
Für Unbefugte
Aufenthalt
verboten
ABB

Taking the initiative

ABB's software development improvement initiative bears fruit

BRIAN P. ROBINSON, JOHN HUDEPOHL –
in the past, software was sometimes considered a sideshow to the main event, the hardware. Now, the software content of a product is often seen as a crucial differentiator. The amount of software in many products has increased dramatically over the years, too. ABB's recognition of these trends can be seen in its strategic acquisition of strong software companies such as Ventyx and Mincom, for example. Software development takes place in all parts of the ABB group and in order to maintain the leading market position of existing software solutions, and to create new ones that are best-in-class, ABB is exploiting modern software engineering methods. A recent group-level initiative brought about significant improvements in software development by focusing on three important aspects: processes, technology and people.

The software content of ABB products is increasing. Indeed, some products are nothing but software. Whereas, in the past, software often merely lent a helping hand to the hardware, it is now frequently seen as a differentiating technology in its own right. Software can be mission-critical, too: Consider the power-stations, industrial plants and infrastructure that rely on well-written and fault-tolerant software for their operation.

Self-scrutiny: Benchmarking ABB's capabilities

To assess ABB's own software capabilities, ABB approached experts in companies that make long-lived, critical infrastructure products. Some of these individuals, along with managers and technical experts from ABB, were formed into a benchmarking team. This team interviewed developers and testers at various ABB software development sites. As a result of these interviews, the team was able to compare ABB's software development capabilities with those of related industries.

> The software development improvement initiative focused on processes, technology and people.

Improvement initiative

Since ABB is a large company with very diverse products, developed in different parts of the world, an initiative was launched at the group level to act on the results of the benchmarking. While the initiative was planned and coordinated at the group level, the people involved were embedded in all five of ABB's divisions. The initiative focused on three main aspects of software development: processes, technology and people.

On the process side, the initiative was tasked with bringing best practice into ABB's development teams.

Under the newly-created software development improvement program (SDIP), these teams created a set of processes that would lead to improvements in their particular area → 1. The work also touched upon the software development life cycle and the ABB gate model.

Title picture
With a product's software content now often being the differentiating factor, software development must follow best-in-class practices.

SDIP creates and defines the best software development practices for the group and the methodology for implementing these.

1 SDIP brings consistency and harmonization to software developed for products like the paper machine drive shown here.

SDIP

SDIP strives towards continuous improvement by assessing progress and regularly validating achievements using the industry-standard capability maturity model integrated (CMMI), a process model provided by the Software Engineering Institute at Carnegie Mellon University, as a reference. SDIP creates and defines the best software development practices for the group and the methodology for implementing these. It specifies, and arranges, global license agreements for the best software tools.

The ABB tool suite seeks to establish a cost-effective and professionally-managed software engineering toolset.

Software development framework

An important result of SDIP is the software development framework. This framework ensures that the latest and best processes are used and it provides tools that embody and automate processes wherever possible. Human factors, such as training and motivation, are also covered by the framework.

Because shorter, more responsive development cycles are demanded, agile or iterative development life cycle models are often used in ABB. Every stage of these includes requirements, design, implementation and some testing.

Software development projects are performed in three phases: concept, development and deployment. The development life cycle used for software projects focuses on requirements, architecture, design, coding, testing, tracking and fixing. It defines the way the software community operates on a day-to-day basis and links to the ABB gate model at key points.

The ABB gate model

The gate model provides a conceptual and operational roadmap for moving a product project from idea to launch, and beyond. It provides a framework for better management of product development projects and it ensures that the line organization is actively involved in the project. At each gate, a decision is made whether to continue or stop. The projects software development life cycle is aligned

2 SDIP brought significant improvement.

"ABB has progressed a lot in software development practices since 2008, and there seems to be an excellent spirit in the organization to keep the improvements going. Software development is now generally taking place at a very professional level. Remember though, that agile programming or CMMI levels are means, but not goals by themselves."

Professor Claes Wohlin, Blekinge Institute of Technology, Sweden

with the gate model process so that inputs required for gate decisions are available at the appropriate time.

Software development technology

While there are many technology areas to consider, so far the initiative has concentrated on selecting and deploying best-in-class development tools across ABB's teams.

The ABB tool suite seeks to establish a professionally managed software engineering toolset. It realizes the vision of a common and integrated software engineering tool chain that simplifies the work flow for users, and increases the visibility of progress via metrics and reports throughout the software development cycle.

A software development life cycle is aligned with the gate model so that inputs for gate decisions are available at the right time.

Standardization and transparency are important watchwords of this global platform. Software engineering tools are employed to provide strong support for deployment of repeatable processes and tool-generated metrics, and reports increase transparency across all stages of a development project.

In other words, the ABB tool suite provides important infrastructure to enable speed, consistency, reliability, and measurement. The backbone of the system includes industry-leading tool packages. These tools streamline development so that products are delivered to the market faster and they support all the basic aspects of software development:

- Requirements, design, test cases, defects, configurations and build management.
- Early defect eradication.
- Effective validation (employing test automation and coverage tools to ensure products are built according to customer requirements).
- Measuring achievement of objectives and targets with data, using key metrics to make gate and milestone decisions.

This common approach also makes it easier to move people around inside the group and leads to a motivated and productive team.

A motivated and productive team

The group initiative also concentrated on training and career development. Software practices advance very quickly, so it is important to keep the practitioners up-to-date. A quality work force of global teams made up of skilled and motivated individuals will undoubtedly be successful in product development. In addition, performance improves when teams are given working processes that are appropriate, mature, and delivered with tools that reduce the effort needed. A competence development framework and software engineering curriculum are used to ensure that software development team members are equipped with the current and future skills needed to fulfill their project roles.

Benchmarking the improvements

After a certain time, the original benchmarking team was reassembled to assess progress. This time, a larger number of development groups were visited. For all of the focus areas in the benchmark, ABB showed significant improvement → 2.

Significant improvements

In accordance with current best practice guidance, ABB had improved software productivity in manageable stages by

3 Process improvement steps

Step	Actions
1	Begin with an assessment and baseline.
2	Initiate an upgrade to management skills in planning and estimating.
3	Start a measurement program to track progress.
4	Improve software defect removal via peer-to-peer reviews and static analysis.
5	Improve defect prevention via better requirements and design.
6	Improve the maintenance process via complexity analysis and restructuring.
7	Improve the conventional development process.

4 Improvement benefits

Software aspect	Percentage improvement
Development productivity	5 – 20%
Maintenance productivity	10 – 40%
Schedule shortening	5 – 15%
Reuse volumes (over 5 years)	From <20% to >75%

Global vendor license arrangements, group-wide implementation of the common toolset and a global tools infrastructure.

focusing on the top-ranked cost and schedule aspects and then carrying out a sequence of process improvement steps on these → 3. According to the latest research, based on information from many organizations, typical software process improvements have the potential to bring tangible benefits → 4.

An SDIP activity is now established in each ABB organization and individuals dedicated to software process improvement have been designated.

A process improvement example common to all was code review, which is an important technique for early defect detection in software development. Code reviews are labor-intensive events performed by developers, so it is important to have a lean, time-efficient process. For example, in the code base for one particular ABB product, significant productivity gains were made by supplementing manual code reviews with the Klocwork [1] static code analysis tool.

As some components in the code have over 100,000 lines of code, tracking code change dependencies during manual reviews was difficult. Klocwork massively reduced rework effort. After this success, extensive training on the tool was organized and its use is now widespread.

Another product that was improved is one that has an annual software release involving millions of lines of code. To address the problems experienced in past product launches, a number of process improvements, based on the CMMI, were put in place for the most recent release.

Information from reviews, for instance, helped balance project requirements against available resources, while code reviews not only helped catch code defects, but facilitated cross-training, mentoring and improved attention to detail.

Footnote

1 http://www.klocwork.com

5 Global license arrangements, a common toolset and a global tools infrastructure are essential for best-in-class software development. Source: Hewlett Packard.

Standardization and transparency are important watchwords of this common global platform.

These relatively small changes had a dramatic impact. The release was superior in terms of timeliness, functionality and perceived quality, and early indications from the validation are that big steps forward in terms of product quality have been made.

The way forward

While ABB has improved its ability to develop high-quality software, there are other best-practice improvements that will help ABB further differentiate its software products from those of its competitors.

Only by continually revisiting software development practices and implementing the best techniques available can ABB thrive in an industrial world where software is becoming ever-more widespread, differentiating and critical. As the life cycles of product development, and indeed of products themselves, shorten, only those with best-in-class development practices will succeed → 5.

Brian P. Robinson
ABB Corporate Research
Raleigh, NC, United States
brian.p.robinson@us.abb.com

John Hudepohl
ABB Technology Ltd.
Zurich, Switzerland
john.hudepohl@ch.abb.com