aspect related to this is that it can have impact on to which country a development project is assigned. This is perhaps especially a concern when the remaining needs to remove are few, at which point normally all needs are good alternatives.

The final selection of which projects to approve is decided centrally within the company. An important part of getting your own project/business-case approved during these meetings is argumentation and marketing. It is mainly “gut-feeling” that controls what needs to be done. One interviewee estimates the political aspect of getting your own project/business case approved to 50%.

Stakeholder Involvement

Seminars are held for internal customers where plans are presented concerning the future plans. In addition, they also gather information concerning needs from the internal customers, which often receive high priority during product planning, since these issues are often directly related to a future business opportunity.

System constraints

Resource constraints

During the end of autumn, when a decision has been made concerning which development projects can fit within the budget, almost 100% of the development resources are assigned.

The total amount of available resources within the company, in form of development capacity, has impact on planning and marketing. For example, during periods with high order volume, fewer resources can be used for new development, which can possibly result in older components being used during order construction; often it requires fewer engineering hours to use existing components, rather than using new components which can really be a better choice. Hence, paradoxically when there is high order volume there is less resources available for new development.

Often the budget is fix, which means that it is not possible to recruit new people, or use consultants, in spite there being many orders. Another aspect of hiring new people is that in short-term it results in lower productivity. Still, the company tries to use as many resources as possible to handle the needs, but there are of course restrictions.

Technological constraints

The people interviewed at the company mainly work with product planning with a time-span of 5–10 years. There are other parts of the company that have similar planning, and these different plans need to be checked for project dependencies. During the interviews we didn't discuss any details of how these dependencies are handled.

Tool Support

The tool Roadmap is used for documenting the roadmaps within the company. It is considered to give a good overview of what is in progress at the company.

Product Strategy & Competitors

Competitors are continuously monitored and sometimes there is need to in detail study new features offered by a competitor. Studying how a competitor has solved a particular problem has possibility of generating own ideas on how to improve your own product offering. Hence, part of the product planning activities is in competitor monitoring.

There is no outspoken strategy concerning how the company's product offering should be in relation to its competitors' products, for example, there is no goal for the company to be a technology leader. Competitor monitoring is mainly for keeping track of what is happening at the different competitors. One contributing factor to this is the high cost awareness within the business, which has been intensified during later years as new competitors have been established in Asia.

Decision material, decision meeting, and consensus

Each proposed need is summarized on one PowerPoint page, which shall contain at least the following information: budget, purpose/motivation of development, product segment affected by the need, ROI, a time-plan. The time before ROI is reached varies; for some needs there can be several years before ROI is reached, but normally the ROI is less than a year.

Often ROI estimations for new market needs are more speculative than technology/improvement needs. For example, if an existing component is replaced with a cheaper component, i.e., a technology improvement, it is fairly easy to compute the savings.

An international team participates during the decision meeting of which development projects to run, and consequently the needs to prioritize. The team consists of representatives from marketing, engineering specialists, and the vice president for sales and engineering. There are often long discussions before reaching a result. The vice president makes the final call in case consensus is not reached. There is competition between the different development centers in terms of getting their own proposals through. Basically it is a competition of having the "best" business-case.

The new company organization has a clear division of functional responsibilities between different company locations. This separation reduces the number of political conflicts between locations. Still there is competition on business-case proposals from different company locations, which partly control the assigned budget to each location. One disadvantage with the new organization is increased reporting.

Documenting release plans

The decision concerning what should be developed is documented using an Excel sheet. This documentation is normally brief, and each development project is responsible for detailing the documentation.

Quality

To be able to remain competitive within this cost-aware line of business, the company puts high efforts into standardization/modularization of their of components to be able to reduce the number of engineering hours spent on order projects, but also to be able to keep low production costs. Another goal of the modularization is to be able to present a diverse offering to customers. Standardization/modularization is a relevant aspect to consider during prioritization of different needs; there is continuous work on improving the modularization.

For almost all projects the company keeps records of all problems occurring during customer operation, and these records do have impact on what becomes prioritized during product planning. For example, the company keeps track of failing components and trends. The quality records are stored in a database. Similar records are stored concerning production costs and problems that arise during production.

Sometimes customer satisfaction surveys are performed, but this is not any primary input to product planning. There is no follow-up concerning how well developed needs meet the projections made in the business-cases used during product planning.

There is no outspoken strategy for how to prioritize between quality improvements and new customer features, instead this is handled on case by case basis. As an example, if there is a component with too low reliability which causes large economic consequences for customers and some other component with low reliability with not as large economic consequences for customers. In such a case it is possible to reason about which component really needs to have high reliability, i.e., reason about the consequences of having low reliability. There is, of course, a relation to the cost of developing and producing a new high reliability component. In terms of quality the company has an outspoken goal of having at least as good quality as its competitors.

Life-cycle costs

A significant part of each development project is to reduce the company internal costs, e.g., costs for development, costs for application engineering, production costs, guarantee costs, and market goodwill. There are many activities whose purpose is to improve the companyŠs total profitability. The interviewees really emphasize the importance of cost reductions.

As an example of how important this matter is, one interviewee has developed an economic model expressing how different product properties impact customers' profitability. The model, for example, takes into account salary costs for customers to operate the product, operational costs, costs for service and repairs, i.e., the model captures the customers "total" economy in a sense. This model has been used by R&D for some months as an aid to estimate how different product changes impact a customer. (There are some adjustable parameters in the model, e.g., for expressing the importance of product cost for the customer and product volume.) This model cannot be used for evaluating new customer features, but rather improvements to the current product offering.

One challenge identified by one interviewee is how to balance the standardization/modularization of the product offering, while at the same time having a diverse product offering to customer, and while acting on a very price sensitive market. In essence, this is concerned with keeping both engineering and production cost low, while still allowing large variation in the product offering. Compared to, for example, car industries where the production volume can be in millions of cars per year, the production volume is limited, which makes it necessary to use other optimization criteria to remain profitable.

Project execution

Project execution was only briefly discussed during the interviews, since the interviewees are not part of the R&D organization. It can be noted, though, that ensuring cost-efficient production is an important part. The interviewed people have insight into development projects, for example, by participating in steering groups for some development projects.

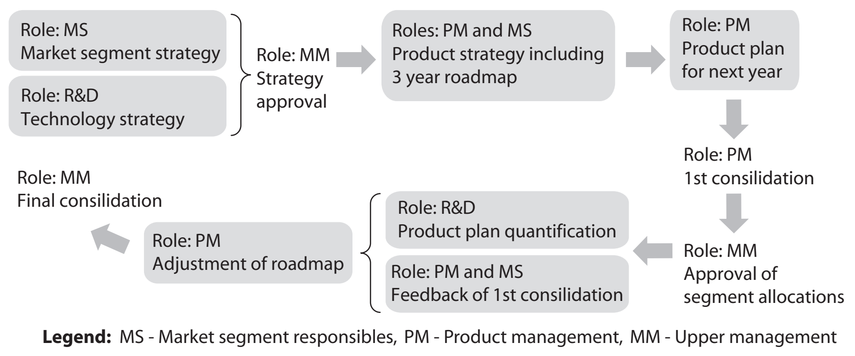

Figure 3.9: Annual process.

3.6 Case 6

The contents of a customer order is based on a list of options/features, which is filled in during sales activities.

At production stops at the company's workshop it is hard to stop the flow of incoming components/material, the same applies for the company's customers, which rapidly causes storage facilities to become full. This is one motivation for the need of high reliability in the products produced by the company.

$3\ rd$ party components

The company is to a high degree dependent on sub-suppliers, since almost all components/material are purchased. However, assembly of the products is performed by the company itself.

In terms of software in the product there are two main areas, where one uses very few $3\ rd$ party components and the other area uses more $3\ rd$ party components, e.g., GUI related components. Changing version of a $3\ rd$ party component can have big impact on the test activities of a release. These changes are, however, rare.

Release/Product Planning

There is a yearly budget process controlling the amount of available funds for each product manager. When the budget has been set, and the top-level goals for what to develop has been decided, the product managers have a somewhat larger freedom in deciding details of the product plan.

An overview of the annual planning process for Case 6 is illustrated in Figure 3.9. It starts by each market segment developing a market segment strategy and by R&D developing a technology strategy. These strategies are reviewed and approved by upper management, which is followed by product management and market segment responsibles developing a product strategy, including a 3 year roadmap. Based on this product management continues and builds a product plan for the next year, which is passed

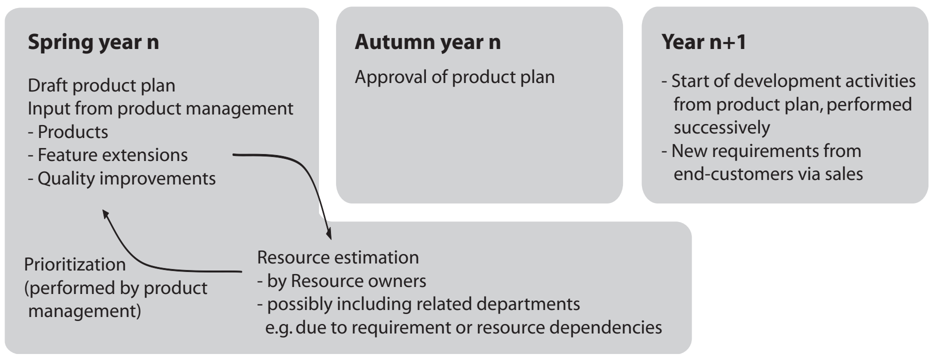
Spring year n
Draft product plan
Input from product management
- Products
- Feature extensions
- Quality improvements
Autumn year n
Approval of product plan
Year n+1
- Start of development activities from product plan, performed successively
- New requirements from end-customers via sales
Prioritization (performed by product management)
Resource estimation
- by Resource owners
- possibly including related departments e.g. due to requirement or resource dependencies

Figure 3.10: Iterative work is required between product management and resource owners when deciding the product plan.

to upper management for approval and for deciding market segment budget allocations. The product plan is passed to R&D for quantification, e.g., making cost and time estimations. At about the same time product management and market segment responsibles provide feedback on the product plan. Based on this feedback and the estimations from R&D, product management updates the product plan. Finally, upper management has the final word of what needs to be done.

Deciding the content of a release is performed by iterative work between product management and the resource owners, as illustrated in Figure 3.10, where the resource owners (from R&D) provide time estimations for realization of different requirements. R&D also provides information concerning the amount of available resources. Primarily the application needs are received from the sales department.

When the initial release plan for the coming year is completed it is sent for review to different parts of the organization (see Figure 3.9, and opinions and proposed changes are collected. These opinions are used for adjusting the release plan.

Generally speaking concerning release planning, and perhaps specifically for the budget process, there is a lot of "gut-feeling" that controls what becomes decided. Also, as one interviewee puts it "sometimes it is the *winds* on upper management levels that control what to prioritize, and then you have to adapt and control according to that". In addition, release planning can be disturbed by reorganizations etc., especially for development projects lasting for several years. Decisions, and reasoning concerning what to include in the budget, is performed in a group.

One interviewee points out that "reorganizations are always messy", since it breaks continuity when new roles and responsibilities need to be established, which in turn can have impact on how the budget is allocated. However, reorganizations is something which one hardly can prepare for, hence, this is handled as good as possible when it occurs.

Normally the requirements from product management are expressed fairly generally, which is not a problem in itself since in early phases one normally doesn't have a clear picture of how to realize different needs. Pre-studies are performed by R&D

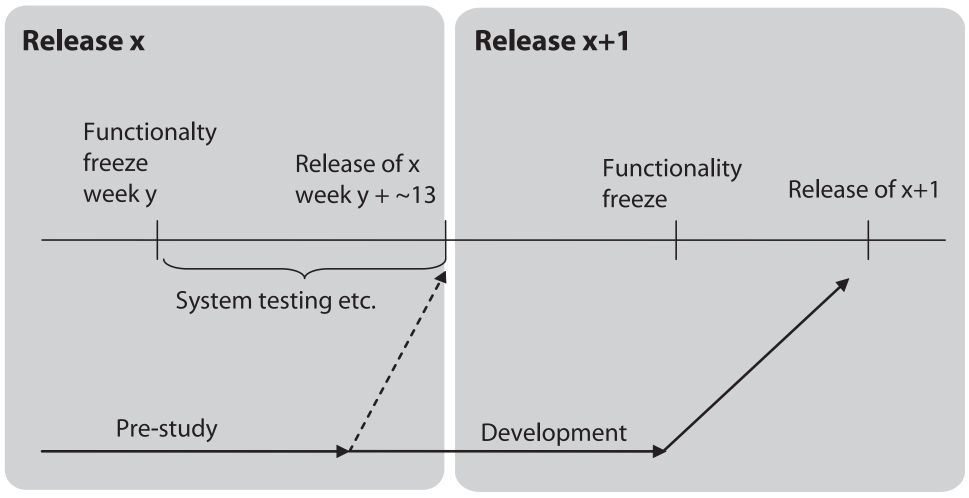

Figure 3.11: Overview of how results are delivered in to a release project.

to detail the requirements received from product management, and for developing time estimations for realization of the requirements.

How to conduct pre-studies is not formally controlled by a process, e.g., there are no specific documents or results which always shall be produced by a pre-study. However, the basic purpose is to refine the requirement(s) and to create a better understanding of the requirement(s). The time and cost estimates, which is a result of pre-studies, are used as an input to product management.

Once a pre-study is completed they normally proceed with realization of the requirement(s). In those cases where a pre-study develops prototype code it sometimes happens that this is directly added to the system and activated/deactivated using configuration options; can be considered to be "beta"-functionality. Normally pre-studies are performed in parallel with a release project, and the pre-studies results in decision material for future releases, as is illustrated in Figure 3.11 (delivery of beta-functionality illustrated using dotted lines in the figure).

One interviewee considers the company to, generally, have improved its release planning process, especially compared to the "chaos" that existed before. However, it would be desirable, if possible, to have some objective way of determining how to balance new customer needs, cost-cut activities, and quality against each other, and preferably some strategy of how to optimize these. In addition, how the budget should be divided between different product areas is an area where improvements are relevant. The product/release planning process is (today) considered to work well, but is experienced to be "heavy" with all meetings (still it is better than the earlier "chaos").

One interviewee lacked a more long-term vision concerning the future for the product, say on 5–10 years perspective. Today most planning is fairly detailed on a 1–3 year horizon, which is perhaps the result of the yearly product plan, a fast moving market

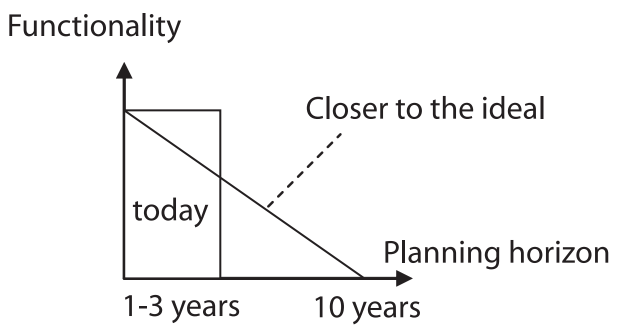

Figure 3.12: Current company planning horizon (1–3 years), and one interviewees ideal picture of planning.

which to large parts is controlled by sales. The interviewee desires a better balance between short- and long-term planning, as illustrated by Figure 3.12, where the plan has high detail in the short-term and low detail in the long-term.

One interviewee desires releases with more strict focus, e.g., releases focused on specific market segments. By having more focused releases the interviewee thinks it would become easier to "sell in" a new release to the market, which would result in a more distinct market impression; today a release is formed by a mix of needs from different areas.

One interviewee noted that it is hard to measure the value of an individual option, e.g., a new customer feature, since it is often of higher value to be able to sell another product. Sometimes it is even the case that is better to give away some product options for free, as long as an order is received for a product; since it results in better economy for the company.

One challenge seen by one interviewee is there being too many different product planning meetings (for different product areas) and too limited focus of each product planning meeting. One possible way of resolving this is to use one product planning meeting per technical field, e.g., one for software and one for mechanical issues. This, on the other hand, would result in there being very many people attending these meetings.

Types of releases

Between two main releases there are usually revisions, which are released on need basis. The first revision after a main release usually contains a lot of functionality/fixes for which there was not sufficient time to make it to the main release. Normally the first revision is released about 4–6 weeks after a main release. The following revisions are released with longer time intervals and contain fewer fixes (most problems are detected quite early).

One interviewee mentioned that "yes, customer features are sometimes introduced in revision releases when there is no time to wait". This is, of course, something they are restrictive with, but some or a few features are usually included.

Scope & Time Horizon

Earlier the company has had 5–6 releases per year, but now they are planning on having strictly 2 releases per year, and revision releases when needs arise. The motivation for having 2 releases (instead of 5–6) is primarily to improve internal efficiency for the company as a whole. Having 2 releases per year increases time-to-market, which can result in lower revenue and increased costs, since product improvements will later become in use. However, at the same time it will reduce the administrative overhead associated with a release, e.g., updating of price lists, educate the organization in the contents of the release, e.g., local sales offices and the part of the company building application on top of the standard software, and update of manuals. All the interviewed people at the company all agree on the benefits of moving to 2 releases per year.

The company is a relatively large organization, which develops and produces complex products, and having 6 months between releases is considered to be a good way of "pacing" the organization; at least when compared with 5–6 releases per year. This will also improve control and coordination of the development projects being part of a release. Yet another reason for having fewer releases is the tough cost-pressure forced by competitors, which increases the need for high internal efficiency in order to remain competitive and profitable within this line of business.

One possible disadvantage of having 2 releases per year is that it can result in development projects aiming to complete by the release date, rather than an earlier point in time. However, it is expected that the total efficiency within the organization will increase.

In case of trouble meeting a release date they consider the following options:

- Moving of independent functions to the next release.
- In case there are dependencies between development projects being part of a release it can result in multiple projects not being able to deliver functionality to a release, i.e., postponed to the next release. However, hypothetically one can consider comprises in exceptional cases, e.g., if some large customer has explicit needs from such a development project. These will need to be handled on case-by-case basis; there is no explicit process for this.
- Customer requirements with no impact on the standard product offering can be excluded from the generic product offering, while still supplying a specific customer with the feature. These features can at the next release be merged into the standard product offering
- It is still possible to postpone a release, in case there are problems with meeting a release date.
- It is possible to move resources between different parts of the organization to strengthen those parts of a release project being behind schedule. However, not all people are capable of performing all kinds of tasks, which puts limits on resources that can be moved.

Earlier the company prioritized the release date, where they promised to deliver all needs “above the red line” (the core functionality) to be included in the release, while all needs below the line was a bonus. Today quality is prioritized before release date.

One interviewee points out that earlier it has happened that upper management has, more or less, ordered changes to an existing release plan without having insight into the consequences of such a decision. With changes in release planning they hope to achieve a better understanding from the organization concerning the impact of making changes to a release plan.

One challenge identified by one interviewee is how to improve time-to-market without lowering the development efficiency. One step in this direction is to improve the modularization of the system such that different components of the system can be released at different dates.

Prioritization mechanism & Objectives

Primarily it is product management that takes care of prioritization of customer needs.

Normally customer needs are prioritized over cost-cut projects. In terms of short-term quality aspects, such as bugs and production stops, these always have highest priority. In the long-term there are plans to improve quality, but usually these quality improvements are speculative investments which result in long return-of-investment time. In case prioritization needs to be done, long-term quality improvements usually have lower priority.

Previously proposed needs/requirements, for which there yet haven’t been sufficient resources, usually have high priority when planning the next main release. Cost-cut activities are ranked according to return-on-investment and risk.

One of the interviewees considers it hard to handle the balance between increasing customer value, initiatives for improved quality, and cost-cut activities. This balance is often handled by “politics, trends, and opinions of upper management”. Furthermore, strong individuals/stakeholders can have easier of getting their ideas through. For example, if a customer is close to making a large order, this can influence a lot of the things that need to be done.

One interviewee desires a more objective way of presenting what is important to do (to alleviate political aspects), and specifically better aid in handling the balance between increasing customer value, quality initiatives, and cost-cut activities.

One interviewee doesn’t consider there to be any value of objective tools/metrics for deciding which requirements/needs to prioritize. Instead the interviewee considers it better to use reference groups, with close customer contact, decide what needs to be prioritized. The interviewee stressed the importance of continuously having the “correct picture of the requirements", i.e., knowing which requirements provides best return-of-investment for the company at the current moment, a picture which constantly changes. To be successful it is also required to implement these needs/requirements in the correct order. The interviewee also considers it to be easier for “strong individuals to get their proposals through”.

One possible downside of using a reference group (as above) is that these people seldom have a good feeling for the quality of the software, and what is possible in that area. This results in some difficulty in prioritizing between quality improvements and new customer features. Again, here it is easier for “strong individuals” to get their will through.

Generally speaking there is a lot of “gut-feeling” that controls what becomes included in a release, and there is a lot of negotiations between different people within the organization before deciding on a release plan.

Several interviewees pointed out that sometimes there are customer problems, or other problems, where one needs to deviate from established processes, and “such flexibility is required, processes isn’t everything!”. There are many things which need to be dealt with on case-by-case basis, and these cases are hard to describe how one acts for each occasion. “It is required with a will to cooperate to find solutions.”

Once per week they go through the list of bug reports and improvement proposals and prioritize what to do. This prioritization is performed together with subject field responsibles. The severity, customer value, and number of customers impacted is estimated for each problem/error. The severity of a problem/error is classified as follows:

- Safety related problems
  These kind of problems/errors are always dealt with.
- Production related problems
- Installation related problems
- Cosmetic problems

They also have estimates on the number of expected errors/problems to be reported, and in case the number of reported errors/problems deviate from the expected it may become necessary to reprioritize the current release project.

All data required for prioritization of a problem/error is stored in a database, where some of the data in the database is mentioned above. The prioritization controls whether a problem will be attended to before the next release. There is one person at the company that decides the action for internally reported problems. The normal action for internally reported problems is to send the problem to a person responsible for the subject field to develop a plan for how to resolve the problem and to perform time estimation. Customer reported problems are treated slightly differently.

One interviewee is very satisfied with how the company handles reported errors/problems, and compared to the interviewee’s earlier experience the company is ahead of many companies in this area.

Stakeholder Involvement

R&D has some impact on what becomes included in a release, since product management during product planning ask R&D to estimate the cost for realization of different needs/requirements. In case something is considered to very hard to introduce, e.g., due to high cost, R&D can motivate this and the need/requirement is removed or modified.

System constraints

The current architecture for the system has been in use for more than 10 years, and consequently there is high investment in it, e.g., exemplified by its millions lines of code and its increasing build time. This, however, makes it harder and harder to introduce new functionality in the system and to test the system. The increased build time has sometimes made developers only check that their code compiles and then directly delivers it to integration testing (without any prior testing), which results in a more costly and "heavy" integration step.

Resource constraints

When working on the budget for the next annual year an engineer is treated as *one* engineer without considering the competence of individual engineers. The budget is set such that all engineers have 100% work load. Later, during project planning and execution some tasks need to be performed by engineers with specific competence, since all engineers aren't capable of performing all tasks, it turns out there aren't sufficiently many engineers available with the required competence. This is partly one of the causes of projects running behind schedule. As one interviewee stated it, "in a sense the early plans, set by the budget, are optimistic and often result in higher development cost".

The competence level at the support organization has been improved. Earlier developers were required to go to customer locations to resolve errors/problems which resulted in undesired disturbances for development projects. Today this is no longer any big issue, partly due to organization changes. Still it is the case for some parts of the application organization that sales require help from R&D, resulting in delays for development projects. However, this often results in R&D becoming knowledgeable in how customer needs need to be prioritized, which sometimes results in R&D having more knowledge than product management in some area. As one interviewee puts it, partly this can be seen as R&D, informally, being part of prioritization.

It has occurred that application needs have reached product planning for the standard platform after the release plan has been set. This becomes a problem since the release plan plans for 100% work load for all resources, which can result in no resources being available to deal with platform functionality required by the application. Another consequence of this is that result in parts of the application engineers not being assigned to any task by product management, which sometimes can be tempting for engineers to develop functionality they consider "good to have".

One interviewee commented the possibility of moving engineers between departments as follows: Today it is rare to move engineers between different departments to cope with overload in different projects. For example, one possibility is to move engineers from application development to development of the standard platform to improve resource utilization. If such flexibility existed one could consider the option of at some point in time use 70% of the available resources for improving the standard platform, and at a later time temporarily place focus on application development. One possible cause and/or consequence of the current lack of flexibility lies in lack of engineers with competence within different product areas. One possible disadvantage of moving engineers between product areas is increased need for training, which possibly temporarily reduces development efficiency.

Technological constraints

A difficulty is in handling the complexity of the product, there are often many projects running in parallel with dependencies between them, and they can be of different nature, e.g., hardware and software. In addition, it can be hard to early see all dependencies.

Tool Support

Within the company they have tried to use Focal Point for prioritization of projects/needs, but found it hard to use when there are big differences in budget for proposed projects/needs, e.g., it is hard to compare projects that have large differences in budget, for example, 20 times difference. Today they don't use Focal Point for requirements/needs prioritization.

Focal Point [5] is primarily used for management of needs/features and to order development projects. For example, in Focal Point they store a list of needs/features proposed a long time ago, but for which there still hasn't been sufficient available resources (due to prioritization). Hence, Focal Point serves as a placeholder for ideas and different needs that exist.

Each need/feature in Focal Point has a description of up to 10 lines concerning what the need concerns. Before a decision is made concerning the realization of a need, a cost-estimation is requested from R&D. Upon reception of the cost-estimation an evaluation is made to see if the estimation has impact on the earlier perception of the priority of the need. If a decision is made to realize the need, the development project order is issued in Focal Point, by marking the need as "new".

All proposed needs/features for the standard platform are stored in Focal Point, but within the application part of the organization this is not strictly enforced, which has its roots in how different organizational needs. Development of the standard platform is usually more complex and involves more people than the application part of the organization.

Needs/Requirements for the next release are stored in an Excel sheet where each need/requirement has up to a 10 line description, a motivation to the need, a reference to the person proposing the need, and a priority (1–4). These needs are prioritized and ranked, and somewhere in this list there needs to be line denoting what can fit within a release's time and resource constraints. One of the interviewees considers this to be a simple, but useful tool during prioritization.

Product Strategy & Competitors

The company has an outspoken strategy to be technology followers. The company's strategy is to have higher quality, easy to use products, more versatile products, open solutions, better error handling, etc. than its competitors. In addition the company should be the most global supplier, at the same time with high local presence. One interviewee estimates the company to work 50% with own innovations and 50% with following competitors' offerings.

Competitors are monitored fairly closely to obtain knowledge concerning their product offering. This knowledge is used to sharpen their own sales argumentation where they are ahead of the competition, but also for planning areas where they are behind the competition.

Some ways of monitoring competitors is by attending trade fairs, buying and testing competitors' products, and to read documentation available from competitors.

Decision material, decision meeting, and consensus

Normally there is a basic business-case for each need, of maximum 3 pages PowerPoint, where numbers are often based on guessing/speculations concerning the particular need's/feature's impact, e.g., market impact, including its development cost. There exists a business-case template.

Documenting release plans

Quality

One interviewee estimates that 50% of the budget is used for quality improvements and bug corrections within some product areas.

One quality indicator is how employees register time for different activities, e.g., fixing bugs and working with improvements is reported separately. However, there is also a grey zone concerning how activities are reported. Today when the product has been on the market for a number of years there is an improvement in quality costs, at least when compared to when the product was introduced. In addition, problem reports and guarantee matters, mainly for hardware, are monitored.

There is a common budget for resolving quality matters, both for correcting bugs and smaller improvements, and new development. How the budget should be divided between new development and quality improvements requires some bargaining/negotiating between product responsibles and the release project. This negotiation takes place during a release project and is followed up once per week.

In case there is need for prioritization between quality improvements and new functionality one needs to internally from R&D create the motivation for the quality improvement, e.g., if there is need for an architectural change. It is primarily a negotiation that controls what becomes prioritized, which is not formally controlled. Proposals for quality improvements can, for example, be initiated by the system responsibles or from specific subject fields depending on the extent of the change. Primarily there are subjective judgments concerning possible benefits of these initiatives, but generally product management have confidence in these estimations and are not questioned.

The company monitors which product options are used on different markets and sometimes options are removed to reduce the amount of articles that need to be maintained. Often cost savings can be made by reducing the number of articles within an area.

At the company they differentiate between internally and externally reported errors/problems. The externally reported problems are mainly handled by the support organization, while there are assigned responsibles for dealing with internally reported problems.

There is no appointed "system architect" at the company, but there is one subject field within the company responsible for the system architecture.

Errors occurring during the product's guarantee time should be reported by the customers, and they are required to report the errors before the guarantee time expires in order to be fixed at no cost to the customer. Using this policy the company counts on all errors being reported.

The company itself monitors its costs for guarantee matters, where for example, work time, material/component cost, and travel costs are included. However, they don't keep track of how their customers are impacted by errors, since such information can be hard to obtain. Still, it is known that production stops at customer installations costs a lot of money.

To resolve quality/guarantee matters it can require resources from development projects. Normally the company tries to handle this by enlisting consultants such that release date and functional content for a release can be maintained. However, this balancing needs to be handled on a case by case basis.

The effect of quality improvements, e.g., for earlier guarantee matters, are followed up by the after market department.

There have been cases where the company has taken on customer deliveries with short time horizon, which has resulted in a number of short-cuts to be taken in order to reach the target delivery date. This is, as one interviewee puts it, "as biting your own tail", since the short-cuts later usually turn up as bug corrections that need to be taken care of to satisfy customer needs. There have been cases where large re-design efforts have been required for earlier developed functionality, i.e., cases where "quick fixes" couldn't be used to resolve the problems.

Life-cycle costs

Hardware development projects do usually have elements of cost reduction, e.g., choice of sub-suppliers and components. There is continuous work to lower the products' hardware costs.

The company uses different strategies for different kinds of customers. For example, for large customer orders there can be considerable order customization, while small customer orders usually are based on a standard offering in order to reduce engineering time for these orders and improve profitability.

Project execution

The company has a release project for the coming release, which in itself can consist of 10–20 different development projects. Once per week (possibly per month) there is a project "sync" meeting, with special focus on project dependencies and project follow up. The purpose of the meeting is to early detect problem/risks with running projects. For example, there are cases when they have relatively late detected that there are project dependencies.

During a release project there can be additional customer features that become included in a project, which in turn requires release planning to be performed continuously during a release project (at least concerning the detailed contents of a release).

New functionality is not allowed to be delivered less than 10 weeks before a main release, in order to allow for time sufficient testing. After the “freeze” for new functionality there is increased resistance to accept deliveries to integration testing. For a delivery to be accepted this late in the process the problem must be critical.

They measure the amount of delivered functionality to integration over time, and they aim to reduce the amount of late deliveries to integration testing. To improve this one interview desires to plan deliveries in an iterative manner to ascertain sufficiently high quality. Instead of developers skipping class tests etc., which results in a peak of internally reported problems that need to be taken care of. Generally the interviewee desires the company to improve in the area of testing and validation.

One interviewee lacked a discussion during the interview concerning software upgrades, which is a problem related to new releases. For example, if a customer has a large amount of computers that need to be updated with a new software release, and this update procedure is manual, then this can be troublesome for customers; since it disturbs customersŠ production and takes a long time.

Mechanics, hardware, and software

The mechanical and hardware content of release does to a large degree control the required software content, i.e., release planning for software is secondary. However, there are areas within the product where hardware and software development can be handled relatively independently. Normally it is required for the software to be completed before the hardware/mechanics.

3.7 Case 7

Company 7 develops and sells a product-line, which is both sold internally as well as directly to external customers. Usually the product-line needs considerable customization to fit a customer installation. The product-line is in large parts based on 3^{rd} party hardware and software components, but is combined with a considerable amount of in-house developed hardware and software. The software functionality is offered to the market in the form of a base system with about 70 options, and an associated license management system. There is high software content in the product-line. Development centers are located in, e.g., Sweden, Germany, India, and USA.

The below description refers to a merged view of the data collected during multiple interviews at the company.

Requirements

One interviewee pointed out that there is a close relation between product planning and requirements management, and there is not much difference between the two at the company. Product/Release planning is conducted in three different steps:

1. Define release profile, which sets focus for the selection of system requirements.
2. Define system requirements.
3. Define product requirements.

Requirements are stored in the tool Focal Point using two main categories:

1. functionality domains or products and
2. into non-functional requirements, such as performance.

One difference between these categories is that the requirements in the first category can more or less be disregarded once implemented, i.e., considered as completed items, while the non-functional requirements remain during the entire product-line life-cycle. Yet, the non-functional requirements can be updated in different releases.

Release/Product Planning

The company uses *release profiles* to describe a high-level strategy for the coming release. Examples of content in such a profile can be improved product properties such as: improved usability and support for more communication protocols, and/or the release profile can be aimed at specific market segments. In a way the release profile is a short-term product strategy for the coming 18 months, which has its roots in the long-term product strategy.

The responsibility to develop the release profile lies on the product-line management. Primarily it is the "gut-feeling" that controls what will become part of the profile, which has its roots in the feedback these people receive from different customers and different people globally across the organization. About 20 people, mainly product managers, take part in deciding the release profile. The work of establishing the release profile can be seen as a first step in planning the next release.

There is no strictly defined process for how to define the release profile, and as one of the interviewees argues “it is not suitable to by mathematics compute what needs to be done”. However, once the release profile is set it is easier to apply mathematics to select the requirements that best match the release profile. The process of developing the release profile is iterative and it is not possible to “compute” the contents of the release profile, instead it is mainly based on “gut-feeling”.

As an aid to decide the release profile they have earlier used, e.g., SWOT analysis and “market-fit-matrix” analysis, where they looked at different markets and where gaps existed in the product offering. One of the interviewees sees the need for “market-fit-matrix” analysis to be less today, when the product has been on the market for a number of years. Another aid in deciding the contents of the release profile is to look at product profitability for each product in the product-line, e.g., sold volumes, possible trends, and changes.

The contents of the release profile can be considered to be controlled by the market, since it is mainly from the market that the needs are collected, e.g., by discussions with major customers. The hard part is, however, in deciding what is most important to act on. Below are some different interviewee responses concerning the release profile and requirements prioritization (not always clear if responses coincide):

1. Deciding the contents of the release profile is not any big problem, consensus is reached fairly fast. One possible reason for this is that these managers are exposed to the same input set, e.g., they meet the same customers.

2. One challenge is to identify the requirements which result in highest customer value. Rarely there is sufficient time to investigate and prioritize all proposed requirements. For example, one early plan contained about 420 requirements, which was later reduced to about 300 requirements, but only about 260 requirements were implemented and released. In addition, during the project there were about 160 formal change requests. About 30% of the 300 requirements didn’t make it to the release and about 15% were new requirements. In summary, there were many requirement changes during the project, both addition and removal of requirements.

3. A complicating issue in deciding the release profile and requirement set is the geographic distribution of the company, both of development centers and of product managers, which results in overhead, since it is not always easy to get hold of people and it increases the need to travel.

4. “It is required to plan correctly to avoid delays.”

5. “There is a well-working process for collecting the right requirements.”

The release profile forms the basis for requirements prioritization, which is performed using Focal Point on the requirement set considered for the next release. Requirements prioritization is most often based on *value*, e.g., customer value, and it isn’t always they look at the cost aspect using Focal Point. Development cost can often not be determined until input is received from R&D, and such estimates/planning is normally done during project planning.

Two interviewees pointed out that during the beginning of a product life-cycle it is usually more important to reach the market rather than focus on quality aspects, i.e., in early life-cycle phases focus should be on items visible in the price list. Later, when a product has been on the market for some time and feedback is received concerning the product, then more focus can be placed on quality aspects. Today more of the focus at the company lies on quality aspects. As an example of a quality aspect, it may be the case that installation requires too long time and there are requests to reduce it by 50%.

Types of Releases

The company uses the following kinds of releases/development:

Main release The main release, which is the focus of this chapter.

Service packs Consists of important bug fixes and sometimes functionality extensions, e.g., extensions promised to specific customers. Usually released in 6 month intervals after product release; the interval is increased with time.

Bug fixes Exclusively contains bug fixes. Usually released once per month.

Customer funded development Development which is considered being of high importance for some customer. The customer pays the main part the development cost. Performed when needs arise.

Scope & Time horizon

Earlier the release cycle was fix and set to 1 year, however, nowadays the release cycle is fix and set to approximately 18 months, which is considered to result in a suitable efficiency for the development organization; project planning, execution, test, upstart activities, documentation, training of the organization, project close out etc. have associated overhead. By having a longer release cycle it is possible to get more time for development of new functionality.

In case there are problems with reaching the target dates for a project or projects, being part of the release, there are three different possibilities:

1. change scope of the release by reducing its functionality,
2. post-pone the release date, or
3. redistribute resources within the organization (also globally).

However, it is always preferred to keep the target date for the release. The motivation for this is that it is more important to reach the market with the functionality that could be developed within 18 months, rather than letting a few projects delay the entire release. The project(s), or functionalities within projects, which are late may cause loss of a few orders, still it is more important to reach the market with the other product improvements such that the effect of the large amount of resources spent on the release becomes visible. At some earlier occasion it has been decided to post-pone the release date, today the release date is prioritized higher.

In case there are critical errors in a released product it can happen that resources are taken from development projects to resolve the error, which can have impact on a project's time plan.

Objectives

As one interview responded, "it is almost always possible to estimate the value of a requirement!" For example, for usability where the profit may seem hard to estimate it is possible to use estimates from different application engineering units, e.g., if it takes 30% longer time to perform customer customization and the business volume is x billions it is possible to form an opinion on profit impact.

Prioritization mechanism

The release profile is used as an aid when prioritizing requirements in Focal Point, and normally the resulting list of requirements from Focal Point controls what will become included in the next release. However, usually complemented with some "easy wins" lower down in the list, e.g., improvements or functionality which have low cost while still providing considerable customer value (in relation to its development cost).

During earlier prioritization sessions in Focal Point they prioritized *value* for the own part of the organization, while today the value for the entire organization is prioritized. This change has quite an impact on what becomes prioritized.

Stakeholder Involvement

Earlier some stakeholders have had low influence over the work of defining the release profile, and all stakeholders haven't had the possibility of monitoring/proposing changes to development projects. In the future it is desired to involve different stakeholders to a higher degree in order to improve the possibility of delivering what stakeholders really expect. Preferably they would like to see stakeholders participate during development projects, e.g., to provide feedback on GUI designs etc. Stakeholders mentioned during one interview was internal customers, sales managers, channeling partners, technical sales support, and the support organization.

R&D is involved during the iterative definition of the requirements for the release, where R&D provides time and cost estimates for selected requirements. During this time it is also possible for the product organization to add quality aspects they consider important to resolve. During the interviews we have not had time to resolve exactly how quality aspects are prioritized against, e.g., new functionality.

System constraints

The existing product offering, and the properties of these, makes some things easier to implement than others. Sometimes there are "low hanging fruits", i.e., requirements, which for a low cost, still provides customer value. This becomes visible to product-line management via the estimations made by R&D for cost and time. (Requirements which requires 1 week to implement and provides low customer value has higher probability of being included in the release than requirements that require 1 year to implement (even if it provides higher customer value).)

During release planning consideration is also taken to which versions of 3^{rd} party components will be used. The more versions the company is willing to support, the more testing is required to verify that it works. Furthermore, the more resources spent on testing, the less can be spent on development of new functionality; more resources will be spent on integration and verification.

Resource constraints

The initial release plan is iteratively developed between the product-line management and R&D, usually using "guesstimates" (partly guessing and partly estimates). In later stages when the release plan scope is more defined, more time is spent on estimates. When R&D performs estimates they take into consideration the amount of available resources and their competencies. For example, if one department has 5 resources that will have impact on the pace in which the department is able to deliver new functionality.

During project planning requirements are assigned for development such that 100% of the project budget is used, but assuming each engineer has 32 hours of available engineering time per week; normally time is also needed for training, administration, meetings, illness etc. However, there is no slack in the budget to handle customer needs discovered during project execution. In such cases *change requests* need to be formally dealt with to modify the scope of a project(s).

Product-line management estimates how many items on the prioritized requirement list from Focal Point that R&D is capable of handling. For this set of requirements a report is generated from Focal Point which goes to R&D for cost and time estimation. When the result is returned to product-line management they decide which requirements will be part of the new release. Normally this work is carried out in an iterative manner where, e.g., R&D requests more information regarding how a certain requirement should be interpreted or suggest alternative methods of implementing a requirement. This iterative work ends with the decision on the contents of the system requirement specification.

Another complicating resource issue is that budgets for different development departments and release time are decided before the content of the next release has been decided. This makes it even harder to optimize for customer value in the release.

Technological constraints

Tool Support

The tool Focal Point [5][8] is used within the group (mainly product managers) deciding the release profile, hence, it is not a tool used by the R&D organization. All requirements are documented in Focal Point and it also holds the status for each requirement, e.g., accepted for prioritization, accepted for implementation, new etc. The database also contains requirements which aren't included in the next release, thereby no requirements are lost. Product-line management uses Focal Point to generate reports, containing prioritized requirements, which are passed to R&D for, e.g., performing resource estimations.

Several of the interviewees respond that they are very satisfied with the tool Focal Point.

Product strategy & Competitors

There are a large number of competitors, within different technology areas, and therefore it is hard for the product-line management to keep track of all competitors. Basically they try to monitor the offering of a few larger competitors, but mainly focus is placed on customers' requirements rather than looking at what competitors offer (and customers' requests can be the result of offerings by competitors). However, there are some people within product management that are assigned to monitor specific areas.

Two of the interviewed persons consider that monitoring of competitors could become more structured, "there is no established process", while one of the interviewed persons considers that the sum of all activities aimed at monitoring competitors results in good coverage of the competitors. All interviewed persons describe that it is primarily customer requests that control what is being developed, rather than following offering by competitors.

As one interviewee pointed out, if information is obtained indicating that a competitor has a new offering, e.g., in form of a sales brochure, it means the competitor already has a sellable product and then you are already too late, since it will take approximately 2 years before it is possible for the company to deliver new functionality to the market; mainly due to the release cycle of 18 months.

Patents are continually monitored, and they patent own ideas.

Decision material, decision meeting, and consensus

It is mainly opinions and thoughts from people in the product-line management that controls the content of the release profile. It is rare to perform any economic forecast, or similar, for the profile, instead it is mainly the feedback from different stakeholders that controls the contents of the profile. For the major system requirements it has occurred that ROI analysis has been performed.

Earlier the company has tried to let the sales organization put numbers on how proposed requirements will impact different market segments, e.g., by ROI analysis. However, for this company there is often a long supply chain before it is possible to judge how the end-customer is affected, since there is often channeling partners etc. During this approach it has also happened that one part of the organization has claimed that they would have no sales at all if not a particular functionality was included in the next release, which was not true.

One of the interviewees considers ROI estimates to be difficult, since these are often very sensitive to some parameter. By adjusting this parameter it is often possible to result in both profit and loss. The part of the organization making the proposal often adjusts such parameters to support their own proposal. To avoid this one often relies on “gut-feeling”, or even, “letting the one yelling loudest get his will through”.

Documenting release plans

Quality

Resolving quality issues doesn’t follow the normal release planning process, i.e., planning of the main release. This is mainly an issue which is handled within each product area. Critical errors which occur at customer locations are always fixed, and in addition a number of less severe errors are fixed. The goal is, of course, to find all errors during the quality assurance process, however, it still happens that errors reach customers.

The number of support cases at R&D and at the support organization, among other things, is followed up on a weekly basis. This feedback controls the contents of the service pack release and bug releases, but not to any larger degree the contents of a main release.

One interviewee from product-line management desires a better dialog with R&D concerning quality issues, since today it is hard to raise these issues to product-line management due to the problem of quantifying quality aspects. Furthermore, it is desired with information on what can be done, and the possible effects of improving quality.

Generally it is considered hard to quantify the value of quality improvements such as, e.g., refactoring of code/architecture. One interviewee mentioned that it is possible that too little focus has been placed on quality in some earlier releases. There is no established process at the company concerning how to prioritize development of new customer features vs. improving quality.

When architectural changes are proposed a risk analysis is performed, which analyzes what will happen if the architectural change is made and what will happen if it is not performed. One interviewee pointed out that it is hard to compute the benefit/profit of performing an architectural change, especially when compared with development of new functionality. A contributing problem is that software changes rapidly, especially when compared with rather conservative business in which these products are used.

There is no formalized process for follow-up of customer satisfaction, i.e., how well the products are received and works at customer locations. However, there are discussions with both internal and external customers concerning what they consider needs to be done, which in a way captures customer satisfaction. The company uses no metrics for customer satisfaction.

Follow-up of forecasts of how new functionality will impact market/customers is in essence never performed. It has been performed in a few cases, but almost always on initiative by a local product manager.

Life-cycle costs

Project execution

A release is conducted as a project, which contains about 20–30 sub-projects. The release project leader is responsible for monitoring the release, including integration aspects etc. Integration aspects are complicated for projects of this size, and needs to be planned carefully. The product-line management doesn't have any scheduled monitoring activities of the release project, but becomes involved if needs arise. Such needs can arise if R&D needs clarifications concerning some requirement, discussion of alternative ways of implementing requirements (with different impact on cost and user aspects). Usually some iteration is required in relation to this.

The sub-projects part of the release project are normally working on 200–300 market requirements. It is hard to monitor project status and project plan deviations for this many projects, not due to lack of processes and methods for project follow-up, but rather due to the high complexity. Today project follow-up is documented using an Excel sheet, but it is hard to get a good overview; it is "like looking through a straw". Another complicating issue is that changes to one market requirement can have impact on several different development projects.

All interviewed people at the company consider the used processes to be good, as exemplified by "it is a good support which is well-established in the organization", "it has flexibility and can be adapted to different situations", and "there is a clear process defining which specifications to write".

Changes to project scope etc. are handled using *change requests*, which is a formal way of making changes to running projects. A change request should specify, e.g., market impact. Today these mainly contain qualitative data and rough estimates, probably due to the difficulty of determining how a change will impact different markets and different customers. Change management within projects was mentioned by one interviewee as an area which needs improvement; they are currently looking at tools to support the process.

One interviewee pointed out that today the product-line management has a form a "total" responsibility for the product-line, but there is no such role for technical aspects of the product-line. There are such roles on product level, but not on system level. Sometimes the persons that have this responsibility on product level have to step in and take this role on system level, and sometimes the product-line management has to resolve this. There are issues which fall between the current roles. One interviewee sees the need for a technical responsible on product-line level.

Part of development projects are also quality assurance roles whose purpose is to verify that sufficient quality has been obtained. For a product to be released it must pass quality assurance tests, and products/releases which fail quality assurance testing aren't release. Quality assurance is treated seriously at the company.

Chapter 4

Summary

Release planning is a company-wide optimization problem involving many stakeholders where the goal is to maximize utilization of the often limited resources of a company and turn them into business benefit. In this report we have documented data collected via interviews concerning the state-of-the-practice for release planning in industry today. This material acts as "raw material", which is the basis for further analysis[1].

[1]The reason for the existance of this report is that scientific papers, based on this data, should be able to reference this report such that reviewers can get a better understanding of the extent of the study.

Appendix A

Letter to Companies

The following introduction to the case-study (written in Swedish) was sent out to interview subjects (and contact persons) before the interviews.

Hej!

Vi arbetar på Mälardalens Högskola med forskning inom release planning och mjukvaruarkitektur och hoppas på att Du är villig på att ställa upp på en intervju där vi avser undersöka hur release planning utförs i industrin idag, dvs., en studie av state-of-the-practice.

Vår målsättning är att intervjun skall ta c:a en timme. Du kommer själv ha möjlighet att granska och kommentera de slutsatser vi dragit från intervjun, allt för att undvika eventuella missfrstånd. Vi kommer dessutom att utföra liknande intervjuer på ett flertal andra företag inom snarlika produktområden och sammanställa resultaten i en rapport. Vår förhoppning är att när väl intervjuerna och analys av insamlat data är genomförda att anordna en workshop där vi presenterar resultatet av studien. Under denna workshop kommer Du ha möjlighet att utbyta erfarenheter med personer i liknande position som du.

En del bakgrund till problematiken med release planning etc. finner Du nedan.

Med vänlig hälsning
Markus Lindgren, Anders Wall, och Christer Norström

> Deciding what to include in future software releases of a product is an important activity with possibility of having major impact on the profitability of a company, since customers usually buy a product based on the value it provides in relation to its cost and quality. This problem is referred to as *release planning*, *product roadmap planning*, and sometimes *requirements prioritization* in different research communities. The aim of release planning is to maximize the value of a release (or releases), while mini-

mizing the cost for developing it, which in essence optimizes the return-of-investment (ROI) for the manufacturer of the product.

In addition, the release planning problem usually has constraints on what can be included in a release, for example, there is often a budget for a release and there are limited available resources. There can also be dependencies between features/requirements which have impact on how they can be allocated to releases. This, and many more items, are input to the release planning process which need to be considered.

What is being published as research results in an area is rarely what is being used in industry. So one aim of our study is to capture state-of-the-practice in industry related to release planning. Examples of areas of our study are: the release planning process, relevant input material to the release planning process, and tool support for release planning.

In the above introductary text we have only briefly touched upon a few of the issues relevant during release planning. This is mainly for you to have an idea of what the study is about. In the end our goal is to improve the release planning process to become more efficient and better meet the needs of industrial companies.

Bibliography

[1] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice* 2^{nd} *edition*. Addison-Wesley, 2003.

[2] P. Carlshamre. Release Planning in Market-Driven Software Product Development: Provoking an Understanding. *Requirements Engineering*, 7(3):139–151, 2004.

[3] STAGE-GATE. Web-site: http://www.prod-dev.com/stage-gate.shtml, March 2007.

[4] Telelogic DOORS — Requirements Management for Advanced Systems and Software Development. Web-site: http://www.telelogic.com/products/doors/doors/index.cfm, March 2007.

[5] Telelogic Focal Point. Web-site: http://www.telelogic.com/corp/products/focalpoint/index.cfm, March 2007.

[6] IEEE Std 1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process, 2005.

[7] H.-W. Jung. Optimizing Value and Cost in Requirements Analysis. *IEEE Software*, 15(4):74–78, 1998.

[8] J. Karlsson and K. Ryan. A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, 14(5), Sept.–Oct. 1997.

[9] G. Mustapic, A. Wall, C. Norström, I. Crnkovic, K. Sandström, J. Fröberg, and J. Andersson. Real World Influences on Software Architecture - Interviews with Industrial Experts. In *Proc.* 4^{th} *Working IEEE/IFIP Conferance on Software Architecture (WICSA)*. IEEE Computer Society, 2004.

[10] G. Ruhe and M. O. Saliu. Art and Science of Software Release Planning. *IEEE Software*, 22(6):47–53, 2005.

[11] M. O. Saliu and G. Ruhe. Supporting Software Release Planning Decisions for Evolving Systems. In *29*th *Annual IEEE/NASA Software Engineering Workshop*, pages 14–26. IEEE Computer Society, 2005.