



□ Henning Wolf

(E-Mail: henning.wolf@it-agile.de)
 ist Geschäftsführer der it-agile GmbH in Hamburg. Er verfügt über langjährige Erfahrung aus agilen Softwareprojekten (XP, Scrum, Kanban) als Entwickler, Projektleiter und Berater. Er ist Autor der Bücher "Software entwickeln mit eXtreme Programming" und "Agile Softwareentwicklung". Henning Wolf hilft Unternehmen und Organisationen, agile Methoden erfolgreich einzuführen.



□ Arne Roock

(E-Mail: arne.roock@it-agile.de)
 arbeitet bei it-agile in Hamburg. Als studierter Germanist interessiert er sich für informative, leicht verständliche und kooperative Kommunikation. Außerdem beschäftigt er sich seit längerem mit den Themen Lean und Kanban sowie Selbstorganisation und Zeitmanagement in der IT.

Mit Feedback zum Erfolg

Wir alle wollen innovative Produkte oder Dienstleistungen entwickeln. Leider reicht es dafür nicht, eine gute Idee zu haben. Entscheidend ist darüber hinaus, dass wir uns schnell und regelmäßig Feedback auf unterschiedlichen Ebenen einholen. Der Artikel stellt die Wichtigkeit von Feedback heraus und beschreibt, was nötig ist, um Feedback konstruktiv zu nutzen.

Einleitung:

Was haben das Telefon, die Musikkassette und der Online-Dienst flickr gemeinsam? Bei allen dreien handelt es sich um Erfindungen, die ursprünglich zu einem ganz anderen Zweck gedacht waren, dann aber in ihrer jetzigen Form erfolgreich wurden. Bell wollte mit seinem Telefon ursprünglich Konzerte übertragen, die Musikkassette sollte eigentlich nur zum Diktieren von Briefen dienen und flickr startete als Feature eines Multiplayer-Online-Spiels, mit dem User ihre Fotos in einem Chatraum austauschen konnten. Die wahr-

ren Innovationen und damit der Erfolg entstanden hier nicht (nur) durch eine geniale Idee, sondern dadurch, dass die Erfindungen in innovativer Weise genutzt wurden und dass sich die Erfinder auf die geänderten Gegebenheiten eingelassen haben.

Feedback, Feedback, Feedback

Was können wir daraus lernen? Natürlich möchten wir auch innovative Produkte oder Dienstleistungen entwickeln. Wer hätte nicht gern Twitter oder den iPod erfunden? Was wir dazu (neben einer Idee) vor allem brauchen, ist Feedback. Denn durch

Feedback erhalten wir das Wissen, das wir dringend benötigen, um unser Produkt in die richtige Richtung zu entwickeln.

Theoretisch kann man sich dieses Wissen natürlich auch auf anderem Wege beschaffen, z. B. indem man sämtliche vorhandene Literatur studiert, gründlich nachdenkt, dann den Plan für das perfekte Produkt aufstellt und mit der Umsetzung anfängt. Bei dieser "wissenschaftlichen Methode" (die ja in der Realität gar nicht so selten anzutreffen ist) gibt es allerdings zwei Probleme: Zum einen bleiben wir dabei immer in unserer eigenen Gedankenwelt gefangen und stellen selten die eigenen Annahmen infrage. Zum anderen bleibt uns gar nicht genug Zeit, um langfristige Pläne aufzustellen und umzusetzen, denn wir haben es mit einem beweglichen Ziel zu tun.

Auch wenn heute beinahe jede Webanwendung über eine Social Media-Funktion und fast jedes elektronische Gerät über einen USB-Anschluss verfügen muss, um dem Standard zu genügen, heißt das noch lange nicht, dass dies in zwei Jahren immer noch so ist. Wer hat schon das schnelle Ende der VHS-Technologie vorhergesehen? Und wer kann schon sicher sagen, ob Twitter nächstes Jahr immer noch zu den meist genutzten Webanwendungen gehört?

Wir brauchen also schnelles Feedback – und zwar zumindest in drei Varianten:



Abb. 1: Das Telefon war ursprünglich nicht zum Telefonieren gedacht.

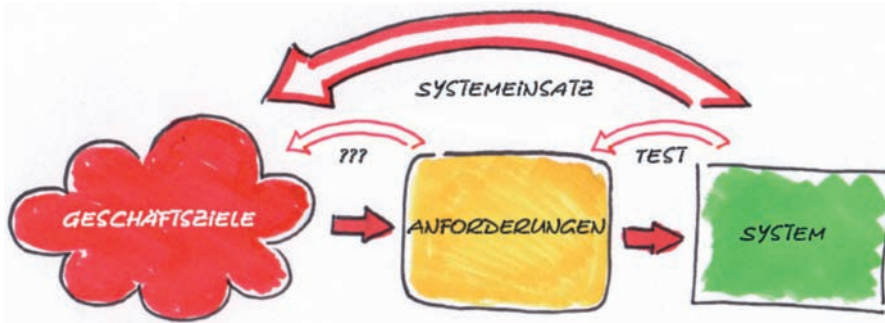


Abb. 2: Das System lässt sich gegen die Anforderungen testen. Aber wie wissen wir, ob die Anforderungen auch zu unseren Geschäftszielen passen? Nur durch schnellen Systemeinsatz!

Feedback mit dem System selbst, mit den Endanwendern und mit den Teammitgliedern.

Feedback mit dem System

Unser System selbst kann uns wichtiges Feedback liefern, wenn es erst einmal läuft. Ist die Infrastruktur ausreichend, die wir uns aufgebaut haben? Halten unsere Server den Nutzerzahlen stand? Bleibt die Geschwindigkeit auch im Produktiveinsatz akzeptabel? Was brauche ich an Support? Was kostet mich der Betrieb – und was muss ich auf der anderen Seite einnehmen, um profitabel zu arbeiten?

Hierfür ist es wichtig, dass wir uns das System in “der freien Wildbahn” ansehen, also im Produktiveinsatz – und nicht in unserer Testumgebung. Und wir tun gut daran, so schnell wie möglich mit einer ersten Version *live* zu gehen. Denn falls die Kosten für den Betrieb unterschätzt wurden oder eine unpassende Infrastruktur gewählt wurde, möchten wir das so schnell wie möglich merken, damit wir noch rechtzeitig gegensteuern können.

Vielleicht entsteht so auch die unangenehme Erkenntnis, dass unser System sich nicht rechnet und wir es beerdigen müssen. Aber auch das sollten wir so früh wie möglich erkennen, um nicht noch mehr Zeit und Geld zu investieren. Wenn schon scheitern, dann wenigstens früh!

Feedback von den Endanwendern

Ganz egal, ob wir eine neue Webanwendung erstellen oder ein Inhouse-System zur Mitgliederverwaltung: Die Endanwender können am Besten beurteilen, was sie wirklich brauchen und was nicht. Die Endanwender entscheiden über den Erfolg oder Misserfolg jedes Systems. Sollten wir an den Anwendern vorbei entwickeln, ist

unser System zum Scheitern verurteilt. Wir tun also gut daran, uns möglichst früh und möglichst häufig von unseren Endanwendern Feedback zu holen – und zwar durch den Einsatz des produktiven Systems (bzw. eines Teils davon).

Wenn wir es schaffen, den Endanwendern früh eine Basisversion zur Verfügung zu stellen und wenn wir gleichzeitig in der Lage sind, uns Feedback von den Anwendern zu holen, dann ist dies mehr Wert als 10 Jahre Marktforschung. Denn auf diese Weise werden unsere Annahmen, wie das fertige System aussehen sollte, schnell bestätigt oder widerlegt - beides ist positiv zu bewerten.

Von den Anwendern erhalten wir die wichtigsten Informationen: Wie gut ist das

System bedienbar? Wo hakt der Workflow? Welche Features werden besonders oft benutzt und welche nicht? Welche weiteren Features werden gewünscht?

Durch dieses Anwender-Feedback erhalten wir oft unerwartete Erkenntnisse hinsichtlich der weiteren Ausgestaltung unseres Produktes: Was wir für unwichtig hielten, sollten wir oftmals schnell ganz nach oben priorisieren. Was in unserer Vorstellung der Renner war, erweist sich oftmals als nutzlos und woran wir vorher gar nicht gedacht hatten, wird zum Top-Feature.

Nur wenn wir uns dieses Feedback einholen, wenn wir es ernst nehmen und wenn wir bereit und in der Lage sind, daraus die nötigen Konsequenzen zu ziehen, werden uns innovative Produkte oder Dienstleistungen gelingen.

Feedback von den Teammitgliedern

In vielen Teams möchten die Entwickler (mit gutem Recht) nicht mehr “Programmierer” genannt werden. Denn sie tun viel mehr als zu programmieren: Sie schätzen, entwerfen, testen, dokumentieren und sie beschäftigen sich mit der Architektur und helfen bei der Beschreibung und dem richtigen Zuschnitt von Anforderungen.

Für innovative Produkte ist deshalb die Feedback-Schleife mit dem Entwicklungsteam von entscheidender Bedeutung. Durch dieses Feedback können wir immer über die Entwicklungsgeschwindigkeit und



Abb. 3: Wir müssen uns Feedback darüber einholen, wie Anwender mit unserem System umgehen, denn letztlich entscheiden immer sie über Erfolg oder Misserfolg.

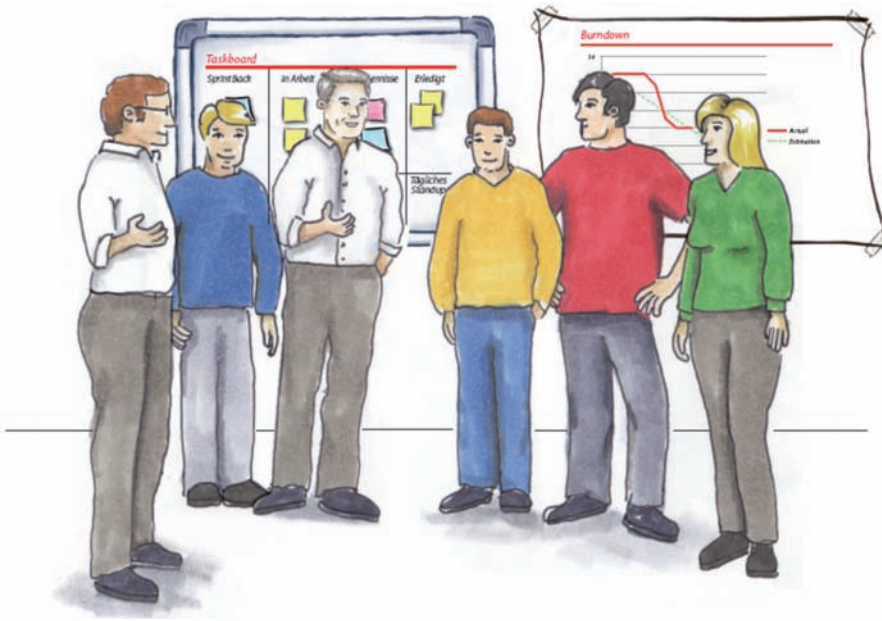


Abb. 4: Entwickler liefern wichtiges Feedback, z. B. durch tägliche Standup-Meetings oder durch die Pflege gut sichtbarer Taskboards und Burndown-Charts.

die Aufwände für aktuelle und eingeplante Aufgaben auf dem Laufenden sein. So wird eine realistische Releaseplanung möglich.

Darüber hinaus brauchen wir den Input des Teams, um Aufwände zu schätzen und erfolgreiches Risikomanagement zu betreiben. So können wir gewährleisten, dass die riskantesten Anforderungen möglichst früh umgesetzt werden und dass der Entwicklungsaufwand stets im vernünftigen Verhältnis zum Geschäftswert steht.

Und natürlich kommen vom Entwicklungsteam wichtige Ideen für unser neues System – sofern wir denn ein offenes Ohr dafür haben.

Was dafür nötig ist

Die Vorteile von vielen, schnellen Feedback-Schleifen sind also bestechend – sie sind aber leider nicht umsonst zu haben! Denn wir wollen unser System ja so erstellen, dass wir in sehr kurzen Zyklen immer wieder neue, produktionsreife Teile dazu bekommen. Gleichzeitig wollen wir immer wieder die Möglichkeit haben, neu zu priorisieren, welche Features als Nächstes umgesetzt werden sollen.

Das bedeutet zum einen, dass wir unsere Architektur so wählen müssen, dass unser System dauerhaft ohne Zusatzaufwände erweitert werden kann. Ein Feature darf also in einem halben Jahr nicht wesentlich teurer sein als heute – ganz egal, welche anderen Features in der Zwischenzeit ent-

wickelt wurden (*flachen Aufwandskurve*).

Im Gegensatz dazu haben klassisch entworfene und entwickelte Systeme meist eine exponentiell wachsende Aufwandskurve, die umso steiler ansteigt, je mehr neue Funktionen später hinzugefügt werden, die zu Beginn der Entwicklung noch nicht bekannt waren. Flache Aufwandskurven lassen sich nur mittels Refactoring erreichen – also dem stetigen Umbau des



Abb. 5: Muss die Lösung wirklich immer in der schnellen (und teuren) „Autobahn“ bestehen? Oder reicht nicht auch der einfache „Sandweg“, etwa wenn ein Feature nur einmal im Monat verwendet wird?

Systems zu einer weiterhin erweiterbaren und änderbaren Architektur.

Zum anderen wollen wir die neuen Systemteile ja immer wieder *produktiv* stellen und nicht nur in einem Testsystem begutachten. Also benötigen wir automatisierte Tests, damit nicht bei jeder Erweiterung umständlich manuell getestet werden muss. Und wir brauchen ein Buildsystem, das so weit automatisiert ist, dass wir praktisch jederzeit „auf Knopfdruck“ ein neues Release erstellen können.

Darüber hinaus muss es uns gelingen, die Anforderungen so auszuwählen und zuzuschneiden, dass sie auch tatsächlich in diesen kurzen Entwicklungszeiträumen vollständig umgesetzt werden können und dabei immer einen Mehrwert für das Gesamtsystem erbringen.

Dies ist alles andere als trivial und erfordert häufig ein Umdenken. In der Regel benötigt ein Produktmanager (oder Product Owner) Unterstützung und viel Erfahrung, bis ihm dies gelingt.

Neben diesen architekturellen und technischen Aspekten sind jedoch auch weiche Faktoren entscheidend: Denn wir müssen bereit sein, Feedback von verschiedenen Seiten anzunehmen und konstruktiv damit umzugehen – auch wenn es häufig unangenehm sein mag und unsere bisherigen Pläne komplett zunichte macht. Dies stellt vielleicht die größte Hürde dar.

Einwände

Man mag einwenden, dass ein solches Vorgehen nicht planbar sei. Das stimmt in letzter Konsequenz auch, wenn man unter Planbarkeit die vollständige Festlegung eines Funktionsumfangs sowie zeitlicher und finanzieller Ressourcen versteht. Echte Innovationen sind in diesem Sinne auch nicht planbar. Allerdings können wir sehr wohl Projekte und Produkte so planen, dass wir ihre Erfolgswahrscheinlichkeit erhöhen, indem wir häufig Softwareversionen ausliefern und wir uns immer wieder Feedback dazu einholen.

Natürlich können wir die Entwicklung auch für zwei bis drei Monate oder sogar länger planen. Wir müssen nur flexibel ob des genauen Inhalts bleiben, wenn wir uns die Zielerreichung und damit den Erfolg nicht verbauen wollen. Die Flexibilität bzgl. der Anforderungen führt zu qualitativ hochwertigen Anwendungen, die auch weiterhin erweiterbar und änderbar bleiben und sich an die sich schnell ändernde Umwelt anpassen.

Auch innerhalb von Unternehmen ist die Entwicklung von internen Anwendungen nach diesem Muster möglich. Es kann zwar notwendig sein, dass man die bisherigen Umfeldvorbereitungen, wie Schulungen und interne Genehmigungsverfahren, drastisch abkürzen muss, dafür erhält man aber eine Flexibilität der Reaktion auf neue Marktgegebenheiten, die ihresgleichen sucht.

Fazit

Wir haben heute die Möglichkeit, Softwareentwicklung mit schnellem Feedback zu steuern, um so zu erfolgreicherem und innovativeren Produkten zu kommen.

Dafür ist es oft nötig, umzudenken und sowohl technische als auch organisatorische Änderungen zu vollziehen. Solche Änderungen fallen niemandem leicht, aber sie lohnen sich! flickr wäre mit Sicherheit nicht so erfolgreich geworden, wenn die Gründer nicht willens und in der Lage gewesen wären, konstruktiv mit der unerwarteten Nutzung der ursprünglichen Software umzugehen und das Feedback der Anwender zu nutzen, um ein wirklich innovatives Produkt zu erstellen.

Softwareentwicklung mit schnellen Feedback-Zyklen kann man selbst mit der eigenen Mannschaft etablieren – oder man holt sich Unterstützung beim Spezialisten. So

bietet it-agile unter dem Namen *Feedback Force* genau solche Entwicklungsteams und fachliche Unterstützung, um Innovationen zu fördern und sich schnell anzupassen.

Das schöne bei dieser Art der Entwicklung: Sie können mit wenig Risiko und Aufwand beginnen, bekommen schnelles Feedback und werden schnell sehen, dass es sich für Sie lohnt!

Mehr Informationen zu Softwareentwicklung mit schnellen Feedback-Zyklen finden Sie unter: www.it-agile.de.

Auf Ihr Feedback zu diesem Artikel freuen sich:

henning.wolf@it-agile.de

arne.roock@it-agile.de