

# Requirements Engineering bei agilen Methoden

Stefan Roock  
it-wps GmbH im Verbund der itelligence AG  
Friedrich-Ebert-Damm 143  
22047 Hamburg  
EMail: [stefan.roock@itelligence.de](mailto:stefan.roock@itelligence.de)

## Einleitung

Agile und leichtgewichtige Methoden wie eXtreme Programming (XP, siehe [Beck2000], [LRW2002]), SCRUM (siehe [SB2001]), Crystal Clear (siehe [Cockburn2002]) oder auch Feature Driven Development (FDD, siehe [Palmer2002]) sind heute in aller Munde. Sie versprechen reduzierte Entwicklungskosten bei hoher Qualität. Dieser Beitrag stellt einige Thesen zum Verhältnis agiler Methoden zum Requirements Engineering (RE) vor und will damit zur Diskussion einladen. Die Thesen basieren auf den Projekterfahrungen des Autors.

## Klassisches RE

Das klassische RE ist geprägt von einem phasenartigen Vorgehen, bei dem zuerst alle Anforderungen komplett ermittelt werden. Die Anforderungen werden vom Requirements Engineer analysiert und auf Basis dieser Gesamtsicht wird die Projektplanung und Aufwandsschätzung durchgeführt.

Die Ideen zyklischer Vorgehensweise sind auch in das Requirements Engineering eingeflossen, so dass das Requirements Engineering selbst zyklisch durchgeführt wird: Die Anforderungen werden erhoben und immer wieder mit den unterschiedlichen Interessensgruppen diskutiert. Es bleibt aber die Grundidee vorhanden, möglichst viele der Anforderungen zu ermitteln, bevor mit dem Programmieren begonnen wird.

## Anforderungsermittlung bei agilen Methoden

Die agilen Methoden machen es sich zunächst sehr einfach (und folgen damit dem XP-Grundsatz: „Do the simplest thing that could possibly work“, siehe [Beck2000]): Es gibt genau einen Kunden, der die Anforderungen kennt, priorisiert und fachliche Fragen zu den Anforderungen beantwortet. Dazu muss er das notwendige Wissen sowie Entscheidungskompetenzen haben. Solche Situationen kommen selten vor. Wenn sie aber vorkommen, kann die Anforderungsermittlung tatsächlich sehr einfach gestaltet werden: Der Kunde schreibt die Anforderungen auf, die Entwickler schätzen die Aufwände je Anforderung und der Kunde priorisiert die Anforderungen. Genau dieses Vorgehen findet sich beim

Planungsspiel des eXtreme Programming und sehr ähnlich bei der Sprint-Planung aus SCRUM.

So entsteht für die Entwickler eine priorisierte Anforderungsliste. Sie setzen die Anforderungen in der gegebenen Reihenfolge um und demonstrieren am Ende einer Iteration (Länge variiert je nach Methode zwischen einer Woche und drei Monaten) den erreichten Projektstand. Auf Basis des gezeigten Systemzustandes wird in der oben beschriebenen Art und Weise die Planung für die nächste Iteration vorgenommen.

Im Gegensatz zum klassischen RE sind die RE-Aufwände bei agilen Methoden also nicht am Projektanfang konzentriert, sondern über die gesamte Projektlaufzeit verteilt.

In vielen Projekten sind die Annahmen agiler Methoden zu einfach. Dann muss der Bereich Anforderungsermittlung adaptiert werden. Die folgenden Abschnitte demonstrieren solche Anpassungen exemplarisch für drei häufig auftretende Probleme.

## Mehrere Kunden

Eine offensichtliche Schwäche der Ein-Kunden-Annahme ist, dass Softwaresysteme heute in der Regel für eine große Menge von Anwendern entwickelt werden, die z.T. ganz unterschiedliche Anforderungen an das System haben. Dazu kommen Anforderungen von Nicht-Betroffenen (z.B. Betriebsrat, Datenschützer), die berücksichtigt werden müssen.

SCRUM hat für dieses Problem gleich die passende Lösung parat: Dort nimmt der Produktmanager die Aufgabe wahr, Anforderungen aus unterschiedlichen Kanälen aufzunehmen, zu konsolidieren und dann die Priorisierung vorzunehmen. Bei XP kann man genau so einen Produktmanager in die Kundenrolle einsetzen. Für das Funktionieren der agilen Methoden ist wichtig, dass es genau eine Person gibt, die priorisiert. Die Anforderungen können sehr wohl aus unterschiedlichen Quellen kommen und wenn die Entwickler Rückfragen zu Anforderungen haben, können sie diese auch direkt an die Quelle richten.

## Unbekannter Anwendungsbereich

Wenn den Entwicklern der Anwendungsbereich weitgehend unbekannt ist, berücksichtigen die agilen Methoden den Aufwand für das Verstehen nicht explizit. Wird auf die oben beschriebene Art und Weise vorgegangen, entsteht bei den

Entwicklern häufig kein konsistentes Gesamtbild des Anwendungsbereiches und des zu konstruierenden Systems. So entstehen Inkonsistenzen in der Begriffsbildung und im System. Es werden immer wieder Irrwege beschritten. Die werden zwar am Ende der Iteration in der Regel entdeckt. Allerdings werden so unnötige Aufwände verursacht. Ein unzureichendes Gesamtbild lässt sich bei XP-Projekten häufig an schwächelnden Metaphern erkennen.

Daher sollten agile Projekte immer mit einer Explorationsphase beginnen. In dieser wird der „Projektraum“ erkundet: Erste Erfahrungen in dem Anwendungsbereich werden gesammelt, essentielle Technologien werden evaluiert (vor allem mit Prototypen), erste Ideen für die Handhabung des Systems werden gewonnen, eine grobe Architekturskizze wird erstellt (Handhabung und Architektur werden bei XP durch die Metapher adressiert.). Außerdem wird in der Explorationsphase ermittelt, welche Anwender- und Interessengruppen wie eingebunden werden müssen und wie die Kunden-/Produktmanagerrolle besetzt werden muss. Nicht zuletzt kann man in der Explorationsphase abschätzen, welche Entwicklungsmethode überhaupt verwendet werden sollte.

Die Explorationsphase soll zwischen 1 und 3 Monaten dauern; ggf. müssen später weitere Explorationsphasen eingeschoben werden.

### **Anforderungen nicht explizit**

Mitunter ist den Anwendern nicht explizit klar, welches ihre Anforderungen sind oder wie sie die Anforderungen formulieren sollen. Hier kann der Produktmanager hilfreich zur Seite stehen. Natürlich kann er dazu das ganze Repertoire des RE einsetzen, wie z.B. Interviews, Szenarios, Workshops.

### **Fazit**

Auch bei agilen Methoden kommen die Techniken des klassischen RE zum Einsatz – sofern sie ein Problem lösen. Die große Änderung betrifft die Einbettung in den Entwicklungsprozess. RE wird bei agilen Methoden kontinuierlich während des ganzen Entwicklungsprozesses durchgeführt. Die Rolle des Requirements Engineer wandelt sich beim Einsatz agiler Methoden. Er wird entweder zum Berater des Kunden oder spielt direkt die Rolle des Produktmanagers.

### **Literatur**

[Beck2000] Kent Beck: „Extreme Programming Explained. Embrace Change“. Addison Wesley 2000.

[Cockburn2002] Alistair Cockburn: „Agile Software Development“. Addison Wesley. 2002.

[LRW2002] Martin Lippert, Stefan Roock, Henning Wolf: „Software entwickeln mit eXtreme Programming“. dpunkt. 2002.

[Palmer2002] Stephen R. Palmer: „A Practical Guide to the Feature-Driven Development.“ Prentice Hall 2002.

[SB2001] Ken Schwaber, Mike Beedle: „Agile Software Development with Scrum“ Prentice Hall. 2001.