OPTIMIZED APPROACH TO SOFTWARE RELEASE PLANNING WITH VOLATILE REQUIREMENTS

S.L Gupta, PhD

Director, Birla Institute of Technology, Extension Centre, Patna

Rachna Soni, PhD

Department of Computer Science, DAV College for Girls, Yamuna Nagar (Haryana), India

Ashish Jolly, PhD

Department of Computer Science, Govt. P.G. College, Ambala Cantt (Haryana) India.

Arti Rana

Amity Institute of Information Technology, Amity University, Noida (UP) India

Abstract:

Software release is one of the most prominent issues involved in software development to decide upon the most appropriate software release plans. Software companies are facing many problems with endless product planning challenges, viz. availability of limited resources, resource bottlenecks, conflicting requirements, complex product dependencies, time-to-market pressures and geographically dispersed stakeholders and project teams. Release Planning is an important and essential part of any type of Software Release Management. The present paper works on complexity of Release Planning with conflicting requirements, contentment of the stakeholders to condense the cost of on the whole software release process.

Keywords: Release Planning, Stakeholders, Volatile Requirements, Optimum Cost

1. Introduction

Release planning become very intricate now a day. The goal of the release planning approach is to deliver maximum value to the stakeholders in least time possible. Various factors come into play, such as the availability of resources and the skilled people to use the limited resources for the effective utilization. A software release is a collection of new and/or changed requirements that form a new product. Release planning for software development assigns features to releases such that most important technical skilled employee, resource, risks and budget constraints are met. Without good release planning ‘critical’ requirements are blocked 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 is a keystone problem, it deals with the proper assignment of the requirements sequence in order to maximize profit, minimize the delay of feedback and return of investment in such a way that dependency and resource constraints are met.

2. Related Work

Release planning decisions are required at an early stage in the development cycle, when uncertainty is unavoidable in the project estimates. Software release Planning and allocation of resources cannot be handled in isolation. “Developing and releasing small increments of requirements, in order for customers to give feedback early, is a good way of finding out exactly what stakeholders want, while assigning a low development effort” [3].
Deciding on the requirements for an upcoming software release is a complex process. With the evident pressures on time-to-market [3] and limited availability of resources, often there are more requirements than can actually be implemented. The market-driven requirement engineering processes [3] have a strong focus on requirement prioritization [4]. The requirement list needs to fulfil the interests of the various stakeholders and takes many variables into consideration. Several scholars have presented lists of such variables, including: importance or business value, stakeholder preference, cost of development, requirement quality, development risk and requirement dependencies [3, 4]. D. Greer and G. Ruhe [5] use genetic approach for optimization of stakeholder priorities within technical precedence constraints for balancing required and available resources for all increments. G. Ruhe works on the method EVOLVE for the continuous planning of incremental software development.

Omolade Saliu and Guenther Ruhe [6], describe ten key technical and nontechnical aspects impacting release planning and proposed approach extends the existing solution method called EVOLVE* to enhance the performance of release planning by the proactive analysis of the risk involved in integrating new features into existing components of the system, identifying the importance of estimating the integration effort for each feature based on system characteristics.

The Bo Yang, Huajun Hu in 2006 worked on the approach taken is to minimize the expected total cost (ETC) of the software project, or further consider the software reliability requirement.

In fact, there exists certain risk that the ATC may exceed the ETC to an intolerable extent, which, attracted a lot of research in the past two decades and several new cost models have been developed in the literature recently In this paper, we study the above mentioned risk problem for software release time determination and propose a new approach which could be helpful for management to control the risk of the project being over-budget [9].

Q.P.Hu, R. Peng, M. Xie, S.H. Ng, G. Levitin in 2011dicussed the model for reliability of multi-release software development process. His study provides the dynamics of software faults during this releasing procedure without the loss of generality. Traditionally, release-time issue is addressed with software reliability models for single release considering when to release each release and balancing the competition in market and the risk of low-quality software [10].

3. Stakeholders in Software Release

The actual challenge is to balance all those contra perspectives and to customize the objective function to the real user needs. Successfully solving the problem of release planning involves satisfying the needs of a varied group of stakeholders. Stakeholder may involve variety of people like:

- Sales representative
- Shareholder
- Project manager
- Product manager
- User (novice, advanced, expert)
- Investor
- Developer

Understandings of who the stakeholders are and their particular needs are key elements in developing an effective solution for the software release planning problem. In general, stakeholders will have different perspectives on the problem and different needs that must be addressed by the release plans in terms of quality, time, or business value [7].

Release planning is a highly constrained process. The technological constraints expressed as dependencies between requirements. In addition to that, effort and resource constraints are addressing limitations on the amount of requirements that can be assigned to a certain release. In addition to effort, (bottleneck) resources are considered whenever specific tasks can be performed by specific types of resources only [8].

4. Key Issues of Software Release Planning

Release planning is a very complex problem including different stakeholder perspectives, competing objectives and different types of constraints. Release planning is impacted by a huge number of inherent constraints. Most of the features are not independent from each other. Typically, there are precedence and/or coupling constraints between them that have to be satisfied. Furthermore, effort, resource, and budget constraints have to be fulfilled for each release. The overall goal is to find a relatively small set of “most promising” release plans such that the overall value and the degree of satisfaction of all the different stakeholders are maximized.

• **Stakeholder involvement**: In most cases, stakeholders are not sufficiently involved in the planning process. Stakeholders involvement reduces the complexity of the problem at hand and if not handled properly, they create a huge possibility for project failures.

• **Informal, not well specified and understood requirements:** There is usually no formal way to describe the features and requirements. Non-standard format of feature specification often leads to incomplete descriptions and makes it harder for stakeholders to properly understand and evaluate features and requirements.

• **Ever Changing requirements** and other problem parameters: Features and requirements always change as the project progresses. If a large number of features increase the complexity of the project, their dynamic nature can pose another challenge. Other parameters such as the number of stakeholders, their priorities, etc., also change with time - adding to the overall complexity.

• **Size and complexity** of the problem: Size and complexity are major problems for project managers when choosing release plans - some projects may have hundreds or even thousands of features

5. Proposed Model

In the proposed model we have assumed that we have n stakeholders and m requirements of stake holders. Assuming m<n and applying sensitivity for cost effectiveness the remaining n-m requirements are taken up according to the managerial decision. The requirement incorporation cost and requirement incorporation efficiency are represented through bipartite graph.

During the optimization of requirement allocation we maintain a potential og *network* and an orientation of D_{og} (denoted by $\vec{D}_{og}$) which has the property that the edges oriented from S(S1,S2….Sm) to R(R1,R2…Rn) form a matching *M*. Initially, og is 0 everywhere, and all edges are oriented from R to S (so *M* is empty). In each step, either we modify og so that its value increases, or modify the orientation to obtain a matching with more edges. We maintain the invariant that all the edges of *M* are tight. We are done if *M* is a perfect matching.

In a general step, let $U_R \subseteq R$ and $U_S \subseteq S$ be the vertices not covered by *M* (so U_R consists of the vertices in R with no incoming edge and U_S consists of the vertices in S with no outgoing edge). Let *Z* be the set of vertices reachable in $\vec{D}_{og}$ from U_R by a directed path only following edges that are tight. This can be computed by breadth-first search.

If $U_S \cap Z$ is nonempty, then reverse the orientation of a directed path in $\vec{D}_{og}$ from U_R to U_S. Thus the size of the corresponding matching increases by 1. If $U_S \cap Z$ is empty, then let.

$\Delta := \min \{ c(i,j) - og(i) - og(j) : i \in Z \cap R, J \in S \setminus Z \}$

Δ is positive because there are no tight edges between $Z \cap R$ and Increase *y* by Δ on the vertices of $Z \cap R$ and decreases og by Δ on the vertices of $Z \cap S$. The resulting og is still a potential. The graph D_{og} changes, but it still contains *M*. We orient the new edges from R to S. By the definition of Δ the set *Z* of vertices reachable from U_R increases (note that the number of tight edges does not necessarily increase).

We repeat these steps until *M* is a perfect matching, in which case it gives a minimum cost assignment. The running time of this version of the method is $O(n^4)$: *M* is augmented *n* times, and in a phase where *M* is unchanged, there are at most *n* potential changes (since *Z* increases every time). The time needed for a potential change is $O(n^2)$.

Notations used:

og= Orientation of graph from E to S.

D_{og}= Directed Network

M= Weights of different edges.

S_i= The stakeholders. i=1,2……………..m.

R_j= The requirements .j=1,2…………n.

$c_{(i,j)}$ = requirement incorporation cost/requirement incorporation time (Case II).

U_S= Subset of S.

U_R= Subset of R.

Z= Set of vertices reachable in D_{og}.

Δ= Minimized value after row reduce & column reduce.

Here two cases are taken firstly in terms of requirement incorporation cost, then secondly in terms of optimal time for software release.

Case I

Here using the above algorithm, M is taken as the requirement incorporation cost to get the optimal solution for Requirement realization to different stakeholders.

	R1	R2	R3	R4	R5
S1	30	29	26	29	21
S2	25	31	29	30	20
S3	18	19	30	19	18
S4	32	18	19	24	17
S5	27	21	19	25	16
S6	19	20	22	19	14
S7	22	30	23	18	16
S8	26	19	26	21	18

Table 1

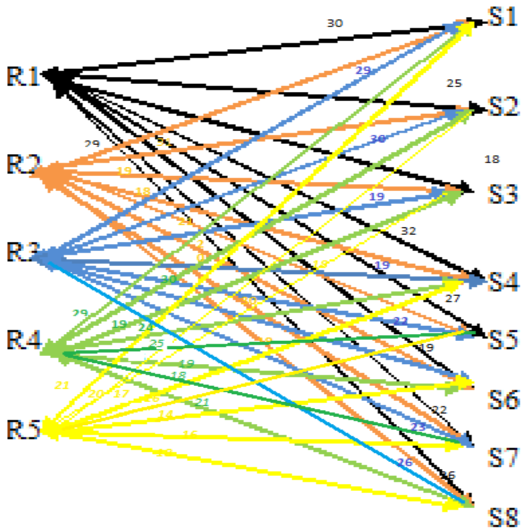

Fig. 1 Cost Network.

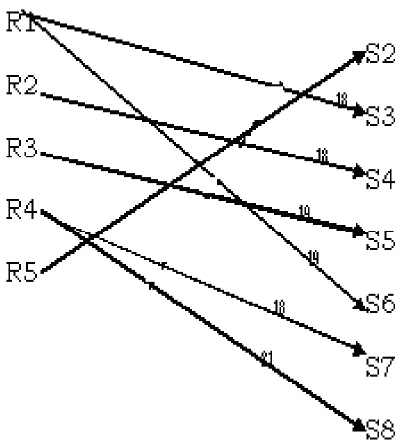

Fig.2: Optimal solution for Cost.

Fig 2 represents the optimal solution for requirement realization to different stakeholders. Here R1 and R4 requirements were assumed to be more implementable ones. The optimal requirement realization cost for the total project comes out to be Rs 133 thousand. Optimally S1 and S3 stakeholders are being considered to only one requirement R4 and S2 and S7 stakeholders are being considered for R1.

Case II

Here the algorithm uses M as requirement incorporation time of particular stakeholder n terms of weekly hours. Table 2 and Fig. 3 represent the data taken.

Table 2

	R1	R2	R3	R4	R5	R6
S1	3	6	2	6	5	0
S2	7	1	4	4	7	0
S3	3	8	5	8	3	0
S4	6	4	3	7	4	0
S5	5	2	4	3	2	0
S6	5	7	6	2	5	0

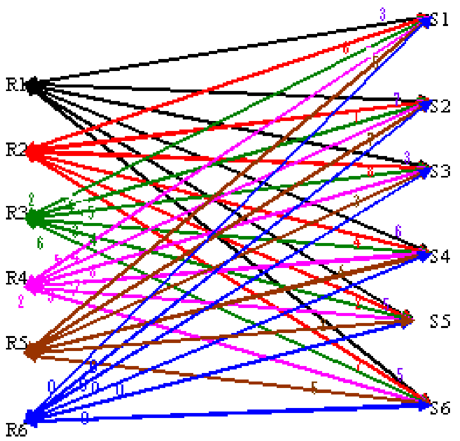

Fig.3

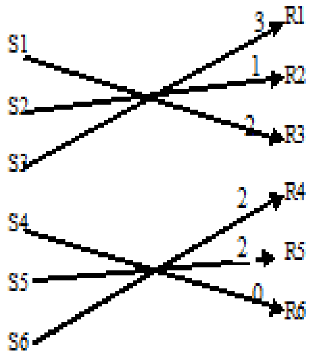

Fig.4

The figure 4 represents the optimal solution for requirement consideration for different stakeholders in terms of time allocation.

6. Conclusion

In this paper we proposed a model for optimum requirement realization cost and time. Volatile requirements are broadly grouped into five groups, stakeholders broadly grouped into eight different categories, and then the approach is applied to find the optimum cost and time to satisfy the maximum requirements of the varied stakeholders on time with minimum cost, maximum quality, design and performance.

References:

Optimized Resource Allocation for Software Release Planning An Ngo-The; Ruhe, G.

Penny, D., "An Estimation-Based Management Framework for Enhancive Maintenance in Commercial Software Products". In: Proc. International Conference on Software Maintenance, pp. 122-130, 2002.

Carlshamre, P., "Release Planning in Market-Driven Software Product Development: Provoking an Understanding". In: Requirements Engineering 7, pp. 139-151, 2002.

Joachim K, Kevin R (1997) A cost-value approach for prioritizing requirements. IEEE Softw 14(5):67–74

Greer D, Ruhe G (2004) Software release planning: an evolutionary and iterative approach. Inf Softw Technol 46:243–253

Omolade Saliu and Guenther Ruhe(2005) Supporting Software Release Planning Decisions for Evolving Systems Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05).

Amandeep, Günther Ruhe, Mark Stanford, Intelligent Support for Software Release Planning, Proceedings PROFES'2004, LNCS Vo. 3009, pp 248-262

Ruhe, G., Ngo-The, A., Hybrid Intelligence in Software Release Planning. Appears in: IJHS, Vol 1 (2004).

Bo Yang, Huajun Hu, Jun Zhou, 'Optimal Software Release Time Determination with Risk Constraint' (2006), on page(s): 393, E-ISBN: 978-1-4244-1461-1, Print ISBN: 978-1-4244-1460-4, Issue Date: 28-31 Jan. 2008

Q.P.Hu1, R.Peng2, M. Xie2,3, S.H. Ng 2, G. Levitin4 , 'Software Reliability Modelling and Optimization for Multi-release Software Development Processes', ISSN: 2157-3611, Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on , 6-9 Dec. 2011.