

Kollabieren oder Kollaborieren

(Link zum Artikel: <http://www.it-republik.de/dotnet/artikel/2973>)

Methodologien, Prozesse und Metaebenen

Text: Uta Kapp, Jean Pierre Berchez

Wenn Begeisterung in einem Softwareentwicklungsteam vorhanden ist, dann wird miteinander kommuniziert und sich engagiert. Dies ist die Voraussetzung für eine effiziente Zusammenarbeit in einem lernenden Unternehmen. Eine Collaboration-Plattform fördert die Kommunikation und unterstützt Vorgehensmodelle und Best Practices eines Teams.

Softwareentwicklung ist im Vergleich zu anderen Ingenieurwissenschaften eine neue Disziplin. Es hat Jahrtausende gedauert, bis man Brücken und Häuser in der Qualität bauen konnte, wie das heute der Fall ist. Selbst das Bauen von Autos ist eine alte Disziplin, wenn man sie mit der Softwareentwicklung vergleicht. Software ist, im Unterschied zu Autos, Häusern und Brücken, ein nichtmaterielles Gut. Es hat also Ähnlichkeit mit Büchern und Musik. Im Laufe der Jahre haben sich Verfahren entwickelt, die sich für das Management von Softwareprodukten bewährt haben. Da die Entwicklung immer komplexer wird und Anforderungen an Geschwindigkeit und Flexibilität wachsen, steigt auch die Notwendigkeit, bei dem Vorgehen zur Entwicklung ständig zu lernen. Der Versuch, Software fabrikmäßig und arbeitsteilig herzustellen, wie es zum Beispiel bei einer Produktionsstraße in der Automobilfertigung praktiziert wird, hat nicht gut funktioniert. Der Wissensverlust bei der Übergabe von einer Disziplin an die andere ist zu groß. Es ist schwierig, alles Wissen zu transferieren, wenn eine Person eine Anforderungsanalyse durchführt, die nächste Person das Softwaredesign, eine weitere Person die Kodierung übernimmt, bevor die vierte Person das Produkt testet. Zu viel Wissen, das intuitiv gelernt wird, geht bei der Kommunikation verloren. Dies führt zu einer Art stille Post. Jeder hat sein eigenes Modell im Kopf. Am Ende kommt ein völlig anderes Produkt heraus, als am Anfang beabsichtigt war. Auch der Ansatz, ein Softwareprodukt vollständig im Voraus zu planen, wie zum Beispiel bei dem Bau einer Brücke üblich und erforderlich, hat sich als nicht praktikabel erwiesen. Die Hälfte der vorher geplanten Funktionen erweist sich als nutzlos oder so nicht einsetzbar. Während der Entwicklung findet ein Lernprozess statt, und die richtige Funktionalität kristallisiert sich oft erst am Prototyp heraus. Dazu sind lauffähige und testbare Versionen in einem frühen Stadium notwendig.

Iteratives Vorgehen

Der Ausweg aus dem Dilemma ist, von vornherein ein iteratives Vorgehen zu wählen. Diese Vorgehensweise besteht aus kleinen, zeitlich fixierten Produktzyklen, deren Ergebnisse jeweils lauffähig sind. Das Ergebnis wird evaluiert. An der neuen Version lernen alle am Projekt Beteiligten und konstruieren so die nächste verbesserte Runde. Das Wissen, das laufend hinzukommt, muss kommuniziert werden und ständig verfügbar sein. Dieser Lernprozess funktioniert nur dann, wenn die Bürokratie auf das notwendige Minimum reduziert wird. Sich selbst organisierende Teams, die ohne Hierarchien auskommen, haben sich hier als am effizientesten gezeigt. Überbordende Bürokratie ist ein Ausdruck von Machtmissbrauch, Kontrollsucht und mangelndem Vertrauen. Es ist im Kern ein Hinweis auf schwache Führungsqualitäten der verantwortlichen Manager. Deshalb ist es wichtig, sich

grundsätzlich mit den Prinzipien von gutem Führungsstil auseinanderzusetzen, bevor man sich mit dem Einsatz von speziellen Vorgehensmodellen und Prozessen für die Softwareentwicklung auseinandersetzt. Erst der nächste Schritt ist es dann, eine Collaboration-Plattform einzusetzen, um diese Prozesse zu unterstützen.

Iteratives und empirisches Vorgehen hat sich bewährt, damit der Kunde und Anwender möglichst viel Funktionalität für seine Investition bekommt und die Ressourcen optimal genutzt werden. Iteratives Vorgehen ist schon so alt wie die Softwareentwicklung selbst. Mit der Bewegung, die agile Softwareentwicklung genannt wird, hat dieses Vorgehen ein strukturiertes Gesicht bekommen. Die Kunst bei iterativen Ansätzen ist es, Spielregeln für den Entwicklungsprozess festzulegen und dafür zu sorgen, dass alle Beteiligten sich daran halten. Es muss genau spezifiziert werden, was in jedem Inkrement erreicht werden soll. Auch wenn Veränderungen als Teil des Lernprozesses im gesamten Entwicklungszyklus gewünscht sind, darf nicht auf Zuruf täglich alles in Frage gestellt werden. Eine klare Modellierung der Software und eine ordentliche Dokumentation sind die Grundlage jeder Entwicklung. Agiles Vorgehen ist nicht gleichzusetzen mit Chaos, sondern mit besonders hoher Disziplin. Diese wird mit Hilfsmitteln zur Selbstorganisation erreicht. Das Projektmanagement-Framework Scrum ist für die Softwareentwicklung populär geworden. Es wurde ursprünglich von Hirotaka Takeuchi und Ikujiro Nonaka als Modell für die industrielle Produktentwicklung vorgestellt [1]. Ken Schwaber und Jeff Sutherland haben dieses Modell am Beispiel der Softwareentwicklung in einen formalen Rahmen gebracht [2].

Collaboration-Plattformen einsetzen

Ein empirisches Vorgehen erfordert einen kontinuierlichen Wissensaustausch im Team und mit den Anwendern. Das stellt Transparenz im Projekt her. Zur Unterstützung der Kommunikationsprozesse hat sich der Einsatz von Collaboration-Plattformen bewährt. Dies ist Software zur Unterstützung von Softwareentwicklung. Es gibt eine Vielzahl von integrierten Collaboration-Plattform-Suiten, die leicht zu bedienen sind und den gesamten Applikations-Lebenszyklus verwalten. Diese Plattformen werden auf zentralen Servern installiert, am besten im Internet, und können mit gängigen Internetbrowsern bedient werden. Die Kommunikation, während des gesamten Lebenszyklus eines Softwareprojektes, von der Planung bis zur Kundenbetreuung, wird so optimiert. Team-Collaboration-Plattformen unterstützen folgende Kommunikationswege:

1. *Wissensaustausch*: Wikis, Foren und andere Technologien, die wir aus der Web-2.0-Welt kennen, finden hier Anwendung für ein flexibles Wissensmanagement. Das Team kann Informationen gemeinsam sammeln und bearbeiten.
2. *Anforderungsmanagement*: Fehlermeldungen, Änderungswünsche und Anforderungen werden in speziell dafür vorbereiteten elektronischen Karteien, den Trackern, auch Work Items genannt, abgelegt. Der letzte Projektstand ist so für alle zugänglich.
3. *Berichtswesen*: Wenn alle Informationen zu einem Projekt in digitaler Form vorliegen, dann kann ein ausgefeiltes Berichtswesen Informationen liefern, die den aktuellen Projektstand offen legen. So wird der Fortschritt messbar. Diese Berichte und Metriken sind auf der Plattform für jedes Mitglied in Selbstbedienung einsehbar.
4. *Benachrichtigungen*: Eine Collaboration-Plattform übernimmt die automatische Benachrichtigung bei Ereignissen, die für Teammitglieder wichtig sind. Dies können

zum Beispiel neue Produktversionen, neue Fehlermeldungen oder andere Neuigkeiten sein.

5. *Zugriffsschutz und Rollenverteilung*: Detaillierter Zugriffsschutz verhindert eine ausufernde Informationsflut. Jeder bekommt nur die Nachrichten zugestellt, die er benötigt. Niemand darf von wichtigen Informationen ausgeschlossen werden.

Links & Literatur

[\[1\] The New New Product Development Game by Hirotaka Takeuchi and Ikujiro Nonaka, Harvard Business Review, January-February 1986](#)

[\[2\] Agiles Projektmanagement mit Scrum von Ken Schwaber und Thomas Irlbeck, Microsoft Press Deutschland; Auflage: 1., Aufl. \(4. Oktober 2007\)](#)

***Uta Kapp** arbeitet als freiberufliche IT-Beraterin und systemischer Coach. Mit einer Kombination aus Fachberatung und Prozessberatung für Softwareprojekte hilft sie Entwicklungsteams bei der Bewältigung der ständig steigenden Komplexität. Hier kommen die agile Softwareentwicklungsmethode SCRUM und Organisationsaufstellungen zum Einsatz. Der Einsatz von Collaboration-Plattformen ist ein weiterer Schwerpunkt.*

***Jean Pierre Berchez** ist Geschäftsführer der [HLSC UG](#) und beschäftigt sich seit mehr als 15 Jahren mit den Themen Projektmanagement, Software Engineering und objektorientierte Softwareentwicklung. In den letzten Jahren liegt sein Interesse insbesondere auf den Themengebieten agile Entwicklung mit Schwerpunkt Scrum sowie "Collaborative" Software Development. Neben seiner Tätigkeit als Geschäftsführer ist Jean Pierre Berchez auch als Lehrbeauftragter an den BAs Stuttgart und Heidenheim sowie an der Hochschule Liechtenstein für die Themen "Anforderungsmanagement", "Scrum" und "Collaborative Software Development" tätig. Er organisiert unter anderem den Community-Event "[Scrum-Day](#)" in Deutschland.*

Original Artikel auf:

<http://it-republik.de/dotnet/artikel/Kollabieren-oder-Kollaborieren-2973.html>

andere Artikel dieser Serie

- [Kollaborieren mit Plattform](#)
- [Integrationen und Schnittstellen – "Wie es Euch gefällt"](#)
- [Bughunting](#)
- [Was macht ein Open Source-Projekt erfolgreich?](#)
- [Metriken, Statistiken, Dashboards](#)
- [Kommunikation zwischen Entwicklern, Benutzern und Management](#)
- [Wikinomics](#)
- [Integration statt Brückenbau](#)
- ["Projekt Gold" oder: Der Weg führt zum Ziel](#)

