ISSN No:-2456-2165

Time Centric Model-An Optimal Solution to Software Release Planning

Sandhia Valsala
College of Business (MIS Department)
University of Buraimi
Sultanate of Oman

Abstract:- Release planning is the process of defining the functionality of related sequence of product releases or iterations The first and foremost process in solving a RP problem is requirement selection and prioritization. The existing RP models concentrates only on selecting and prioritizing the requirements that need to be included in the current release without giving much importance to scheduling. The high software failure rate and the complex relationship between requirements support the argument that scheduling should be integrated into release planning activity. Proposed in this paper is a time centric model that integrates prioritization and scheduling into release planning.

Most of the existing models when tested with varying dependency between requirements showed that as the number of requirement dependency increases there are more chance for the project to be delayed i.e., these prioritized requirements when scheduled to the development teams exceeds the release date; it never always guarantee an on time schedule. A Time Centric Model can overcome this issue of project delay by guaranteeing an on time delivery and this algorithm selects the best combination of prioritized requirements that can always be scheduled to development teams such that it finishes on time. This approach provides a schedule that will always meet the target release date and thus provides an optimized solution to prioritization and scheduling problem in software release planning.

Keywords:- *Release Planning, Time Centric, Prioritization, Scheduling, Requirements.*

I. INTRODUCTION

Software Development process involves four phases namely Requirement Management, Design and development, Software Delivery and Maintenance [1]. One of the key activities in requirements management phase is "Release Planning". Release planning is the process of selecting the right group of requirements to be included in each [2]. Most organizations do this informally by choosing the requirements based upon individual experience and tacit knowledge of experts. If planning is done badly it could increase risks such as budget overruns and project delay. The key to successful software projects is in making the right requirements decisions. These decisions should also respect the needs of users and other stakeholders in addition to satisfying various other system constraints. These constraints include requirement dependencies, cost, resource constraints, and development effort, risk etc.

Release Planning is considered to be the most important activity in the requirement engineering phase [3] whose sole aim is to identify the best set of requirements that could be included in the upcoming release satisfying various constraints like the business value of the requirement, stakeholder influence, cost, development effort, risk, requirement dependency, resource availability etc. Since a good release plan is a blueprint of upcoming releases the main focus should be on the selection and prioritization of requirements satisfying the variables described above. Most of the existing release planning models end their activity with the selection of prioritized requirements which are ready for implementation in the current release [4].The release planning activity is extended to include scheduling aspect which is often considered as another step after release planning. Scheduling these selected requirements to various development teams is also important so as to ensure on time release. If software release planning activity just ends with prioritization and selection of optimal set of requirements, later during the implementation phase it can pose a problem when these requirements cannot be scheduled efficiently to the development teams. It will be too late to go for a reprioritization at this stage. So it is necessary to integrate both these activities (Prioritization and Scheduling) into Software Release Planning (SRP).

The objective of SRP is not only to find the best composition of requirements to be included in a release but also to schedule these requirements to development teams so that the project will meet the target release date. If release planning is just about selecting the most optimal set of requirements there is no guarantee that these selected requirements when scheduled to development teams will meet the target release date. Another factor that strongly supports the claim of including scheduling as an important activity in release planning is the fact that most of the requirements are interdependent and these dependencies further increase the complexity of the project plan.

II. LITERATURE REVIEW

A total of 31 release planning models have been considered in this study. A number of requirement prioritization techniques have been applied on these models to deal with this multi- aspect optimization problem which includes Analytical Hierarchy Process (AHP), Fuzzy logic, Integer Linear Programming (ILP) and Models using

ISSN No:-2456-2165

Genetic Algorithm [5]. Cost Value Approach (CVA) [6] prioritizes requirements following the rule "maximize quality and minimize cost". The model focuses on prioritizing requirements based on stakeholder's preference by making a pairwise comparison of customer requirements according to their relative cost and value. Incremental Funding Method (IFM) [7] aims at maximizing the Net Present Value (NPV) while selecting requirements. Evolutionary and Iterative Approach (EVOLVE) and its family series solves RP problem by an evolutionary and iterative approach. Evolve-Star (Evolve*) [8] improves the existing release planning models by combining the strength of mathematical models with experts knowledge. EVOLVE [9] uses Genetic algorithm to provide optimum allocation of requirements to releases considering stakeholder preference and also balances resources to all releases. Extension of Evolve (EVOLVE+) [10] is a combination of genetic algorithm with an iterative solution method. The solution, if it satisfies the risk constraint, precedence constraint and resource constraint are selected. System Evolve-Star (S-EVOLVE*) [11] apart from considering other constraints also considers the knowledge of existing system in deciding on the requirements that need to be included in the next release. Financial Evolve Star (F-EVOLVE*) [12] in addition to other constraints selects features that give high returns while solving RP problem. Evolve Extended (EVOLVE ^{ext}) [13] is an extension of EVOLVE * model that combines computational and human intelligence by finding an optimal balance between computing stakeholder priorities and bottleneck resources. Art and Science of Release Planning (AHPSRP) [9] is a hybrid release planning model that uses human intuition to formalize the problem and applies computational algorithm to generate optimal solutions. Evolutionary EVOLVE+ [14] is an extension of hybrid intelligence approach EVOLVE* by combining computational complexity and cognitive complexity. Next Release Problem (NRP) [15] uses a heuristic approach (GA and Simulated Annealing) in solving RP problem by considering various factors like customer's value and cost in feature selection. Multi Objective Release Planning (MORP[16] follows the following approach in solving RP problem, maximize stakeholder preference and minimize project risks respecting available resource and requirement dependency. Multi Objective Next Release Problem (MONRP) [17] takes in two objective functions while solving RP problem, maximize customer satisfaction and minimize required cost. Bi- Objective Release Planning for Evolving Systems (BORPES) considers existing system in making release decisions by including highly coupled features in the same release. Quantitative Win Win (AEQWW) [11] proposes a small number of possible sets of requirements from which the actualdecision maker can choose and select the most appropriate solution. Analytical Model of Requirement Selection (AMRSQE) [9] uses queuing theory concept by having a queue of requirements with different quality estimated based on market value and development effort. Quality Performance (QUPER) [18] is an extension of CVA model and considers requirement quality as the main factor for requirement selection. A Mathematical Formulation for Flexible Release Planning (AMFFRP) [18] uses ILP method to find an optimal set of requirements with maximum projected revenue against available revenues. Release Planning with Feature trees (RPFT) [19] uses feature tree to structure the requirements by using AND, OR and REQUIRE dependencies and utilize feature trees for planning the release. MAX-MIN Ant System with a dynamic Roulette Wheel Release Planning (MMASDRW-SRP)[20] uses Ant Colony Optimization (ACO) by adopting a roulette wheel strategy to obtain solutions within a reasonable amount of computational cost. Release Plan Simulator (REPSIM-1) [21] follows an hybrid approach which ensures stability to release plan as features assigned to releases may change with time. RP with Fuzzy Effort Constraints (RPUFEC) [14] uses Fuzzy logic in solving RP problem by finding a release plan that maximizes stakeholder satisfaction. It also handles uncertainties regarding effort estimation. Quality Improvement Paradigm (QIP) [22] uses genetic algorithm in finding the most optimal solution but considers data from previous releases in making this decision. An Optimization technique for RP (AOTRP) [23] provides a solution to software release planning by optimizing revenue against available resources in a given time period. Fuzzy Model for Dependence Constraints in RP (FMDCRP) [14] uses fuzzy logic to handle uncertainty in identifying the structural dependency constraints between requirements. Fuzzy Optimization Model (FOMRP) [14] also addresses uncertainty in release planning using fuzzy logic. Consensus Driven value Based RP approach (CDVBRPA) [24] is an efficient method of release planning used in small organizations. The method uses value-based and consensus-driven approach in solving RP problems. Post Release Analysis of Requirement Selection Quality (PARSEQ) [25] is aimed to improve release planning decisions that are made in previous releases. It is generally a Post analysis that does reprioritization by re-estimating the cost and value of each requirement based on which incorrect decisions (about requirement selection) are inspected. Risk Driven Method for Extreme Programming (RDMXP_XP) [26] is a risk driven method for XP release planning by creating a list of feasible plans and then assessing the risk of each feasible plan and ultimately stakeholders decide on a suitable release plan. Release Planning and Constraint Programming (RP and CP) [27] is a hybrid approach combining the strength of Constraint Programming (CP) and Release by (RP). An Integer Linear Programming (ILP) Model to solve Software Release Planning and Scheduling [28] is the only model that uses Scheduling aspects in solving RP problem. An ILP formulation is used to solve the RP problem which maximizes revenue and also finds an on-time schedule. It is claimed to be the only model that integrates scheduling into software release planning.

It is essential to strengthen the existing RP models by integrating both the process of prioritization and scheduling into RP process. Also there is a need to study the effect of varying requirement dependency during the prioritization and scheduling phase while solving an RP problem.

ISSN No:-2456-2165

III. TIME CENTRIC MODEL

In software release planning process, currently available prioritization and scheduling mechanisms do not analyze time delay privileges. On increasing the requirement dependencies, it can be seen that there are more chance for the project to exceed the release date i.e. the project will not be completed on time. As a result, it is necessary to employ a time delay analysis mechanism while performing prioritization and scheduling. So it is necessary to develop a release planning model that will provide a group of requirements that can always be scheduled on time. The model guarantees an on- time-delivery schedule, but this additional constraint may affect the revenue of the selected requirements. The aim of this proposed model is to provide a timely system and it is extremely useful when there is strict release deadline. Hence this model *is called the "Time Centric Model".*

Suppose if we have strict deadline on release date the Revenue Centric models cannot guarantee an optimal solution. So a Time Centric model is proposed which always selects that group of requirements which can be scheduled to development teams in such a way that it will never exceed the target release date. It always guarantees an on time project delivery. In those cases where there is no delay the Revenue Centric Model and Time Centric Model choose the same group of requirements. But in those cases where there is a delay, the Time Centric model chooses another set of requirements that can be completed on time. In other words Time Centric Model is a 0% delay model. But being too strict on time deadline causes this model to compromise on revenue.

The proposed model does requirement prioritization by the Genetic Algorithm. Initially m number of requirements is selected. Most of the requirements are independent with each other and some of the requirements are dependent with some other requirements. Hence, requirements prioritization initially fixes the dependencies between the requirements based on the six dependency factors such as exclusion, combination, cost- based, implication, the time based and revenue based dependencies. Here, we consider the time dependency as an important factor to avoid time delay.

This model integrates both the activities of prioritization and scheduling and adds a time delay analysis mechanism which is triggered whenever there is a delay in project schedule. When this delay is triggered it goes back to the prioritization process to select the next group of optimal requirements. This process is looped until the algorithm generates a group of requirements that can be scheduled within the release date. The proposed model framework is depicted in figure given below.

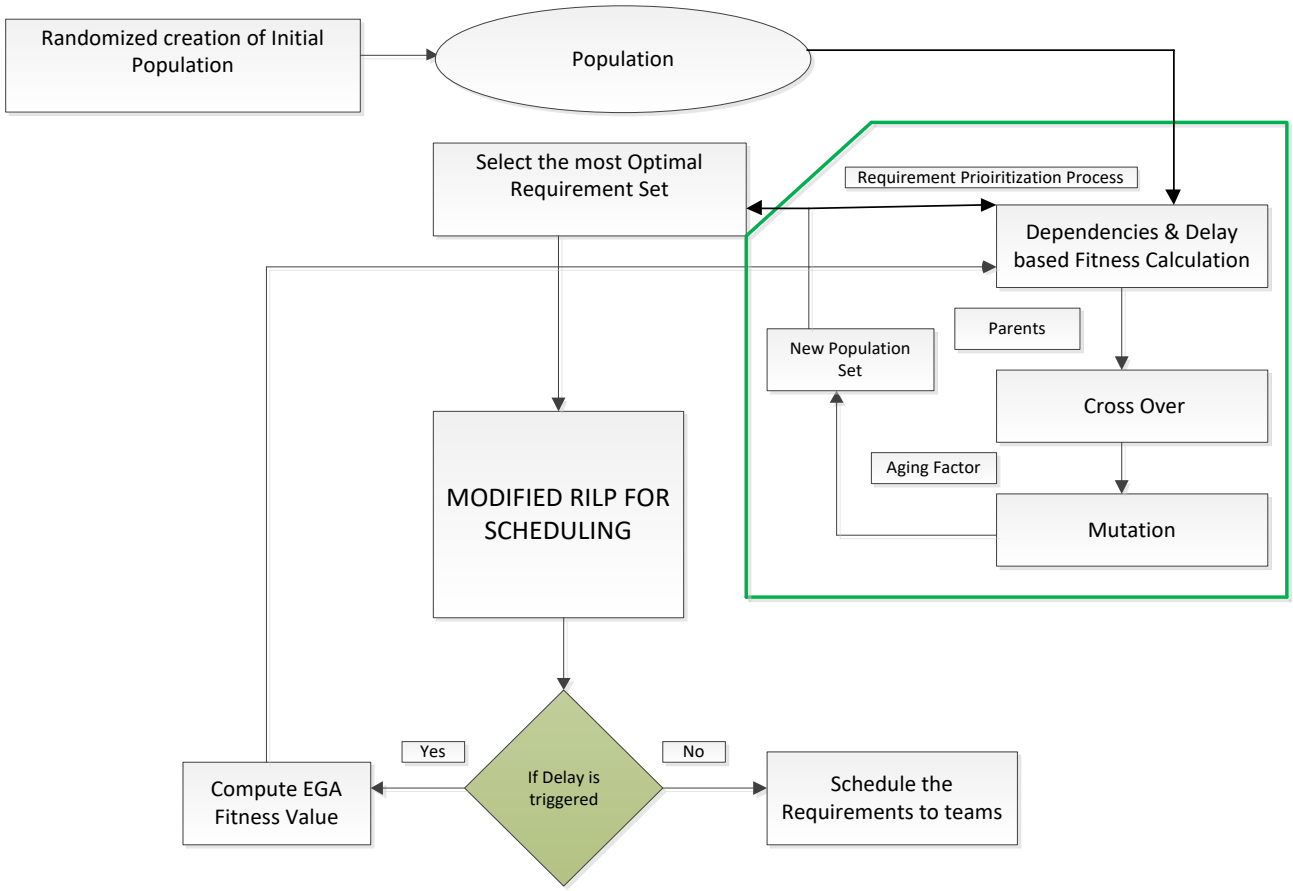

Fig 1:- Proposed Time Centric Model

The ultimate aim of this algorithm is to create a timely schedule a time delay analysis mechanism is added to this algorithm to check if there is a delay. If a delay is triggered during the scheduling process it goes back to prioritization process and another set of requirements is selected. The process is repeated until we find that group of requirements which can be scheduled to development teams within the target release date.

The scheduling delay is computed by using below equation

$delay_g = Total\ Scheduling\ days - release\ days$

$delay_g$ will be a positive value for delayed cases.

For non-delay cases set $delay_g$ <=0

In delayed cases set $Monitor\ Alert\ trigger$ to 1 and compute the fitness value as follows:

$$EGA\ FITNESS\ VALUE = delayg + Monitor\ Alert\ trigger\ X\ 10^1$$

- Influence of Varying Dependency on Multiple Data Set
The model was tested against multiple data sets. The main aim of this simulation was to check if the model schedules the prioritized requirements always on schedule even when we change the number of dependency between the requirements. The following requirement dependency was considered in these simulations. The simulation results are shown below.

99 Req Data Set	Influence of varying dependency using Revenue Centric Model			
	Dependency Ratio	Average Days	No of Delays	Revenue
Small set (8 req,60 days)	10	60	0	1570.56
	20	60	0	1212.49
	30	60	0	998.25
	40	60	0	856.67

Table 1:- Varying Dependency and Scheduling Results on Time centric model(99 Req data set)

76 Req Data Set	Influence of varying dependency using Revenue Centric Model			
	Dependency Ratio	Average Days	No of Delays	Revenue
Small set (5 req,38 days)	10	38	0	706.6
	20	38	0	486.85
	30	38	0	354.05

Table 2:- Varying Dependency and Scheduling Results on Time centric model(76 Req data set)

Satisfying various explicit and implicit objectives and constraints. This process of assigning requirements or features to different increments or releases is known as Software Release planning (SRP). Majority of the existing release planning models concentrate only on generating an optimal set of requirements and give least importance to the scheduling aspect. These models when tested with varying dependency between requirements and it was seen that as the number of requirement dependency increases there are more chance for the project to be delayed i.e., these prioritized requirements when scheduled to the development teams exceeds the release date; it never always guarantee an on time schedule. To overcome this issue of project delay which still exists on increasing the number of dependencies between the requirements a Time Centric model is proposed. Hence an on time delivery is guaranteed with this algorithm selecting the best combination of prioritized requirements that can always be scheduled to development teams such that it finishes on time. This model employs a time delay analysis method while performing prioritization and scheduling and thus provides a schedule that will always meet the target release date and thus proposed is an optimized solution to prioritization and scheduling problem in software release planning. The aim of this proposed Modified Hybrid model is to provide an on-time project delivery and it is extremely useful when the release deadline is very strict. The model guarantees an on-time- delivery schedule, but this additional constraint makes the model to compromise on revenue. The Time Centric model always selects that group of requirements that can be completed within target release date. This model has no influence on requirement dependency and has to compromise on revenue for having an additional on-time delivery constraint.

IV. CONCLUSION

One of the most important activities during requirements engineering is the selection and prioritization of requirements

ACKNOWLEDGMENT

I would like to express my heartfelt thanks to my respectable and friendly guide Dr. Anil R Nair, Ph.D.,his wise academic advice and ideas have played an extremely vital role in the work presented in this paper. Without his support, this research work would not have been possible.

On analyzing the results it can be concluded that the Time Centric Model model saves time by providing timely schedule, but this additional constraint causes the model to compromise on revenue. The proposed Time Centric model guarantees an on time project delivery and is independent on the number of dependencies between requirements. From the simulations it is clear that the model has to compromise on revenue for having an additional on-time delivery constraint.

REFERENCES

[1]. Dabbagh, M., & Lee, S. P. (2014). An Approach for Integrating the Prioritization of Functional and Nonfunctional Requirements. *The Scientific World Journal Volume, Article ID-737626*, pp.1-12.
[2]. Danesh, A., & Ahmad, R. (2012). "Companies Approaches in Software Release Planning – Based on Multiple Case Studies. Journal of Software, Vol 7(2), pp.471-478.
[3]. Lai, R., & Ali, N. (2013). A Requirements Management Method for Global Software Development. Advances in Information Sciences (AIS), Vol 1, pp.38-58.
[4]. Maglyas, A., & Fricker, S. (2014). The preliminary results from the software product management state-of-practice survey. Springer International Publishing, Vol 182, pp.295- 300.
[5]. Svanberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S. B., & Shafique, M. U. (2011). A Systematic Review on Strategic Release Planning Models. Information Software Technology, Vol 52(3), pp.237-248.
[6]. Carlshamre, P., Sandhal, K., Lindvall, M., Regnell, B.,Nattoch, J. D. (2001). An Industrial Survey of RequirementInterdependencies in Software Product Release Planning.*RE'10,IEEE Computer Scociety*, pp.84-91.
[7]. Denne, M., & Huang, J. (2004). The Incremental Funding Method: Data Driven Software Development. IEEE, Vol 21(3), pp.39-47.
[8]. Ruhe, G., & Ngo, A. (2004). Hybrid Intelligence in Software Release Planning. *International Journl Hybrid Intell.Syst, Vol 1*, pp.99-110.
[9]. Ruhe, G., & Sailu, O. (2005). The Art and Science of Software Release Planning. *IEEE Software*, pp.47-53.
[10]. Ruhe, G., & Des, G. (2003). Quantitative Studies in Software Release Planning under Risk and Resource Constraints. Proceedings of the 2003 International Symposium on Empirical Software Engineering, (pp. 262-270).
[11]. Sailu, O., & Ruhe, G. (2005). Supporting Software Release Planning Decisions for Evolving Systems. 29th Annual IEEE/NASA Software Engineering Workshop.
[12]. Maurice, S., Ruhe, G., Sailu, O., & Ngo, A. (2006). Decision Support for Value Based Software Release Planning. *Journal of Value Based Software Engineering*, pp.247-261.
[13]. Ruhe, G., & Momoh, J. (2005). Strategic Release Planning and Evaluation of Operational Feasibility. Proceedings of the 38th Annual Hawaii International Conference on System Sciences, (p. p313).
[14]. Ngo, A., & Ruhe, G. (2007). A Systematic Approach for solving the Wicked Problem of Software Release Planning. Soft Comput, Vol 12, pp.95-108.
[15]. Freitas, F. G., Coutinho, D. P., & Souza, J. T. (2011). Software Next Release Planning Approach through Exact Optimization. *International Journal of Computer Applications, Vol 21*, pp.1-8.
[16]. Colareo, F., Souza, J., Carmo, R., Padua, C., & Mateus, G. R. (2009). A New Approach to Software Release Planning. *Brazilian Sympoium on Software Engineering(SBES '09)*, (pp. 207-215). Brazil.
[17]. Zhang, Y., Harman, M., & Mansouri, A. (2010). The MultiObjective Next Release Problem. Proceedings to the 9th annual conference on Genetic and Evolutionary Computation, (pp. 1129-1137).
[18]. Regnell, B., Svensson, R. B., & Olsson, T. (2008). Supporting Road Mapping of Quality Requirements. IEEESoftware, pp.42-47.
[19]. Fricker, S., & Schumacher, S. (2012). Release Planning with Feature Trees :Industrial case. *Information and Software Technology*, pp.288-305.
[20]. [20]. Yu, Q. H., Chuan, W. C., & Hsang, H. C. (2013). Applying a MAX MIN Ant System with a Dynamic Roulette Wheel Strategy to Software Release Planning.
[21]. Pfahl, D., Al-Emran, A., & Ruhe, G. (2007). A System Dynamics Simulation Model for Analyzing the stability of Software Release Plans. Software Process Improvement and Practice, Vol 12(5), pp.475-490.
[22]. Amandeep, A., Ruhe, G., & Stanford, M. (2004). Intelligent Support for Software Release Planning. Proceedings to PROFES, LNCS, Vol 3009, pp. 248-262.
[23]. Akker, V. d., Brinkkemper S, Diepen G, & Versendaal J. (2008). Software Release Planning through Optimization and What If Analysis. Information and Software Technology, Vol 50(12), pp.101-11.
[24]. Zemer, S., Sampaio, F., & Stalhane, T. (2006). A decision Modelling for analyzing requirements configuration trade-offs in time-constrained Web Application Development. *18thInternational Conference on Software Engineering an Knowledge Engineering*, (pp. 144-149).
[25]. Du, G., Ritcher, M. M., & Ruhe, G. (2006). An

Explanation Oriented Dialogue Approach and its Application to Wicked Planning Problems. Journal of Computing and Informatics, pp.223-249.

[26]. Mingshu, L., Meng, H., Fengdi, S., & Juan, L. (2006). A Risk Driven Method for Extreme Programming Release planning. Proceedings to 28th International Conference on Software Engineering, pp.423-430.

[27]. Przepiora, M., Karimpour, R., & Ruhe, G. (2012). A Hybrid rlease Planning Method and its Empirical Justification. ESEM. Swedon.

[28]. Li, C., Akker, V. d., Brinkkemper, S., & Diepen, G. (2010). An Integerated Approach for requirement selection and scheduling in Software Release Planning. Journal of requirement Engineering, Springer-Verlg, Vol 15, pp.369-375.