

Qualität ist das beste Rezept

# Mit agilem Qualitätsmanagement zum Projekterfolg

Elmar Borgmeier

Softwarequalität zu gestalten, bleibt eine Herausforderung für das Projektmanagement. In den letzten Jahren prägten zwei gegenläufige Entwicklungen das Bild: Einerseits gewann das klassische Qualitätsmanagement an Bedeutung, andererseits etablierten sich immer mehr Verfahren der agilen Softwareentwicklung, die geradezu entgegengesetzte Schwerpunkte setzen. Gerade im Umfeld von Java und Eclipse entstanden zahlreiche qualitätsunterstützende Werkzeuge für die Softwareentwicklung. Agiles Qualitätsmanagement bietet die Chance, diese Vielfalt im Rahmen eines methodischen Qualitätsmanagements systematischer zu nutzen, ohne die Flexibilität des agilen Projektmanagements zu verlieren.

► Es müssen nicht immer die spektakulär gescheiterten Großprojekte sein, die auf das Thema Softwarequalität aufmerksam machen. Auch in der täglichen Praxis sind die enormen Effizienzpotenziale durch Qualitätssteigerungen für Entwickler, Projektmanager und Führungskräfte immer wieder augenfällig. Doch wie kann eine Organisation diese Potenziale für sich am besten nutzen? Wie entwickelt sie aus dem projektbezogenen Einsatz der aktuellen Werkzeuge wie Maven, Findbugs etc. eine nachhaltige, in der Organisation verankerte Qualitätsstrategie? Auf diese Fragen sind in den letzten Jahren zwei Antworten gegeben worden, wie sie unterschiedlicher nicht sein könnten: Empfohlen werden einerseits die Einführung eines ingenieurmäßigen Qualitätsmanagements und andererseits die Umstellung auf agile Prozesse. Beide Ansätze werden im Folgenden kurz vorgestellt und durch ihre Kernmerkmale charakterisiert.

## Klassisches Qualitätsmanagement

Qualitätsmanagement (QM) wurde eingeführt, um Qualität systematisch zu steuern. Der systematische Ansatz ist schon deshalb notwendig, weil unter Qualität unterschiedliche Dinge verstanden werden, die alle ihre Berechtigung haben: Entwickler achten auf Klarheit des Designs und wartbaren Code, die Verantwortlichen für den Betrieb achten auf Stabilität und Änderbarkeit, Anwender auf vollständige Umsetzung ihrer Anforderungen und Software-Ergonomie, Projektleiter auf die Einhaltung des Projektprozesses und Auftraggeber auf Budget- und Termintreue.

Um sich innerhalb dieser vielfältigen Ziele auszurichten, geht das klassische Qualitätsmanagement von den Gesamtzielen des Unternehmens aus, und leitet daraus die Zielsetzungen für Softwareprojekte ab. Dieser Ansatz wird insbesondere beim *Total Quality Management* (TQM) verfolgt. Eine Konkretisierung von TQM ist das Modell der European Foundation for Quality Management (EFQM), das in einem Gesamtbild Prozesse und Ergebnisse im Sinne der verschiedenen Zielgruppen Kunden, Mitarbeiter und Gesellschaft betrachtet. Besonders der ganzheitliche Ansatz überzeugt daran! Die Deutsche Gesellschaft



für Projektmanagement (GPM) verwendet ein daraus abgeleitetes Schema auch zur Bewertung von Projektqualität im Rahmen ihres Modells *Project Excellence*.

Qualitätssteigerungen einer Organisation können nur schrittweise umgesetzt werden. Daher haben sich im Softwarebereich neben den bekannten Konzepten der ISO 9000ff. vor allem so genannte Reifegradmodelle etabliert, die eine stufenweise Verbesserung vorsehen. Beim CMMI-Modell der Carnegie-Mellon University wird dabei der Reifegrad einer Organisation insgesamt betrachtet, das europäische Gegenstück SPICE bewertet den Reifegrad einzelner Prozesse.

Die Vorteile der genannten Verfahren zum Qualitätsmanagement liegen vor allem darin, dass sie den Führungskräften einer Organisation einen Handlungsrahmen bieten, innerhalb dessen Qualität systematisch weiterentwickelt und der erreichte Stand bewertet werden kann. Qualität kann gesteuert und geprüft werden. Unterhalb der Managementebene, bei den Entwicklern und Projektmanagern, finden die Verfahren jedoch weit weniger Akzeptanz. Die Hauptkritikpunkte sind:

- ▼ Klassische Verfahren zum Qualitätsmanagement sind stark prozessorientiert. Auch sehr gut kontrollierte Prozesse können bei Entwicklungstätigkeiten allein aber noch nicht die Produktqualität sicherstellen.
- ▼ Prozess-Standardisierung orientiert sich häufig an den größten und anspruchsvollsten Szenarien und ist damit dann überdimensioniert für kleinere und mittlere Softwareprojekte.
- ▼ Um eine externe Auditierbarkeit zu gewährleisten, wird häufig die Erstellung von Dokumenten vorgesehen, die das eigentliche Projektteam für sich als wenig hilfreich und als bürokratische Zusatzarbeit betrachtet.

Mangelnde Akzeptanz führt dann regelmäßig zu einer Pro-forma-Umsetzung des Qualitätsmanagements, mit der die eigentlichen Ziele nicht mehr erreicht werden können. Letztlich wirken die Maßnahmen sogar kontraproduktiv und beschädigen das Image des Qualitätsmanagements bei den technisch orientierten Beteiligten erheblich. Gerade die Entwicklung negativer Einstellungen zur Organisation von Qualität ist besorgniserregend und, wenn sie sich einmal eingeschlichen hat, schwierig zu korrigieren.



## Agile Verfahren

Andererseits entstanden agile Verfahren wie XP, Scrum oder die Crystal-Clear-Methodiken, die sich als Gegenentwurf zum klassischen Qualitätsmanagement verstehen lassen: Der Schwerpunkt liegt auf Kommunikation statt auf Prozessen, auf der Software selbst anstatt auf umfassender Dokumentation, auf aktivem Änderungsmanagement statt auf hoher Planbarkeit.

Da das Wort „agil“ inzwischen in durchaus verschiedenen Varianten verwendet wird, ist es hilfreich, zwischen zwei etablierten Begriffen zu unterscheiden:

- ▼ **Agile Softwareentwicklung:** bezieht sich auf die Gestaltung des eigentlichen Entwicklungsprozesses im Sinne agiler Prinzipien. Es gibt unterschiedliche Ausprägungen, die bekannteste ist das *eXtreme Programming (XP)*.
- ▼ **Agiles Projektmanagement:** bezieht sich auf die Gestaltung des Gesamtprojektablaufs, also auf die Anpassung von Prozessmustern an die konkreten Erfordernisse für das IT-Projekt. Agiles Projektmanagement kann auch in IT-Projekten eingesetzt werden, in denen nicht Softwareentwicklung durch ein Entwicklungsteam im Mittelpunkt steht, also etwa bei der Einführung von Standardsoftware.

Leider finden sich in der internationalen Literatur zur agilen Softwareentwicklung wenig spezifische Aussagen zum Umgang mit Qualität. Aus den gewählten Formulierungen wird implizit klar, dass die Autoren eher darauf setzen, dass Qualität von selber entsteht, wenn erfahrene „Softwerker“ gut zusammen arbeiten. (Eine Ausnahme sind die Autorinnen Christiane Gernert, die in ihrem Buch [Ger03] Qualitätsmanagement ernsthaft integriert, sowie Jutta Eckstein [Eck04], die zumindest Vorschläge zur Zusammenarbeit von Qualitätskontrolle und -sicherung mit den Projektteams macht.)

Auch wenn in die agilen Verfahren bereits von ihren Ansätzen her eine Qualitätsorientierung integriert ist, so bedeutet der Verzicht auf ein explizites und bewusstes Qualitätsmanagement doch eine wesentliche Einschränkung:

- ▼ Ohne explizite Qualitätssteuerung setzt sich der subjektive Qualitätsbegriff der Beteiligten durch – und ein typischer Softwareentwickler neigt dazu, z. B. Belange des Betriebs dabei niedriger zu gewichten.
- ▼ Bei Qualität gibt es nicht nur die Entscheidung für mehr oder weniger, sondern auch für die eine oder andere Art von Qualität: beispielsweise wurde eine Zeit lang gern Javascript eingesetzt, um Websites interaktiver und benutzerfreundlicher zu gestalten. Wegen fortlaufender Änderungen an Javascript in neuen Browserversionen musste die Software dann aber immer wieder angepasst werden. Deswegen galt dann in vielen Unternehmen der Verzicht auf Javascript als Qualitätsmerkmal. Aktuell hat sich der Trend wieder gedreht, mit dem AJAX-Konzept steht wieder Interaktivität im Vordergrund.

- ▼ Eine nur implizite Qualitätsorientierung ist nur innerhalb eines Projektes wirksam und trägt wenig zum Fortschritt der Gesamtorganisation bei.

Nicht zu unrecht steht daher gerade das Management den agilen Verfahren häufig skeptisch gegenüber, ihre Leistung für die Organisationsentwicklung ist in der bisherigen Form zu wenig greifbar, auch wenn die konkreten Projekterfolge sehr wohl für die agilen Methoden sprechen.

## Agiles Qualitätsmanagement

Daher wurden in jüngster Zeit Ansätze zum agilen Qualitätsmanagement (*Agile QM*) entwickelt, also einem expliziten Verfahren zur Steuerung von Qualitätsmaßnahmen, das sich mit den Prinzipien der agilen Verfahren verträgt. Die Leitlinien dafür sind:

- ▼ Agiles QM fügt sich flexibel und angemessen in ein konkretes Projektvorgehen ein – es ist dem Projektleiter und -team Hilfsmittel, kein unverstandener Overhead.
- ▼ Agiles QM ist ergebnisorientiert und daher technologiezentriert. Es umfasst konkrete und technische Maßnahmen zur Sicherung der Softwarequalitätsmerkmale.
- ▼ Agiles QM zeigt Qualität als besondere Leistung und motiviert die Kompetenzträger zur Übernahme von Qualitätsaufgaben.

Agiles QM setzt sich damit bewusst von den rein prozessorientierten Ansätzen des klassischen Qualitätsmanagements ab, um mit einem personen- und technologiezentrierten Ansatz diejenigen Mitarbeiter zu erreichen, die in der Vergangenheit von überbordenden Formalismen abgeschreckt wurden.

Im Folgenden wird ein spezifisches Denkmodell für Agiles QM definiert, das die Leitsätze aufgreift und konkret umsetzbar macht. Das Modell nennt sich *AspectQ* in Anlehnung an AspectJ, weil es wesentliche Gedanken aus der aspektorientierten Programmierung auf die Konzeption eines Qualitätsmodells überträgt. AspectQ basiert darauf, Qualitätsthemen als mehrere zusätzliche Aspekte des Projektvorgehens zu betrachten. Diese zusätzlichen Aspekte werden separat vom Projektprozess festgelegt und dann an zu definierenden Stellen in den Projektprozess eingefügt (s. Abb. 1).

Die Klassifikation der Qualitätsthemen in mehrere Aspekte erfolgt so, dass innerhalb eines Aspektes ein möglichst starker Zusammenhang der Maßnahmen besteht. Es ist ausdrücklich *nicht* so, dass sich die Aspekte einem Projektabschnitt zuordnen lassen. Sie werden durchgängig in den Ablauf „eingewo-

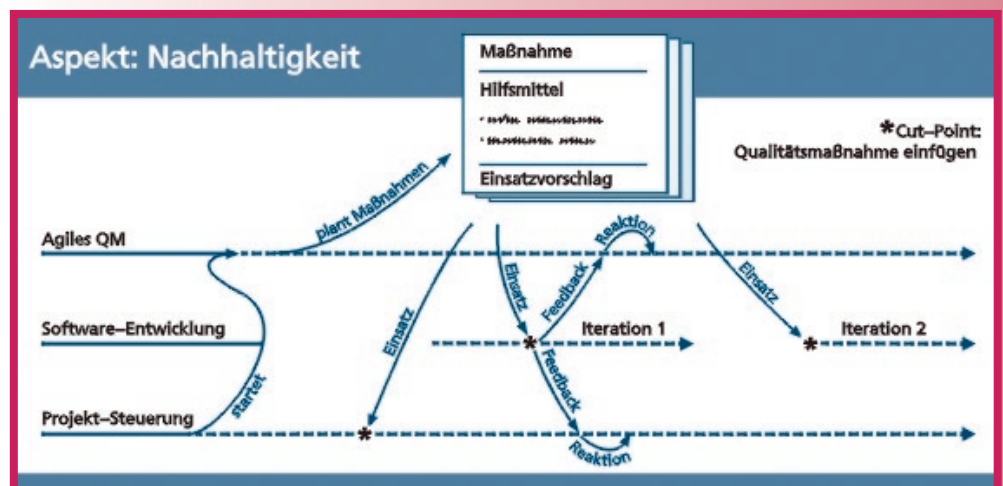


Abb. 1: Softwareentwicklung, Projektsteuerung und agiles Qualitätsmanagement unterstützen sich gegenseitig



### Aspekte von Qualität

Qualität besitzt verschiedene Dimensionen, die jeweils mit eigenständigen Maßnahmenbündeln adressiert werden. Deshalb unterscheidet AspectQ die folgenden Aspekte:

- ▼ *Kunden-bezogene Qualität*: Stellt die Sicht des Auftraggebers dar – Qualitätskriterien sind hier Faktoren wie Kosten, Termine oder Investitionssicherheit.
- ▼ *Anwender-bezogene Qualität*: Die Qualitätsthemen, zu denen primär ein Anwender beitragen kann, also im Wesentlichen funktionale Eigenschaften von Software, deren Spezifikation und Test, sowie der Umgang mit Klärungen und Änderungen während des Projekts. Dazu gehören auch der Umgang mit nicht ausgesprochenen Erwartungen und die Benutzerfreundlichkeit der Software.
- ▼ *Technologie-bezogene Qualität*: Qualitätsmerkmale des ausführbaren Programms, die zu stark mit der Technik verknüpft sind, als dass sie von den Anwendern allein vorgegeben werden könnten. Vor allem Performance, Stabilität und Sicherheit sind hier zu adressieren.
- ▼ *Innere Qualität*: Qualitätsmerkmale, die zunächst nur im Quellcode selbst erkennbar sind und deshalb auch dort geprüft werden. Dazu gehören Klarheit des Designs, saubere Code-Strukturen, Verständlichkeit und Wartbarkeit.
- ▼ *Nachhaltigkeit*: Der Nutzungszeitraum von Software beginnt erst mit dem Abschluss des Entwicklungsprojektes. Deshalb ist es wichtig, von Anfang an den Betrieb der Anwendung und ihre Betreuung im Blick zu haben. Keine leichte Aufgabe, da Entwickler und Betreiber Software ganz unterschiedlich sehen. Maßnahmen zu diesem Aspekt adressieren vor allem Themen wie Konfigurierbarkeit, Überwachung, einheitliches Änderungs- und Release-Management, Offenheit der Schnittstellen und Fehlertoleranz.

ben“ und bilden damit das *Quality Pattern* des Projektes. Die in AspectQ verwendete Einteilung in die fünf Aspekte Kunden-, Anwender- und Technologie-bezogene Qualität, innere Qualität und Nachhaltigkeit ist im Textkasten zu diesem Artikel erläutert. Wichtig ist, dass durch die explizite Unterscheidung der Aspekte die Komplexität des Qualitätsbegriffs sichtbar gemacht wird, und die Diskussion um die jeweilige Priorisierung von Qualitätszielen gefördert wird. Basierend auf diesem Grundansatz definiert AspectQ zwei Ebenen von QM-Verfahren: einmal das Verfahren innerhalb eines Projektes, und dann das Verfahren zum projektübergreifenden „Qualitätslernen“ in einer Organisation.

Innerhalb eines Projektes findet die Qualitätsplanung in speziellen Workshops statt, in denen die Maßnahmen und ihre Einsatzzeitpunkte für alle Aspekte festgelegt werden. Gerade bei Java-Enterprise-Projekten mit Eclipse als Entwicklungsumgebung kennt jedes Projektteam eine Vielzahl möglicher Aktivitäten zur Qualitätssteigerung: Werkzeuge wie JUnit oder Plug-Ins wie Findbugs stehen zur Verfügung, der Nutzen von Prototyping und Code Reviews ist belegt. Ziel ist daher vor allem, die effektivsten Maßnahmen für das konkrete Projekt zu identifizieren und ein angemessenes Set an Maßnahmen

zu vereinbaren. Das bewusste Durchdenken der unterschiedlichen Qualitätsaspekte im Entwicklerteam fördert aber auch neue Ansätze zu Tage, z. B. in Abstimmung mit dem Rechenzentrum die Monitoring-Fähigkeiten der Java-Enterprise-Plattform besser zu nutzen.

Alle Maßnahmen werden in Form von Maßnahmen-Karten beschrieben und im Intranet veröffentlicht. Dieser Fundus an vorgefertigten Maßnahmen-Karten bildet dann nicht nur einen Grundstock für die Workshops anderer Projekte, sondern ist auch die Basis für „Qualitätslernen“ der Organisation. Qualitätsmanagement wird nicht aus einem formalen Gesamtkonzept heraus in die Projekte getragen, sondern aus den Erfahrungen der Projekte heraus in die Organisation hinein entwickelt. Die Maßnahmenbeschreibungen im Intranet umfassen Angaben zu den Randbedingungen, unter denen der Einsatz einer bestimmten Maßnahme sinnvoll erscheint, zu den Projekten, die bisher diese Maßnahme eingesetzt haben, und zu den Erfahrungen dieser Projekte mit den Maßnahmen. Daraus lassen sich direkt Bewertungen ableiten und in einem evolutionären Prozess die erfolgreichsten Maßnahmen identifizieren. Diese können dann standardisiert und durch geeignete Hilfsmittel noch besser unterstützt werden – ihre Akzeptanz ist dann bereits gesichert.

### Fazit

Klassisches Qualitätsmanagement hat die explizite Steuerung von Qualität als eigenständige Aufgabe etabliert. Agiles Projektmanagement hat die Bedeutung des Faktors Mensch gegenüber formalen Prozessen wieder in den Vordergrund gestellt. Agiles Qualitätsmanagement strebt an, beides zu verbinden, indem es in die Projektteams selbst das Qualitätsmanagement als explizite Aufgabe hineinträgt und es über eine weniger formale „Verpackung“ leichter zugänglich macht. Erfolgreich ist es dann, wenn es die Akzeptanz der letztlich im Projekt handelnden Personen findet, und gleichzeitig Ergebnisse liefert, die über das Projekt hinaus auf der Organisationsebene sichtbar sind und dem Management einen Ansatz bieten, das Organisationslernen zum Qualitätsmanagement systematisch voranzutreiben.

### Literatur

- [Ger03] Ch. Gernert, *Agiles Projektmanagement*, Hanser, 2003
- [Eck04] J. Eckstein, *Agile Softwareentwicklung im Großen. Ein Eintauchen in die Untiefen erfolgreicher Projekte*, dpunkt, 2004



**Elmar Borgmeier** ist Technologie-Vorstand der syngenio AG in Stuttgart und seit neun Jahren als Berater und Projektleiter tätig. Seine Schwerpunkte liegen in den Bereichen SOA und Agiles Qualitätsmanagement. E-Mail: [elmar.borgmeier@syngenio.de](mailto:elmar.borgmeier@syngenio.de).