

University of Groningen

Design of a Methodology to Support Software Release Decisions

Sassenburg, J.A.

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2006

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Sassenburg, J. A. (2006). *Design of a Methodology to Support Software Release Decisions: Do the Numbers Really Matter?*. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

PREFACE

*“Research is to see what everybody else has seen and to think what nobody else has thought.”
-- Albert Szent-Gyorgyi --*

Software is a major worldwide industry. Software pervades a multitude of products and is an important corporate asset, with demand still increasing. Without software, most organizations could not survive in the current marketplace, and both the impact of software on everybody’s life and our dependence on software is rapidly increasing. Everyone uses software nowadays. We find software embedded, for example, in watches, coffee machines, cars, televisions, aeroplanes, phones, reservation systems, and medical equipment. It cannot, however, be denied that software engineering is still a discipline with much potential for improvement. Software projects are characterized by schedule and budget overruns, and the delivery of unreliable, and difficult to maintain, software products.

A relatively unexplored area in the field of software management is the implementation or release decision, deciding whether or not a software product can be transferred from its development phase to operational use. Many software manufacturers have difficulty in determining the ‘right’ moment to release their software products. It is a trade-off between an early release, to capture the benefits of an earlier market introduction, and the deferral of product release, to enhance functionality, or improve quality.

A release decision is a trade-off where, in theory, the objective is to maximize the economic value. Inputs into the release decision are expected cash inflows and outflows if the product is released. When should a product be released? What is the market window? What are the expectations of customers and end-users?

A release decision deals with the difficulty of verifying the correct implementation of functional and non-functional requirements. How much testing is needed? Software manufacturers are challenged to find the optimal level of information, as information has its price in cost and time.

The decision-making process to release a product will normally involve different stakeholders, who will not necessarily have the same preferences for the decision outcome. This means elements of individual and group behaviour are likely to surface when discussing a release decision to find a satisfactory consensus.

A decision is only considered successful if there is congruence between the expected outcome and the actual outcome, which sets requirements for decision implementation.

In this research project these different perspectives are reviewed, explored in-depth, both from a theoretical and from an empirical point of view, by studying practical examples. The results are used in a proposed methodology to improve strategic software release decisions, characterized by the existence of large prospective financial loss outcomes, including the presence of high costs for reversing a decision. Based on validation results in a practical setting, it is concluded that this methodology has a descriptive and a judgmental character, and can therefore support understanding, analysing, assessing and improving the capability of software manufacturers in this problematic area.

