# Towards a Capability Model for the Software Release Planning Process — Based on a Multiple Industrial Case Study

Markus Lindgren[1], Rikard Land[2], Christer Norström[2], and Anders Wall[3]

[1] ABB Force Measurement, Västerås, Sweden
markus.lindgren@mdh.se
[2] School of Innovation, Design, and Engineering
Mälardalen University, Västerås, Sweden
rikard.land@mdh.se, christer.norstrom@mdh.se
[3] ABB Corporate Research, Västerås, Sweden
anders.wall@se.abb.com

**Abstract.** Software release planning is an important activity for effectively identifying the customer needs generating best business, especially for incremental software development. In this paper we propose a capability model for improving the release planning process of an organization. Using this model it is possible to 1) determine the capabilities of an organization's release planning process, and 2) identify areas for improvement. The model is based on empirical data from a multiple case study involving 7 industrial companies, all being producers of software intensive systems. We also provide examples of how the proposed capability model can be applied using the companies from the study.

## 1 Introduction

Release planning can be seen as 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 [1]. As input to release planning is a set of *needs* that, when implemented as part of a product, provides some business/customer value. Release planning results in a decision of what to include in future *release(s)* of a product, and consequently, a decision of what *not* to include; normally the cost of implementing all proposed needs exceeds the budget allocated to a release. Thus, the *set of needs* needs to be prioritized in order to maximize the business value of the included needs. In addition, there are constraints that must be considered during release planning [1], such as, time-to-market and dependencies between needs. An overview of some relevant aspects of release planning is illustrated in Fig. 1.

Poorly performed release planning can result in "wrong" features being released to the market and/or being released at the "wrong" point in time. Another possible impact of poor release planning is inefficient use of the resources available to an organization. Ultimately, release planning impacts how successful and profitable an organization can become.

A. Jedlitschka and O. Salo (Eds.): PROFES 2008, LNCS 5089, pp. 117–132, 2008.

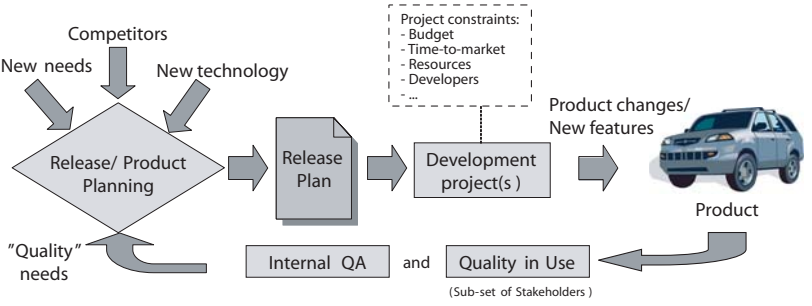


**Fig. 1.** Overview of relevant aspects of release planning

In this paper we propose a capability model for improving the release planning process of an organization, which is based on empirical data from a multiple case study on software release planning in industry [2,3,4]. Using this capability model it is possible to:

- Determine the capabilities of an organizations release planning process.
- Identify areas for improvement.

We also illustrate how this capability model can be applied to determine the capabilities of a company's release planning process, using the companies from our multiple case study as examples.

The outline of this paper is as follows: Section 2 presents a selection of related work, Section 3 describes the research method, Section 4 presents the release planning capability model, Section 5 presents examples of using the release planning model, Section 6 provides a more extensive release planning example from industry, putting things into context, and Section 7 discusses future work. Finally, in Section 8 we summarize the paper.

## 2 Related Work

Release planning research is mainly focused on formalizing the release planning problem, typically by formulating the problem as an optimization problem, where customer value should be optimized while subject to a set of constraints [1,5]. In addition, there are a number of tools being developed implementing these algorithms [6]. However, there is also work indicating that the release planning problem in itself is "wicked" and therefore hard, and possibly unsuitable, to formulate as an optimization problem [7].

In [8,1] two different approaches to release planning research are discussed, referred to as the *art* and *science* of release planning. The approaches formulating release planning as an optimization problem belong to the science approach, while we in this paper are more focused on the art of release planning. We discuss how a company can perform release planning from a more practical point of

view. Furthermore, it is rare to consider how the existing system impacts release planning, exception being [8,6], which again are science approaches.

Requirements engineering is related to release planning, e.g., prioritization of needs is a common problem. However, release planning is a more general problem, since it, e.g., considers resource constraints [6]. Focal Point is one example of a requirements prioritization tool based on the work presented in [9].

Research within the area of process improvement is active, where the perhaps most well-known model and most used in practice is the CMMI [10]. CMMI is focussed on an organization's capabilities and maturity of running product development project(s). It specifies practices that must be adhered to in order to reach a specific CMMI level, where CMMI level 1 represents an organization with lowest maturity and CMMI level 5 represents the highest maturity. However, CMMI provides little detail on how to perform release planning. The CMMI process areas being related to release planning are *project planning*, *requirement development*, and *requirement management*. Within these process areas there are practices for capturing and managing requirements, but when it comes to how to select which features to include in the next release of a product there is no information or practice; CMMI in several cases merely states "resolve conflicts". This paper can be seen as an extension of CMMI that addresses some areas where CMMI provides little or no information.

There are also a number of standards which have parts related to release planning, for example, IEEE Std. 830:1998, 1220:2005, 12207:1996, and 15288:2002. IEEE Std. 830 specifies how a complete, correct, and non-ambiguous software requirement specification should be written. IEEE Std. 1220 specifies the tasks required throughout a system's life cycle to transform stakeholder needs, requirements, and constraints into a system solution. IEEE Std. 12207 specifies a common framework for software life cycle processes, similarly IEEE Std. 15288 defines a framework for systems engineering life cycle processes. As these are standards, they rarely state how to perform a specified required task, instead they state that the task must be performed (somehow). In this paper we provide more "hands-on" approaches to release planning; surprisingly little is concerned with release planning in these standards.

To conclude, existing research provide little information concerning how to improve an organizations release planning process, and there is little work on how to consider the quality of the existing system during release planning.

## 3 Research Method

In this paper we investigate the effectiveness of performing software release planning. We do this by proposing an initial model expressing the capabilities of an organizations release planning process, which aid in identifying areas for improvement. The model is a first attempt in this direction which we expect to be refined and become more detailed over a period of time. This work is based on empirical data collected during a multiple case study involving 7 industrial companies [4]. For confidentiality reasons there are no company names, no names of

interviewed people, and no absolute numbers on, e.g., budget, in this paper nor in [4], but where possible we present relative figures.

In our study we have used semi-structured interviews as the primary data collection method (with a common line of questions/topics), sometimes complemented by documents received from the interviewees. The main alternative, direct observations, has not been used due to the topic being studied containing company sensitive information and partly due to practical limitations.

In conducting the study we have followed the recommendations by Yin [11] for multiple case studies. We have addressed *construct validity* in multiple ways. First, there have always been two researchers present during each interview in order to reduce possibilities of misunderstandings. Second, interview notes have been taken during each interview, which have been sent to interviewees for approval [4]. Third, we have had two test interviews to improve our interview setup. In total we have interviewed 16 people (excluding test interviews), typically 2–4 persons per company to achieve data triangulation.

To strengthen the *internal validity* of our study we have used multiple researchers when performing analysis and made use of pattern-matching techniques, and we have considered rival explanations. To increase *reliability* of our study, all collected data, and derivations thereof, are stored in a database accessible only to the researchers in the study, e.g., interview notes and merged notes per company. In addition, the study design is documented, which includes the interview questions. Using this material it is possible to trace conclusions to collected data, and vice verse. To counter *researcher bias* multiple researchers have been involved in most of the steps of this study. Furthermore, companies that the researchers in the study are affiliated with are excluded from the study.

Thanks to industrial contacts we have been able to find a relatively large number of companies and persons willing to participate in our study, which aids in increasing the *external validity* of our results. However, there is risk that the selection is not fully generalizable to other domains and/or nationalities/cultures. In selecting people to interview we have asked our contact person(s) at each company for references to people working with release planning. Our interviews have mainly been with product managers, managers, and project leaders.

All companies in the study develop software intensive embedded systems with a typical life cycle of 10–20 years. However, the companies are in different product domains, e.g., automation, telecommunication, and automotive. Table 1 presents some relative data concerning the characteristics of products developed, produced, and sold by the studied companies in order to provide a feel for their main characteristics. In Table 1 *Volume* refers to the produced product volume, while the rows *% Software*, *% Electronics*, and *% Mechanical* is our subjective judgment of the products' software, electrical, and mechanical content, which in turn reflect the amount of resources these companies invest in these areas. Case 3 is excluded from the table since it is a management consulting company that has no products. The case numbering in Table 1 is consistent with [4,2].

We are partly using a grounded theory approach [12] in this research, since we define a model based on observed data. The model we present in this paper

**Table 1.** Charachteristics of companies in the study, where VH = Very high, H = High, M = Medium, L = Low, and VL = Very low

| Case | 1 | 2 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| Volume | VH | H | VH | L | M | L-M |
| % Software | L-M | L-M | H | L-M | M | H |
| % Electronics | M | M | H | M | M | M |
| % Mechanical | H | H | L | H | M | L |
| Employees | H | H | VH | L-M | VL | VL-L |
| % in R&D | VL | VL | H | L-M | VL | VL |

has partially been validated in a workshop with participants from our study. However, since we created the model based on collected data, the same data cannot be used to validate the model. Nevertheless, it can be used to motivate and illustrate the model until further studies validate it.

Proper validation of the model requires a baseline with which to compare, for example, the state before changing the release planning process in an organization, and then collect data after introducing the change; as has been done for CMMI [13]. We have not yet reached such a state in this research.

## 4 Improving the Release Planning Process

In its generalized form, release planning can be considered to consist of three different process activities, as illustrated in Fig. 2:

**Elicit Needs.** Collect stated, and unstated, needs from the different stakeholders. Other literature, e.g. [8], refers to this activity as requirements elicitation, which typically refers to a phase within a development project. We mainly refer to need collection occurring prior to forming the development project.

**Make Release Decision.** Prioritize the needs such that the cost and schedule for realization fits within the constraints of the release, and decide the contents of the release. Again, this is an activity that primarily occurs prior to the development project.

**Realize Needs.** Typically performed as product development project(s) within the research and development (R&D) part of the organization, where the prioritized needs are implemented as part of a product(s).

These process activities can, but need not, be performed in sequence; typically these are continuous activities with data flow between them.


**Fig. 2.** Overview of the release planning process

In this section we present a capability model for improving an organization's release planning process, inspired by the Test Process Improvement (TPI) [14] framework, and partly by the CMMI [10], therefore there are some similarities. Our focus is placed on need elicitation and making the release planning decision, while we are aware of there being other important areas within release planning as well, such as, choice of time horizon and resolving need dependencies.

Each of the following sub-sections describe *key-areas* within the three process activities from Fig. 2. For each key-area a capability scale is presented, with levels from A–D, where A represents lowest capability and D represents highest capability. Using these descriptions it is possible to pinpoint the capability for an organization within each key-area. While describing the key-areas we also present some examples of how the key-area can be applied in practice, using examples from our multiple case study [4]. However, it should be noted that we do not suggest that it is always economical for an individual organization to strive for level D in each key-area.

We have derived the set of key-areas based on what we have observed as being important activities in our study; the set we present is in no way guaranteed to be complete. Within each key-area we have ranked observed data from the cases within each area to form the capability levels. The ranking has partly been validated in a workshop with participants from our study. However, we expect our proposed capability model to be refined and detailed in the future; this is only a first attempt at building a capability model for release planning. For example, in this model there are capability levels from A to D, however, it is not necessary for there being four levels within each key-area. Furthermore, the model has been devised with a focus on software, although it may be possible to apply the model in other domains as well.

In TPI [14] there are also key-areas with levels A–D. In addition, there is a *test maturity matrix* relating the level within each key-area to a test maturity scale, from 0 to 13. For example, for an organization to have rating 3 it is required to have level A in the key-area *Estimation and planning*, level B in *Test specification techniques*, etc. In our work we currently haven't reached a state where we can present a similar maturity matrix for release planning. Yet, the key-areas will allow an organization to identify possible improvement key-areas.

### 4.1 Elicit Needs

We have identified the following key-areas within the process activity elicit needs: *need Elicitation* and *need documentation*.

**Need elicitation:** Refers to how and from which stakeholders needs are elicited. Elicited needs are prioritized, in other key-areas, and a decision is made of what to include in a release. The capability scale is as follows:

**Level A. Adhoc.** Needs are collected when opportunities arise, for example, during meetings with customers or other stakeholders. Typically this activity is unstructured and performed by product management.

**Level B. Formal Path.** Each (important) stakeholder group has a formal path for passing need requests to product management. Typically these needs are collected prior to upcoming release planning decisions. This formal path should be described in the process description for the organization.

A stakeholder group refers to a specific "type" of stakeholders, e.g., the end-customer can be one such type. Other such possible types are developers, testers, and commissioners. What differentiate these are that they each use the product in different ways, and therefore also have different needs.

**Level C. Stakeholders Prioritize Needs.** Needs are collected using both A and B, but with the addition of each stakeholder group also assigning priorities to the needs. Product management, which makes the release decision, receives a set of prioritized need lists, one from each group, and is required to make the release plan decision and to be able to motivate this decision.

**Level D. Stakeholders Rate Needs Based on Product Strategy.** An extension of C where the internal stakeholders of the company assign priorities based on the product and/or company strategy; external stakeholders prioritize needs according to level C.

**Example 1.** Case 1 fulfils level B by having three parts within the organization, which each focus on a separate area of the product [4]. These areas are *product features*, *product quality*, and *cost-cut* (mainly related to production cost), which each propose needs to product management; illustrated in left part of Fig. 3. In a similar way Case 4 has a formal path for collection of feature needs and quality needs; illustrated in right part of Fig. 3.

**Example 2.** Case 4 develops a product platform used by two other parts of their organization, $O_1$ and $O_2$ [4]. One way in which Case 4 fulfils level B for need elicitation is that product management for $O_1$ and $O_2$ propose needs to product management for Case 4. In addition, system responsibles from $O_1$, $O_2$, and from Case 4 propose needs to product management for Case 4, as is illustrated in Fig. 4. Furthermore, they apply a principle, called "one-voice", where each group (i.e., each line to product management in Fig. 4) prioritize the set of needs before passing them to product management for Case 4, thereby fulfulling level C.

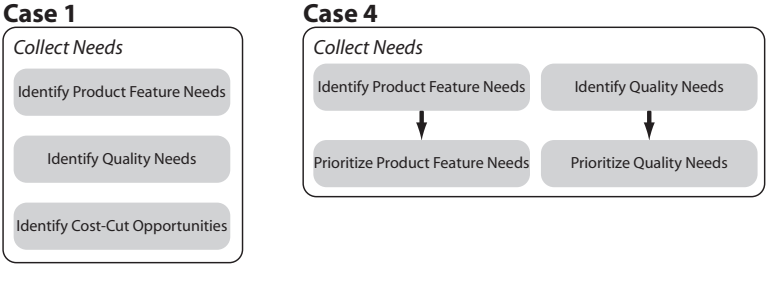


**Fig. 3.** Two examples of a formal path existing for need elicitation

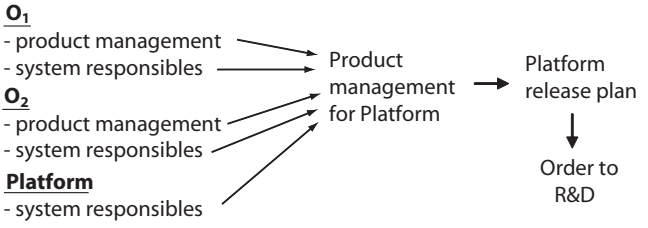


**Fig. 4.** Example of having a formal path for need elicitation

**Documentation:** For a need to become eligible for prioritization into a release there typically needs to be some documentation, e.g., a short description of what business benefit the need aims to fulfil. The capability scale is as follows:

**Level A. Adhoc.** Only a short description of the need, if any.
**Level B. Template.** A common template for need documentation consisting of at least: a short description of the need, an initial cost estimate for realizing the need, an initial return-of-investment calculus, and a statement concerning the consequences on the existing system of introducing the need. Typically, this documentation should be at most one page.
**Level C. Tool Support.** The information from Level B are stored in a tool/database, which can be accessed by all people involved in need elicitation.
**Level D. Type of Need.** An extension of Level C where the needs are classified according to at least the following types: new feature, quality improvement, cost-cut.

**Example 3.** In our study [4] all the companies use some form of template for the proposed needs. However, its form range from a short statement, a one page statement (as in level B), to templates with 3 PowerPoint slides.

### 4.2 Make Release Decision

These are the key-areas we have identified belonging to making the release decision: *decision Material*, *product strategy*, and *release plan decision*.

**Decision material:** In addition to the need documentation described in Section 4.1 there can be different types of studies that refine the needs and produce decision material, which complement the mentioned need documentation. The purpose of the decision material, produced via studies, is both for increasing confidence of data and risk reduction. Typically this is related to, e.g., refining cost-estimates, determine consequences for the existing system, refining return-of-investment calculus. The capability scale is as follows:

**Level A. Adhoc.** No formal decision material is used, instead the decision material is formed by the "gut-feeling" [2] of the individuals involved in making the release plan decision.

**Level B. Unstructured Pre-study.** A pre-study is performed with purpose of refining a need proposal. The results produced by the pre-study partly depend on which individual(s) perform the pre-study, and partly on the people ordering the pre-study.

**Level C. Structured Pre-study.** A structured pre-study [2], compared to an unstructured (level B), has a standardized set of issues which should be investigated in the pre-study. It considers alternatives, as is described in the process area Technical Solution in CMMI [10].

**Level D. Feasibility Study.** This is more oriented towards the solution for how to realize the proposed need(s) into the existing system, and investigating different alternatives. In addition to performing a structured pre-study, as in level C, a feasibility study is performed with the goal determining how the proposed need(s) can be realized into the existing system, the resources required to complete the task, refine cost-estimates, and address market issues; see feasibility study on Wikipedia.

**Example 4.** The development projects in Case 1 are to large degrees concerned with production issues, since their products have large mechanical content (see Table 1) and the production is both complicated and costly. Therefore, before any decisions are made they need to know the consequences for production (as far as possible). These consequences are investigated via structured pre-studies, illustrated in left part of Fig. 5.

Case 4 on the other hand delivers a product platform to $O_1$ and $O_2$ (previously mentioned in Example 2), which in turn use the platform to develop products. In case the platform is delivered at the wrong point in time or with too poor quality, this will have consequences on the efficiency for $O_1$'s and $O_2$'s development projects. To reduce this risk, and to improve confidence in the decisions being made are the correct ones, Case 4 use both pre-studies and feasibility studies; before and after each study it is possible to re-prioritize the proposed needs, illustrated in Fig. 5.

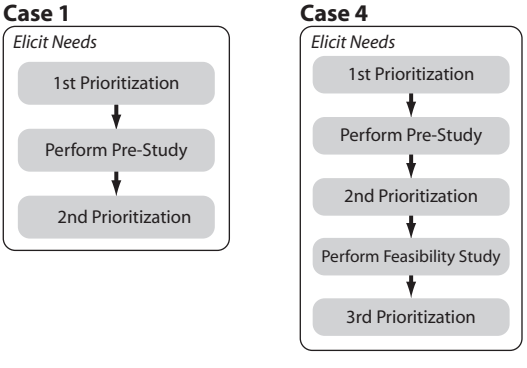


**Fig. 5.** Two examples of using studies to refine need proposals

**Product strategy:** Having a well-defined and clearly communicated product strategy often helps employees within an organization to know what to strive for; also an aid in daily prioritization of tasks. So far we have not yet reached a state where we can present a capability scale. Still, we present some examples from our industrial cases [4]:

**Adhoc.** Each release basically has its own focus and is not directly related to any product/business strategy.

**Release Profile.** Case 7 uses a *release profile*, which sets the top-level priorities for the next release. For example, the profile can be aimed at improving usability and/or extending the set of supported communication protocols.

**Product Strategy.** Each product produced by Case 1 has an attribute profile, consisting of more than 20 attributes, defining the target properties of the product. Furthermore, the attribute profile is clearly linked to the company strategy. Case 2 also has a clearly defined product strategy, with 8 core values and 3 premium values; these values are used when determining the business value of proposed development projects.

Another possibility expressing the product strategy is by using the *measure of effectiveness* (MOE) defined in IEEE Std. 1220:2005, which is an explicit mathematical expression "... by which an acquirer will measure satisfaction with products produced by the technical effort." The expression should capture the top-level goals, i.e., should not contain too many details. The difference compared to the previous strategy, is that it is measurable.

**Release plan decision:** The needs elicited in Section 4.1, including its documentation and decision material, is used as a basis for prioritization and followed by a decision of what needs to be included in the next release. We have not yet reached a state where we can present a full capability scale, the first two levels are presented, while the following two are proposals which might fit into the capability scale (these have not been observed in the industrial cases):

**Level A. Adhoc.** Needs are prioritized based on the "gut-feeling" [2] of product management.

**Level B. Tool Support.** A decision support tool, such as Focal Point, is used to aid in making the decision. The prioritization should make use of the product/company strategy.

**Metrics (not based on observed data).** Use metrics, e.g., production cost, maintenance costs, used product options, sold product volume, quality-in-use, and prediction of these variables, as support when making decisions. We have not yet defined a specific set of metrics, but to reduce the number of decisions based on "gut-feeling" [2] more objective data must be used.

**Optimizing (not based on observed data).** Once proper metrics are in place it might be possible to introduce optimization. For example, by adapting current "science" approaches [1,6] to consider these metrics when computing release plans. Further research is required to reach such a position.

### 4.3 Realize Needs

How to realize needs in development projects is not within the scope of this paper; refer to, e.g., CMMI [10] for a description of practices/key-areas. However, one related issue is given an organization with certain capabilities of its release planning process and a certain maturity for performing development projects, e.g., determined using CMMI, which of these two areas should be improved in order to have best effect?

The maturity and capabilities of performing development projects controls the *efficiency* with which needs can be implemented, while release planning is more focused on *effectiveness*, i.e., making sure that the correct features and quality improvements are released to customers. Consequently, it is not certain that a company with 100% efficiency in its development projects is the most successful one. This indicates there being a need for being at least as good at release planning as performing development projects.

## 5 Application of the Capability Model

In this section we evaluate the release planning key-areas presented in Section 4 for the companies in our multiple case study [4]. This evaluation is based on qualitative reasoning of the empirical data from our study, which is the same data used in developing the release planning capability model. Based on the results from the evaluation we also present suggestions for how to improve the release planning processes for the companies. We lack data for making explicit conclusions in some cases due to using a grounded theory approach in building our capability model, i.e., the model was constructed after collecting the data.

The results of our analysis is summarized in Table 5; case numbering is consistent with [4,2]. Below we comment on each key-area requiring further analysis to reach a conclusion concerning the capability level.

**Need Elicitation:** Case 1 fulfils level B since it has a formal path for product features, quality improvements, and cost-cut; as discussed in Example 1. Case 4 fulfils level C by having a formal path, for product features and quality needs, and by the stakeholders prioritizing the needs; as discussed in Example 2.

**Table 2.** Capability levels for each release planning key-area and company

| Key-Area | Need Elicitation | Need Documentation | Decision Material | Product Strategy | Release Plan Decision |
|---|---|---|---|---|---|
| Case 1 | Level B | Level A | Level C | Product strategy | Level A |
| Case 2 | Level B-D | Level A? | Level C | Product strategy | Level A? |
| Case 4 | Level C | Level B | Level D | Adhoc | Level A/B |
| Case 5 | Level A | Level B | Level A-B? | Adhoc | Level A |
| Case 6 | Level A | Level B | Level B | Adhoc | Level A |
| Case 7 | Level A | Level A-B | Level B | Release profile | Level B |

We have insufficient data for clearly determining the capability level for Case 2, but it is somewhere between level B-D on the capability scale. Development projects are rated using the product strategy, indicating part support for level D, but we lack data concerning the existance of a proper formal path (level B) for need elicitation and if stakeholders prioritize needs (level C).

For Case 5 our interviews cover release planning with a planning horizon of 5–10 years, and therefore we lack data for the more near time planning. Still, the data we have indicate level A.

Case 6 and Case 7 are rather similar, both having level A. This judgement is made since they do not have any formal path for needs from R&D. Though, there are formal paths from sales and marketing.

**Need Documentation:** Almost all companies in the study have some form of template for documenting proposed needs. For example, Case 5 and Case 6 document needs using up to three PowerPoint slides, with cost estimates, business impact, and a time plan. Case 4 uses a "one-pager" containing a slogan, business benefit, estimated cost, and impact on other parts of the system. Case 7 has an Excel template but seems to lack cost-estimate, but there may be other templates not covered during the interviews. Case 1 seems to use only a short-description, but which is refined in pre-studies and later stored as a change request in a database. For Case 2 we have no exact data.

**Decision Material:** Case 1 fulfils level C, as discussed in Example 4, and Case 4 fulfils level D, as also discussed in Example 4. Case 2 performs structured pre-studies and projects are also required to rate their impact on the 8 core values defined in their product strategy. The data we have from Case 6 and Case 7 indicate that they perform pre-studies, but the format for these pre-studies is not strictly defined (level B). We lack data for Case 5, but based on our impression from our interviews we suspect they are between level A–B.

**Product Strategy:** Case 1 and Case 2 have clearly defined product strategies as discussed in Section 4.2. Case 7 uses a release profile defining the top-level goals for the next release. Case 4, Case 5, and Case 6 did not seem to have a defined product strategy that had impact of how needs where prioritized, instead these companies based their decisions making "good business".

**Release Plan Decision:** The release plan decision is usually made in a group discussion, where typically stakeholders need to "lobby" for their own case. In case there is data support the proposed need, e.g., a customer survey, then such data often has strong impact beneficial. Case 1, Case 4, Case 5 seem to handle in the decision making in similar ways. Case 7 uses the tool Focal Point [9] as an aid in decision making. Case 6 has tried using Focal Point, but considers to be a bit awkward when comparing needs with very different costs.

### 5.1 Improvement Proposals

Here we discuss some possible ways in which the companies in our study can improve their release planning processes.

Case 4 has highest capability level, among the companies in the study, for need elicitation, decision material, and need documentation; see Table 5. We have data from other sources indicating that Case 4 also has highest CMMI level among the investigated companies. We have not looked further into this issue. Areas where Case 4 possibly can improve is by defining a more clear product strategy and by employing decision support tools to a greater extent.

Case 7 can improve their need elicitation by having a formal path for R&D and possibly other important stakeholders. Probably they will benefit of using R&D for performing structured pre-studies, which in turn should result in better time and cost-estimates for development.

Case 6 is in a similar position as Case 7; improvement areas being need elicitation and decision material. Possibly they can improve by using a release profile. Case 1 has a very large product volume compared to most of the other cases (see Table 1) and make use of many metrics and customer surveys to track their performance. Possibly they can improve their need documentation.

For Case 2 and Case 5 we lack data for making good improvement suggestions. Still, it should be noted that Case 2, compared to the other companies, seems to take more decisions on lower organizational levels.

## 6 The Problem into Context

The different process activities and key-areas described in Section 4 can be implemented in many different ways. For example, it is possible to use them in a staged/waterfall manner or in an iterative way. What suits an organization usually depends on the business context, and therefore it is not necessarily related to release planning capabilities. However, to provide a better understanding for how the key-areas can be combined we provide one example from our study.

**Example 5.** One basic understanding concerning release planning for Case 4 [4] is that *there will be changes* during a release project, e.g., there will always be needs which aren't thought of during the initial planning of a release and there will be changes to proposed needs. To cope with this they do not assign more than 50% of the release budget to the initial release plan, the remaining 50% is planned to be used for needs and changes during the release project.

During the initial phases of release planning there is usually 4 times as many needs as can fit within the budget of a release. In order to identify the needs that provide best return-of-investment/customer benefit they iteratively refine needs using pre-studies and feasibility studies. These investigations explore more needs than can fit within the budget allocated to a release. 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, as is illustrated in Fig. 6. Furthermore, the people capable of performing pre-/feasibility studies are often

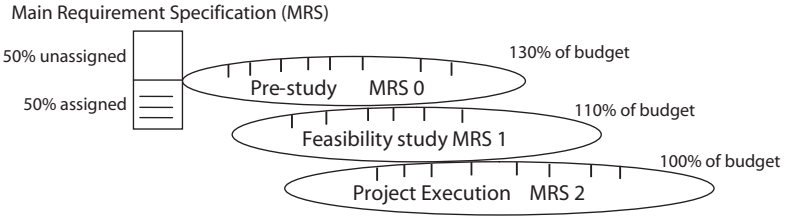


**Fig. 6.** Overview of iterative need refinement

a scarce resource, therefore these studies are performed throughout the release project; the vertical lines in Fig. 6 indicate start of study.

The of goals of Case 4's release planning process are:

**Efficiency.** Obtain efficient use of the resources available to the organization, including, e.g., R&D, local sales offices, and marketing.

**ROI.** The goal of basically all companies is to generate profit, therefore investments should be made such that return of investment is maximized.

**Flexibility.** For most organizations it is desirable to have a flexibility that allows late detected needs, e.g., new customer needs, to be included into the current release, thereby enabling a short time-to-market.

**Risk reduction.** In all product development, and within most organizations, there are different kinds of risk. For product development one such potential risk is underestimating cost and/or time for development.

## 7 Future Work

Here we discuss some possibilities for future work resulting from this paper. There are possibly other "key-areas" for release planning processes, which remain to be identified. Work needs to be performed to develop a maturity matrix for release planning, as exists for TPI [14]. Improve the release planning process capability model, such that it relies less on subjective judgements, and rather use more objective data, e.g., measured directly on the quality and costs associated with a product. Further validation and refinement of the model is also required.

## 8 Conclusion

Software release planning is an important activity for selecting the needs that best fulfil customer needs and at the same time provide a sound return-of-investment to the organization developing the software. Ultimately, release planning impacts how successful an organization can become.

In this paper we present a capability model for improving the release planning process of an organization. Using this model it is possible to:

1. Determine the capabilities of an organization's release planning process.
2. Identify areas for improvement.

The model is based on empirical data obtained from a multiple case study on release planning in industry involving 7 different companies. All companies in the study are producers of software intensive embedded systems, where the products have a relatively long life cycle; typically in the range of 10–20 years. In the paper we also apply our proposed capability model on the companies in our study, and illustrate how the capabilities of their release planning processes can be determined and discuss opportunities for improvement.

## Acknowledgements

This work was partially supported by the Knowledge Foundation (KKS) via the graduate school Intelligent Systems for Robotics, Automation, and Process Control (RAP), and partially supported by the Swedish Foundation for Strategic Research (SSF) via the strategic research centre PROGRESS.

## References

1. Ruhe, G., Saliu, O.: Art and Science of Software Release Planning. IEEE Software 22(6), 47–53 (2005)
2. Lindgren, M., Norström, C., Wall, A., Land, R.: Importance of Software Architecture during Release Planning. In: Proc. Working IEEE/IFIP Conference on Software Architecture (WICSA) 2008. IEEE Computer Society, Los Alamitos (2008)
3. Lindgren, M., Land, R., Norström, C., Wall, A.: Key Aspects of Software Release Planning in Industry. In: Proc. 19th Australian Software Engineering Conference, IEEE Computer Society, Los Alamitos (2008)
4. Lindgren, M.: Release Planning in Industry: Interview Data. Technical Report MDH-MRTC-219/2007-1-SE, Mälardalen Real-Time Research Centre (2007)
5. Jung, H.W.: Optimizing Value and Cost in Requirements Analysis. IEEE Software 15(4), 74–78 (1998)
6. Saliu, M.O., Ruhe, G.: Supporting Software Release Planning Decisions for Evolving Systems. In: 29th Annual IEEE/NASA Software Engineering Workshop, pp. 14–26. IEEE Computer Society, Los Alamitos (2005)
7. Carlshamre, P.: Release Planning in Market-Driven Software Product Development: Provoking an Understanding. Requirements Engineering 7(3) (2004)
8. Saliu, O., Ruhe, G.: Software release planning for evolving systems. Innovations in Systems and Software Engineering 1(2) (2005)
9. Karlsson, J., Ryan, K.: A Cost-Value Approach for Prioritizing Requirements. IEEE Software 14(5) (1997)
10. CMMI Product Team: CMMI for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008, Carnegie Mellon — Software Engineering Institute (2006)
11. Yin, R.K.: Case Study Research: Design and Methods (Applied Social Research Methods), 3rd edn. Sage Publications Inc., Thousand Oaks (2003)
12. Strauss, M., Corbin, J.M.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, 2nd edn. Sage Publications, Thousand Oaks (1998)

13. Gibson, D.L., Goldenson, D.R., Kost, K.: Performance Results of CMMI-Based Process Improvement. Technical Report CMU/SEI-2006-TR-004, Carnigie Mellon — Software Engineering Institute (2006)
14. Andersin, J.: TPI — a model for Test Process Improvement. Technical report, University of Helsinki, Department of Computer Science (2004)