Agile collaboration on a global infotainment project

7 guidelines
for successful
agile
collaboration

Elektrobit and Ford worked together on the development of a global infotainment project. The collaboration was based on agile principles. Through this collaboration, we identified seven guidelines that can also be applied to other projects.

As innovation cycles become faster and complexity grows, agile methods are increasingly used for software projects in the automotive industry.

For agile collaboration to really work, a number of prerequisites must be met. The first prerequisite is a common understanding of the principles on which the collaboration is based. Elektrobit's experience shows that this goes far beyond the use of methods such as Scrum or Kanban - it describes a basic mindset. The second, mandatory prerequisite is a strong relationship of mutual trust between customer and supplier. In many cases, this requires long-term collaboration and successfully implemented projects. The following sections use the collaboration between Elektrobit and Ford for reference. Starting in 2007, the collaboration is now a long-term strategic software partnership, spread across three continents.

In this project, specific procedures were derived from the theoretical principles laid out, for example, in the "Agile Manifesto."

Manifesto for

Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Quelle: http://agilemanifesto.org/iso/en/manifesto.html

Figure 1: The "Agile Manifesto" describes the mindset of agile software development.

Seven guidelines supported both team-internal communication and coordination with the customer:

▶ **1. Deliver as quickly as possible.** Nightly builds allow the customer to view the current status on the actual product and provide direct feedback. A live system demonstrates features and concepts better than tickets and specifications. Their development means additional work and they often leave room for interpretation.

▶ **2. Integrity.** Like any good craftsman, each team member should aim for high-quality results. One tried-and-tested approach is pair programming: One developer writes code while the other focuses on the design. Permanent testing that is automated to the greatest extent possible is another mandatory aspect. In test-driven development, the developers first write the test and then the actual algorithm.

▶ **3. Decide as late as possible** and therefore remain able to act. This does not mean postponing decisions but making them at the right time. Decisions, for example, on the system architecture or user interface that are made too early create dependencies and can involve expensive, time-consuming corrections. It makes sense to start the project based on substantiated assumptions and make the necessary decisions just in time. This allows later correction of requirements, if necessary.

▶ **4. Quickly eliminate waste.** Obsolete code fragments or useless components should quickly be gotten rid of. Are there any elements or activities that do not add value? Working on unnecessary features, overfulfilling requirements or "programming in advance" should be avoided. Project managers face the following questions: Is later use or reuse possible? Does a supposed detour promise to deliver faster results or more reliability in the long term?

▶ **5. Empower the team.** The goal is a self-organizing, motivated and efficient team. Detailed planning is left to the team – based on tools such as Kanban boards. Metrics such as tasks completed per iteration, error correction period or return rates are used to ensure the project's success. For single tasks, the motto is "push, don't pull": The team itself decides who is working on which task. The Product Owner sets global priorities.

▶ **6. Reflect at regular intervals.** Feedback and ideas from the team significantly contribute to the product's quality and process efficiency. Specifically, Elektrobit abandoned the Scrum method at a later project stage. Due to the focus on error correction, for example, planning two-week sprints was no longer possible and required unnecessary effort. The project team opted to use the Kanban method. Its focus on optimized throughput made it better suited to the specific project situation.

▶ **7. See the whole.** Each team member is encouraged to think outside the box and assume responsibility for the whole product. For example, debugging – regardless of who may have caused the error – should always be performed in the entire system and not be limited to the team member's own scope of work. As a result, testing is not restricted to the development environment. It also takes place in the actual vehicle.

Best practices: Maintaining a backlog together with the customer promotes transparency. Ford can also view the source code at any time.

Spread over three continents, the development teams successfully organized their internal collaboration on the basis of these principles. To this end, they use a single global ticket queue from which the individual teams derive their tasks that are then visualized on local boards. These proven tools also support the collaboration of mixed teams of the customer's employees, suppliers involved and Elektrobit. Best practices also include agile tenders and contractual models such as the Agile Fixed Price.

Although agile methods are already being widely used, a consistent radical change is not yet in sight. The main reason for this is that a combination of different agile methods is needed to meet the industry's requirements. This is complex and, aside from the trust and confidence that has already been mentioned, requires extensive management know-how. However, this project shows that, with the relevant experience, using agile methods works well and, above all, pays off.

Figure 2: Local, 10 m² (ca. 108 sq. ft.) magnetic board.
Personalized magnets support appropriate task assignment.

Author:
Christian Mies
Elektrobit Automotive GmbH
Christian Mies is Head of Automotive Consulting
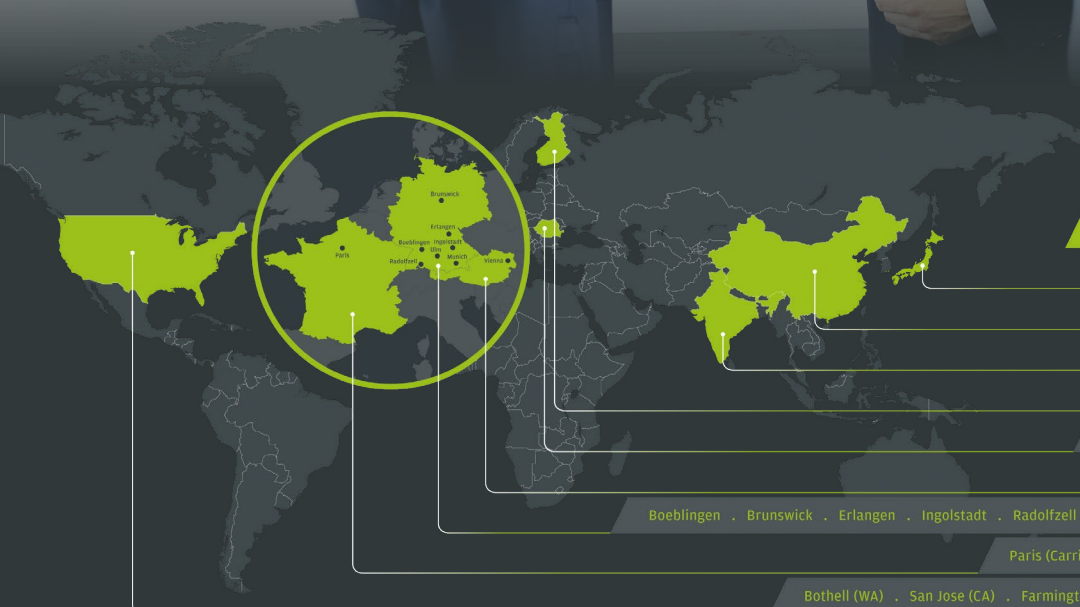at Elektrobit Automotive GmbH in Erlangen.

Notes

Notes

Notes

7 guidelines for successful agile collaboration

Elektrobit (EB) — Locations

Tokyo . Nagoya	**Japan**
Beijing . Shanghai	**China**
Bangalore	**India**
Oulu	**Finland**
Brasov . Timisoara	**Romania**
Vienna	**Austria**
Boeblingen . Brunswick . Erlangen . Ingolstadt . Radolfzell . Munich . Ulm	**Germany**
Paris (Carrières-sur-Seine)	**France**
Bothell (WA) . San Jose (CA) . Farmington Hills (MI)	**USA**

About Elektrobit (EB)

EB is an industry leading supplier of automotive software and has a proven record in embedded and connected software development for over two decades. As a dedicated partner to the automotive industry, EB provides technologies and flexible software platforms, tools, and services to help automotive manufacturers and their suppliers to deliver the best products and services in order to meet the needs of drivers.

EB

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
sales@elektrobit.com
www.elektrobit.com