

### Trends im Software Engineering

# Agile Welten

Von Hans-Henning Herzog

*Offenbar besteht ein grosser Bedarf nach Erneuerung in der Softwarebranche. Nirgendwo sonst ist das Scheitern so teuer und gleichzeitig so weit verbreitet. Doch wie kann der Kunde sein Softwareprojekt besser steuern und wie können die Teams Software effizienter entwickeln? MQ sprach darüber mit Karol Frühauf, Experte im Software-Projektmanagement und Leiter der SAQ Fachgruppe Informatik.*

**H**err Frühauf, warum wird Qualität im Software Engineering immer wichtiger? Überall greift Software in unser Leben ein. Immer mehr Software wird gebraucht, und das in immer kürzerer Zeit. Entwicklungen müssen immer schneller auf den Markt kommen. Und es gibt neue Bewegungen, auf welche Art man Software entwickeln kann oder soll. Interessant daran ist, dass die Qualität vorrangig wird.

*War das früher weniger der Fall?* Früher hat man viel über Funktionalität gesprochen und wenig über Qualität. Und am Schluss,

**Karol Frühauf**, Dipl. El. Ing., 1968 bis 1987 Software-Entwickler und Projektleiter bei BBC Brown Boveri AG in Baden, zuletzt in Führungsfunktion, seit 1987 Mitgründer und Teilhaber der INFOGEM AG, Informatiker Gemeinschaft für Unternehmensberatung, Leiter der SAQ Fachgruppe Informatik, Schulungstätigkeit in der Schweiz, in Deutschland, Tschechien und der Slowakei, zahlreiche Publikationen und Tagungsbeiträge. INFOGEM AG, Rütistrasse 9, CH-5401 Baden, Tel. +41 (0)56 222 65 32, karol.fruehauf@infogem.ch

wenn der Termindruck gross wurde, hat man Abstriche bei der Qualität gemacht und weniger bei der Funktionalität. Das war das klassische Vorgehen.

*Was hat sich daran heute geändert?* In der agilen Softwareentwicklung, wie sie genannt wird, kommt fast so etwas auf wie im alten Handwerker-Zunftwesen, also das Bewusstsein, dass man nur richtige Arbeit abliefern. Es wird genau das Notwendige entwickelt. Vielleicht machen wir weniger, wenn die Zeit oder das Geld ausgeht, aber das, was wir machen, ist richtig und nützlich. Das ist der Grundtenor dieser agilen Vorgehensweise.

*Woher stammt der Begriff?* Der kommt aus den USA. Populär wurde er, als nach 1999 Kent Beck und andere ihre Bücher zu «Extreme Programming» herausbrachten. In kleinen Schritten und kleinen Häppchen nähert man sich zielgerichtet den Anforderungen

des Kunden. Während der Entwicklung gibt es viele Feedback-Schleifen, und jedes Häppchen wird vom Kunden geprüft, sodass man nach Wochen statt nach Jahren erkennt, ob man in die falsche Richtung geht.

*Was heisst Qualität in diesem Zusammenhang?*

Nach agilem Verständnis ist Qualität geprägt vom Grad der Erfüllung von Anforderungen und Erwartungen. Wichtig: Der Kunde bestimmt, was Qualität ist, er wählt die Anforderungen aus und er legt fest, wie ihre Erfüllung am fertigen Produkt festgestellt wird.

*Die Sicht des Kunden und Nutzers ist alles entscheidend?*

Genau. Nach Noriaki Kano gibt es bekanntlich drei Arten von Anforderungen:

### In kleinen Häppchen zum Erfolg

die Basisanforderungen, die man nicht formuliert, aber natürlich erfüllen muss, so wie jedes Auto Räder hat, dann der mittlere Bereich der Leistungsanforderungen, über diese spricht der Kunde, weil er dort eine Lösung sucht, und dann gibt es die Begeisterungsfaktoren, etwas mit einem überraschenden Zusatznutzen, womit die Kunden

nicht rechnen, weil sie nicht wissen, dass es möglich ist.

*Was wäre denn etwas ganz Besonderes? Was Begeisterung auslöst?*

Die Klimaanlage oder ABS im Auto haben vor Jahren so etwas ausgelöst. Aber die Eigenschaften verändern sich mit der Zeit. Heute gehört die Klimaanlage zur Leistungs-, für viele sogar zur Basisanforderung.

*Agiles Vorgehen und die Qualitätsbrille, beides hat Konsequenzen für die Art, wie Software überhaupt entwickelt wird?*

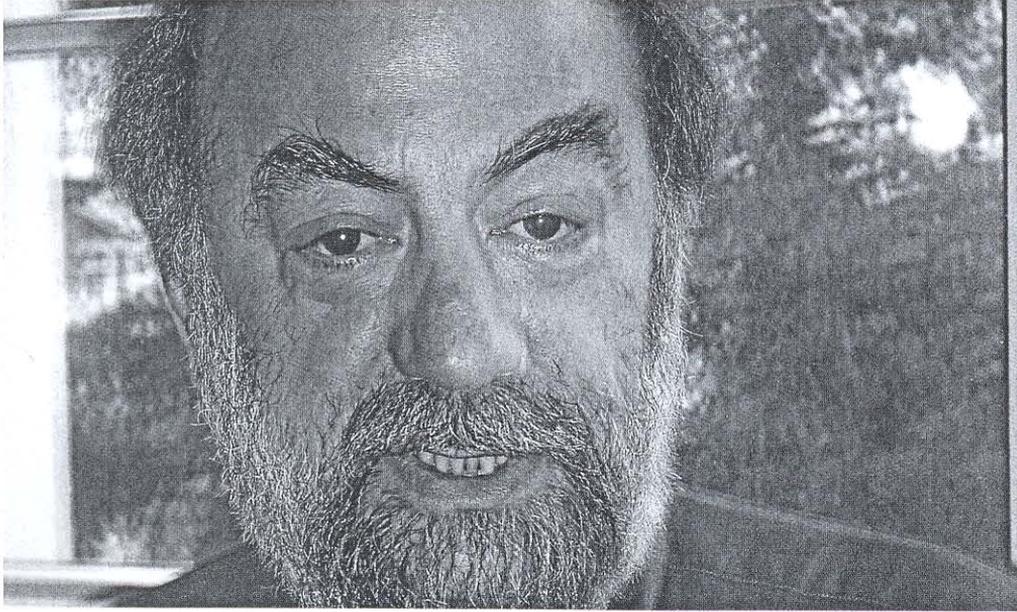
Ja, das wurde zum Teil völlig auf den Kopf gestellt. Zum Beispiel bei der Entwicklung einer Komponente, da überlegt man zunächst, wie man sie prüfen kann und dann erst beginnt man die Software zu schreiben. Beim klassischen Vorgehen spezifiziert man, was man haben will, dann entwirft man es, kodiert und dann testet man es. Beim Extreme Programming dagegen erfasst man die Anforderungen und fragt sich, wie prüfen wir die Software, die diese Anforderungen erfüllen soll. Erst dann entwickeln wir die eigentliche Software. Man macht die Messanordnung, ohne dass das Messobjekt eigentlich vorhanden ist.

*Und dieses Vorgehen hat sich heute durchgesetzt?*

Nein, noch nicht. Aber es gibt immer mehr Leute, die es anwenden. Es ist auch nicht einfach. Mental ist das sehr problematisch, weil die Denkweise durch die «Erziehung» geprägt ist, und die ist zumeist klassischer Natur.

*Ich würde mich auch fragen, wie ich etwas testen soll, das noch nicht vorhanden ist ...*

Am ehesten vergleichbar vielleicht mit dem Nachweis der Elementarteilchen. Der Vorteil des «test-driven development» aus



## QUALITÄT SICHERN

### «Man muss nicht alle Anforderungen kennen, bevor man losmarschiert» Karol Frühauf

Projektmanagementsicht ist der Nachweis des Fortschritts. Wenn der Programmierer mit dem Testprogramm fertig ist, dann drückt er den Knopf und bei allen Testfällen kommt «gescheitert» he-

#### Erst prüfen, dann machen

raus; alles andere wäre ein Wunder, weil noch kein Code da ist. Beim Programmieren kann er immer wieder den Testknopf drücken und wenn er zurück ist mit dem Kaffee, sieht er wunderbar den Fortschritt.

*Zurück zu der iterativen Vorgehensweise. Was bedeutet das praktisch für den Kunden?*

Das Wichtigste: Man muss nicht alle Anforderungen kennen, bevor man losmarschiert. Wenn der Kunde eine Vision hat und genügend darüber weiss, was ihm Nutzen bringt und wie viel er bereit ist dafür auszugeben, dann heuert er ein Team von Entwicklern an. Zusammen bestimmen sie, welche von den bekannten Anforderungen den grössten Nut-

zen bringen und vom Team in einem Monat entwickelt werden können. Und das wird realisiert. Wichtig: Der Kunde hat das Recht, es sich unterwegs anders zu überlegen, wenn er bereit ist, auf bereits in Auftrag gegebene Anforderungen zu verzichten.

*Eine Änderung der Anforderungen während der Entwicklungsarbeit ist also jederzeit möglich?*

Ja, ich sage meinen Seminarteilnehmern immer: Das ist ein Naturgesetz, es hat keinen Sinn, sich dagegen zu stemmen, das gehört zum Projekt. Wenn ich es als «Gravitationsgesetz der Softwareentwicklung» benenne, dann schmunzeln die meisten.

*Und in einem Monat liegt ein Ergebnis vor?*

Ja, das ist die Regel. Die Idee ist, dass man innerhalb eines Monats oder in sechs Wochen, je nachdem wie man die Iteration legt, etwas produziert, das man nutzen kann. Nicht Papier, sondern ein funktionsfähiges Programm.

*Und so geht es Schritt für Schritt weiter?*

Ja, so tastet man sich in kleinen Häppchen an das Endprodukt

heran. Jeder Schritt wird intensiv diskutiert und abgenommen. Sobald der realisierte Nutzen es rechtfertigt, wird eingeführt. Und dann kommen die nächsten Anforderungen dran, usw.

*Gibt es dafür auch praktische Beispiele? Wie konkret ist das?*

Die erste Anwendung ist das Lohnsystem bei Chrysler. Das ist das Paradebeispiel. Da hat man zum ersten Male solche Ansätze des Extreme Programming gewählt. Und hat in zwölf Monaten erreicht, was vorher in drei bis fünf Jahren vergeblich versucht wurde. Das sagt die Legende. Es gibt Beispiele in der Telekommunikationsindustrie, Versicherungswesen und anderen Branchen.

*Das alles ist heute der Topstandard für die Entwicklung?*

Anfang 2000 war es ein Hype, inzwischen hat es sich beruhigt, aber es wird nachhaltige Konsequenzen für die Softwareentwicklung haben. Es wird nicht so radikal, wie es heute formuliert wird, überleben, aber viele Ideen und Techniken werden Eingang finden in jegliche Vorgehensweise, die kommen wird. Davon bin ich überzeugt.

*Kann die Softwareentwicklung nicht auch vom klassischen Projektmanagement profitieren?*

Sicher kann man da viel übernehmen. Aber Software ist anders. Software ist nicht «materiell». Es gibt keine physikalischen Grenzen. Deshalb überschätzt man sich leicht in Projekten. Zum Beispiel, dass man zu komplexe Sachen an die Hand nimmt. Weil alles, was denkbar ist, auch in Software umsetzbar ist. Softwareentwicklung ist eigentlich nur Denkarbeit. Wir müssen nicht Material verformen, löten oder schweissen. Wir müssen nur Gedachtes in Sprache giessen und transformieren, damit die Maschine es ausführen kann.

*Aber es gibt schon Grenzen, allein finanziell, aber auch technisch ...*

Ja, aber die technischen sieht man nicht am Anfang. Das stellt man erst fest, wenn man unterwegs ist. Wichtig ist, wann erkennt man die Grenze. Das Agile löst nicht alle Fragen. Aber es hilft uns, die grossen Probleme früher zu erkennen. Das ist der bescheidene Anspruch von Ken Schwaber, dem Vater von «Scrum».

*Schon wieder ein neuer Begriff und eine neue Methode?*

Scrum ist eine agile Spielart des Projektmanagements und ist nicht nur in der Softwareentwicklung anwendbar. Scrum wurde

### Das Agile lässt Probleme früh erkennen

zum Beispiel bereits in der Entwicklung von Herzschrittmachern eingesetzt.

*Scrum und die anderen agilen Methoden fordern nicht nur die Teams in der Entwicklung, sondern vor allem auch die Kunden heraus. Wie können die als Nichtspezialisten überhaupt mitreden?* .....→

## QUALITÄT SICHERN

Vor allem müssen die Kunden, auch wenn sie keine Informatikausbildung haben, fähig sein, ihre Anforderungen zu äussern und die Abnahmekriterien zu bestimmen. Um die Aufgabe zu bewältigen, sind sie auf Weiterbildung angewiesen. Dass sie die benötigten Kenntnisse erworben haben, können sie seit ein paar Jahren mit einem Zertifikat bescheinigt bekommen.

*Was für Zertifikate kann man erwerben?*

Die Vorreiter waren bei uns die Software-Tester. Die Prüfungsfragen für den SAQ Certified Software Tester werden vom Swiss Testing Board zusammengestellt. Basis bildet der vom International Software Testing Qualifications Board (ISTQB) erstellte Lehrplan, der mittlerweile in ca. 30 Ländern Anwendung findet.

### Zertifikate für Nichtinformatiker

Unser Swiss Testing Board arbeitet sehr aktiv auf der internationalen Ebene mit. Wie in Deutschland das German Testing Board auch.

Seit einem Jahr gibt es auch den Professional for Requirements Engineering. Hier werden die Fragen vom International Requirements Engineering Board (IREB e.V.) erstellt, der auch den Lehrplan erarbeitet hat. Unser Swiss Requirements Engineering Board leistet dazu einen wesentlichen Beitrag.

*Wer vergibt die Zertifikate?*

Die Personalzertifizierungsstelle der SAQ nimmt die Prüfungen ab und stellt die Zertifikate aus. Heute führt SAQ etwa 1500 zertifizierte Tester und gut 150 zertifizierte Requirements Engineers.

*Was ist der Nutzen des Zertifikats?*

Bankfachleute können ihrem (neuen) Arbeitgeber vorweisen, dass sie neben dem Bankfach auch das Metier des Testens verstehen. Das wird immer wichtiger, weil sie es ja sind, die in ihrem Unternehmen und in ihrer Organisation gehalten sind, die Software zu testen, bevor sie in Betrieb geht.

*Die SAQ hält bei den Zertifizierungen die Fäden in der Hand?*

Und zwar über die SAQ Fachgruppe Informatik. Die Boards werden von der Fachgruppe ernannt, und die Fachgruppe ist die übergeordnete Lenkungs- und Überwachungsinstanz, zum Beispiel auch bei Rekursen.

*Die Fachgruppe Informatik hat ja gerade im Januar 2008 ihr 25-Jahr-Jubiläum gefeiert. Sie sind ihr Leiter. Was macht die Fachgruppe konkret?*

Unser Ziel ist, der Schweizer Wirtschaft auf dem Wege zu besseren IT-Systemen, zu besserer Software behilflich zu sein. Wir haben zurzeit ungefähr 70 Mitglieder. Davon sind 25 bis 30 wirklich aktiv. Vor allem in unseren Arbeitsteams. Das ist unser wichtigstes Standbein. Hier diskutieren wir unterschiedlichste Informatikthemen und tauschen Erfahrungen aus. Leute in ähnlicher Position und mit ähnlichen Problemen, über Firmengrenzen hinweg, arbeiten an bestimmten Themen. Das bringt sehr viel.

*Was sind das für Themen?*

Sehr erfolgreich war zum Beispiel das «Vademecum des Software-Testens». Dann haben wir ein Assessment-Modell für IT-Betriebe aufgestellt.

*Organisiert die Fachgruppe auch Weiterbildungsveranstaltungen?*

In diesem engen Sinn, nein. Neben der Zertifizierung mit den

### Publikationen zum Thema

- Kent Beck: Extreme Programming – das Manifest. Die revolutionäre Methode für Softwareentwicklung in kleinen Teams, Addison-Wesley, 2000, ISBN 3-8273-1709-6.
- Ken Schwaber: Agile Project Management with Scrum. Microsoft Press, 2004, ISBN 0-735-61993-X (aktuelles Standardwerk über Scrum).
- Henning Wolf, Stefan Roock, Martin Lippert: eXtreme Programming: eine Einführung mit Empfehlungen und Erfahrungen aus der Praxis, dpunkt, 2. Auflage, 2005, ISBN 3898643395.
- Wikipedia verfügt in seinen Themenportalen über detaillierte Informationen zum Extreme Programming und Scrum.

### Internationale Gruppen

- The Agile Alliance, mit ca. 4000 Mitgliedern weltweit derzeit eine besonders aktive Gruppe. Der nächste Konferenztermin: Agile 2008 Conference – Toronto, 4. bis 8. August 2008 ([www.agilealliance.org](http://www.agilealliance.org)).
- XP Days Germany, Konferenz über eXtreme Programming und agile Softwareentwicklung. Zielpublikum sind Softwareentwickler, Projektleiter, IT Manager, Tester, Architekten und Coaches. Die Konferenz wird jährlich von andrena objects AG (Karlsruhe) und akquinet AG (Hamburg) ausgerichtet. ([www.andrena.de](http://www.andrena.de), [www.akquinet.de](http://www.akquinet.de)).
- [www.scrumalliance.org](http://www.scrumalliance.org) – offizielle Website der internationalen Scrum-Gruppe.

### SAQ Fachgruppe Informatik

- Infos über [www.saq.ch](http://www.saq.ch); zum Certified Software Tester: [www.software-tester.ch](http://www.software-tester.ch); zum Requirements Engineering: [www.certified-re.de](http://www.certified-re.de).
- Anlässlich des Swiss Testing Day (19. März, Kongresshaus Zürich) können Interessierte in Tutorials Prüfungsvorbereitungen zum Certified Tester – (inkl. Prüfung) besuchen (Leiter Reto Müller, SynSpace AG, ein Tag in D, 18. März, Zürich), [www.swisstestingday.ch](http://www.swisstestingday.ch), Veranstalter: International Software Testing Qualifications Board (ISTQB).

Boards geht es beim dritten Standbein um die Streuung von Wissen. Früher haben wir Konferenzen organisiert. Heute favorisieren wir «Zug um Zug»-Veranstaltungen.

*Was verstehen Sie darunter?*

Der Name kommt daher, weil es sich um Abendveranstaltungen in der Nähe von Bahnhöfen handelt, Bern, Olten, Zürich, Basel. Jedes Jahr konzentrieren wir uns auf ein Thema. Das wird auf der Jahresversammlung durch einen Keynote-Speaker eingeläutet. Jetzt läuft zum Beispiel Software-Wartung. Frühere Zug um Zug-Themen waren Testen und Prozesse. Es bewährt sich, Mitglieder der Fachgruppe und auch Nichtmitglieder nehmen mit viel Elan und Interesse daran teil.

*Verfügt die Fachgruppe über internationale Verbindungen?*

Wir sind international vernetzt, aber kaum offiziell. Zwar gibt es

auf EOQ-Ebene auch eine Software-Fachgruppe. Aber die EOQ hat nie die Fachgruppenarbeit unterstützt. Für Brüssel existieren wir gar nicht. Das motiviert die Leute in der Softwaregruppe nicht gerade. Wir haben einfach auf persönlicher Grundlage internationale enge Kontakte, natürlich vor allem zum deutschsprachigen Raum. Interessanterweise haben weder DGQ noch ÖVQ solche Fachgruppen wie wir. Da sind andere Vereinigungen in die Bresche gesprungen.

*Was wünschen Sie sich für die Zukunft?*

Natürlich, dass die Fachgruppe noch lange so intensiv oder noch intensiver weiter arbeitet und ihre Wirkung steigert. Dann bleibt sie sicher stark verankert in der SAQ und in der Schweizer Software-engineering-Landschaft.

*Karol Frühauf, vielen Dank für dieses Gespräch.* ■