International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Research Paper
Available online at: www.jrrset.com
UGC Approved Journal No: 45483
Volume 8, Issue3 Pages 29-30

OPTIMIZED RESOURCE ALLOCATION FOR SOFTWARE RELEASE PLANNING

[1]Dr.Muntha Raju,[2]Mr. Mohd Mohammed Ali,[3] Ms. A. Annapurna
[1]Professor, [2]Associate Professor, [3]Assist professor
[1,2,3,]Dept of CSE ,Shadan College of Engg & Tech

ABSTRACT- Incremental software development offers products in releases where each release provides additional or modified functionality compared to the previous release. Incremental delivery of software products provides business value earlier and allows for quicker reception of customer feedback. Release planning for incremental software development assigns features to releases such that technical, resource, risk, and budget constraints are met. A feature can be offered as part of a release only if all of its necessary tasks are done before the given release date. We assume a given pool of human resources with different degrees of productivity to perform different types of tasks. In the context of release planning, the question studied in this paper is how to allocate these resources to the tasks of implementing the features such that the value gained from the released features is maximized.
Keywords: Portland Pattern Repository, OPTIMIZE RASORP

1. INTRODUCTION

Planning of software releases and allocation of resources cannot be handled in isolation. To address the inherent difficulty of this process, we propose a two-phase optimization approach called OPTIMIZE RASORP that combines the strength of two existing solution methods.
Phase 1 applies integer linear programming to a relaxed version of the full problem.
Phase 2 uses genetic programming in a reduced search space to generate operational resource allocation plans. The method is evaluated for a series of 600 randomly generated problems with varying problem parameters. The results are compared with a heuristic that locally allocates resources based on a greedy search.

2. LITERATURE REVIEW

A release planning meeting is used to create a release plan, which lays out the overall process. The release plan is then used to create iteration plans for each individual iteration. It is important for technical people to make the technical decisions and business people to make the business decisions. Release planning has a set of rules that allows everyone involved with the process to make their own decisions. The rules define a method to negotiate a schedule everyone can commit to. The essence of the release planning meeting is for the development team to estimate each user story in terms of ideal programming weeks. An ideal week is how long you imagine it would take to implement that story if you had absolutely nothing else to do. No dependencies, no extra work, but do include tests. The customer then decides what story is the most important or has the highest priority to be completed. User stories are printed or written on cards. Together developers and customers move the cards around on a large table to create a set of stories to be implemented as the first (or next) release.
A useable, testable system that makes good business sense delivered early is desired. When planning by time multiply the number of iterations by the process velocity to determine how many user stories can be completed. When planning by scope divide the total weeks of estimated user stories by the process velocity to determine how many iterations till the release is ready. Individual iterations are planned in detail just before each iteration begins and not in advance. The release planning meeting was called the planning game and the rules can be found at the Portland Pattern Repository. When the final release plan is created and is displeasing to management it is tempting to just change the estimates for the user stories. The estimates are valid and will be required as-is during the iteration planning meetings. Underestimating now will cause problems later. Instead negotiate an acceptable release plan. Negotiate until the developers, customers, and managers can all agree to the release plan. The base philosophy of release planning is that a process may be quantified by four variables; scope, resources, time, and quality. Scope is how much is to be done. The major advantage is tasks can be defined to an even more fine-grained level. In addition, managerial support and other tasks can be considered here as well.

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Research Paper
Available online at: www.jrrset.com
UGC Approved Journal No: 45483
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

3. EXISTING SYSTEM

A major problem faced by companies developing or maintaining large and complex systems is determining which elements of a typically large set of candidate features should be assigned to which releases of the software. In addition, there is the question of how to assign resources accordingly. The solution of this complex problem is of pivotal importance for process success. Without good release planning, critical features are not provided at the right time. One of the key limitations of current release planning methods is the lack of a systematic process to balance the appropriate delivery of features with the resources available. The solution of this complex problem is of pivotal importance for process success. Without good release planning, critical features are not provided at the right time. One of the key limitations of current release planning methods is the lack of a systematic process to balance the appropriate delivery of features with the resources available.

4. PROPOSED SYSTEM

In this process we augment the scope of release planning by examining details of the resource allocation necessary to actually develop the features. For that, we assume that each feature is decomposed into a sequence of tasks such as design, implementation, and testing. These development tasks can be defined to an even more fine-grained level. In addition, managerial support and/or other tasks can be considered here as well. The major advantage is tasks can be defined to an even more fine-grained level. In addition, managerial support and other tasks can be considered here as well.

5. CONCLUSION

The expected practical benefit of the planning method is to provide release plan solutions that achieve a better overall business value (e.g., expressed by the degree of stakeholder satisfaction) by better allocation of resources. Without ignoring the importance of the human expert in this process, the main contribution of the paper is seen in making the overall process more objective and more qualified in terms of the results. This also increases transparency of solutions, especially when compared to making subjective ad hoc decisions. Plans for resource allocation and provision of features need to be adjusted in the course of a process. It is important to have a qualified starting point for these necessary adjustments caused by different changes in process parameter.

REFERENCE

1. S. Acun˜ a, N. Juristo, and A.M. Moreno, “Emphasizing Human Capabilities in Software Development,” IEEE Software, vol. 23, no. 2, pp. 94-101, Mar./Apr. M. Crissis, M. Konrad, and S. Shrum, CMMI—Guidelines for Process Integration and Product Improvement. Addison-Wesley, 2006.
2. A. Amandeep, G. Ruhe, and M. Stanford, “Intelligent Support for Software Release Planning,” Proc. Fifth Int’l Conf. Product Focused Software Process Improvement, pp. 248-262, 2004.
3. J. Blazewicz, W. Domschke, and E. Pesch, “The Job ShopScheduling Problem: Conventional and New Solution Techniques,”European J. Operational Research, vol. 93, no. 1, pp. 1-33, 1996.
4. L. Briand, J. Feng, and Y. Labiche, “Experimenting with Genetic Algorithms and Coupling Measures to Devise Optimal Integration Test Orders,” Software Eng. with Computational Intelligence, pp. 204-234, Kluwer Academic Publishers,