

Boris Gloger: Positionspapier Projekte 2.0



Kontinuierliches Liefern von Produktteilen

Wie baut man in 180 Tagen einen völlig neuen Kampffjet? Konstrukteur Kelly Johnson von Lockheed Martin hat es 1943 einfach so gemacht: Alle dafür notwendigen Ingenieure in ein Zelt packen, Störungen fern halten und die Experten selbstorganisiert machen lassen. Ergebnis: Die P-80, fertig entwickelt nach 143 Tagen. Und die Geburt von „Skunk Works“, der weitgehend autarken, bürokratiefreien Entwicklungsabteilung für radikale Innovationen bei Lockheed Martin.

Agile Prinzipien und Vorgehensweisen sind also nichts Neues oder auf die Entwicklung von Software beschränkt. Angesichts wachsender Komplexität wurden sie in den 1990er-Jahren allerdings in der Softwareentwicklung wiederentdeckt und als nun offiziell „agil“ bezeichnete Verfahren wie XP, Crystal oder eben Scrum erfolgreich wiederbelebt. Heute sind auch traditionelle Branchen wie der Maschinenbau zu kurzen Entwicklungszyklen gezwungen und stoßen dabei mit der klassischen Projektplanung an ihre Grenzen. Der Stolperstein ist die Meinung, dass Fehlannahmen zu Beginn eines Projekts billiger auszugleichen seien als am Ende des Projekts, wie man auch im PMBOK nachlesen kann. Auf den ersten Blick erscheint diese Annahme logisch, allerdings beweist die Praxis immer wieder das Gegenteil – das zeigen zum Beispiel die Zahlen des „Chaos Manifesto“ der Standish Group, einer fortlaufenden Studie zum Erfolg von IT-Projekten. Selbstverständlich ist es auch bei klassisch geführten Projekten prinzipiell möglich, in der Anfangsphase Prototypen zu entwickeln und Tests durchzuführen. Prototypen in einer frühen Phase wären leicht einzuplanen. Aber das eigentliche Prinzip des Lernens durch die Prototypen, die den Projektverlauf rigoros ändern können, ist in diesem Denkschema nicht explizit vorgesehen. Dieses Lernen, das laut Nonaka erst durch gemeinsames Arbeiten, Experimentieren und Ausprobieren entsteht, wird beim traditionellen Gedankenmodell des Projektmanagements nicht implizit mitgedacht, sondern müsste erst erklärt werden (wenn es überhaupt gewollt wird).

Jeff Sutherland und Ken Schwaber erkannten bei ihren Projekten in der Softwareentwicklung, dass kleine Entwicklungsteams mit Teammitgliedern, die im Wesentlichen alle Skills haben, Software-Applikationen effektiver, schneller und mit höherer Qualität lieferten als große Teams. Basierend auf dem Artikel von Ikujiro Nonaka „The New New Product Development Game“ nannten sie diese neue Methode des Teammanagements **Scrum**. Dabei verstanden die beiden „Väter“ Scrum immer als reine Management-Rahmenmethode. Daher sagt Scrum wenig bis gar nichts darüber aus, wie Software entwickelt werden soll und kann also auch für alle anderen Management-Fragen und vor allem abseits der Softwareentwicklung verwendet werden.



Scrum – ein Paradigmenwechsel

Scrum entwickelte sich aus den Ideen des Knowledge Managements und propagiert die gleichen Vorgehensprinzipien wie das Toyota Production System (TPS):

- Kleine **selbstorganisierte und cross-funktionale Teams**, in denen die Teammitglieder verschiedene Aufgaben durchführen können.
- Der **Deming-Cycle – Plan-Do-Check-Act** – liegt als kontinuierlicher Verbesserungsprozess zugrunde.
- Der **One-Piece-Flow**: Es ist immer nur ein Teil in Arbeit.
- „**Waste**“ wird wenn immer möglich beseitigt – in Scrum nennen wir das „Impediments“.
- Die Arbeit wird von den Teams gepullt, nicht von außen reingepusht. Indem die Kontrolle über den Arbeitsprozess beim Team bleibt und das Team auch Fehler im System rückmeldet, optimiert sich der „Produktionsfluss“ von innen heraus.

Scrum denkt Projekte vom Prozess der Produktentwicklung her: Projekte sind nichts anderes als ein kontinuierlicher Produktentwicklungsprozess - also sind auch die Prinzipien der Produktentwicklung anwendbar. Dazu gehören

- die Ideen des **Design Thinking**, wie es Unternehmen wie IDEO oder Apple vormachen und
- die Prinzipien des **Lean Managements**, das den Wertschöpfungsfluss von Materialien und Produkten durch die Fertigung optimiert.

Beiden Prinzipien ist eines gemeinsam: Sie schaffen einen Raum, in dem Teams ungestört und selbstorganisiert an den besten Lösungen arbeiten können. Ikujiro Nonaka nennt diesen Raum „Ba“. Auch das Toyota Production System belässt das Tempo der Arbeitsleistung beim Team. Scrum gibt diesen Räumen den Managementrahmen und optimiert so die Abläufe der Organisation von innen heraus.

Agile Verfahren können auf diese Weise ganze Produktentwicklungszyklen managen, ganze Abteilungen und ganze Unternehmen - nicht nur in der Softwareentwicklung, sondern in der gesamten Industrie: Maschinenbau, Architektur, Medizintechnik uvm. Die größte Herausforderung dabei ist allerdings die Änderung des Mindsets von Top und Executive Management, Teams und des Einzelnen.

Herausforderungen und Handlungsfelder

Wenn Organisationen dem Ansatz „Projekte 2.0“ folgen wollen, stehen sie vor den gleichen Herausforderungen wie vor 10 Jahren: Die Organisations- und Projektmanagementansätze sind nach dem Vorbild des Wasserfallmodells ausgelegt, in Deutschland und der Schweiz ist das V-Modell stark verbreitet. Dadurch sind Organisationen entstanden, die sich in Einkauf, Verwaltung, Management, Forschung und Entwicklung völlig nach diesem stark sequenziellen Prozessmodell ausgerichtet haben. Projektmanagement 2.0 bewirkt, dass diese Bereiche stärker und kontinuierlicher zusammenarbeiten und stärker involviert sind.

Die zweite wesentliche Herausforderung: Das mittlere Management muss wieder lernen mitzuarbeiten – inhaltlich. Projektteams, in denen jeder Mitarbeiter eigenverantwortlich seine

Aufgabe erledigt, brauchen erfahrene Personen, die Mitarbeitern und Teams dabei helfen, Entscheidungen zu treffen. Sie brauchen keine Manager, die in Unkenntnis der Sachlage Entscheidungen über die Köpfe der Mitarbeiter hinweg treffen.

Innovative Prinzipien ...

- **Offenheit** ist ein Wert in Scrum, mit dem aber nicht Transparenz gemeint ist. Offenheit bedeutet, dass alle Beteiligten von sich aus über alle Aspekte, Meinungen, Ideen, Probleme und Fehler reden. Es bedeutet aber auch „Offenheit für Lösungen“. Mit Projekten 2.0 werden von Projektteams Antworten auf Fragen gegeben, bei denen weder die Frage, noch die Lösung bekannt ist.
- **Community:** Die Basis von Community ist Freiwilligkeit und die Bereitschaft sich einzubringen. Scrum-Teams beschließen jede Woche aufs Neue, woran sie arbeiten wollen.
- **Bedeutung:** In Scrum wird u.a. immer an der Präzisierung dessen gearbeitet, warum etwas getan werden soll und was für den Einzelnen als Gewinn im Projekt steckt.
- **Aktivismus:** „Doing as a Way of Thinking“ – es geht ums Tun im Sinne von Anfangen, Ausprobieren und Erfahrungen sammeln. Nicht endlos reden, sondern handeln! Dabei dürfen Fehler passieren, sie sind Grundlage des Lernens. Aber Achtung: Es ist kein Freibrief für Herumstolpern und das Wiederholen der ewig gleichen Fehler.
- **Zusammenarbeit:** Alle agilen Methoden sind teambasiert. Ein Team, das gut zusammenarbeitet, gewinnt immer gegen das einzelgängerische Genie. Selbst Einstein gewann seine Ideen aus dem engen Kontakt mit den anderen bedeutenden Physikern seiner Zeit.
- **Autonomie** bedeutet nicht Beliebigkeit, sondern die Entscheidungsfähigkeit des Einzelnen innerhalb eines gegebenen Rahmens, der jedoch immer wieder zur Disposition gestellt werden kann.
- **Glücklicher Zufall:** Ohne Glück und die zündende Idee geht auch in einem agilen Projekt nichts.
- **Dezentralisierung:** Entscheidungen werden an die Menschen delegiert, die die Arbeit machen.
- **Experimentieren:** „Fail fast“ zu geringen Kosten. Schnelles Überprüfen, iteratives und inkrementelles Entwickeln von fertigen Produktteilen. Darum geht es.
- **Geschwindigkeit** - schnell sein. Aber nicht das Gerede von der Performance von Teams ist damit gemeint. Manchmal ist stundenlanges Nachdenken wichtig.
- **Vertrauen** – in sich selbst, in das Team, und in den Kunden. Ohne Vertrauen entstehen gigantische Transaktionskosten. Schwarze Schafe gibt es immer, aber es lohnt nicht, ständig zu kontrollieren. Hier schließt sich der Kreis: Das geht nur, wenn umgekehrt eine hohe Offenheit herrscht.

... und innovative Werkzeuge

Nr.	Werkzeug	Kurzbeschreibung
1	Taskboards	Machen Sie die Arbeit Ihres Teams mit Hilfe eines Taskboards transparent und sichtbar
2	Daily Scrum	Ersetzen Sie Weekly Team Meetings durch tägliche Stand Ups vor dem Taskboard.
3	Product Backlogs / Input Cues	Eine Liste von Aufgaben und Lieferungen, die diszipliniert von oben nach unten abgearbeitet werden.
4	Retrospektiven	Reflektieren und verbessern Sie mit Ihrem Team alle zwei Wochen die Art und Weise, wie Sie miteinander arbeiten.
5	Klare Verantwortlichkeiten und Gewaltenteilung	Es gibt vier Managementverantwortlichkeiten in einem agilen Projekt: Lieferung/Qualität, Produktivitätssteigerung, den Wert der Lieferung und die Infrastruktur. Verteilen sie diese Verantwortungen auf unterschiedliche Menschen in Ihrem Team.
6	User Stories	Kundenzentriert statt anforderungsgetrieben – schreiben Sie die Funktionalitäten auf, die Sie liefern wollen. Überlassen Sie die Details den Teams
7	Cycle Time statt Aufwand	Messen Sie die Durchlaufzeit, also die Zeit vom Auftrag bis zur Lieferung. Finden Sie genug Samples und machen Sie einen statistischen Forecast, statt aufwendiger Aufwandsplanung
8	Kunde, Kunde, Kunde	Lassen Sie Ihre Teams mit den Kunden und Usern gemeinsam arbeiten. Requirements Engineering – streichen Sie diese Funktion. Bringen Sie das Team in Kontakt mit der Realität
9	Design Thinking statt Anforderungsmanagement	Der Kunde weiß im Grunde nicht, was er will – liefern Sie ihm, was er braucht. Eine Lösung für sein Problem.

Weiterführendes Material

1. Scrum

- a. Gloger, Boris: Scrum. Produkte zuverlässig und schnell entwickeln. 4., überarbeitete Auflage, München 2013.
- b. Gloger, Boris/Häusling, André: Erfolgreich mit Scrum: Einflussfaktor Personalmanagement. Finden und Binden von Mitarbeitern in agilen Unternehmen. München 2011.

2. Bedeutung

- a. Amabile, Teresa/Kramer, Steven: How Leaders Kill Meaning at Work. In: McKinsey Quarterly, January 2012.



- b. Denning, Stephen: The Springboard. How Storytelling Ignites Action in Knowledge-Era Organizations. New York 2000.
- c. Denning, Stephen: The Leader's Guide to Storytelling. Mastering the Art of Business Narrative. 2nd edition, San Francisco 2011.

3. Taskboards und Flow

- a. Anderson, David: Kanban. Successful Evolutionary Change for Your Technology Business. Sequim, Washington 2010.
- b. Liker, Jeffrey: The Toyota Way. 14 Management Principles from the World's Greatest Manufacturer. New York 2003.
- c. Liker, Jeffrey/Convis, Gary: The Toyota Way To Lean Leadership. Achieving and Sustaining Excellence Through Leadership Development. New York 2011.

4. Positive Verstärkung und agiles Führen

- a. Blanchard, Kenneth/Oncken, William/Burrows, Hal: The One Minute Manager Meets the Monkey. New York 1991.
- b. Probleme lösen statt Anforderungen managen
- c. Cagan, Marty: Inspired. How to Create Products Customers Love. Kindle Edition, Sunnyvale, California 2008.
- d. Drucker, Peter F.: Management. Tasks, Responsibilities, Practices. New York 1985.

5. User Stories

- a. Cohn, Mike: User Stories Applied. For Agile Software Development. Boston 2004.
- b. Teams, Teams und nochmal Teams
- c. Catmull, Ed: How Pixar Fosters Collective Creativity. In: Harvard Business Review, September 2008, 86(9), S. 64-72.
- d. Fisher, Kimball: Leading Self-Directed Work Teams. A Guide to Developing New Team Leadership Skills. 2nd edition, New York 2000.

Über den Autor

Die Wiesbadener Bücherei Red Adair und das Eindringlings-Detektionssystem der Enterprise haben Boris Gloger als Kind nachhaltig beeindruckt und genau genommen sein Verständnis von Scrum geprägt. Er liest in jeder freien Minute. Deswegen erschließen sich ihm in der Regel mehrere und überraschende Sichtweisen auf ein Problem – und entsprechend auch mehrere Lösungen. An Feuerwehrheld Red Adair beeindruckt ihn das überlegte, zielgenaue Vorgehen in brenzligen Situationen. Und dass ein System selbst erkennen kann, wenn etwas schief läuft – so wie es das Raumschiff Enterprise kann – ist für ihn nichts anderes als eine Metapher auf Scrum, das genau so direkt Fehler in einer Organisation aufzeigt. 2002 hat er sein erstes Scrum Team bei der österreichischen ONE (heute Orange Telecommunication Austria) zum Erfolg geführt. Nach der Ausbildung zum ersten Certified ScrumTrainer bei Ken Schwaber und Boris Gloger wesentlich dazu beigetragen, dass sich Scrum in Europa, Südafrika und Brasilien als de facto Standard der agilen Softwareentwicklung durchgesetzt hat.