Ad Hoc Versus Systematic Planning of Software Releases – A Three-Staged Experiment

Gengshen Du, Jim McElroy, and Guenther Ruhe

Laboratory for Software Engineering Decision Support, University of Calgary
2500 University Drive NW, Calgary, AB, T2N 1N4, Canada
{dug, mcelroy, ruhe}@cpsc.ucalgary.ca

Abstract. Release planning addresses the process of deciding which requirement of an evolving software system should be assigned to which release. We study two fundamentally different software release planning approaches: (i) ad hoc planning and (ii) systematic planning. Ad hoc planning is mainly based on human intuition, experience and communication. Systematic planning, based on formalization, assumes a quantitative description of the problem, and application of optimization algorithms for its solution.

We have performed a controlled experiment intended to investigate hypotheses related to confidence, understanding, and trust related to the two approaches. The stated hypotheses were based on an explorative pre-study and prior industrial release planning projects. Although limited in scope and size, the experiment provided interesting insight into the performance of the stated approaches. Overall, systematic planning based on tool support increased confidence into the solutions and was trusted more than ad hoc planning.

Keywords: Release Planning Process, Controlled Experiment, Decision Support tool, Confidence, Understanding, Trust.

1 Introduction

Requirements engineering is a decision-rich problem-solving activity [2]. Software release planning is an important part of that activity when incremental software development is considered. In its simplest description, release planning is the process of assigning features or requirements to releases of a product. The functionalities of the product are additive, but it is important to offer the right features at the right time. The overall goal of release planning is to find the most promising release plans such that some stated objective, such as the degree of satisfaction for all the stakeholders or the overall business value, is maximized and available resource constraints are met.

Release decisions are complex, especially when considering problems involving several hundred features and a large number of widely diversified stakeholders. It becomes an even harder problem when resource estimates, resource capacities, and dependencies between features are taken into account. For a more detailed description of the problems and existing solution algorithms, we refer to [11].

There is a lack of evidence in evaluation of technologies in general [8], and of release planning in particular [3]. Addressing trust in solutions generated by algorithms

J. Münch and M. Vierimaa (Eds.): PROFES 2006, LNCS 4034, pp. 435 – 440, 2006.
© Springer-Verlag Berlin Heidelberg 2006

not easily understood by end-users, Lehtola et al. [7] conducted an empirical investigation to compare two well known methods of prioritizing requirements. The first one is based on pair-wise comparison of requirements, the other one is Wiegers' method [12] which is a cost-benefit method. One of the main findings was that participants mistrusted results they got from Wieger's method. So how can we achieve a higher degree of understanding, confidence and trust into proposed solutions on the user side?

We have performed a controlled experiment intended to investigate hypotheses related to confidence, understanding, and trust related to the two approaches. For that, we compared the (i) informal voting, manual and ad hoc generation of a release plans with (ii.1) the black box type of usage of an intelligent decision support (DSS-RP) system for release planning called ReleasePlanner to perform computer-based voting and generation of release plans. In addition, we have applied (ii.2) a white box usage of DSS-RP where the users were provided with both the results generated from the problem input and with further explanations and insights into the rationale of the proposed solutions resulting from performing two re-planning scenarios.

2 Software Release Planning in a Nutshell

Release planning in an ad hoc fashion focuses on human intuition, communication and human capabilities to decide which requirements should be selected to go into which releases. Physical meetings with stakeholders have to be arranged to elicit their priorities. Normally, this is hard to arrange. During the meetings, the expression of priorities is influenced by the persons attending. Based on that, a more or less accurate understanding of the real priorities is achieved. The actual planning using this understanding occurs on a manual basis, eventually including rounds of negotiations. This process can be supported by a list of the requirements to be released and/or story cards for a description of the requirements or use cases.

The systematic approach for release planning generates plans based on a formalization of the problem. This involves maximizing an objective function constituted from stakeholder ranking of the requirements based on urgency and/or value where different additional parameters for adaptation to the problem context are included. The actual optimization is further defined by a family of technological, resource and/or budget constraints. The result of this process is a set of five alternative solutions where each solution is at least 95% optimal. In addition, the solutions are maximally diversified among each other. For further details, we refer to [11].

ReleasePlanner is a decision support system (DSS-RP) that uses the above optimization approach as part of an evolutionary problem solving approach integrating human and computational intelligence. For industrial experience using the technology we refer to [1], [5], and [9].

3 Experimental Setup

The experiment was carried out at the University of Calgary, Canada. Nine Master's and PhD students majoring in software engineering participated. Further details can be found at http://sern.ucalgary.ca/~dug.