STRATEGIC RELEASE PLANNING AND EVALUATION OF OPERATIONAL FEASIBILITY

Guenther Ruhe[1)] & Joseph Momoh[1)2)]

[1)] Laboratory for Software Engineering Decision Support
University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4
ruhe@ucalgary.ca
[2)] Trema Laboratories Inc.
Suite 200, 5970 Centre Street SE
Calgary, Alberta, Canada T2H 0C1
Joseph.momoh@trema.com

ABSTRACT

Strategic planning (or road-mapping) of software releases addresses the assignment of requirements to releases on a strategic level. Effort, finance and risk constraints are considered to determine strategic release plans. The goal is to find an optimal balance between competing stakeholder priorities and bottleneck resources. However, strategic planning has to be supplemented by more fine-grained operational planning as typically performed in project management.

The paper describes mechanisms by which to reduce the complexity of strategic and operational planning to a series of data and formulae that objectively represent input from all stakeholders and can easily reported, analyzed and manipulated. The capability provides improved planning and re-planning in a dynamic business environment, including the ability to validate strategic plans against operational limitations and revise as necessary.

For performing strategic planning, we present the research prototype ReleasePlanner^{TM}. Real-world experience in performing strategic planning using ReleasePlanner is reported from a case study at Trema Laboratories Inc.

Keywords*: Release Planning, Strategic Planning, Operational Planning, Decision Support, Stakeholder, Resource Planning, ReleasePlanner, Case Study, Tool Support.*

1. INTRODUCTION

The requirements engineering process is a decision-rich problem solving activity [Aurum & Wohlin 03]. One of the most prominent issues involved in incremental software development is to decide upon the most promising software release plans while taking into account diverse qualitative and quantitative project data. Such decisions have become even more complicated in the presence of stakeholders having different relative importance and different preferences in the presence of constraints about sequencing and coupling of requirements in various increments, and taking into account resource allocation for implementation of requirements. The goal is to account for all these factors and to come up with suggestions for the most satisfactory release plans.

Release planning can be done in very different ways [Bagnall et al. 01], [Penny 02], [Greer & Ruhe03], Karlsson et al. 04], [Ruhe & Ngo-The 04]. Distinguishing parameters in approaching release planning are: Considered time-horizon of planning, objects of planning, degree of formality in the planning procedure, inclusion of additional constraints as part of the planning process, degree of stakeholders involvement, and finally the actual solution technique that is applied to obtain release plans.

In its easiest way, release planning is done in an ad hoc manner, eventually supported by spreadsheet computation. Extreme programming [Beck 01] conducts release planning by performing planning games. The customer presents the desired requirements to the programmers. With cost estimates and some knowledge of the importance of the requirements, the customer lays out a plan for the project. This approach has obvious limitations in terms of its scalability, its effectiveness and its efficiency. It is unable to address competing stakeholder interests and to address any type of resource constraint.

In this paper, we will consider a more sophisticated approach that is based on formal problem description and its solution using specialized optimization procedures. This is an extension of an existing method EVOLVE*, as presented in [Ruhe & Ngo-The 04]. We will call the resulting method EVOLVE^{ext}. EVOLVE^{ext} is based on the interplay between strategic and operational planning. The difference is in terms of the planning time horizon, the objects to be planned and the granularity of planning. EVOLVE^{ext} also provides a more flexible system of stakeholder voting. The new scheme encompasses groups of requirements and allows flexible assignment of stakeholders to requirements or groups of requirements.

EVOLVE^{ext} is based on the paradigm of software engineering decision support [Ruhe 03]. It combines computational and human intelligence to approach inherent difficulties of the "wicked" planning problem, including:

- Requirements are not well specified and understood,
- change of requirements and other problem parameters,
- stakeholder opinions that compete with each other,
- problem size and complexity (considering all the resource and technological constrains), and
- uncertainty and incompleteness of data related to effort, resources, and inherent risks.

Following this introduction, Section 2 examines strategic release planning. A formal problem description is derived including main objectives and constraints. Section 3 discusses solution approach EVOLVE* and its implementation in the intelligent decision support tool ReleasePlanner^{TM}. Section 4 addresses the question to whether to evaluate the operational feasibility of the proposed strategic plan to realize the predetermined requirements for the immediately next release. This includes planning of tasks and their required resources for the individual increments within that release. Section 5 studies the strategic release planning at Trema Laboratories Inc. and the required extensions of ReleasePlanner to fully cover the interplay between strategic and operational planning are studied in Section 5. Finally, Section 6 summarizes the paper and presents conclusions for future work.

2. STRATEGIC PLANNING OF SOFTWARE RELEASES

We consider a set of requirements $\Re = \{r_1,\ldots,r_n\}$. Whenever applicable without ambiguity, $\{1,2,\ldots,n\}$ is used instead. The problem is to assign the most attractive requirements to a set of J subsequent releases. Based on that, we define Boolean decision variables $x(i,j) \in \{0,1\}$ with

(1) $x(i,j) = 1$ if i is assigned to release j and $x(i,j) = 0$ otherwise.

Each requirement i is assumed to be assigned at most once expressed as

(2) $\sum_j x(i,j) \leq 1$

The set of all variables x(i,j) with $i = 1,\ldots,n$ and $j = 1,\ldots,J$ is represented by an array x. We consider two types of dependencies between requirements: coupling and precedence relationships. Two requirements i1, i2 from $\Re$ are said to be coupled if they have to be assigned to the same release. The set of all pairs of coupled requirements is denoted by C. Precedence between requirements i1 and i2 from $\Re$ means that i1 is not allowed to be released later than i2. The set of all pairs of requirements in precedence relationship is denoted by P. Expressed in terms of variables x(i.j), coupling and precedence relationship is formulated as (3) and (4), respectively.

(3) $x(i1,j) = x(i2,j)$ for all fixed releases j and all pairs $i1,i2 \in C$

(4) $\sum_j (J+1-j)\, x(i1,j) \geq \sum_j (J+1-j)\, x(i2,j)$ for all pairs $i1,i2 \in P$.

Formula (4) ensures that requirement i1 is never released later than requirement i2. More general cases of requirements dependencies are studied in [Carlshamre et al. 01]. To realize requirement i we also consider the estimate of the overall realization effort. At this stage, effort subsumes all human effort performed by the different roles in the development process. The amount of effort necessary to realize requirement i is denoted by effort(i). For each increment j to be released, there is an effort capacity bound Cap(j). The respective constraints are

(5) $\sum_i \mathrm{effort}(i)\, x(i,j) \leq \mathrm{Cap}(j)$ for all releases $j = 1,\ldots,J$

A release plan $x \in X$ is called a feasible solutions if it satisfies constraints (2) – (5). The set of all feasible plans is denoted by X.

We assume q different stakeholders abbreviated by $S_1,\ldots,S_q$. Each stakeholder S_p is assigned a relative importance $\lambda_p \in (0,1)$. The relative importance of all involved stakeholders is typically assigned by the project or product manager. If it is difficult to actually determine these weights, pair-wise comparison using the analytic hierarchy process [Saaty 80] can be used as a support. We assume that stakeholder weights are normalized to one.

The increasing focus on value creation as a result of software development implies the question of how the different features or requirements impact the value of the software. Typically, there are different and conflicting priorities between different stakeholders or groups of stakeholders. To determine the most attractive requirements and product portfolio all priorities have to be evaluated to the best knowledge available.

There are different ways to evaluate the priority of a requirement from a stakeholder perspective. For our purposes, we consider two dimensions of priority, a value-based and an urgency-based prioritization. Value addresses the assumed impact on the value of the final product, and is considered to be independent of time. Urgency addresses the time-to-market aspect to reflect market needs and competitor analysis information.

Intuitively, what we are trying to achieve is to assign requirements of high value and high urgency to the first release.

The judgment of stakeholder p with regard to requirement i is denoted by value(i,p) where value(i,p) represents the perceived value of the requirement i for stakeholder p. We are using a nine-point scale of measurement which could be replaced by another (e.g., five-point) scale depending on the degree of knowledge about the subject. As a guideline, we define

- value(i,p) = 1 requirement i is of very low value
- value(i,p) = 3 requirement i is of low value
- value(i,p) = 5 requirement i is of moderate value
- value(i,p) = 7 requirement i is of high value
- value(i,p) = 9 requirement i is of extremely high value

Release planning is based on an evolving set of requirements. The number of releases to be considered in advance may vary from case to case. We use "option k" for the assignment of a requirement to release k. Because of the high degree of requirement volatility, it does not make sense to plan too many releases in advance. For this paper (and without loss of generality), we will only consider two releases in advance. This is considered to be a good compromise of looking into the future while accepting the uncertainty and volatility of the problem. Consequently, as a result of release planning, each requirement is assigned to exactly one of three possible cases:

- next release (option 1),
- next but one release (option 2), or
- postponed or not (yet) considered for implementation (option 3).

Urgency addresses the degree of satisfaction with these three possibilities from the individual stakeholder perspective. Consideration for urgency can be motivated by the time-to-market aspect of certain product requirements. For each feature i, the stakeholder p is asked to represent his or her satisfaction with the situation that feature i is assigned to option k (k=1,2,3). His or her judgment is expressed by sat(i.p,k) $\in$ {1,…,9} with

(6) Sat (i.p) = (sat(i.p,1) , sat(i.p,2), sat(i.p,3)) for all i and all stakeholder S_p

(7) $\Sigma_{k=1,2,3}$ sat(i.p,k) = 9 for all requirements i and all stakeholder $S_{p.}$

The higher the number of votes the stakeholder assigns to k, the more satisfied (s)he would be if the requirement is put in the respective option. For example, a vote of (9,0,0) would express extreme urgency for the associated requirement. Similarly, (3,6,0) would express a preference to have the requirement preferably in the next plus one release. Voting (3,3,3) indicates no preference in terms of when the requirements should be released.

Typically, not necessarily all stakeholders are necessarily able or comfortable to give their evaluation related to all requirements. A stakeholder p is called active with respect to requirement i, if (s) has provided the evaluation, and is called passive otherwise. The set of active stakeholders with respect to requirement i is denoted by P(i). For determining the weighted average priority WAP(i,k), only active stakeholder priorities are considered.

Further flexibility in the strategic planning process is introduced by the possibility of varying the importance of stakeholder weights λ_p and of weighting the importance ξ_1 and ξ_2 of options 1 and 2, respectively. This results in the objective function F(x) defined as

(8) $F(x,\lambda,\xi) = \xi_1 \Sigma_{x(i)=1} WAP(i,1) + \xi_2 \Sigma_{x(i)=2} WAP(i,2)$ with

(9) $WAP(i,k) := [\Sigma_{p\in P(i)} \lambda_p \cdot value(i.p) \cdot sat(i.p,k)] / [\Sigma_{p\in P(i)} \lambda_p]$ for all requirements i and k = 1 and 2.

For a fixed vector λ of stakeholder weights and a fixed vector ξ of importance, the strategic release-planning problem RP(λ,ξ) becomes

(10) Maximize $F(x, \lambda,\xi)$ subject to $x \in X$.

3. STRATEGIC PLANNING OF RELEASES USING RELEASEPLANNER

The solution of (10) is implemented by the tool suite ReleasePlanner^{TM} (see www.releaseplanner.com). It is based on a solution approach called EVOLVE* [Ruhe & Ngo-The 04] that combines the computational strength of special purpose optimization algorithms with the flexibility of an iterative solution method. At all iterations, an integer optimization algorithm is applied to determine the most promising solutions of constrained release planning. The three main phases of the incremental and evolutionary approach are:

Phase 1 – Modeling

Modeling focuses on the formal description of the dynamic real world to make it suitable for computational intelligence based solution techniques by defining all of the related decision variables, constraints, stakeholder evaluations, etc.

Phase 2 – Exploration

Computational techniques, such as the ones built into the tool ReleasePlanner^{TM}, are used to explore the solution space with many alternatives.

Phase 3 – Consolidation

In the consolidation phase decision makers investigate the solution alternatives. This leads to the problem with reduced size and complexity for the next iteration. The evolutionary character of the approach is illustrated in Figure 1.

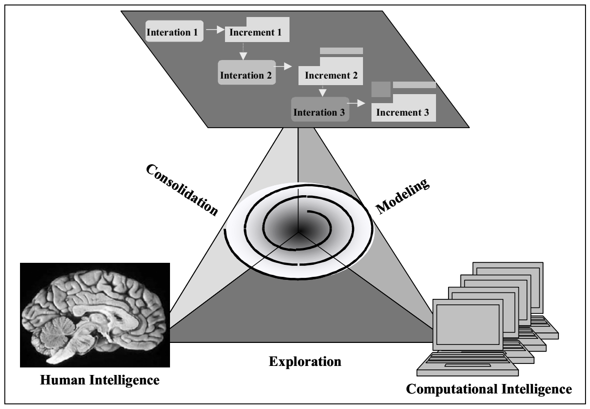


Figure 1. EVOLVE* as a mediator between real problem world, computational intelligence based tools, and human intelligence.

The main requirements of the intelligent decision support tool ReleasePlanner^{TM} are:

- Greater input and interaction with even remote stakeholders (sales representatives, shareholder, developer, different classes of user) because it is web-based. This interaction results in a much better chance to achieve high customer satisfaction with the products to be developed
- Ease of customization for the different types of users and the different usage scenarios. After defining the usage mode, only respective functionality is activated. This increases user-friendliness.
- Compatibility with commercial tools addressing requirements management (as a prerequisite for using release planning) and project management (for actually planning and controlling the performance of the plan).
- Computational strength of its core algorithms that are based on innovative ideas (genetic algorithm) and their novel implementation. This results in very effective procedures to generate near-optimal solutions for large-scale problems of high complexity.
- Modelling support is offered. As validated models are a mandatory prerequisite for meaningful results, different objectives (value maximization, optimal customer satisfaction), different schema of evaluation, and different types of constraints (effort, resource, precedence, finance, risk) can be included.
- Flexibility: ReleasePlanner^{TM} is able to work with different levels of abstraction. The lowest level is that of individual requirements, the highest level is (groups of) requirements or even applications.
- Process-driven usage guidelines as well as experience and knowledge support: There is a process framework for how to conduct planning. Each task is the object of adding or retrieving related knowledge and experience (Knowledge management component).
- Portability. The tool can be used for different platforms and operating systems.

ReleasePlanner was successfully used already in a series of industrial environments (see [Amandeep et al. 04]), [Dantsigner 04], [Momoh 04]. We will report about a case study experience in Section 5.

4. OPERATIONAL FEASIBILITY OF STRATEGIC PLANS

Strategic planning of software releases addresses stakeholder priorities and different types of constraints. However, strategic planning has to be supplemented by more fine-grained operational planning as typically performed in project management. Operational planning as a refinement of strategic planning is performed for the immediately next release under consideration. Within this release, we introduce sub-releases also called increments. The subject of planning is no longer requirements. Instead, we consider the individual tasks for realizing these requirements.

Resources are an essential asset for planning increments and releases. Consideration of just 'effort' as it is done on the strategic level is a rough approximation and does not consider individual types of resources. For each increment, we consider all necessary resources for performing the tasks assigned to this increment. Definition of the actual resource types is done by the project manager. Example resource types are:

- Requirements specification
- Analysis and design
- Implementation
- Testing
- Money
- Risk

To formalize the problem, we consider a set of requirements $\Re(1) = \{1,\ldots,n1\} \subset \Re$ that is assigned to the immediately next release as a result of strategic planning. The release under consideration is subdivided into H increments. To validate the operational feasibility of the proposed strategic plan, we consider all the individual resources necessary to perform the different tasks within each increment. Let's assume that K different resource types are considered to realize the requirements of $\Re(1)$. In correspondence to that we study K different tasks defined for realizing requirement i.

(11) $T(i) = \{task(i,1),\ldots,task(i,K)\}$

In other words, we model tasks as being in one-to-one correspondence to the different resource types. Each task task(i,k) reflects the portion of resources of type $k \in \{1,\ldots,K\}$ necessary for realizing requirement i. task(i,k) is assumed to consume the amount effort(i,k) of resource type k.

We define the binary decision variable y(i,k,h) as defined in (12). The vector of all binary variables is abbreviated by y.

(12) $y(i,k,h) = 1$ if task(i,k) is assigned to increment h (and $y(i,k,h) = 0$ otherwise).

In each increment h, we expect an available amount of Cap(k,h) of resource type k. That means, we have a set of capacity constraints formulated as

(13) $\sum_{i=1,\ldots,n1} \text{effort}(i,k)\, y(i,k,h) \leq \text{Cap}(k,h)$ for all increments h and all resource types k.

In the same way as introduced for strategic level between requirements, we consider precedence and coupling constraints between tasks for realizing the requirements.

The set of all pairs of coupled tasks within release 1 (the immediately following) is denoted by C(1). Coupling between tasks a1 and a2 means that task a1 has to be assigned to the same increment as task a2. Precedence between tasks a1 and a2 means that a1 is not allowed to be released later than a2. The set of all pairs of requirements in precedence relationship is denoted by P(1). Expressed in terms of variables y(i,k,h), coupling and precedence relationships are formulated in analogy to(3) and (4), respectively. Both types of constraints are defined for tasks of the same requirement or for tasks of the same resource type.

We will abbreviate the set of all assignments y(i,k,h) of tasks to individual increments satisfying all these constraints by Y(Cap). This indicates the dependency of Y from the capacity vector Cap.

Using the notation introduced above, we are able to formulate three types of problems related to the question of validating the feasibility of the proposed strategic release plan. More specifically, we consider the feasibility of the strategic plan with respect to the tasks and available resources within the immediately following next release.

Feasibility Problem 1: Check for Feasibility

Does there exist an assignment y^* such that $y^* \in Y(\text{Cap})$ fulfills all constraints.

In that case, we would be sure that the strategic plan is feasible in the sense that the more fine-grained consideration of tasks and resources does not create any additional conflicts for the immediately following release. We call the strategic plan operationally feasible.

However, if this is not the case, it might be the case that a proposed strategic plan is not operationally feasible. In that case two possible derived problems are of importance. Both of them are addressing the question of how to make the proposed plan feasible with a smooth adjustment of either the capacities available or by reducing the functionality of the first release. Problem 2 is to find the cheapest extensions of resources necessary to allow feasibility. Problem 3 is the value-optimal reduction of set $\Re(1)$.

Feasibility Problem 2: Optimal Resource Extension

Assuming the extension of resource type k for increment h by one unit costs ext_cost(h,k), we are looking for the cheapest extension ΔCap of available capacities to allow feasibility of the proposed strategic plan.

(14) $\min\{ \sum_h \sum_k \text{ext\_cost}(h,k)\, \Delta\text{Cap}(h,k)$
subject to
$y \in Y(\text{Cap} + \Delta\text{Cap}\}$

(14) is a specialized integer linear programming problem that can be solved in seconds of time for mid-size problems (up to thousand objects to be planned).

Feasibility Problem 3: Optimal Reduction of Functionality

Assuming that for each requirement $i \in \Re(1)$ we have determined the weighted average value(i) of stakeholder values. The problem becomes to find the optimal subset $\Re^*$ of $\Re(1)$ such that there is a feasible plan y^* related $\Re^*$ and

(15) $\sum_{i \in \Re^*}$ value(i) is maximum.

Problem (15) is a subset selection problem, where the solution of each sub-problem is NP-Complete itself. To approach this high problem complexity, fast heuristics will be applied.

5. CASE STUDY AT TREMA LABORATORIES INC.

5.1 Background

Trema Group is a provider of strategic software solutions for the financial industry. The software development focus is on providing a fully integrated cash and treasury management product suite designed to support front to back office treasury operations, as well as specific applications for cash management and accounting. Requirements are added to the product incrementally. So, planning for future releases becomes extremely important for the business.

The requirements for the next releases of the suite of products come from different sources such as contractual commitments, market positioning, technology opportunities, sales analysis, customer needs and existing functionality enhancements. The planning for the next releases is also tied to the strategic priorities which could change from release to release. The strategic priorities include:

- Marketability,
- quality,
- performance,
- scalability,
- customer functionality,
- technology,
- stability, and
- contractual commitments

To better understand existing challenges in planning of releases, a series of questionnaire based interviews were performed. The 23 participants were project managers (8 participants), product manager (5), development manager (6), and functional architects (4).

The questionnaire contained ten key challenges facing the industry and participants were asked to rank them in order of magnitude of the problem. Opportunity was provided for participants to include other challenges. The sample for the survey is small but care was taken in drawing conclusions from the results through follow-up interviews with respondents to explore their understanding of some of the terms and issues in the survey in more depth.

A distribution of key factors is shown in Figure 2. The category 'Others' consists of all the other minor factors identified during the survey as shown in the table above. A priority ranking of the key challenges is given in Table 1.

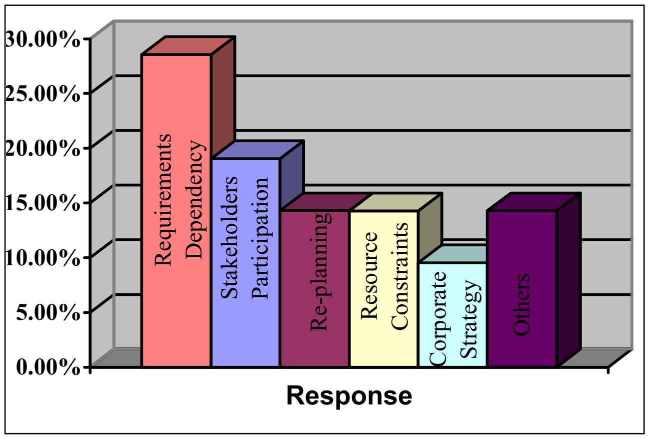


Figure 2: Distribution of key factors.

Challenging Factors	**% of Response**
No easy way to get good overview of dependencies of requirements	26%
Stakeholders participation difficult especially in a distributed environment	22%
Difficulty in re-planning due to late breaking requirements or changing stakeholder priorities	17%
Managing resource constraints and their effect on a release is tedious	13%
Difficulty of translating corporate strategy to the product through the plan	9%
Assigning the right resources and skills to tasks to be performed in each requirement	4%
Visibility of the required skills and roles to implement each requirement.	4%
Adequate interaction among the stakeholders during the selection of requirements for a release	4%
No easy way to obtain stakeholder votes for requirements	0%
No easy way to generate reports showing the overview of stakeholder voting for each requirement	0%

Table 1. Priority ranking of release planning challenging factors at Trema.

5.2 Baseline

Current deficits in release planning and requirements prioritization are reported in [Lethola et al. 04] and [Ruhe & Ngo-The 04]. As with most software organizations, the planning process is currently mostly ad-hoc with use of several spreadsheets through endless meetings with stakeholders in several different locations to arrive at an acceptable content of the releases.

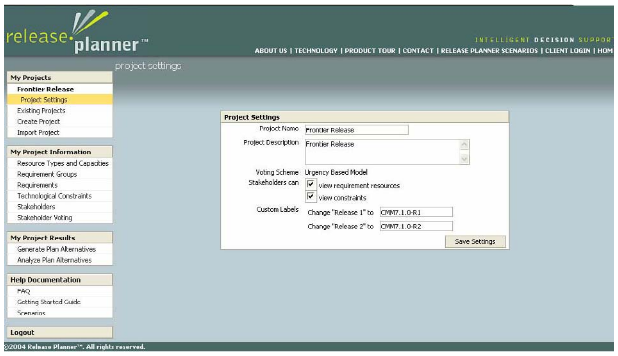
The planning process is performed in two stages. The first stage is the development of the longer term roadmap also known as the strategic release plan. This usually spans three or more releases over one or more years. The roadmap is used as a strategy and planning document for the organization. The ultimate goal is to provide an optimal longer term release plan that will help fulfill the product's mission based on all the requirements that have been received. The roadmap is generally taken as a living document which evolves over time and becomes more concrete and offer more details as one gets closer to each of the releases on the roadmap. This strategic plan is also revised based on the lessons learnt from each of the releases as they are scheduled and executed. From the roadmap, the focus is usually on the near-term release and on how to achieve successful scheduling and execution of that release.

The inventory of requirements from which the roadmap or strategic release plan is developed from will usually contain over 500 items that are requested to be in future releases. Some of these items are dependent on other items in the inventory. There are five or more stakeholders with varying interests in each of the requirements participating in prioritizing these items. The stakeholders are located in different places around the world. The process of ranking and agreeing the content of the releases takes several days using spreadsheets to calculate and collate the results. Some of the problems with the ad-hoc approach include:

- There is usually no way to know all the alternative solutions considering all the constraints and dependencies between the requirements.
- Getting all the stakeholders together for meetings and go through all the requirements takes time.
- Every stakeholder has a different opinion of what is important and the process of negotiation among the stakeholders can take too long.
- Requirements are continuously changing and evolving. The impact of the changes on the strategic plan is not immediately obvious.

5.3 Improvement

Trema Laboratories Inc. found through the initial trial that ReleasePlanner^{TM} provided very good ways to manage the top four challenges in the release planning process identified in the survey results above. The time to generate release plans and to alternate release plans because of changing requirements was drastically reduced to clicks of buttons from a process that took several days. Figures 3 to 5 below show some of the screens in the process of defining the above roadmap using ReleasePlanner$^{TM:}$



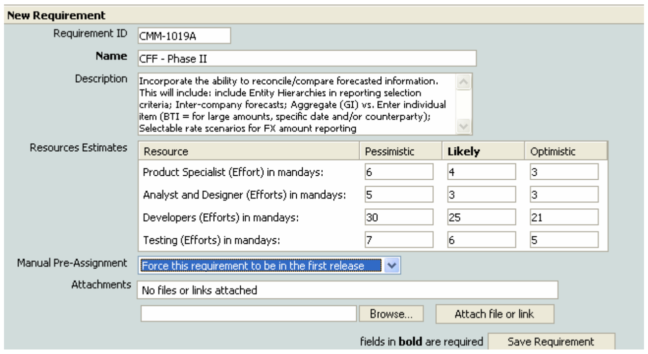
Project Settings

Project Name	Frontier Release
Project Description	Frontier Release
Voting Scheme	Urgency Based Model
Stakeholders can	✓ view requirement resources
	✓ view constraints
Custom Labels	Change "Release 1" to CMM7.1.0-R1
	Change "Release 2" to CMM7.1.0-R2

Figure 3: Defining the project settings.

The efforts for the release is measured in person-days and the totals available to be allocated to each of the releases have been determined based on the availability of the team members as shown in Figure 4 below.

The requirements are ranked based on the urgency model.



New Requirement

Requirement ID	CMM-1019A
Name	CFF - Phase II
Description	Incorporate the ability to reconcile/compare forecasted information. This will include: include Entity Hierarchies in reporting selection criteria; Inter-company forecasts; Aggregate (GI) vs. Enter individual item (BTI = for large amounts, specific date and/or counterparty); Selectable rate scenarios for FX amount reporting

Resources Estimates

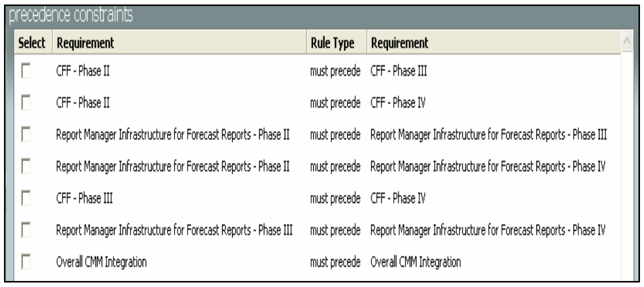
Resource	Pessimistic	**Likely**	Optimistic
Product Specialist (Effort) in mandays:	6	4	3
Analyst and Designer (Efforts) in mandays:	5	3	3
Developers (Efforts) in mandays:	30	25	21
Testing (Efforts) in mandays:	7	6	5

Manual Pre-Assignment	Force this requirement to be in the first release
Attachments	No files or links attached

fields in **bold** are required

Figure 4: Capturing of the requirements.

The roadmap had a total of 49 requirements to be released over a period of one year. Certain requirements such as the building of the framework for future requirements are required to be in the early or first release in the roadmap. Some of the requirements must precede other requirements in the roadmap. In addition, we have six coupling constraints



precedence constraints

Select	Requirement	Rule Type	Requirement
☐	CFF - Phase II	must precede	CFF - Phase III
☐	CFF - Phase II	must precede	CFF - Phase IV
☐	Report Manager Infrastructure for Forecast Reports - Phase II	must precede	Report Manager Infrastructure for Forecast Reports - Phase III
☐	Report Manager Infrastructure for Forecast Reports - Phase II	must precede	Report Manager Infrastructure for Forecast Reports - Phase IV
☐	CFF - Phase III	must precede	CFF - Phase IV
☐	Report Manager Infrastructure for Forecast Reports - Phase III	must precede	Report Manager Infrastructure for Forecast Reports - Phase IV
☐	Overall CMM Integration	must precede	Overall CMM Integration

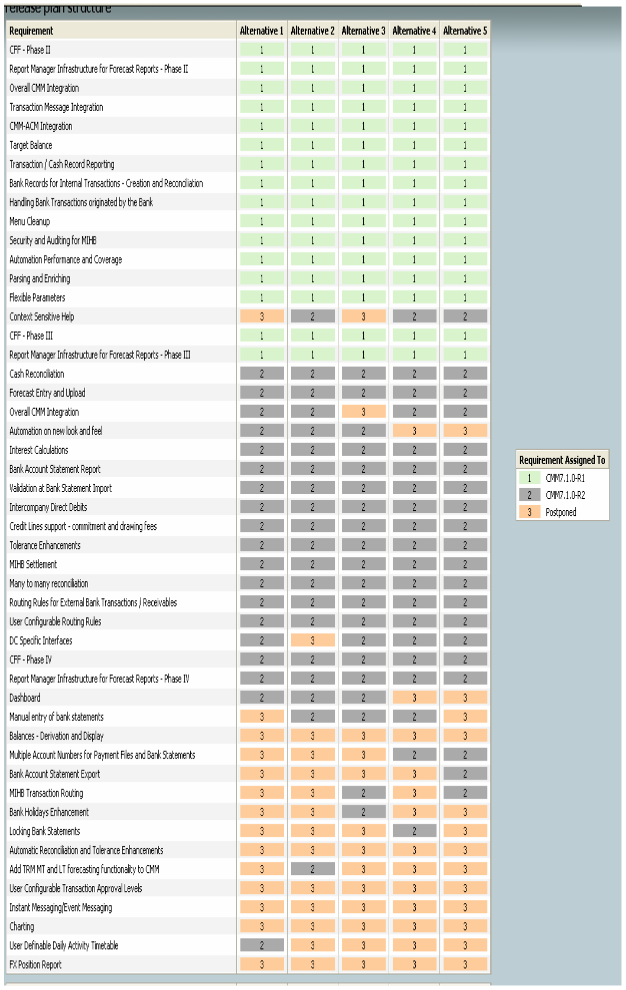
Figure 5: Defining the precedence constraints.

There are six stakeholders involved in voting on the roadmap. At the end of the voting exercise by all stakeholders in the project, release plan alternatives were generated by ReleasePlanner^{TM} as shown in Figure 6 below.

At the end of the trial of using the intelligent decision support for both strategic and operational release planning at Trema Laboratories Inc., the general agreement is that ReleasePlanner^{TM} provides benefits to the organization. Some of these benefits are tangible and some are intangible. The intangible benefits are difficult to realistically evaluate but the good thing in this study is that most of the intangible benefits are generally positive. In summary the benefits include:

- Improvement in pre-planning: Pulling together the roadmap and generating the release plans takes 60% of a full time job for a product manager and a release manager. This effort spent for this activity depends on the size of the organization and the complexity and size of the product. Bigger organizations tend to expend more effort expended for this activity. Keeping an overall view of all the requirements and the dependencies between the requirements makes this extremely difficult especially when dealing with hundreds of requirements. Also, gathering the efforts to implement each requirement and establishing the capacity versus the required effort for a release is a tedious process. ReleasePlanner^{TM} provides a quick overview of these requirements and the generation of alternative release plan becomes trivial through the push of a button on the keyboard. During the case study we experienced a reduction of the time spent in pre-planning to less than 20% compared to the 60% using an ad hoc approach.

- Improved and quick re-planning capabilities: The re-planning process takes even further time and efforts. Whenever there is a change in a requirement, in the estimated size of a requirement or a change in stakeholder priority there is always a need to revise the plan to accommodate the change. This process consumes about 30% of a release manager's job including the time of other stakeholders involved in the release. It is estimated that the other stakeholders (6 in this case study) spend an average of 3 hours a week for re-planning during the release. The findings in the case study indicate that a saving of 75% was achieved through the use of ReleasePlanner^{TM}.



release plan structure

Requirement	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5
CFF - Phase II	1	1	1	1	1
Report Manager Infrastructure for Forecast Reports - Phase II	1	1	1	1	1
Overall CMM Integration	1	1	1	1	1
Transaction Message Integration	1	1	1	1	1
CMM-ACM Integration	1	1	1	1	1
Target Balance	1	1	1	1	1
Transaction / Cash Record Reporting	1	1	1	1	1
Bank Records for Internal Transactions - Creation and Reconciliation	1	1	1	1	1
Handling Bank Transactions originated by the Bank	1	1	1	1	1
Menu Cleanup	1	1	1	1	1
Security and Auditing for MIHB	1	1	1	1	1
Automation Performance and Coverage	1	1	1	1	1
Parsing and Enriching	1	1	1	1	1
Flexible Parameters	1	1	1	1	1
Context Sensitive Help	3	2	3	2	2
CFF - Phase III	1	1	1	1	1
Report Manager Infrastructure for Forecast Reports - Phase III	1	1	1	1	1
Cash Reconciliation	2	2	2	2	2
Forecast Entry and Upload	2	2	2	2	2
Overall CMM Integration	2	2	3	2	2
Automation on new look and feel	2	2	2	3	3
Interest Calculations	2	2	2	2	2
Bank Account Statement Report	2	2	2	2	2
Validation at Bank Statement Import	2	2	2	2	2
Intercompany Direct Debits	2	2	2	2	2
Credit Lines support - commitment and drawing fees	2	2	2	2	2
Tolerance Enhancements	2	2	2	2	2
MIHB Settlement	2	2	2	2	2
Many to many reconciliation	2	2	2	2	2
Routing Rules for External Bank Transactions / Receivables	2	2	2	2	2
User Configurable Routing Rules	2	2	2	2	2
DC Specific Interfaces	2	3	2	2	2
CFF - Phase IV	2	2	2	2	2
Report Manager Infrastructure for Forecast Reports - Phase IV	2	2	2	2	2
Dashboard	2	2	2	3	3
Manual entry of bank statements	3	2	2	2	3
Balances - Derivation and Display	3	3	3	3	3
Multiple Account Numbers for Payment Files and Bank Statements	3	3	3	2	2
Bank Account Statement Export	3	3	3	3	2
MIHB Transaction Routing	3	3	2	3	2
Bank Holidays Enhancement	3	3	2	3	3
Locking Bank Statements	3	3	3	2	3
Automatic Reconciliation and Tolerance Enhancements	3	3	3	3	3
Add TRM MT and LT forecasting functionality to CMM	3	2	3	3	3
User Configurable Transaction Approval Levels	3	3	3	3	3
Instant Messaging/Event Messaging	3	3	3	3	3
Charting	3	3	3	3	3
User Definable Daily Activity Timetable	2	3	3	3	3
FX Position Report	3	3	3	3	3

Requirement Assigned To	
1	CMM7.1.0-R1
2	CMM7.1.0-R2
3	Postponed

Figure 6: Structure of qualified release plan alternatives generated by ReleasePlanner^{TM}. Each column represents one alternative. Final selection is done by the human expert.

- Increased participation of stakeholders in the planning process: Through the ad hoc approach, stakeholders will usually need to meet in a location to discuss, prioritize and finalize the content of the release. In a distributed organizations where the stakeholders are based in different locations, this becomes very difficult and expensive considering travel costs. With ReleasePlanner^{TM} stakeholders are able to participate from wherever they are in the prioritization process for as long as they have access to the internet. This saves travel cost and increases the level of participation. The added cost of not participating early in the process is in re-planning.
- Improved stakeholder satisfaction through active participation: It is established through the case study that stakeholders feel a sense of ownership of the content of the release because of their active involvement in the requirement selection and prioritization. This leads to stakeholders having better satisfaction with the release.
- Provides open, collaborative atmosphere among stakeholders: Stakeholders are able to interact directly with the system. This also leads to a higher level of satisfaction in the process.
- Provides ease of setting priorities and decision making: Priority setting by stakeholders is improved and allows for quick decision making by management on the content of the release. There is a more focused approach to delivering the requirements in the release by the project and development teams.
- Improves project planning, product quality and quick product wins thorough the incremental requirement selection and focus on the optimal selection of requirements.
- Improves time to market of required features: Since the appropriate selection of requirements are delivered in the release, the product gets to market with these required features. This potentially leads to increased market share through customer satisfaction with the product. During the course of the case study, revenue increased by $1,750,000.00. All of this increase cannot be attributed to the use of the new release planning system but attributing 1% of this increase to the system was deemed acceptable.

5.4 Evaluation of Operational Feasibility

Once the release plan has been generated and agreed to by the stakeholders as the optimal requirement selection, the second stage in the planning process is to ensure that the next immediate release can be delivered based on the available resources, skills and constraints. Moving from strategic to operational planning of a release is very important for long term success of the organization.

At this stage, the key roles, skills and tasks required to implement each requirement are determined and estimated at a granular level such that the scheduling of the tasks can be achieved in on a two week increment. The key roles and tasks for each requirement are described as follows:

- The product specialist acts as the liaison between the business people and the software developer. This role focuses on what the end users need to be more effective at their jobs and not particularly concerned with the technical details. The product specialist is responsible for translating the business requirements in to a functional specification for the developers to work from.
- The analyst or designer has the task of reviewing existing product state and translating the detailed requirements from the functional specification into a design that can be implemented by the software developer.
- The software developer writes the code that would run on the selected infrastructure while ensuring that the new codes integrate with existing functionalities in the product. The developer provides some level of unit testing and automation of the test cases.
- The tester comes up with all the ways that a user is expected to use the requirement and documents them in test cases.

The size of efforts required for each task varies from requirement to requirement. Typically, the process of scheduling and sequencing of requirements in a release is a manual effort using spreadsheets and later feeding the results into a project management tool such as MS Project. Most of the tough decisions are made manually and ad-hoc before loading the data into the project management tool for tracking purposes. In a manual decision making process, it may be difficult to consider all the constraints and there is usually not enough time and resources to generate different scenarios. This could result in a potentially good sequence being ignored.

Every role addresses a different aspect of the requirement as a whole. Since each requirement will require different roles as previously described, the resources available to the release must take into consideration the different types of tasks to be executed for each of the requirements and the resources necessary for realizing them. task(i,k) is assumed to consume the amount of effort(i,k) of resource type k (Product specialist, analyst and designer, software developer, and Tester) for each increment h.

6. SUMMARY AND CONCLUSIONS

Determining the most appropriate requirements is one of the hardest and one of the most crucial problems of software development [Lethola et al. 04]. Competing objectives have to be balanced [Davis 03]. More than half of failing software projects can be directly attributed to flawed requirements [Standish Group 02].

We are addressing strategic planning of requirements in terms of assigning them to proper releases. Strategic planning on that level is crucial for later more detailed planning and successful project performance [Anton 03]. Strategic software release planning is characterized by the fact that the quantity and quality of information available is typically low; that the processes and available decision parameters are dynamically changing and that a number of conflicting stakeholder interests with different objectives and constraints must be balanced and optimized.

The main contributions of the paper are three-fold: (i) enlargement of the flexibility in stakeholder voting as performed in the predecessor approach EVOLVE*; (ii) extension of the capabilities of strategic release planning by evaluating the operational feasibility of proposed release plans, and (iii) real-world evaluation of the intelligent decision support tool ReleasePlanner^{TM}. In a more fine-grained planning of the increments of the immediately following release we have studied three different options for (ii): (ii.1) The plan can be realized with the resource capacities available, (ii.2) The plan is not feasible, and we are looking for the most cost-effective extension of capacities to make it realizable, or (ii.3) The plan is not feasible and we are looking for the value-optimal reduction of the release functionality.

We have reported an encouraging experience of using the ReleasePlanner tool at Trema Laboratories. This adds to an already existing experience as described in [Amandeep et al. 04]. Future work will be devoted to implementation, to the conceptual ideas for evaluating operational feasibility into the tool suite and to evaluate the overall performance of the approach EVOLVE^{ext}. To adress the area of matching methodology to problem domain as stated by [Glass 04], we believe that EVOLVE^{ext} is more applicable, as the problem becomes larger and more complex. In addition to that, we foresee more benefit for its application, as processes of an organization mature.

Acknowledgements

The authors would like to thank the Alberta Informatics Circle of Research Excellence (iCORE) for its financial support of this research. Also, thanks to Trema Laboratories, Inc. for providing access to data and other

information for the research. The reviewer's comments helped to make the paper more readable.

References

[Amandeep et al. 04]
Amandeep, A., Ruhe, G., Stanford, M., "Intelligent Support for Software Release Planning", Proceedings PROFES'2004, Lecture Notes on Computer Science, Vol. 3009, pp 248-262.

[Anton 03]
Anton, A. I., "Successful Projects Need Requirements Planning", IEEE Software, 20(6), pp. 44-46, 2003.

[Aurum & Wohlin 03]
Aurum, A., Wohlin, C., "The Fundamental Nature of Requirement Engineering Activities as a Decision-Making Process", Information and Software Technology 2003, Vol. 45 (2003), No. 14, pp. 945-954.

[Bagnall et al. 01]
Bagnall, A. J., Rayward-Smith, V. J., and Whittley, I. M., "The Next Release Problem", Information and Software Technology, 43 (14), pp. 883-890, 2001.

[Beck 01]
Beck, K., "Extreme Programming Explained", Addison Wesley, 2001.

[Carlshamre et al. 01]
Carlshamre, P., Sandahk, K., Lindvall, M., Regnell, B., and Nattoch Dag, J., "An Industrial Survey of Requirements Interdependencies in Software Release Planning", In Proceedings of the 5th IEEE International Symposium on Requirements Engineering, pp. 84-91, 2001.

[Davis 03]
Davis, A. M., "The Art of Requirements Triage",IEEE Computer, 36(3), pp 42- 49, 2003.

[Dantsigner 04]
Dantsigner, E., "Practical Release Planning and Management", University of Calgary, Laboratory for Software Engineering Decision Support, TR 006/04, 2004, 29p.

[Glass 04]
Glass, R. L., "Matching Methodology to Problem Domain", Communications of the ACM, 47(5), 2004.

[Greer & Ruhe 04]
Greer, D. and Ruhe, G., "Software Release Planning: An Evolutionary and Iterative Approach", Information and Software Technology, 46(4), pp. 243-253, 2004.

[Lethola et al. 04]
Lethola, L., Kauppinen, M., and Kujala, S., "Requirements Prioritization Challenges in Practice", Proceedings PROFES'2004, Lecture Notes on Computer Science, Vol. 3009, pp 497-508.

[Momoh 04]
Momoh, J.A., "Applying Intelligent Decision Support to Determine Operational Feasibility of Strategic Software Release Plans". MSc Thesis, University of Calgary, Department of Electrical and Computer Engineering (in preparation), 2004.

[Penny 02]
Penny D. A., "An Estimation-Based Management Framework for Enhancive Maintenance in Commercial Software Products," In Proceedings of International Conference on Software Maintenance, pp. 122-130, 2002.

[Ruhe 03]
Ruhe, G., "Software Engineering Decision Support - A New Paradigm for Learning Software Organizations", Advances in Learning Software Organization, Lecture Notes in Computer Science, 2640 (2003), Springer, pp. 104-115, 2003.

[Ruhe & Ngo-The 04]
Ruhe, G., Ngo-The, A. (2004), "Hybrid Intelligence in Software Release Planning". International Journal of Hybrid Intelligent Systems, Volume 1, No. 2, pp. 99-110.

[Saaty 80]
Saaty, T.L., "The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation", McGraw-Hill, New York, 1980.

[Standish Group 02]
Standish Group Research. "What are your Requirements?", http://www.standishgroup.com/, 2002.