University of Groningen

Design of a Methodology to Support Software Release Decisions

Sassenburg, J.A.

Publication date:
2006

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.

CONCLUSIONS AND RECOMMENDATIONS

208

12 CONCLUSIONS AND RECOMMENDATIONS

"The outcome of any serious research can only be to make
two questions grow where only one grew before."
-- Thorstein Veblen --

12.1 Introduction

The first Chapter of this thesis starts with a quotation from Brooks (1975), who writes: *'Complexity is the business we are in and complexity is what limits us'*. Or, as Beizer (1990) puts it: *'Software complexity grows to the limits of our ability to manage that complexity'*. This appears to be true. In this study, it is found that software product development involves many uncertainties, stemming from different sources. Examples are performance, schedule, development cost, technology, market and business. This appreciably complicates the predictability of software product development in general, and the market entry decision in particular. Market entry, or the release decision, is investigated in this study. For many software manufacturers, especially those operating in mass markets, this is the point of no return. A software release decision can be seen as a trade-off between early release to capture the benefits of an early market introduction and the deferral of product release, to enhance functionality and/or improve quality.

The objective of this study is to investigate how strategic software release decisions can be improved. A release decision is of strategic value when large prospective financial loss outcomes to the software manufacturer and/or the customers/end-users of the software are present. The study approach follows two main steps. 1.) Existing theory is studied and exploratory case studies are conducted. 2.) The results obtained are used to design a methodology with two basic requirements: a.) The methodology should enable a software manufacturer to understand the different aspects relevant to strategic software release decisions, and b.) It should offer the possibility of assessing its capability in this area, and act as an instrument to identify areas of possible improvement. The need to improve strategic software release decisions is addressed in Chapter 1; revealing the increasing impact of software on society, the increasing complexity of software products and the increasing importance of strategic software release decisions. In Part 1, Chapters 3 and 4, the results of the Exploration phase are described, with a study of existing theory and seven exploratory case studies. Based on the results obtained, the **Release Decision Methodology** for strategic software release decisions is designed in Part 2, Chapters 5 through 10. The methodology is based on three styles of decision-making behaviour, namely maximizing, optimizing and satisficing behaviour, complemented by relevant issues for decision implementation. Different process areas are identified in the **Release Decision Methodology**, and for each process area underlying practices are derived. Part 3, Chapter 11, describes the results of the Testing phase, in which a second series of case studies is conducted to validate the assumed descriptive and judgemental character of the methodology in a practical context.

In this Chapter, the primary and secondary research questions are revisited in Section 12.2, while the external validity of the methodology is discussed in Section 12.3. Next, the research philosophy, approach and strategy are reviewed in Section 12.4. Suggestions for further research are identified in Section 12.5, where after closing remarks on the subtitle of this thesis are made in Section 12.6.

12.2 Review of Research Questions and Results

In this Section, the answers to the four secondary research questions raised in Chapter 5 are summarized, and findings on the primary research question, as raised in Chapter 1, are given.

12.2.1 1st Secondary Research Question

How to model the market entry trade-off for a software product?

In Chapter 6, following the perspective on maximizing behaviour, the NPVI-method is presented, combining existing models from the semiconductor industry with a method for the comparative evaluation of software development strategies, based on NPV-calculations. The method is tailored to software release decisions but still has a general character, so its applicability is not limited only to software release decisions. The method presented can be used for the economic valuation of different release alternatives by determining the differences between a test strategy and a base strategy. The method can be extended, for example, by including flexibility, such as time-to-build and growth options, as another premium at the lowest level. Use of the method is not necessarily restricted to specific manufacturer types. Environmental and internal conditions under which a software manufacturer operates will determine the market, or demand, window and thus affect the expected cash inflows and outflows, while the underlying method is not influenced.

The NPVI-method receives little further attention in this study after its definition in Section 6.4. One or more software manufacturer environments willing to validate the NPVI-method were sought to apply the method in commercial environments with the presence of a high competition level and a medium to high number of potential customers. However, two manufacturers who showed initial interest were environments where the researcher was active as a consultant at that time. As discussed in Section 2.8, it was decided not to conduct research in software manufacturer organizations where the researcher was, or had been, recently active as a consultant. Further research toward the practical applicability of the NPVI-method is recommended (Section 12.5).

12.2.2 2nd Secondary Research Question

To what extent can an optimal level of information be determined
as input to the software release decision-making process?

In Chapter 7, the concept of maximizing behaviour is further investigated. Rather than assuming decision-makers possess all the relevant information for making choices, information itself should be regarded as an input that has a price in time and money. A decision-maker is confronted with the problem that, apart from determining the demand and supply functions, a trade-off between the costs and benefits of searching for information is required. This leads to optimizing behaviour instead of maximizing behaviour, however it is concluded that the difference is marginal, as the primary objective of a decision-maker remains profit maximization. The point of optimality where the maximal net asset value is reached, is influenced by many different factors [product characteristics, applied development process, technology used, available resources] and a software manufacturer is confronted with problems in determining this point accurately, one of them the problem of infinite regress. Other complicating factors are the lack of proven software reliability estimation models with a predictive character, and the overly complex relationships between all parameters involved. To be able to optimize, assumptions about reality have to be made, often leading to simplifications. In a practical context, software manufacturers will therefore look for a zone of cost

effectiveness instead of the point of optimality, as, in a practical setting, this point can, neither *ex ante*, nor *ex post*, be accurately determined.

The second series of case studies indicate that it is possible to identify a zone of cost effectiveness. In one case, the available information was close to the zone of cost effectiveness. In another case, some informants claimed that certain re-tests were redundant, as they did not increase the information level for the implemented functionality and stability of the product. In this case, an advantage to the project was the availability of relatively stable product requirements and the possibility of comparing test results with results from an existing product. This enabled strong statements about the reliability of the product, and increased the information level. However, in all cases studied, this zone could only be determined in qualitative terms, and not in quantitative terms. Further research toward mechanisms to determine this zone more accurately is recommended (Section 12.5).

12.2.3 3rd Secondary Research Question

> *What effects, stemming from individual and group behaviour,*
> *play a role in the software release decision-making process?*

In Chapter 8, conclusions are drawn on effects stemming from individual and group behaviour in the software release decision-making process. Intra-individual conflicts arise from the concept of bounded rationality and lead to *satisficing* behaviour. The negotiated decision-making strategy is best applicable for strategic software release decisions. A release decision matches well with the interacting group type. The characteristics of effective groups as described by McGregor (1960) are considered relevant. The proposed decision rule is consensus, as an important criterion is the need of support for successful implementation. On the political aspects of software release decisions, the different bases of power are mapped on the processes and strategies, as described by Stokman *et al.* (2000). It is argued that a high presence of 'management of meaning' and a low presence of 'challenge' and 'exchange' processes and strategies are favourable in strategic software release decisions. A high presence of 'management of meaning' processes and strategies implies that possible differences in positions, or aspiration levels, are reduced through the acceptance of convincing information. Several sources of conflict relevant for group decision-making are discussed, with two important for software release decisions, namely the inter-dependence between individuals or units, and the divergence of objectives. It is concluded that the availability of a commonly-shared product development strategy, accepted by all stakeholders involved, is an important factor in reducing the likelihood of the presence of these sources of conflict.

The issues identified are implicitly integrated in the methodology as practices. As such, they are addressed during the second series of case studies, with a focus on the decision-making theory of Stokman *et al.* (2000), also applicable to the decision-making process of strategic software release decisions. In one case, inter-dependence between two stakeholders led to a controversial issue and one stakeholder was therefore left out of the decision-making process. The responsibility for decision implementation was therefore transferred to a temporary taskforce, as the stakeholder removed did not want to implement the decision made. The decision-making process itself was characterized by the presence of 'challenge' processes and strategies due to insufficient information. In the other two cases, the decision-making process was characterized by a high presence of 'management of meaning' processes and strategies due to information close to, or within, the zone of cost effectiveness. Other validated characteristics of the decision-making process are a negotiated decision-making strategy, an interacting group type, with consensus as the decision rule.

12.2.4 4th Secondary Research Question

Which issues are important to increase the likelihood of a successful implementation of the software release decision?

On the implementation aspects of software release decisions the following conclusions were drawn in Chapter 9. Decision implementation is an important factor contributing to overall decision success. Important issues identified for decision implementation are the acknowledgement that a budget upfront should be reserved for corrective actions, and that the implementation of the decision should be carefully monitored to ensure the actual decision outcome is close to the expected outcome before the project team is discharged and product responsibility is transferred [single-loop learning]. The decision appraisal, after the official discharge of development responsibilities and transfer of product responsibility, is a pre-requisite for any software manufacturer wanting to improve its capabilities through organizational learning [double-loop learning]. The appraisal should not limit itself to either the decision-making process or the decision outcome, but include both issues, and the entire project history. By identifying strengths, weaknesses and possible areas for improvement, the organization's memory can be augmented for use in future projects.

The issues identified were all integrated into the methodology as practices. As such, they were validated during the second series of case studies. The reservation of a maintenance budget and monitoring the product rollout are confirmed as important practices where, in two cases, minor to major corrective actions were needed to eliminate discrepancies between the expected and actual decision outcome. Only in one case, was the project officially discharged and appraised, although no evidence was found that the results of the appraisal were archived for future use. This is a definite shortcoming as 'good projects have a memory'. In this case where most practices were successfully implemented, a formal appraisal of the project would have given an instrument to identify the best practices. A consideration here could have been to promote them to the standard development process, thus enabling future projects to implement them and improve the manufacturer's capability.

12.2.5 Primary Research Question

In this study, the dimensions of the problem of deciding when to release a software product are investigated using the existing body of knowledge, complemented by a study of current practices in software manufacturer organizations. The results obtained lead to the following definition of a strategic software decision:

> *A strategic software release decision is the act of choice function in a collective, managerial decision-making process, deciding to transfer a software product from its development phase to operational use with the existence of large prospective financial loss outcomes to the software manufacturer and/or its customers, including the presence of high costs to reverse the software release decision once it is made. It is a non-routine decision and will normally have a long-term horizon.*

The results are combined with theories from different disciplines and used to propose a methodology to structure software release decisions. The methodology, consisting of four process areas with underlying practices, was validated in three case studies in a practical context, with this methodology having a descriptive character in the environments studied. When the information level in the decision-making process increases, thereby reducing uncertainty, the pre-release cash outflows [development cost] will increase due to continued testing, and the post-release cash outflows are likely to decrease [more stable product]. From a decision-making perspective, the presence of 'management of meaning' processes and strategies will increase, and the presence of 'challenge' processes and strategies will decrease.

It is validated in the same three case studies that the methodology, with its underlying practices, offers the possibility of assessing the success of a strategic software release decision in the environments studied and identifying which improvements lead to increased decision success. Especially in the first case studied, successful implementation of the first two process areas of the methodology could have prevented the organization from releasing a product so difficult to maintain that the decision was later made to start re-developing the whole product. The methodology offers an instrument for increasing the success of strategic software release decisions.

If all practices are successfully implemented, the quality of the decision outcome and the quality of decision implementation will be within an optimal zone. Below this zone, uncertainty will be high and is likely to lead to challenges among stakeholders, whereas, above this zone, the costs of additional information are likely to be economically unjustifiable. Placement within the optimal zone is expected to increase the likelihood that the decision will result in the attainment of objectives. This judgmental character of the methodology enables a software manufacturer to pro-actively aim for decision success by implementing the formulated practices.

It is concluded that the primary research question, as formulated in Chapter 1 of this thesis, *'How to improve strategic software release decisions?'* is satisfactorily answered.

12.3 External Validity of the Methodology

The methodology is validated in similar software manufacturer environments where software products are developed for internal use. A review of the methodology however indicates no practices specific to software manufacturers. The characteristics of a specific software manufacturer influence the product development strategy and market entry strategy chosen, but not the methodology. The product specific market and product requirements influence the required information level, but not the methodology. All practices derived need proper implementation, independent of the characteristics of the software manufacturer type, and the extent to which the practices are implemented follows the specific circumstances. No major obstacles therefore exist to applying the methodology successfully in either similar or other software manufacturer environments.

The methodology is designed for strategic software release decisions, a requirement stated in Section 1.6, which is the reason to adopt Harrison's (1987) Process Model, instead of the Organizational Model, as the reference model. The methodology ensures all relevant stakeholders, representing different perspectives, become actively involved before a project is started [in the proposal phase] and stay involved until the product released functions well in its operational environment. This multi-perspective approach enables the sharing of knowledge among stakeholders, and where problems arise all perspectives are represented to evaluate an alternative course of action. When strategic value is not as important, there is less need for a formal decision-making process, and this does not necessarily require the involvement of higher management. When considering this methodology for more routine release decisions, it should be carefully considered, for each practice, if its implementation gives sufficient added value and whether it requires the involvement of higher management levels. Taking this into account, the methodology can also be used for software release decisions without strategic value.

Using the methodology is not necessarily restricted to software release decisions only. Although the methodology is designed for strategic software release decisions, it has a general nature, in focusing on the decision-making process, extending it by defining, and controlling, the decision objectives, the definition and collection process of information as input to the decision-making process, and the implementation and evaluation of the release decision. These

are common aspects of decision-making and its usage for other product development decisions can be considered, with a possible revision of the underlying practices.

A review of the methodology indicates no practices specific to software. However, the lack of transparency on purpose and quality of software on one hand, and the verification and validation problems with software on the other hand, are two sources of uncertainty and are strong arguments to adopt a methodology especially in cases where the release decision is of strategic value. In other product development engineering disciplines both uncertainty and strategic decision value can be present as well, especially in cases where new products are developed and introduced to new, or existing, markets. This indicates the methodology can be of interest beyond the scope of software product development. Further research is needed in this area to be able to make stronger statements (see Section 12.5).

12.4 Research Philosophy, Approach and Strategy

In this Section the research philosophy, research approach and research strategy followed are discussed, with considerations as follow:

1. *Research Philosophy*. As described in Section 2.3, it is not possible to classify this study as either positivistic/realistic or interpretivistic. The study incorporates characteristics of both philosophies. A structured approach was followed throughout this study, looking for facts and causes of behaviour. This is a characteristic of a positivistic philosophy. On the other hand, it was confirmed in the case studies, that it is difficult to obtain reliable quantitative data for developed products, and the processes followed, as there is general lack of agreement on software measurement and a lack of maturity of measurement in software engineering. It was, for example, often difficult, if not impossible, to retrieve the initially-planned schedule and pre-release cash outflows, or the actual schedule and pre-release cash outflows, and an indication of the post-release cash outflows needed for corrective maintenance.

2. *Research Approach*. The research approach is classified as inductive (see Section 2.4), as a methodology is designed on the basis of the results obtained in the exploratory case studies, while facts were collected in a qualitative way as opposed to a quantitative way (Saunders *et al.* 2003, pp.86-87). The lack of maturity of measurement in software engineering would have made it difficult to acquire the number of samples, of sufficient size, required to generalize conclusions (Saunders 2003, *et al.* p.87). However, some conclusions are drawn about the external validity of the methodology, as summarized in Section 12.3. Having followed an inductive approach, the designed methodology is not seen as the only possible explanation for describing, and assessing, strategic software release decisions. It is likely this methodology can be further extended and refined.

3. *Research Strategy*. Case studies as a research strategy offer a good opportunity to study software release decisions in a practical context; its primary criterion being the lack of research results on the study phenomenon. A major requirement is the use of multiple sources of evidence; in this study interviews, questionnaires and documentation. The questionnaires returned were often not fully completed or different informants interpreted questions differently. The interviews and documentation were used to complete information and to remove misinterpretations. The interviews are considered especially important. By creating an atmosphere of mutual trust valuable information was obtained, especially on the positions of different stakeholders in the decision-making process. In case of surveys as an alternative research strategy, it is questioned whether this information would have been obtained. In addition, as mentioned before, questionnaires returned were often not fully completed or different informants interpreted questions differently. The case study criteria used were successful in the sense that results of all case studies conducted could be used. The case studies have been very time-consuming. Several further organizations initially showed an interest in participating in this study,

but retracted after exploring discussions. Most participating organizations had to be visited more than once: preparation of the case study, interviews that could not always be planned for the same period (causing increased travelling), and approval and presentation of the results. What was the added value of the case studies in this study? In the first place, a review of the methodology reveals that many derived practices were based on the case study results of the Exploration phase (see process areas *'Release Definition'* and *'Release Information'*). Secondly, the case study results have supported the researcher when designing the methodology by having at his disposal examples from a practical setting.

12.5 Directions for Further Research

The multi-disciplinary approach to strategic software release decisions presented is an attempt to describe strategic software release decisions and to offer an instrument for identifying areas for improvement. It offers room for further research in this area of interest. During the study, eight issues of particular interest are identified for further investigation, as described hereunder.

1. *Further research on the completeness of the methodology.* In Section 11.5, the completeness of the methodology is questioned, using case J as a reference. In this case all relevant practices for release decision success were satisfactorily implemented. The feedback session with the involved project members and a review of the methodology did not reveal additional elements. It is however concluded that due to the limited number of case studies, additional research is needed in similar and other software manufacturer environments to further explore the completeness of the methodology.

2. *Application of the NPVI-method in a commercial environment.* One of the initial objectives of this study was to demonstrate the possibility of defining an economic model illustrating the evaluation and comparison of different release alternatives. For this reason, the NPVI-method was defined. As explained in Section 12.2, no environments were found to validate the usability of this method in a practical context. Further research could be conducted to determine its potential benefits in a practical context, and this may possibly lead to further extensions and/or refinements of the method, for example, by integrating time-to-build and growth options (see Section 6.6).

3. *Mechanisms to more accurately determine the zone of cost effectiveness for information perfection.* In this study, the zone of cost effectiveness is introduced for information perfection (Chapter 7). Below the zone of cost effectiveness, marginal value still outweighs marginal cost, whereas above the zone of cost effectiveness marginal cost outweighs marginal value. The concept of a zone of cost effectiveness is qualitatively used in this study, but further quantitative research in this area can be beneficial for software manufacturers, as this will help them make better trade-offs facing a release decisions: continue testing or release the product? This is a difficult issue to investigate, however one could focus, for example on the measures needed for evaluating reliability and maintainability dependent on associated risk levels [An extension on the work of Rae and Robert (1995)].

4. *Factors that influence the aspiration levels of stakeholders identified.* The presence of uncertainty due to imperfect information and cognitive limitations on one hand and the presence of sources of conflict on the other hand might give rise to different aspiration levels for stakeholders involved in a release decision, as discussed in Chapter 8. In this study, some time was spent on trying to identify factors, which influence the different aspiration levels of the stakeholders identified. However, as this was not explicitly addressed in the exploratory case studies, insufficient evidence was available to make any strong statements. Further research in this area may give more insight into potential sources for aspiration level differences, which might facilitate the reduction, or even elimination, of these differences.

5. *Models to predict the pre-release level of reliability and the post-release maintenance effort*. In this study, it has been repeatedly stressed that software manufacturers are confronted with serious problems when trying to report the pre-release level of product reliability obtained and the expected post-release maintenance cost, based on the level of reliability and the maintainability of the resulting product. The applicability of existing theory is limited (Chapter 3), and the exploratory case studies confirm this to be a problem area (Chapter 4). This hampers the determination of the zone of cost effectiveness, especially for larger and more complex software products. This problem area has been known for decades, but no solution has been proposed that has found wide acceptance. The 'traditional' development methods are not able to cope with this, possibly implying that the release trade-off question will become more difficult in the near future due to increasing uncertainty. It might be worthwhile, although ambitious, to pursue research in the area of totally new development approaches, eliminating, or at least reducing, this uncertainty level and moving in he decision-making process from complete uncertainty to informed uncertainty.

6. *Research into the possible correlation between decision characterization of software release decisions and the process maturity level*. In Section 4.4.3 the question is raised whether a correlation between decision type and process maturity exists. Will higher process maturity change a non-routine decision type to a routine decision type? Motivation might be the improved control of the development process. For example, the availability of historical data should make much stronger predictions about the reliability level possible, through the use of software reliability prediction models like COQUALMO (Chulani 1999) and Orthogonal Defect Classification (Chillarege *et al.* 1992) and approaches like Bayesian nets (Fenton and Neil 2001) and Goal Structuring Notation (Kelly 1998), see Section 3.3.2, thus reducing uncertainties in the decision-making process. The exploratory case studies gave no answer to this question, but it would be interesting to investigate whether increased process capability leads to this phenomenon. If so, it would reveal a strong argument for investing in process improvement. A related aspect is how higher process maturity influences the zone of cost effectiveness. In Section 10.3, it is argued that higher maturity would lead to a less uncertain and less expensive zone of cost effectiveness, and decision-making, increasingly based on sharing convincing information. This might be interesting to further investigate as well, as it also builds a strong case for process improvement investments.

7. *Applicability of the designed methodology to other engineering domains*. Discussing the external validity of the designed methodology, the question is raised whether the validity of the methodology might go beyond the context of software development. This question has remained unanswered. It could be worth pursuing this question further to find a satisfactory answer. Further research could identify the extent to which other engineering disciplines could benefit from this methodology, and possibly identify important differences.

8. *Integration with theories covering other life-cycle stages*. In the life-cycle approach to software products, theories and supporting methods, methodologies and techniques are developed to optimize the efficient and effective development and maintenance of software products throughout the life-cycle; from strategic planning, through development, through operations and through to end-of-life decisions (Berghout 2002). Notable recent publications are:

   - On strategic planning:

     'Evaluation of information system proposals: design of a decision support method' (Berghout 1997);

   - On development:

     'Product focused software process improvement' (Solingen 2000);

   - On operations:

'The Alignment of Operational ICT' (Klompé 2003).

A direction for future research might be to integrate the different theories into one coherent theory covering the entire life-cycle, which could be performed in a step-wise approach, for example, by first studying the relationship of this proposed methodology to other theories.

12.6 Closing Remark: Do the Numbers Really Matter?

The subtitle of this thesis is *'Do the numbers really matter?'* Of course they matter. Identifying the numbers and developing the methods to interpret them should be a prominent issue for Information Systems (IS) and Software Engineering (SE) research. In general, software manufacturers will only invest in the development of new software products when the expected net present value of their investments is positive. This is true, both for software manufacturers, selling their developed software products to external customers, and for software manufacturers investing in software products [information technology], to improve the efficiency and/or effectiveness of their internal organization.

However, in a practical context, the determination of the optimal release time from a quantitative, financial perspective is difficult, if not almost impossible, due to complete uncertainty, as opposed to certainty, or informed uncertainty (Section 4.4.3). Sources of this uncertainty are threefold.

- ❖ The state of the art in software engineering technology is such that building software components and products in a predictable way with predictable behaviour is uncommon. Although new innovations may be, or become, available, their application in the software industry is severely limited at this stage.
- ❖ Information has its price in time and cost, forcing decision-makers to make a trade-off between search costs and opportunity costs.
- ❖ Decision-makers simplify the real world, as they cannot escape the diverse psychological forces that influence individual behaviour. Combined with the potential presence of sources of conflict, this may lead to the situation where different stakeholders experience difference aspiration levels. As such, satisficing behaviour where decision-makers try to find consensus and choose a satisfactory release alternative is a good characterisation of the software release decision-making process found in the environments studied.

The focus of this study is to address software release decisions from a quantitative, economic perspective on one hand and from a qualitative, decision-making perspective on the other hand. Increased attention to numbers, by gathering valid information [including historical data] to compare, and evaluate, different release alternatives [for example, the presented NPVI-method] and sharing the results among decision-makers is important to reduce uncertainty levels to a more acceptable level, so differences in aspiration levels of stakeholders involved in the decision-making process, are reduced, or eliminated, through convincing information. This is an important contribution to reducing uncertainty, and thus minimizing situations where people lives are put at risk, especially for software products where reliability, safety and security are important non-functional requirements.

The results of this study are a contribution towards improving the capability of software manufacturers, by offering a methodology that helps understand, assess and improve their software release decision-making processes. However, successful adoption of the methodology requires that software manufacturers reach the zone of cost effectiveness for the perfection of information; a zone where numbers make business sense, and can be convincingly used to support informed decision-making. It is likely that uncertainty will increase due to ever-increasing software size, and the absence of substantial improvements in defect potentials and

removal efficiencies. Without the availability, and successful adoption, of ways to significantly improve software productivity and software quality, it will become more complex for software manufacturers to attain release decision success. In such a situation it is likely that the release decision-making process will be dominated by a high presence of 'challenge' processes and strategies and that the numbers will be increasingly less complete and less reliable: they still matter but have less value and will probably be ignored, leading to intuitive decision-making instead of informed decision-making.