

Towards Comprehensive Release Planning for Software Product Lines

Muhammad Irfan Ullah and Guenther Ruhe

Laboratory for Software Engineering Decision Support, University of Calgary, Canada
{ullah, ruhe}@cpsc.ucalgary.ca

Abstract

Release Planning (RP) plays an important role for the success of incremental product development. Proper planning includes consideration of stakeholder preferences, resources and their capacities, as well as product and business objectives. The complexity of this process is getting even larger when looking for releases of software product lines (SPL). SPL is considered as a viable and important software development paradigm allowing companies under certain conditions to realize order-of-magnitude improvements in time to market, cost, productivity, quality, and other business drivers.

In this paper, we present ongoing research in the process to develop a comprehensive and formalized model for planning and optimizing releases for SPL. We have identified key issues that are unique to RP of SPL. Some of them are highlighted by an example modified from literature for illustrative purposes.

1. Introduction

A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [1]. A software release is a collection of new and/or changed features that form a new product.

Release planning (RP) for incremental software development assigns features to releases such that most important technical, resource, risk and budget constraints are met. Without good release planning ‘critical’ features are jammed into the release late in the cycle without removing features or adjusting dates. This might result in unsatisfied customers, time and budget overruns, and a loss in market share [2].

Release planning for software product lines (RP-PL) involves a number of new aspects, which are not applicable in traditional systems. It has to consider

multiple products being developed by multiple teams for different customers. The research question is to go beyond planning and managing just one, but a couple of different products integrated into a product family.

Currently, there is no RP-PL model that allows addressing the problem in a rigorous formalized manner generating optimal solutions or at least qualified solutions. Most of the popular SPL development methodologies focus on the scoping process. Riebisch et. al. [3] include release planning as part of the scoping process for SPL. They have proposed Decision Tables to prioritize features for assignment to releases. Their approach is semi-formal and does not support quantitative evaluation of release plans. Release Matrix approach proposed by Taborda [4] provides a holistic view of SPL release management however, lacks the formalism needed to generate optimal solutions. For the case of one product, the hybrid approach called EVOLVE* was designed for exactly that purpose. It improves existing methods for release planning by combining the strength of mathematical models with the subtleness of experts’ knowledge [5]. It uses computationally intelligent techniques such as evolutionary computing and principles of multi-criteria decision support in combination with human intelligence [5]. The challenge is to extend this approach to RP-PL.

In this paper, we present ongoing research in the process to develop a comprehensive and formalized model for planning and optimizing releases for SPL. The paper is subdivided into five sections. In the following section 2, we present some terminology and existing results in the context of RP-PL. Section 3 is devoted to discuss the formal model of [5] and mapping of RP-PL problem to reuse existing functionality of the ReleasePlanner™ decision support system [6] developed for optimized release planning of one product. An illustrative example for the problem statement and its solutions is given in Section 4. Finally, Section 5 gives an outlook to future research.

2. State of the Art

2.1. One Product Release Planning

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. This is called release planning. Release planning considers stakeholder priorities and different types of constraints. The output of the release planning process is a set of candidate assignments of features to releases. They are supposed to represent a good balance between stakeholder priorities and the shortage of resources. In each release, all the features are executed following one of the existing software development paradigms including analysis, system design, detailed design, implementation, component testing, system testing, and user testing.

An overview of existing methods for release planning of one product was given in [7]. The following deficits were analyzed:

1. There is no major focus on addressing system constraints. The attempt in [8] assumes operational risk of system failure can be given probabilistic values, without deriving such information from the architecture, code base, and other historical data of the system.
2. There are not enough decision support tools that are fully developed and based on sound formalism, except ReleasePlannerTM.
3. Release planning has been largely focused on “fixed release intervals” and no current work on planning with flexible time intervals.

2.2. Scoping vs. Release Planning for SPL

What is the difference between RP-PL and the scoping activity being an established part of any PL development approach? In general, release planning prioritizes features for assignment to releases while scoping focuses on identification and principal categorization of features. In the strict sense [9], scoping identifies the individual products, features assigned to products and features assigned to core assets of all products. RP deals with prioritizing the features (common and variable) and assigning them to releases. Scoping develops the product portfolio [10] without considering the technological constraints, resource capacities and consumptions. Because of the inherent complexity, we suggest to make this process more rigorous and optimized. Based on the output of scoping process, this will allow generating qualified

alternative release plans in a transparent and repeatable manner.

2.3. Release Planning – Conventional Systems vs. Software Product Lines

RP-PL presents some new challenges, which do not exist in the context of conventional systems. We provide a list of new questions. For some of them, our proposed approach will show how to handle them in a more formal manner.

1. Managing delays in development of core assets due to conflicting requirements of products so that it does not affect product development.
2. Evolving core assets with competing requirements of various products.
3. Synchronizing different product cycles.
4. Allocating resources across product teams.
5. Improving quality, productivity and time-to-market goals while maximizing stakeholders’ satisfaction.
6. Moving excess resources across product development teams to increase resource utilization.
7. Quickly reacting to opportunities for new product development based on existing core assets.

At this stage our proposed RP-PL model can handle points 1 and 2 mentioned above with the help of what-if analysis. For point 3, our model is based on the assumption that all the products have same release dates. For point 4, our model develops the release plans with an assumption that only one team is developing all the products. Our model uses stakeholder preferences in a quantitative manner to develop the release plans thus it fulfills point 5. Since we are considering only one team therefore our RP-PL model does not address point 6. Our model can handle any number of products and therefore it is possible to introduce new products and generate release plans for them based on the existing core assets. Consequently, it satisfies point 7.

3. Modeling and Problem Statement

Ruhe and Ngo-The [5] have presented a hybrid release planning model for conventional product development. It is able to produce optimal release plan solutions by combining the strength of computational intelligence and the knowledge and experience of human experts [5]. In this section we will present the formal notation used by the model [5] and later map the release planning problem for software product lines

on it. In a SPL there are two types of features, i.e. core asset features (F_n) and product specific features (F_m). In our model we are assuming that these two types of features are already decided during the scoping process and core asset features are used by all the products while each product specific feature is required by only one of the products. Therefore, the total set F of features in the SPL is given as $F = F_n + F_m = \{f(1), f(2), \dots, f(n+m)\}$. These features can be assigned to one of the K possible release options. This is described by decision variables $\{x(1), x(2), \dots, x(n+m)\}$ where $x(i) = k$ if we assign feature i to release option $k \in \{1, 2, \dots, K\}$ and $x(i) = 0$ if feature i is postponed.

The model considers two types of dependencies amongst features: coupling and precedence. Coupled features must be released together while the precedence relationship imposes to release features in a certain order [5].

We assume an available resource capacity $\text{Cap}(k)$ for each release k . For simplicity we only consider a single type of resource although the model [5] considers capacities for different types of resources. If feature $f(i)$ requires an amount $r(i)$ of development resources then each release plan x must satisfy

$$\sum_{x(i)=k} r(i) \leq \text{Cap}(k) \quad (1)$$

Stakeholders are extremely important in release planning. Assume a set of stakeholder $S = \{S(1), \dots, S(q)\}$. Each stakeholder p can be assigned relative importance $\lambda(p) \in \{1, \dots, 9\}$. $S(p) = 1$ indicates the lowest and $S(p) = 9$ the highest degree of stakeholder importance. Each stakeholder prioritizes every feature based on two criteria using a nine point scale. The first criterion is represented by $\text{value}(p, i)$ and the second criterion is represented by $\text{urgency}(p, i)$.

The objective is to maximize a function $F(x)$ among all release plans x subject to satisfaction of resource constraints described in equation 1. The function $F(x)$ depends on a number of factors like value of the release, importance of stakeholders, urgency of a feature and its value to stakeholders.

$$F(x) = \sum_{k=1 \dots K} \sum_{i: x(i)=k} \text{WAS}(i, k) \quad (2)$$

with

$$\text{WAS}(i, k) = \xi(k) [\sum_p \lambda(p) \cdot \text{value}(p, i) \cdot \text{urgency}(p, i)] \quad (3)$$

In (3), $\xi(k)$ expresses the relative (normalized to 1) importance of release k . This release planning model is the core functionality of the decision support system ReleasePlanner™ [6]. The goal of our research is to extend its applicability to allow also release planning

for SPL. As a first step in this direction, we have mapped RP problem for SPL on [5] and used ReleasePlanner™ [6] to generate optimal release plans. For that, we have made the following assumptions.

1. Core asset and product specific features along with their mapping on the products are available from the scoping process.
2. All the products in the SPL have the same release dates for respective releases.
3. Only one team is developing all the products in SPL.
4. No customization effort is required to use a core asset in a product.

4. Hypothetical Case Study

We have developed an example by looking at a number of case studies presented in literature and adapted it to highlight some issues relevant to RP. The purpose of this example is to illustrate the mapping of RP problem for SPL on [5] and generate optimal release plans using ReleasePlanner™ [6].

Consider a SPL with two products, Product A and Product B. There are ten candidate features in the product line. The assignment of features to products and core assets (C.A.) is shown in figure 1. The graphical representation shown in figure 1 is adapted from [11] but here we are only interested in precedence and coupling types of interdependencies.

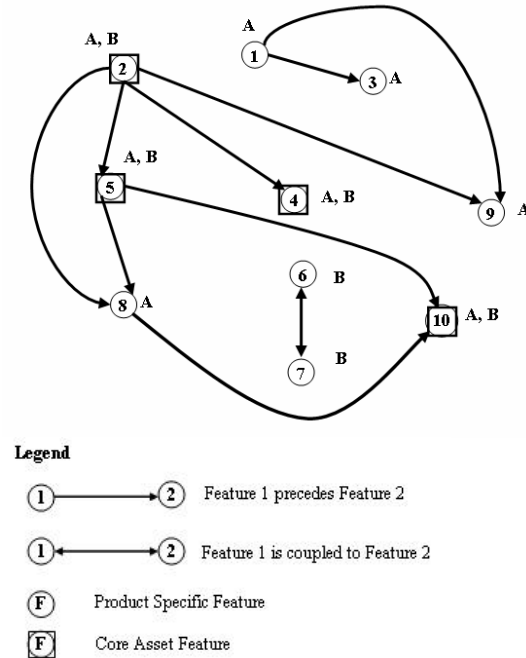


Figure 1: Graphical view of features in the SPL

In this example we are considering $K = 2$ releases. There is one type of development resource and its capacity is $Cap(1) = 55$ person days and $Cap(2) = 45$ person days for release 1 and 2 respectively. The relative importance of releases are $\xi(1) = 0.7$ and $\xi(2) = 0.3$, respectively. The coupling and precedence constraints amongst the features are illustrated in figure 1. There are two stakeholders $S(1)$ and $S(2)$ with relative weights $\lambda(1) = 8$ and $\lambda(2) = 5$. The resource requirements for the ten features are given in table 1. Stakeholders' votes on all of the features for value and urgency criteria are also shown in table 1.

Table 1: Effort estimates and stakeholders' votes

Feature $f(i)$	Effort $r(i)$ (p. days)	Stakeholder $S(1)$		Stakeholder $S(2)$	
		Value(1, i)	Urgency(1, i)	Value(2, i)	Urgency(2, i)
f(1)	5	6	7	4	2
f(2)	14	5	8	7	6
f(3)	7	6	7	5	6
f(4)	16	4	2	6	7
f(5)	15	7	6	6	4
f(6)	10	6	7	6	3
f(7)	5	5	7	7	7
f(8)	8	6	6	5	8
f(9)	13	3	1	4	3
f(10)	10	5	3	7	7

ReleasePlanner™ presents the decision maker with up to five qualified alternative solutions. The term “qualified” here refers to solutions proven to be at least 95% optimal for the stated objective function. We have selected two solutions provided by the ReleasePlanner™ to discuss them in more detail.

4.1. Solution 1

Table 2 presents the structure of solution 1 generated by the ReleasePlanner™. It lists down the features and the releases in which they will be developed. A core asset feature will only need to be developed once for all the products. In this release plan feature f(3) of Product A is postponed (PP).

Table 2: Release plan for SPL: Solution 1

Product A			Product B	
R1	R2	PP	R1	R2
f(2)	f(1)	f(3)	f(2)	f(4)
f(5)	f(4)		f(5)	f(10)
f(8)	f(9)		f(6)	
	f(10)		f(7)	

4.2. Solution 2

Table 3 presents structure of the second alternate solution provided by the ReleasePlanner™.

Table 3: Release plan for SPL: Solution 2

Product A			Product B		
R1	R2	PP	R1	R2	PP
f(2)	f(1)	f(10)	f(2)	f(4)	f(10)
f(5)	f(3)		f(5)		
f(8)	f(4)		f(6)		
	f(9)		f(7)		

4.3. Analysis

Both of the release plans presented above are within 97% quality range with respect to objective function, however, they differ in their structure. One of the major differences is in terms of postponed features. In solution 1, f(3) is postponed while in solution 2, f(10) is postponed. Postponement of f(10) affects both the products since it is a core asset. ReleasePlanner™ also provides comparison of the solution alternatives in terms of resource utilization, stakeholders' satisfaction and performance with respect to prioritization criteria. Based on the requirements of the product, their respective importance, time-to-market and stakeholders, the project manager can then select one of the solution alternatives.

The strength of our technique is based on its ability to combine human knowledge and expertise with computational power to generate different scenarios for supporting pro-active release planning. It also supports generation of what-if scenarios by varying resource capacities, release weights, stakeholder weights or by pre-assigning features to specific releases. One can imagine the complexity of this task when there are tens of products in the product line with hundreds of features to assign to releases while satisfying the constraints. On top of this, the release plan has to maximize the satisfaction of stakeholders' of multiple products having conflicting interests. With the help of ReleasePlanner™ decision support system, release planning of a product line with any number of products and features can be performed efficiently.

5. Work-in-Progress

Some of the assumptions mentioned in section 3, under which we have mapped the RP-PL problem on [5] are restrictive and do not necessarily reflect most of the real world situations. We are currently working to

removing these assumptions by extending [5]. Extensions will include:

1. Allowing stakeholders to vote on developing a feature as core asset or product specific.
2. Ability to generate release plans with multiple teams, each developing one product of SPL.
3. Ability to move excess resources across product and core assets development teams to improve quality of release plan solutions.
4. Assigning relative weights to products in the product line based on their time-to-market requirements, economic benefits and stakeholders, etc.
5. Evaluate impact of different organizational structures i.e. separate core assets team vs. product teams developing core assets on release plan solutions.

Acknowledgements

We thank the Alberta Informatics Circle of Research Excellence (iCORE) and the Higher Education Commission of Pakistan (HEC) for their financial support of this research. We also thank the anonymous reviewers for their detailed and valuable comments.

References

- [1] <http://www.sei.cmu.edu/productlines>.
- [2] D. Penny, "An Estimation-Based Management Framework for Enhance Maintenance in Commercial Software Products," In Proceedings of the *International Conference on Software Maintenance*, Montreal, Canada, 2002, pp. 122-130.
- [3] M. Riebisch, D. Streitferdt and I. Philippow, "Feature Scoping for Product Lines," in Proceeding of *PLEES'03, International Workshop on Product Line Engineering The Early Steps: Planning Modeling and Managing*, Erfurt, Germany, September 2003.
- [4] L. Taborda, "Generalized Release Planning for Product Line Architectures," in Proceedings of the *SPLC 2004, The Third Software Product Lines Conference*, Boston, USA, August 2004, pp. 238 – 254.
- [5] G. Ruhe and A. Ngo-The, "Hybrid Intelligence in Software Release Planning," *International Journal of Hybrid Intelligent Systems*, Vol. 1(2004), pp. 99 - 110.
- [6] <http://www.releaseplanner.com>
- [7] O. Saliu and G. Ruhe, "Supporting Software Release Planning Decisions for Evolving Systems," in Proceedings of the *29th IEEE/NASA Software Engineering Workshop*, Greenbelt, MD, USA, April 2005, pp. 14 – 24.
- [8] D. Greer, "Decision Support for Planning Software Evolution with Risk Management," in Proceedings of the *16th International Conference on Software Engineering and Knowledge Engineering*, Banff, Canada, 2004, pp. 503 – 508.
- [9] K. Schmid, "A Comprehensive Product Line Scoping Approach and Its Validation," in Proceedings of the *24th International Conference on Software Engineering*, Orlando, USA, May 2002, pp. 593 – 603.
- [10] K. Schmid, "Scoping Software Product Lines – An Analysis of an Emerging Technology," in *Software Product Lines – Experience and Research Directions*, P. Donohoe, Ed. Kluwer Academic Publishers, 2000.
- [11] P. Carlshamre et. al., "An Industrial Survey of Requirements Interdependencies in Software Product Release Planning," in Proceedings of the *Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp. 84 – 91.