

Agile Zusammenarbeit in einem globalen Infotainment-Projekt

Die Zusammenarbeit mit Ford bei der Entwicklung eines globalen Infotainment-Projekts hat Elektrobit nach agilen Prinzipien organisiert. Dabei identifizierte der Softwarespezialist sieben Leitlinien, die ebenso auf andere Projekte übertragen werden können.

Schnellere Innovationszyklen und die steigende Komplexität führen dazu, dass auch für Softwareprojekte in der Automobilindustrie zunehmend agile Methoden genutzt werden. Damit agile Zusammenarbeit wirklich funktioniert, müssen jedoch mehrere Voraussetzungen erfüllt sein. Die erste ist ein gemeinsames Verständnis der Prinzipien, auf denen die Zusammenarbeit beruht. Nach Erfahrung von Elektrobit geht dies weit über den Einsatz von Methoden wie Scrum oder Kanban hinaus und beschreibt vielmehr eine grundsätzliche Denkweise. Zweite, unabdingbare Voraussetzung ist ein ausgeprägtes Vertrauensverhältnis zwischen Auftraggeber und Lieferant. Dies stellt sich häufig erst nach längerer Kooperation und erfolgreich realisierten Projekten ein.

Im Folgenden dient die Zusammenarbeit zwischen Elektrobit und Ford als Referenz. Sie startete bereits 2007 und ist heute eine langfristig angelegte, strategische Software-Partnerschaft. In diesem konkreten Projekt wurden aus den theoretischen Prinzipien, wie sie etwa im „agilen Manifest“ formuliert sind, konkrete Handlungsweisen abgeleitet.

Sieben Leitlinien halfen bei der team-internen Kommunikation ebenso wie bei der Abstimmung mit dem Kunden:

1 So schnell wie möglich liefern. Auf Basis von „Nightly Builds“ kann der Kunde den aktuellen Status am tat-

Die Denkweise agiler Softwareentwicklung beschreibt das „Agile Manifest“.

sächlichen Produkt nachvollziehen und unmittelbares Feedback geben. Ein Live-System demonstriert Features und Konzepte besser als Tickets und Spezifikationen, deren Erstellung zusätzlichen Aufwand bedeutet und die häufig Interpretationsspielraum lassen.

2 Integrität. Wie bei einem guten Handwerker muss der Anspruch jedes Teammitglieds an sein Arbeitsergebnis hoch sein. Bewährt hat sich unter anderem „Pair Programming“: Ein Entwickler schreibt den Code, der andere konzentriert sich auf die Konzeption. Außerdem ist ständiges, so weit wie möglich automatisiertes Testen, unabdingbar. Im „Test Driven Development“ schreiben die Entwickler erst den Test und dann erst den eigentlichen Algorithmus.

3 Entscheidungen möglichst spät treffen und dadurch handlungsfähig bleiben. Das bedeutet nicht, Entscheidungen aufzuschieben, sondern sie zum richtigen Zeitpunkt zu treffen. Zu früh getroffene Festlegungen etwa zur Systemarchitektur oder der Benutzeroberfläche schaffen Abhängigkeiten und können aufwändige Korrekturen nach

sich ziehen. Sinnvoll ist, das Projekt auf Basis fundierter Annahmen zu beginnen und nötige Entscheidungen „just in time“ zu treffen. So kann man Anforderungen bei Bedarf nachträglich korrigieren.

4 Abfall schnell entsorgen. Es gilt, sich zügig von veralteten Code-Fragmenten oder unnützen Bestandteilen zu verabschieden. Gibt es Elemente oder Tätigkeiten, die nicht zur Wertschöpfung beitragen? Zu vermeiden sind etwa Arbeiten an unnötigen Features, die Überfüllung von Anforderungen oder „Programmierung auf Vorrat“. Projektverantwortliche müssen hier abwägen: Kommen ein späterer Einsatz oder eine Wiederverwendung in Betracht? Verspricht ein vermeintlicher Umweg auf lange Sicht schnellere Ergebnisse oder höhere Zuverlässigkeit?





Die personalisierten Magneten auf dem zehn Quadratmeter großen Magnetboard helfen bei einer angemessenen Aufgabenverteilung.

5 Vertrauen ins Team. Ziel ist ein selbstorganisierendes, motiviertes und effizientes Team. Detailplanungen bleiben dem Team überlassen – auf der Grundlage von Werkzeugen wie einem Kanban-Board. Zur Sicherstellung des Projekterfolgs dienen Metriken wie die abgearbeiteten Aufgaben pro Iteration, die Bearbeitungsdauer von Fehlern oder Rückläuferquoten. Für Einzelaufgaben gilt dabei das Motto „Pull statt Push“: Das Team entscheidet selbst, welches Mitglied welche Aufgabe bearbeitet. Übergeordnete Prioritäten setzt der Product Owner.

6 Regelmäßig reflektieren. Feedback und Ideen aus dem Team tragen wesentlich zur Qualität bei. Konkret hat Elektrobit die Nutzung des Scrum-Verfahrens in einer späteren Projektphase wieder aufgegeben, weil durch den Fokus auf Fehlerbehebung, etwa die Planbarkeit von Zwei-Wochen-Sprints nicht mehr gegeben war und unnötigen Aufwand erforderte. Die dann eingesetzte Kanban-Methode, mit dem Fokus auf optimierten Durchsatz, war besser geeignet.

7 Das ganze Bild sehen. Jedes Teammitglied ist aufgefordert, über den Tellerrand hinaus zu blicken und Verantwortung für das gesamte Produkt zu übernehmen. Beispielsweise sollten Fehlersuchen – unabhängig vom Verursacher – immer im Gesamtsystem erfolgen und nicht nur auf den eigenen Umfang bezogen sein. Tests finden daher nicht nur in der Entwicklungsumgebung, sondern auch im realen Fahrzeug statt.

Best Practices Ein gemeinsam mit dem Kunden gepflegtes Backlog fördert die Transparenz. Auch der Source-Code ist jederzeit von Ford einsehbar. Ihre interne Zusammenarbeit haben die über drei Kontinente verteilten Entwicklerteams ebenfalls auf der Grundlage dieser Prinzipien erfolgreich organisiert. Sie nutzen dazu eine einzige, globale Ticket-Queue, aus der die einzelnen Teams ihre Aufgaben ableiten und auf lokalen Boards visualisieren. Diese bewährten Werkzeuge unterstützen auch die Zusammenarbeit gemischter Teams aus Mitarbeitern des Kunden, involvierten Zulieferern sowie von Elektrobit. Sinnvoll sind in diesem Zusammenhang auch agile Ausschreibungen und Vertragsformen wie der agile Festpreis.

Obwohl agile Methoden bereits genutzt werden, ist noch kein konsequenter Umbruch in Sicht. Vor allem weil eine Kombination verschiedener agiler Methoden benötigt wird, um den Anforderungen der Branche gerecht zu werden. Dies ist aufwändig und setzt, neben dem bereits erwähnten Vertrauen, großes Management-Know-how voraus. Dass der Einsatz mit der entsprechenden Erfahrung gut funktioniert und dass es sich vor allem rentiert, zeigt das vorliegende Projekt. ■



Elektrobit (EB) Automotive GmbH
www.elektrobit.com



Christian Mies leitet den Geschäftsbereich Consulting bei der Elektrobit Automotive GmbH in Erlangen.

Elektrobit

Elektrobit (EB) ist ein führender Anbieter von Automotive Software. Das Unternehmen steht für eine rund 30 Jahre andauernde Erfolgsgeschichte in der Entwicklung von Embedded- und Connected-Softwarelösungen. Als kompetenter und engagierter Partner unterstützt EB Automobilhersteller und -zulieferer mit Technologien, flexiblen Softwareplattformen, Tools und Dienstleistungen dabei, den Erwartungen von anspruchsvollen Autofahrern gerecht zu werden. In partnerschaftlicher Zusammenarbeit mit branchenführenden Unternehmen entwickelt EB die nächste Generation intelligenter, flexibler und kosteneffizienter Softwarelösungen für das Fahrzeug und ist aktiv an der Weiterentwicklung wichtiger Industriestandards beteiligt. EB verfügt über langjährige Erfahrung in sicherheitskritischen Umgebungen – von Navigation und Infotainment über ECUs bis hin zu Fahrerassistenz und automatisiertem Fahren. EB-Software wird in über 100 Millionen Fahrzeugen weltweit genutzt.

Kerndaten

- Geschäftsführer: Alexander Kocher, Gregor Zink
- Hauptsitz: Erlangen, Deutschland
- 20 Niederlassungen: China, Deutschland, Finnland, Frankreich, Indien, Japan, Österreich, Rumänien, USA
- Weltweit über 1 700 Mitarbeiter
- EB ist eine Tochtergesellschaft der Continental AG.



Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen
Telefon: +49 (0) 91 31/77 01-0
Web: www.elektrobit.com
E-Mail: sales@elektrobit.com