

## Argumente für ein agiles Wissensmanagement

Gerhard Melchard

*„Der Ball ist rund und ein Fußballspiel dauert 90 Minuten.“ Aus dem aktuellen Anlass der kurz bevorstehenden Fußballweltmeisterschaft soll einmal dieses Diktum der deutschen Fußballlegende Sepp Herberger als Alternative für den Hinweis auf die Konstanz der Veränderung dienen. Mit diesen Worten hat er auf die Dynamik und die Unvorhersehbarkeit eines Fußballmatches hingewiesen. Der Ball ist rund, damit das Spiel die Richtung ändern kann. Und wir erwarten auch von einem interessanten Fußballmatch, dass es die Richtung wechselt, dass einmal auf das eine Tor gestürmt wird und dann wieder auf das andere. Nichts ist langweiliger als ein einseitiges Spiel oder gar ein statisches und regungsloses Geplänkel und Gestoche im Mittelfeld. Wenn ein Match spannend ist, kann niemand wirklich antizipieren, wie sich der nächste Spielzug entwickeln wird. Die Spannung konstituiert sich aus dieser Unvorhersehbarkeit. In der Vorbereitung und in der Halbzeit eines Matches können Taktiken besprochen und im Team vereinbart werden, eine Strategie kann ausgearbeitet werden, aber es kann nicht minutiös vorgeplant werden. Was in den 90 Minuten wirklich passiert, weiß beim Anpfiff keiner.*

### Softwareentwicklung als Fussballspiel

Die Metapher des Fußballspiels scheint mittlerweile auch besser für die Softwareentwicklung geeignet als die des Hausbaus. Software Engineering wird heute nicht mehr so sehr als die ingenieurmäßige Planung und Konstruktion eines Artefakts verstanden, sondern eher als ein kooperatives Spiel. Soft facts sind wichtiger als hard facts. Die Validität eines Entwurfs vom Reißbrett ist nicht der entscheidende Erfolgsfaktor sondern die Kommunikation der beteiligten Akteure. Die im Rahmen eines IT-Projektes entworfene Software ist Ergebnis von intersubjektiven Aktivitäten. *Panta rhei* lautet das Motto. Nicht die Einhaltung eines Plans stellt die Maxime dar, sondern das Reagieren auf die Veränderungen, die ohnehin im Laufe des Projekts stattfinden werden. Exakte ausformulierte Vorgaben sind in diesem

Kontext nicht möglich, da sich die Ziele ohnehin im Laufe des Projekts ändern werden. Sobald sie fixiert sind, werden sie obsolet. On-the-fly-tuning, also ständige Anpassung an volatile Rahmenbedingungen, erhöht die Erfolgchancen eines Projekts im Spannungsfeld des magischen Dreiecks von Qualität, Zeit und Kosten.

### Prinzip der Agilität

Diesen Umständen gerecht zu werden, ist Anspruch des Prinzips der Agilität. Dieser Begriff, bedeutet Lebendigkeit, geistige und körperliche Wendigkeit, eine Eigenschaft, die auch von einem guten Fußballer erwartet werden kann. Unter agiler Softwareentwicklung wird die Anwendung des Agilitätsprinzips auf den Softwareentwicklungsprozess verstanden. Sie zeichnet sich durch eine geringe Bürokratisierung und die Definition von einige wenigen Regeln aus, die die Effektivität eines Softwareentwicklungsprojekts gewährleisten sollen. Erstmals wurden diese 2001 von 17 Softwaremethodikern in einem abstrakten Kanon, dem „Agile Manifesto“, formuliert. Es postuliert das Primat von

- Individuum und Interaktion vor Prozessen und Werkzeugen,
- funktionierender Software vor umfangreicher Dokumentation,
- Kundenzusammenarbeit vor Vertragsverhandlungen und
- Reaktion auf Änderung vor Planerfüllung

Agiles Projektmanagement heißt nicht Absenz von Methodik oder gar *laissez faire*. Gerade in einem Umfeld, in dem sich die Rahmenbedingungen rasch ändern können, muss die Richtung klar definiert werden, um ans Ziel zu gelangen. Allerdings braucht es dafür nicht eine Methodenschule im Sinne eines konsistenten Regelwerks, sondern ein eklektizistische Selektion verschiedener Methoden und Techniken. Sowie in der postmodernen Kunst nicht die Homogenität eines Stils bestimmend ist, sondern die Rekombination und das Zitat, werden in der agilen Softwareentwicklung Werte, Prinzipien und Prozesse, die einem bestimmten Muster folgen, zusammengefasst.

Unter agilen Prozessen werden Vorgehensweisen verstanden, die sich agiler

Methoden und Prinzipien bedienen. Bekannte Ansätze in diesem Zusammenhang sind:

- Crystal
- Feature Driven Development FDD
- Adaptive Software Development ASD
- Scrum
- Extreme Programming XP

An dieser Stelle sei ganz besonders auf Scrum hingewiesen, weil der namensgebende Begriff aus dem Sport stammt. Beim Rugby versteht man darunter die Besprechung des Teams auf dem Spielfeld, bei der der nächste Durchbruch, Sprint genannt, geplant wird. Bei Scrum werden Projekte in Serien von 30-tägigen Zyklen (Sprints) bearbeitet. Dabei sind tägliche Scrums unter Beteiligung des Auftraggebers vorgesehen, um auf die Änderungen und Priorisierungen angemessen reagieren zu können.

Kurze Entwicklungszyklen und inkrementelles Vorgehen sind Charakteristika dieser agilen Prozesse. Implementierung geht vor Spezifikation. Lieber wird ein Rapid Application Prototyp entwickelt und mit dem Kunden abgestimmt als detaillierte Analyse und Design ausformuliert. Iterationen mit möglichst raschem Kundenfeedback haben sich mit der Objektorientierung als dem Paradigma des Software Engineering etabliert. Bei Extreme Programming ersetzen die direkte Kommunikation und der entwickelte Sourcecode die Dokumentation der Implementierung.

Das konventionelle Projektmanagement ist mit solchen Verfahren inkompatibel. Die starre Orientierung an einem Vorgehens- bzw. Phasenmodell führt nur dazu, dass auf Biegen und Brechen Milestones eingehalten werden, aber nicht die gewünschte Lösung geliefert wird. Hier ist ein wesentlich flexibleres Vorgehen gefragt, das rasch und proaktiv mit Risiken fertig werden kann. Riskomanagement ist eine Schlüsseldisziplin des agilen Projektmanagements.

### **Risiken eines Softwareentwicklungsprojekts**

Eines der Hauptrisiken eines Softwareentwicklungsprojekts ist es, die Kundenanforderungen zu verfehlen. Daher kommt dem Requirement Engineering eine besondere Bedeutung zu. Die Anforderungen werden in Form von Workshops mit dem

Anwender definiert. Dabei finden die gängigen Werkzeuge wie Flipchart und Whiteboard Anwendung. Für die Spezifikation können neben der Umgangssprache auch Methoden wie UML oder ERD verwendet werden. Dabei werden üblicherweise nicht teure Casetools eingesetzt, sondern Standardanwendungen wie Powerpoint, Visio oder Dia. So entsteht in Form des Anforderungsmodells der wichtigste Bezugspunkt für die Planung und Steuerung des Projekts.

Die Dokumentation dient dabei nicht mehr als Kriterium für die Verfolgung des Projektfortschritts. Das Ziel ist vielmehr die Unterstützung der Kommunikation im Team, mit dem Kunden und mit den Stakeholdern. Aufgabe des Projektmanagers ist dabei die Steuerung und Koordination der Kommunikation. Wichtig ist dabei auch die Angemessenheit der eingesetzten Mittel zur Zielerreichung. Der getätigte Aufwand muss so gerade noch ausreichend sein.

### **Methoden der Wissensrepräsentation**

Agile Softwareentwicklung erfordert agiles Projektmanagement. Dieses wiederum braucht auch leichtgewichtige, emanzipatorische Methoden der Wissensrepräsentation. Ein geeignetes Medium für die Darstellung der Anforderungen, die auch der Distribution des Wissens im Projektteam dient, ist ein WikiWiki, eine strukturierte über Links verbundene Seitensammlung im Inter- bzw. Intranet, die nicht nur von allen gelesen sondern auch geändert werden kann. Der Begriff, der übrigens aus der Sprache der hawaiianischen Ureinwohner stammt, bedeutet schnell. Das Wiki eignet sich grundsätzlich für die Zusammenfassung von Projektdokumenten. So können dort die Use Cases, SW-Architektur-Charts, Geschäftsprozessmodelle, Testcases usw. abgebildet werden. Es kann über das Projektende hinaus auch als Medium für die Repräsentation von Wissen über die entwickelte Applikation verwendet werden.

Eine weitere Möglichkeit der Unterstützung des Wissenstransfers besteht in der Anwendung von Weblogs, meist nur Blogs genannt. Dabei handelt es sich ebenfalls um eine webbasierte Publikation, die Einträge enthält, die in der Regel in umgekehrter chronologischer Reihenfolge angeordnet sind. Hier können relevante Termine, wie Deployments

angekündigt, Neuigkeiten bekanntgegeben, sowie bestimmte Erfahrungen manifestiert und zugänglich gemacht werden.

Last but not least sei noch auf eine wesentliche soziale Technik hingewiesen, die der Aufbereitung der im Projekt erworbenen Erfahrungen dient. Im Rahmen eines Reflexionsworkshop werden die Lessons Learned erarbeitet, um sie über das Projekt hinaus zu persistieren. Zur Entwicklung einer lernenden Organisation kann damit zumindest auf der Ebene des Teamlearnings ein Beitrag geleistet werden. Für die pointierte Beschreibung der Motivation, das im Laufe des Projekts durch Erfahrung gewonnene Wissen für Folgeaktivitäten verfügbar zu machen, soll zum Abschluss noch einmal ein Zitat Sepp Herbergers dienen: „Nach dem Spiel ist vor dem Spiel.“

### Der Autor

Gerhard Melchard, MSc. ist derzeit Teamleader und Projektmanager in einem Telekommunikationsunternehmen. Er hat langjährige und extensive Erfahrung in Softwareentwicklung, IT und Organisation insbesondere im Bereich Business Analysis, Methodensupport sowie Projekt- und Prozessmanagement.




---

### Literatur

#### Agiles Projektmanagement, Definition im Projektmanagement-Glossar.

<http://www.projektmagazin.de/glossar/gl-0775.html?pmSession=439b852861c6576f4a092e6d75c363ee>. (gelesen am 15.5. 2006).

**Kent Beck.** Extreme Programming. Das Manifest. Addison Wesley. München 2003.

**Alistair Cockburn.** Agile Software Development. Addison Wesley. Boston 2002.

**Alistair Cockburn.** What the agile toolbox contains. <http://alistair.cockburn.us/crystal/articles/wtadc/whattheagiletoolboxcontains.htm>. (gelesen am 15.5.2006).

**Christiane Gernert.** Agiles Projektmanagement oder wieviele Regeln brauchen unsere Projekte? [http://www.stz-itpm.de/link.doc/AgilesPM\\_Gernert\\_20050128.pdf](http://www.stz-itpm.de/link.doc/AgilesPM_Gernert_20050128.pdf). (gelesen am 15.5. 2006).

**Christiane Gernert.** Agiles Projektmanagement. Risikogesteuerte Softwareentwicklung. Carl Hanser Verlag. München, Wien 2003.

**Manifesto for Agile Software Development.** <http://agilemanifesto.org>. (gelesen am 15.5.2006).

**Gernot Starke.** Agile Entwicklungsprozesse. An Flexibilität wachsen statt daran zu scheitern. <http://www.b-agile.de/de/index1.html>. (gelesen am 15.5.2006).