

Scrum & Co.

Agile Entwicklung von Embedded-Systemen

24.01.14 | Autor / Redakteur: Remo Markgraf * / Franz Graser



Die Umstellung des kompletten Entwicklungsprozesses inklusive des Systemtests auf agile Methoden und eine enge Kopplung mit der Hardwareentwicklung führen zum erfolgreichen Einsatz agiler Konzepte.

Häufig wird die agile Softwareentwicklung in eine traditionelle Systementwicklung nach dem V-Modell/XT eingebettet. Leider wird hier meist verkannt, dass die agilen Vorteile dabei kaum zum Tragen kommen, da sie von zusätzlichem Aufwand bei der Einbettung aufgezehrt werden.

Entgegen dem agilen Manifest müssen etwa Entwicklungsdokumente vorab erstellt werden, um dem traditionellen Prozess zu genügen. Als Ausweg werden die Dokumente als Teil der Lieferung in die agilen Backlogs aufgenommen. Der agile Vorteil durch die Lean-Prinzipien – erst ausprobieren, dann machen, dann dokumentieren – geht so völlig verloren.

Und das ist nur ein Beispiel für vielfältige Schwierigkeiten bei der Einbettung. Ein viel effizienterer Weg ist der komplette Umstieg auf eine agile Systementwicklung. Dabei werden neben der Softwareentwicklung auch der Systemtest und der Integrations- und Gesamtsystemprozess agil umgesetzt. Nur die Hardwareentwicklung wird weiter traditionell geführt und iterativ mit den agilen Prozessen verknüpft.

Schritte zur erfolgreichen Einführung agiler Methoden

Der Umstieg von der traditionellen Entwicklung zum agilen Systemprozess wird am besten in kleinen Schritten vollzogen. Zunächst startet man mit der agilen Softwareentwicklung (etwa Scrum) und erweitert nach der Eingewöhnungsphase die

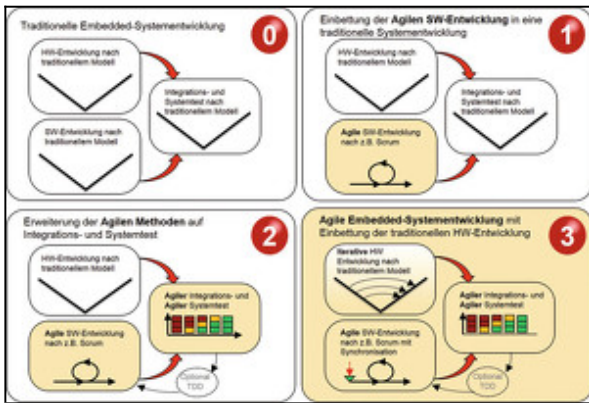
Reichweite der agilen Prozesse auf den Systemtest. Diese Reihenfolge liegt auf der Hand, da nach der Einführung der agilen Entwicklung die Kluft zwischen Entwicklern und Systemtestern verstärkt wird, da die einen schon agil, die anderen noch traditionell arbeiten.

Es liegt also nahe, auch den Systemtest agil umzustellen. Die Sprintziele der Entwicklungs-/Testteams werden dazu sukzessiv in der Reihenfolge Entwicklungsziele, Testvorbereitung, Integrationstest, Systemtest und Abnahmetest verschoben. Bei mehreren parallelen Teams werden die Teams zu Beginn in ihrer traditionellen Zusammensetzung belassen; nur die Aufgaben werden nach den agilen Prinzipien erledigt. Nach erfolgreicher Einführung werden die Teams neu nach Themengebiet zusammengesetzt. Architekten, Entwickler und Tester bilden gemeinsame Teams, die ihr Thema von der Idee bis zur Kundenabnahme begleiten. Damit erreicht man eine klare Identifikation der Teammitglieder mit ihrem Thema. Der Kundennutzen der einzelnen Themengebiete wird größer, da jedes Teammitglied den Blick auf das Gesamtergebnis hat.

Da der übergeordnete Entwicklungsprozess noch traditionell ist, treten auch jetzt an den Schnittstellen zwischen den Prozessen Verluste auf, die mögliche Vorteile zum Teil aufzehren. Der schwierige Schritt, die Umstellung auf den agilen Systemprozess, steht nun an. Aufgrund der Umstellung der Softwareentwicklung und des Systemtests wurde „in-house“ bereits Erfahrung mit Umstellungsprozessen gesammelt; daher wird der System-Umstellungsschritt relativ reibungslos ablaufen. Wesentlicher Erfolgsfaktor ist, den Auftraggeber und den Vertrieb ins Boot zu holen.

Die Aussage „Ich kann ihnen sagen, wann Sie die Lieferung bekommen – nur leider nicht, was!“ wird die meisten Kunden wohl eher abschrecken. Doch die Vorteile sind gerade für den Endkunden erfreulich. Er braucht das gewünschte System zu Projektbeginn nicht bis in das letzte Detail durchzuplanen. Auch die Detailanforderungen dürfen iterativ mitwachsen. Der Kunde sieht das System wachsen, kann erste Erfahrungen sammeln und wesentlich präziser seine Vorstellungen äußern. Am Ende entsteht ein System mit optimiertem Kundennutzen.

Um vertraglichem Wildwuchs vorzubeugen, empfiehlt es sich, den Kunden in den agilen Systemprozess einzuweihen und eine Bepreisung nach Storypoints auszuhandeln. Storypoints werden in der Aufwandsschätzung für jedes Feature festgelegt und sind ein Maß für die Komplexität zur Entwicklung des Features. Ein Angebot nach Storypoints ist sowohl bei Aufwands- als auch Festpreisangeboten möglich, die eine bestimmte Anzahl an Storypoints garantieren. Zu Beginn einer Iteration kann der Kunde bei der Auswahl der Features mitbestimmen und so seine Prioritäten sichern.



Natürlich hält der agile Systemprozess weitere Herausforderungen bereit. Die Einbettung der Hardwareentwicklung nach dem traditionellen V-Modell sei hier stellvertretend erläutert.

Schwierigster Teil der Aufgabe ist die Abschätzung der Dimensionierungsanforderungen. Im traditionellen Modell wird zunächst das System detailliert spezifiziert. Die Anforderungen sind somit bekannt. Ob das System letztlich auch so realisiert

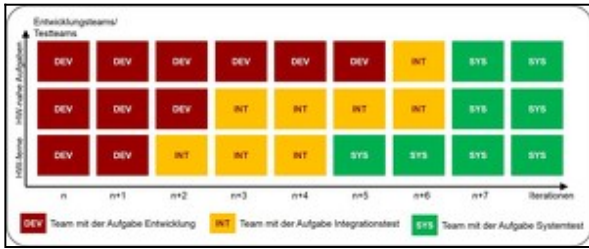
wird, hängt natürlich davon ab, ob sich die spezifizierte Lösung tatsächlich umsetzen lässt. Im agilen Ansatz wird dieses Problem geschickt umgangen, indem erst ausprobiert wird und sowohl das System als auch die Dokumentation iterativ wachsen.

Einbettung der Hardwareentwicklung gemäß dem V-Modell

Um eine grobe Abschätzung der Dimensionierung zu ermöglichen, muss auch im agilen Systemprozess das Gesamtsystem zumindest grob architekturell geplant werden. Basierend auf dieser Grobplanung schätzt man die Dimensionierung ab und verfeinert sie mit jeder Iteration. Zum Glück gibt es viele Pin-kompatible Controller in verschiedenen Leistungs- und Speichergrößen, sodass eine Korrektur nicht immer gleich ein teures Redesign bedeutet. Die HW selbst entsteht trotz V-Modell iterativ. Zunächst verwendet man wenn möglich ein Standard-Entwicklungsboard für die ersten SW-Entwicklungsschritte. Zweckmäßigerweise beginnt man mit den möglichst HW-unabhängigen Anteilen. Mit Fertigstellung des ersten HW-Prototypen schwenkt man mit einem Teil der SW-Entwicklung auf den Prototypen um.

Im Laufe der weiteren Iterationen werden sukzessive die weiteren SW-Entwicklungen auf den Prototypen verlegt, sofern dessen Stabilität dies erlaubt. Implizit wird dabei durch Testen und Verifizieren der SW-Funktionen, die zuvor auf dem Entwicklungsboard liefen, auf den Integrationstest des Systems umgeschwenkt. Dieser Prozess setzt sich kontinuierlich mit dem korrigierten Prototypen und dem ersten Hardware-Release fort.

Vergleichen wir nun die traditionelle Produktentstehung mit der agilen Systementwicklung, so stellen wir erfreut fest, dass die agile iterative Methode dem natürlichen Erfinderdrang



Vergleichen wir nun die traditionelle Produktentstehung mit der agilen Systementwicklung, so stellen wir erfreut fest, dass die agile iterative Methode dem natürlichen Erfinderdrang – fast möchte man sagen Spieltrieb – der Entwickler sehr entgegenkommt. Nach einer Eingewöhnung kommt ein Entwickler wesentlich besser damit

zurecht, etwas termingerecht abliefern zu müssen, als damit, sich in der Sache vorweg zu sehr festzulegen. Teamgeist und die höhere Eigenverantwortung sorgen so für qualitativ hochwertigere und früher verfügbare Lösungen. Die Kosten der Entstehung sinken damit. Durch den Umstieg auf den agilen Systemprozess kann wesentlicher Mehraufwand bei der Einbettung agiler Teilentwicklungen vermieden werden – die agilen SW-Entwicklungsvorteile lassen sich für das gesamte Embedded-System ausschöpfen.

Wer den Umstieg auf die agile Systementwicklung erfolgreich geschafft hat, der ruht sich nicht aus, sondern sucht gezielt nach weiteren Optimierungsmöglichkeiten. Es muss doch noch besser gehen! Mit dem Schritt in Richtung Test Driven Development lassen sich die Vorteile des agilen Testens weiter ausbauen und damit die Qualität und Systemstabilität weiter verbessern. Warum nicht gleich bei der Grobspezifikation des Systems die System-Testfälle gestalten und das System so kontrolliert wachsen lassen? Test Driven Development bedeutet in diesem Zusammenhang, dass nicht nur die Testfälle vor der Codierung der SW entstehen, sondern gleich die Testfälle für das Gesamtsystem.

Software as a Service (SaaS)

Web-basierte Oberflächen für Embedded-Systeme sind längst State of the Art. Jeder Billig-Router für 20 Euro bietet diese Möglichkeit. Smartphones und Smart-TV verfügen über komplexe Applikationen, die einem ständigen Update-Prozess unterliegen.

ERGÄNZENDES ZUM THEMA

- ▼ Tiefere Einblicke in die Agile Entwicklung
- ▶ Tiefere Einblicke in die Agile Entwicklung

Agile Entwicklungsmethoden werden auch im Embedded-Umfeld immer populärer. Nicht zuletzt deshalb, weil ihre Prinzipien – erst ausprobieren, dann machen, dann dokumentieren – den Bedürfnissen der Entwickler entgegenkommen. Es ist also ratsam, rechtzeitig vorzusorgen und die Technologie-Speerspitze zu bilden. Unter anderem zum Thema „Agile Entwicklung von Embedded-

Systemen“ bietet MicroConsult Coaching und ein zweitägiges Training an. Mehr Infos auf:
<http://www.microconsult.de/um/agile-dev>.

Die Updates nicht mehr herunterzuladen, sondern gleich in der Cloud zu belassen und dem Kunden als SaaS anzubieten, bringt einige Vorteile: Der Hersteller braucht nicht mehr verschiedene, sich im Feld befindliche Systemversionen zu warten. Der Kunde hat ohne Installations- und Wartungsaufwand stets die aktuellste Softwareversion in Betrieb. Bei Problemen hat der Servicetechniker sofort Zugriff auf das System. Doch was bedeutet dies für das gesteuerte Embedded-System? Das System muss mit dem Server im Internet kommunizieren, und der Anwender bedient nicht das System selbst, sondern die App im Internet.

Zusammenfassung

Neben der steigenden Bedeutung von Haftungsfragen und der Betriebssicherheit wird für die Entwicklung ein immer breiter gefächertes Wissen unverzichtbar. Das Problem liegt häufig in einem unzureichenden Know-How der Embedded-Softwareentwickler. Es ist also jeder gut beraten, rechtzeitig vorzusorgen und die Technologie-Speerspitze zu bilden. Nach C und C++ sind nun Python, Ruby/Rails, Cloud, SaaS und agile Methoden die wichtigsten Ausbildungsziele.

//FG

* * Dipl.-Ing. (Univ.) Remo Markgraf ist Management-Trainer bei der MicroConsult GmbH. Er verfügt über langjährige Projekt- und Führungserfahrung in der Softwareentwicklung.

Copyright © 2014 - Vogel Business Media

Dieser Beitrag ist urheberrechtlich geschützt.
Sie wollen ihn für Ihre Zwecke verwenden?
Infos finden Sie unter www.mycontentfactory.de.

Dieses PDF wurde Ihnen bereitgestellt von <http://www.elektronikpraxis.vogel.de>

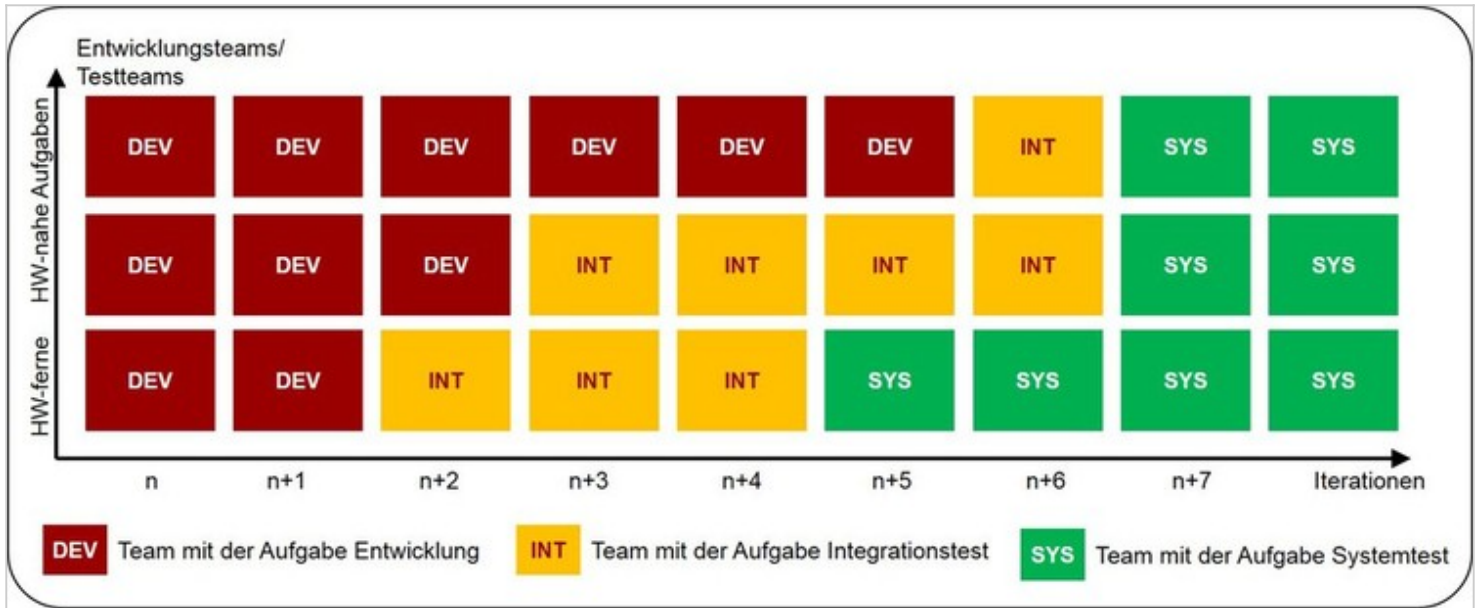


Bild 3: Erweiterung der agilen Methoden auf den Integrations- und Systemtest (Grafik: MicroConsult)

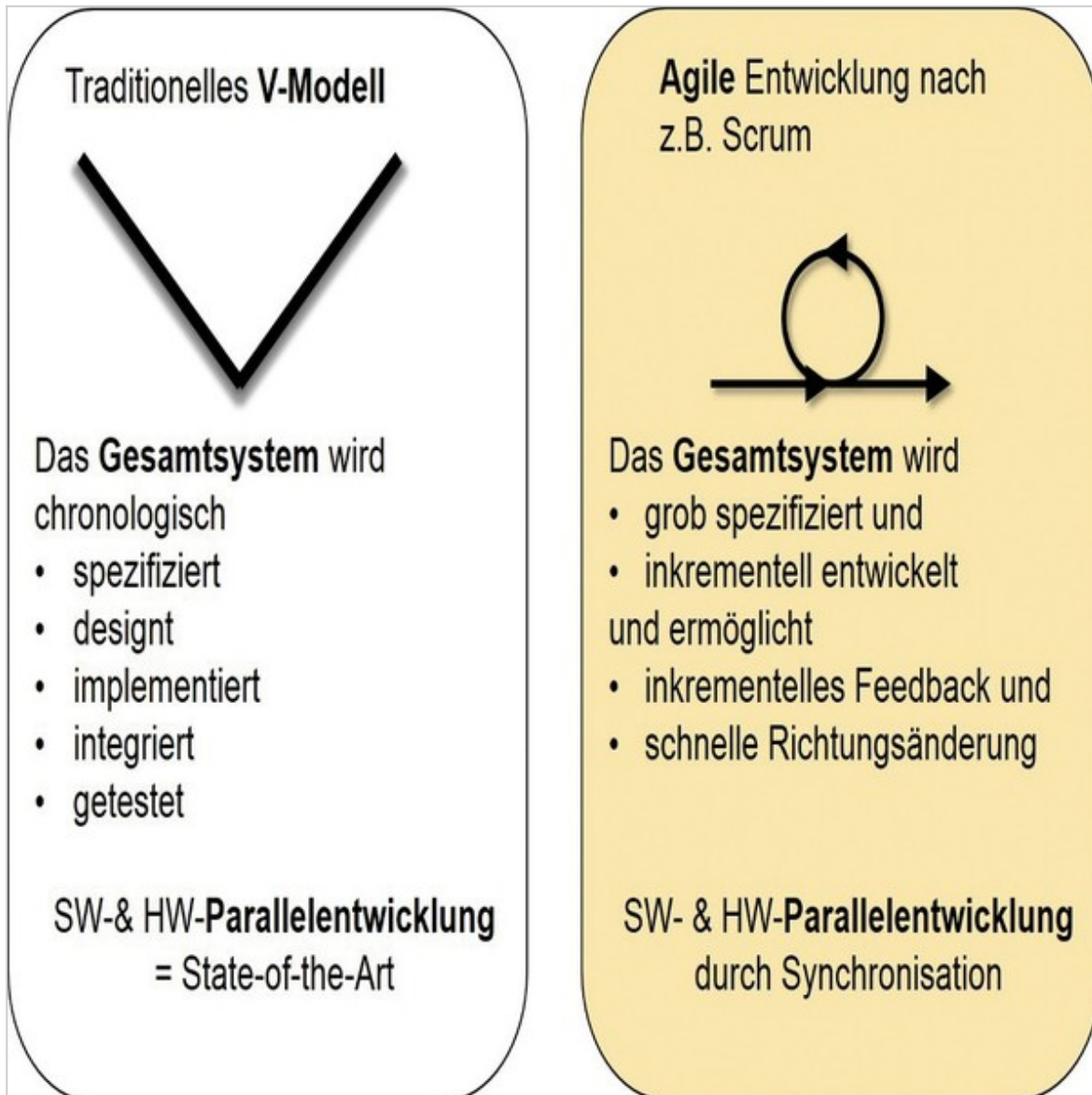


Bild 1: Gegenüberstellung der Grundsätze der traditionellen (links) und der agilen Systementwicklung (Grafik: MicroConsult)