



□ Stefan Mieth

[E-Mail: Stefan.Mieth@aitgmbh.de]

ist für die AIT als Berater im AIT Team SystemPro Team tätig. Zu seinen Schwerpunkten gehört die Anpassung von Microsofts Application-Lifecycle-Management-Plattform an kundenspezifische Anforderungen. Er unterstützt Unternehmen bei der Einführung und Anpassung des Visual Studio Team Foundation Servers. Das AIT TeamSystemPro Team besteht aus Experten rund um Team Foundation Server und Application Lifecycle-Management.

Microsoft spricht [skrum]

Immer mehr Unternehmen setzen auf diesen bekanntesten Vertreter der agilen Softwareentwicklung. Ist Scrum einfach? Nein, Scrum ist nicht einfach. Den Grundsätzen von Scrum zu folgen, erfordert Mut, vollständige Transparenz und das Vertrauen darauf, Fehler machen zu dürfen. Es bedarf also eines Teams, das offen alle auftretenden Probleme systematisch der Reihe nach abarbeitet und daraus lernt, um Projekte erfolgreich mit Scrum zu realisieren. Die dafür notwendigen Werkzeuge und Methodiken sollten deshalb nicht schon die erste Hürde darstellen. Dieser Artikel zeigt eine exemplarische Einführung in Scrum-basiertes, werkzeuggestütztes Arbeiten.

SCRUM

Würde das Wort „Scrum“ im Duden stehen, so könnte der Eintrag wohl ungefähr wie folgt aussehen:

Scrum, das; bezeichnet ein schlankes Prozessmodell, das den agilen Prozessen zum Projektmanagement und der Softwareentwicklung zugeordnet wird.

Was zwar richtig, aber nicht annähernd ausreichend wäre. Es handelt sich also um ein schlankes und agiles Modell. Doch wie sieht das nun praktisch aus?

Die Maxime in Scrum erfüllen das Agile Manifest [agi]. Das bedeutet, dass die Kommunikation zwischen den Projektbeteiligten und die Projektbeteiligten als Individuen wichtiger sind, als die eingesetzten Werkzeuge. Scrum schreibt daher keine speziellen Werkzeuge oder gar Werkzeugketten vor. Was im Mittelpunkt steht, ist das Ergebnis, der erschaffene Wert. So liegt dem Product Owner am Ende einer jeden Iteration ein „Potential Shippable Increment“ – also ein potenziell auslieferbarer Teil des Produkts vor. Auch folgt Scrum der These, dass die andauernde enge Zusammenarbeit mit dem Kunden wichtiger ist, als anfänglich vertraglich beschriebene – und vielleicht bereits veraltete – Anforderungen. Änderungen in den Anforderungen werden als Teil des Projektes und nicht als Störung empfunden.

Der Prozess basiert auf drei Bereichen:

- Der erste Bereich bezieht sich auf eine klare, dreigeteilte Rollenverteilung. Der *Product Owner*, vertritt die Auftraggeberseite und definiert und priorisiert die Anforderungen. In der Rolle des Mentors befindet sich der *Scrum Master*. Er stellt den methodischen Kern des Prozesses dar und ist ein zertifizierter Fachmann in seinem Bereich.

Zu seinen Aufgaben gehören die Leitung und Einberufung der Planungsmeetings und die Sicherstellung der Einhaltung des Scrum-Prozesses. Das *Scrum-Team* umfasst 5-10 Mitglieder und setzt sich aus den verschiedenen Bereichen der Entwicklung zusammen – ein multifunktionales Team.

- Der zweite Teilbereich umfasst die Projektbesprechungen und spiegelt damit die Maxime der Kommunikation als

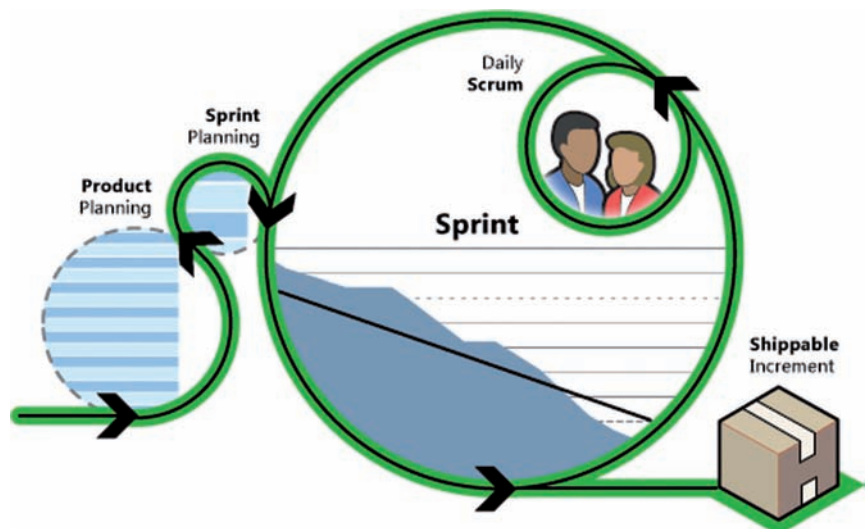


Abb. 1: Der typische Scrum-Ablauf

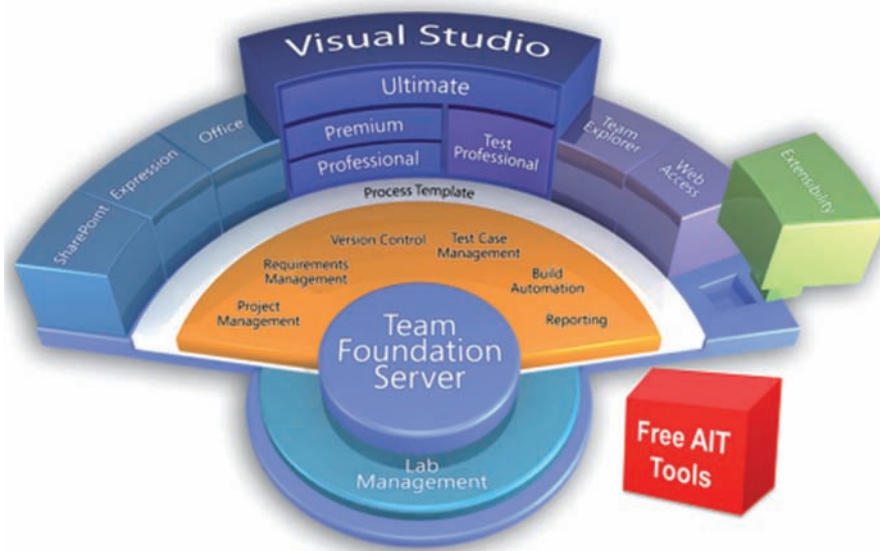


Abb. 2: Modularer Aufbau der TFS-Komponenten

Eckpfeiler in Scrum wider. Jeder Projektabschnitt besitzt ein speziell dafür vorgesehenes Besprechungskonzept, das die Abstimmung des Projektes und der Projektbeteiligten optimal gestaltet. Es ist die Aufgabe des Scrum-Masters diese einzuberufen und zu leiten (siehe auch [Abbildung 1](#)).

- Der dritte Bereich stellt das Rückgrat der Produkthanforderungen dar: die Backlogs. Aus dem Product Backlog, in dem der Product Owner seine Anforderungen sammelt und priorisiert, wählt das Team für jede Iteration – hier Sprint genannt – genau die Arbeitsmenge nach der Priorisierung des Product Owners, die es in dem kommenden Sprint umsetzen kann. Diese für den Sprint versprochenen Arbeitseinheiten werden, als „committed“ gekennzeichnet, im Sprint Backlog abgelegt und während des Sprints von den Teams umgesetzt.

Es ist ersichtlich, dass sich gerade aus der Rollenverteilung und dem eigenverantwortlichen Arbeiten der Teams ein Verantwortungsgefüge bildet, das ein hohes Maß an Transparenz und Disziplin von allen beteiligten Seiten erfordert. Fester Bestandteil der Planung und der Meetings ist das sogenannte Scrum Board, das die Backlog-Items physisch visualisiert.

Die Leiter vor der Hürde

Der Team Foundation Server (TFS) [tfs] ist die Application Lifecycle Management

(ALM) Plattform für Softwareentwicklung im Team von Microsoft. Über den TFS können Projekte geplant, erstellt und verwaltet werden. Der TFS 2010 beinhaltet unter anderem Komponenten für das Projekt- und Requirement-Management, Versionskontrolle, Reporting und Build Automatisierung und ist somit der Kern der teamorientierten Softwareentwicklung (siehe auch [Abbildung 2](#)).

Auf ihn kann mittels verschiedener Clients zugegriffen werden. Der bekannteste Vertreter ist wohl das Visual Studio, mit dem überwiegend die Entwickler ihre Codeänderungen versioniert auf dem Server ablegen oder Buildautomation steuern können. Aber auch der Softwarearchitekt, der Tester wie auch der Projektmanager greifen auf diese gemeinsame Datenbasis zu. Letzterer – der Manager – ist nicht an der Versionierung des Codes, sondern vielmehr an der Auswertung der Anforderungen und des Projektstatus interessiert, welche er über die integrierten Berichte komfortabel auswerten kann.

Erst die Verwendung einer solchen zentralen Plattform bei der Projektrealisierung ermöglicht das aktive Leben eines einheitlichen, strukturierten und vor allem funktionstüchtigen Prozessmodells. Um dies zu nutzen, unterstützt der TFS die Verwendung von Prozessvorlagen und wird bereits mit zwei Vorlagen aus dem Hause Microsoft ausgeliefert: zum einen der formale Ansatz, umgesetzt in der „MSF for CMMI“- Vorlage, zum anderen der agile Ansatz im „MSF for Agile Software Development“.

Der Nachwuchs ist da

Mit der jetzt erschienenen *Microsoft Visual Studio Scrum 1.0-Vorlage* [vsd] wurde dieses Portfolio nun um einen sehr schlanken und agilen kleinen Bruder erweitert. Wie der Name schon suggeriert, soll diese Prozessvorlage Anwendern helfen, die zu Beginn des Artikels beschriebenen Methoden und Schritte zu Scrum umzusetzen und sie so durch diesen Prozess zu führen. Die Produktivität und Eigenverantwortung der Teams stehen dabei weiterhin im Vordergrund.

Wie viel Scrum steckt drin?

Das Ziel, das mit der Scrum 1.0-Vorlage anvisiert wird, ist, den Scrum-Prozess auf dem TFS abzubilden und lebbar zu gestalten. Der durch die Prozessvorlage vorgegebene Rahmen muss daher leichtgewichtig sein und die Agilität des Teams unterstüt-

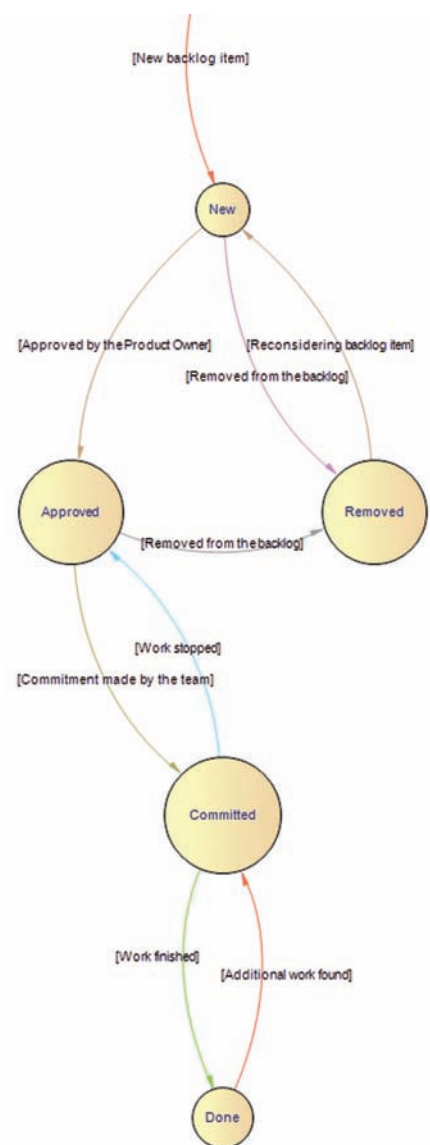


Abb. 3: Die PBI-Statusmaschine

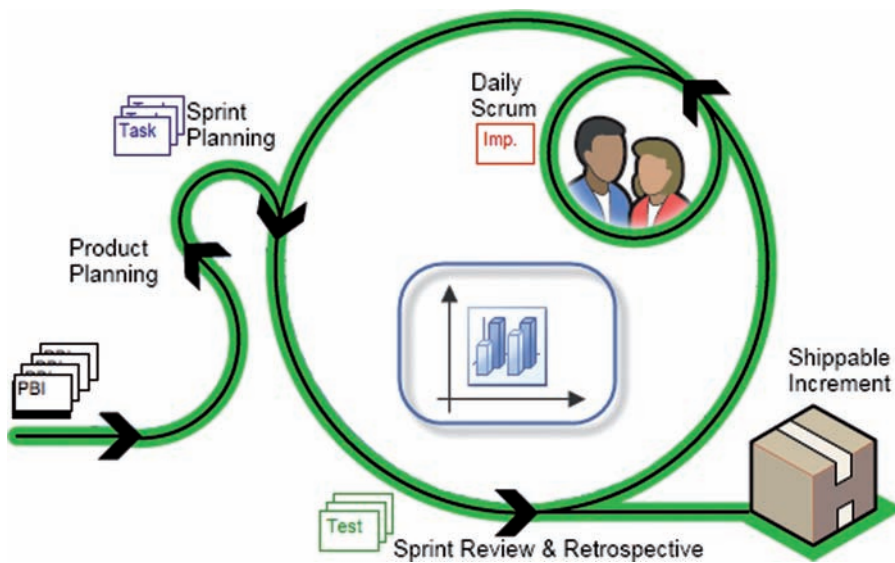


Abb. 4: Welches Meeting mit welchem Workitem?

zen und fördern. Es müssen zum Beispiel Work Items für das Backlog und die Tasks existieren, die durch die Scrum-typischen Phasen wie „In Progress“ oder „Done“ geführt werden. Auch ein Mechanismus zur Sprintplanung muss vorhanden sein. Und nicht zuletzt muss auch die Abarbeitung der Backlog Items, z. B. in Burn-Down-Charts, visualisiert werden können.

Schauen wir also einmal ob der neue, agile kleine Bruder halten kann, was er verspricht.

Zunächst einmal gilt es, die Informationselemente – die Work Item Typen, die das Backlog darstellen – zu betrachten. Mit den Typen Sprint, Product Backlog Item, Bug, Task, Impediment und Test Case existieren alle zur Organisation, Planung und Umsetzung der Sprints – und letztlich des Projektes – benötigten Elemente. Betrachten wir zum Beispiel die aufeinander folgenden Zustände des Tasks genauer, so erkennen wir den Ablauf von *todo* über *in progress* nach *done*.

Das deckt sich mit der Definition in Scrum. Dasselbe erkennen wir auch bei den Product Backlog Items, die vom Product Owner freigegeben (*approved*) und anschließend vom Team für einen Sprint angenommen, also *committed*, werden (siehe auch Abbildung 3).

Ärmel hoch – jetzt wird gescrumt!

Zu Beginn der Produktplanung nimmt der Product Owner die Anforderungen auf und

legt neue *Product Backlog Items (PBI)* an. Da er die meisten Anforderungen sowieso in Word organisiert hat, kann er diese durch Zusatztools wie *WordToTFS* [ait] bequem aus Word in den TFS übertragen. Er wählt nun durch Priorisierung seine

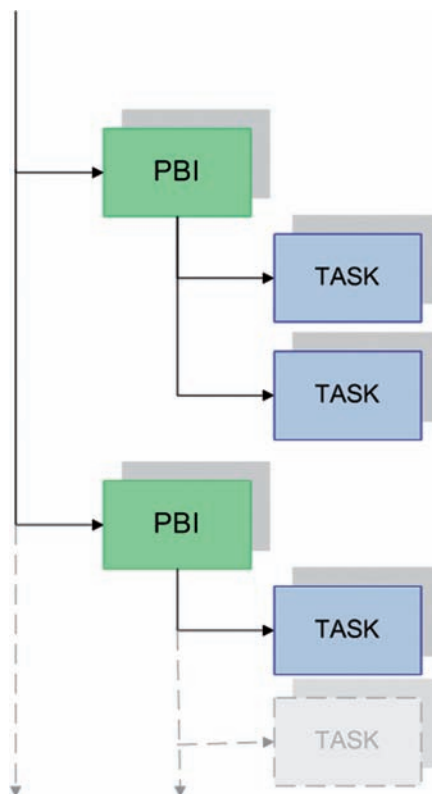


Abb. 5: Sprint Backlog-Hierarchie

Favoriten und setzt deren Status auf „*approved*“. So kann das jeweilige Backlog Item beim nächsten *Sprint* vom Team berücksichtigt werden.

Der Scrum Master plant in Absprache mit dem Product Owner den Sprint. Hierzu legt er den Start und das Ende des Sprints im Sprint Item fest und beruft das *Sprint Planning Meeting* ein (siehe auch Abbildung 4).

Das Team verständigt sich im ersten Teil des Sprint Planning Meetings auf das „Was“ für den nächsten Sprint. Es werden die Backlog-Items nach der Priorisierung des Product Owners ausgewählt, die in dem nächsten Sprint bearbeitet werden sollen.

Im zweiten Teil des Sprint Planning Meetings bespricht das Team das „Wie“. Hierbei werden die Teilaufgaben, die zur Realisierung notwendig sind, identifiziert und in *Tasks* manifestiert. Wichtig ist der Aspekt, dass ein Task mit dem jeweiligen Backlog-Item verknüpft wird, zu dem es gehört (siehe auch Abbildung 5). Hierzu wird der Team Explorer als Client zum TFS verwendet. Am Ende des Tages steht somit ein Aufgabenkatalog für den nachfolgenden Sprint – der *Sprint Backlog*.

Die einzelnen Tasks sind den jeweiligen Spezialisten des Teams zugeordnet, dem Designer, Architekten oder Entwickler, der diese priorisiert und während des Sprints von üblicherweise 2 bis 4 Wochen abarbeitet und seine aufgewendeten Zeiten gewissenhaft einträgt.

Durch diese täglichen Eintragungen werden während des Sprints Reports, sogenannte Burndown Charts generiert (siehe auch Abbildung 7). Sie geben Aufschluss über den Sprintstatus und darüber, ob das Sprintziel erreichbar ist oder eben nicht.

Zur Teamkommunikation dienen die täglich stattfindenden *Daily Scrum Meetings*. Hier wird in maximal 15 Minuten von jedem Teammitglied kommuniziert, was es gestern gemacht hat und was es heute machen wird.

Durch kostenfreie Zusatztools, wie dem *AIT Task Board* [aits], kann das typischerweise verwendete Scrumboard aus Post-its durch ein virtuelles Board ersetzt werden (siehe auch Abbildung 6). Es bietet einen schnellen und einfachen Überblick über alle aktiven Arbeiten und ermöglicht nicht nur ein interaktives Scrum-Meeting, denn es ist für jedes Prozess-template konfigurierbar, Scrum, CMMI, Agile, oder Ihr eigenes. Der

Vorteil liegt klar auf der Hand: Es können direkt die anstehenden Tasks visualisiert und modifiziert werden.

Beim Daily Scrum Meeting werden auch Impediments aufgenommen. Impediments sind Störungen, die auftreten und den Projektverlauf negativ beeinflussen. Es ist die Aufgabe des Scrum Masters, diese zu beheben oder in seltenen Fällen einen Sprintabbruch in die Wege zu leiten.

Auch das Impediment ist ein Work Item-Typ der Prozessvorlage. Am Ende des Sprints soll ein fertiges Inkrement stehen, d.h. es muss auch getestet werden. Mithilfe von Test Cases wird dieser Vorgang ebenfalls im TFS reproduzierbar dokumentiert.

Die in den Sprints und über die gesamte Projektdauer gesammelten Schätzwerte, Testergebnisse und Änderungsdaten sind durch die Verwendung einer einheitlichen Datenbasis – dem TFS – über Reports auswertbar und dienen als Referenzwert für die Planung des nächsten Sprints oder auch des nächsten Produktes. Schätzwerte, wie etwa über die Team Velocity (Arbeit je Zeit), können somit verfeinert und Aussagen sowie Prognosen über bevorstehende Projektabschnitte getroffen werden.

Durch die Scrum-typischen Burndown-Charts werden diese Daten grafisch aufbereitet und dem Team, dem Product Owner und nicht zuletzt dem Scrum Master präsentiert. Sie visualisieren den Projektfortschritt aufgrund der geschätzten restlichen Aufwendung und berechnen, ob das Sprint-Ziel über- oder untertroffen wird.

Fazit

Microsoft liefert mit dem TFS eine Plattform, die von Haus aus viele Prozessmodelle unterstützt. Jetzt wird mit der Scrum 1.0-Prozessvorlage auch SCRUM „out-of-the-box“ unterstützt. Microsoft hat damit den Projekverantwortlichen die Erweiterung an die Hand gegeben, auf die die Scrum-Fangemeinde schon lange gewartet hat. Die Vorlage ist mit ein wenig Scrum-Verständnis fast schon intuitiv anwendbar und könnte durch die Vielzahl der Planungs- und Auswertungsmöglichkeiten Euphorie unter den potenziellen Product Ownern auslösen. Sie täuscht fast schon darüber hinweg, dass es eben doch noch eines Scrum Masters und eines wahren Teams für einen erfolgreich gelebten Scrum Prozess bedarf. ■

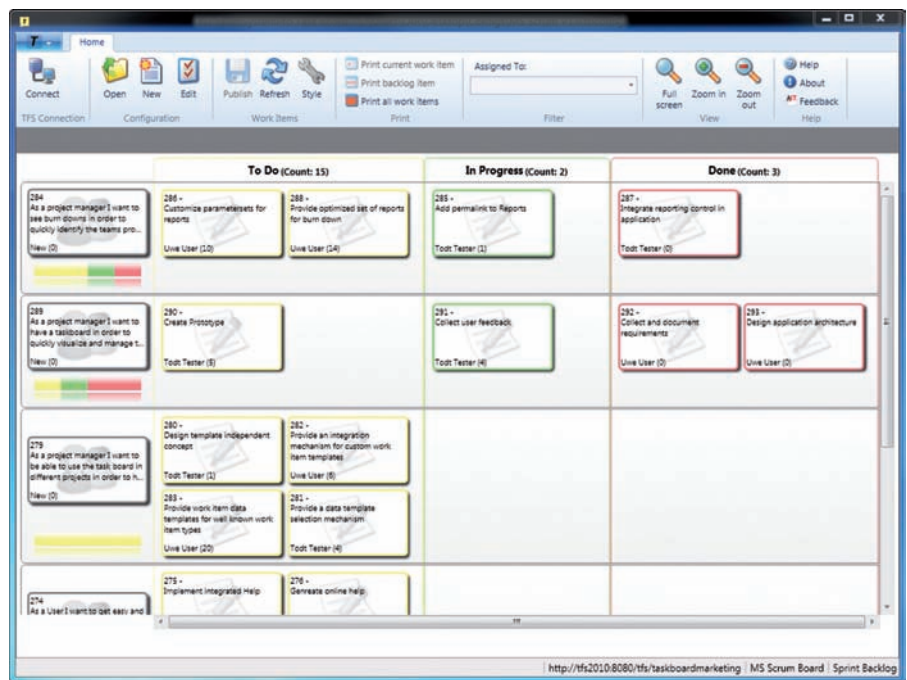
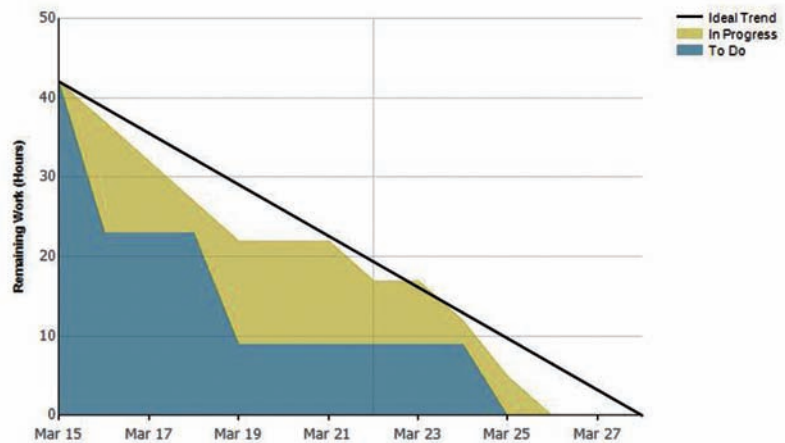


Abb. 6: Das Taskboard

Sprint Burndown

Indicates the team's progress towards completing its work for a sprint.



Questions This Report Helps Answer

- How much work remains in the sprint?
- Is the team on track to finish work for the sprint?
- When will the team finish work for the sprint?
- How much work for the sprint is in progress?

[How to Use This Report](#)

Parameter Values

Sprint:	Release 5/Sprint 6
Area:	All (No Filter)
Start Date:	03/15/2010
Finish Date:	03/28/2010

Generated: 5/21/2010 8:07:19 AM

By: REDMOND\abjork

Data Updated: 5/21/2010 7:35:50 AM

Abb. 7: Der Sprint Burndown Chart

Referenzen

- [agi] <http://agilemanifesto.org/>
- [tfs] <http://www.microsoft.com/germany/visualstudio/products/team/visual-studio-team-foundation-server.aspx>
- [vsd] <http://visualstudiogallery.msdn.microsoft.com/de-de/59ac03e3-df99-4776-be39-1917cbfc5d8e>
- [ait] http://www.aitag.com/word_to_tfs0.0.html
- [aits] http://www.aitag.com/ait_scrum_taskboard.0.html