



CHANCE FÜR DIE QUALITÄTSSICHERUNG IN DER SOFTWAREENTWICKLUNG

# Agiles Vorgehen macht Programme besser

Christian Graf, Erlangen

Der Einsatz agiler Vorgehensmodelle hat sich in vielen softwareproduzierenden Unternehmen durchgesetzt. In regulierten Branchen wie der Medizintechnik bewährt sich schon heute Lean Software Development mit XP, Lean und Scrum. Agile Praktiken orientieren sich an den Grundprinzipien des Total Quality Managements und bieten die Chance zur nachhaltigen Verbesserung der Qualitätssicherungsprozesse.

Die Einführung neuer Methoden der Softwareentwicklung wie Extreme Programming (XP), Lean Thinking oder Scrum führt zu tiefgreifenden Umstellungen in allen Prozessen der betroffenen Bereiche und der bereichsübergreifenden Zusammenarbeit. Ist sie erfolgreich, winken:

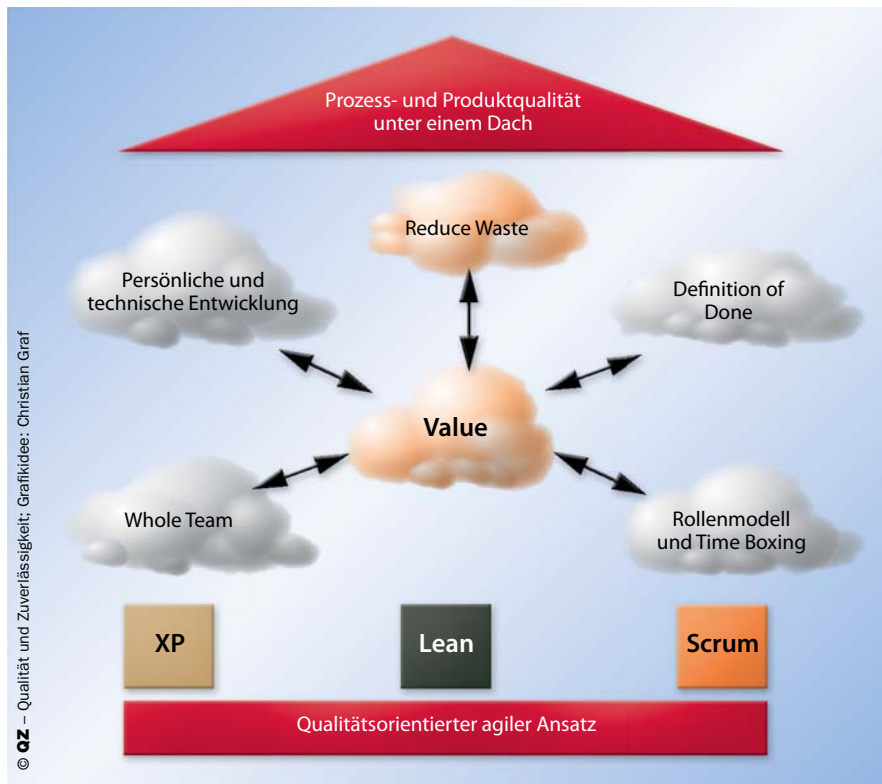
- stabilere Produkte,
- kürzere Entwicklungszeiten,
- stabilere Anforderungen,
- höhere Kundenzufriedenheit und
- effiziente Qualitätssicherung.

Die Erfahrungen der letzten Jahre zeigen, dass diese Ziele sogar im sicherheitskritischen Umfeld erreicht werden können.

Es ist aber nicht gesagt, dass der Weg, den man in der eigenen Organisation wählt, auch zu diesem Erfolg führt. Denn ein einzelnes Verfahren buchstabengetreu zu übernehmen, reicht nicht aus. Um zu verstehen, wie sich die unterschiedlichen Ansätze gegenseitig ergänzen, muss man sie genauer betrachten (siehe Infoboxen). Folgendes Vorgehen hat sich bewährt:

- Ziele definieren und einen oder mehrere Pilotprojekte planen,
- Pilotprojekte durchführen,
- Erfahrungen aus den Pilotprojekten sammeln und die Erreichung bzw. Erreichbarkeit der Ziele bei einem organisationsweiten Rollout prüfen,
- eigene Vorgehensweisen anpassen und in weiteren Teams, Projekten und Abteilungen zur Anwendung bringen.

Diese Ziele müssen messbar sein: Ein Benchmark-Wert zum Vorher-Nachher-



**Bild 1.** Extreme Programming (XP), Lean Thinking und Scrum bilden die Säulen einer stabilen Softwareentwicklung. Das Wertschöpfungsprinzip ist der Motor agiler Prozesse. Es definiert das Verständnis von Prozess- und Produktqualität und damit direkt die Umsetzung agiler Prinzipien im konkreten Projekt.

Vergleich ist notwendig. Zum Lernprozess agiler Denkweisen gehört es, die Prioritäten von Anforderungen neu und in der Regel realistischer einzustufen. Um zu messen, ob Entwicklungsaufwände dadurch effizienter eingesetzt werden, fragt man, wie zielgerichtet das Team arbeitet,

und betrachtet die angefallenen Aufwände in Aktivitätengruppen wie etwa Design, Coding & Unit Testing, Systemintegration, Integrations- & Systemtest, Team-Entwicklung, Technisches Management, Werkzeug-Infrastruktur und Support nach außen. Diese Aktivitätengruppen

## ► EXTREME PROGRAMMING (XP)

### Prinzipien und Werte für gute Software

Extreme Programming (XP) ist ein Satz von Prinzipien, Werten und Good Practices, die es einem Team von Softwareentwicklern ermöglichen, seine bestmögliche Leistung zu entfalten. Typische XP-Merkmale sind:

- Alle erforderlichen Fähigkeiten und Kenntnisse müssen im Team vorhanden sein. Aber nicht jeder muss alles können.
- Lösungen sollen so einfach wie möglich gehalten und nachgewiesenermaßen stabil sein. Dazu gehört schnelles Feedback während der Entwick-

lung. Entsprechend werden Techniken wie das „Test First“-Prinzip und Continuous Integration eingesetzt.

- Beim Pair Programming lösen Team-Mitglieder mit unterschiedlichen Fähigkeiten eine Aufgabe im Paar, um sich gegenseitig zu ergänzen oder voneinander zu lernen.

Eine konsequente Anwendung von Extreme Programming erfordert eine hohe Disziplin der Team-Mitglieder und den Willen, stets das eigene Tun zu überdenken und die eigene Arbeitsweise stetig weiterzuentwickeln.

lassen sich den in Scrum üblichen Tasks zuordnen. Die verknüpften Aufwände sind damit auch in agilen Projekten erfassbar.

Zur Messung der Qualität der umgesetzten Anforderungen betrachtet man deren Reife gemessen an der Anzahl und den Ursachen der Fehler, die nach Freigabe (Done) durch das Team und den Product Owner von Dritten entdeckt werden.

### Auf das Zusammenspiel der Methoden kommt es an

Agile Methoden einzuführen bedeutet, neue Arbeits- und Denkweisen zu erlernen und die eigene Prozess- und Planungswelt neu zu strukturieren. Hierfür muss man eine ausreichende Eingewöhnungszeit einkalkulieren und verstehen, wie sich Scrum, XP und Lean Thinking gegenseitig ergänzen (Bild 1). Ein qualitätsorientierter agiler Ansatz verwendet das Rahmenwerk von Scrum zur Team- und Projektsteuerung sowie XP als Methodenkatalog und Ideengeber für eine effiziente Softwareentwicklung. Im Softwarebereich unterstützen XP-Methoden die Umsetzung der Lean-Prinzipien hervorragend.

Den Fortschritt bei der Methodenumsetzung kann der Scrum Master mit dem Team sichtbar machen, indem er in der Retrospektive nach jedem Sprint einschätzen lässt, zu welchem Anteil angestrebte oder vereinbarte Praktiken eingesetzt werden.

Welche XP-Methoden verwendet werden, was die Definition of Done (DoD) und die Definition of Ready (DoR) in Scrum beinhalten, muss sich am Wertschöpfungsprinzip orientieren. Zu entscheiden, was Wertschöpfung im konkreten Fall bedeutet, ist eine Aufgabe, an der alle Prozessbeteiligten (auch erfahrene Projektleiter und Qualitätsmanager) teilhaben müssen und bei der unkonventionelles, kreatives Denken gefragt ist. Auf keinen Fall darf man alle Storys über einen Kamm scheren. Ein Beispiel: In der Medizintechnik kann eine als kritisch eingestufte Software in kritische und unkritische Komponenten zerlegt werden. Die kritischen Komponenten müssen anders entworfen, dokumentiert und getestet werden als die unkritischen. Um das Endprodukt sicherer zu gestalten, hat ihre Entwicklung vielleicht Auswirkungen auf das Design der Hardware. Typische überflüssige Aktivitäten wären hier: »

**Literatur**

- **Mackert, O.; Hildenbrand, T.:** Von wasserfallartigen Softwareentwicklungsmodellen hin zu Lean Software Product Development. Rundbrief des Fachausschusses WI-MAW der Gesellschaft für Informatik 2011, S. 48-59
- **Koschek, H.:** Geschichten vom Scrum. D Punkt Verlag, 2010
- **Martin, R. C.:** Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008
- **Schoonderwoert, N. van:** Embedded Agile Project by the Numbers With Newbies. agile 2006, S. 351-366
- **Poppendieck, M. u. T.:** Leading Lean Software Development. Addison-Wesley Longman, 2010
- **Rasmussen, R.; Hughes, T.; Jenks, J. R.; Skach, J.:** Adopting Agile in an FDA Regulated Environment. Agile Conference, 2009
- **Sutherland, J.; Schwaber, K.:** The Scrum Guide (2011). <http://www.scrum.org/scrumguides>

**Autor**

**Christian Alexander Graf**, geb. 1970, ist für die Method Park Software AG als Senior Consultant tätig. Er ist stellvertretender Leiter der Fachgruppe Medizintechnik Erlangen-Nürnberg beim Arbeitskreis Software Qualität und Fortbildung (ASQF). Seit 2005 ist er Dozent an der Dualen Hochschule Baden-Württemberg in Mannheim. Hier hält er Vorlesungen zu Mathematik, Grundlagen der Informatik, Algorithmen und Datenstrukturen sowie zur Software-Qualitätssicherung.

**Kontakt**

**Christian Alexander Graf**  
**T 09131 97206-464**  
**christian.graf@methodpark.de**

**www.qz-online.de**

Diesen Beitrag finden Sie online unter der Dokumentennummer: **399125**

- die Sicherheitsklasse aller Komponenten vor der Sicherheitsklasse des Softwaresystems zu identifizieren und
- für unkritische Komponenten dieselben Aktivitäten wie für kritische durchzuführen.

Die notwendigen Aktivitäten werden rein über Akzeptanzkriterien Story-abhängig festgelegt. Das ist aber unübersichtlich und im Nachhinein schwer nachvollziehbar. Eine transparente Vorgehensweise ist, einen Satz von DoR- und DoD-Templates inklusive einer Checkliste mit den durchzuführenden Aktivitäten für unterschiedliche Situationen, wie eben sicherheitskritische und nichtkritische Storys, zu verwenden. Eine Rückverfolgbarkeit auf entsprechende regulatorische Prozessanforderungen kann dann durch Kopplung der typischen agilen Prozesswerkzeuge mit einem QM-Prozesswerkzeug erreicht werden. In diesem Werkzeug sind die entsprechenden Templates mit den zugehörigen Prozessaktivitäten verknüpft, ohne dass mit den Entwicklungsteams ein zusätzlicher Overhead entsteht.

**Fähigkeiten im Team sollten diversifiziert werden**

Die Zusammensetzung des Teams ist ein weiterer Erfolgsfaktor. Nach dem „Whole Team“-Prinzip in XP müssen im Team alle benötigten Fertigkeiten vertreten sein. Das

betrifft Fähigkeiten im Software-Engineering, Software-Test und Werkzeug-Kenntnisse genauso wie das Wissen um die Einsatzgebiete der Software. In der Medizintechnik müssen zum Beispiel Risikomanagement, Anforderungen an Gebrauchstauglichkeit, Werkzeugvalidierung und regulatorische Anforderungen an die Dokumentation im Team bekannt sein und gelebt werden. Auch überschätzen Softwareentwickler tatsächlich häufig die Sicherheitsklasse einer Komponente. Das führt zu unnötigen Mehraufwänden. Gleichzeitig werden Aspekte der Gebrauchstauglichkeit oft unterschätzt. Im Laufe der agilen Fortentwicklung ist es deswegen sinnvoll, Spezialisten hierfür entweder in die Teams zu integrieren oder Teammitglieder fortzubilden. Es gibt Firmen, die ihre Mitarbeiter dazu beispielsweise in Krankenhäusern hospitieren lassen.

**Organisationsweiter Erfolg mit QM**

Das Zusammenschweißen vieler Teams zu einer funktionsfähigen Organisation ist auch für agile Ansätze der schwerste Schritt. Meist werden Vertreter der Einzelteams in einem „Scrum of Scrums“ in übergeordnete Teams entsendet, die die Zusammenarbeit der Teams und Abteilungen koordinieren. Die Erfahrung zeigt, dass das nicht ausreicht. Neben den Pro-

**▶ LEAN THINKING****Schlankes Denken für wirtschaftliche Prozesse**

Lean Thinking basiert auf den Ideen des Toyota-Produktionssystems (TPS). Wichtig ist dabei das richtige Verständnis des Begriffs „Wertschöpfung“ im eigenen Kontext. Alles Handeln wird daran ausgerichtet und optimiert. Alle Tätigkeiten, die nicht der Wertschöpfung dienen, gelten als überflüssig (Waste).

Das klingt einleuchtend, ist aber in der Praxis alles andere als einfach zu realisieren. Typische Fehler sind falsche Vorstellungen und Konzepte von Wertschöpfung und wie diese gemessen werden kann oder die Fokussierung auf die reine Beseitigung von Waste, ohne die Optimierung der Wertschöpfungskette im Auge zu behalten. Die Folge ist eine strategisch falsche Ausrichtung von Teams

und Abteilungen, von Produkten und im schlimmsten Fall des gesamten Unternehmens. Um ein falsches Konzept zu vermeiden, werden im Lean-Kontext weitere Prinzipien wie Transparenz, Synchronisation, Perfektion, One Piece, Zero Defects, Pull-Prinzip, Timebox und eine ganze Flut von Kreativ-Techniken eingesetzt.





## ► SCRUM

### Wiederholung ist das Erfolgsgeheimnis

Scrum ist ein iterativer Ansatz zum Management von Projekten und ein flexibles Rahmenwerk zur Lösung komplexer Aufgaben. Den Kern dieses Rahmenwerks bilden:

- das aus Iterationen bestehende Prozessgerüst,
- das Rollenmodell mit Product Owner, Scrum Master und (Entwickler-)Team,
- Transparenz, Betrachtung des Ist-Zustands und Anpassung an den Soll-Zustand.

Hier kommt es auf das strikte Einhalten der vorgegebenen Zeitrahmen an (Timeboxing), sei es für eine Iteration der Produkterstellung (Sprint) oder für Besprechungen. Ressourcen und der Umfang dessen, was in einer Iteration umzusetzen ist, dürfen variieren – der Zeitrahmen bleibt fest.

Was es in einer Iteration erledigen kann, entscheidet allein das Team. Der Product Owner gibt vor, was umzusetzen ist und mit welcher Priorität. Letztere wird anhand des Geschäftswerts der umzusetzenden Dinge (Backlog Items) und des mit diesen verbundenen Risikos bemessen. Diese Backlog Items müssen klar und verständlich als Definition of Ready (DoR) formuliert werden. Und die Beteiligten müssen sich einig sein, wann sie als abgearbeitet gelten, also eine Definition of Done (DoD) erstellen. Was nicht als fertig gilt, darf auch nicht an einen Endkunden ausgeliefert werden. Damit gehören insbesondere Qualitäts-



und Prüfkriterien notwendigerweise in die DoD.

Die Scrum-Grundregeln lassen dem Anwender vollkommene Freiheit in der Definition des Geschäftswerts, den Definitionen von Ready und Done und der Auswahl weiterer Methoden. Das Vorgehensmodell macht weder Vorgaben zu Inhalten des Anforderungsmanagements oder der Entwicklungsmethodik noch zur Qualitätssicherung. Die Skalierung von Scrum auf mehrere Teams oder über ganze Bereiche einer Organisation hinweg wird ebenfalls nicht adressiert. Die diesbezüglichen Bedürfnisse einer Organisation müssen aktiv in die Scrum-Implementierung eingebracht werden. Dennoch ist Scrum alles andere als Anarchie: Die Erfüllung der Anforderungen an die Scrum-Rollen und die Einhaltung der Timebox-Regeln erfordern ein hohes Maß an Disziplin von allen Beteiligten.

duct Ownern muss hier das Qualitätsmanagement aktiv vertreten sein. Denn Qualitätsplanung und -lenkung erfolgen auf dieser Ebene, während die eigentliche Qualitätssicherung und -verbesserung häufiger auf Teamebene als auf Abteilungsebene stattfinden.

Im Scrum of Scrums sorgen Qualitätsmanager aktiv dafür, dass einerseits die globalen Bedürfnisse einer Organisation in den agilen Teams gelebt und andererseits teamübergreifende Bedürfnisse zur Qualitätssicherung hinsichtlich Tools, Testumgebungen und ähnlichem unterstützt werden. Um beide Aufgaben erfüllen zu können, sollten Qualitätsmanager das Wertschöpfungsprinzip anwenden können und verstehen, wie die Teams arbeiten. Der ideale QM-Beauftragte eines agilen Unternehmens ist nicht Verwalter der Prozessabweichungen, sondern führender Lean Coach unter den Führungskräften. □