



**Sven Hubert**

(sven.hubert@aitgmbh.de)

ist Bereichsleiter und Software Process Consultant im AIT TeamSystemPro-Team und MVP für Visual Studio ALM. Er hilft Teams bei der Verbesserung der Softwareentwicklung und berät bei der Einführung und Anpassung des Visual Studio Team Foundation Server. Seine Erfahrungen vermittelt er zudem als Autor des TFS-Blogs, in Magazinen und in Vorträgen.



**Stefan Rauch**

(stefan.rauch@boschrexroth.de)

ist bei Bosch Rexroth in einem Entwicklungsbereich für die Einführung von Softwareentwicklungsprozessen verantwortlich. Er nutzt den Team Foundation Server als Grundlage für die Umsetzung agiler Verfahren und einer durchgängigen Toolkette.

## Werkzeuge für Product Owner bei Bosch Rexroth

Wer als Produktmanager arbeitet oder die Rolle des Product Owner innehat, verwendet wahrscheinlich Microsoft Office Word, um Anforderungen aufzunehmen und zu dokumentieren, Excel, um Abschätzungen zu Kosten und Aufwand festzuhalten und Prioritäten zu vergeben, sowie ein Ticket-System oder Bug-Tracking-Tool, um mit Entwicklung und Qualitätssicherung zu kommunizieren. Leider haben diese Systeme keine gemeinsame Datenbasis: neben der Tatsache, dass die Daten mehrfach angelegt und gepflegt werden müssen, bedeutet jeder Übergang einen Informationsverlust. Einige Informationen stehen nur in Word, einige sind nur in Excel, andere sind nur im Ticket-System zu finden. Von den UML-Modellen, die mit Visio, Enterprise Architect oder ähnlichen Werkzeugen erstellt wurden, ganz zu schweigen. Der Product Owner zerreißt sich also zwischen diesen verschiedenen Tools der unterschiedlichen Ansprechpartner. Dieser Artikel zeigt, wie eine integrierte Werkzeugkette mit gemeinsamer, zentraler Datenbasis für die bekannten Tools aussehen kann.

### Menschliche und andere Herausforderungen

Die größte Herausforderung in Entwicklungsprojekten ist die Kommunikation mit dem Kunden oder involvierten Fachabteilungen.

Wenn Missverständnisse zwischen Auftraggeber und Auftragnehmer erst spät im Projekt zutage treten, wird die Korrektur eventuell kostspielig – je später, desto teurer. Aktuelle Projektdaten und passende Sichten auf diese sind somit eine wichtige Voraussetzung für die Durchführung von Softwareprojekten. Beides hilft, Missverständnisse und Lücken in Annahmen, Spezifikationen und Planungen früh zu erkennen und zu beseitigen. Transparenz in der Planung schafft Vertrauen – Vertrauen zwischen Auftraggeber und Auftragnehmer, aber auch zwischen Entwicklung und Management.

Passende Prozesse sind hierfür eine notwendige Voraussetzung. Ein entscheidendes Erfolgskriterium beim Etablieren dieser Prozesse sind die passenden Tools. So nutzt Bosch Rexroth den Microsoft Team Foundation Server (TFS), um die eigene

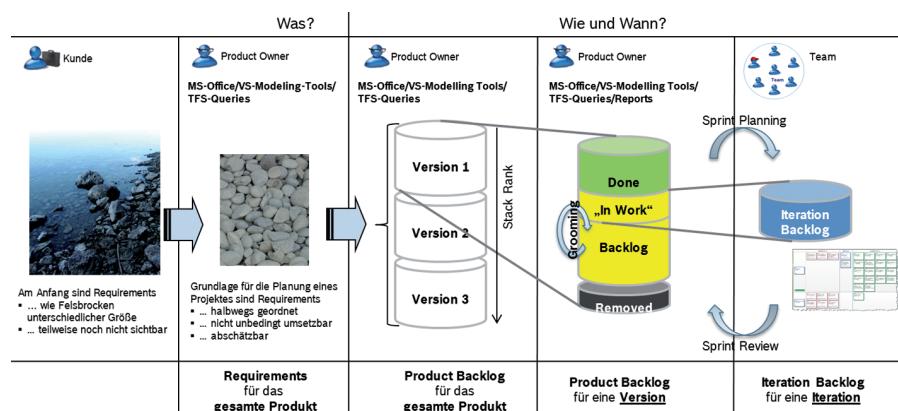
Methodenkompetenz mit einer durchgängigen Datenbasis zu verknüpfen.

**Abbildung 1** zeigt, wie aus einer Kundenidee ein Eintrag im Backlog eines Entwicklungsteams wird. Bindeglied zwischen Kunde und der Entwicklung ist der Product Owner (auch Produktmanager genannt). Er hat die Aufgabe, seine Kunden von den entwickelten Produkten zu begeistern. Indem er die Vorgaben aufarbeitet, versetzt er das Entwicklungsteam in die Lage, die

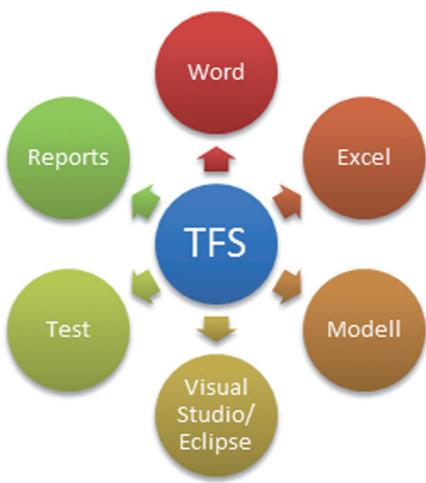
Wünsche seiner Kunden in Produktinkremente zu verwandeln.

Im Wesentlichen muss er dazu:

- Die zunächst häufig unstrukturierten Informationen des Kunden als Anforderungen aufnehmen, konkretisieren und strukturieren.
- Lücken in den Anforderungen identifizieren und schließen.



**Abb. 1:** Informationspipeline – von der Idee bis zum Iterationsbacklog



**Abb. 2:** TFS als Rückgrat für alle Product Owner-Werkzeuge

- Die Anforderungen priorisieren und in eine Abarbeitungsreihenfolge bringen.
- Fragen des Teams beantworten und Entscheidungen zu kurzfristigen Änderungen treffen.
- Funktionale Lösungskonzepte mit dem Team erarbeiten.
- Für eine passende Detaillierung sorgen.
- Akzeptanzkriterien für die formulierten Anforderungen definieren.

Im Rahmen seiner Tätigkeiten erzeugt der Product Owner eine Menge Informationen, die er irgendwo verwalten muss.

### Aus Word in die Datenbank

Für die Verwaltung der Anforderungen wurde bei Bosch Rexroth nach einer Möglichkeit gesucht, neben den gängigen Bearbeitungsmöglichkeiten über Excel oder ein Webportal, die während des Projekts entstehenden Artefakte wie Requirements, Use Cases und Testfälle untereinander zu verknüpfen (vgl. Abbildung 2).

Zu Beginn eines Entwicklungsvorhabens sieht sich ein Product Owner je nach Projektart mit sehr unterschiedlichen Voraussetzungen konfrontiert (vgl. Abbildung 1, Spalte 1). Während für die Weiterentwicklung von bereits existierenden Produkten die Kunden oft schon sehr strukturierte und konkrete Anforderungen an die Entwicklung übergeben, ist das Bild bei neuen Produkten in der Regel deutlich unschärfer.

Egal mit welcher Form von Kundenwünschen es der Product Owner zu tun hat, seine Aufgabe ist die Dokumentation und das Strukturieren der vorhandenen

Informationen. Als geeignetes Werkzeug hierfür hat sich Word etabliert. Es gibt zunächst keine Struktur für Aufzeichnungen vor und gewährt somit den erforderlichen Freiraum.

Im Rahmen der Anforderungserhebung hat es sich allerdings bewährt, gewisse Vorlagen für aufzunehmende Punkte und Strukturen vorzugeben. Im schlechtesten Fall wird der Freiraum von Word für unstrukturierte Prosa „missbraucht“. Word ist durch seine Verbreitung aber eine gute Grundlage für die Kommunikation und aktive Mitarbeit aller Projektbeteiligten.

In Word vorstrukturierte Kundewünsche synchronisiert der Product Owner mit dem von AIT entwickelten, kostenlosen Word-Plug-in „WordToTFS“ bidirektional mit der Datenbank des TFS. Diese ist das zentrale Repository für alle Projektdateien und beinhaltet die „einzige Wahrheit“ im Projekt.

Die sogenannten „Work Items“ (Überbegriff für alle Requirements, Workpackages, Use Cases usw.) werden in Word in Form von Tabellen dargestellt, deren Format und Layout pro Projekt frei definiert werden kann. Somit ist eine Anpassung an unternehmensinterne Vorgaben und Prozesse ohne Weiteres möglich. Dokumententemplates führen den Product Owner durch die Anforderungsanalyse und dienen ihm als Checkliste, um alle wesentlichen Informationen zu den Anforderungen einzuholen.

Zur Beschreibung der Work Items ohne Weiteres können in Word auch Bilder und andere Elemente verwendet werden. Dies

versetzt den Product Owner in die Lage, zur Beschreibung von Anforderungen neben Text auch mit Skizzen und Kennfeldern zu arbeiten, was besonders im technischen Umfeld sehr wichtig ist

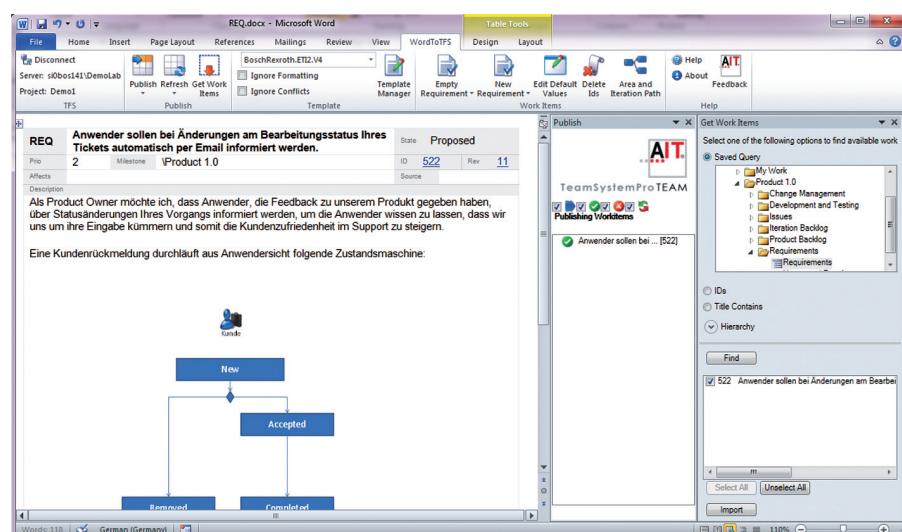
Ein Beispiel bietet Abbildung 3. Im linken Bereich ist das Dokument mit Grafiken zu sehen. Die Anforderung ist als Tabelle im Word-Dokument hinterlegt und wird mit dem TFS synchronisiert (rechts im Bild). Eine allgemeine Übersicht über andere Werkzeuge zur Verwaltung von Anforderungen und Aufgaben mit TFS 2012 finden Sie unter [SvH12].

### Lebendige Modelle

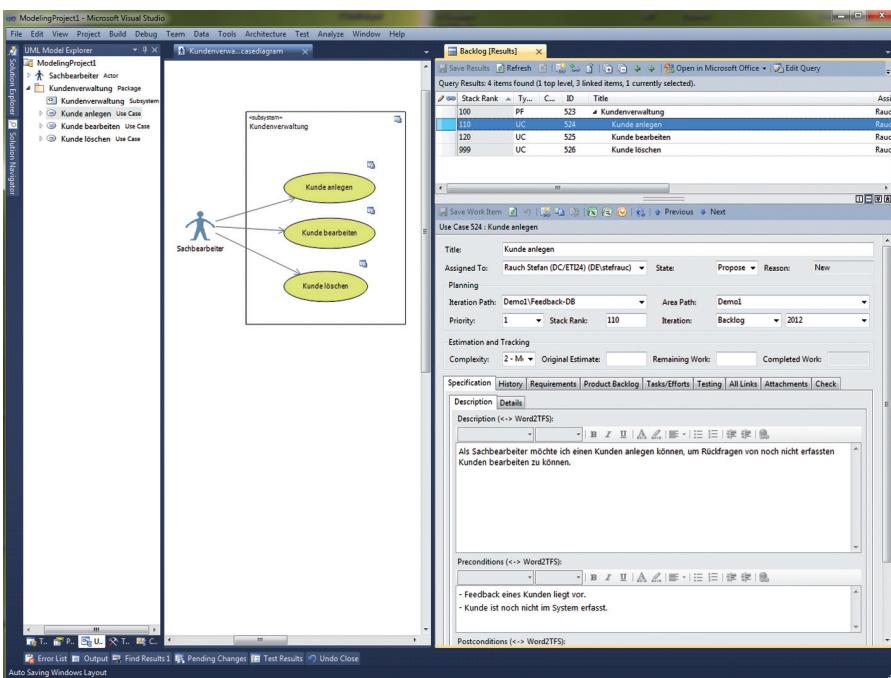
Um die im ersten Schritt vom Kunden aufgenommenen Informationen in einen Product Backlog mit planbaren Arbeitsaufträgen für die Entwicklung zu transformieren, baut der Product Owner ein Anforderungsmodell auf (vgl. Abbildung 1, Spalte 2). Für die Darstellung der funktionalen Anforderungen hat sich das Arbeiten mit Use Cases als adäquate Methode erwiesen.

Im Anforderungsmodell hat er die Möglichkeit, die Informationen weiter zu strukturieren und Verbindungen in Form von Generalisierungs- und Abhängigkeitsbeziehungen in das Modell aufzunehmen. Die Diagramme im Modell eignen sich als sehr gute Arbeitsgrundlage für Diskussionen mit Kunden und Entwicklern.

Der Product Owner orientiert sich dabei an den zu spezifizierenden Attributen für einen Use Case, wie Vor- und Nachbedingungen, essenziellem Ablauf usw. Bei der Spezifikation der Use Case-Attribute sowie



**Abb. 3:** WordToTFS in Aktion



**Abb. 4:** Use Case-Modell mit verknüpften Work Items

beim Verknüpfen von Use Cases untereinander oder mit Akteuren werden häufig noch weitere Use Cases entdeckt, an die bis dahin niemand gedacht hatte.

Als Werkzeug hierfür werden die Visual Studio-Modeling-Tools genutzt, die in der Ultimate Edition von Visual Studio das Erstellen von UML-Diagrammen ermöglichen. Die so modellierten Use Cases können mit Work Items im TFS verknüpft werden. Somit werden die UML-Modelle zu einer weiteren „Sicht“ auf die Daten im TFS-Repository.

Die Standardfunktionalität der Visual Studio Modeling Tools erlaubt es dem Product Owner, einzelne UML-Modelllemente wie Subsysteme oder Use Cases selbst mit Work Items im TFS zu verknüpfen. Leider werden dabei die Beziehungen, wie z. B. Abhängigkeiten oder Generalisierung zwischen UML-Modelllementen, nicht auf die verbundenen Work Items übertragen, sondern müssen nochmals manuell nachgepflegt werden.

Zur Lösung dieses Problems wurde in Zusammenarbeit mit AIT für Bosch Rexroth ein Visual-Studio-Plug-in entwickelt, das diese Lücke schließt. Das Plug-in erzeugt automatisch für jedes Diagrammelement ein entsprechendes Work Item inklusive der gewünschten Verknüpfungen, die die modellierten Beziehungen repräsentieren ([siehe Abbildung 4](#)). So stehen dem Product Owner in seinem Product Backlog

auch die Informationen zur Verfügung, die er in seinem UML-Anforderungsmodell erstellt hat.

*Beispiel aus Abbildung 5: Das UML-Subsystem „Kundenverwaltung“ wird auf das gleichnamige Work Item vom Typ PF (Product Feature) abgebildet. Die drei Use Cases aus dem UML-Diagramm werden auf jeweils ein Work Item vom Typ UC (Use Case) abgebildet und mit einer*

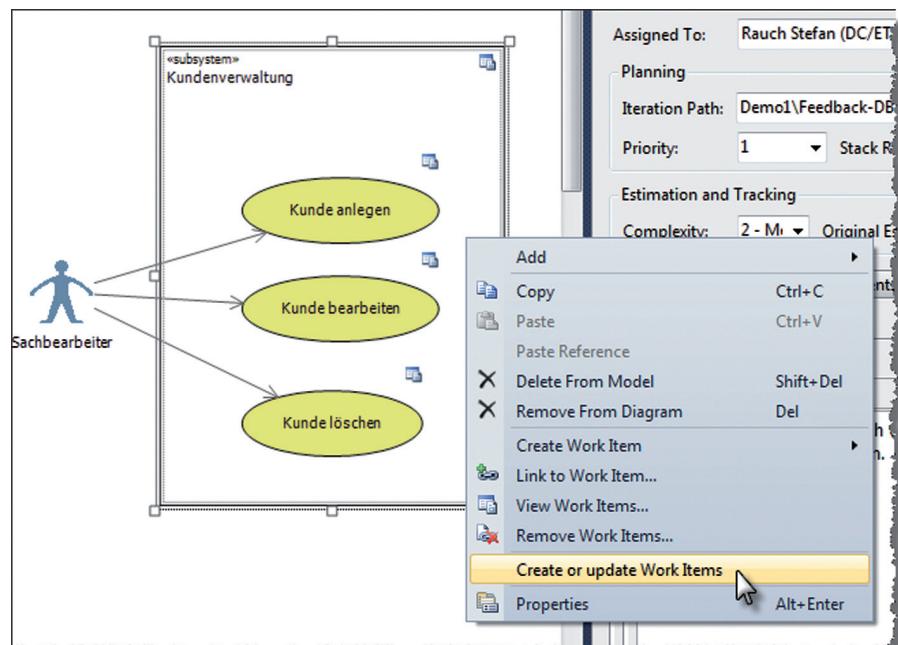
„Child“-Beziehung an das entsprechend übergeordnete Product Feature gebunden.

Zur weiteren Detailierung eines Use Case stehen dem Product Owner mehrere Möglichkeiten offen. Er kann z. B. weitere UML-Diagramme (State- oder Activity-Diagramme usw.) nutzen, um Abläufe detailliert zu beschreiben. Diese Vorgehensweise ist empfehlenswert, wenn er mit Kunden oder der Entwicklung noch Diskussionen führen muss: es steht eine visuelle Diskussionsgrundlage zur Verfügung.

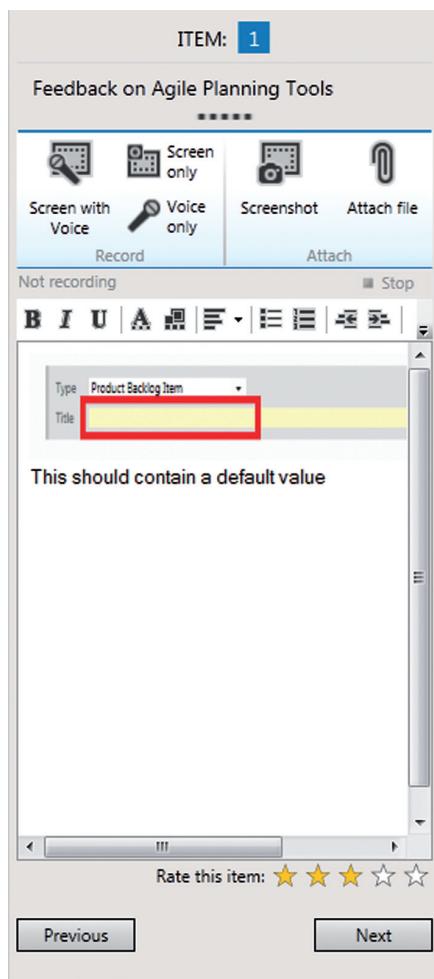
Eine andere Möglichkeit ist, die Akzeptanzkriterien direkt in Form von Test-Case-Work-Items zu formulieren und diese als weitere Spezifikationselemente mit dem Use Cases zu verknüpfen.

Ob ein Product Owner Anforderungen zunächst frei aufnimmt und daraus zu einem späteren Zeitpunkt Use Cases ableitet oder die Kundenanforderungen direkt als Use Cases formuliert, erfordert ein gewisses Maß an Fingerspitzengefühl. Die Erfahrung hat gezeigt, dass bei „Greenfield“-Anwendungen mehr Spielraum erforderlich ist, während bei der Erweiterung von bestehenden Applikationen Kunden oft schon selbst in Use Cases denken.

Im Zusammenspiel mit WordToTFS [AIT] bietet die Visual Studio Modellierungsumgebung dem Product Owner eine professionelle Arbeitsumgebung für die Erfassung und Dokumentation von Kundenanforderungen. Durch die Vor-



**Abb. 5:** Use Case-Diagramme mit einem Klick mit dem TFS synchronisieren.



**Abb. 6:** Feedback kann mit dem Feedback-Tool gegeben werden, welches an den Desktop andockt und z. B. die Aktionen des Nutzers am Bildschirm aufzeichnet.

gehensweise mit Use Cases wird der Product Owner zusätzlich methodisch beim Aufbau des Anforderungsmodells unterstützt.

### Product Backlogs

Da Anforderungen schon als Work Items im TFS erfasst sind, können sie als Grundlage für den Aufbau des Product Backlogs verwendet werden ohne eine weitere Toolgrenze überwinden zu müssen.

Damit ein Team Anforderungen in Form von Use-Case- oder Product-Feature-Work-Items umsetzen kann, müssen diese bestimmte Bedingungen erfüllen: sie sollen SMART sein (Specific/Spezifisch, Measurable/Messbar, Attainable/Machbar, Resourced/Zugewiesen, Timely/Terminierbar (vgl. [Ral12]).

Sind die Anforderungen noch zu grob, um diese Kriterien zu erfüllen, stehen Product Owner und Entwicklungsteam vor der Herausforderung einen passenden Zuschnitt zu finden. Eine Möglichkeit dieser Aufgabe methodisch durchzuführen, ist die Anwendung der Hamburger Methode (siehe [goj12]).

Ein Use Case verwandelt sich dann in einen „Hamburger“, dessen essenzielle Schritte durch den Belag repräsentiert werden. Ein Biss am Hamburger entspricht einem Arbeitspaket in einem Sprint. Das Team hat die Aufgabe, den Burger Biss für Biss zu „verspeisen“. Ein Blick in dieses Vorgehen lohnt sich und macht Spaß.

### Keine Gewähr ohne Bestätigung

Es wird immer wichtiger, schon während des Designs oder der Entwicklung alle wichtigen Projektbeteiligten einzubinden. Der Product Owner zum Beispiel wird zwar ständig in Entscheidungen zu den nicht spezifizierten Details einbezogen. Doch auch die Fachabteilung sollte schon vor dem Release Feedback geben und so Einfluss auf die Funktionen nehmen können.

Mit TFS 2012 können Anfragen für Feedback gezielt via Mail an Projektbeteiligte versendet werden. Diese enthalten einen Link auf die Installation des Feedback-Tools. Das Werkzeug erlaubt es dem Benutzer, unkompliziert Kommentare abzugeben, Screenshots zu erstellen sowie ein Video (natürlich mit Ton) aufzuzeichnen. Sämtliches Feedback wird in Form von Work Items in der Datenbasis des TFS gesammelt – versandet also nicht in irgend einer Inbox. Der Product Owner kann so später das Feedback auswerten und mit einer neuen oder bestehenden Anforderung verknüpfen.

Damit werden späte Änderungen an Anforderungen transparent und die Entscheidungen auch lange nach der Umsetzung nachvollziehbar.

### Fazit

Eine durchgängige Toolkette gewährleistet eine gemeinsame Planungsbasis – in früheren Phasen erstellte Artefakte werden nicht mehr verwendet, weil sie in einem unbekannten, sprich „veralteten“, Zustand sind. Eine integrierte Werkzeugkette stellt für alle Disziplinen geeignete Editoren zur Verfügung und arbeitet auf einer zentralen Datenbasis. Word2TFS, die Anbindung der Visual Studio Modellierumgebung sowie die Standardeditoren des TFS sind eine umfassende, integrierte Umgebung für die Entwicklung und Verwaltung von Anforderungen. Die durchgängige Toolkette hilft auch bei der Umsetzung der Anforderungen und der Nachverfolgung der Entwicklungsaktivitäten. ■

### Links

- [**SvH12**] <http://de.slideshare.net/SvenHubert/tfs-2012-whats-new-in-alm-with-team-foundation-server-overview>
- [**AIT**] <http://www.aitgmbh.de/index.php?id=222>
- [**Ral12**] <http://blog.ralfw.de/2012/06/smarte-nutzenpakete.html>
- [**goj12**] <http://gojko.net/2012/01/23/splitting-user-stories-the-hamburger-method/>