Integrating Business and Software Development Models

Christina Wallin, Fredrik Ekdahl, and Stig Larsson, ABB

Today, software product development cannot generally be regarded as successful. Only about one of four software development projects are completed on time and on budget, with all the features and functions originally specified.1 Running a software project is a complex task in itself; making the resulting product a commercial success is even harder.

Software development life-cycle models and business decision models contribute to the control of product development in different ways. However, both kinds of models have limitations. SDLMs do not ensure that resources are used in the right projects, that the market is available, or that the organization is ready for a release. Similarly, business decision models do not support software development, so development might take place with uncontrolled changes and inadequate time for verification and validation.

Thus, successful software product development requires that the project use both a business decision model and an SDLM. This requires careful definition of the interfaces, or mapping, between the two model types, as well as to any other model related to software product development. The ABB Gate Model, presented here, supports decision makers with business-relevant project and product information, increases mutual understanding and improves visibility between decision makers and developers during product development, educates decision makers in software engineering problems and solutions, and educates developers in business issues.

How business issues hurt software development

Many business-related problems face software product development. First, stakeholders typically scrutinize their software development projects from a business perspective only at startup, if at all; they do not revisit the business case over the course of the project. Often, they do not identify market, technology, or schedule problems until the project has gone astray.

Second, because new technology drives software product development, stakeholders typically examine a project’s business aspects less carefully than the technical solu-
This is of course a serious mistake, especially when a project is targeting a market that is new to the organization and when knowledge about this market is limited. Unfortunately, limited knowledge often leads to even less activity in trying to understand the business aspects.

Third, decision makers who don’t understand the basics of software engineering change the target continuously without looking at resulting costs and delays. This is probably a result of the common view that developers can easily adapt software to last-minute requirements. However, decisions to change or add new functionality often overlook the tasks that go along with code changes—for instance, changed architecture and design documentation, changed user documentation, regression testing, redesign of test cases for verification and validation, and changes to training, marketing and support material, and so on.

Finally, project managers can feel squeezed in the middle. Typically, decision makers want facts as soon as possible, but ask for finalized documents. For example, a manager might want to know if the selected technical solution is feasible, but instead asks if the detailed design document is ready. On the other side, developers might not think they can provide enough information when the decision makers want it. They often think that business decision models imply a waterfall-like development life cycle, so those who want to use modern development practices might resist using any such model. Also, modern practices such as the Unified Process and Extreme Programming require iterative and incremental development, which leads to late finalization of documents.

Business decision models

Delivering a product with expected quality and functionality, on time, and on budget is seldom enough to achieve commercial success. It is at least equally important to choose the right product development projects and to have a mechanism for closing down projects that no longer show sufficient potential.

Good business decisions are based on facts elicited through careful evaluation of key elements of the business situation—for example, market, competitors, technical feasibility, strategy, intellectual property, product quality, and resource availability. To facilitate the collection of relevant facts in time to make business decisions, many organizations use a well-defined process.

Several well-known business decision models exist, of which Cooper’s Stage-Gate Process Model is one example (see the sidebar on page 30 for more information). Typically, they comprise a number of different development stages separated by decision points, often referred to as decision gates. The gates represent distinct decision points at which stakeholders decide the project’s future.

Software development life-cycle models

Several SDLs support software development projects. When correctly implemented, they help projects deliver products with expected quality and functionality, on time and within budget.

Most SDLs divide the development life cycle into several phases, generally three to five. However, there are almost as many names for these phases as there are SDLs (see Figure 1). Phase names typically indicate the main activity performed in that phase and do not distinguish the concerns of project management and software development. This article uses the life-cycle phases defined in Microsoft’s Synch-and-Stabilize Life Cycle, the Unified Software Development Process, and Extreme Programming as examples. These models are commonly known, and their life-cycle phase names cannot be confused with software development activities such as analysis, design, implementation, verification, and validation, as described in the traditional waterfall model. Moreover, these three approaches’ phase names indicate the product’s maturity rather than the development activities performed.

In most SDLs, passing a major milestone marks the transition from one development phase to the next (see Figure 2). Of the three models just listed, only XP does not mention milestones. The Unified Process uses the three anchor-point milestones that Barry Boehm defined (Life-Cycle Objectives, Life-Cycle Architecture, and Initial Operational Capability) to mark each phase’s conclusion and the stakeholders’ commitment to move ahead. The UP also adds a Product Release milestone that concludes the Transition phase. Synch-and-Stabilize identifies three major milestones, each concluding a phase.

Delivering a product with expected quality and functionality, on time, and on budget is seldom enough to achieve commercial success.
Cooper’s Stage-Gate Process Model, shown in Figure A, breaks the development project life cycle into six stages and five gates. Each stage consists of a set of parallel activities, of which software development is only one, performed by different functions 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 development project.

The stage before the actual development project starts, the Discovery stage, begins with an idea for a new product or product version. Generally, a product manager collects the information needed as input to the first business decision gate.

Gate 1, the Idea Screen decision point, follows the Discovery stage and is the first occasion where decision-makers commit resources to the product development project. The product manager presents the idea to the stakeholders from development, marketing and sales, service and maintenance, manufacturing, training, and so on, who together decide whether to start a development project based on the idea.

During the first product development stage, the Scoping stage, the main objective is to assess the market and technology to identify key product requirements.

Gate 2, the Second Screen decision point, essentially repeats the previous gate, although with more rigorous requirements and based on the information gathered during the Scoping stage.

The second development stage, Building the Business Case, includes a detailed investigation that clearly defines the product, market, organization, development project, competitors, intellectual properties, and so on in preparation for deciding whether developing the product is feasible.

Gate 3, the Go to Development decision point, is the gate prior to the Development stage and the last chance to stop the project before the organization makes significant investments. A go decision at this point represents both a financial and resource commitment to the project as well as an agreement on the product and project definition established during the Building the Business Case stage.

The third stage, Development, mainly deals with the product’s physical development according to the product and project definitions. The deliverable from this stage should be a product ready for beta testing.

Gate 4, the Go to Testing decision point, is based on a postdevelopment assessment to ensure that the product and project are still attractive to the market and to the organization. A go decision at this point is an agreement on the verification and validation plans and also on marketing and operation plans.

In stage four, Testing and Validation, the product is verified and validated in-house or at friendly customers’ sites.

Finally Gate 5, the Go to Launch decision point, is the last point at which the project can be killed and the product cancelled. A go decision here is an approval of the marketing and operation plans and the startup of full production or operation. The final stage, Launch, includes, for example, activities for marketing and sales and for production or operation.
Both UP and S&S also use minor milestones; in the UP, each iteration ends with a minor milestone, whereas S&S uses a number of predefined minor milestones concluding various subprojects.

**Mapping business decision models and SDLMs**

A milestone is a scheduled event that marks the completion of one or more important tasks. The project manager uses milestones to measure and show achievements and development progress. At a milestone, a predefined set of deliverables should have reached a predefined state to enable a review. A gate, on the other hand, is a go-or-no-go decision point in the product development cycle, where all relevant business facts are brought together. At each gate, the decision maker uses the results from the preceding stage's activities together with a decision criteria checklist as input to the business decision.

Developers should not treat gates as software development milestones (see Figure 3), but they must pass some key milestones to be able to supply the decision maker with the required information in time before the gate. These important milestones could be called pregate milestones; they reflect the mapping between the business decision model and the SDLM. Of course, pregate milestones are not only in the software development plan but also, for example, in plans for marketing and competitor management, business, intellectual property management, training, customer service, quality assurance, hardware development, and so on. The project should pass all pregate milestones in all plans before the corresponding business decision at the gate.

Mapping a business decision model’s gates to an SDLM’s major milestones is straightforward (see the examples in Figure 4). A go decision at the first gate is a prerequisite to start software development as well as all the other activities. At this point, we can start the project if we decide that the intended product is a strategic fit, attractive to the market, and technically feasible. We can then use major milestones in the software development life cycle as pregate milestones corresponding to the business decision gates. If the gates outnumber the major milestones, we must select suitable minor milestones as pregate milestones.
Mapping SDLMs and the ABB Gate Model

To raise the quality of its product development business decisions, ABB developed the ABB Gate Model, a project control model reminiscent of Cooper's Stage-Gate. The ABB Gate Model consists of eight gates: gates 0 through 5 are true decision gates where the project can actually be canceled; gates 6 and 7 are used for follow-up and for a retrospective investigation of project experiences.

Mapping the UP major milestones and the ABB Gate Model gates is almost as straightforward as mapping to Cooper's Stage-Gate. It is only before ABB's Gate 3, Confirm Execution, that the UP is missing a pregate major milestone. Here, project management can choose a minor milestone indicating the finalization of an iteration or subphase as a

Table 1
Mapping the ABB Gate Model gates and the major Unified Process milestones

<table>
<thead>
<tr>
<th>ABB gate</th>
<th>Gate's purpose</th>
<th>UP's corresponding major milestone</th>
<th>Milestone's content</th>
</tr>
</thead>
<tbody>
<tr>
<td>G0</td>
<td>Agree to start project</td>
<td>–</td>
<td>Project start</td>
</tr>
<tr>
<td>G1</td>
<td>Agree on project scope</td>
<td>Life-Cycle Objectives</td>
<td>Software's scope set</td>
</tr>
<tr>
<td>G2</td>
<td>Agree on requirements and project plan</td>
<td>Life-Cycle Architecture</td>
<td>Stable architecture and planned software development schedule, staff, and cost</td>
</tr>
<tr>
<td>G3</td>
<td>Confirm consensus regarding proposed technical solution</td>
<td>–</td>
<td>Minor milestone should be selected</td>
</tr>
<tr>
<td>G4</td>
<td>Agree on the product's readiness for piloting and market introduction</td>
<td>Initial Operational Capability</td>
<td>Software ready for beta testing</td>
</tr>
<tr>
<td>G5</td>
<td>Agree on release</td>
<td>Product Release</td>
<td>Software's formal release</td>
</tr>
</tbody>
</table>
pregate milestone. Table 1 summarizes the requirements for the ABB Gate Model gates and for the UP’s major milestones.

Mapping the ABB Gate Model to XP resembles mapping to the UP but adds one complication. Because the time for planning in an XP project should be short, separating Gate 1 and Gate 2 is unnecessary (see Figure 5). (The recommended time for the planning phase in XP projects is about one week.) The proposed solution is to combine Gate 1 and Gate 2 and use the end of the planning phase as the point in time for a combined Gate1/Gate2.

When ABB first introduced a common decision model for product development, one of the developers’ most common concerns was that adapting to the ABB Gate Model seemed to force the projects to use the waterfall development model. To clarify this issue, ABB made available to its developers all the mappings this article describes.

So far, the results are promising. decision makers, project managers, and software engineers have reacted well to these mappings. Initial results show enhanced communication between the developers and the decision makers, increased focus on business aspects, and increased understanding of the differences between the models.

Current work focuses on making the mappings more widely known and used throughout ABB. By making these mappings available and broadly understood, ABB expects easier adaptation to future SDLMs, with new approaches to software development.

**Acknowledgments**

We recently presented a more detailed and theoretical version of this article at the 28th Euromicro Conference 2002. It is available in the proceedings published by the IEEE Computer Society.

**References**


For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

**About the Authors**

**Christina Wallin** is a research engineer at ABB Corporate Research, Sweden, and helps support software process improvement at several ABB sites. She is also a PhD candidate at Mälardalen University. Her research interests include management involvement in software development. She received her MSc from Luleå University of Technology, Sweden. Contact her at Mälardalen University, Computer Science and Engineering Dept., SE-721 23 Västerås, Sweden; christina.wallin@mdh.se.

**Fredrik Ekdahl** heads the software engineering process group at ABB Corporate Research. He received his MSc in industrial engineering and management and a PhD in quality engineering and management from Linköping University. Contact him at ABB Corporate Research, SE-721 78 Västerås, Sweden; fredrik.ekdahl@se.abb.com.

**Stig Larsson** is responsible for product development processes for the ABB group. He received his MSc in electrical engineering from the Royal Institute of Technology in Stockholm. Contact him at ABB Corporate Research AB, SE-721 78 Västerås, Sweden; stig.bm.larsson@se.abb.com.

**Figure 5. Mapping ABB Gate Model gates and Extreme Programming milestones.**

![Diagram showing the mapping between ABB Gate Model gates and Extreme Programming milestones.](image-url)