

# EINE SACHLICHE ROMANZE: SCRUM UND EVOLUTIONÄRES REQUIREMENTS-ENGINEERING

Agile Praktiken und Requirements-Engineering befinden sich heute in einem scheinbaren Konflikt. Dabei verschenken beide Disziplinen wertvolle Errungenschaften des jeweils anderen Bereichs. Anders als im gleichnamigen Gedicht Erich Kästners führt diese sachliche Romanze im aktuellen Projektgeschehen zu belegbaren Erfolgen. Auch die jüngeren Initiativen aus der Scrum Alliance weisen auf diesen Weg hin. Anhand von Erfahrungen aus langjähriger agiler Arbeit zeigen wir in diesem Artikel, wie Scrum-Experten mit evolutionärem Requirements-Engineering Softwareprojekte auf effektive Weise zum Erfolg führen können.

## Die Chance erkennen

Inspiziert durch den scheinbaren Widerspruch im Titel des Kästner-Gedichtes aus dem Jahr 1928, haben wir den scheinbaren Konflikt zwischen Scrum und *Requirements-Engineering (RE)* analysiert und mit den Ergebnissen aus der erfolgreichen Projektpraxis verglichen. Gerade größere Projekte, die mit mehreren Teams Softwaresysteme inkrementell erstellen, bergen heute Risiken für die Qualität der Resultate und die erfolgreiche Adaption von Scrum gleichermaßen.

Der Ansatz von Scrum lässt – wohl nicht ganz unabsichtlich – freien Raum für Methoden des RE. Deren Resultate ermöglichen wiederum ein strukturiertes Vorgehen bei Backlog-Management und Sprint-Planung. Evolutionäres RE kann hier entscheidende Beiträge leisten, indem es Scrum bei der Initialisierung des *Product-Backlog* und beim Bearbeiten der Backlog-Einträge adäquat unterstützt.

Wie bei der sachlichen Romanze lohnt es sich, Konflikte neutral zu analysieren und als Chance für eine Weiterentwicklung zu nutzen. Dazu betrachten wir im Folgenden die Gemeinsamkeiten beider Welten in der konkreten Projektpraxis – nicht ganz unabsichtlich im Zeitraffer einer romantischen Begegnung.

## Der erste Blick

Der erste Augenkontakt zwischen Scrum und RE bleibt unverbindlich, leicht ließe sich auf Desinteresse schließen:

„[...] this includes detailed requirements analysis [...] Scrum is silent on how this work is done“ (vgl. [Sut07]).

Hier klammert Scrum das konkrete Vorgehen explizit aus oder legt dieses in die Verantwortung der Rolle des *Product Owners (PO)*. In der Folge hängen die Ergebnisse in hohem Maße von dessen persönlichen Erfahrungen und Fähigkeiten ab.

Die Offenheit von Scrum ermöglicht den Einsatz bewährter Methoden und Praktiken. Zu den vom PO geforderten Tätigkeiten zählen das Formulieren von User-Stories, das Priorisieren und Ordnen von Backlog-Einträgen, die Release-Planung und letztlich die Abnahme der Ergebnisse. Gemeinsam mit dem Team soll er die Einträge im Backlog „groomen“, d. h. formulieren, analysieren, aufteilen und verfeinern. Aus der in Scrum geforderten detaillierten Anforderungsanalyse ergibt sich, dass hierfür grundlegende RE-Kenntnisse nicht hinderlich sind.

## Gemeinsamkeiten von RE und Scrum

In Scrum lassen sich sinnvolle Entsprechungen für mehrere Aktivitäten des RE finden (vgl. auch [Hoo05]). Das Erstellen von User-Stories findet sich beispielsweise bereits im Lehrplan für den „Certified Professional for Requirements Engineering“ des International Requirements Engineering Boards (vgl. [IREB11]).

Die Projektpraxis zeigt, dass unsere Auftraggeber mehr am Erreichen ihrer Ziele interessiert sind als an den Mitteln und Wegen, anhand derer die Ergebnisse erstellt werden. Die Konzentration auf das gemeinsame Ziel hat sich daher bewährt:

„[...] zu verstehen, was der Kunde benötigt, und sicherzustellen, dass er genau dies auch erhält“ (vgl. [Hoo05]).



Thomas Mödl

(E-Mail: [Thomas.Moedl@dialogdata.de](mailto:Thomas.Moedl@dialogdata.de)),

Mitgründer der DialogData GmbH & Co. KG, entwickelt seit zwei Jahrzehnten abwechselnd Requirements und Software. Er ist Referent bei den REDays und der OOP.



Susanne Mühlbauer

(E-Mail: [Susanne.Muehlbauer@HooD-Group.com](mailto:Susanne.Muehlbauer@HooD-Group.com))

ist Senior Consultant bei der Hood GmbH. Neben Beratung im Requirements-Engineering zählen agile Ansätze zu ihren Schwerpunkten. Sie ist Referentin bei den Konferenzen REConf, Scrum Day und IEEE RE Conference.

Übertragen auf Scrum, bedeutet dies, dass das Entwicklungsteam verstehen soll, welchen Mehrwert der PO durch die neue Software erhalten möchte. Dies erfolgt in Scrum durch die Vorstellung der Backlog-Items vor dem Team durch den PO. Scrum verwendet hier explizit den allgemeinen Begriff *Backlog-Item* – in der Praxis treffen wir hingegen überwiegend auf den Einsatz von User-Stories. Der Oberbegriff *Item* schafft den Freiraum, beispielsweise eine Ablaufskizze in UML, einen Use-Case oder eine beigefügte Videoaufzeichnung zu nutzen. Das ursprüngliche Ziel bleibt so im Sinne von beiden – Scrum und RE – erreichbar, auch bei freier Wahl der Spezifikationsmethode:

„[In Sprint Planning] one focuses on understanding what the Product Owner wants [...] many people like to articulate the requirements in terms of user stories“ (vgl. [Sut07]).

Der Anteil an mündlich besprochener und schriftlich dokumentierter Anforderung ist also frei wählbar, solange die Verantwortlichen für die Umsetzung – also das

- Das Produkt-Backlog enthält verschiedene Einträge wie Kunden-Features, aber auch Forschungsaufgaben oder Ziele zur Verbesserung des Engineerings [...]
- Die Einträge im Produkt-Backlog unterscheiden sich stark hinsichtlich Größe und Aufwand.
- Dass Scrum das Schreiben detaillierter Spezifikationen verbietet, gehört ins Reich der Legenden; vielmehr entscheiden der Product Owner und das Team für jeden Eintrag neu, wieviel Detail dort erforderlich ist. [...]
- Das initiale Product Backlog Refinement [...] erstreckt sich über ein paar Tage oder eine Woche. Es beinhaltet einen Vision-Workshop sowie eine detaillierte Anforderungsanalyse [...]

**Kasten 1:** Evolutionäres RE und Scrum passen gut zusammen.

Team – verstehen, was sie zu realisieren haben (siehe Kasten 1). Die geschilderten Gemeinsamkeiten erlauben es uns, einen ersten Versuch vorzunehmen, die konkreten RE-Aktivitäten den entsprechenden RE-Aspekten in Scrum-Projekten zuzuordnen (siehe Abbildung 1 und Tabelle 1).

Wir verzichten bewusst auf eine verfeinerte Betrachtung des *Sprint Planning 1* und *2* (vgl. [Hoo05]), da bei reinem Scrum die ausreichende Qualität der selektierten Storys im Sprint-Backlog durch das fortlaufende *Backlog-Grooming* (auf Deutsch „Striegeln“) gewährleistet ist und somit vorausgesetzt wird. Dabei sollte jedoch nicht versäumt werden, auf weitere Potenziale hinzuweisen, die RE in der Praxis für beide Sprint-Planning-Meetings bereithält (z. B. Modellieren oder Systemanforderungen erheben).

Dazu unterscheiden wir zwischen der Vision, der Initialisierung des Backlogs, den fortlaufenden Tätigkeiten der Weiterentwicklung des Backlogs (Backlog-Grooming) und allen Tätigkeiten, die für die Rolle des Scrum-PO vorgesehen sind. Die Gegenüberstellung der genannten Bestandteile von Scrum mit RE-Aktivitäten zeigt, wo beide die gleichen oder unterstützende Ziele verfolgen. Parallel nennen wir die etablierten agilen Praktiken sowie dazu hilfreiche Methoden des RE.

Im Folgenden betrachten wir die in Tabelle 1 dargestellten Gemeinsamkeiten genauer. Dabei stellen wir ausgewählte RE-Methoden vor, die in Scrum angewendet werden können, ohne den agilen Grundgedanken aus den Augen zu verlieren. Die Scrum-Community hat dies bereits erfolgreich für Methoden wie Kano (vgl. [Kan84]) oder MuSCoW (vgl. [Cle04]) praktiziert.

Den grundsätzlichen Rahmen für ein Scrum-Projekt steckt die Vision ab. In der agilen Praxis werden hierfür Praktiken ein-

gesetzt, die es erlauben, das geplante Produkt in wenigen Sätzen zu skizzieren, wie (*Vision Box*, *Elevator Statement*, *Press Release*, vgl. [Coh06]). Eine sinnvolle Ergänzung für die Erstellung der Produkt-Vision sind die *Stakeholder-Analyse* und das Festlegen von Systemgrenzen und -umfang (*Scope*). *Zielmodelle* wiederum tragen durch Zieldekomposition (siehe Abbildung 2) dazu bei, die Vision eines Systems zu verfeinern. Für diese Zielmodellierung werden beispielsweise Baumdarstellungen verwendet (vgl. [IREB11]).

Im Rahmen der *Backlog-Initialisierung* wird das Product-Backlog mit ersten Anforderungen bzw. Backlog-Items befüllt. Dieser Vorgang entspricht der RE-Aktivität „Erheben“. In der agilen Praxis finden wir Methoden zur Erhebung aus dem RE wieder, wie den (*Story Finding*) *Workshop*.

Hier gibt die Vision vor, welche Anforderungen bzw. User-Stories für das Projekt relevant sind. Ergänzend hierzu grenzt ein definierter Scope den Erhebungsumfang ein. Sind die relevanten Stakeholder durch eine Stakeholder-Analyse identifiziert, können die Story-Finding-Workshops zielgerichtet erfolgen.

RE bietet vielfältige Erhebungsmethoden, die auch im Rahmen eines agilen Projekts nützlich sind, wie Rollenspiele oder *Brain Writing*. Die Auswahl der Methoden hängt jedoch stark von den projektspezifischen Gegebenheiten ab, sodass wir dies hier nicht weiter vertiefen.

Das *Spezifizieren* von Anforderungen erfolgt in agilen Projekten vorwiegend mittels User-Stories und Gesprächen. Erstere werden mit *Akzeptanzkriterien* versehen. Akzeptanzkriterien lassen sich hervorragend ermitteln, indem *W-Fragen* gestellt werden (siehe Kasten 2).

Es lohnt sich, bereits bei der initialen Erstellung eines Product-Backlogs die verwendeten Begriffe zu definieren, beispielsweise in einem Glossar oder Fachdatenmodell. Dadurch wird vermieden, denselben Sachverhalt mit unterschiedlichen Begriffen zu beschreiben. Das gemeinsame Verständnis wird dadurch von Anfang an sichergestellt.

Während der gesamten Projektlaufzeit beschäftigt sich das Scrum-Team parallel zu den Sprints mit dem Product-Backlog. Dieses *Backlog-Grooming* oder auch *Backlog Refinement* umfasst alle Tätig-

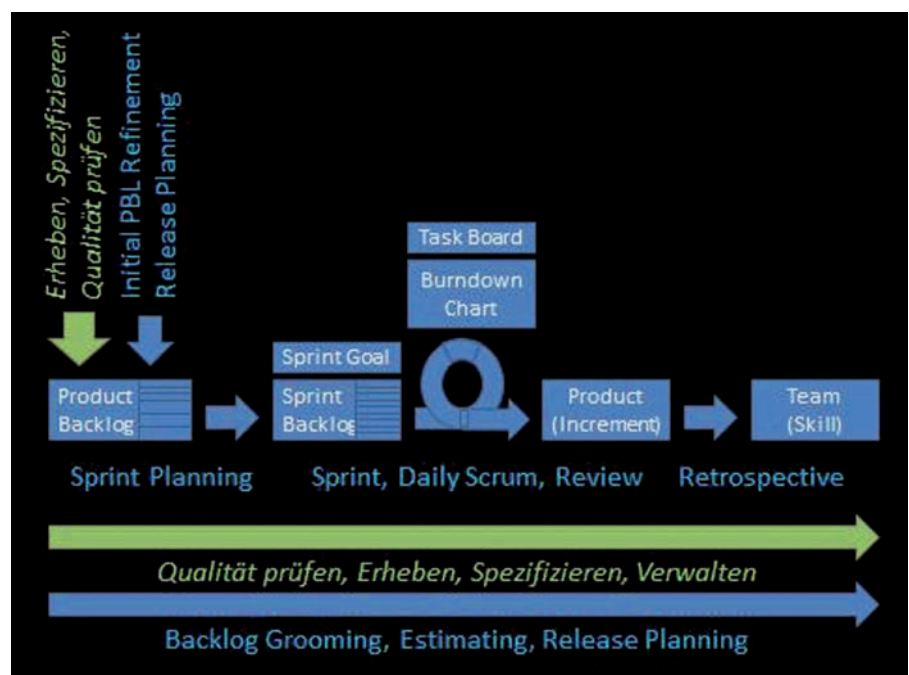


Abb. 1: RE im Scrum-Ansatz.

keiten zur Pflege und Detaillierung des Product-Backlogs für zukünftige Sprints. Aus RE-Sicht gehören hierzu das *Verwalten der Anforderungen* (inklusive Priorisierung) sowie die *Prüfung* derselben gegen definierte *Qualitätskriterien*. Für User-Stories finden hierzu die *INVEST-Kriterien* Anwendung (siehe Kasten 3).

Die Qualitätsprüfung findet implizit statt. Der PO ist verantwortlich dafür, dass zum Beginn eines Sprints eine ausreichende Anzahl von Backlog-Items vorhanden ist, die „ready“ für die Implementierung sind (vgl. [Bea09]). Die Prüfung kann er jedoch nur mit Hilfe des Teams und auf Basis der INVEST-Kriterien durchführen. PO und Team erhalten bei der Schätzung der Backlog-Items einen Hinweis darauf, ob einzelne INVEST-Kriterien verletzt sind (siehe Kasten 3). Eine Verletzung des Kriteriums *Sized Appropriate* würde so bei-

spielsweise in einer Splitting der betreffenden Story resultieren.

Die Forderung nach voneinander unabhängigen Backlog-Items (*I* für Independent) kann der PO auf fachlicher Seite selbst sicherstellen. Über eine Suche nach im Glossar definierten Begriffen, lassen sich potenziell voneinander abhängige Backlog-Items finden, da diese dieselben Sachverhalte betreffen. Ebenso lassen sich in Backlog-Items eines bestimmten Stakeholders Abhängigkeiten vermuten.

Wesentliche Aspekte des Backlog-Groomings sind die laufende und wiederholte *Schätzung* und *Priorisierung* der Backlog-Items. RE bietet diverse Methoden, die direkt oder indirekt die Schätzung unterstützen, wie zum Beispiel die Bündelung von Anforderungen oder die Analyse von Änderungen, die wir an dieser Stelle nicht weiter vertiefen. Zur Priorisierung lassen sich wiederum die Ergebnisse des

*Scopings* und der *Stakeholder-Analyse* sinnvoll einsetzen. Über den Scope lassen sich Anforderungen identifizieren, die nicht im Rahmen des aktuellen Projekts umgesetzt werden sollen. Über eine Priorisierung der Stakeholder lassen sich auch deren Anforderungen priorisieren.

Während des Backlog-Groomings werden außerdem neue User-Stories hinzugefügt (oder auch herausgenommen), detailliert oder in weitere User-Stories zerlegt (*Story Splitting*). Es findet also ein Erheben und Spezifizieren von Anforderungen statt. Das Festlegen des Projekt-Scope und der relevanten Stakeholder ermöglicht auch hier eine zielgerichtete Anforderungsdefinition. Um User-Stories besser zu verstehen oder weiter zu detaillieren, bietet sich der Einsatz von Modellierungstechniken an, wie z. B. Aktivitätsdiagramme oder Use-Case-Diagramme.

Ein ganz wesentlicher Aspekt in Scrum allgemein und während des Backlog-Groomings im Besonderen ist die Betonung der Kommunikation zwischen allen Beteiligten. Der PO ist dafür verantwortlich, dass diese Konversation stattfindet – dadurch kann der schriftliche Spezifikationsanteil reduziert werden.

Eine Reihe weiterer RE-Aktivitäten sind bei Scrum in der Rolle des PO angesiedelt. Das *Konsolidieren* und *Abstimmen* von Anforderungen durch die Stakeholder findet in Scrum keine entsprechende Aktivität. Ziel der Konsolidierung ist es, Redundanzen und Widersprüche aufzudecken und diese zu bereinigen. Für diese Aktivität greifen wir auf die identifizierten Stakeholder zurück: Stakeholder mit konkurrierenden Zielen haben häufig konkurrierende, Stakeholder mit ähnlichen Zielen wiederum ähnliche Anforderungen. Über ein Glossar findet der PO User-Stories, die sich auf die gleichen Sachverhalte beziehen, die sich gegebenenfalls widersprechen oder die zusammengefasst werden können. Bei einer zunehmenden Anzahl von Backlog-Items ist diese Aktivität – besonders also bei größeren Projekten oder bei Projekten mit mehreren Scrum-Teams – dringend geboten.

Den Aspekt des Abstimmens von Anforderungen – d. h. das Rückspiegeln der aufgenommenen Anforderungen an die Stakeholder zur Bestätigung des gemeinsamen Verständnisses – beachtet Scrum nicht explizit. Es wird davon ausgegangen, dass der PO im Idealfall selbst der Kunde ist oder in irgendeiner Form sicherstellt, dass

Scrum	RE-Aktivitäten	Etablierte Agile Praktiken	Ausgewählte RE-Methoden
Vision	Scoping	Vision Box Elevator Statement Press Release	Stakeholder Analyse Systemgrenzen/-Kontext Zielmodelle
Backlog Initialisierung	Erheben	Vision Story Finding Workshop	Priorisierte Stakeholder Scope Diverse Erhebungsmethoden
	Spezifizieren	User Story Karte Akzeptanzkriterien	Glossar/ Fachdatenmodell W-Fragen
Backlog Grooming/ Refinement	Anforderungs- qualität prüfen	Wiederholte Schätzung und Priorisierung INVEST	Suche nach Begriffen aus Glossar/ Fachdatenmodell Suche nach Stakeholder Qualitätskriterien
	Verwalten (inkl. Priorisieren)	Wiederholte Schätzung	Diverse Methoden
		Wiederholte Priorisierung	Priorisierte Stakeholder Scope
		Story Splitting	W-Fragen
	Erheben	Story Splitting	Priorisierte Stakeholder Scope
Spezifizieren	User Story Konversation	Modellierungstechniken Glossar/ Fachdatenmodell	
Product Owner Rolle	Konsolidieren	-	Suche nach Begriffen aus Glossar/ Fachdatenmodell Suche nach Stakeholder
	Anforderungen abstimmen	-	Review

Tabelle 1: RE-Entsprechungen in Scrum.

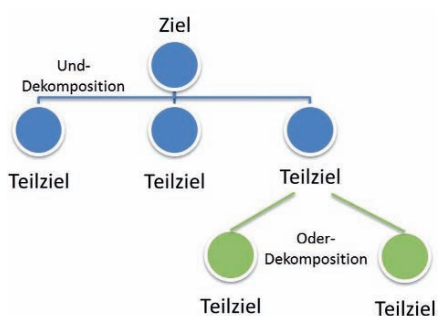


Abb. 2: Zieldekomposition.

er den Kunden bzw. die Gesamtheit der Stakeholder angemessen vertritt. Dies ist unserer Erfahrung nach in kleineren Projekten und einem wenig komplexen Umfeld der Fall, allerdings auch nur dort. Die erwähnte Rückspiegelung erfolgt bei Scrum implizit im Sprint-Review, sofern Anwender oder Stakeholder daran beteiligt sind. Wurde das Ziel verfehlt, wird dies im schlimmsten Fall erst nach dem Sprint erkannt.

### Die Chance nutzen

Mit den hier aufgeführten Entsprechungen stellt RE eine sinnvolle Unterstützung für Scrum-Projekte bereit. Demgegenüber treffen wir in der Praxis häufig auf stark vereinfachte Interpretationen von Scrum (siehe **Kasten 4**), die in Projekten mit wenigen Mitarbeitern und Teams auch funktionieren können. Wir raten jedoch davon ab, diese ungeprüft zu übernehmen.

Das Bereitstellen umsetzbarer User-Stories im Backlog ist eine Herausforderung, die sich in größeren Scrum-Projekten oft ergibt. Um die Aktivitäten des Backlog-Grooming zu unterstützen, betrachten wir abschließend die spezifischen Aspekte, die der Umgang mit Anforderungen in Scrum-Projekten mit sich bringt. Dabei erheben wir keinen Anspruch auf Vollständigkeit, nichtsdestoweniger haben sich Betrachtungsweise und Vorgehen bewährt. Diese Analyse soll auch die weitere Zusammenarbeit von Scrum und RE ermöglichen, indem sie die zu erfüllenden Aufgaben konkretisiert.

Agile Methoden versuchen im Allgemeinen, die Komplexität und die Risiken im Umgang mit Anforderungen zu reduzieren. Neben der Prüfung und Annahme der Stories durch das Team zum Sprint-Beginn und der direkten *Face-to-Face*-Kommunikation verfolgt Scrum dies durch zwei weitere Prinzipien: Die Beschränkung des Bearbeitungsumfangs auf einen Sprint von

zwei bis vier Wochen und die evolutionäre Entwicklung der Anforderungen zum spätest möglichen Zeitpunkt. Das bedeutet für die Kombination mit RE:

#### 1. Prinzip: Überschaubarer Zeitraum und Umfang

Durch Konzentration auf einen überschaubaren Umfang reduzieren sich die Aufwände, die sonst durch Pflege einer umfangreichen Anforderungsspezifikation entstehen können. Da diese gewöhnlich mit dem Bearbeitungsvolumen überproportional ansteigen, sind dadurch nennenswerte Vorteile erzielbar.

Zudem werden hier die Vorteile des Prototyping genutzt: Die Stakeholder entwickeln ihre Vorstellung besser und drücken diese konkreter aus, wenn sie dazu bereits eine in Teilen verwendbare Software vor Augen haben. Die Möglichkeiten zu Fehlinterpretationen reduzieren sich, genau wie deren negative Auswirkungen, auch wegen der zeitnahen Kontrolle durch das Review am Sprint-Ende.

#### 2. Prinzip: Evolutionäre Entwicklung von Anforderungen

Die evolutionäre Entwicklung von Anforderungen bedeutet, den Weg von der Produktvision über die Release-Planung zur Sprint-Planung wiederholt zu gehen. Die Vision wird zu Beginn einmal skizziert und dann in Iterationen schrittweise weiterentwickelt und verfeinert. So füllt sich das Product-Backlog, aus dem die Auswahl des

- Wer...?
- Wann...?
- Wie ..?
- Wie oft ?
- Wie lange...?
- Mit Welchen Daten...?
- ...

Kasten 2: W-Fragen.

anstehenden Umfangs für einen Sprint erfolgt. Der Scrum-PO ist verantwortlich für eine ausreichende Anzahl vorbereiteter Backlog-Items (Zustand „ready“) und bestimmt deren Reihenfolge im Backlog.

Das Scrum-Team soll sich in einem anstehenden Sprint auf die wichtigsten und dringlichsten Projektaufgaben konzentrieren und Überbearbeitung soll vermieden werden. Bislang empfahl Scrum dazu die Priorisierung von Backlog-Items anhand ihres Geschäftsnutzens, des dafür zu leistenden Aufwands und eventueller Risikoeinflüsse. In manchen Fällen gleicht der PO so einem Bauherren, der das Erdgeschoss seines Einfamilienhauses vor dem Keller errichtet, weil ihm das Wohnzimmer mehr Nutzen bietet als der Weinkeller. Neuere Ansätze ergänzen diese Form der Priorisierung durch eine – im Gesamtkontext des angestrebten *Return on Invest (ROI)* zu definierende – Ordnung (vgl. [Coc11]).

Evolutionäres RE wird hier zum

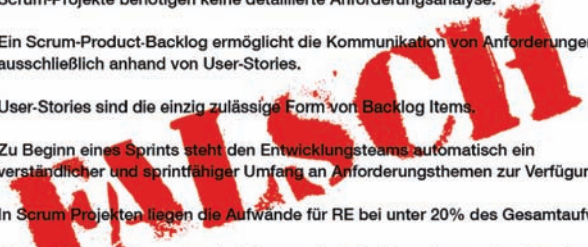
I	Independent	Unabhängig	Backlog-Items sollen voneinander unabhängig sein.
N	Negotiable	Verhandelbar	Backlog-Items stellen keine vertragliche Spezifikation dar; Inhalt und Umfang können mit dem Product-Owner noch verhandelt werden.
V	Valuable	Wertvoll	Backlog-Items sollen einen Wert liefern.
E	Estimable	Schätzbar	Die Implementierungsumfänge der Backlog-Items müssen vom Team geschätzt werden können-
S	Sized right	Richtige Größe	Jedes Backlog-Item muss in weniger als einem Sprint realisiert werden können.
T	Testable	Testbar	Jedes Backlog-Item muss nach seiner Implementierung getestet werden können.

Kasten 3: INVEST-Kriterien (nach [Wak03]).



Die aktuelle Projektpraxis basiert – vor allem bei der Neueinführung von Scrum – zu einem großen Anteil auf folgenden stark vereinfachten Annahmen, die sich nicht in der offiziellen Scrum Dokumentation finden:

1. Scrum-Projekte benötigen keine detaillierte Anforderungsanalyse.
2. Ein Scrum-Product-Backlog ermöglicht die Kommunikation von Anforderungen ausschließlich anhand von User-Stories.
3. User-Stories sind die einzig zulässige Form von Backlog Items.
4. Zu Beginn eines Sprints steht den Entwicklungsteams automatisch ein verständlicher und sprintfähiger Umfang an Anforderungsthemen zur Verfügung.
5. In Scrum Projekten liegen die Aufwände für RE bei unter 20% des Gesamtaufwandes.
6. Mit der Rolle des Scrum Product Owners sind die Kompetenzen des Requirements Engineering, des Product Management und der Projektplanung/-steuerung abgedeckt.



Kasten 4: Vereinfachte Annahmen.

Bestandteil der Projektplanung in Scrum. Der PO führt letztere nämlich implizit über die Anordnung der Backlog-Items durch: Das Backlog dient so mehr als Instrument der Projektsteuerung als zur Spezifikation von Anforderungen.

Das Backlog-Grooming verfolgt das Ziel, die in User-Stories gefassten Features evolutionär weiter aufzuteilen, zu verfeinern und zu ergänzen. Der Begriff Evolution bedeutet hier auch, dass es sich um eine gerichtete Entwicklung handelt. Das angestrebte, potenziell lieferbare Resultat eines Sprints soll stets einen Beitrag zu der gesamten mit der Systemeinführung verfolgten Zielsetzung leisten. Scrum-Projekte benötigen also eine stabile und nachhaltige Ausrichtung, die sich nicht mit jedem neuen Sprint gänzlich neu erfindet. Erfahrene Projektmanager stellen hier häufig die Frage: Wie können wir sicherstellen, dass sich nicht im nächsten Sprint genau das ändert, was wir im vorhergehenden Sprint realisiert haben?

**Deferred Decisions**

Um evolutionäres RE zu verstehen, muss man das zu Grunde liegende Konzept der *Deferred Decisions* betrachten. So führt manchmal der Begriff „evolutionär“ zu der Frage, ob übergeordnete Projektziele, wie sie sich üblicherweise im Rahmen der Formulierung einer Produktvision finden, auch zu einem späteren Zeitpunkt noch komplett neu- und umdefiniert werden dürfen. In unserem Beispiel bedeutet das, das Haus ohne Keller zu bestellen und plötzlich dann doch noch einen Keller haben zu wollen.

Die ersten, grob-granularen Festlegungen auf ein Projektziel sollten sich im Verlauf eines Projekts nicht grundlegend ändern. Vielmehr dienen sie als Leitplanken, die in eine bestimmte Richtung führen. Sind diese Leitplanken richtig gezogen, erweisen sich die daraus resultierenden Einschränkungen nicht als zufällig oder hinderlich, sondern als hilfreich und notwendig. Ähnliche

Prinzipien finden sich in vielen Bereichen unseres Lebens. Um es mit den Worten des Philosophen Karl Popper auszudrücken: „Jede Struktur erwirbt ihre Fähigkeit zu stützen durch die Aufgabe von Freiheitsgraden.“

Auf unser Beispiel übertragen bedeutet das: Im späteren Projektverlauf dürfen wir zwar die Farbe des Hauses ändern, nicht jedoch den Zweck des gesamten Projekts bzw. den Bebauungsplan. Auf dem Weg von der Vision zu den technischen Detailsentscheidungen für ein System treten neue Anforderungen und Wünsche auf, die irgendwo zwischen den genannten Extrembeispielen liegen. Häufig bestehen dann unterschiedliche Einschätzungen bei den Beteiligten, ob eine Änderung von grundlegender Art ist oder nicht.

Aber wie können bereits zum Projektstart alle grundlegenden fachlichen und technischen Fragestellungen geklärt werden? Um die richtigen Antworten als Leitplanken für die Umsetzung zu erhalten, unterscheiden wir zwischen langfristig feststehenden Anforderungen und kurzfristig veränderlichen Anforderungen. Erstere stellen die Leitplanken dar, innerhalb derer sich unser Projekt evolutionär entwickeln soll. Letztere lassen sich kurzfristig treffen und sind vergleichsweise leicht modifizierbar.

Für die Unterscheidung zwischen beiden verwenden wir das Prinzip der aufschiebbaren Entscheidungen (*Deferred Decisions*). Dies lässt sich darstellen, indem wir die kurzfristig veränderlichen und langfristig feststehenden Anforderungen nach hoher und niedriger Volatilität einordnen.

Wichtig ist hierbei die Betrachtung der Auswirkungen, die eine Änderung der jeweiligen Anforderung auf das System hätte. Dies hilft uns zu unterscheiden, ob eine Anforderung gleich zu Beginn eines Projekts feststehen muss. Damit ist nicht

Anforderung	Volatilität	
	hoch (d.h. kurzfristig veränderlich)	niedrig (d.h. langfristig feststehend)
Fachlich	aufschiebbar	nicht aufschiebbar
Technisch		
Auswirkungen	lokal	global

Tabelle 2: Veränderlichkeit von Softwareanforderungen.

ausgedrückt, dass sie nicht feststehen kann – selbst wenn sie verschiebbar wäre.

Wir halten diese Unterscheidung für wesentlich, da die Erwartungen und Meinungen über die Änderbarkeit von Software oft weit auseinander klaffen. Eine Terminologie wie in **Tabelle 2** könnte den ersten Schritt zu einem besseren Verständnis darstellen.

#### RE für Deferred Decisions

Entscheidungen, die zum richtigen Zeitpunkt getroffen werden, unterstützen die RE-Methoden der Zielfindung, -schärfung und -verfeinerung. Zieldekomposition unterstützt dabei, Ziele entsprechend ihrer Veränderlichkeit darzustellen. Dabei bilden Teilziele, die in einer Und-Beziehung zueinander stehen, tendenziell eher langfristig feststehende Anforderungen ab. Teilziele, die in einer Oder-Beziehung stehen, müssen genauer untersucht werden, da sie Lösungsalternativen beschreiben können (**siehe hierzu Abbildung 2**).

Die evolutionäre Entwicklung von Anforderungen erhält dadurch die Gerichtetheit, die einen zielorientierten Projektablauf ermöglicht. Dabei setzen wir bislang auf Unterstützung folgender Scrum-spezifischer RE-Methoden:

- Die Erfassung übergeordneter Ziele als User-Stories und evolutionäre Verfeinerung (Use-Case-Diagramme, Story-Mapping, Zielmodellierung).

- Erweiterung des Backlogs um Modelle, die eine feature- und komponentenorientierte Ausrichtung auf den Planungsebenen erlauben (z. B. Aktivitätsdiagramm, Klassendiagramm).

#### Happy End

RE-Techniken unterstützen den PO methodisch dabei, User-Stories auf handhabbare und verständliche Einheiten herunterzubrechen. Diese Einheiten dienen als Basis für die Planung inkrementeller Entwicklungs-etappen und die Organisation parallel arbeitender Teams. Redundanzen, Überlappungen

und Unvollständigkeiten in diesen Planungseinheiten können vermieden werden.

Die vereinfachenden Interpretationen, mit denen der Einsatz des Scrum-Ansatzes häufig unterlegt ist, lassen wenig Raum für den Einsatz bewährter RE-Methoden: Diese erscheinen kontraproduktiv und auf das Erstellen umfangreicher Dokumente angelegt.

RE-Beiträge können dem PO aber auch zu aussagekräftigen Backlog-Einträgen und einer in sich schlüssigen Backlog-Struktur verhelfen: Der erfolgreiche Weg führt somit über die sachliche Romanze zwischen Scrum und RE. ■

#### Literatur & Links

**[Bec11]** A. Becker, Akzeptanzkriterien im klassischen und agilen Testumfeld, in: OBJEKTspektrum Themenspecial Testing, 2011

**[Coc11]** A. Cockburn, It's Ordered – Not Prioritized!, August 03, 2011, siehe: <http://www.scrumalliance.org/articles/367-its-ordered-not-prioritized>

**[Coh06]** M. Cohn, Agile Product Management, 2006, siehe:

[http://www.mountaingoatssoftware.com/system/presentation/file/11/SDBP05\\_AgileProductMgmt.pdf?1267636389](http://www.mountaingoatssoftware.com/system/presentation/file/11/SDBP05_AgileProductMgmt.pdf?1267636389)

**[Hoo05]** C. Hood, R. Wiebel, Optimieren von Requirements Management & Engineering, Springer-Verlag 2005

**[IREB11]** IREB, Certified Professional for Requirements Engineering, Lehrplan, Elicitation and Consolidation Advanced Level, 2011, siehe:

[http://www.certified-re.de/fileadmin/IREB/Lehrplaene/CPRE\\_Elicitation\\_and\\_Consolidation\\_Lehrplan\\_Version\\_1.0.pdf/](http://www.certified-re.de/fileadmin/IREB/Lehrplaene/CPRE_Elicitation_and_Consolidation_Lehrplan_Version_1.0.pdf/)

**[Pop85]** K. Popper, K. Lorenz, Die Zukunft ist offen, Serie Piper 1985

**[Sut07]** J. Sutherland, K. Schwaber, The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, Boston Scrum, Inc., 2007

**[Wak03]** B. Wake, INVEST in Good Stories, and SMART Tasks, 2003, siehe: <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>