

Agile Requirements mit dem Team Foundation Server 2012

UI-Planung via Storyboard

Die agile Softwareentwicklung genießt inzwischen eine große Beliebtheit. Bei der Einführung von Agilität stehen meistens die Änderungen für die Softwareentwickler im Vordergrund. Schon bald stellt man aber fest, dass auch auf Seite des Requirements Engineering große Veränderungen anstehen. Der neue Team Foundation Server (TFS) 2012 bietet neben vielfältigen Verbesserungen bestehender Funktionalitäten neue Konzepte, um das agile Requirements Management besser zu unterstützen.

von Thorsten Ruf und Roland Simon

Das klassische Requirements Engineering gilt im Kontext von Agilität als bürokratisch und schwerfällig. Mit der lieb gewonnenen Flexibilität im Projektverlauf ist es verschwendete Mühe und vor allem Zeit, vor Beginn der Implementierung sämtliche Anforderungen detailliert in einem seitenstarken und komplexen Anforderungsdokument zu spezifizieren. In Projekten mit etablierten agilen Vorgehensweisen zeigt die Erfahrung, dass auch das Requirements Engineering an den agilen Ansatz adaptiert werden muss. Somit wird im Projektverlauf nicht nur iterativ-inkrementell implementiert, sondern auch das Requirements Enginee-

ring muss fortlaufend erfolgen. Hierbei ergibt sich eine stetige Verfeinerung der Anforderungen und somit ein fortlaufendes Auffüllen der bestehenden Wissenslücken im Domänen-Know-how des Kunden.

Im agilen Requirements Engineering werden sehr hilfreiche, klassische Techniken jedoch keinesfalls außer Acht gelassen. Hier liegt es in der Hand des erfahrenen Requirement Engineers, die passende Technik zum richtigen Zeitpunkt anzuwenden. Er muss sicherstellen, dass komplexe Sachverhalte mit geeigneten Methoden beschrieben werden und das Entwicklungsteam die Anforderungen für einen Sprint gut verstehen kann. Zum Projektstart im agilen Umfeld bedarf es einer detaillierten Beschreibung der ersten Anforderungen in Form eines initialen Product

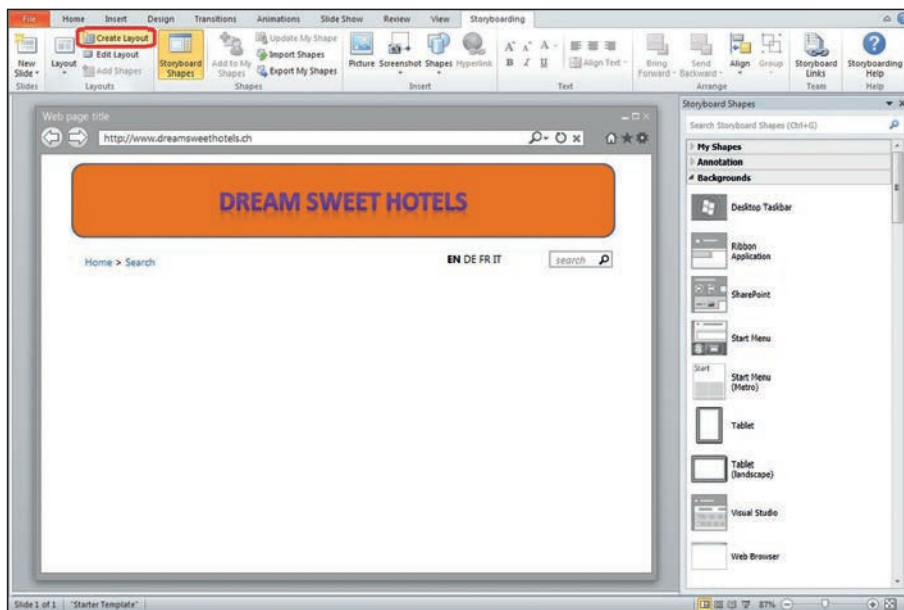


Abb. 1: Erstellen eines Templates

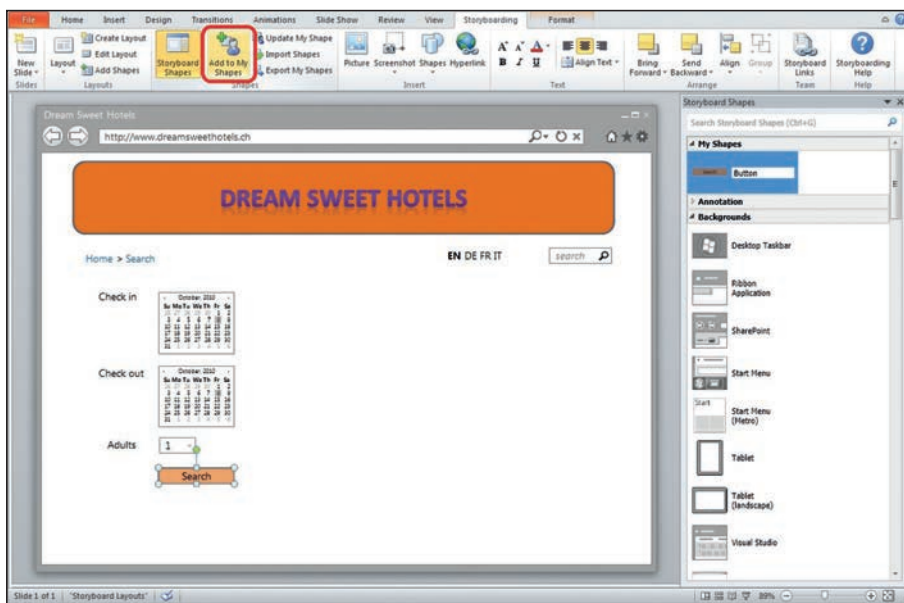


Abb. 2: Hinzufügen eines Custom Shape

Backlogs, das im TFS 2012 noch einfacher und schneller per Drag-and-Drop-Funktionalität verwaltet werden kann. Diese ersten Anforderungen werden jedoch nur so detailliert spezifiziert, um deren Kosten, Risiken und Nutzen abschätzen und priorisieren zu können, und soweit verfeinert bis das Entwicklungsteam für die ersten ein bis zwei Sprints beschäftigt werden kann. In dieser Phase hat es sich bewährt, auf die klassischen Techniken in Form von Workshops und Interviews zu setzen. Nach jedem Sprint ist es von großer Bedeutung, für die realisierten Anforderungen ein Feedback bei den Stakeholdern einzuholen, um die Anforderungen zu verifizieren und zugleich neue Anforderungen zur Aufnahme ins Product Backlog zu identifizieren. Der TFS 2012 stellt für die weitere Verfeinerung von Anforderungen ein einfaches Storyboarding-Tool zur Verfügung, mit dessen Hil-

fe User-Interface-Skizzen erstellt werden können. Diese Storyboards können mit den Work Items des TFS verknüpft werden und stehen dem Entwickler bei der Umsetzung direkt zur Verfügung. Sie ergänzen die textuelle Beschreibung der User Story. Mit der ebenfalls neu eingeführten Feedback-Request/Response-Funktionalität bietet der TFS 2012 eine Möglichkeit, bei Stakeholdern aktiv ein Feedback einzuholen und sie näher an das Geschehen heranzubringen. Alle Projektbeteiligten können schnell und einfach Einfluss auf die Weiterentwicklung des Projekts nehmen. Schauen wir uns als erstes das neue Storyboarding des Team Foundation Server 2012 genauer an.

Storyboarding

Das TFS-Team setzt für das Erstellen von Storyboards auf PowerPoint. Dies mag im ersten Moment verwundern. Präsentationen und Storyboards haben doch nicht viel gemeinsam, möchte man einwenden. Aber sowohl Programme für die Erstellung von Präsentationen als auch Storyboards verwenden einen Designer für die grafische Gestaltung. Ein weiterer gewichtiger Pluspunkt für PowerPoint ist die Verbreitung beim Zielpublikum. Die Benutzeroberflächen-Skizzen sollen ja nicht primär von Softwareentwicklern erstellt werden, sondern von den Stakeholdern mit viel Geschäftswissen und engem Kontakt zum Endbenutzer. Diese haben vielfach PowerPoint nicht nur bereits installiert, sondern auch fundierte Erfahrungen mit der Anwendung. Eine kritische Anmerkung hierzu finden Sie im Kasten „Benutzerkreis

Storyboarding“.

Damit Storyboards kosteneffizient erstellt werden können, hat das TFS-Team ein paar PowerPoint-Erweiterungen bereitgestellt. Die auffälligste ist das zusätzliche Ribbon Storyboarding. In diesem sind die Funktionen für das Erstellen von Storyboards zusammengefasst. Das TFS-Entwicklerteam wählte also den gleichen Ansatz wie bei der Erweiterung für Microsoft Excel mit dem Team Ribbon. Dieses wird für die Kommunikation von Excel mit dem TFS eingesetzt.

Durch das Starten von PowerPoint über das Windows Startmenü *Microsoft Visual Studio 2012 > Powerpoint Storyboarding* wird ein für das Storyboarding vorbereitetes Template geöffnet. Als erster Schritt werden über das Storyboarding Ribbon die Storyboard Shapes eingeblendet. Unter „Backgrounds“ kann eine

vorbereitete Form gewählt werden. Wir ziehen den Webbrowser-Background für eine Webapplikation auf die Folie – Entschuldigung, das Storyboard (da wir uns im PowerPoint befinden, verwenden wir für das bessere Verständnis auch die Begriffe aus der Präsentationstechnik). Neben dem Anpassen der Texte des Webbrowser-Hintergrundes werden Benutzeroberflächenkomponenten wie Header und Footer hinzugefügt. Zuerst werden nur Komponenten hinzugefügt, die auf allen Screens vorkommen, mit dem Ziel, ein Template zu erstellen. Das Template wird einmalig erstellt und kann anschließend mehrfach wiederverwendet werden. Sobald man mit dem Template zufrieden ist, kann es über den Befehl `CREATE STORYBOARD` zu den Templates hinzugefügt werden. Für diesen Schritt muss es noch nicht abschließend designt und aufgebaut sein (Abb. 1). Ein großer Vorteil der Templates ist, dass sie nachträglich editiert werden können und auf sämtlichen Folien nachgeführt werden. Ist ein Benutzeroberflächenlayout schon umgesetzt, muss es nicht im PowerPoint Storyboarding nachgezeichnet werden. Über das integrierte Screenshot-Tool kann das bestehende Layout als Hintergrundbild in PowerPoint importiert und im Template eingebunden werden.

Nun steht die Grundstruktur der Benutzeroberfläche. Jetzt geht es um das Erstellen der einzelnen Seiten. Hierfür wird eine neue Folie mit dem oben erstellten

Template erstellt. Auf diese Folie werden die einzelnen UI-Komponenten gezogen. Für das Gestalten können neben den Storyboard-spezifischen Funktionen die umfangreichen Standardfunktionalitäten von PowerPoint verwendet werden. Bei den Storyboard Shapes gibt es schon viele vorbereitete Formen. Den Backgrounds sind wir schon oben begegnet. Unter Common gibt es die Standard-Controls, wie Button, Textbox, Text, Radiobuttons etc. Man findet in der Shape-Galerie aber auch ausgefallenerere Elemente, wie Street Map und Tag Cloud. Für die Benutzeroberflächen (klassisches) Windows, Metro und Windows Phone gibt es schon vorgefertigte Controls im entsprechenden Style. Es werden also schon die neuesten Benutzeroberflächentechnologien aus dem Hause Microsoft unterstützt.

Benutzerkreis Storyboarding

Bedenkt man die Lizenzpolitik von Microsoft, ist es fraglich, ob alle Stakeholder als Benutzerkreis vom Storyboarding betrachtet werden: Bei der Betaversion muss Visual Studio in der Premium-, Ultimate- oder Test-Professional-Version lokal installiert werden, um die Funktion „PowerPoint Storyboarding“ benutzen zu können. Die Autoren hoffen, dass in der Release-Version die Möglichkeit geboten wird, diese Funktion auch auf PCs ohne Visual Studio verwenden zu können [1].

Anzeige

entwickler.press

Oliver Sturm

F#

Einstieg und praktische Anwendung

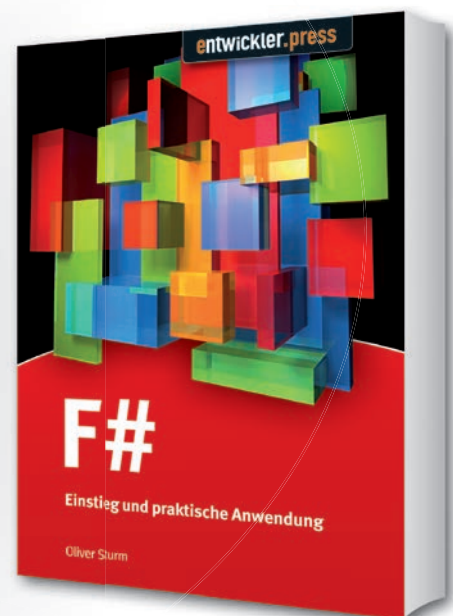
190 Seiten, Juli 2012

- PRINT** ISBN: 978-3-86802-083-0
Preis: 29,90 € / 30,80 € (A)
- PDF** ISBN: 978-3-86802-270-4
Preis: 18,99 €
- EPUB** ISBN: 978-3-86802-608-5
Preis: 18,99 €

Mit Visual Studio 2010 hielt eine neue Programmiersprache in der .NET-Welt Einzug: F#. Das Stadium der Forschungssprache hat F# hinter sich gelassen und stellt Ihnen mit seinem hybriden Ansatz Möglichkeiten zur Verfügung, die weit über das Können anderer .NET-Sprachen hinausgehen. Die objektorientierte Welt ist sogar besser als in C# unterstützt und die funktionale Programmierung eröffnet in vielen Bereichen neue und effizientere Wege.

Suchen Sie nach der leistungsfähigsten .NET-Sprache? Mögen Sie funktionale Ideen wie LINQ? Und erwarten Sie von einer Sprache größtmögliche Flexibilität sowie beste Hilfestellung in jeder Programmiersituation? Dann sind F# und dieses Buch genau das Richtige für Sie. Anhand von klaren Beispielen ermöglicht Oliver Sturm einen einfachen Einstieg in die neue Denkweise von F# und bietet dem Leser die Möglichkeit, schon nach kurzer Zeit eigene produktive Anwendungen zu erstellen.

Mehr Infos zum Buch finden Sie unter: www.entwickler.press.de/fsharp



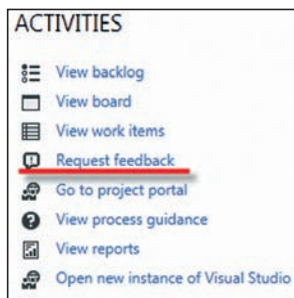


Abb. 3: Web Access „Request feedback“

Obschon viele Controls zur Verfügung stehen, werden Sie schnell eigene Controls und Control-Gruppen zur Wiederverwendung gestalten wollen. Ein einfaches Beispiel ist ein farblich gestalteter Button für eine Web- oder WPF-Anwendung. Dieser kann über den Befehl ADD TO MY SHAPES im Storyboard Ribbon zu den Shapes hinzugefügt werden (Abb. 2). Ab jetzt können sie genau gleich wie die Standardkomponente

verwendet werden. Die Autoren möchten noch auf eine Falle hinweisen: Die eigenen Shapes können aktualisiert werden. Dies betrifft aber nur die Vorlage und deren zukünftige Verwendungen. Ist eine Form einmal verwendet, wird sie nicht mehr automatisch an die neue Formatierung angepasst. Diese müssen von Hand nachgeführt werden. Also gestalten Sie die eigenen Shapes vollständig, bevor Sie diese vervielfachen. Dies steht im Gegensatz zu den Templates, die automatisch angepasst werden.

Mit dem bloßen Erstellen eines Storyboards hat man noch nicht viel erreicht. Es muss noch allen Projektbeteiligten zur Verfügung gestellt und von ihnen verwendet werden. Bei dem agilen Vorgehen gehören die Storyboards wie die restlichen Requirements zur User Story. Im TFS 2012 wird die User Story als Product Backlog Item abgebildet. Dieses Item hat einen Extra-

Tab für Storyboards. Um ein Storyboard hinzuzufügen, muss es an einer so genannten „Shared Location“ abgespeichert sein. Im ersten Moment empfindet man dies als mühsam und umständlich. Denkt man darüber nach, entdeckt man einen sehr großen Vorteil der gewählten Lösung: Ein Storyboard beschreibt vielfach die Benutzeroberfläche von mehreren User Stories. Das gleiche Board sollte also zu mehreren Product Backlog Items verlinkt werden können. Wird es bei mehreren Items als Attachment hinzugefügt, kann es nicht mehr an einem zentralen Ort erweitert und angepasst werden. Ist das Storyboard an einer Shared Location abgelegt, kann es zentral editiert werden und alle Stakeholder haben Zugriff auf die aktuellste Version. Im TFS bietet sich die Dokumentenbibliothek der Projekt-SharePoint-Seite geradezu als Shared Location an. Zumal SharePoint von Haus aus das Editieren von mehreren Benutzern managen kann. Hierfür muss das obligatorische Check-out auf Ebene „Dokumentenbibliothek“ eingeschaltet werden. Zurück zu unserer erstellten Storyboard-Datei. Sobald Sie diese in einer SharePoint-Dokumentenbibliothek gespeichert haben, steht sie zum Verlinken mit den Product Backlog Items zur Verfügung. Wenn Sie das Storyboard über SharePoint öffnen, steht der Ribbon Button „Storyboard Links“ zur Verfügung. Über diesen kann die Datei mit bestehenden TFS Work Items verlinkt werden.

Bevor wir das Thema Storyboarding verlassen, zur Vollständigkeit noch einen Verweis auf ein anderes Storyboarding-Tool aus dem Hause Microsoft: Wem die vorhandene Funktionalität in PowerPoint nicht ausreicht, sollte sich die Applikation SketchFlow [2] ansehen. Diese ist Bestandteil von Microsoft Expression Studio in der Ultimate-Version. Sie kann als sehr umfangreiche Alternative zum PowerPoint Storyboarding verwendet werden und bietet noch weitreichendere Möglichkeiten.

Request Feedback

Ein großer Vorteil der agilen Entwicklung ist, dass am Ende jeder Iteration ein lauffähiger, neuer Release der Applikation entsteht. Die Stakeholder haben damit die Möglichkeit, frühzeitig zu testen und sich über den aktuellen Stand der Entwicklung zu informieren. Es ist wichtig, für jedes Release ein fundiertes Feedback von allen Projektbeteiligten einzuholen, um die Erwartungen mit dem Projektfortschritt abzugleichen. Dies geschieht sehr häufig in unstrukturierter und mündlicher Form im Rahmen des

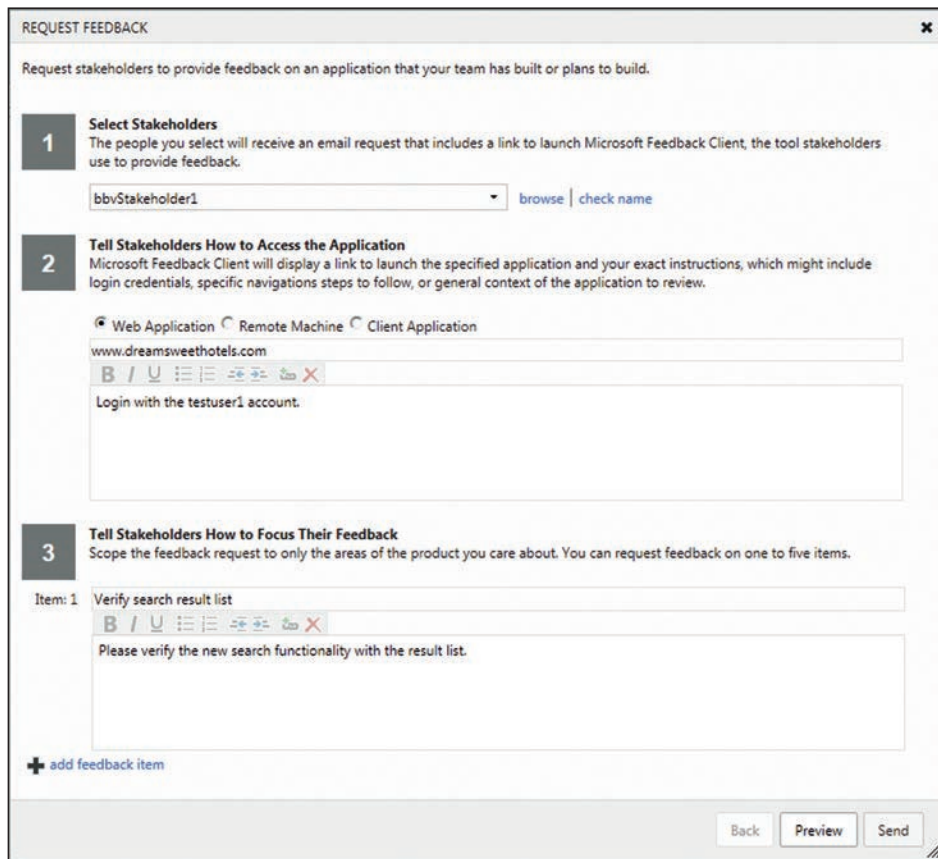


Abb. 4: Request-Feedback-Formular

Sprint Reviews. Hierbei besteht die Gefahr, dass durch den gesteckten Rahmen Änderungswünsche nicht oder nur teilweise aufgenommen werden. Zudem nehmen im Normalfall nicht alle Stakeholder am Sprint Review teil. Trotzdem sollten alle Stakeholder die Möglichkeit haben, ihr Feedback zu dem aktuellen Stand abzugeben. Hierzu bietet der Team Foundation Server 2012 mithilfe der neu integrierten Feedback-Funktionalität die Möglichkeit, aktiv ein Feedback von Stakeholdern einzuholen und dieses auch in strukturierter Form im TFS zu verwalten. Ein Feedback Request kann aktuell nur über TFS Web Access erstellt werden. Hierzu findet man bei den Activities den Link „Request feedback“ (Abb. 3).

Das Formular zum Versenden des Request, wie in **Abbildung 4** zu sehen ist, enthält neben der Angabe eines oder mehrerer Stakeholder noch weitere Informationen; beispielsweise wo der Stakeholder die Applikation mit dem neuen Feature testen kann. Zudem können bis zu fünf Einträge hinzugefügt werden, auf die sich das Feedback bzw. die Stakeholder bei ihrem Review konzentrieren sollen.

Die Stakeholder erhalten per E-Mail die Aufforderung, ihr Review durchzuführen und das Feedback zu erfassen. Zur Verwendung kommt hierbei das Tool „Feedback Client“, das beim Stakeholder installiert sein muss. Die Nachricht enthält ebenfalls einen Link zum Download des „Feedback Client“. Dieser steht bereits in mehreren Sprachen zum Download zur Verfügung. Startet der Stakeholder den Feedback Client durch Verwendung des Links „Start your feedback session“ direkt aus der E-Mail heraus, werden die Details des Requests, die im Formular zuvor eingegeben wurden, angezeigt (Abb. 5). Der Applikationslink führt den Benutzer direkt zu der zu testenden Applikation (Abb. 6).

Der Benutzer führt das Review der Applikation selbständig durch und kann neben Textkommentaren zusätzliche Screenshots, Videos und Sprachkommentare hinzufügen. Zudem lässt sich ein Rating in Form von ein bis fünf Abstufungen dem Review hinzufügen. Durch das Abschließen des Feedbacks wird im Team Foundation Server dem vorhandenen Feedback Request ein neues Work Item vom Typ *Feedback Response* erstellt und kann vom Team bei der Planung des nächsten Sprints mit einbezogen werden. Anhand des Feedbacks können weitere User Stories erfasst und zeitnah umgesetzt werden. Mithilfe des Feedback-Tools fühlt es sich fast so an, als ob der Stakeholder beim Review direkt daneben sitzt. Das bringt einen enormen Vorteil in der Zusammenarbeit und verbessert die zeitnahe Kontrolle des aktuellen Fortschritts während des Projekts.

Fazit

Das Storyboarding ist eine ideale Diskussionsgrundlage für das Definieren von Benutzeroberflächen und Anforderungen. Fachpersonen können ihre Erwar-

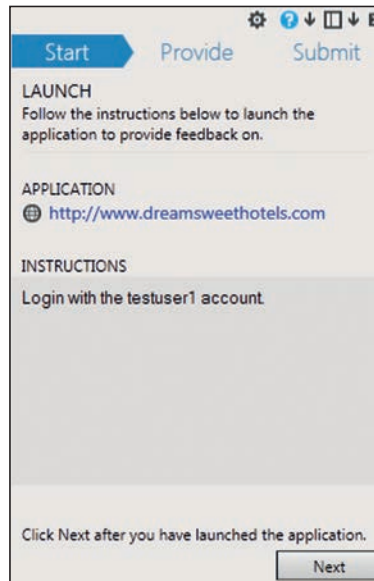


Abb. 5: Feedback-Client-Start

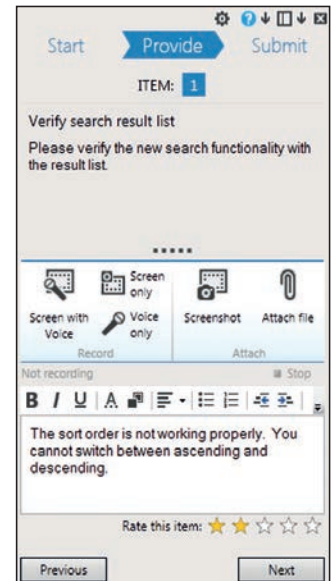


Abb. 6: Feedback-Client-Verwendung

tungen an die zu erstellende Applikation meistens am besten über Benutzeroberflächen-Skizzen ausdrücken. Diese sind, im wahrsten Sinn des Wortes, bildlicher als Textbeschreibungen in User Stories. Durch die Benutzeroberflächen-Skizzen können Missverständnisse nahezu ausgeschlossen und die Projekteffektivität gesteigert werden. Durch Einsatz von Continuous Deployment und Feedback Manager kann unmittelbar nach jedem Sprint bei den relevanten Stakeholdern eine konstruktive Kritik eingeholt werden. Neben dem sofortigen Reagieren auf Änderungswünsche, können die Fachpersonen in das Projektboot geholt werden. Ihnen wird nicht eine pfannenfertige Anwendung hingestellt, sie dürfen aktiv mitgestalten und werden die neue Applikation auch als „ihre“ ansehen. Die Akzeptanz der Benutzer hat man schon vor dem ersten produktiven Release. Mit dem Team Foundation Server 2012 ist es Microsoft gelungen, das agile Projektmanagement hervorragend zu unterstützen.



Die Autoren **Thorsten Ruf** und **Roland Simon** sind langjährige Software Engineers bei der bbv Software Services AG. Sie sind verantwortlich für die Einführung und den Betrieb des TFS bei Inhouse-Softwareprojekten. Sie unterstützen zahlreiche Kunden sowohl bei der

Einführung als auch bei Migrationsprojekten bis hin zu individuellen Anpassungen des Team Foundation Server.

Links & Literatur

- [1] <http://www.microsoft.com/visualstudio/11/en-us/products/compare>
- [2] http://www.microsoft.com/expression/products/Sketchflow_Overview.aspx