

# IT-PROJEKTE SICHER LEITEN – WAS LEISTET AGILES PROJEKTMANAGEMENT?

Globalisierte Wirtschaftsstrukturen, verschärfte Konkurrenz, beschleunigte Produktionszyklen sowie der Wandel vom Anbieter- zum Käufermarkt stellen neue Herausforderungen an das Management von IT-Projekten. Gefragt sind eine große Änderungs- und Anpassungsfähigkeit an Kundenwünsche und die Kompetenz, flexibel und schnell neue Produkte entwickeln und integrieren zu können. Veränderungen während des Entwicklungsprozesses schaffen jedoch ein hohes Maß an Unsicherheit im Projektumfeld – zumal die strikte Einhaltung von Budget- und Terminvorgaben unerlässlich ist. Der Artikel stellt agiles Projektmanagement im Vergleich zu traditionellen Vorgehensweisen vor und zeigt, wie in IT-Projekten Unsicherheit und Unbestimmtheit produktiv eingesetzt werden können, um außergewöhnliche Erfolge zu erzielen.

## Kundenzentrierte Produktentwicklung

Innovative Produkte werden heute in enger Rückkopplung mit den Kunden gestaltet. Der Kunde wird vom Randfaktor zum Mittelpunkt der Entwicklung neuer IT-gestützter Systeme. Die korrespondierenden Prozesse entstehen nicht mehr vorab am Reißbrett, sondern werden parallel als Teil des Produkts entwickelt. War es in der Vergangenheit ausreichend, die internen Prozesse zu analysieren und durch IT-Unterstützung effizienter zu machen, können Wettbewerbsvorteile heute nur erreicht werden, wenn Produkte und IT-Systeme kundenzentriert entwickelt werden.

Eine kundenzentrierte Produktentwicklung bedeutet aber, dass Veränderungen der Anforderungen während der Projektlaufzeit nicht länger die Ausnahme sind, sondern zur Regel werden. Aber lassen sich IT-Projekte in solch dynamisierten Umfeldern überhaupt noch budgetieren? Versagen bei sich ständig ändernden Anforderungen nicht sämtliche Steuerungs- und Kontrollmechanismen des uns vertrauten Projektmanagements?

Dass man angesichts von Unsicherheit, schemenhaften Anforderungen und komplexen Kundenwünschen erfolgreich Produkte entwickeln kann, zeigen diverse Projekte global agierender Unternehmen. Prominentes Beispiel ist die 1-Click-Bestellung von Amazon. Der weltgrößte Buchhändler, wie auch Google oder Yahoo, verwendet dabei stets den gleichen Trick: Er stellt dem Endkunden in sehr frühen ersten

Tests eine minimale Produktfunktionalität zur Verfügung. Die Kommentare und Anregungen der Kunden dazu werden übernommen und gehen in die Anforderungsdefinition der nächsten Entwicklungsschritte ein. Die Kunden werden so zu Produktmanagern und Ideengebern. Auf diese Weise sind die genannten erfolgreichen Unternehmen sicher, genau die Produkte zu entwickeln, die der Kunde wünscht. Und zudem werden kostenintensive Fehlentwicklungen vermieden.

Diese Herangehensweise steht im deutlichen Gegensatz zu den Möglichkeiten, die klassisches Projektmanagement in seiner



Christian Setzwein (E-Mail: christian.setzwein@setzwein.com) ist Turnaround-Manager und Leiter internationaler Großprojekte in unterschiedlichsten Branchen. Er ist Geschäftsführer der Setzwein IT-Management GmbH.



Eicke Bücking (E-Mail: eicke.buecking@dv-ratio.com) ist als Berater und Projektmanager für die DV-RATIO tätig und spezialisiert auf den Bereich der JEE-Anwendungsentwicklung.

Abfolge linear gekoppelter Entwicklungsschritte (Phasenmodell) bietet. Denn dort steht das Produkt erst am Ende des Entwicklungsprozesses – nach Abschluss aller anderen Phasen – für ein Kunden-Feedback bereit.

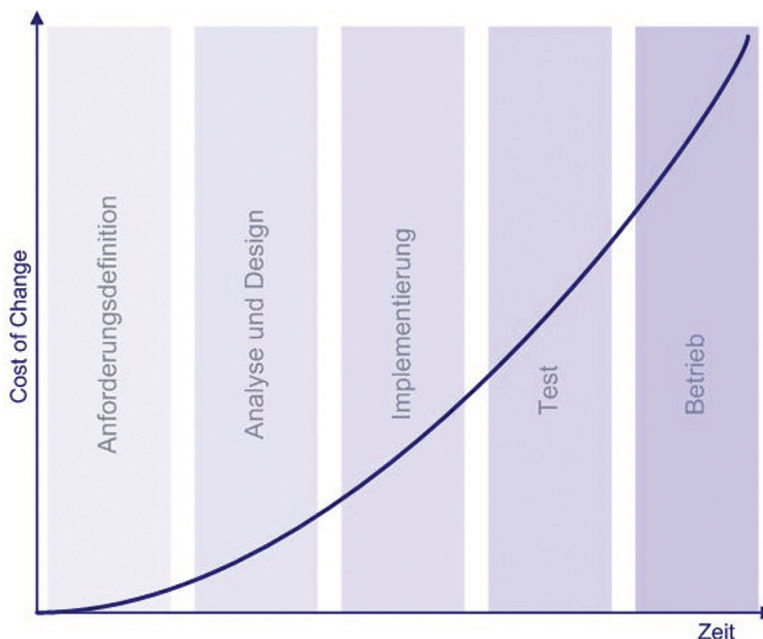
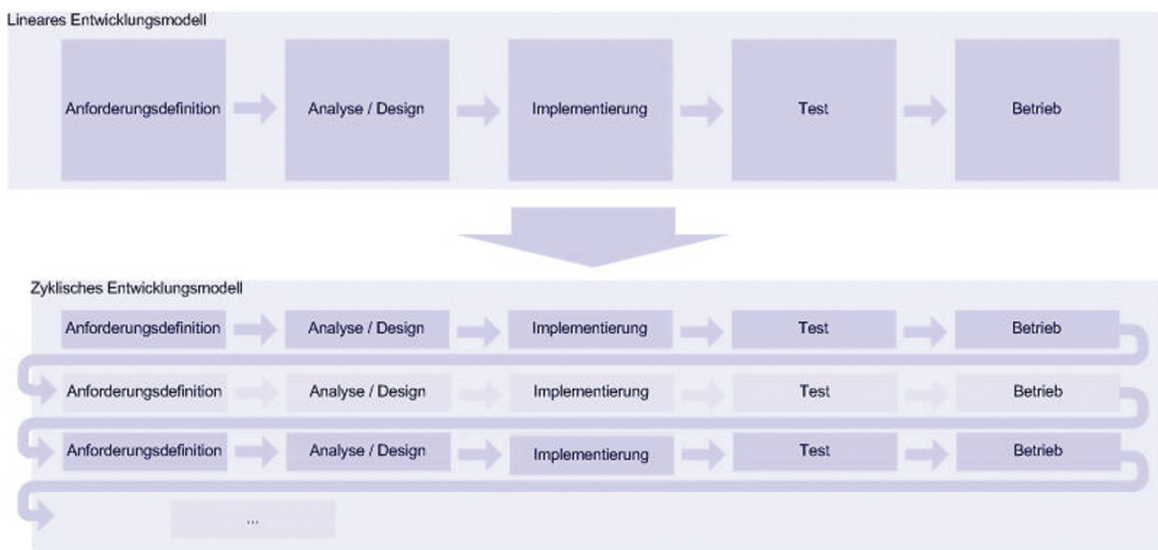


Abb. 1: „Cost of Change“



Statt schwerer linearer Phasen viele leichtgewichtige Iterationen

Abb. 2: Statt schwerer linearer Phasen viele leichtgewichtige Iterationen

### Neue Anforderungen an das IT-Management

Aus der wachsenden Kundenzentrierung und der Notwendigkeit, Kundenwünsche in den laufenden Entwicklungsprozess zu integrieren, ergeben sich zwangsläufig neue Herausforderungen an die Methoden des IT-Projektmanagements. Gewährleistet werden müssen:

1. die schnelle Lieferung von funktionsfähigen Produkten, die vom Kunden frühzeitig genutzt werden können,
2. die Möglichkeit zur Veränderung der Anforderungen an das Produkt während der Projektlaufzeit,

3. die Minimierung der Kosten für kundenspezifische Anpassungen,
4. die zuverlässige Budgetierung des Projekts, obwohl die Anforderungen nur schemenhaft bekannt sind.

Klassisches und agiles Projektmanagement gehen mit diesen Herausforderungen unterschiedlich um. Beide Methoden stimmen in der Annahme überein, dass die Kosten für Veränderungen steigen, je später diese im Projekt durchgeführt werden. Veränderungen während der Anforderungsdefinition sind deutlich weniger kostenintensiv als während der Testphase oder des Betriebs (siehe Abb. 1).

### Paradoxe Prozesse im klassischen Modell

In der Praxis des klassischen Projektmanagements zieht man daraus den Schluss, möglichst alle Veränderungen ausschließlich in der Phase der Anforderungsdefinition zuzulassen. Durch genaue Planung und möglichst detaillierte Anforderungsanalyse sollen Unbestimmtheiten und Änderungen in späteren Phasen vermieden werden. Diese auf den ersten Blick einleuchtende Vorgehensweise bedeutet jedoch, dass für diese Phase viel Zeit benötigt wird. Dies widerspricht dem Ziel, möglichst schnell lauffähige Systeme für die Kunden zur Verfügung zu stellen (1).

Feature List	
<input checked="" type="checkbox"/>	Auswahl der möglichen Kalkulationsarten
<input checked="" type="checkbox"/>	Kalkulation von einfachen Leasingraten
<input type="checkbox"/>	Ausdruck der Kalkulation
<input type="checkbox"/>	Berechnung der Kfz-Steuer
<input type="checkbox"/>	Änderung der Laufzeit
<input checked="" type="checkbox"/>	Kalkulation mit Wartungsservice
<input checked="" type="checkbox"/>	Kalkulation mit Reifenservice
<input checked="" type="checkbox"/>	Kalkulation mit Versicherung
<input type="checkbox"/>	Kalkulation mit GEZ
<input type="checkbox"/>	Vorgabe des Kalkulationszins
<input checked="" type="checkbox"/>	Online-Hilfe

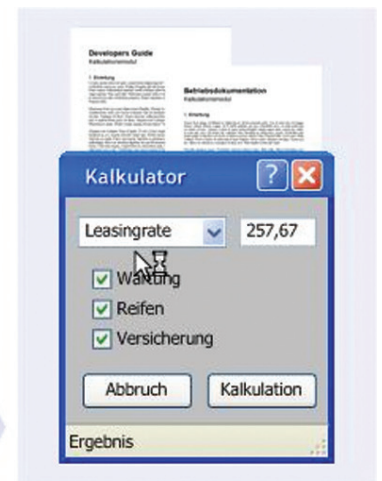


Abb. 3: Von der Feature-Liste zum Produkt

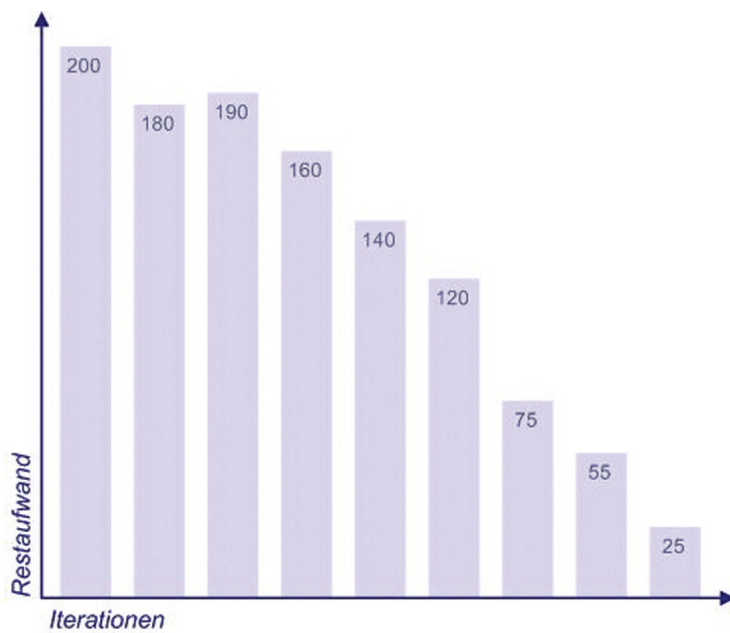


Abb. 4: „Burndown Chart“

Änderungswünsche in den folgenden Phasen werden durch ein *Change-Request-Verfahren* (CR) abgewickelt. Das kann bedeuten, dass Änderungen durch die aufwändigen Prozeduren für CRs womöglich gar nicht zugelassen werden, was der Anforderung widerspricht, auf Änderungswünsche der Kunden individuell zu reagieren (2). In diesem Fall werden dann zumeist notwendige Änderungen in einem Folgeprojekt durchgeführt. Das erzeugt versteckte Projektkosten – die Kosten werden ja nicht mehr dem eigentlichen Projekt zugerechnet – und widerspricht dem Ziel einer sauberen Budgetierung (4). Zudem sind die Kosten für die Veränderung überproportional teuer, weil sie sehr spät durchgeführt werden (3).

Werden CRs hingegen innerhalb des Projekts zugelassen, so werden sie fast immer erst zum Projektende erkannt und implementiert. Im klassischen Kontext werden CRs deshalb so spät erkannt, weil das fast lauffähige System dem Kunden für gewöhnlich erst im Szenario eines Abnahmetests zur Verfügung gestellt wird. Die Bearbeitung von CRs am Projektende widerspricht dem Ziel, die Kosten für Veränderungen möglichst gering zu halten (3). Außerdem treten Kosten für CRs in derartigen Konstellationen als zusätzliche Kosten auf, was wiederum dem Ziel einer sauberen Budgetierung entgegenläuft (4).

Wo klassisches Projektmanagement es nicht schafft, die vier Herausforderungen miteinander zu vereinbaren, gelingt es den

agilen Methoden, alle Ziele unter einen Hut zu bringen.

**Produktive Unbestimmtheit: das Unplanbare nutzen**

Im Mittelpunkt agiler Vorgehensweisen steht das Prinzip, veränderte Anforderungen durch den Kunden in jedem Fall zuzulassen (2) – „embrace change“, wie Kent Beck es bezeichnet (vgl. [Bec04]). Anstatt von unverrückbaren Vorgaben auszugehen, erkennt man an, dass die Anforderungen an das zu entwickelnde System am Projektanfang nur schemenhaft bekannt sind. Ein eingängiges Beispiel für eine extreme Form produktiver Unbestimmtheit ist die Bilderplattform „Flickr“. Ursprünglich wollte das kleine kanadische Unternehmen um Catarina Fake ein web-basiertes Multiplayer-Spiel entwickeln. Nur am Rande bot man die Möglichkeit an, Bilder im Web hochzuladen. Diese Option wurde von den Kunden jedoch so intensiv genutzt, dass man das zunächst geplante Konzept fallen ließ und zu einem neuen, außerordentlich erfolgreichen Geschäftsmodell gelangte.

Eine unbestrittene Stärke agiler Entwicklungsmethoden ist also der flexible Umgang mit Veränderungen. Doch wie werden die anderen Ziele – Budgetsicherheit, Kostenminimierung für wechselnde Anforderungen und frühzeitige Bereitstellung von Funktionalität – realisiert? Wie ist es möglich, ein Budget festzulegen, wenn die Anforderungen nur sche-

menhaft bekannt sind? Wie kann man schnell zu einem lauffähigen Produkt kommen, wenn doch die Anforderungen an das Produkt ständig verändert werden dürfen? Und sind die ständigen Änderungen nicht sowieso zu teuer?

**Steuerung durch Iterationen und Feature-Listen**

Grundlegend für die Methodik des agilen Projektmanagements (zur Einführung vgl. [Hig04], [Pic05]) ist es, die Phasen Anforderungsanalyse, Design, Implementierung, Test und Betrieb nicht nur ein einziges Mal zu durchlaufen, wie es im klassischen Projektmanagement der Fall ist. Man durchläuft die Phasen stattdessen mehrfach in so genannten Iterationen (siehe Abb. 2). Die typische Dauer einer Iteration beträgt dabei zwei bis sechs Wochen.

Das zentrale Instrument zur Steuerung der Anforderungen ist die Feature-Liste. Sie enthält eine Kurzbeschreibung der geplanten Funktionalität. Durch Hinzufügen und Weglassen von Features in der Liste kann der Veränderung an die Anforderungen eines Produkts Rechnung getragen werden (siehe Abb. 3).

Eine initiale Feature-Liste wird zu Beginn eines Projekts festgelegt. In Zusammenarbeit mit dem Kunden, zumeist in mehrtägigen Workshops, werden die Features besprochen. Ein gemeinsames Verständnis der Funktionalität wird erarbeitet und die Features werden vom Kunden priorisiert. Alle Features der initialen Liste werden geschätzt und das Budget für das Gesamtprojekt wird aufgrund dieser Schätzung festgelegt. In komplexen Fällen wird die Schätzung durch Probeimplementierungen in Teilbereichen abgesichert und erst auf dieser abgesicherten Basis budgetiert (zum Verfahren agiler Schätzung und Budgetierung vgl. [Coh05]). Die Phase einer durch umfangreiche Dokumente gestützten langen Anforderungsdefinition wird im Vergleich zum klassischen Projektmanagement wesentlich abgekürzt. Allein dies minimiert die Kosten späterer Veränderungen deutlich (3).

Im weiteren Projektverlauf wird zu Beginn jeder Iteration überprüft, welche Features implementiert werden sollen. Dies geschieht durch Umpriorisierung durch den Kunden oder durch das Hinzufügen neuer Features, die erst im weiteren Projektverlauf in den Blick geraten sind. Die



Methodik gewährleistet damit vor jeder Iteration das Zulassen von Veränderungen (2). Budgetsicherheit wird dadurch garantiert, dass für neu hinzukommende Anforderungen diejenigen Features aus der Liste gestrichen werden, die vom Kunden am niedrigsten priorisiert wurden (4). Das klassische CR-Verfahren wird somit durch ein agiles *Exchange-Request*-Verfahren ersetzt. Im Gegensatz zum klassischen CR-Verfahren werden durch vereinfachte Regularien Verwaltungsaufwände reduziert (3).

Das Ergebnis jeder Iteration ist ein lauffähiges IT-System, das die zu ihrem Beginn festgelegten Features implementiert. Das Ergebnis schließt Test und notwendige Dokumentation mit ein. Kosten entstehen nur für Anforderungs- und Designdokumentation und Tests von Features, die tatsächlich benötigt werden (3).

Bereits nach der ersten Iteration steht ein lauffähiges System zur Verfügung. In einigen agilen Projekten reicht diese erste Version bereits aus, um vom Kunden genutzt werden zu können. In jedem Fall wird dem Kunden nach möglichst wenigen Iterationen ein Release zur Verfügung gestellt (2).

### Sicherheit durch Agilität

Dadurch, dass der Kunde nach jeder Iteration die lauffähige Anwendung prüft, bemerkt er notwendige Interventionen rechtzeitig. Durch die andauernde Prüfung der Ergebnisse nach jeder Iteration erhält

er zudem genau das, was er am Projektende wirklich möchte – und nicht das, was er dachte, am Projektanfang haben zu wollen. Die Vermeidung der Entwicklung von nicht benötigter Funktionalität birgt ein enormes Potenzial zur Kosteneinsparung.

Zusätzliche Sicherheit schafft agiles Projektmanagement durch das Prinzip der Messbarkeit von Status und Projektfortschritt (vgl. [Sch04]). Die Feature-Liste und der tägliche Arbeitsfortschritt werden gegenüber dem Kunden transparent gehalten. Tägliche Restaufwandsschätzungen und deren Darstellung in *Burn-Down Charts* (vgl. [Pic05]) lassen Probleme im Projekt frühzeitig sichtbar werden (siehe Abb. 4). Dadurch bleiben stets alle Handlungsoptionen offen und es kann schnell und zielgerichtet interveniert werden.

Tägliches Kompilieren und *Deployment* auf die Zielplattform sowie automatisches Testen bringen unablässig den tatsächlichen Zustand des Systems und des Projekts ans Licht. Viele Unternehmen – darunter Weltmarktführer in ihren Domänen, wie z.B. SAP oder Siemens – setzen heute Methoden des agilen Projektmanagements ein. Einige von ihnen nutzen dabei zur Herstellung von Transparenz eine sehr anschauliche Technik: Ein defektes System wird durch eine rote Lampe und ein lauffähiges, automatisch korrekt getestetes System durch eine grüne Lampe für alle Mitarbeiter im Unternehmen sichtbar gemacht. Sichtbarkeit und Messbarkeit

sorgen angesichts variabler Anforderungen und wachsender Komplexität für ein hohes Maß an Sicherheit im Projektumfeld.

### Fazit

Agiles Projektmanagement bietet die Möglichkeit, innovative Produkte kundenindividuell herzustellen: im vorgegebenen Zeit- und Kostenrahmen und mit den gewünschten Features in höchster Qualität. Was hierbei zunächst als Unsicherheitsfaktor erscheint – das Zulassen von Veränderungen während des Entwicklungsprozesses – erweist sich bei genauerem Hinsehen als große Stärke und Überlegenheit agiler Methoden gegenüber klassischen Ansätzen. Denn gerade durch die flexible Methodik im Umgang mit Komplexität und Veränderung wird Sicherheit geschaffen. ■

### Literatur

- [Bec04] K. Beck, *Extreme Programming Explained*, Addison-Wesley, 2004
- [Coh05] M. Cohn, *Agile Estimating and Planning*, Prentice Hall, 2005
- [Hig04] J. Highsmith, *Agile Project Management*, Addison-Wesley, 2004
- [Pic05] R. Pichler, *Agiles Projektmanagement: eine Einführung*, in: *OBJEKTSpektrum* 1/05
- [Sch04] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, 2004