

Drei Ingenieure und die agile Systementwicklung

27 Juni 2019 | **Product Engineering** | [Thomas Rahn](#)

Lesezeit: 6 Minutes

Schneller, flexibler und vor allem immer „in time“ und „in budget“ - immer mehr Unternehmen versuchen, die Herangehensweisen und Methoden aus der agilen Softwareentwicklung auf die System- oder Geräteentwicklung zu übertragen. Sie verschlanken ihre bisherigen Produkt-Entwicklungsprozesse (PEP), Meilensteine und Gate-Reviews weichen einem kontinuierlichen oder iterativen Vorgehen. Doch für einen erfolgreichen Transfer zu einer wirklich agilen Systementwicklung ist es nicht damit getan, nur Methoden aus der Software zu kopieren und Prozesse auszudünnen.

Bei der Systementwicklung arbeiten unterschiedliche Ingenieursdisziplinen zusammen, die alle ihre Eigenarten, speziellen Randbedingungen und unterschiedlichen Herangehensweisen haben. Es gibt wohlbegründete Unterschiede in der Herangehensweise der verschiedenen Ingenieursdisziplinen, mit denen bewusst umzugehen ist. Es geht vor allem um die spezifischen Freiheitsgrade im Design, bzw. um die typischen Constraints bei der Entwicklung eines serienfähigen Gerätes. Ich möchte hier also nicht stereotypisch davon sprechen, ob und wie man bei der Hardware-Entwicklung schneller zu Prototypen kommt. Wir wissen alle um die Fortschritte bei der Platinenfertigung oder durch 3D-Druck. Und auch iteratives Vorgehen kennen Elektroniker oder Kunststoff-Experten.

Agile Systementwicklung braucht die richtigen Zwischenziele

Für eine wirklich agile Systementwicklung ist ein umfassenderer Change-Prozess erforderlich. Dabei ist es entscheidend, alle Mitarbeiter im Unternehmen mitzunehmen und agile Entwicklungsteams zu formen, vom Produktmanagement bis zu den Entwicklern. Dieser Change-Prozess setzt sich mit den jeweiligen Herangehensweisen auseinander und versucht, den besten Methoden-Mix zu etablieren. Der entscheidende Punkt für das Gelingen einer agilen Systementwicklung ist es aber, die richtigen Zwischenziele zu setzen. Hierzu braucht es viel Erfahrung und vor allem ein gutes Einfühlungsvermögen in die beteiligten Disziplinen.

Bevor ich auf ein Projektbeispiel eingehe, möchte ich als Analogie den Hausbau heranziehen: Wie wäre es denn, wenn ein Programmierer, ein Elektroniker und ein Konstrukteur mein neues Haus bauen sollen – jeder auf seine Art? Am Ende möchte ich nicht nur ein MVP haben (in diesem Fall entspräche das einer Blockhütte ohne Fenster und ohne Warmwasser), sondern ein nettes Eigenheim mit allem Komfort, in dem ich gut alt werden kann. Wir lassen unsere drei Ingenieure mal beginnen:

Der Programmierer

Den Hinweis mit dem Warmwasser hat er gehört und macht sich als erstes an die Ausgestaltung des Badezimmers. Ein richtiger Wellnessstempel soll es werden. Er braucht dafür natürlich etwas mehr Raum (im echten Leben wären das CPU Performance, Speicher, Netzwerk-Bandbreite, usw.), aber bisher hat er von seinen Kollegen noch keine Vorgaben bekommen. Auch an die vielen Versorgungsanschlüsse denkt er. Für sein selbstgewähltes Aufgabengebiet erstellt er jede Menge Mocks, so dass virtuelles Wasser aus dem Hahn kommt und auch der Abfluss funktioniert. Am Ende des ersten Bauabschnitts gibt es viele grüne Unit-Tests und für den nächsten Sprint ist das erste Release in Modulbauweise gedacht.

Der Elektroniker

Er hat früh eine Vorstellung, was alles für den Rohbau und die Installationen nötig ist und wie eine Raumaufteilung aussehen könnte, was er braucht von der Fußbodenheizung bis zu den Fenstern (im echten Entwicklungsprojekt sind das die Baugruppen und Funktionsgruppen). Für die kniffligen Fragestellungen besorgt er Material im Fachhandel, wälzt Kataloge und fragt auch einige spezialisierte Anbieter an. Am Ende des ersten Bauabschnitts gibt es einige Aufbauten – etwa für die Duscharmatur, die Wasserverteilungen, Zähler und Sperrhähne inklusive Vernetzung und automatisierter Steuerung. Alles noch etwas groß und überdimensioniert, mit vielen Rohrstücken zusammengesetzt, aber das warme Wasser fließt aus dem richtigen Hahn.

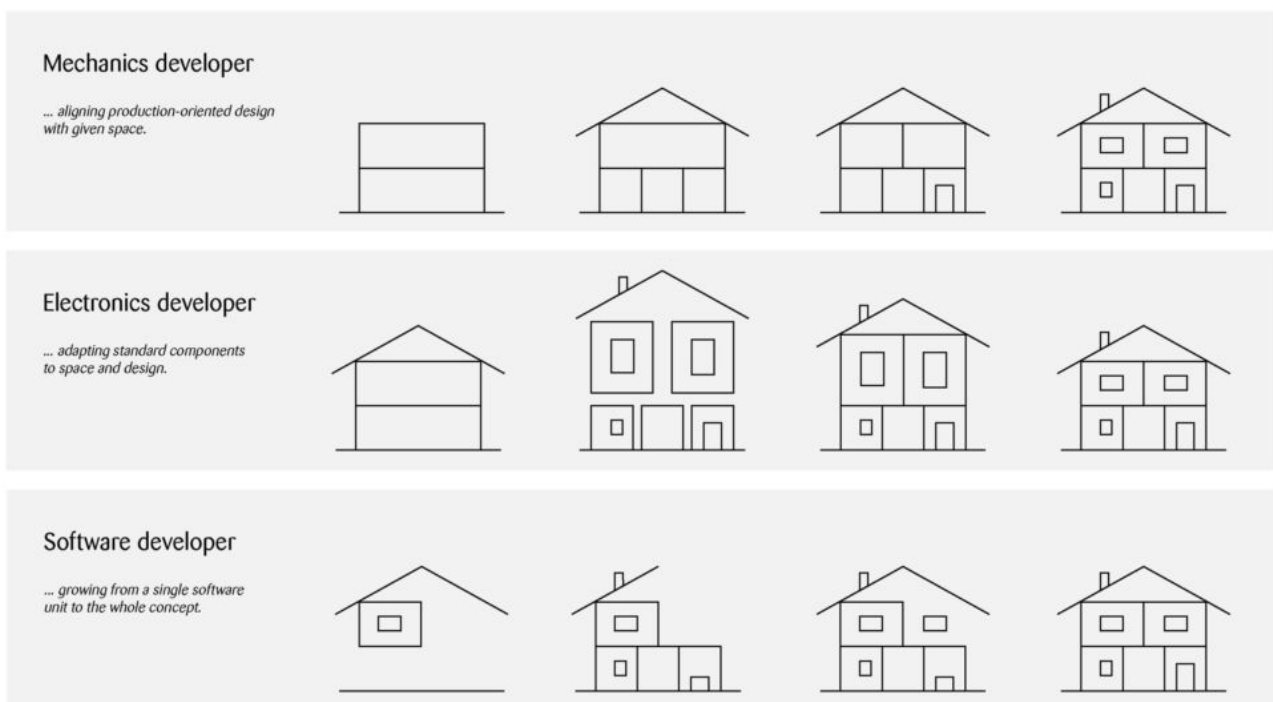
Der Konstrukteur

Ihn bewegt, dass der Bauplatz nicht sehr groß ist und auch wenig Platz für einen Kran oder einen Betonlaster bleibt. Und wie soll alles zusammenpassen und sich um ein Treppenhaus fügen, in das später vielleicht auch noch ein Treppenlift passt? Nachdem auch er die kniffligen Funktionen identifiziert und priorisiert hat, beginnt er mit einer Konzept-Betrachtung, macht Skizzen und Varianten. Er blendet die weniger kritischen Details aus und prüft erst einmal, wie die verschiedenen Räume und Installationen um einen Treppenlift herum passen könnten (das wäre die Auswahl von Funktionsprinzipien und ein Bauraumkonzept im echten Entwicklungsprojekt). Am Ende des ersten Bauabschnitts steht ein Gerüst, das den Umriss des Hauses zeigt, und es sind viele Schnüre gespannt, die die Innenraum-Aufteilung andeuten. In einem 3D-Modell kann ich bereits ein paar weitere Räume sehen. Für den Treppenlift gibt es eine Vorkonstruktion, aber in dem Aufbau reicht dafür zum jetzigen Zeitpunkt auch eine Leiter.

Die erste Bauphase beim klassischen Vorgehen besteht daraus, eine Baugrube auszuheben und die Grundplatte zu gießen – wobei hoffentlich alle Anschlüsse berücksichtigt werden. Also weit entfernt von dem Zwischenstand bei unserem fiktiven Hausbau. Zwar kann jeder der

drei Ingenieure mir konkrete Ergebnisse zeigen, die ich ansehen, ausprobieren und bewerten kann. Eigentlich eine gute Grundlage, um die nächsten Schritte zu vereinbaren. Alle Beteiligten sind sich sicher, einen guten Fortschritt erzielt zu haben und erwarten keine großen Probleme.

Building a house by a



Aber warum bleibt mein komisches Bauchgefühl bei der Vorstellung der Zwischenergebnisse? Möglicherweise, weil ich mir nach wie vor einige Fragen stelle: Werden die Einzelteile des Elektrikers am Ende zusammenpassen? Werden die Zimmer noch groß genug sein, wenn Baulücke und Wandstärken mit viel Vorsicht abgeschätzt sind? Sind alle Rohre und Leitungen verlegt? Wie kommt das Badezimmer en bloc in den ersten Stock? Und das sind nur eine kleine Auswahl der ungeklärten Fragen. Auf den Punkt gebracht: Hier ist das System Engineering auf der Strecke geblieben. Jede Fachdisziplin hat die Aufgabenstellung aus ihrer Sicht und mit ihren Erfahrungen interpretiert und ist sie mit einem anderen Fokus angegangen.

Wie soll aus den drei Teilen das Traumhaus entstehen?

Sicherlich hat jeder der drei aus seiner Warte die wichtigen Themen bearbeitet, eine gute Lösung gefunden und auch die Schnittstellen zu den anderen bedacht. Trotzdem kann ich als

Bauherr nach dem ersten Bauabschnitt nicht erkennen, wie aus den drei Teilen später mein Traumhaus werden soll.

Warum ist das Vorgehen so unterschiedlich? Entscheidend sind die Freiheitsgrade, Randbedingungen und die typischen Projektrisiken der drei Disziplinen:

- In der Elektronikentwicklung sind wir auf Standardkomponenten angewiesen, wenn es nicht ein Custom-ASIC mit allen Risiken und Aufwand werden darf. Hier geht es in den frühen Projektphasen darum, kritische Funktionen mit den ausgewählten Bauteilen nachzuweisen. Übliche Schaltungselemente des Gerätes, die wenig Risiko darstellen, können später ergänzt werden, ebenso Anpassungen an Kontur und Bauraum.
- Bei der mechanischen/mechatronischen Konstruktion sind Bauraum und Materialien/Fertigungsverfahren ein begrenzender Faktor für den Lösungsraum, entscheidend auch die später anvisierte Stückzahl. Daher wird hier sehr früh versucht, mit Bauraummodellen zu einer Aufteilung zu kommen, und auf ein produktionsgerechtes Design geachtet.
- Für die Software ist es sehr einfach, gegen virtuelle Schnittstellen zu arbeiten. Es ist oft einfacher, eine begrenzte Software Unit gegen Mocks zu implementieren als die Umsetzung aller Software Units gleichzeitig zu beginnen.

Sieht man diese unterschiedlichen Schwerpunkte gerade schon für die frühen Projektphasen, dann wird klar, dass es für eine gelungene agile Systementwicklung immer gemeinsame Etappenziele braucht. Etappenziele oder Integrationspunkte, die eine der offenen technischen Fragestellungen beantworten oder ein System-Feature umsetzen. Zu dem Integrationspunkt gehört auch die adäquate Festlegung des Reifegrads der Entwicklungs(zwischen)ergebnisse (Qualitätskriterien). Diese Ziellinie gilt es während des Projekts immer im Auge zu behalten.

Ein Beispiel aus der agilen Praxis bei Zühlke

Doch wie kann so eine Zielsetzung in der Praxis der agilen Systementwicklung aussehen? Schauen wir uns dazu ein Beispiel aus unserem Projektalltag bei Zühlke an: Bei der Entwicklung eines neuen Gerätes stand die Sensorik als kritisches Feature und Hauptfunktion für den Auftraggeber im Mittelpunkt. Wir mussten daher schon zu einem frühen Zeitpunkt zeigen, dass diese Funktionalität im notwendigen Umfang und unter den Anwendungsbedingungen erreichbar ist. Das Projektteam definierte dazu ein erstes Integrationsziel. Alle Disziplinen mussten ihren Teil beitragen, um diesen Funktionsnachweis zu erreichen. Wichtig war die Vereinbarung zu Scope und Qualität (im Sinne des Reifegrades der Umsetzung) für diesen Integrationspunkt, damit er frühzeitig erreicht werden konnte, aber auch aussagekräftig war. Das sind wieder die zwei zentrale Werte – Timeline und Kundennutzen – der agilen Entwicklung.

Als Zielsetzung für diesen Integrationsschritt wurde vereinbart, dass die „Rohdatenerfassung möglich ist“. Daraus ergab sich eine Priorisierung für die Vielzahl der Anforderungen, die vorrangig zu bearbeiten waren, während für andere Punkte bewusst Workarounds vorgesehen wurden:

- Die Analogsignale sollten möglichst gut erfasst werden und möglichst nah am späteren Produkt sein. Das bedeutete, im Aufbau bereits kurze Wege zur Verstärkerstufe zu ermöglichen und auch den Einsatz einer seriennahen Anlogschaltung. Andererseits mussten wir noch keine eigene Stromversorgung konzipieren, sondern konnten noch auf ein hochwertiges Labornetzteil zurückgreifen. Für die nachgelagerte Datenerfassung genügte ein Eval-Board der ausgewählten CPU, anstatt eines komplett ausentwickelten PCB.
- Die mechanische Konstruktion musste die Kontaktierung und Fixierung innerhalb des verfügbaren Bauraums erreichen und dabei Platz für die Anschlüsse der Elektronik lassen. Dafür konnten wir bei diesem Zwischenschritt die Größe und Kraft des Antriebs frei wählen. Darüber hinaus genügte es, wenn die Komponenten nur einige hundert Messungen durchhielten und defekte Teile im Aufbau notfalls schnell getauscht werden konnten.
- Die Steuerungssoftware musste einen Messablauf ermöglichen. Sie brauchte diesen allerdings nur mit fest eingestellten Parametern zu unterstützen. Die freie Konfigurierbarkeit von Parametern und Schwellen mittels PC-Software konnte in einem späteren Schritt erfolgen.
- Eine weitere Anforderung an die Software war es, die Datenerfassung und Auswertung in Echtzeit zu ermöglichen. Im ersten Schritt konnten wir dafür die Rohdaten zunächst im RAM zwischenspeichern – diese wurden erst in einem nachgelagerten Arbeitsgang ausgewertet.
- Die Bedienung des Aufbaus sollte über ein feststehendes Kommando-Interface erfolgen. Dafür genügte aber zunächst eine schnell verfügbare Remote-Konsole ohne GUI. Die GUI wurde erst später entwickelt.

Natürlich hätte man viele der zurückgestellten Anforderungen auch gleich umsetzen können, um es „von Anfang an richtig zu machen“. Aber dann wäre die Integration nicht nur bereits planmäßig viel später gewesen, es wären bis dahin auch erhebliche zusätzliche Projektrisiken aufgelaufen. Für uns hat es sich bewährt, im Rahmen unserer agilen Entwicklung möglichst häufig auf Systemebene zu integrieren und gegen Systemanforderungen, bzw. mit Systemabläufen zu testen. Auf dieser Abstraktionsebene ist es auch mit wenig Aufwand möglich, die Ziele und Inhalte der Integrationen früh genug zu definieren. Nach unserer Erfahrung erreichen wir so einen sehr gut vorhersagbaren Fortschritt und können vor allem während des Projektes noch flexibel auf Erkenntnisse und Änderungen reagieren.

Welche Methodik und Best Practices das im Alltag ermöglichen, darüber berichten wir immer wieder in unseren Blogs und auf Konferenzen, demnächst auf einem [Seminar mit dem Schwerpunkt Agiler Systementwicklung](#) für Medizin-Produkte .

Weitere spannende Blogbeiträge zum Thema:

- [Agile Methoden in der Systementwicklung](#)
- [Agile System Development](#)
- [Time, Budget und Innovation – dank agiler Systementwicklung](#)
- [Die Architektur der Vorreiter](#)