

JETZT MIT **it**fokus

**NEU**

10 EURO

itmanagement



# itmanagement

OKTOBER 2008

www.it-daily.net



**Agiles Qualitätsmanagement**

# Ewige Veränderung

Itverlag



SONDERDRUCK FÜR

**andrena**  
OBJECTS



## Agiles Qualitätsmanagement

# Ewige Veränderung

*Gegenstand des Artikels ist eine Verbindung des Besten aus zwei Welten: den Vorzügen eines ganzheitlichen, normierten Qualitätsmanagements mit den Vorteilen agiler Verfahren zur Softwareentwicklung.*



Agile Verfahren sind noch relativ jung. Sie beruhen auf einfachen, leichtgewichtigen Methoden, die im Gegensatz zu schwergewichtigen Prozessen wie dem Rational Unified Process (RUP) oder dem V-Modell die Dynamik in Softwareentwicklungsprojekten sowie die Anpassung an sich ständig ändernde Rahmenbedingungen und Anforderungen adressieren. Der Change Request wird nicht als Ausnahmesituation behandelt, die das Projekt immer wieder vor große Herausforderungen stellt, sondern als Standardsituation akzeptiert und als solche methodisch integriert.

Die Problematik zu schwergewichtiger und damit letztlich ungeeigneter Entwicklungsprozesse gilt auch für

gängige, fortwährende und alle Bereiche eines Unternehmens erfassende Tätigkeit, die dazu dient, Qualität als Systemziel einzuführen und dauerhaft zu garantieren. TQM wurde in der japanischen Autoindustrie eingesetzt und schließlich zum Erfolgsmodell gemacht. Zu den wesentlichen Prinzipien der TQM-Philosophie zählen die Folgenden:

- Qualität orientiert sich am Kunden.
- Qualität wird mit Mitarbeitern aller Bereiche und Ebenen erzielt.
- Qualität umfasst mehrere Dimensionen, die durch Kriterien operationalisiert werden müssen.
- Qualität ist kein Ziel, sondern ein Prozess, der nie zu Ende geht.
- Qualität bezieht sich nicht nur auf

Qualitätsmanagementsysteme in der IT: die Dynamik des Software Engineerings, neue Sprachen, neue Arbeitsmethoden, neue Konzepte, kurzum: die Schnelligkeit der Innovationszyklen erfordert leichtgewichtige, adaptive Verfahren in dieser jungen Ingenieursdisziplin.

Gegenwärtig rückt die agile Vorgehensweise mehr und mehr in den Blickpunkt. Nicht nur kleine Unternehmen sondern auch die Großen der Branche wie SAP und Microsoft setzen zunehmend auf agile Verfahren.

## Total Quality Management

Total Quality Management (TQM) ist ein grundlegender Ansatz zur Qualitätssicherung. Er bezeichnet die durch-

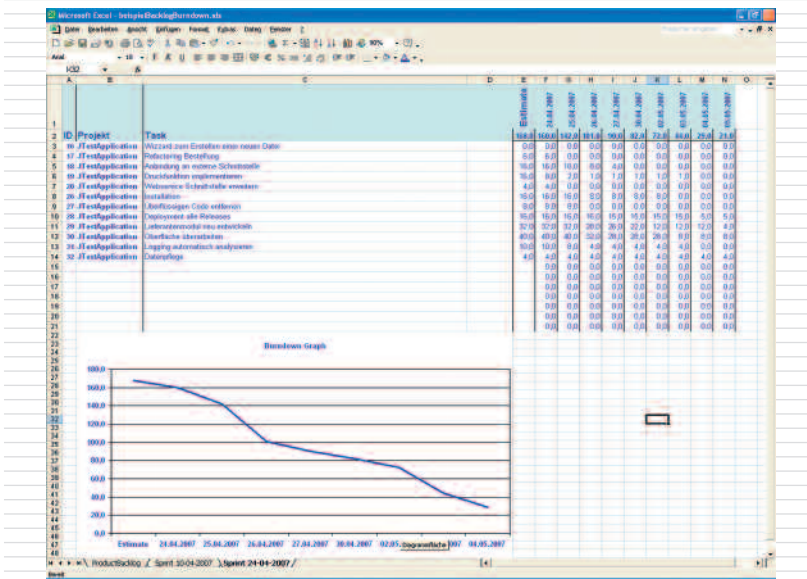
Produkte, sondern auch auf Dienstleistungen.

- Qualität setzt aktives Handeln voraus und muss erarbeitet werden.

## Aspekte der Qualität

Je nach Sicht und Rolle des Beteiligten können im Software Engineering verschiedene Aspekte von Qualität unterschieden werden. Aus Sicht des Kunden sind Einhaltung des Budgets, termingerechte Fertigstellung, Nachhaltigkeit und Sicherung des ROI zu nennen, aus Sicht des Anwenders Fehlerfreiheit, Performance und Bedienerfreundlichkeit (äußere Qualität).

Im Lebenszyklus von Software fallen etwa 80 Prozent der Kosten für Änderungen und Erweiterungen im An-



Quelle: andrena objects ag

**Bild:** Sprint Backlog und zugehöriger Burndown Chart.

schluss an die Primärentwicklung an. Software lebt also, wird immer wieder an neue Erfordernisse angepasst. Wie gut dies gelingt, hängt also maßgeblich von der inneren Struktur beziehungsweise inneren Qualität ab, der Verständlichkeit, Wartungs-, Erweiterungs- und Änderungsfreundlichkeit. Die Nachhaltigkeit und die Sicherung des ROI sind stark mit diesen Aspekten verknüpft. Genau das ist häufig das zentrale Problem: der Kunde sieht die innere Qualität nicht. Die aus mangelnder innerer Qualität resultierenden Probleme werden erst nach geraumer Zeit spürbar, falls sich zum Beispiel die Kosten für einfache Änderungen stark erhöhen oder Anpassungen gar nicht mehr möglich sind.

### Qualitätsmanagementsysteme für die Softwareentwicklung

Für die Softwareentwicklung gibt es spezialisierte Systeme zum Qualitätsmanagement (CMMI-Dev, SPICE), die im Wesentlichen den Reifegrad einer Organisation oder einzelner Prozesse adressieren. Der Grundgedanke hierbei ist, dass Verbesserungen nur schrittweise eingeführt werden sollten.

Diese Verfahren bieten folgende Vorteile:

- In erster Linie überzeugt der ganzheitliche Ansatz. Nicht nur die Softwareentwicklung als Kern der Tätigkeit eines Unternehmens, sondern alle Bereiche werden dem Qualitätsmanagement unterworfen.
- Es bestehen Kontroll- und Steuerungsmöglichkeiten für das Management.
- Die Einführung des Qualitätsmanagements erfolgt nicht komplett in einem Zug sondern schrittweise, was den Betriebskörper nicht überfordert und jahrelang in Bewegung hält.

Dem gegenüber sind folgende Nachteile festzuhalten:

- **Zuviel Struktur:** Die Definition von Formalismen und Vorgehensweisen für das gesamte Unternehmen schert alles über einen Kamm. Das kann für kleinere Projekte und Vorhaben zu einem signifikanten Hindernis werden.

- **Zuviel Papier/Bürokratismus:** Die Verfahren gehen auf höheren Reifestufen mit einem erheblichen Formalismus einher. Erfahrungsgemäß korreliert die Menge produzierter Schrankware nicht mit der Qualität der Prozesse und Produkte.
- **Geringe Akzeptanz auf technischer Ebene:** Von den Entwicklungsteams werden die einzuhaltenden Formalien und die zu produzierenden Dokumente als wenig nützlich und ohne Effekt auf die Qualität ihrer Arbeit empfunden.
- **Fokus auf formaler Korrektheit:** Durch Audits wird nur die formale Einhaltung der Normen etc. geprüft. Ob Qualitätsmanagement tatsächlich gelebt wird oder nur pro Forma (zwecks Zertifikat), kann der Auditor nicht beurteilen.
- **Prozessorientierung:** Dies ist insofern ein Vorteil, da alle Tätigkeiten des Unternehmens in das Qualitätsmanagement einbezogen werden. Ein entscheidender Nachteil ist allerdings, dass die Qualität des Produkts wenig Berücksichtigung findet. Das ist insbesondere bei der Softwareentwicklung fatal, da hier die innere Qualität, das heißt die Verständlichkeit, Wartungs-, Erweiterungs- und Änderungsfreundlichkeit der Software in vielen Unternehmen und Projekten vernachlässigt wird.
- **Geringe Effizienz:** aus den genannten Problemen resultiert letztendlich eine geringe Effizienz der Systeme, die Verfahren entsprechen nicht den Grundsätzen der schlanken Produktion

### Agile Methodik

Die agile Softwareentwicklung stellt den Kunden und diejenigen Menschen in den Vordergrund, die an der Erstellung des Softwareprodukts beteiligt sind. Sie realisiert damit die Prinzipien des TQM



Im Lebenszyklus von Software fallen etwa 80 Prozent der Kosten für Änderungen im Anschluss an die Primärentwicklung an.

bezüglich Kundenorientierung und Einbeziehung aller Mitarbeiter. Insbesondere steht das Entwicklungsteam im Fokus, hier wird der Hebel zur Sicherung der Produktqualität angesetzt. Eine weitere Grundlage der agilen Philosophie ist schlanke Produktion, insbesondere wurden Aspekte des von Toyota entwickelten Toyota Way beziehungsweise des Toyota Production System für das Software Engineering adaptiert. Das den Toyota Way prägende Grundprinzip ist Kaizen. Kaizen heißt im wörtlichen Sinn „Veränderung zum Besseren“, bedeutet aber in der japanischen Praxis vor allem die Philosophie der ewigen (sichtbaren) Veränderung und bezieht alles am Produktionsprozess beteiligte mit ein: Mensch, Maschine, Umwelt, Methode, Material, Management. Kaizen erfordert Mitarbeiter, welche über die Qualifikation und die Motivation verfügen, den Prozess zu verbessern. Auftretende Probleme müssen sofort gelöst werden, Verbesserungsmaßnahmen müssen direkt von den Mitarbeitern umgesetzt werden können, was nur mit Hilfe von flachen Hierarchien gelingt.

Langjährige Erfahrung hat gelehrt, dass Softwareentwicklungsprojekte einer immensen Dynamik unterliegen. Es gibt häufig neue Bestimmungen, Change Requests oder einfach nur Änderungen der Prioritäten. Deshalb muss die Softwareentwicklung schnell auf Änderungen reagieren können. Des Weiteren ist die Qualität der Software sehr stark vom Know How, von der Motivation und Kreativität der Entwicklungsteams abhängig.

Um beides – Flexibilität und optimale Teamleistung – zu gewährleisten und das Prinzip Kaizen zu integrieren, wurden fünf Werte und 14 Prinzipien der agilen Softwareentwicklung aufgestellt, die sozusagen die Axiome darstellen, auf denen alle Methoden basieren. Die

## o Die Techniken des Extreme Programming

### Automatisiertes Testen:

Bei der Programmierung jeder Funktion wird gleichzeitig Testsoftware geschrieben, die immer wieder automatisiert ausgeführt wird und sicherstellt, dass das Programm korrekt arbeitet. Diese Erfindung ist wohl der wichtigste und erfolgreichste Beitrag zum modernen Software Engineering.

### Pair Programming:

Zwei Teammitglieder entwickeln zusammen, um die Anzahl der Fehler zu reduzieren und die Wissenszirkulation im Team zu gewährleisten. Dies ist insbesondere bei schwierigen Anforderungen und Programmfundamenten wichtig.

### Kollektives Eigentum:

Die Software gehört wie auch alle Anforderungen dem ganzen Team. Jeder ist (mehr oder weniger) für alles verantwortlich. Dies ermöglicht die schnelle Kompensation von Ausfällen, etwa durch Krankheit oder Urlaub.

### Permanente Integration:

Jederzeit ist ein lauffähiges Gesamtsystem vorhanden, in dem neue Programmteile getestet und angepasst werden können.

### Kundeneinbeziehung:

Schnelles Feedback und klare, präzise Absprachen verhindern Missverständnisse.

### Refactoring:

Die Struktur oder innere Qualität der Software wird ständig verbessert, um die Nachhaltigkeit und Sicherung des ROI zu gewährleisten.

### Keine Überstunden:

Überstunden reduzieren die Leistungsfähigkeit deutlich und erhöhen die Fehlerzahl dramatisch.

fünf Werte sind: Kommunikation, Einfachheit, Feedback, Mut und Respekt. Um der Einfachheit und dem Mut Genüge zu tun, wird hier auf die Auflistung der Prinzipien verzichtet.

Zu den wichtigsten agilen Methoden gehören Extreme Programming (XP) als Kerntechnik des Programmierens und Scrum für Steuerung, Planung, Fortschrittskontrolle und Prozessoptimierung. Aufgrund der mehrjährigen positiven Erfahrungen sind beide dem Stand der Technik im Software Engineering zuzuordnen.

## Extreme Programming

Der Name ist bewusst provokativ gewählt. Im Gegensatz zu RUP oder V-Modell wird im Sinne schlanker Produktion der Schritt im Gesamtprozess, in dem die Software programmiert wird, fokussiert und aufgewertet: Hier entscheiden sich Qualität und Effizienz im Software Engineering.

Der wesentliche Kerngedanke von XP ist die Sicherstellung der inneren und äußeren Produktqualität. Die wichtigsten Techniken sind im Kasten oben aufgeführt.

## Scrum

Ziel ist die ständige Verbesserung des Softwareentwicklungsprozesses, um höchste Qualität bei niedrigstem Aufwand zu erreichen. Der Weg ist die Weiterentwicklung aller an dem Prozess Beteiligten, der Entwicklungsprozesse,

## Die agile Softwareentwicklung

stellt den Kunden und die Menschen in den Vordergrund, die an der Erstellung des Softwareprodukts beteiligt sind.



[www.andrena.de](http://www.andrena.de)

der Arbeitsmittel und Methoden unter Beibehaltung der Grundannahmen (analog zum Toyota Production System).

Um den vielfältigen Spezifika und der Dynamik im Software Engineering Rechnung zu tragen, ist Scrum daher kein vollständiges System von Regeln und Anweisungen, sondern ein bewusst einfach gehaltener Rahmen, der an konkrete Bedingungen anzupassen oder zu ergänzen ist.

Es gibt die folgenden drei Rollen:

- Der Product Owner entspricht dem Kunden beziehungsweise Auftraggeber. Er legt das Ziel und die Prioritäten fest.
- Das Team arbeitet die Anforderungen gemäß der Priorisierung ab.
- Der Scrum Master überwacht den Prozess.

Der Entwicklungsprozess wird in kleine, geschützte Iterationen zerlegt, Sprints genannt. Geschützt meint hier, dass in diesem Zeitraum keine Änderungen von Aufgaben und Vorgehensweisen vorgenommen werden dürfen. Diese Sprints sollen, je nach Möglichkeit, zwischen zwei und vier Wochen lang sein. Diese kurzen Iterationen ermöglichen die schnelle Anpassung an neue Anforderungen. Gleichzeitig steht nach jeder Iteration eine lauffähige Software zur Verfügung, die getestet werden kann.

Ein Sprint läuft wie folgt ab:

- Der Product Owner legt die Prioritäten der Anforderungen fest.
- Das Team legt anhand der Prioritäten die zu bearbeitenden Anforderungen im Sprint fest und schätzt diese.
- Das Team arbeitet die Anforderungen der Reihe nach ab.

Der Produktionsfortschritt wird im Burndown Chart ersichtlich. Täglich

wird zu Arbeitsende der Restaufwand für Anforderungen geschätzt.

Am Ende des Sprints gibt es einen Review, bei dem der Kunde überprüfen kann, ob die Anforderungen richtig umgesetzt wurden.

Zuletzt gibt es noch eine Retrospektive, in welcher der Prozess an sich untersucht wird. Die Fragen „Was war gut“ und „Was kann noch verbessert werden“ sind hierbei zentral. Dies ist der entscheidende Punkt zur Prozessoptimierung. Verbesserungsmöglichkeiten werden im darauffolgenden Sprint umgesetzt.

### Agiles Qualitätsmanagement

Da Extreme Programming nur die Kernmethodik der Programmierung adressiert und Scrum eine bewusst einfache Planungs- und Steuerungsmethode ist, sind im Sinne von TQM ergänzende Mechanismen für ein ganzheitliches Qualitätsmanagement erforderlich.

In jüngster Zeit gab es verschiedene Konzepte, um Qualitätsmanagementsysteme zu schaffen, die mit der agilen Philosophie verträglich sind (etwa AgileQ von Elmar Borgmeier).

Das von andrena objects realisierte Qualitätsmanagementsystem (QMS) basiert auf folgenden Komponenten:

- EN ISO 9001:2000 stellt den formalen Rahmen, der mittels nachfolgender Verfahren mit Leben gefüllt wird.
- Scrum dient der Planung, Fortschrittskontrolle und der kontinuierlichen Verbesserung.
- Extreme Programming zur Sicherung der Produktqualität.
- ISIS, ein von andrena konzipiertes Navigationssystem für Prozess- und Produktqualität.

EN ISO 9001:2000 ist prozessorientiert und durch die Beschränkung auf die

Metrik Kundenzufriedenheit als Steuergröße für die Prozessverbesserung für alle Dienstleistungsbereiche geeignet. Die Norm erfasst alle für die Produktqualität wesentlichen Bereiche der Unternehmenstätigkeit. Als Konsequenz der Allgemeingültigkeit ist die Norm sehr schlank gehalten und stellt für agiles Software Engineering einen idealen, ergänzenden Systemrahmen dar.

ISIS ermöglicht eine kennzahlenbasierte Steuerung und Optimierung sowie Transparenz. Neben der Kundenzufriedenheit werden weitere, die Qualität des Prozesses und der Software kennzeichnende Metriken erfasst und bewertet. Monatlich werden die Werte für das Management und den Auftraggeber publiziert. Teamintern dienen die Daten zur Steuerung und Optimierung des Entwicklungsprozesses. Eine detaillierte Beschreibung von ISIS findet sich in Sanierung von Softwaresystemen, it-management September 2008.

### Fazit

Entscheidend für den Erfolg des QMS ist:

- Das Vermeiden von Schrankware.
- Das Verhindern der Papierlawine.
- Die hierdurch erst mögliche Erkenntnis aller Beteiligten, dass das QMS die Qualität der Arbeit verbessert und nicht die Arbeit selbst behindert.

Angestrebt werden muss ein Optimum zwischen der Detailliertheit von Verfahrensanweisungen beziehungsweise Vorgaben und der für Effizienz und Qualität erforderlichen Flexibilität der Einzelprojekte. Die für die Softwareentwicklung bei andrena maßgebenden Verfahrensanweisungen umfassen acht Seiten Papier, die gelebt werden. Durch eine weitgehend automatisierte Erfassung und Verwaltung von Aufzeichnungen und Qualitätsdaten behindern wir die Arbeit nicht mit der Produktion als unnützlich empfundener Dokumente. Das alles wird von allen Beteiligten als positiv empfunden, nach einjährigem Einsatz sind erste Verbesserungen der Prozesse messbar.

DR. EBERHARD KUHN,  
DR. ANDREAS ARNOLD



Kaizen heißt wörtlich „Veränderung zum Besseren“, bedeutet aber in der japanischen Praxis vor allem die Philosophie der ewigen Veränderung.

## ► Leistungen von der andrena objects ag

- Entwicklungsunterstützung
- Projektmanagement /QS Begleitung
- Beratung
- Sanierung /Migration
- Lösungen /agiles Geschäftsmodell

## ► ObjektForen 2008

18.11.2008

ObjektForum Frankfurt

**„SOA: Befgriffe, Muster, Fallbeispiele“**

Referent: Prof. Dr.-Ing. Arne Koschel (FH Hannover)

Museum für Kommunikation, Frankfurt 18:30

18.11.2008

ObjektForum Berlin

**„Build-Prozesse für Geschäftsanwendungen mit Apache Maven“**

Referente: Alphonse Bendt (akquinet AG),

Dr. Torsten Fink (akquinet AG)

FU Berlin, Hörsaal der Informatik, Takustrasse 9

24.11.2008

ObjektForum Stuttgart

**„Domain-Driven Design –**

**Was Baggerfahren von guter Softwareentwicklung unterscheidet“**

Referente: Marco Klemm (Berater),

Ludger Solbach (Deutscher Sparkassen Verlag GmbH)

Alte Scheuer Degerloch, 18:30

01.12.2008

ObjektForum Karlsruhe

**„Server Side Eclipse“**

Referent: Dr. Frank Gerhardt (Gerhardt Informatics Kft.)

Technologiepark Karlsruhe, 18:30

15.12.2008

ObjektForum München

**„Von der modellbasierten Anwendungsentwicklung zur generierten Projektdokumentation“**

Referent: Andreas Ditze (MID GmbH)

Sofitel Hotel München, 18:30

Die Informationen über alle unsere Veranstaltungen, ObjektForum, XP Days, Entwicklertag, als auch Anmeldung und aktuelle Stellenangebote finden Sie unter [www.andrena.de](http://www.andrena.de)