

A novel method of optimized Resource Allocation for Software Release Planning

Sri Manoj Kumar Tyagi

KI university
Department of ECM
Vaddeswaram

r.maradugu@gmail.com

M.Pushpa Rekha

kl university
Department of ECM
Vaddeswaram

kolli.vamsikrishna@gmail.com

K. Vamsikrishnareddy

kl university
Department of ECM
vaddeswaram

sudhakarklc@gmail.com

A. Sudhakar

kluniversity
Department of ECM
vaddeswaram

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 project is how to allocate these resources to the tasks of implementing the features such that the value gained from the released features is maximized.

Key words: human resources, productivity, incremental software development.

1. Introduction

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 project is how to allocate these resources to the tasks of implementing the features such that the value gained from the released features is maximized. A release planning meeting is used to create a release plan, which lays out the overall project. 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 project to make their own decisions. The rules define a method to negotiate a schedule everyone can commit to. In strategic planning, resource allocation is a

plan for using available resources, for example human resources, especially in the near term, to achieve goals for the future. It is the process of allocating resources among the various projects or business units

1.1 PROBLEM DEFINITION

This project is regarding how to allocate the resources to the tasks. A feature can be offered if all of its necessary tasks are done before the given release date. Without good release planning, critical features are not provided at the right time. Incremental software development provides additional or modified functionality. In this project we examine details of the resources. Each feature is decomposed into a sequence of tasks. The system is a web based system. The administrator has full control on the system. In the context of release planning, the question studied is how to allocate these resources to the tasks of implementing the features such that the value gained from the released features is maximized. Planning of software releases and allocation of resources cannot be handled in isolation. The solution of this complex problem is of pivotal importance for project success. Without good release planning, critical features are not provided at the right time.

2. SYSTEM ANALYSIS

2.1 Release planning

A release planning meeting is used to create a release plan, which lays out the overall project. 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 project 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 project velocity to determine how many user stories can be completed. When planning by scope divide the total weeks of estimated user stories by the project 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.

2.2 Resource allocation

Assign the available resources in an economic way. It is part of resource management.

2.3 Strategic planning

In strategic planning, resource allocation is a plan for using available resources, for example human resources, especially in the near term, to achieve goals for the future. It is the process of allocating resources among the various projects or business units. The plan has two parts: Firstly, there is the basic allocation decision and secondly there are contingency mechanisms. The basic allocation decision is the choice of which items to fund in the plan, and what level of funding it should receive, and which to leave unfunded: the resources are allocated to some items, not to others. There are two contingency mechanisms. There is a priority ranking of items excluded from the plan, showing which items to fund if more resources should become available; and there is a priority ranking of some items included in the plan, showing which items should be sacrificed if total funding must be reduced.

2.4 Software project management

is a sub-discipline of project management in which software projects are planned, monitored and controlled.

2.5 Software development process

A software development process is concerned primarily with the production aspect of software development, as opposed to the technical aspect. These processes exist primarily for supporting the management of software development, and are generally skewed toward addressing business concerns. Requirements analysis is a term used to describe all the tasks that go into the instigation, scoping and definition of a new or altered computer system. Requirements analysis is an important part of the software engineering process; whereby business analysts or identify the needs or requirements of a client; having identified these requirements they are then in a position to design a solution. Risk management is the process of measuring or assessing risk and then developing strategies to manage the risk. In general, the strategies employed include transferring the risk to another party, avoiding the risk, reducing the negative effect of the risk, and accepting some or all of the consequences of a particular risk.

2.6 Project planning, monitoring and control

The purpose of project planning is to identify the scope of the project, estimate the work involved, and create a project schedule. Project planning begins with requirements that define the software to be developed. The project plan is then developed to describe the tasks that will lead to completion. The purpose of project monitoring and control is to keep the team and management up to date on the project's progress. If the project deviates from the plan, then the project manager can take action to correct the problem. Project monitoring and control involves status meetings to gather status from the team. When changes need to be made, change control is used to keep the products up to date.

2.7 Iterative and Incremental development

It is a cyclic software development process developed in response to the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interaction in between. The iterative and incremental development is an essential part of the Rational Unified Process, the Dynamic Systems Development Method, Extreme Programming and generally the agile software development frameworks

2.8 Iterative development

Iterative development slices the deliverable business value (system functionality) into iterations. In each iteration a slice of functionality is delivered through cross-discipline work, starting from the model/requirements through to the testing/deployment. The unified process groups iterations into phases: inception, elaboration, construction, and transition. Inception identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated.

Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements. Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements. Transition delivers the system into the production operating environment. Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full. The unmodified "waterfall model". Progress flows from the top to the bottom, like a waterfall.

2.9 Implementation guidelines

Guidelines that drive the implementation and analysis include:

Any difficulty in design, coding and testing a modification should signal the need for redesign or re-coding.

Modifications should fit easily into isolated and easy-to-find modules. If they do not, some redesign is needed.

Modifications to tables should be especially easy to make. If any table modification is not quickly and easily done, redesign is indicated.

Modifications should become easier to make as the iterations progress. If they are not, there is a basic problem such as a design flaw or a proliferation of patches.

Patches should normally be allowed to exist for only one or two iterations. Patches may be necessary to avoid redesigning during an implementation phase.

The existing implementation should be analysed frequently to determine how well it measures up to project goals.

Program analysis facilities should be used whenever available to aid in the analysis of partial implementations.

3. Motivation

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 project success. Without good release planning, critical features are not provided at the right time. This might result in dissatisfied customers, time and budget overruns, and decreased competitiveness in the marketplace. 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 project we examine details of the resources. Each feature is decompose into a sequence of tasks such as design, implementation, and testing. These development

tasks can be defined to an even more fine-grained level. The proposed system is a web based system. The administrator has full control on the system. He maintains the employee details in a database and divides the entire process in to number of tasks.

Each task is assigned to the employees according to their efficiency and also resources available. Task should be completed within the given time .The employees can view their tasks and do according to that.

5. Advantage

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.

As it is a web based project complexity is decreased.

6. Module description

6.1 Decomposition of feature

In this module, 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 other tasks can be considered here as well.

6.2Data Collection

We define the requirements of the release planning. This includes human and nonhuman resources that are required. We collect information regarding what are the resources that are available. Requirements include hardware and software .

6.3Computation of upper bound

In this module, we can generate an upper bound for the maximum value achievable. This can significantly reduce the computational effort and allows solution of problems of small and medium size. In this we assume an upper bound for the given feature like time required so that the task can be completed in time.

6.4 Optimization

In this we optimize the release planning. That means we divide the tasks into higher priority and lower priority tasks. So that resources can be allocated according to that. We can give more preference to the higher priority tasks. And also equal workload to all the employees.

6.5 Looking for resources

In addition to human resources, nonhuman resources are considered for each release as well. We assume M different types of nonhuman resources. This results in the capacity constraints.

6.6 Allocation of resources

This allows us to consider more or less skilled developers with a higher or lower productivity. Considering all the results such as human productivity, number of tasks, and available resources they are allocated to different developers to complete the task within given time.

7. 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 project. It is important to have a qualified starting point for these necessary adjustments caused by different changes in project parameters.

References

1. S. Acuña, 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 Shop Scheduling Problem: Conventional and New Solution Techniques," *European J. Operational Research*, vol. 93, no. 1, pp. 1-33