



□ Frank Maar

(E-Mail: Frank.Maar@microsoft.com)

ist Technologieberater für Softwareentwicklungstools und -prozesse für Großkunden bei der Microsoft Deutschland GmbH. Eine optimale Beratung erfordert eine Balance in dem Dreieck Menschen, Tools und Prozesse zu finden. Dieser Herausforderung stellt er sich jeden Tag aufs Neue und mit Freude.

## Agile Softwareentwicklung mit dem Visual Studio Team System nach Scrum

Softwareentwicklung verlief vor einigen Jahrzehnten noch spontan und unkontrolliert. Dann wurden formale Prozesse, wie z.B. V-Modell XT, eingeführt, die einen geregelten Ablauf auf Softwareprojekte übertrugen. Mit diesem Vorgehen konnten Auslieferungstermine, Qualität und Budget viel genauer eingehalten werden. Formale Prozesse gehen von der Annahme aus, dass sich die Anforderungen der Anwender im Verlaufe des Projektes nicht ändern. Diese Eigenschaft sorgt dafür, dass formale Prozesse häufig für Festpreisprojekte verwendet werden. Agile Prozesse wurden eingeführt, um schneller auf sich ändernde Anforderungen reagieren zu können. Geschäftsprozesse sind nicht mehr in „Stein gemeißelt“ und Fachbereiche erwarten eine schnelle Abbildung der Prozesse auf IT. Die Wahl der Entwicklungsprozesse hängt von gesetzlichen Vorgaben, Unternehmensstandards oder aktuellen Trends ab. Scrum ist solch ein Trend.

Prozesse an sich können in Form von Dokumenten beschrieben werden. Wie ist dann allerdings sichergestellt, dass wirklich nach dem Prozess gearbeitet wird? Lasten- und Pflichtenheft, Testspezifikation, Architekturbeschreibung und Projektpläne sind sicherlich wichtig und reichten bisher in vielen Projekten aus. Welche Anforderungen sind wirklich implementiert worden? Welche Anforderungen wurden tatsächlich getestet? Solche Fragen werden meist mit Excel-Listen beantwortet, die manuell vom Projektleiter gepflegt werden. Die Aktualität und die Güte der Daten beruhen darauf, dass immer und immer wieder die gleichen Fragen gestellt werden. Projektleiter, Entwickler und Tester haben aber viele andere Dinge zu tun, als immer wieder die gleichen Fragen zu klären. Einen Zwischenschritt stellt das Verwenden mehrerer Repositories dar. In größeren Projekten gibt es ein Repository für die Versionsverwaltung, Anforderungsmanagement, Testing und Build. In welcher Version ist welcher Bug aufgetreten? Wo ist er wieder behoben worden und welche Dateien beinhalten die Fehlerkorrektur? Das sind keine einfach zu beantwortenden Fragen. Ein zentrales Data-Repository, in dem alle projektrelevanten Daten gesam-

melt werden, ist an dieser Stelle die Antwort. Vom Anforderungsmanagement, über Projektplanung, Architekturdefinition, Testing und Releasemanagement arbeiten alle Projektbeteiligten mit dem gleichen Repository. Das System muss für kleine Teams von 5 bis 10 Entwicklern und

auch für Entwicklerteams mit 1.000 bis 3.000 Entwickler nutzbar sein. Verteilte Entwicklung ist heutzutage ebenfalls Standard.

Das Visual Studio Team System stellt eine integrierte und offene Software-Entwicklungsplattform für kleine und gro-

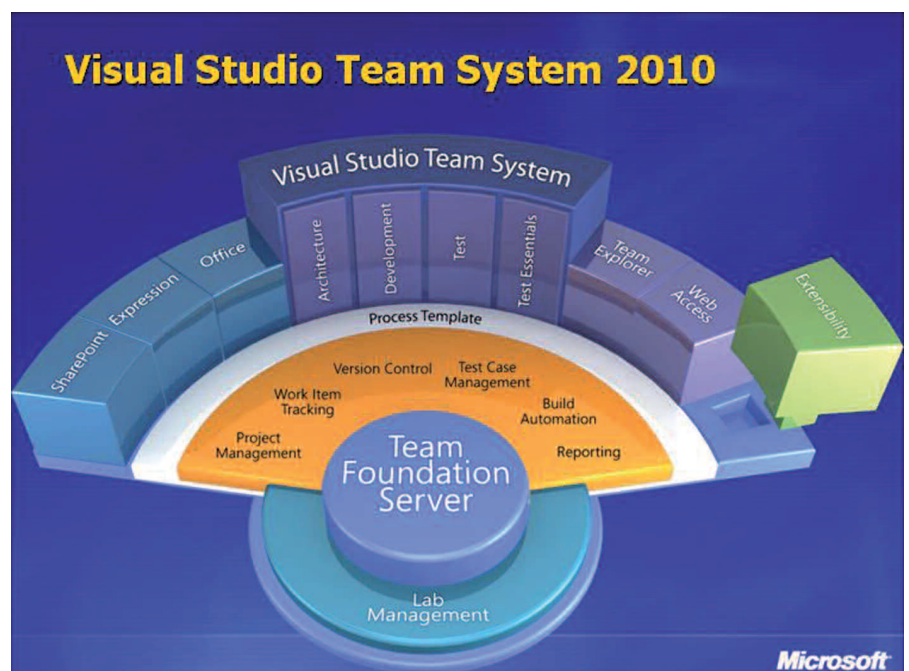


Abbildung 1: Elemente des Visual Studio Team Systems

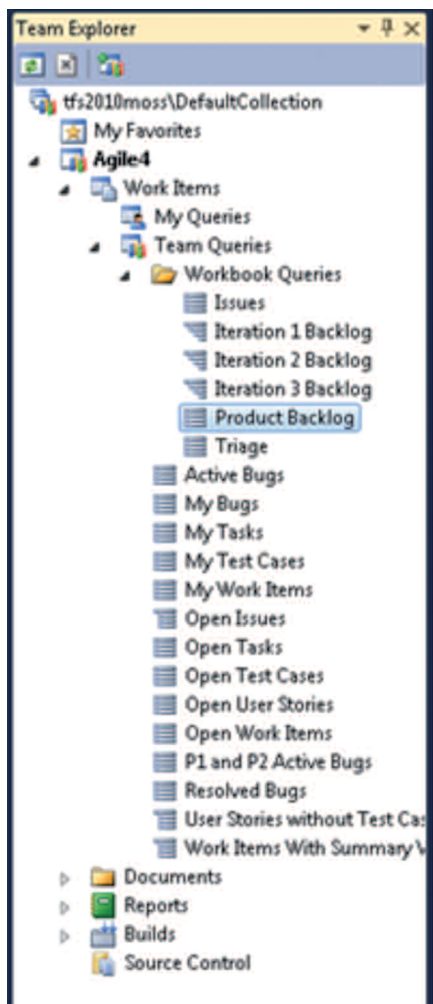


Abbildung 2: Abfrage Product Backlog Im Team Explorer

ße Teams dar. Es können verschiedene Process Templates verwendet werden. In den Process Templates sind z.B. alle Typen von Arbeitspaketen (Work Items) und deren Workflows hinterlegt. Wenn ein neues Softwareprojekt angelegt wird, können ähnlich wie bei Word Templates verwendet werden. Dadurch kann ein neues Projekt innerhalb von 5 Minuten angelegt werden und über dieses Template wird sichergestellt, dass tatsächlich nach dem vorgegebenen Prozess gearbeitet wird. Für Scrum gibt es bereits eine Vorlage unter <http://www.scrumforteamssystem.com>, die bei Bedarf mit dem Process Editor angepasst werden kann. Unternehmen orientieren sich in aller Regel an Standardprozessen, benötigen aber mitunter zusätzliche Attribute bei Arbeitspaketen (z.B. das Attribut Datenbanksystem bei Bugs) oder weitere Genehmigungsschritte. Das Team System und auch die Entwicklungsprozesse sind unabhängig von der verwendeten Technologie. Eine Unterstützung von

.NET-Projekten mit Visual Studio wird sicherlich als gegeben vorausgesetzt. Java-Entwickler und Softwareentwickler mit anderen Betriebssystemen als Windows können mit Produkten von Partnern (z.B. <http://www.teamprise.com>) ebenfalls mit dem Team Foundation Server arbeiten.

Das Visual Studio Team System unterstützt alle Prozessschritte des Application Lifecycle Management, also in den Schritten Planung, Design, Entwicklung, Build und Test. Der Team Foundation Server ist eine Serverkomponente, die auf einem Windows Server 2003 oder 2008 installiert wird. Es wird empfohlen, den Team Foundation Server als Standarddienst (Shared Service) der zentralen IT anzubieten, damit der Betrieb für diesen Dienst nur einmal aufgebaut wird und die Kosten gering gehalten werden. Die anderen Komponenten sind Clients, mit denen auf die Daten des Team Foundation Servers zugegriffen werden können.

In der Phase der Projektplanung werden die User Stories vom Product Owner im Team Foundation Server erfasst. Als Tool zur Eingabe der Daten kommen hierfür Excel, Project, Web Access oder auch Visual Studio in Frage. Um den Nutzer nicht zu überfordern, sollte ein Tool als Standard festgelegt werden. Der Product Backlog und die einzelnen Iteration Backlogs liegen als fertige Abfragen vor, die in all den genannten Tools aufgerufen werden können. Die Wahl des Clienttools hängt von den Vorlieben und der Infrastruktur ab. Project ist für den erfahrenen Projektleiter. Damit können dann auch Gantt-Diagramme erstellt werden. Excel für die gelegentliche oder eher listenorientierte Bearbeitung von Anforderungen und Arbeitsaufgaben. Der Web Access wird gerne von den Benutzergruppen verwendet, die eher selten damit arbeiten oder deren Client-Infrastruktur nicht bekannt ist. Mitarbeiter im Help-Desk können darüber

z.B. dem System Bugs mitteilen. Visual Studio ist für Softwareentwickler konzipiert worden, kann aber ebenfalls zur Eingabe von User Stories verwendet werden. Ein typischer Anwendungsfall sind kleine Teams, die einzig mit Visual Studio arbeiten. Als minimale Visual Studio Komponente reicht hierfür übrigens der Team Explorer. Dokumentation in Word- oder PDF-Form kann mit den User Stories verknüpft werden, sodass ein Teil der Daten strukturiert und ein Teil unstrukturiert erfasst werden kann.

In dem Bild sehen Sie Bestandteile eines Projektes im Team Foundation Server, wie Work Items, Documents, Reports, Builds und Source Control. Die farblich hervorgehobene Abfrage „Product Backlog“ liegt z.B. bereits als Standardabfrage vor.

Ab der Version 2010 stehen UML-Diagramme zur Verfügung, sodass auch Use Case Diagramme erstellt und diese mit User Stories verknüpft werden können.

Alle User Stories werden im Product Backlog gesammelt und die relative Größe und Komplexität in Story Points geschätzt.

In **Abbildung 3** wurde die Abfrage „Iteration 3 Backlog“ in Excel aufgerufen. Alle User Stories mit den zugehörigen Tasks werden dann in Listenform dargestellt. User Stories und Tasks werden als Parent-Child-Beziehung erfasst, sodass in der Darstellung alle Tasks nach einer User Story zu dieser gehören und nach rechts eingerückt mit Title 2 bezeichnet werden. Es erfolgt dann eine Zuweisung der Aufgabe an ein Teammitglied und eine Angabe der geplanten Stunden.

Die Multi-Ressource-Planung ist eine Stärke von Project. Diese Aufgabe kann aber auch mit Excel-Arbeitsmappen durchgeführt werden. Wochenarbeitszeit und arbeitsfreie Zeiten wurden in den Excel-Blättern Settings und Interruptions erfasst und sind Basis der berechneten Auslastung.

Ein Teammitglied ist in dem Beispiel in

 A screenshot of a Microsoft Excel spreadsheet titled 'Iteration 3 Backlog - Microsoft Excel'. The spreadsheet displays a list of work items with columns for ID, Work Item, Stack, Title 1, Title 2, State, Story Points, Assigned To, Remaining Work, and Completed Work. The data is as follows:
 

ID	Work Item	Stack	Title 1	Title 2	State	Story Points	Assigned To	Remaining Work	Completed Work
54	User Story	1	As a customer I want to customize my landing page		Active	5	Sunder Raman		
26	Task		Test all the login scenarios		Active		Gregg Boer		3
27	Task		Add error handling to all navigation elen		Active		Jeff Beehler		4
28	Task		Add transaction handling to all stored pr		Active		Aaron Bjork		5
29	Task		Run perf analysis on all database calls fo		Active		Jeff Beehler		6
30	Task		Test the login screen		Active		Jeff Beehler		5
56	Task		Test all the navigation from the main me		Active		Stephanie Cuthbert		4
9	User Story	2	As a new customer I want log in so that I can use the		Active		8 Sunder Raman		
14	User Story	3	As a new customer I want log in so that I order thi		Active		5 Sunder Raman		

Abbildung 3: User Stories und Tasks

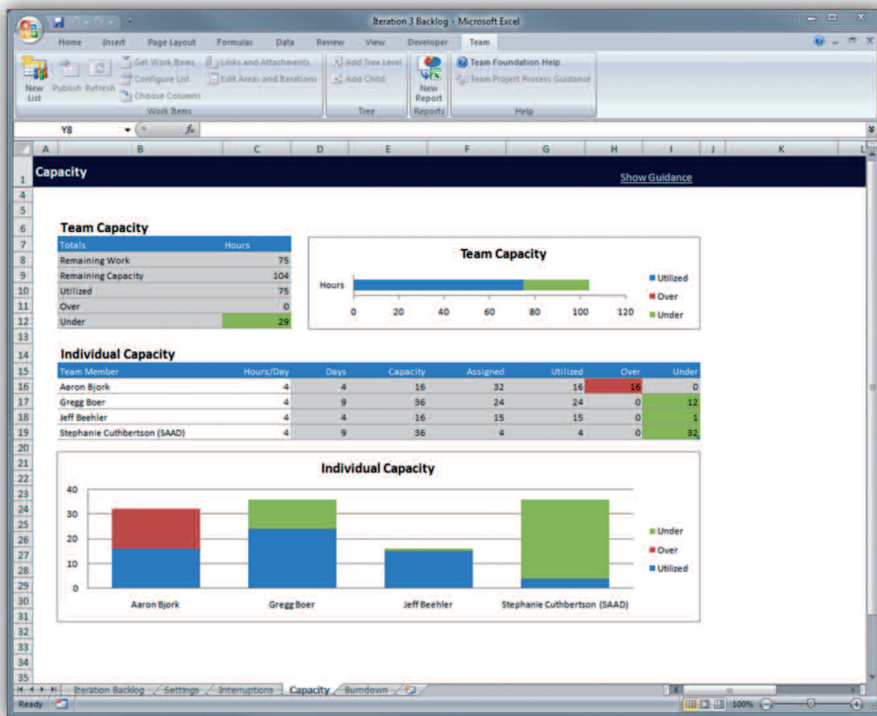


Abbildung 4: Auslastungsgrad im Team

der Abbildung überlastet. Dessen Aufgaben könnten auf andere Teammitglieder verteilt werden, weil in dem gesamten Team noch Reserven vorliegen. Wenn das gesamte Team überlastet wäre, müssten User Stories aus der Iteration herausgenommen werden. In der Projektdurchführung ist es wichtig, alle wichtigen Kennzahlen auf einen Blick zu sehen. Hierfür wird ein Dashboard mitgeliefert, wobei auch bei Bedarf in die Einzelberichte hineingeschaut werden kann.

Das Erstellen von Berichten ist oftmals sehr aufwändig, weil Daten aus unterschiedlichen Tools gesammelt und aufbereitet werden müssen. Im Team Foundation Server hingegen wird ein Data Warehouse auf der Basis der Analysis Services des SQL Servers genutzt, bei dem bereits ein einheitliches Datenmodell vorliegt. Als Berichtsgenerator werden die Reporting Services verwendet. Die Reporting Services sind Bestandteil des SQL Servers und liegen in der Beschreibungssprache Reporting Definition Language vor. Im Datawarehouse werden alle Attribute der Arbeitspakete, CheckIns, Build- und Testvorgänge mit dem zeitlichen Verlauf gespeichert. Die Build-Protokolle werten diese Daten aus und stellen dar, welche Arbeitspakete in dem erstellten Release eingefügt und welche Tests durchgeführt wur-

den. Sie können auch Ihre eigenen Berichte erstellen, da die Struktur des Data Warehouses dokumentiert und auf den MSDN-Seiten abgerufen werden kann.

Die Rückmeldungen, welche Arbeitspakete vom Entwickler abgearbeitet wurden,

erfolgen direkt aus der Entwicklungsumgebung heraus und können über Regeln auf dem Server erzwungen werden. Darüber wird sichergestellt, dass die Rückmeldungen wirklich erfolgen und der Entwickler in seinen gewohnten Tools bleibt.

In der Version 2010 des Visual Studio wird dem Tester erstmals ein eigenständiger Client angeboten, in dem die Tests geplant und durchgeführt werden können. Videos, Screenshots und Systeminformationen können im Fehlerfall an Bugs geheftet werden, sodass der Entwickler die notwendigen Informationen hat, um den Fehler zu beheben. Wenn das noch nicht zur Fehlerbehebung reicht, kann ein Debuggen mit Informationen aus dem Produktsystem (historical debugging) oder ein Snapshot einer virtualisierten Umgebung verwendet werden. Damit wird das zeitaufwändige Nachstellen von Fehlersituationen vermieden.

Ein automatisierter Build-Vorgang, der auch die Tests beinhalten kann, rundet die Qualitätssicherung ab. Test- und Build-Vorgänge können bei zu hoher Auslastung auf andere Server ausgelagert werden.

Agile Methoden erfordern auch agile Tools. Mit dem Visual Studio Team System und dem Scrum-Template steht eine integrierte Toolplattform zur Verfügung, die mit einem geringen Aufwand eine maximale Scrum-Unterstützung bietet. ■

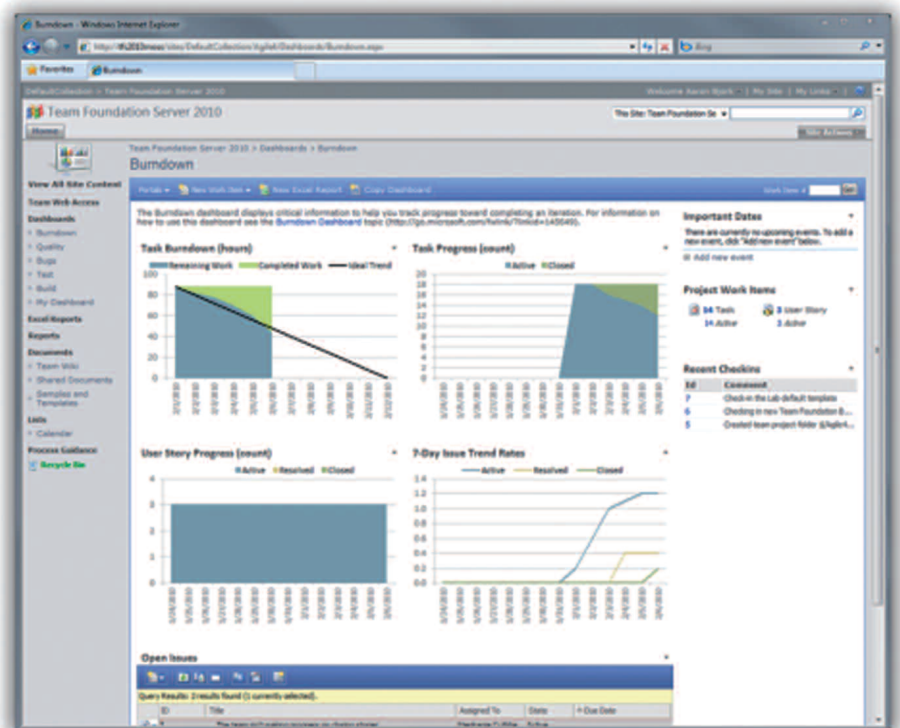


Abbildung 5: Dashboard mit Burn-Down-Chart