



Alexander Merten

(E-Mail: alexander.merten@hood-group.com) ist Diplom-Informatiker und betreut als Senior Consultant die Themengruppe „Agile“ der HOOD-Group. Als zertifizierter Scrum Master und Scrum Product Owner spricht er auf internationalen und nationalen Konferenzen über das Thema Requirements Engineering (RE) und Agilität.



Robert Stevenson

(E-Mail: robert.stevenson@hood-group.com) ist zertifizierter Product Owner und seit 2002 Senior Consultant bei der HOOD Group. Er hat sich darauf spezialisiert, Requirements Engineering (RE) in IT-Projekten zu verbessern und unterstützende Softwarewerkzeuge einzuführen. Er besitzt über 20 Jahre IT-Erfahrung in Deutschland und in Großbritannien.



Sabine Wildgruber

(E-Mail: sabine.wildgruber@hood-group.com) ist Diplom-Informatikerin und Betriebswirtin. Sie ist für Projektdurchführung, Coaching sowie Trainings zuständig und betreut das Kernthema Requirements Engineering (RE) mit den Nachbarprozessen wie Change- oder Testmanagement. Sie ist zertifizierter Product Owner.

# Agile Praktiken im Requirements Engineering – Bewährte Methoden auf neuem Terrain

Agile Vorgehensmodelle rücken den Menschen ins Zentrum der Entwicklungsprozesse. Sie stellen Kommunikation und Zusammenarbeit über Dokumentation und strikte Formalisierung dieser Zusammenarbeit [Bec01]. Dennoch ist das Ziel agiler Projekte dasselbe, wie das klassisch geprägter Projekte: profitable, stabile und kundengerechte Produkte zu erstellen, die am Markt erfolgreich ankommen. Erfolg ist die Erfüllung von Anforderungen [Hoo92]. Um diesen Erfolg zu erzielen, suchen und finden Agilisten Wege, den Kunden in engen Zyklen kontinuierlicher einzubinden und so das eigene Produkt näher an den Anforderungen des Kunden auszurichten. Dabei spielt das Erheben und Verwalten dieser Anforderungen eine zentrale Rolle. In dem agilen Vorgehensrahmenwerk Scrum beispielsweise finden sich Anforderungen in der Product Vision, dem Product Backlog bis hinab zum Sprint Backlog und den zugehörigen Tasks. Doch mit welchen Methoden können Anforderungen erhoben und verwaltet werden, ohne in den Sog schwergewichtiger Prozesse zu geraten? Dieser Artikel richtet sich an Mitglieder agiler Projekte, deren Aufgaben die Erhebung von Anforderungen einschließt. Im Folgenden stellen wir einige Best Practices vor, die zeigen, wie bekannte und erprobte Methoden des Requirements Engineering (RE) erfolgreich in einem agilen Projekt eingesetzt werden können.

## What's my product – Scoping, Stakeholder und Schnittstellen

Auch im agilen Umfeld beginnt nachhaltiges Requirements Engineering damit, den Produktumfang und die betroffenen Stakeholder „festzuzurren“: Welches Produkt will ich entwickeln, was gehört dazu und wer ist davon betroffen? Typischerweise sind diese Überlegungen Teil der Produktvision.

### Den Rahmen finden – Agiles Scoping

Im Produkt-Scoping wird festgelegt, was Teil des zu entwickelnden Produktes ist und was nicht: Was kaufe ich von außen zu und was entwickle ich selbst. Diese Entscheidungen sind maßgeblich für Architektur und viele weitere Entscheidungen im Projekt.

**Best Practices:** Modelliere den Scope grafisch, z. B. durch Kontextdiagramme! Notiere, welche Art von Schnittstelle zwischen dem

Produkt und dem, was „außen“ liegt, besteht! Bespreche den Scope mit verschiedenen Projektmitgliedern und Stakeholdern! Entwickle die Scopedefinition kontinuierlich weiter, er wird nie vollständig sein!

### Mit wem spreche ich? – Stakeholder finden und sammeln

Stakeholder sind nicht nur Endnutzer. Als Stakeholder bezeichnet man jede Person und Personengruppe, die mit dem Produkt direkt interagiert oder auf irgendeine Weise in den Produktlebenszyklus eingebunden ist. Dies erstreckt sich von Projektmanagern, über Entwickler, über das Wartungs- und Supportteam bis hin zu Interessensvertretungen und normativen Instituten. Diese zu sammeln und entsprechend ihrer Wichtigkeit in einem Projekt zu behandeln, ist eine Schlüsselaufgabe auch in agilen Entwicklungsprojekten.

**Best Practices:** Beginne mit der Definition des Scopes! Identifiziere Stakeholder, z. B.

anhand von System- und Prozessschnittstellen (Personen mit ihrer Rolle und ihren Kontaktdaten, aber auch Institutionen)! Beginne mit dem Wichtigsten und arbeite dich dann iterativ in die Tiefe! Nutze existierende Checklisten zur Ergänzung der Stakeholderliste! Bei einer großen Anzahl von Nutzern, empfiehlt sich die Identifikation von Repräsentanten bestimmter Rollen als Ansprechpartner!

**Beispiele wichtiger Fragen zur Identifikation Ihrer Produktstakeholder:**

- Wer **bezahlt** für das Produkt?
- Wer **spezifiziert** das Produkt?
- Wer **entwickelt** das Produkt?
- Wer **testet** das Produkt?
- Wer **wird** durch das Produkt **betroffen**?
- Wer **wartet** das Produkt?
- ....

Abb. 1: Checkliste für Produktstakeholder

**Mit wem spreche ich zuerst? – Stakeholder priorisieren**

Nicht jeder Stakeholder muss vollständig auf seine Anforderungen untersucht werden. Und nicht jeder Stakeholder hat für das Produkt die gleiche Bedeutung. Stakeholder zu priorisieren kann dabei helfen, die wichtigsten Stakeholder zu finden und sich auf diese zu konzentrieren.

**Best Practices:** Lege fest, welche Stakeholder Einfluss auf den Projekterfolg haben! Überlege auch, wie es um ihre Verfügbarkeit bestellt ist, um z.B. schwierig erreichbare Stakeholder früh zu erreichen! Lege fest, zu welchen Themen ein Stakeholder dir weiterhelfen kann – so kannst du die Stakeholder finden, die für die anstehenden Sprints entscheidend sind!

**Produktgrenzen kennen – Schnittstellen identifizieren**

Die Schnittstellen an der Produktgrenze, dem Scope, gliedern sich in drei Hauptgruppen:

- technische Schnittstellen,
- Benutzerschnittstellen (oft auch Human Machine Interface [HMI] genannt) und
- Prozessschnittstellen.

Je besser man diese Grenzen des Produkts kennt und die Erwartungen der Gegenseite versteht, desto sicherer kann ein Produkt entwickelt werden.

**Best Practices:** Stelle regelmäßig sicher, dass Stakeholder das gleiche Verständnis der jeweiligen Schnittstellen haben! Diskutiere, was von welcher Seite geliefert werden soll! Beziehe Personen mit dem richtigen Fachwissen ein: Architektur und Entwicklung bei technischen Schnittstellen, Grafik und Graphical User Interface [GUI] bei Benutzerschnittstellen und Business Analyse bei Prozessschnittstellen! Nutze Modelle und Grafiken, um das gemeinsame Verständnis festzuhalten!

**Den Kreis schließen – Schnittstellen, Stakeholder und Scope**

Scoping, Stakeholderanalyse und Schnittstellendefinition sind also auch in agilen Projekten ein wichtiger Aufsattpunkt für die Produktentwicklung. Doch keine der drei Aktivitäten ist jemals abgeschlossen. So gehört es zum Beispiel zu den Aufgaben eines guten Scrum Product Owners, diese drei Aktivitäten vor jedem Sprint durchzuführen und Synergieeffekte zu nutzen.

Best Practices: Nutze Fragetechniken, um bei der Befragung von Stakeholdern weitere Stakeholder und Schnittstellen zu ermitteln! Besprich mit den Stakeholdern den Scope und verfeinere kontinuierlich! Suche zu einer Schnittstelle explizit nach geeigneten Stakeholdern, wenn sie fehlen! Ergänze dein Verständnis über die Umwelt deines Produktes!

**Die Qual der Wahl – Erhebungstechniken**

Sind die richtigen und wichtigen Stakeholder gefunden, ist die Anforderungserhebung der nächste Schritt im agilen Requirements Engineering. Unter Erhebung verstehen wir das Herausfinden oder Ermitteln von Anforderungen an das Produkt. Dazu existiert eine Vielzahl von Vorgehensweisen, die alle ihre Vor- und Nachteile haben [Wie05]. Im Weiteren wollen wir an vier Beispielen zeigen, wie RE-Methoden genutzt werden können, um effektiv Anforderungen zu finden und zu detaillieren.

**Best Practices:** Wähle bewusst für jede Situation eine Erhebungstechnik aus; Wäge Vor- und Nachteile ab! Reicht eine Technik nicht aus, ergänze sie ggf. mit weiteren Techniken!

**User Stories**

In agilen Projekten gelten User Stories als verbredete Darstellungsform zur Dokumentation von Anforderungen. Die Nutzung von User Stories zum Erheben von Anforderungen ist eine weitere Nutzungsform. Hierbei wird das systematische Suchen von Rollen und Zielen, die ein Stakeholder mit einer User Story verfolgt, genutzt, um Anforderungen zu erheben.

**Best Practices:** Nutze standardisierte Vorlagen bei User Stories! Frage bei funktionalen Stories die nicht-funktionalen Aspekte ab und definiere sie als Abnahmekriterien!

**Anforderungserhebungsworkshops**

Der Anforderungserhebungsworkshop ist die Erhebungstechnik, die in den meisten agilen Projekten zur Anwendung kommt. Es ist eine strukturierte Besprechung, in der ausgewählte Gruppen von Betroffenen und Experten zusammenarbeiten, um gemeinsam Anforderungen zu erheben und ggf. weitere Arbeitsergebnisse zu erzeugen, die zusätzliche Informationen enthalten. In agilen Projekten kann das Produkt solcher Anforderungserhebungsworkshops beispielsweise eine Menge an User Stories sein.

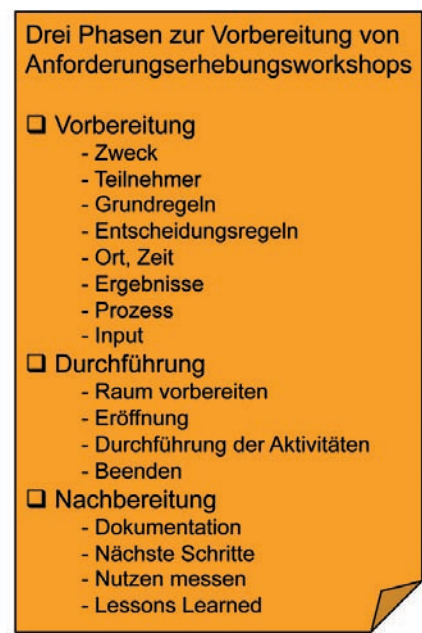


Abb. 2: Checkliste für Workshops [Val]

Dabei kommt der Vorbereitung des Workshops und der zielgerichteten Durchführung eine hohe Bedeutung zu. **Abbildung 2** zeigt eine Checkliste zur Sicherstellung der Effizienz eines solchen Anforderungserhebungsworkshops.

**Best Practices:** Setze Workshops ein, um User Stories zu erheben.

**Checklisten**

Checklisten bestehen aus Fragen, die z.B. den Stakeholder dazu inspirieren sollen, bisher nicht betrachtete Aspekte der/des zu entwickelnden Software/Systems auf Anforderungen zu durchleuchten. Diese Technik kann bei der Erhebung von User Stories einerseits dazu verwendet werden, um neue User Stories zu finden, andererseits ist sie anwendbar, um wichtige Details, wie zum Beispiel Akzeptanzkriterien von User Stories aufdecken zu können. Besonders gut eignen sich Checklisten, wenn bereits Erfahrung mit der Anwendungsdomäne oder mit der Art der/des zu entwickelnden Software/Systems vorhanden ist.

**Best Practices:** Nutze Checklisten, um deine Fragetechnik und Effizienz zu steigern! Nutze sie, um immer wiederkehrende Aspekte und Fragen nicht zu vergessen! Überarbeite sie regelmäßig und tausche dich mit anderen Projekten aus!

**Metafragen**

Metafragen ergänzen das Vorgehen in Workshops. Sie dienen dazu, eine angewandte Fragetechnik zu verbessern und deren Erfolg zu prüfen. Im Kontext der Erhebung von User Stories helfen Metafragen dabei, neue User-Rollen zu identifizieren und sicherzustellen, dass der Workshop möglichst effizient zu den richtigen User Stories führt. Beispiele hierfür sind:

1. Glauben Sie, dass ich die richtigen Fragen stelle? Frage ich genug?
2. Sind Sie die richtige Person für diese Fragen? Frage ich zu viel? Wen könnte ich noch Fragen?
- ...

**Best Practices:** Nutze Metafragen, um kontinuierlich die Effizienz von Workshops zu steigern.

**635-Brainwriting**

Beim 635-Brainwriting suchen 6 Teilnehmer jeweils 3 User Stories, für die sie jeweils 5 Minuten Zeit haben. Sobald die ersten 5 Minuten abgelaufen sind, gibt jeder Teilnehmer seine User Stories an den nächsten Teilnehmer weiter. In den nächsten 5 Minuten werden die User Stories vom jeweils nächsten Teilnehmer weiterentwickelt (z. B. Präzisierung, Detailtiefe oder Formulierung). Dies wird solange wiederholt, bis jeder Teilnehmer seine ursprünglichen User Stories wieder zurück erhalten hat. Dieses Verfahren eignet sich, um in kurzer Zeit an viele neue und bereits qualitativ hochwertige User Stories zu gelangen.

**Best Practices:** Steuere die Auswahl der initialen User Stories durch ein vorhergehendes Brainstorming und wähle gezielt Themen aus! Priorisiere das Entstandene im Nachhinein!

**Zusammenfassung**

Wir konnten hier eine exemplarische Auswahl an Erhebungstechniken aus dem RE betrachten, die sich gut dazu eignen, um das Erheben von User Stories effizienter zu gestalten. Weitere geeignete Erhebungstechniken wären z. B. Befragung bzw. Interviews, Beobachtung, Szenarien, Brainstorming, Mindmaps, Use-Cases.

**Was will der Kunde wirklich? – Unterscheiden von Problem und Lösung**

**Die Frage nach dem „Warum“ – Bedürfnisse suchen und finden**  
 „Als Benutzer will ich ein Register-Tab aus der Seitenleiste heraustrennen, sodass ich festlegen kann, welche Tabs angezeigt werden, damit ich sie beliebig auf meinem Bildschirm anordnen kann.“ Das ist es, was der Nutzer möchte. Aber ist es auch genau das, was er braucht?

Im Requirements Engineering ist es ein wichtiger Grundsatz, darauf zu achten, eine Problemstellung von ihrer Lösung bewusst zu trennen. In dem oben dargestellten Beispiel erfahren wir zwar, was der Nutzer will, aber nicht warum. Sein eigentliches Problem bleibt uns unbekannt. Wahrscheinlich gibt es aber eine Reihe von Situationen, in denen der Anwender mehrere Informationen parallel anzeigen können möchte und daher diese Anforderung formuliert.

Diese Informationen, die für den Kunden den eigentlichen Geschäftswert darstellen, sind jedoch verloren gegangen. Wenn dem Projektteam die Problemstellung bekannt ist, hat es die Möglichkeit, alternative Lösungen zu der Problemstellung vorzuschlagen.

**Best Practices:** Frage so oft wie möglich „Warum?“. Es ist dein Mantra bei der Anforderungserhebung!

Unterstütze deine Stakeholder dabei, das Problem zu beschreiben und sich bei der technischen Umsetzung vom Team beraten zu lassen!

**Gezielte Trennung – Ein Ansatz mit User Stories und Use Cases**

Im Folgenden skizzieren wir einen Ansatz, der es uns in agilen Projekten ermöglicht, mit einfachen Mitteln Problem und Lösung gezielt voneinander zu trennen (Abb. 3).

User Stories und Use-Cases sind Darstellungsformen, die in agilen Projekten weit verbreitet sind und den Vorteil haben, im Gegensatz zu vielen grafischen Modellen oder formalen Darstellungsformen, für alle Stakeholder leicht verständlich zu sein. Daher eignen sie sich bestens, um die Kommunikation zwischen Projektmitgliedern und Stakeholdern zu fördern.

Die typische Form von User Stories mit 1 bis 3 Sätzen bietet sich geradezu an, um sie zur Repräsentation von Problemstellungen zu nutzen. Use-Cases mit ihren Ablaufbeschreibungen eignen sich dagegen sehr gut, um die Umsetzung solcher Problemstellungen zu beschreiben. Zudem können Use-Cases in zweiter Instanz dazu genutzt werden, die Benutzerdokumentation des Produkts zu pflegen. Dabei ist es nicht notwendig, Use-Case-Beschreibungen bis ins letzte Detail zu spezifizieren. Abhängig von Komplexität, Priorisierung und Erfahrung des Entwicklungsteams, genügt es, den positiven Ablauf darzustellen, sodass die Beteiligten sehen, wie die Umsetzung funktioniert bzw. funktionieren wird.

Die unterschiedliche Repräsentation von Problem- und Lösungsinformation führt dazu, dass sowohl dem Projektteam als auch den Stakeholdern stets klar ist, ob gerade über das Problem oder über die Lösung gesprochen wird. Zudem lässt sich prüfen, ob für gefundene Use-Cases (Lösungsansätze) klar ist, welche User Stories (Problemstellungen) durch sie erfüllt werden sollen. Das zu Anfang gezeigte Abhandenkommen der Problemstellung wäre auf diese Weise nicht entstanden und das Projekt gewinnt zugleich ein Mittel zur späteren Rückverfolgbarkeit zwischen Implementierung und Anforderungen.

**Best Practices:** Nutze Use-Cases gezielt um die Funktionsweise des Produktes, und damit die gewählte Lösung darzustellen! Achte darauf, dass User Stories das Problem beschreiben!

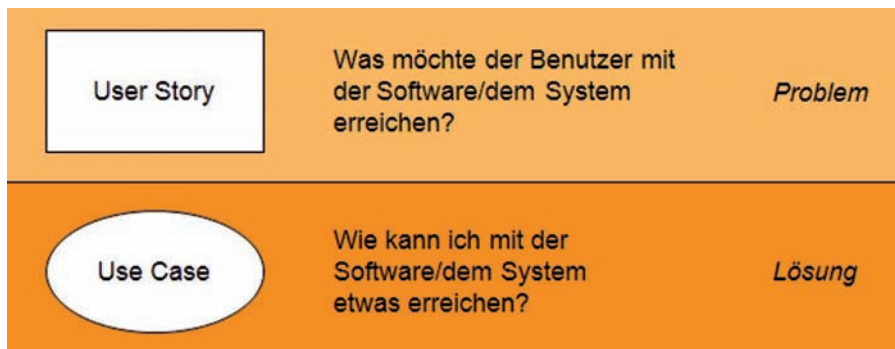


Abb. 3: Trennen von Problem und Lösung

**Alles in einem Schritt – Testfälle und Anforderungen gemeinsam erzeugen**

Nachdem wir nun die Anforderungen aus unseren Stakeholdern herausholen konnten und Problem und Lösung getrennt sind, stellt sich die Frage nach der Abnahme und dem Testen.

**Zwei Fliegen mit einer Klappe – Akzeptanzkriterien erheben**

Zu Beginn des Projektes sowie zu Beginn jeder Iteration werden neue Anforderungen besonders intensiv erhoben und bestehende Anforderungen angepasst. Dies stellt hohe Erwartungen an den Anforderungsverantwortlichen, z.B. den Product Owner in Scrum. Agile wie auch klassische Vorgehensmodelle fordern zusätzlich, dass Akzeptanzkriterien für jede Anforderung direkt bei der Erhebung definiert werden. Diese können dann wiederum als Testgegenstand genutzt werden, um Testfälle zu erzeugen.

**Best Practices:** Zeige den Stakeholdern die Vorteile, wenn Erwartungen in Form von Akzeptanzkriterien definiert werden! Nutze die Best Practices zur Anforderungserhebung auch für die Erhebung von Akzeptanzkriterien! Insbesondere sollten Akzeptanzkriterien messbar sein; beispielsweise sollten Toleranzgrenzen gemeinsam definiert werden!

**Plan to test – Testfallstrategien frühzeitig definieren**

Akzeptanzkriterien sind nichts anderes als die Zielsetzung der Testfalldefinitionen. **Abbildung 4** verdeutlicht den Zusammenhang.

Wird also direkt bei der Definition der Anforderungen auch die Definition der Akzeptanzkriterien beendet, so werden zwei wichtige Ziele in einem Schritt zu Beginn der Iteration erreicht. Darüber hinaus müssen wir uns auch bereits zum Start eines Projektes die Frage nach möglichen Teststrategien im Gesamtteam stellen. So können wir das Risiko reduzieren, von sehr hohem Realisierungsaufwand für Testumgebungen überrascht zu werden. Immerhin brauchen einige Testumgebungen genauso lange wie das System selbst, um entwickelt zu werden.

**Best Practices:** Definiere frühzeitig die Teststrategie und baue entsprechende Testumgebungen auf! Beginne mit dem Beschreiben der

**Bereiten Sie sich von Beginn an auf ihre erfolgreiche Projektabnahme vor!**

**Backlog-Liste**

Backlog No	Title	User Story	Acceptance Criteria	...
PB_0724	Discount	User Story	There is a fixed list of discounts available	...
PB_0956	Approval	User Story	Acceptance Criteria	...
PB_0035	Title	User Story	Acceptance Criteria	...
PB_0144	Title	User Story	Acceptance Criteria	...

**1**

**Definiere Abnahme Kriterien**

**Testfall-Liste**

Testfall No	Title	Description	Ref to Backlog	...
T_1020	Discount	In the fixed list there are the following defined discounts available: 5, 10, ...	PB_0724	...
T_1045	Approval	Test Case description	PB_0956	...

**2**

**Erstelle grobe Liste der Testfälle**

*Abb. 4: Von der Anforderung zum Testfall*

notwendigen Testfälle so früh als möglich – die Stabilität der Anforderungen während einer Iteration mindert das Risiko von Fehlerarbeit! Bei Scrum: Nimm ggf. die Erstellung der notwendigen Tests in die Definition of Done (DOD) auf!

**Den Fortschritt spüren – Spaß mit agilem Requirements Engineering**

Einer der zentralen Aspekte agiler Vorgehensmodelle ist das iterative Vorgehen. Durch kleinere und kürzere Schritte wird der Projektfortschritt in den Vordergrund gerückt und schnell für das Team sichtbar. Dies bringt nicht nur mehr Spaß an der Arbeit, sondern ermöglicht auch eine Reihe von weiteren Vorteilen.

**Schnelle Ergebnisse – präziseres Feedback**

Auch in Zeiten agiler Entwicklung fällt es vielen Menschen leichter, zu beschreiben, was sie „eigentlich“ wollen, wenn sie sehen, was sie bekommen werden. Sie brauchen eine Basis zur Beurteilung, um ihre Anforderungen genauer zu fassen. Wird also ein funktionierendes Produkt-Inkrement schnell geliefert, fördert dies die Qualität der Anforderungserhebung frühzeitig und wird direkt in kommenden Iterationen belohnt.

**Best Practices:** Notiere Kommentare aus den regelmäßigen Kundenabnahmen von Iterationsergebnissen als mögliche, neue Anfor-

derungen! Versuche zu verstehen, was den Kunden bewegt und gib ihm eine konkrete Ausgangsbasis in Form eines Produkt-Inkrement dafür!

**Teamegeist – Anforderungen besser verstehen**

Sobald sich die Kommunikationswege der Entwicklungsbeteiligten verkürzen, baut jede Person dieses Teams schneller ein Verständnis über die vorliegenden Anforderungen auf. Wissen kann effizienter geteilt werden. Alle kommen näher zusammen und fühlen sich bei der Umsetzung von Anforderungen oder bei der frühzeitigen Klärung von Konflikten involviert. Tägliche Kurzabstimmungen und gemeinsame Klärung über Fragen und Missverständnisse zu Anforderungen lassen auch Stakeholder, einen regelmäßigen Fortschritt erkennen. Das Vertrauen zwischen Entwicklern und Stakeholdern wächst. So können Stakeholder und Team leichter auf gleichem Kurs bleiben.

**Best Practices:** Stakeholder-Vertreter, wie z.B. der Scrum Product Owner, sollten täglich Zeit für das Team einplanen, um Fragen zu beantworten! Steht ein Stakeholder nicht direkt für Fragen zur Verfügung, Sorge dafür, dass die Fragen an ihn nicht verloren gehen! Suche nach Wegen, ihn einzubinden! Nicht zur Verfügung stehende Experten können schnell zum Projektrisiko werden. Handle frühzeitig und suche gegebenen

### Soziale Aspekte – Gefahren agiler Vorgehensmodelle beherrschen

Trotz aller Vorteile bringen agile Vorgehensmodelle viele kleine und große Veränderungen in Bereichen des sozialen Miteinanders mit sich. Werden diese sozialen Faktoren nicht erkannt und beachtet, können sie einen negativen Einfluss auf Teile des Teams haben. Sowohl die gesteigerte Transparenz als auch die starke Forderung und Förderung der engen Zusammenarbeit können ungewohnt und anstrengend sein.

Wichtige Gründe für negativen Einfluss sind zum einen der wahrgenommene Anstieg des Zeitdrucks aufgrund kurzer (manchmal täglicher) Fertigstellungstermine und zum anderen der Anstieg der Arbeitsmenge in eng zusammenarbeitenden Teams. Schon die Tatsache, dass beispielsweise im Scrum ein Sprint „kurz“ ist, kann zu Stressanstieg bei Teammitgliedern führen. Geknüpft an die Tatsache, dass nun auch noch plötzlich häufigere und engere Interaktionen mit den Kollegen erforderlich sind, kann dies für manches Teammitglied einen Alptraum bedeuten. Ein erfahrener Projektleiter ist daher umso mehr gefragt, diese Probleme zu erkennen und Lösungen zu suchen. In Scrum beispielsweise ist dies die Hauptaufgabe des Scrum Masters.

**Best Practices:** Akzeptiere „Servant Leadership“ als ein wichtiges Merkmal agiler Vorgehensmodelle! Mache Dich mit Team-Building-Methoden vertraut! Achte auf die Auswirkungen von Veränderungen bei verschiedenen Persönlichkeitstypen!

### Erholung – Spaß am Erfolg der agilen Arbeit genießen

Ein letzter Tipp zum Abschluss: Plane, dich zu erholen und genieße die vielen kleinen Erfolge die du mit agilen Vorgehensweisen erlebst! Du hast bei der Arbeit im agilen Entwicklungsumfeld viel Energie gebraucht, um den Teamerfolg sicherzustellen. Vergiss nicht, dir diesen Erfolg auch bewusst zu machen. Belohne dich dafür und widme beispielsweise den letzten Teil des Arbeitstages oder der Woche sozialen Aspekten. Gehe zum Beispiel nach draußen, um kurzes Luft zu schnappen oder um gemeinsam mit Teamkollegen etwas trinken und essen zu gehen. Vielleicht genießt du deinen Arbeitsfortschritt auch einfach durch bewusstes Ablenken von Projektthemen (Cooling Down). Dieses bewusste Erholen vom effizienten Arbeiten hast du dir nicht nur verdient, sondern du wirst es manchmal auch brauchen. Viel Erfolg!

### Fazit

Mit einem gutem Requirements Engineering (RE) steht und fällt ein Projekt – auch

in Zeiten agiler Vorgehensweisen. Häufig werden jedoch die Nachteile suboptimalen REs unterschätzt. Nicht zuletzt, da sich agile Methoden zumeist stark auf die Entwicklungsabläufe fokussieren und wenig Aussagen über ein angemessenes RE treffen. Wird ein agiles Vorgehen jedoch mit entsprechenden RE-Methoden angereichert, kann dies die Erfolgsaussichten des Projekts signifikant erhöhen. Einige Möglichkeiten wie dies geschehen kann, wurden hier vorgestellt. ■

### Referenzen

[Bec01] K. Beck et al. (2001)

[www.agilemanifesto.org](http://www.agilemanifesto.org)

[Hoo92] Colin Hood 1992: „Erfolg ist Erfüllung von Anforderungen“

[HuW05] C. Hood, R. Wiebel. 2005. Optimieren von Requirements Management & Engineering, Springer, Berlin, Kap. 5.5 „Definieren des Scopes und identifizieren der Stakeholder“, S. 83 – 85

[Wie05] Simon Wiedemann, PhD, Erheben von Anforderungen – Ein Leitfaden der HOOD Group, März 2005

[Val] Uwe Valentini, HOOD-Group, Erfolgreiche Anforderungserhebung mit Workshops 1. Auflage