Combining Models for Business Decisions and Software Development

[1]Christina Wallin, [2]Stig Larsson, [3]Fredrik Ekdahl, [1]Ivica Crnkovic
[1]Mälardalen University, Department of Computer Engineering, Västerås, Sweden
{christina.wallin, ivica.crnkovic}@mdh.se
[2]ABB Automation Technology Products AB, Västerås, Sweden, stig.bm.larsson@se.abb.com
[3]ABB Corporate Research, Västerås, fredrik.ekdahl@se.abb.com

Abstract

Today there is a number of established software development lifecycle models (SDLMs) supporting software development. Correct implementation of these models helps develop software products the right way, but this does not ensure that the right products are developed. Successful product development companies often use business decision models (BDMs) to facilitate the selection of products and projects for investment, but these models do not necessarily facilitate actual development of the software. One of the current challenges in the software community is to combine BDMs and SDLMs, including mapping of business decision gates and major lifecycle milestones. This is needed to achieve synergies between the two model types and to support the development the right products the right way, as well as to gain control over company investments. This paper analyzes two BDMs, proposes mappings to an established SDLM, and describes experiences of using them in a large, multinational engineering company.

Keywords

Business decision, software product development, gate, major milestone, software development lifecycle

1. Introduction

Development of software products is typically done in projects, using different SDLMs. Used properly, these enable the projects to deliver products with expected quality and functionality, on time and on budget. This is however not enough for a commercial success.

It is at least equally important that the right product development projects are selected and that sound *business decisions* govern the projects until the product is launched. Progression from one development lifecycle phase to another should be based on a deliberate business decision, verifying that the development project and the product are still feasible to the development organization and the market. Organizations successful in developing new products typically use some kind of BDM to achieve this [1].

Using both a BDM and a SDLM can however introduce problems in the interaction between business decisions and software development. As there are different models (both for business decisions and software development) it might not be obvious how these models interface and how they should be synchronized. Another problem is that some software developing organizations, introducing a BDM, experience that it forces them to use a particular type of SDLM, which might not be perceived as appropriate from the development point of view.

Consequently, mappings of SDLMs and BDMs are needed for clarity and to fully realize the synergies between these two model types.

This paper aims to demonstrate the concept of mapping BDMs and SDLMs. The outline of the paper is as follows: Section 2 describes two BDMs, the Stage-Gate™ Model [1] and the ABB Gate Model for Product Development [3]. Section 3 gives a short overview of software development lifecycle phases and major milestones as defined in the Unified Process [2]. Section 4 discusses the mapping of business decision- and software development lifecycle models by introducing pre-gate milestones. Finally the last section is a summary of experiences, and concludes with plans for future work.

2. Business Decision Models

A business decision should be based on the results from evaluating the market and competitors, the technology feasibility, the business strategy, intellectual property rights, product quality and status, and available resources

within the organization. Today, many organizations use a well-defined model for preparing and making business decisions in product development projects. Such BDMs generally consist of a number of different development stages separated by business decision gates. Cooper's *Stage-Gate™* model [1][4] is a good example of a BDM.

2.1 Cooper's Stage-Gate™

Cooper's Stage-Gate™ model, shown in Figure 1, divides the development project lifecycle into six stages, separated by five gates. Each stage consists of a set of parallel activities and processes performed by different actors within an organization. Each activity in each stage is designed to gather information needed as input to the upcoming business decision gate and to reduce risks associated with the creation or evolution of a product.

The first stage of Cooper's model is the *discovery* stage. It begins with an idea for a new product or product version. During the *scoping* stage, the main objectives are to assess market and technology and identify the key product requirements. During the *business case* stage, information needed to decide if it is feasible to develop the product is gathered. The *development* stage mainly deals with the development of the product according to the product and project definitions. In stage four, *testing and validation*, the product is finally verified and validated, and the final stage, the *launch* stage, includes activities for marketing and sales, and for production or operation.

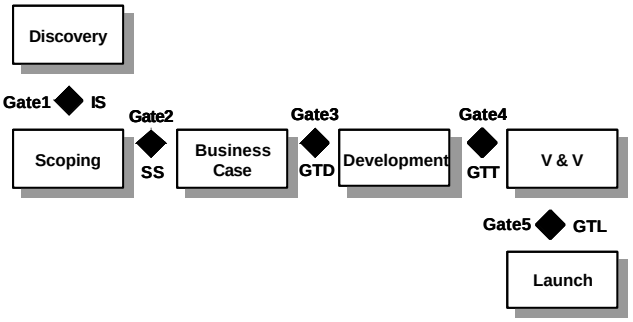

Figure 1. Cooper's Stage-Gate™ Model

The gates between each product development stage represent distinct business decision points. The procedures at each gate are similar; the results from the activities performed in the stage preceding the gate, together with a decision criteria checklist are used as input to the business decision. The output from the gate is a go/no-go/hold/redo decision accompanied by relevant plans for the next stage.

Cooper's Stage-Gate™ model uses five gates: Gate 1, *idea screen* (IS), is the first occasion where resources are committed to the project. Gate 2, *second screen* (SS), is essentially a repetition of Gate 1, although more rigorous and based on the information gathered during the scoping stage. Gate 3, *go to development* (GTD), represents the last chance to stop the project before significant investments are made. A go decision at this point represents both a financial and resource commitment and an agreement on the product and project definition established during the build-business-case stage. Gate 4, *go to testing* (GTT), is based on a post-development assessment to make sure that the product and project are still attractive to the market and to the organization. Finally Gate 5, *go to launch* (GTL), is the last point at which the project can be stopped and the product cancelled.

Cooper's Stage-Gate™ Model is widely accepted in development and manufacturing industry. Many companies use this or slightly modified versions of the model, and experience reports from the companies are positive [5][6][7]. One of the advantages of a BDM is the permanent awareness of the business goals and the presence of clear alternatives for the decisions. Another advantage is that the commitment of funding for a project is low at the start and increases as the project progresses when the stakeholders become more confident that the project will ultimately be successful. A difficulty with using BDMs is the significant requirements on the information needed for the business decisions already early in the project. If the development lifecycle model used is not well defined and accurately synchronised with the BDM, there is a high risk that the information used for the decision is not well prepared. Another consequence of poor synchronisation of the models is the risk of spending unnecessary effort and time to produce information that is neither needed nor a natural result of the activities conducted so far in the project.

2.2 The ABB Gate Model

The ABB Gate Model for Product Development [3] defines eight gates where major business decisions are made. The model serves as a framework for the various activities, e.g. software development, hardware development, competitor management, intellectual property management, training and marketing, etc. included in a product development project.

There are several reasons why ABB developed its own model. The primary reason is that the model has grown from the company's own experience and to some extent from shared experience with other companies (such as the PROPS [8] model from Ericsson). This approach gives a good tuning of the model to specific company needs, but requires extra effort for development and maintenance, and increases the risk of ending up with a model that is not in line with general trends or applicable de-facto standards.

The ABB Gate Model does not explicitly define any stages. It is implicitly assumed that the selected SDLM

includes the phases or stages needed. The results from the development activities provide information used as input to the gates. This way the ABB Gate Model is not used as an independent and self-sustained process but it is closely related to the development processes. When using a BDM as an independent process, there is an apparent risk for a mismatch with the development process. Experiences from ABB show that such a mismatch (in strategy, in goals, in procedures and techniques) can lead to misunderstandings between the developers and management, to decisions based on faulty information, to decisions made on the wrong level, or even to decisions not being made at all. On the other hand, good synchronization between the models increases both decision and development quality. For this reasons it is crucial that the BDM is distinguished from, but tightly coupled with the SDLM. One way to achieve this is to utilize gates as the decision points, and refrain from specifying lifecycle stages as activities separate from the decision process.

2.2.1 Gate Model Roles

There are two specific roles defined in the ABB Gate Model, the *Gate Owner* and the *Gate Assessor*.

The Gate Owner is the person, or group of persons, that have the responsibility and authority to decide if a product should be developed or not, and if a development project should run or not. The Gate Owner is also responsible for the funding of the development project and for the availability of required resources. Generally this is the product or customer responsible.

The task of the Gate Assessor is, on behalf of the Gate Owner, to evaluate the product and the project before a gate, to produce a gate assessment report, to suggest a gate decision, and to present the assessment result at a gate meeting. The Gate Owner appoints the Gate Assessor and it is recommended that the Gate Assessor is external to the development project, to be able to be as objective as possible. This is a demanding role, as the Gate Assessor needs to be both experienced and competent, trusted by the Gate Owner and respected by the organization.

2.2.2 Gate Procedure

The gate procedure in itself is fairly simple and consists of only two activities, the gate assessment and the gate meeting. Input to the gate assessment is documents prepared by the project, interviews with project stakeholders and a gate assessment checklist. The assessment is done over an extended period, typically a calendar week, and involves both the project manager and the gate assessor. The output from the assessment is a report (a slide presentation) addressing the checklist items for the current gate. The assessment report together with other relevant material is made available to the gate meeting participants prior to the gate meeting to give them time to prepare. At the gate meeting, a go/no-go/hold/redo decision is made for the project. After the gate meeting the decision and any identified actions are communicated to all stakeholders.

2.2.3 Gates

The first six gates (Gate 0 to Gate 5) are decision points used to determine whether or not the development project should continue. Gate 6 and 7 are used to close the development project and to capture experience.

Input to Gate 0, *Start Project* (SP), is a feasibility study report or a project proposal including analysis of the market, competitors, intellectual properties, product strategy, risks, needed resources and required technology.

At Gate 1, *Start Project Planning* (SPP), the development project scope should be defined in terms of functions, features and quality as well as business constraints such as for example time to market, in such detail that it can be used for planning.

At Gate 2, *Start Execution* (SE), the project should be planned in terms of specified requirements, effort, time and cost estimates, procedures for quality assurance, risk management, configuration management and so on.

At Gate 3, *Confirm Execution* (CE), all major risks should be addresses and all technical solutions proposed.

At Gate 4, *Product Introduction* (PI), all functions and features should be implemented and the product should be ready for *Beta*, or acceptance-test and marketing.

At Gate 5, *Product Release* (PR), the product should be ready for release to the market or customer.

At Gate 6, *Close Project* (CP), the development project should be closed and product hand-over to manufacturing and/or service and maintenance should be confirmed.

At Gate 7, *Retrospective Investigation of Project* (RIP), an evaluation of the project and product should be done to evaluate its business success.

3. Software Development Lifecycle Models

The software development lifecycle can be divided into a number of phases, generally 3-5, indicating the main focus of the development work at that time; investigating the scope, constructing the software or deploying the results. There is almost as many names for these phases as there are software-developing organizations, but in the context of this paper the lifecycle phases defined in the Unified Process (UP) [2] will be used. According to UP, each software development lifecycle consists of four phases:

inception, *elaboration*, *construction* and *transition*. Each phase can then be further subdivided into steps (iterations), see Figure 2.

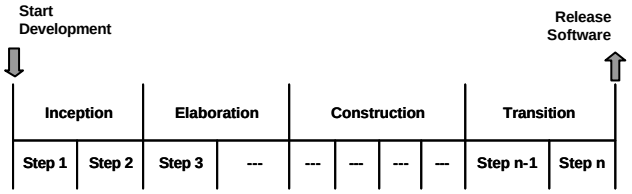

Figure 2. Unified Process Development Lifecycle Phases

The progression from one phase to the next is made when passing a major milestone. A milestone is defined as a scheduled event that marks the completion of one or more important tasks and it is used to measure achievements and development progress. At a milestone, a predefined set of deliverables should have reached a predefined state to enable a review.

In his article "Anchoring the Software Process" [9] Barry Boehm describes the three critical milestones essential for successful management of system development:

Life Cycle Objectives (LCO) - stakeholders' agreement on the system's top-level objectives such as: system boundaries, operational concept, system requirements, system and software architecture, and development lifecycle plan and feasibility rationale. The primary goal of the inception phase, which is concluded with the LCO milestone, is to set the technical scope of the software, outline the architecture, identify critical risks, and build a proof-of-concept prototype.

Life Cycle Architecture (LCA) - stakeholders' agreement on the system's elaborated objectives, especially the system and software architecture, and complemented with a risk assessment and risk management plan. The goal of the elaboration phase, concluded with the LCA milestone, is a stable software architecture, identified significant risks, specified quality requirements, most functional requirements captured, and planned schedule, cost and resources.

Initial Operational Capability (IOC) - the system is prepared for operation and support, the deployment site is prepared and users, operators and maintainers are prepared and trained. The general objective of the construction phase, concluded with the IOC milestone, is a software capable of initial operation, that is, it is ready for *Beta* testing.

The UP has adopted these three critical milestones as major software development milestones, modified the interpretation of them slightly and added a *Product Release* milestone to end up with four major software development milestones, see Figure 3. By the end of the transition phase, concluded with the PR milestone, the software should be tested, corrected and ready for a formal release, including all documentation and all required preparation of the manufacturing or operation environment.

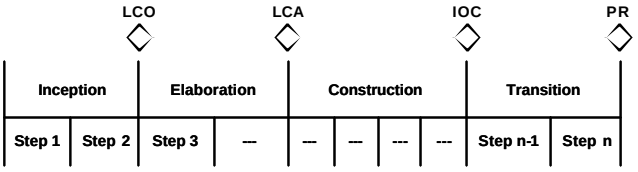

Figure 3. Major Milestones in the Unified Process

4. Model Mapping

To synchronize business decisions with development activities, in particular software development, a set of points for synchronization is needed. As the gates in the BDMs and major milestones in the SDLMs are clearly distinguishable, it is natural to use them for this purpose. It is desirable that the selection of SDLM is independent of the BDM used.

4.1 Pre-Gate Milestones

Gates should not be milestones in the software development plan as they belong to different processes. However, to be able to perform the assessment of the product and project before a gate, the required information has to be available; i.e. typically some key milestones have been passed. These milestones can be designated as *Pre-Gate Milestones*, see Figure 4. Note that pre-gate milestones are not specified only in the software development plan, but also in the marketing plan, competitor monitoring plan,, intellectual property management plan, training plan, service plan, quality assurance plan, hardware development plan and so on. Consequently, all pre-gate milestones, in all plans, ought to be passed before the assessment of the product and project takes place.

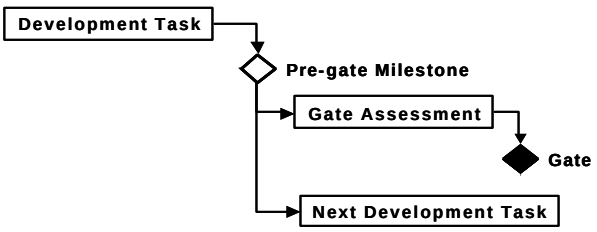

Figure 4. Pre-Gate Milestone

All product development tasks continue during the gate assessment but if a decision to stop the project is made at the gate, all planned tasks are cancelled.

4.2 Mapping Gates and Milestones

Using the concept of pre-gate milestones, mapping Cooper's Stage-Gate™ model product development gates and the UP software development major milestones becomes straightforward, see Figure 5. A go decision at Gate 1, idea screen (IS), is a prerequisite for the software development, as well as all other activities. Each major milestone in the UP can then be used as a pre-gate milestone to the corresponding business decision gate. An evaluation of the technical feasibility, an analysis of the market, an evaluation of the development and manufacturing/ operation capability, an estimation of development time and cost, and an investigation of any legal and regulatory constraints, provide input to Gate 2, second screen (SS). The scope and technical feasibility of the software should be defined in the UP inception phase.

A detailed technical appraisal, detailed market investigations and market research studies, as well as competitive analyses, investigations of needed internal investments, and detailed business and financial analyses provide input to the go-to-development (GTD) decision at Gate 3. A stable software architecture and planned schedule, staff and cost for the software development is the result of the UP elaboration phase.

Concurrent with the technical construction, market analysis and customer feedback activities are undertaken. Regulatory, legal and patent issues are resolved and test plans, market launch plans, production or operation plans are developed. When all this is done the project is ready to pass Gate 4, go-to-test (GTT). By the end of the UP construction phase, the software is ready for Beta testing.

During Beta testing, the market should be evaluated to determine expected market share and revenues, and the business and financial analyses should be revised. All this information has to be regarded at Gate 5, go-to-launch (GTL), as this is the last point at which the product can be stopped before a major commitment to production, service, maintenance, training and so on is made. Output from the UP transition phase, which is a software ready for formal release.

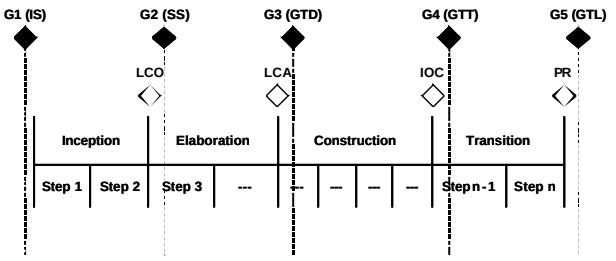

Figure 5. Mapping between Cooper's Stage-Gate™ model gates and Unified Process major milestones

This sample mapping illustrates that there is a good match between the information needed at the gates, and what is required to pass the milestones.

4.3 ABB Gate Model Mapping

The mappings of the ABB Gate Model gates and the UP major milestones much resembles that of Cooper's Stage-Gate™ model and UP, see Figure 6.

As in the Stage-Gate™ model mapping, a go decision at the ABB Gate Model Gate 0, start project (SP), is a prerequisite for starting the software development subproject, as well as the other subprojects. The goal of ABB Gate Model Gate 1, start project planning (SPP), is to agree on the project scope and the scope and technical feasibility of the software should be defined in the UP inception phase. The goal of ABB Gate Model Gate 2, start execution (SE), is to agree on the project plan and the goal of the UP elaboration phase is a stable software architecture and planned schedule, staff and cost for the software development. The goal of ABB Gate Model Gate 4, start introduction (SI), is to agree on the product readiness for piloting and market introduction, and the goal of the UP construction phase is to provide software ready for Beta testing. Finally the goal of ABB Gate Model Gate 5, release product (RP), is to agree on the product readiness for release and the goal of the UP transition phase is software ready for a formal release.

It is only before ABB Gate Model Gate 3, confirm execution (CE), that a major milestone in the UP, usable as a pre-gate milestone, is missing. Instead, a minor milestone indicating the finalization of an iteration [2] should be selected and used as a pre-gate milestone in the software development subproject.

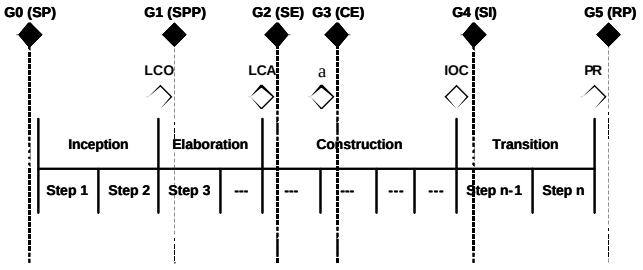

Figure 6. Mapping of ABB Gate Model gates and Unified Process major milestones

The main purpose of Gate 3 is to agree on the proposed technical solution, which must be taken into account when selecting an appropriate minor milestone. Experience indicates that it is good practice to decompose the UP construction phase into sub-phases and to mark the completion of the first sub-phase with a software release milestone that is also used as a pre-gate milestone to Gate 3.

At this point the software has been integrated for the first time and the technical solution can be assessed. Consequently, it is an appropriate time for a gate decision.

When working with mappings in ABB, the experience is that in order to allow sufficient time for assessments and gate meeting preparations, the pre-gate milestones need to be passed at least two weeks before the corresponding gate.

5. Conclusion and future work

By combining a business decision model with a software development lifecycle model, but at the same time recognizing the need for two separate models, several advantages are achieved. In this paper we have illustrated how an organization can have the possibility to select a software development lifecycle model independently of the chosen business decision model. This also makes it easier for the organization to adopt future software development lifecycle models. Through a proper combination of the two model types, organization will also avoid trying to meet the need for a business decision model with the means of a software development lifecycle model, or vice versa.

The combination of models is theoretically simple, but may lead to misconceptions, e.g. that a business decision model forces the use of a waterfall like software development lifecycle model. This is typically a result of a misunderstanding of the gate concept. Instead of evaluating the business aspects of the software project and product at appropriate points in time, the gate assessments are used to "tick off" that deliverables are completed and the gates are regarded only as additional milestones.

The mapping between the ABB Gate Model and local development lifecycle models has shown positive results so far, with some variations. Organizations using UP have adopted the ABB Gate Model smoothly without serious problems. In some cases the organizations have experienced problems gathering all the information needed for the decisions at the early gates, especially at gate 2, which actually is an indication of the organization's maturity level.

The first organizations implementing the mapping between gates and major milestones have experienced a higher degree of management understanding as well as increased speed in the development. The projects have reduced the non-value added project tasks, and the clearly identified business decision model has made it possible to introduce iterative development in the software development subprojects, since the requirement from management on project status visibility is satisfied by the gate model.

Future efforts will be focused on mapping the ABB Gate Model and a wider set of software development lifecycle models, such as different variants of incremental or evolutionary models. To achieve this, a generic method for mapping different business decision models to software development lifecycle models will be developed. At the same time, the long-term effects of the deployment and use of the ABB Gate Model together with different software development lifecycle models will be analyzed. Also, experiences from the combination of business decision models and development models for other product development activities, such as intellectual property development or development of marketing and sales material, are needed to get the whole picture of developing the right software products in the right way.

6. References

[1] Cooper Robert G., *Winning at new Products*, Third Edition, ISBN 0-7382-0463-3, Perseus Publishing, 2001

[2] Jacobson Ivar, Grady Booch and James Rumbaugh. *The Unified Software Development Process*, ISBN 0-201-57169-2, Addison-Wesley, 1999, pp. 8-13, 102-104 and 410.411.

[3] GP-PMI 9AAD102113, *ABB Gate Model for Product Development 1.1*, ABB

[4] Cooper Robert G., Doing it right, Winning at new Products, Ivey Business Journal July/August, 2000

[5] Cormican, K. and O'Sullivan, D., Product Manager: A Decision Support Tool for Design Engineers". Proceedings of the European Network of Excellence on Advanced Methodologies and Tools for Manufacturing Systems International Scientific-Technical Workshop, 1999, Ufa, Russia.

[6] Johansson, J.; Nilsson, L. Product planning at an Electrolux subsidiary Engineering and Technology Management, 1998. Pioneering New Technologies: Management Issues and Challenges in the Third Millennium, *IEMC '98 Proceedings*

[7] National Renewable Energy Laboratory, Stage Gate Management in the Biofuels Program. (2001). 41 pp.; *NICH Report No. MP-510-31541.*

[8] Ericsson Project Management Institute, project-management model PROPS, *http://www.ericsson.com/epmi/index.shtml*

[9] Boehm Barry. *Anchoring the Software Process*, IEEE Software, July 1996, pp. 73-82.