Paper presented at the European SEPG, June 17th, 2003 (London)

"When can the software be released?"

Software product development from an economic perspective

Abstract
In this paper, a conceptual model is presented, offering a framework to steer software product development from an economic perspective. Through a series of mini-cases, in various organisations, it was found that the determination of a satisfactory moment to release a software product is a problem that most organisations struggle with. In the first place, the business case as the initial rationale for a project stays insufficiently aligned with the actual status of the project as it progresses. Secondly, at crucial decision points, there are many uncertainties and comparing and evaluating the different alternatives is performed limited by the time available. Finally, there is a lack of proper evaluation of the business case results and the correctness of the various decisions made. Together, these prevent organisations from improving their capabilities in this area. The conceptual model presented combines existing methods and techniques into an overall framework. It helps organisations to build an economic case for their projects and control the progress of these projects. It supports the decision-making process to determine a satisfactory moment to release a software product.

Keywords
Software economics, business case, decision-making, release criteria, time-to-market, net present value, real option, estimation methods, software architecture evaluation, defect growth models, reliability, maintainability.

1. Introduction

Technicians are not economists and economists are not technicians. At secondary school, this difference already starts to surface. Future technicians concentrate on technical subjects and are only mildly interested in economic subjects. They might learn something about macroeconomic issues, but in few cases is knowledge gained in the field of business economics. Future economists chose different subjects, and mathematics is only included when really necessary to go to university. At universities the gap is further widened. Technical studies pay only minor attention to economics and economic studies do not investigate the area of product development as it takes place at the operational level in organisations. In industry, this gap avenges itself. Technicians start working in projects and experience themselves the issues relevant at the operational level. Economists joining industry find themselves busy at strategic and tactical level and spend their time on issues such as market investigations and product management. Economists in the role of product managers and technicians in the role of project leaders or software architects speak different languages and operate in different worlds. It prevents organisations from selecting and controlling projects in a way such that they add maximum value to the organisation. It might be well true that this gap is one of the most important reasons for software project failures. Today's challenge is to focus continuously on the realisation of business benefits, which can only be accomplished by developing software products from an economic perspective. In other words: bridge the gap.

In July 2002 a research project in this field started. The first phase of this project was an orientation towards current practices in software industry. Apart from a thorough literature study (theoretical approach), mini-cases were conducted within seven Dutch and Swiss organisations (empirical approach). These case studies investigated the way products were developed and released with special attention to the definition, the deployment and the evaluation of release criteria and on the final release decision. In the next (current) research phase a conceptual method is being specified (to be concluded in August 2003). This method aims to support the important decision points during software product development in such a way, that a satisfactory moment for releasing the product can be derived which meets the formulated business case. Currently, preparations are being taken to validate the specified method in industry through action research (experimental approach). This validation will take place in the third phase of the project. The method will be validated in two different organisations, after which the overall conclusions of the research project can be drawn.

This article reports the results obtained so far. First, the research area will be explored to get a better understanding of the issues related to software product release (paragraph 2). Thereafter, the results of the mini-cases will be discussed together with the introduction of a reference model for managerial decision-making (paragraph 3). These paragraphs summarise the orientation phase of the research project. The results are used to specify a conceptual method that can be used to steer software product development from an economic perspective (paragraph 4). This paper ends with the conclusions so far, including a description of open issues to be further investigated.

2. First exploration of the research area

In this paragraph the research area is explored. Attention is given to the possible strategic goals of a software supplier. Further some possible tradeoffs between release criteria are discussed.

Strategic Supplier Goals
Grady describes three possible strategic goals of software suppliers [GRA 1992]:

- Maximize customer satisfaction. This is accomplished mainly by offering products, which will both satisfy and delight customers. Other factors are important as well, for instance the price of the product and the required service level and maintenance efforts and costs from a customer's point of view. Maximizing customer satisfaction for a project means that essential ***product needs*** must be identified and implemented.
- Minimize engineering effort and schedule. Improving productivity is important, as it will help to decrease development costs (from which a customer may also benefit). Shortening development times will help to deliver products faster, which can be a highly competitive advantage in today's marketplace (and can offer another benefit to a customer). Minimizing engineering effort and schedule for a project means that work must be performed efficiently to reduce costs and that ***time to market*** must be minimized.
- Minimize defects. Minimizing defects during development will limit the amount of rework. This will also have a positive impact on minimizing engineering effort and schedule. Further, post-release costs will probably decrease as the product contains fewer defects. The customer will also benefit from reliable products and thus minimizing defects will have a positive contribution to customer satisfaction. Minimizing defects for a project means that product must be developed with a high ***reliability***, which can be accomplished by applying appraisal methods like reviews and inspections.

Since the early nineties many supplier have initiated process improvement programs to improve these strategic capabilities. There is however little evidence, that conformance to process standards guarantees good products, which meet customers' demands [KIT 1996]. However, this criticism may be unfair as popular process models (CMM, CMMI-SE/SW) and standards (ISO 9001:2001, ISO 15504) also insist that process improvement enhances product quality [KIT 1996].

When developing products, not all goals are equally important for a project. It depends on the lifetime of a product, which goal has highest importance. It can be seen from Figure 1 that the priorities of the goals shift during the evolvement of a product.

Market description	Introduction	Early Adopters	Mainstream	Late Majority	End of Life
Buyer profile	Technology enthusiasts	Visionaries	Pragmatists	Conservatives	Skeptics
Importance	1. Time to market 2. Product Needs 3. Reliability	1. Time to market 2. Product Needs 3. Reliability	1. Reliability 2. Time to market 3. Product Needs	1. Reliability 2. Product Needs 3. Time to market	1. Reliability 2. Product Needs 3. Time to market

Figure 1: Project priorities as a function of a product's lifecycle [MOO 1995].

Costs and time to market versus quality

Developing software products is normally characterized by business pressure to minimize costs and time to market. It is often stated that delivering a higher quality product does not necessarily mean that development costs will increase. This is only partially true. For instance, striving for higher reliability and maintainability through investing in appraisal techniques like reviews and inspections will be paid back up by a decrease in the repair costs of finding and fixing defects. There is however an optimal level (see Figure 2). Beyond this point a further increase in appraisal costs will not have a net positive effect (for sake of completeness, one should also take into account the effects on post-release costs):

delta (appraisal costs) + delta (repair costs) < 0

Whether a further increase in appraisal needs is justified or not will depend on the specific circumstances, for example market characteristics. In some cases, reliability is a top priority and one cannot afford to deliver below a certain level.

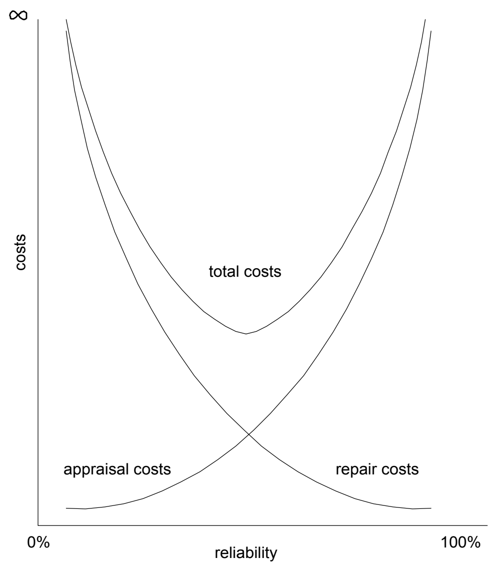

Figure 2: Appraisal costs versus repair costs.

Time to market is also influenced by the phase in the product's lifetime as well as other characteristics of a market, for example the level of competition. Figure 1 depicts the release priorities of a software product as they evolve through a product's lifetime. Figure 3 depicts some examples of profit models related to time to market. When, for instance, the entry of a new product is delayed in a market with heavy competition, the probability of a supplier capturing the advantages of "Early Adopters" will decrease.

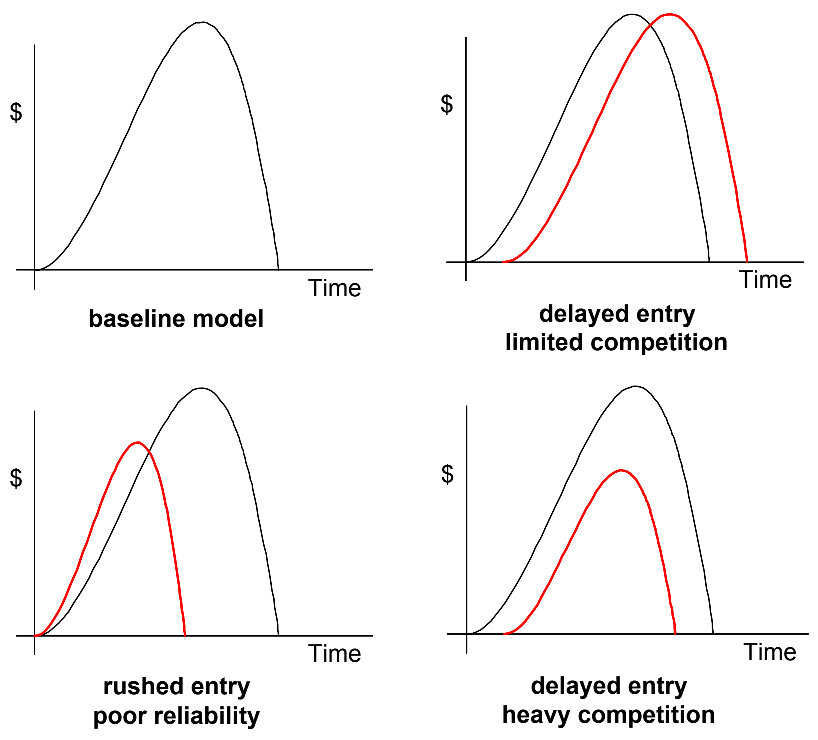

Figure 3: Examples of profit models.

Reliability and Maintainability
Although reliability and maintainability are, in most models and standards, defined as a non-functional product needs, explicit attention is usually given to them. The reason for this lies in the fact that reliability is considered important in nearly every software product and the impact of not meeting reliability requirements may be dramatic on users and their environments. Further, both reliability and maintainability determine, to a great extent, the short-term and long-term post-release or operational costs. And with respect to the total life-cycle costs of a software product, these costs are in general a multiple of the original development costs: ratios of 70:30 have been reported [NOS 1990]. The difficulty is to determine how these non-functional product needs can be deployed and evaluated during product development. The ISO 9126 standard [ISO 1991] for instance defines quality attributes and related indicators. However, there is no description of how the lowest level metrics can be used to evaluate non-functional product needs at a higher level during product development. Instead, methods are developed that combine qualitative and quantitative information.

With respect to reliability, software defect prediction models have been developed since the seventies. Fenton and Neil have studied the most widely accepted models. They identified severe problems such as [FEN 1999]:

- There is no distinction made in different notions of 'defect'.
- Statistical methods are often flawed.
- Product size is wrongly assumed to be a causal factor for defects.
- Obvious causal factors are not taken into account.
- Black box models hide crucial assumptions.
- The models cannot handle uncertainty.

They conclude that as a result these models provide little support for determining the reliability of a software product. Their study also showed that the number of pre-release faults is not a good indicator of the number of post-release faults. The problem is that many software suppliers use the pre-release fault counts as a measure for the number of post-release faults, e.g. the reliability of the released product.

These research outcomes combined with a further investigation led to the conclusion by Fenton and others that Bayesian nets offer a model that takes into account the crucial concepts missing from classical approaches [FEN 1998].

A Bayesian net is a graphical network (see Figure 4) together with an associated set of probability tables. The nodes in the net represent uncertain variables and the arcs in the net represent causal/relevance relationships between the variables. Classical prediction methods do not take these relationships into account, but focus on correlation between variables (for instance size and defects). The probability tables for each node provide the probabilities of each state of the variable of that node. For nodes without parents these are just the marginal probabilities while for nodes with parents these are conditional probabilities for each combination of parent state values [AGE 2002].

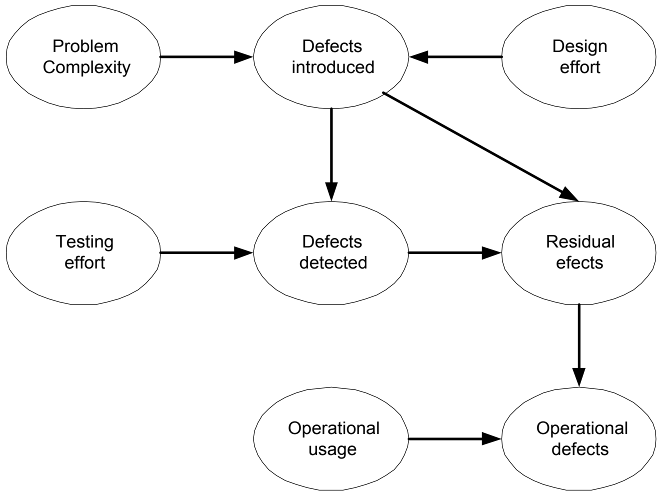

Figure 4: Example of Bayesian Net for defects [AGE 2002].

Once a Bayesian net has been set up, evidence about variables (as soon as available) can be entered. All the probabilities will be updated accordingly, offering valuable information concerning variables we are interested in predicting.

With respect to maintainability, there are even fewer methods and techniques available to assess its value during product development. Instead, classic design rules like maximal cohesion, minimal coupling and information hiding are often used. They are assumed to contribute implicitly to a high level of maintainability [PAR 1979]. Further research in this area is conducted [BOS 2000b].

Conceptual Economic Model

The one and only appropriate measure a software supplier would place on the decision to release a product or not is the profit difference. Suppose a software supplier produces a product to be sold at a price ***p***. Profits are revenues (price ***p*** times quantity ***q***) minus costs (pre-release or development costs, post-release or operational costs):

supplier profits = revenues – costs = p . q – pre-release costs – post-release costs

(In this simplified model other costs such as production costs and sales costs are not taken into account)

In order to be able to predict profits the following sub-questions related to expected revenues and costs must be answered:

- Which product needs have been implemented and tested?
- What are the current levels of reliability and maintainability compared to their target values?
- What are estimated sales figures (price, quantity, reputation) when the product is released now?
- What are estimated post-release costs when the product is released now?

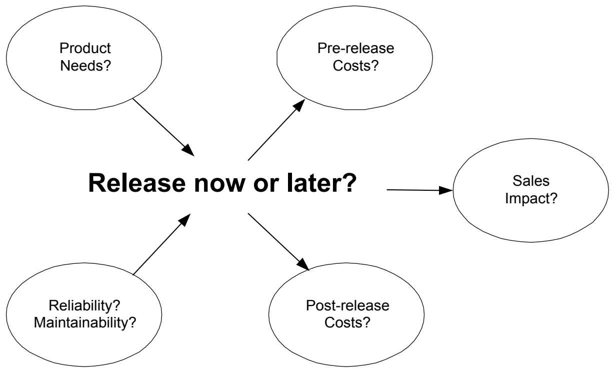

Figure 5: Decision factors for product releasing (supplier perspective).

As a second step, other situations must be considered. Delaying the time to market will have several consequences depending on the phase in the product's lifetime and the characteristics of a market as discussed before. Extending the development time will bring extra product features and higher reliability that may have a positive effect on price and quantity, but the impact may be negative as well in a highly competitive market.

supplier profits` = p` . q` – pre-release costs` – post-release costs`

In order to be able to predict profits in these situations the following sub-questions related to expected revenues and costs must be answered:
Sub-questions to be answered are:

- What are the additional pre-release costs to improve and extend product needs?
- What are the additional pre-release costs to improve the reliability and maintainability?
- What will be the impact on sales (price, quantity, reputation) when market introduction is delayed?
- What will be the impact on post-release costs when reliability and maintainability are improved?

With this information, the profit difference of the supplier can be calculated:

delta (supplier profits) = supplier profits – supplier profits`

In practice, these questions remain unanswered in most cases. There is no analytical approach to model the supplier profits as a function of functional product needs and non-functional product needs such as reliability and maintainability. A combination of the following non-analytical methods is used to decide when a software product is "good enough" to release [RTI 2002]:

- A "sufficient" percentage of test cases run successfully.
- Statistics are gathered about what code has been exercised during the execution of a test suite.
- Defects are classified and numbers and trends are analysed.
- Real users conduct beta testing and report problems that are analysed.

- Developers analyse the number of reported problems in a certain period of time. When the number stabilizes or remains below a certain threshold, the software is considered “good enough”.

There may even be cases, however, in which suppliers ship the product when there is no budget or schedule left for further testing and repair activities.

Answering the questions raised concerning product features, reliability, maintainability, costs and sales impact is on the other hand not a task that can be easily accomplished. It requires a mature development process to define, deploy and evaluate the release criteria. Further, all stakeholders within the supplier organisation will have to work closely together with the development team. Product management must play a central role by defining and managing the entire lifecycle of each product. Marketing or sales must be involved to deliver inputs with respect to sales impact analyses and profit models. The group or department responsible for maintenance of released products must be involved to define maintenance criteria as part of release criteria and give input with respect to post-release costs.

A conceptual economic model could also incorporate the profits of the end-user(s) of a product.

delta (economic welfare) = delta (supplier profits) + delta (end-user profits)

What are the profit differences for the end-user in case the product features are extended or the reliability is improved, e.g. when releasing the product is delayed? This will be difficult to determine in most cases as the end-user’s profile heavily depends on product and market characteristics.

3. Results of mini-cases

In this paragraph the results of the conducted mini-cases are presented. First, the reference model derived from a general decision-making model will be described.

Decision-making

What is a decision-making process? Figure 6 shows both the interrelatedness of six functions and their sequential organisation [HAR 1987, page 40]:

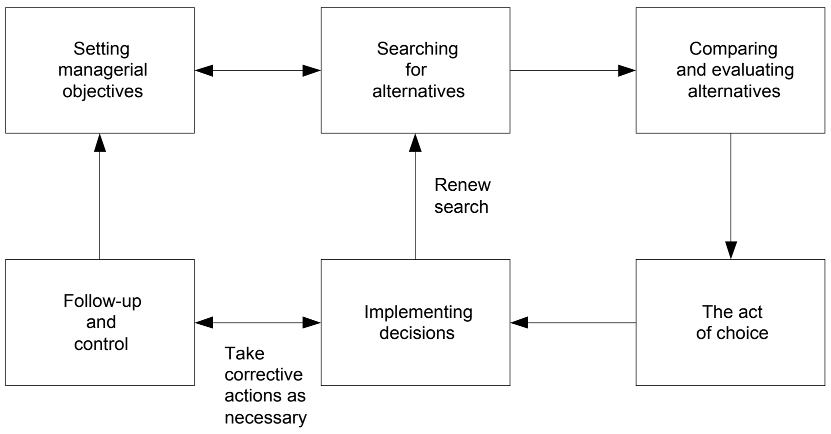
Setting managerial objectives
Searching for alternatives
Comparing and evaluating alternatives
Renew search
Follow-up and control
Take corrective actions as necessary
Implementing decisions
The act of choice

Figure 6: The decision-making process.

The functions of decision-making are [HAR 1987, pp. 38-39]:

1. *Setting managerial objectives*. The decision-making process starts with the setting of objectives, and a given cycle within the process culminates upon reaching the

objectives that gave rise to it. The next complete cycle begins with the setting of new objectives.

2. *Searching for alternatives*. In the decision-making process, search involves scanning the internal and external environments of the organisation for information. Relevant information is formulated into alternatives that seem likely to fulfil the objectives.
3. *Comparing and evaluating alternatives*. Alternatives represent various courses of action that singly or in combination may help attain the objectives. By formal and informal means alternatives are compared based on the certainty or uncertainty of cause-and-effect relationships and the preferences of the decision maker for various probabilistic outcomes.
4. *The act of choice*. Choice is a moment in the ongoing process of decision-making when the decision maker chooses a given course of action from among a set of alternatives.
5. *Implementing the decision*. Implementation causes the chosen course of action to be carried out within the organisation. It is that moment in the total decision-making process when the choice is transformed from an abstraction into an operational reality.
6. *Follow-up and control*. This function is intended to ensure that the implemented decision results in an outcome that is in keeping with the objectives that gave rise to the total cycle of functions within the decision-making process.

Derived reference model

In Figure 7, a reference model is presented that places the product development activities of a supplier organisation in a broader perspective. Senior Management at a strategic level defines a *product road map*, describing the long-term expectations with respect to business and technology developments. Business developments are addressed in terms of changes in the marketplace and the organisation. Technology developments are addressed in terms of adoption of new technologies and new application of existing technologies. The product road map is the input for Product Management at a tactical level to derive *business cases*. A business case is used to define the rationale for a project that is initiated to develop a product (either a new product or a newer version of an existing product). It describes the expected revenue for the supplier organisation taking into account the expected development or pre-release costs (to develop the product) and operational or post-release costs (to produce, deploy and maintain the product).

The business case defines the *external product needs and constraints* as input to a project at operational level. The external product needs describe the required functionality seen from the perspective of the customer(s). Distinction can be made into functional needs and non-functional needs. The functional needs describe the functionality that must be offered by the product. The non-functional needs define product properties and put the constraints upon the functional needs. They determine the behaviour of a product. Examples are: reliability, safety and accuracy. There are often referred to as quality attributes. In the non-functional needs, the compliance to external standards is included in addition. Constraints determine the boundaries of a project and may, for instance, be limitations with respect to budget and lead-time of the project and cost price of the final product.

Internal stakeholders define *internal product needs and constraints*. The internal product needs are also expressed in functional and non-functional needs. Functional needs describe for instance the documentation that is needed to produce, deploy and maintain the resulting product. Non-functional needs describe for instance the compliance to internal standards.

The combination of the *external product needs* and constraints and the *internal product needs* and constraints define the scope of a project. Release criteria can be directly derived from them in global terms. They are defined as *the particular criteria of a project and its resulting products that are taken into account to make the decision whether or not to release the product*. During the project the product is developed, during which time the defined release criteria are used to select the best design alternative (meeting the needs and constraints) and deployed afterwards to lower-level process and product attributes. Suppose that lead-time and budget are constraints then this will also put constraints on each component as defined in the product design. Suppose that reliability and maintainability are part of the non-functional needs, they will have to be deployed in some way to the defined components. It may not always be possible to conduct a simple mathematical breakdown of a non-functional need. In that case implementation rules may be defined that will implicitly contribute in meeting the non-functional need at product level.

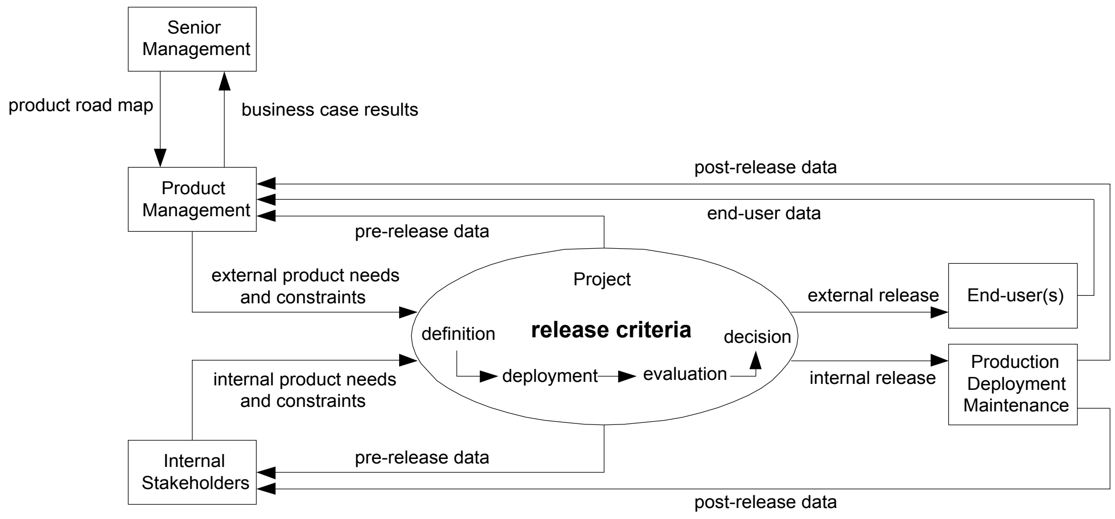

Figure 7: Reference model.

During development the project must stay aligned with both Product Management and the internal stakeholders. The status of the project is obtained by evaluating the defined and deployed release criteria. Currently measured values and predictions of final values form the *pre-release data*. A steering committee may be in place to discuss the *pre-release data*, combined with any new insights. For instance, the business case may have been changed due to market developments or the service department may come up with additional product needs.

The continuous alignment of the status of the project with the status of the *external product needs and constraints* and the *internal product needs and constraints* will finally lead to the situation where the release decision can be made. Release alternatives to be considered are:

- Do not release the product and cancel the project.
- Release the product later.
- Release the product now.

In the following paragraph, more attention is paid to this release decision, addressing it from an economic perspective. A distinction is made between externally releasing the product to its intended end-user(s) and internally releasing the product to the internal stakeholders for production, deployment and maintenance activities.

After the product has been released, assuming that the project is not cancelled, data must be gathered to determine the result of the business case. A distinction is made between *end-user data* (for instance the revenues of the product and the customer satisfaction) and *post-release data* (for instance the costs for corrective maintenance). Evaluation of these data might result in changes to the *product road map* and future business cases, as well as removal of organisational process deficiencies.

This conceptual model will be easily recognised by industrial companies, developing commercial products for external customers (business-to-business or business-to-consumer). A valid question here is whether this model can also be applied in those organisations, where an IT-department develops IT-solutions to support the own internal administrative organisation (as occurs for instance in banks, insurance companies, government). There will of course be differences. Examples are:

- There are no external end-users, but internal end-users.
- The level of competition is different (in fact, there might be no competitors at all).
- It will be harder to express a business case in financial terms (profits).
- The introduction of the product will affect the organisational processes within the own organisation.

On the other hand, the main principles stay the same. Product management is directed by a long-term strategy (the product road map) and there must be a clear business case as the rationale for each project. A release decision is to be made based upon the evaluated release criteria and after the product has been released externally and internally, the result of the business case can be determined in quantitative or qualitative terms.

Results of mini-cases

Mini-cases have been conducted in seven Dutch and Swiss organisations. The selected organisations were both industrial organisations (B2B markets with competition) and administrative organisations such as a bank and an insurance company (internal products for support of their administrative processes). Selection criteria regarding the maturity of the product development process (especially at strategic/tactical level, but also at operational level) have been high in order to reveal as much possible relevant information. The reference model as described in the previous paragraph was used during interviews with 3-6 people in each organisation. Further, available process descriptions and project documentation were studied. Finally, all organisations filled in a questionnaire with questions related to the characteristics of the organisation, its products and the market.

The main conclusions of these case studies were the following:

- ***There was no strong alignment between the business case and the project during the execution of the project***. All organisations used a business case as the rationale for the selected projects, stating both the expected costs and benefits. [1] However during the project, Product Management and the project failed to report to each other explicitly about the current status of the business case (new insights) and the current status of the project (progress so far and estimates to completion).
- ***Alternatives were not explicitly compared and evaluated against the business case***. Implicitly this happened in most cases, however at crucial decision moments (project definition, product design) no evidence was found why one alternative was selected above another alternative, using criteria derived from the business case. Available methods and techniques for comparison and evaluation were, in most cases, known but not used.
- ***The final release decision was made without being able to accurately estimate operational costs***. In all cases, reliability and maintainability were defined as important non-functional product needs as they determine to a great extent the operational costs after product release. High reliability reduces corrective maintenance effort and high maintainability reduces both corrective maintenance effort and adaptive/perfective maintenance effort. In all cases, these non-functional needs were not deployed to lower level components as identified in the selected

[1] In two cases, the benefits were not stated as these projects delivered an IT infrastructure. In one case this infrastructure was used by other projects, in the other case clients paid an annual fee for using the infrastructure. In both cases it was felt that it was hard or even impossible to allot revenues to a specific product release.

product design or software architecture. It was only during testing that much effort was spent on meeting a high level of reliability. However, in all reliability and maintainability were not expressed in financial terms.

- ***There was no strong feedback loop in the product development cycle***. After product release, there were no specific actions undertaken to evaluate the result of the business case as a whole and the results of the implemented decisions at crucial moment during development (project definition, product design, product release). In only one case there was a plan to evaluate the business case at predefined moments after product release by the business project leader, who was assigned the responsibility for the investments made. In all cases, there was no system in place to analyse the defects found after product release and use the results to remove process deficiencies in product development.

In Figure 8, the results are illustrated in the used reference model.

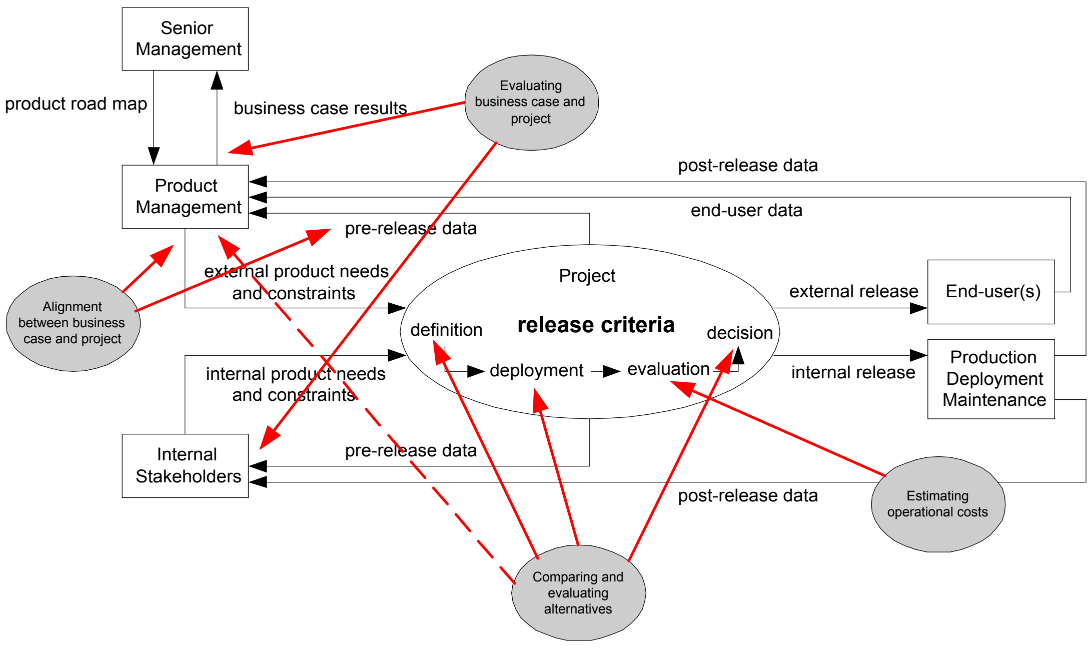

Figure 8: Results of the case studies.

4. Towards a Conceptual Method

Interdisciplinary framework

This research project aims to specify a decision-support method to find a satisfactory economic moment to release a software product. One could have expected here *optimal* instead of *satisfactory*. The basic difference is that in the optimal situation the decision maker is assumed to have all the alternatives against which to apply the release criteria, whereas in the satisfactory situation the decision maker merely applies the release criteria to any minimally satisfactory alternative that is good enough to meet the objective. This is especially relevant for those situations where multiple stakeholders have to make the decision in an open environment where not all variables are known. The method will concentrate on the disciplines *Economics and Statistics* and *Mathematics*. It would, however, be naïve to eliminate the effects that originate from other disciplines as depicted in Figure 9.

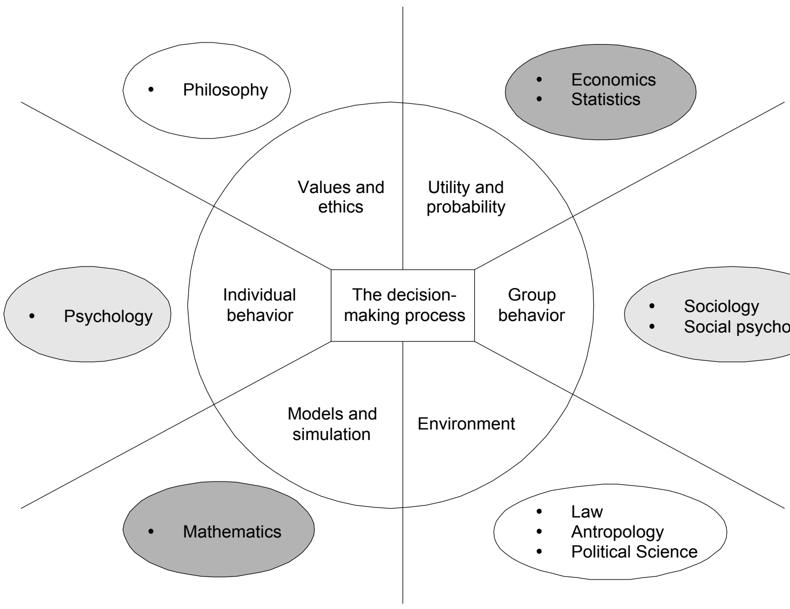

Figure 9: Interdisciplinary framework for decision-making.

The method to be specified must take into account the effects from the disciplines *Psychology*, *Sociology* and *Social Psychology*, addressing individual and group behaviour. These effects must either be recognised and be made explicit or measures must be specified to reduce or even eliminate their effects.

The central research question has become:

How to specify a method that can be used to determine a satisfactory economic moment to release a software product, assuming that a release decision is an investment activity and taking into account the effects of individual and group behaviour?

Product releasing as an investment activity

In the research project, it is argued that a release decision is an investment activity. The business case is in fact reviewed again. The difference is that pre-release costs or investments ***I*** have been made, development time ***T*** has elapsed and after the release the assets ***C*** will be generated to be reduced by the post-release costs or operational costs ***M*** (Figure 10).

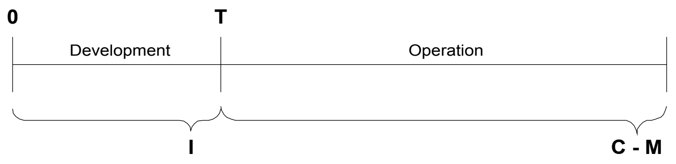

Figure 10: Simplified economic model.

The Net Present Value can be calculated as:

$$NPV = (C - M) / (1 + d)^T - I \qquad \text{(Net Present Value)}$$

The NPV-method is the most widely accepted traditional discounted cash flow method. During recent years, there has been a tendency to apply option-pricing theory to information technology investments, referred to as the real option approach ([BOE 2000] and [SUL 1999]). Important reasons for the choice are that the NPV-method is static and does not incorporate management flexibility to stop a project for instance (staged investment or time-to-build option). Further, the NPV-method does not include the possibility to start second-stage projects (build option). However, there is also criticism of the real option approach. It is very complex and still leaves management with the difficult task to estimate input parameters. Extending NPV with Decision Tree Analysis offers the possibility to incorporate the time-to-build option [RIB 1997].

When a release decision is an investment activity, it can also be argued that preceding steps in product development are preliminary release decisions, in which release criteria are defined, deployed and evaluated. Differences are that the final release decision will be the point of no return where the product is transferred from the development phase to the operational phase, and that economic and technical uncertainties existing at the start of product development have been reduced or even eliminated. In Figure 11 four steps are distinguished, three preceding steps and the final step to release the software product (staged investment).

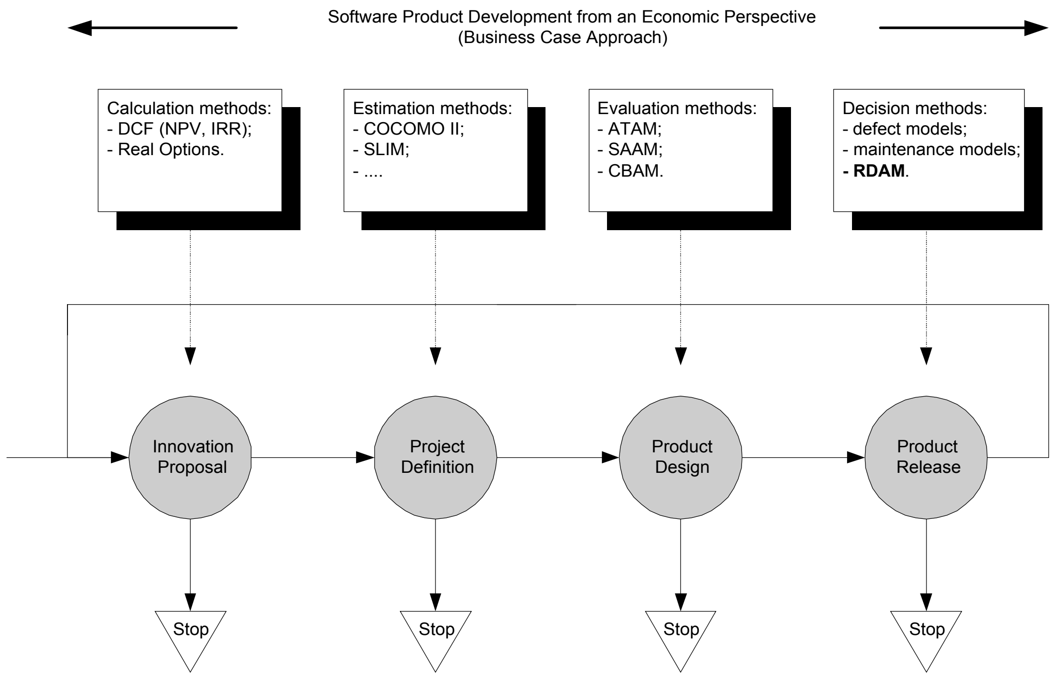

Figure 11: Product development steps (staged investment) and possible supporting methods.

The main characteristics of each step are:

- ***Innovation Proposal (Why build the product?)***. The first step is building the business case for a new product or another version of an existing product [REI 2002]. In this step, the external release criteria are defined in global terms. Examples of these criteria could be: functionality, quality, time-to-market, pre-release or development costs, post-release or operational costs (corrective, adaptive and perfective maintenance) and compliance to external standards. Calculation methods such as the traditional discounted cash flow [ERD 1999] and the newer real option approach [SUL 1999] may be used here to build the case.
- ***Project Definition (How to build the right product?)***. The second step is defining the boundaries of the project, taking into account both the external release criteria (from the business case) and combining them with the internal release criteria, like compliance to internal standards and additional quality criteria. Project estimation methods like COCOMO II [BOE 2001] and SLIM Estimate [PUT 1992] may be used here to make the optimal trade-off between functionality, quality, lead-time and costs. Different project alternatives may be evaluated with multiple stakeholders using the Win-Win Negotiation Model [HIN 2001]. This step to make the Project Definition might lead to changes in the business case as better insights are gained.

- ***Product Design (How to build the product right?)***. The third step is to evaluate different design or architecture alternatives. Important criteria are both the defined quality criteria (such as reliability, maintainability, see [ISO 1991]) and the expected costs. Supporting methods here are for instance ATAM [KAZ 1998], SAAM [DeS 1995] and CBAM [ASU 2001]. Another good reference here is [BOS 2000a]. This step may again lead to changes in the business case.
- ***Product Release (When to stop building the product?)***. After the product has been implemented, tests will be started to test to which degree the functionality and quality have been implemented correctly. At the same time, relevant information is gathered to support the final release decision. No supporting methods have been found other than defect prediction models to make quantitative or qualitative statements about reliability ([CHI 1992], [LYU 1996]) and assessment models [BOS 2000b] to quantify maintainability.

The research project primarily focuses on the last process step, which is the specification of the decision-support method *RDAM* (Project Acronym: *Release Decision Analysis Method*), but it will prescribe the other three preceding steps with respect to the persons and knowledge to be involved, possible cognitive constraints that may play a role in each step, the supporting methods that can be used in each process step and the preconditions in applying the method.

Refined Decision-Making Process
How do the product development steps, as described in this paragraph, correspond to the business case driven approach to the functions in the decision-making process as described in the paragraph 3? This has been depicted in Figure 12. The function *Setting managerial objectives* in the sense of software product development is in fact defining the business case.

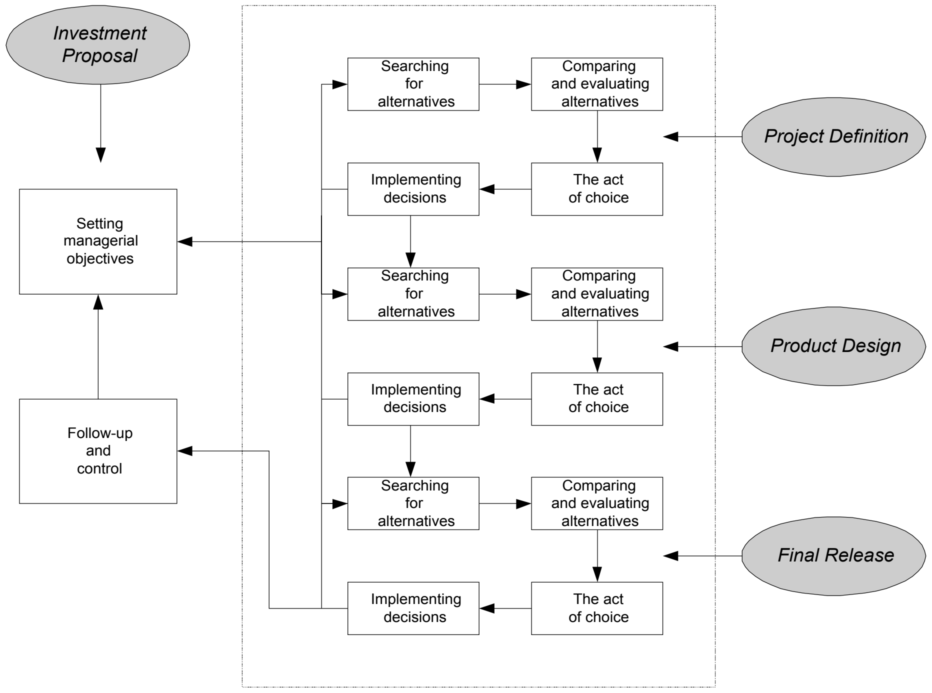

Figure 12: Relationship between decision-making process and product development steps.[2]

The next four functions *Searching for alternatives -> Comparing and evaluating alternatives -> The act of choice -> Implementing decisions* are in fact repeated three times: when defining

[2] In order to reduce the complexity of this figure, no attention has been given to the fact that during *Setting Managerial Objectives* and *The act of choice* (three times) one has the possibility to stop product development (time-to-build option).

the project, when selecting the product design and when making the final release decision. Note here, that when searching for alternatives in each step, it may be decided to review the defined managerial objectives, or in other words adjust the business case.

5. Conclusions

In this article the context of a business case driven approach to software product development has been described, focusing on adding economic value to an organisation. The results of the exploration of literature and the mini-cases are currently used to specify a framework in which existing methods and techniques are brought together. The objective of this framework is to support decision-making at crucial moments during product development, using the business case as the reference to compare and evaluate alternatives.

The method is not ready yet. At this stage the following issues are still open and need to be further investigated and discussed:

- What is a software product? Basically, four types of products can be distinguished:
  a. Commercial products being sold to external customers (business-to-business or business-to-consumer markets).
  b. Products to save time and money by automating manual labour.
  c. Products to be used for building other products (like an IT infrastructure).
  d. Products to improvement the organisational effectiveness by doing tasks completely different to better achieve the wanted results.

  Should the method be applicable to all types of products (eventually in different forms) or should the restriction be made to concentrate primarily on for instance commercial products?
- How must the method deal with the opposed effects from other disciplines like Psychology, Sociology and Social Psychology? How can these effects be recognised? How can they be reduced or eliminated?
- What is the best model of choice at each product development step? Should it be a strictly financial analysis or are non-financial issues to be included as well? If so, do they have to be quantified some way? Does it imply, that the economic/financial decision is transformed into a decision to be supported by a multi-criteria analysis method?

Further research on these issues is currently undertaken.

About the author

Hans Sassenburg (*hsassenburg@se-cure.ch*) received a Master of Science degree in Electrical Engineering from the Eindhoven University of Technology in 1986 (The Netherlands). During his study years he founded a one-person consulting firm and worked as an independent consultant. In 1996 he co-founded a new consulting and training firm (Alert Automation Services b.v.), which was sold in 2000. From 1996 till 2001 he worked as a guest lecturer and assistant professor at the Eindhoven University of Technology. In 2001 he moved to Switzerland where he founded a consulting firm (SE-CURE AG, *www.se-cure.ch*). Having finished his first book *"Software Engineering: van ambacht naar professie"* (ISBN 90-72194-64-0), he started a Ph.D. research at the Faculty of Economics at the University of Groningen (The Netherlands) under the supervision of Prof. Egon Berghout (*e.w.berghout@eco.rug.nl*).

References

[AGE 2002] *"Software Metrics and Risks"*, Technical Report, Agena Limited, 11 February.

[ASU 2001] *"A Foundation for the Economic Analysis of Software Architectures"*, J.Asundi, R.Kazman, Third International Workshop on Economics-driven Software Engineering Research, Toronto (Canada).

[BOE 2000] *"Software Economics: A Roadmap"*, B.W.Boehm, K.J.Sullivan, ACM Press.

[BOE 2001] *"Software Cost Estimation with COCOMO II"*, B.W.Boehm et al, Prentice-Hall.

[BOS 2000a] *"Design and Use of Software Architectures"*, J.Bosch, Addison-Wesley.

[BOS 2000b] *"Assessing Optimal Software Architecture Maintainability"*, J.Bosch et al., fifth European Conference on Software Maintainability and Reengineering.

[CHI 1992] *"Orthogonal Defect Classification – A Concept for In-Process Measurements"* R.Chillarege et al, IEEE Transactions on Software Engineering, Vol.18, No.11.

[ERD 1999] *"Comparative evaluation of software development strategies based on Net Present Value"*, H.Erdogmus, First International Workshop on Economics-driven Software Engineering Research, Toronto (Canada).

[FEN 1998] *"Assessing Dependability of Safety Critical Systems using Diverse Evidence"*, N.Fenton et al, IEE Proceedings Software Engineering, 145(1), pp. 35-39.

[FEN 1999] *"A Critique of Software Defect Prediction Research"*, N.Fenton and M.Neil, IEEE Transactions on Software Engineering, Vol. 25, No. 5.

[GRA 1992] *"Practical Software Metrics for Project Management and Process Improvement"*, R.B.Grady, Prentice Hall, pp. 22-24.

[HAR 1987] *"The Managerial Decision-Making Process"*, E.F.Harrison, Houghton Mifflin Company.

[HIN 2001] *"A Requirements Negotiation Model Based on Multi-Criteria Analysis"*, H.In et al, International Symposium on Requirements Engineering, Toronto (Canada).

[ISO 1991] *"ISO/IEC 9126: Information Technology - Software Product Evaluation - Quality characteristics and guidelines for their use"*, ISO.

[KIT 1996] *"Software Quality: The Elusive Target"*, B.Kitchenham, S.L.Pfleeger, IEEE Software, Vol. 13, No. 1, pp. 12-21.

[LYU 1996] *"Handbook of Software Reliability Engineering"*, M.R.Lyu, McGraw-Hill.

[MOO 1995] *"Crossing the Chasm"*, G.Moore, Harperbusiness, New York.

[NOS 1990] *"Software Maintenance Management: changes in the last decade"*, J.T.Nosek, P.Palvia, Software Maintenance: Research and Practice (2).

[PAR 1997] *"Designing Software for Ease of Extension and Contraction"*, D.L.Parnas, IEEE Transactions on Software Engineering, March, pp. 128-137.

[PUT 1992] *"Measures for Excellence: Reliable Software On Time Within Budget"*, L.H.Putnam, W.Myers, Yourdon Press Computing Series.

[REI 2002] *"Making the Software Business Case"*, D.Reifer, Addison-Wesley.

[RIB 1997] *"Option pricing for IT valuation: a dead end"*, P.M.A.Ribbers et al, in E.W.Berghout, & D.S.J.Remeny, (Eds.), Evaluation of Information Technology, (pp. 67-75) Delft University Press.

[RTI 2002] *"The Economic Impacts of Inadequate Infrastructure for Software Testing"*, Planning Report 02-3, Prepared by RTI for National Institute of Standards and Technology, U.S. Department of Commerce.

[SUL 1999] *"Software Design as an Investment Activity: A Real Options Perspective"*, K.J.Sullivan et al, Real Options and Business Strategy: Applications to Decision Making, L.Trigeorgis, ed., (London, England: Risk Books), pp. 215-261.