entwickler.press

TFS Jumpstart

Per Express zum Application Lifecycle Management

Tobias Richling, Michael Klei

2. aktualisierte Auflage

Tobias Richling, Michael Klei

TFS Jumpstart

Per Express zum Application Lifecycle Management
aktualisierte 2. Auflage

entwickler.press

Tobias Richling, Michael Klei
TFS Jumpstart. Per Express zum Application Lifecycle Management
aktualisierte 2. Auflage

ISBN: 978-3-86802-356-5

© 2017 entwickler.press
Ein Imprint der Software & Support Media GmbH

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Ihr Kontakt zum Verlag und Lektorat:
Software & Support Media GmbH
entwickler.press
Schwedlerstraße 8
60314 Frankfurt am Main
Tel.: +49 (0)69 630089-0
Fax: +49 (0)69 630089-89
lektorat@entwickler-press.de
http://www.entwickler-press.de

Lektorat und Korrektorat: Björn Bohn, Martina Raschke
Copy-Editor: Nicole Bechtel
Satz: Dominique Kalbassi
Umschlaggestaltung: Maria Rudi
Titelbild: © filo | istockphoto.com

Belichtung, Druck & Bindung: Media-Print Informationstechnologie GmbH, Paderborn

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktion jeglicher Art (Fotokopie, Nachdruck, Mikrofilm, Erfassung auf elektronischen Datenträgern oder anderen Verfahren) nur mit schriftlicher Genehmigung des Verlags. Jegliche Haftung für die Richtigkeit des gesamten Werks kann, trotz sorgfältiger Prüfung durch Autor und Verlag, nicht übernommen werden. Die im Buch genannten Produkte, Warenzeichen und Firmennamen sind in der Regel durch deren Inhaber geschützt.

Inhaltsverzeichnis

Vorwort

Arbeiten in kurzen Zyklen, schnelles Liefern neuer Releases, dabei volle Transparenz und Nachvollziehbarkeit sowie eine hoher Grad an Automatisierung im Entwicklungsprozess – das sind Standards in der heutigen Softwareentwicklung. Microsoft bietet mit seinem Application-Lifecycle-Management-System, Team Foundation Server (TFS) und Visual Studio Team Services (VSTS), eine umfangreiche Plattform, die Teams bei diesen Aufgaben hervorragend unterstützt.

Nachdem Microsoft – zumindest im Bereich der Entwicklungswerkzeuge – selbst einen Wandel hin zu einer agilen Organisation durchlebt hat und diese Werkzeugkette kontinuierlich verbessert, kann man auch als Nutzer des Systems stets von Weiterentwicklungen profitieren. Quellen dafür gibt es viele, seien es die MSDN selbst, englischsprachige Bücher oder diverse Blogs.

Doch insbesondere deutschsprachige Literatur ist in diesem Bereich vergleichsweise wenig vertreten. Ich und meine Kollegen bei der AIT GmbH finden, Communityarbeit macht nicht nur Spaß, sondern ist ein wichtiger Bestandteil des gemeinsamen kontinuierlichen Lernens. Die Weitergabe der eigenen Erfahrungen an die Community ist daher äußerst wertvoll und dies kann über verschiedene Kanäle geschehen, neben sehr informellen User Groups oder Meet-ups über Konferenzen, aber auch Blogs, Webcasts, Fachartikel und Bücher. Mit unserem deutschsprachigen Blog unter *http://blog.aitgmbh.de* leisten wir hierzu unseren Beitrag.

Deshalb freut es mich besonders, dass die Autoren Tobias Richling und Michael Klei mit der zweiten Auflage Ihres Werks über den TFS einen weiteren wertvollen Beitrag zur deutschsprachigen Fachliteratur für Application Lifecycle Management (ALM) in der Microsoft-Technologiewelt leisten. Der TFS und VSTS erfreuen sich großer Verbreitung und haben einen beachtlichen Reifegrad erreicht – und das nicht ohne Grund.

Der TFS hat in den letzten Jahren eine rasante Entwicklung hingelegt. Nachdem die ersten Versionen 2005 und 2008 noch sehr rudimentär waren, eine solide, zentrale Versionskontrolle und ein einfaches Work-Item-System geboten hatten, ging es seit 2010 in großen Schritten in Richtung einer praxisnahen und integrierten DevOps-Werkzeugkette. Mit den Versionen 2010, 2012, 2013, 2015 und nicht zuletzt 2017 wurden ganz neue Disziplinen im Entwicklungsprozess erschlossen und abgedeckt. So wurden nicht nur das Test-Management und eine sehr gute Deployment-Unterstützung in Form von Release-Management eingeführt, sondern auch auf wichtige Trends in der Softwareentwicklung reagiert. Die zusätzliche Unterstützung und volle Integration von Git als Versionskontrolle, ein komplett neues und modernes Build-System sowie die Öffnung der gesamten Plattform für die Community sind dabei wesentliche Meilensteine.

Das vorliegende Buch bietet Neulingen in dem Bereich einen guten Einstieg und ist den Experten ein gutes Nachschlagewerk. Schon das einfache Durchblättern und Reinschnuppern macht Spaß, da die Autoren immer wieder mit anschaulichen Darstellungen arbeiten, die dazu einladen, sich mit einem Themenkomplex näher zu beschäftigen.

Die Kapitelstruktur erlaubt es, gezielt zu einzelnen Themenbereichen zu springen, die einen im Alltag gerade interessieren. So startet das Buch mit einem ersten Blick auf das gesamte System und ist daher besonders geeignet für jemanden, der noch keine Berührungspunkte damit hatte, sich einen ersten Eindruck zu verschaffen. Doch auch für die Administration bietet dieses Kapitel wertvolle Tipps.

Im weiteren Verlauf stützt sich die Struktur auf die wesentlichen Subsysteme des TFS. Der Umgang mit dem Work-Item-System wird ausführlich beschrieben, gefolgt von den beiden Versionsverwaltungen Git und Team Foundation Version Control. Auch dem alten und neuen Build-System wird ein eigenes Kapitel gewidmet. Für alle, die danach noch weiter eintauchen wollen, bietet das letzte Kapitel einen gelungenen Abschluss. Hier wird das TFS API sehr praxisnah und anhand verschiedener Beispiele erklärt.

Außerdem sind im Buch viele Screenshots, schematische Abbildungen und auch Codebeispiele zu sehen. Diese illustrieren die technischen Zusammenhänge sehr schön und ermöglichen das Erarbeiten eines Sachverhalts, auch wenn man nicht gerade eine komplette Umgebung vor sich hat, sondern sich ein paar Seiten gemütlich am Sonntagnachmittag auf der Couch zu Gemüte führt.

Thomas Rümmler
MVP Visual Studio and Development Technologies
AIT GmbH & Co. KG

1 Einleitung

Bereits seit Anfang 2013 kann man auf verschiedenen Wegen an eine kostenlose Ausgabe von Microsofts ALM-Plattform, dem Team Foundation Server (TFS), gelangen. Die erste mögliche Option ist eine vor Ort (On-Premises) zu installierende Version des TFS Express, die sich namentlich und technisch in die Reihe der übrigen Express-Produkte, zum Beispiel SQL Server, stellt. Die zweite Variante ist die Nutzung der Visual Studio Team Services in der Cloud. Die Installation eines TFS Express ist schnell getan, und für die Cloud fallen überhaupt keine Aufwände für ein Set-up an. Damit hat Microsoft dem TFS seine schlimmsten Stacheln gezogen: die hohen Systemanforderungen, die aufwendige Installation und die hohen Lizenzkosten.

In den letzten Jahren hat sich die Welt rund um den TFS weiterentwickelt, und viele dieser Entwicklungen sind auch am TFS nicht spurlos vorübergegangen. Zwei der bedeutendsten Änderungen sind die Zunahme der Popularität von Git und die breite Akzeptanz von cloud-basierten Systemen. Diese beiden Entwicklungen haben den TFS geprägt, denn nun bietet der TFS eine vollwertige Integration von Git. Auch das Build-System ist, der aktuellen Entwicklung folgend, deutlich verschlankt worden und unterstützt nun sowohl die Windows-Plattform mittels PowerShell-Skripten als auch den Rest der Welt mittels Node.js. Man hat dazugelernt in Redmond und erfindet das Rad nicht neu, sondern bindet bestehende Lösungen ein. Diese neuen Entwicklungen kann man am besten mit der cloud-basierten Version des TFS verfolgen. Die On-Premises-Variante hinkt immer ein bisschen hinterher – das gilt leider insbesondere auch für die kostenlose Express-Variante. Es bleibt zu hoffen, dass die Sicherheitsbedenken, die viele Leute in Bezug auf ein Versionskontrollsystem in der Cloud plagen, durch Rechenzentren in Deutschland und andere vertrauensbildende Maßnahmen gemindert werden können.

Es gibt also neue, einfache Möglichkeiten, sich mit dem TFS zu befassen und ihn als Plattform für das Application Lifecycle Management einzusetzen. Die beiden häufigsten Ausreden gelten nicht mehr! Und der Einstieg ist in der Tat nicht schwer, wie schon der Umfang dieses Buchs beweist.

Also dann: Nichts wie ran an den TFS! Auch wenn die Cloud-Version noch nichts für den beruflichen Alltag ist: Privates Probieren kostet nichts (ehrlich!). Vielleicht kann man ja mit dem einen oder anderen tollen Feature einem Kollegen oder dem Chef den Mund wässrig machen. Wir wünschen Ihnen viel Spaß beim Entdecken der vielen tollen Features des TFS und hoffen, dass dieses Buch Sie auf Ihrer Rundreise durch den TFS mit vielen hilfreichen Tipps unterstützen kann.

1.1 Für wen ist dieses Buch?

Das Buch eignet sich hervorragend für Einsteiger in den Bereich ALM mit dem TFS. Wenn Sie bisher noch keine oder nur wenig Erfahrung mit Quellcodeverwaltung und Co. haben und den Einstieg in den TFS wagen wollen, halten Sie das richtige Buch in den Händen. Das Buch beschreibt die wesentlichen Bereiche des ALM-Prozesses und beginnt jeweils mit einer Erklärung zur Nutzung der entsprechenden Funktionen im Sinne eines Benutzerhandbuchs.

Sie sind auf der Suche nach einem schnellen Überblick über die Kernfeatures des TFS? Dann ist insbesondere das erste Kapitel das richtige für Sie. Hier machen wir einen Rundflug über den TFS. Jedes weitere Kapitel bietet zunehmend tiefere Einblicke in die einzelnen Bereiche.

Wenn Sie den TFS bereits nutzen, zum Beispiel zur Quellcodeverwaltung, kann dieses Buch Ihnen helfen, die weiteren Potenziale des TFS zu erschließen. Jedes Kapitel enthält neben der Erklärung der grundlegenden Bedienung erweiterte Beispiele zur Anpassung des TFS. Hier kann auch für erfahrene Nutzer des TFS noch der ein oder andere hilfreiche Tipp versteckt sein.

Auch wenn Sie den TFS schon lange und in allen Bereichen nutzen, kann dieses Buch Ihnen noch von Nutzen sein. Wenn die Anpassbarkeit des TFS an ihre Grenzen stößt, eröffnet das TFS-API neue Wege. In einem eigenen Kapitel werden Beispiele für die Automatisierung des TFS unter Zuhilfenahme des mächtigen API beschrieben. Hierbei werden verschiedene technologische Aspekte des TFS-Ökosystems angesprochen.

Wenn Sie in Ihrem Unternehmen eine große TFS-Installation verwenden, einschließlich Warehouse, SharePoint und Co., und Sie nach Lösungen dieser Größenordnung suchen, wird dieses Buch Ihnen nicht viel weiterhelfen. Auch für Administratoren hat das Buch nur wenig zu bieten. Zwar wird die grundlegende Administration des Servers beschrieben, der Fokus liegt jedoch auf den funktionalen Bereichen.

1.2 Aufbau und Inhalt

Der Schwerpunkt dieses Buchs liegt auf der Cloud-Variante, den Visual Studio Team Services, und dem TFS Express. Das Buch beschränkt sich auf eine Teilmenge der Features vom TFS, die zum Großteil von diesen beiden Produktvarianten abgedeckt werden. Viele Features der Vollversion des TFS fallen damit aus diesem Buch heraus. Dazu gehören zum Beispiel die SharePoint-Integration, das Data Warehouse und das Test Lab. Dadurch bleibt das Buch schlank und auf das Wesentliche fokussiert: den Kern des ALM-Prozesses. Dieser besteht aus drei Säulen: der Anforderungsanalyse, die Quellcode- und Versionsverwaltung sowie Builds. Im Wesentlichen folgt das Buch dieser Roadmap, indem jedes Kapitel einer dieser Säulen gewidmet ist.

Seit dem ersten Erscheinen dieses Buchs 2013 hat sich jedoch im TFS viel getan, vorrangig im Bereich Git und der Adaption der Cloud, und diese Entwicklungen sind massiv in die

Struktur des Buchs mit eingeflossen. Außerdem wurde das Feedback, das wir von Lesern erhalten haben, berücksichtigt. Im Bereich der Work Items wird stärkeres Augenmerk auf die Features des Cloud-TFS gelegt, da diese denen des On-Premises-TFS weit überlegen sind. Die Säule der Versionskontrolle ist in zwei Kapitel zerfallen: ein Kapitel zum Thema Git und ein Kapitel zur Team-Foundation-eigenen Versionskontrolle. Aufgrund der steigenden Popularität von Git erhält dieser Kontrahent hier in der Reihenfolge den Vorzug. Das Kapitel zum Thema Build deckt das alte und das neue Build-System ab. Die Ausführungen zum alten Build-System wurden dabei deutlich gestutzt. Die Ausführungen zum neuen Build-System sind aufgrund von dessen vereinfachter Struktur so kompakt, dass beides zusammen noch in ein Kapitel passt. Die Fallstudien und Erläuterungen zum API, die vorher Teil der einzelnen Kapitel waren, wurden in ein eigenes Kapitel ausgelagert. Alle Beispiele beziehen sich auf das REST-API des TFS und sind so aufgebaut, dass sie auf jeden Fall in der Cloud-Version des TFS funktionieren. Auf diese Weise bleiben die Kernkapitel fokussierter und man findet an einer Stelle zusammenhängend alles zum Thema API.

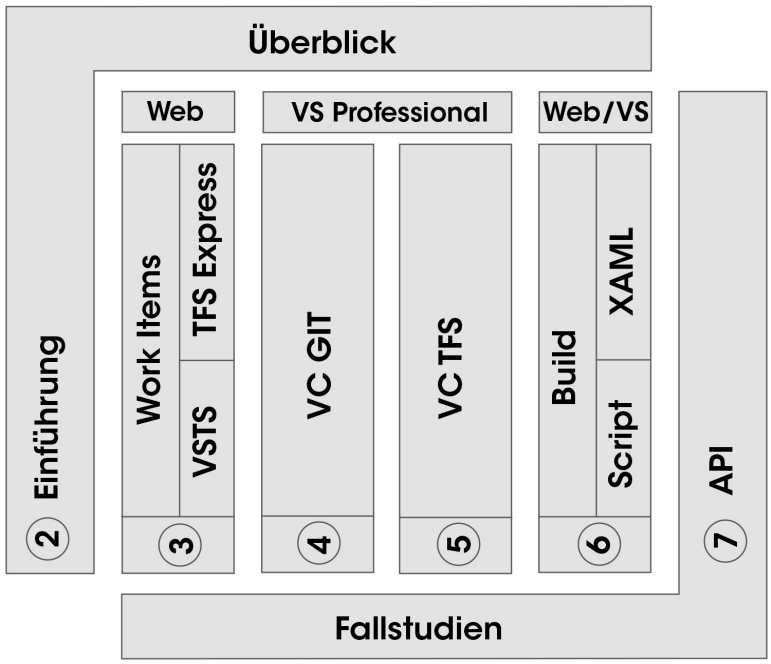

Abbildung 1.1: Aufbau des Buchs

Die einzelnen Kapitel enthalten zusammengefasst folgenden Inhalt:

- Kapitel 2 – Überblick und Installation/Administration: Auch wenn die Installation einfach geworden ist, wird hier noch einmal erklärt, wie es geht. Es werden die Varianten „TFS-Express-Installation“ und „Anmeldung bei den Visual Studio Team Services“ be-

schrieben. Außerdem enthält das Kapitel einen Schnelldurchlauf durch alle weiteren Themen des Buchs und bietet somit einen optimalen Überblick über das Produkt.

- Kapitel 3 – Work Items und Prozessvorlagen: Hier wird kurz der Scrum-Prozess umrissen, der vom TFS neben anderen Prozessen unterstützt wird. Danach wird beschrieben, wie sich Scrum in der Anforderungsanalyse und in der Aufgabenbearbeitung im TFS manifestiert. Nennenswert sind dabei insbesondere die agilen Planungstools und das Taskboard. Falls die vorhandenen Work Items Ihren Anforderungen nicht genügen, wird beschrieben, wie Work Items erweitert und angepasst werden können.
- Kapitel 4 – Git-Versionskontrolle: Das Kapitel startet mit einer Einführung in Git anhand der Kommandozeile, um einen unverstellten Blick auf die Konzepte in Git zu ermöglichen. Im Folgenden wird die Nutzung von Git aus dem Visual Studio heraus beschrieben. Hierbei wird besonders auf Umsteiger von TFVC auf Git eingegangen. Damit ist dieses Kapitel kein Ersatz für ein Buch über Git, sondern soll Umsteigern den Einstieg erleichtern. Es wird beschrieben, wie man einige der speziellen Features der TFVC in Git abbilden kann.
- Kapitel 5 – TFS-Versionskontrolle: Dieses Kapitel beschreibt den Umgang mit der Quellcode- und Versionsverwaltung im TFS. Zunächst werden die grundlegenden Mechanismen, wie das Einchecken von Code, erläutert. Im weiteren Verlauf werden insbesondere Features wie das Unterbrechen von Aufgaben und Codereviews beschrieben.
- Kapitel 6 – Team Build: Dieser Abschnitt stellt die Möglichkeiten der automatisierten Erstellung von Code des TFS dar. Das Kapitel zerfällt in zwei identisch strukturierte Teile für das neue skriptbasierte Build-System und das alte auf XAML-Workflows basierende System. Beginnend mit der Architektur wird erläutert, wie ein automatisierter Build aufgesetzt und ausgeführt werden kann und wie sich der vorgegebene Workflow verändert.
- Kapitel 7 – API und Fallstudien: Hier werden die drei Säulen Work Items, Versionskontrolle und Build aus Sicht des REST-API des TFS betrachtet. Es wird gezeigt, wie man API-Aufrufe authentifiziert und welchen gemeinsamen Aufbau alle Aufrufe haben. Die Struktur der APIs für die drei Bereiche wird beschrieben, ebenso wie übliche Aufgaben, die man über das API erledigen kann. Von diesen werden dann einige anhand von Beispielen vorgestellt. Fallstudien über die Anpassung von Work Items, das Reagieren auf TFS-Ereignisse sowie das Erstellen eigener Build-Tasks runden das Kapitel ab.

Wo immer es möglich war, haben wir in diesem Buch die deutsche Version von Visual Studio und des TFS verwendet, und damit auf ein deutliches Leserfeedback reagiert. Erfreulicherweise hat sich die Qualität der Übersetzung im Laufe der Jahre aber auch deutlich verbessert. Der Cloud-TFS steht jedoch nicht in einer deutschen Version zur Verfügung, daher bleibt es hierbei beim Englischen.

1.3 Danksagungen

Ein Buch schreibt man ja nie allein. Daher möchten wir uns bei all jenen bedanken, die bei der Entstehung des Buchs geholfen haben. Dazu gehören die Mitarbeiter des entwickler.press-Verlags, die unter der Anleitung von Lektorin Martina Raschke akribisch sämtliche Fehler aus dem Manuskript entfernt haben.

Ein weiterer Dank geht an meine Frau Judith, die durch ihre Geduld und ihre Ermutigung die Entstehung dieser zweiten Auflage des Buchs begünstigt hat, auch wenn es sich gegenüber der ursprünglichen Planung zeitlich mal ein bisschen verschoben hat – was natürlich heißt, dass es länger dauerte, als geplant. Vielen Dank für die vielen Stunden Kinder hüten und die Abende, die du ohne meine Gesellschaft verbringen musstest, weil ich mit Buchschreiben beschäftigt war. Ohne diese Unterstützung hätte ich weder die Lust noch die Zeit gehabt, dieses Projekt fertig zu stellen. Du bist die Beste, finde ich – aber das weißt du ja bereits. Außerdem haben noch viele Freunde und Bekannte das Buch gelesen und uns wertvolle Hinweise gegeben. Hierzu gehört insbesondere Thomas Rümmler von der AIT GmbH, den wir für das Vorwort gewinnen konnten.

Wir haben uns bemüht, ein sinnvolles Buch mit logischer Struktur und relevanten Beispielen zu schreiben, das den Leser nicht mit einer Masse an Seiten erschlägt. Dennoch kann es vorkommen, dass Ihnen etwas nicht ganz klar wird. Oder Sie finden, man hätte das ein oder andere anders niederschreiben sollen? Sie finden das Buch einfach nur super? Dann freuen wir uns, von Ihnen zu hören, sowohl für Kritik, Anregungen als auch Lob. Dazu erreichen Sie uns unter *tobias.richling@thinkexception.net* und *mklei@meta-objects.net*. Oder Sie schauen mal auf Twitter vorbei und folgen @trichling oder Sie werfen einen Blick auf den Blog *http://thinkexception.blogspot.de/*.

Wir wünschen viel Spaß beim Lesen!

Tobias Richling

Michael Klei

Februar 2017

2 Überblick und Installation / Administration

Die Installation des TFS ist über die vergangenen Produktversionen immer einfacher geworden. Was früher wie ein epischer Marathon angemutet hat, stellt sich heute als einfache assistentengeführte Installation dar. Es gibt aber immer noch, beziehungsweise seit neustem, verschiedenen Installationsvarianten mit eigenen Vor- und Nachteilen. In diesem Kapitel

- werden die verschiedenen Installationsvarianten des TFS vorgestellt
- wird ein kurzer Rundgang durch alle Themenbereiche des Buchs gemacht
- wird eine grobe Übersicht über die Konfigurationsmöglichkeiten des TFS gegeben

2.1 Die Wege zum TFS

2.1.1 Das Visual-Studio-Ökosystem

Seit der Version 2012 des TFS hat sich rund um das ALM-Flaggschiff von Microsoft vieles verändert. Die deutlichste Änderung ist, dass nun alle Entwicklerprodukte für alle Plattformen unter dem Namen Visual Studio zusammengefasst sind – dies zeigt sich vor allem an der Startseite für all diese Produkte: Unter *www.visualstudio.com* findet man den Einstiegspunkt zu allen Visual-Studio- und TFS-Produktlinien.

Die Produktlinien im Bereich Visual Studio sind nach dem Prinzip „aus Zwei mach Eins" verschlankt worden.[1] Microsoft hat die Professional- und Premium-Produktlinien zu einer einzelnen Professional-Version verschmolzen. Anstelle der Ultimate-Edition gibt es nun Visual Studio Enterprise. Alle Produkte sind neben dem Erwerb einer normalen Lizenz auch als jährliches oder monatliches Abonnement lizenzierbar. Im Vergleich kostet Visual Studio Professional als eigenständige Lizenz 636 €, ein Monatsabo 38 € pro Monat und ein Jahresabo 456 €, was genau den zwölf Monatsraten à 38 € entspricht. Sparen kann man also über die jährliche Zahlungsweise nicht. Die Enterprise-Edition ist als eigenständige Lizenz nur im Rahmen eines MSDN-Abonnements verfügbar und kostet dann 7 634 €. In der monatlichen Abovariante kostet die Version 210 €, das macht 2 530 € pro Jahr.

Wer nicht so tief in die Tasche greifen möchte, für den gibt es mittlerweile bereits mehrere Lösungen. Zum einen wären das nach wie vor die Express-Editionen von Visual Studio. Hier gibt es einzelne Produkte für Universal-Apps, Desktop- und Webentwicklung. Diese

1 *http://www.microsoft.com/visualstudio/deu/products/compare*

leicht verwirrende Produktgestaltung wurde durch die Visual Studio Community Edition verschlankt: Hierbei handelt es sich funktional um eine Professional-Edition, die mit lizenzrechtlichen Einschränkungen verbunden ist. Somit hat man ein kostenfreies Visual Studio zur Verfügung, mit dem man für alle Plattformen und Sprachen entwickeln kann.

Wer es gerne noch etwas schlanker hätte, der kann noch auf das neue Visual Studio Code zurückgreifen.[2] Hierbei handelt es sich um eine überwiegend textgesteuerte IDE, die speziell für Entwicklung auf anderen Plattformen entstanden ist. In dieser Version wurde auf sämtliche grafischen Tools und Assistenten verzichtet. Übrig geblieben ist ein überdurchschnittlich mächtiger Texteditor mit integrierten Debuggingtools, IntelliSense, Git-Integration und vielem mehr. Dabei hat Visual Studio Code einen klaren Fokus auf die Entwicklung von modernen Webapplikationen mit ASP.NET Core und Node. Es integriert sich gut in einen Entwicklungsworkflow, der neben einem leistungsfähigen Editor noch auf Kommandozeilentools wie npm, yo und Konsorten aufbaut.

Die zweite Säule des Visual-Studio-Ökosystems bildet eben der Team Foundation Server. Dieser wird zum einen als Version zum selbst installieren angeboten, zum anderen kann man ihn als gehosteten Dienst in der Cloud nutzen, dann hört er auf den Namen Visual Studio Team Services.

Wer seinen TFS in den eigenen vier Wänden betreiben möchte, hat die Wahl zwischen der vollwertigen Lizenz oder der kostenlosen TFS-Express-Variante. Der TFS selbst schlägt mit rund 636 € zu Buche, jede weitere Zugriffslizenz (CAL) kostet nochmal das gleiche. Die Kosten der gehosteten Cloud-Variante richten sich nach der Anzahl der Nutzer und der genutzten Dienste.

Für kleine Teams bietet Microsoft seit der Version 2012 nun auch für den TFS eine kostenlose Express-Version an. Diese unterliegt, wie auch andere Vertreter der Express-Familie, diversen Einschränkungen gegenüber der Vollversion.

Nachfolgend werden die drei Möglichkeiten, in den Genuss der Nutzung des TFS zu kommen, einmal kurz vorgestellt. Den Anfang macht die On-Premise-Vollversion, die im Folgenden keinen weiteren Raum erhält. Ihr gegenüber steht die Express-Variante, deren Installation ebenfalls vorgestellt werden soll. Den Abschluss bildet die Cloud-Variante.

2.1.2 TFS On-Premise

Diese Installationsvariante ist wohlbekannt, bezeichnet sie doch die bisherige TFS-Installation „vor Ort“. Obwohl die grundlegende Installation auch einfacher geworden ist, hat man hier so viele architektonische Möglichkeiten, dass das Aufsetzen eines Servers schon wieder abschrecken kann.

Die Möglichkeiten reichen von „alles auf einer Maschine“ wie beim TFS Express bis hin zu großen Deployments mit eigenem Application Server, Datenbankserver, SharePoint

2 *https://code.visualstudio.com/*

und Data Warehouse. Alle Installationsmöglichkeiten zu beschreiben wäre ermüdend und würde auch klar den Rahmen sprengen, daher konzentriert sich das Buch auf die Möglichkeiten, die sich mit der Express- und der Service-Variante ergeben.

2.1.3 TFS Express

Im Frühjahr 2012 hat Microsoft zum ersten Mal den TFS Express angekündigt. Seitdem hat es eine Aktualisierung des Produktes gegeben. Zum Zeitpunkt der Drucklegung war dies die Version 2015 Update 1. Die aktuelle Version kann im Netz heruntergeladen werden.[3] Und was steckt dahinter? Ganz im Sinne der übrigen Produkte der Express-Serie von Microsoft zunächst einmal, dass es kostenlos ist, dafür aber einigen Einschränkungen unterliegt. Der TFS Express ist von den Hardwareanforderungen so moderat, dass er sich problemlos auf einem Laptop oder einem normalen Desktop-PC installieren lässt. Er begnügt sich mit allen aktuellen Clientbetriebssystemen von Windows 7 bis Windows 10. Eine Installation auf einem Serverbetriebssystem ist natürlich ebenfalls möglich.[4]

Im Lieferumfang des TFS Express sind die wesentlichen Säulen des Application-Lifecycle-Management-Prozesses enthalten: die Erfassung und Verwaltung von Work Items zur Anforderungsanalyse und Fehlerverfolgung, die Versionskontrolle zur Verwaltung des Quellcodes und Team Builds zum automatischen Erstellen der Software und für Continuous Integration. Bis zu fünf Benutzer können den Server nutzen. Sollte das Team größer werden, so besteht die Möglichkeit, weitere Lizenzen in Form von CALs (Client Access Licences) zu erwerben. Alternativ kann man, ohne Datenverlust befürchten zu müssen, auf eine größere Edition des TFS umsteigen. Der kleine Bruder des TFS arbeitet nur mit dem kleinen Bruder des SQL Server zusammen und ist bezüglich der Datenbank dessen Beschränkungen unterworfen.

Die Installation läuft in zwei Phasen ab. In der ersten Phase, die auch die meiste Zeit in Anspruch nimmt, werden alle Dateien vom Installationsmedium auf die lokale Festplatte kopiert. Hier wählt man auch den Installationsordner aus, standardmäßig ist das *C:\Programme\Microsoft Team Foundation Server 14.0*. Es werden ca. 2 GB Festplattenspeicher benötigt.

Die zweite Phase ist die Konfiguration der gewünschten Features. Diese wird von einem Assistenten geleitet. Auf der ersten Ebene wählt man die Art der Installation aus. Für die Erstinstallation wählt man Team-Foundation-Anwendungsserver konfigurieren: Dadurch wird der TFS „komplett“, also auch inklusive des SQL Server, installiert. Möchte man eine bestehende Installation erweitern, so kann man Nur für Logikschicht auswählen: Die Kerndienste des TFS werden über Webseiten/Webdienste bereitgestellt. Für Szenarien mit hoher Verfügbarkeit ermöglicht es diese Option, diese Dienste über einen Lastenausgleich zur Verfügung zu stellen. Auch wenn man den TFS auf einen anderen Server verschieben möchte oder eine Notfallwiederherstellung durchführen muss, kann man diese Option auswählen. Hat man bereits eine bestehende Installation und möchte

3 *http://www.microsoft.com/visualstudio/eng/downloads#d-team-foundation-server-express*
4 *https://msdn.microsoft.com/library/vs/alm/tfs/administer/requirements#Clientcompatibility*

auf die neue Version aktualisieren, so ist die Option Upgrade die richtige Wahl. Der Bereich der Team Builds wurde in der Version 2015 komplett neu gestaltet und basiert nun nicht mehr auf der Workflow Foundation. Die „alten" XAML-Builds werden aber noch unterstützt. Wer sie nutzen möchte, muss im Zuge der Installation die Option XAML-Builddienst konfigurieren auswählen.

Nachfolgend wird der Prozess der vollständigen Installation beschrieben. Auf der Willkommensseite teilt der Assistent mit, dass der TFS unter dem lokalen Dienstkonto installiert und auf Port 8080 bereitgestellt wird. Es wird festgestellt, ob bereits ein SQL Server installiert ist. Wenn ja, wird dieser verwendet, andernfalls wird er mit installiert. Danach wird geprüft, ob alle Systemanforderungen erfüllt sind. Sollte hier mal etwas schieflaufen, wartet der Installer mit erstaunlich präzisen Meldungen darüber auf, wie der Fehler zu beheben ist. Sind alle Tests bestanden, geht es über den Button Konfigurieren zur eigentlichen Installation (Abbildung 2.1).

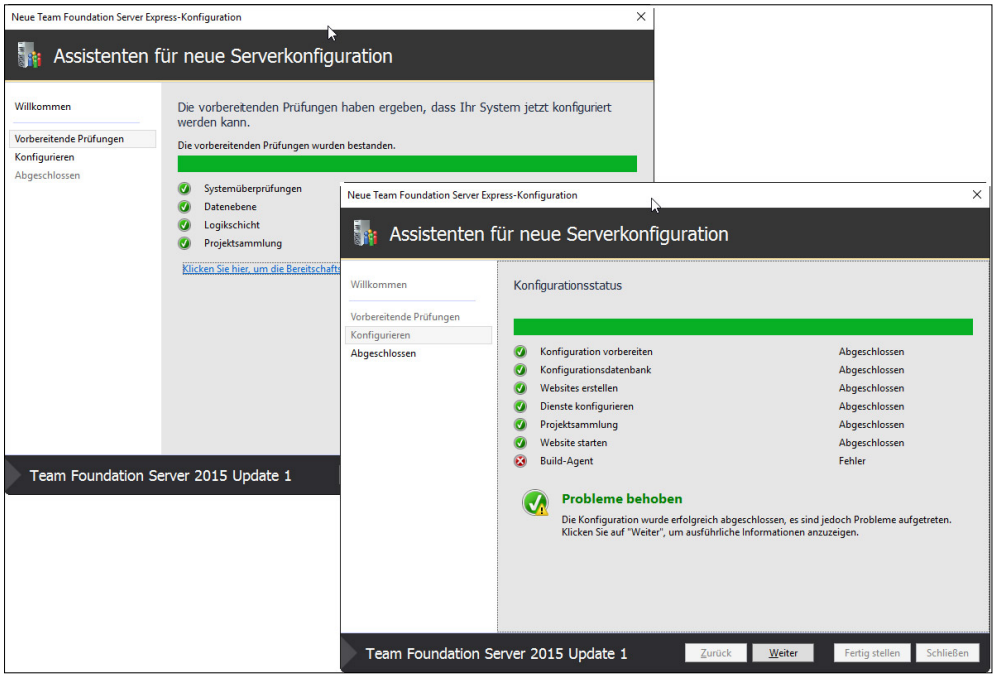

Abbildung 2.1: Installation des TFS Express

Nachdem das Set-up seine Arbeit getan hat, erhält man eine Zusammenfassung über den Erfolg der Installation. Sollte es dabei – wie auch in der Abbildung zu erkennen – zu Problemen gekommen sein, findet man in den Installationslogs Hinweise zur Fehlerbehebung. Diese sind unter *C:\ProgramData\Microsoft\Team Foundation\Server Configuration\Logs* zu finden.

Das Set-up versucht bereits, einen Build-Agenten für die Ausführung von Team Builds mit zu installieren. Dieser Schritt unterscheidet sich wesentlich von den vorherigen Ver-

sionen. Die Build-Agenten werden in den Ordner *C:\TfsBuildAgents* installiert und als Windows-Dienst mit dem Namen *VSO Agent* gefolgt vom Computernamen bereitgestellt. Dieser Dienst soll zunächst mit dem lokalen Dienstkonto ausgeführt werden. Mit dieser Konfiguration lässt sich der Dienst jedoch nicht starten. Um den obigen Fehler zu korrigieren, muss man also lediglich in der Diensteverwaltung einen anderen Benutzer für den Dienst einstellen. Dem neuen Build-System ist ein eigenes Kapitel gewidmet, für den Moment soll es genügen, einen funktionsfähigen Build-Agenten installiert zu haben. Wer bereits mit dem TFS 2012 und dessen XAML-Builds gearbeitet hat, der kann im Anschluss direkt die „alten" Build-Dienste installieren.

Nach dem Set-up öffnet sich die Team-Foundation-Server-Managementkonsole (Abbildung 2.2). Diese dient der Konfiguration des Servers und kann auch zum nachträglichen Konfigurieren weiterer Features, zum Beispiel der XAML-Build-Dienste, genutzt werden. Für den Anfang gibt es hier jedoch nicht viel zu tun.

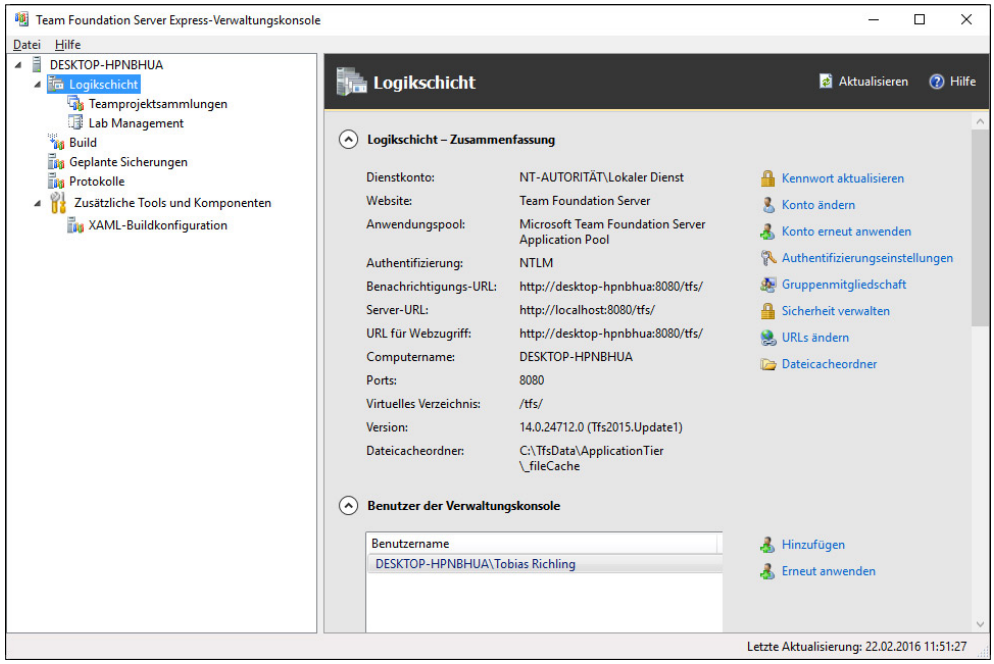

Abbildung 2.2: Die TFS-Managementkonsole

2.1.4 Visual Studio Team Services

Wer gar keine Lust auf eine Installation hat, kann auch Visual Studio Team Services verwenden. Dabei handelt es sich um einen TFS in der Cloud. Zum Starten geht man auf *www.visualstudio.com* und wählt dort „Weitere Informationen zu Team Services". So gelangt man zur Seite in Abbildung 2.3. Dort findet man auch Einzelheiten zum Preismodell.

Um sich anzumelden, benötigt man ein Microsoft-Konto. Fünf Benutzer sind pro Konto kostenlos. Weitere Benutzer kosten monatlich einen gewissen Betrag. Nutzen zum Beispiel zehn Personen den Dienst, so zahlt man monatlich ca. 25 €, dies entspricht 5 € pro weiterem Nutzer. Bei 50 Nutzern werden knapp 300 € fällig, das entspricht knapp 7 € je Benutzer zwischen 11 und 100. Ab 100 Benutzer wird es dann wieder günstiger, hier fallen nur noch knapp 4 € an, ab 1 000 Benutzern nur noch weniger als 2 € pro weiterer Lizenz, konkret wären das dann knapp 3 700 €.

Der Zugriff auf die agilen Boards, Arbeitsaufgaben und Fehler ist für eine beliebige Anzahl an Personen kostenlos enthalten. Auch MSDN-Abonnenten ab einer gewissen Stufe können in beliebiger Anzahl hinzugefügt werden. Außerdem gilt eine CAL für die Team Services auch für eine lokale TFS-Umgebung. Somit kann ein Nutzer, für den man eine Cloud-Lizenz bezahlt, auch in einer lokalen TFS-Umgebung quasi kostenlos mitarbeiten. Über einen Preisrechner kann man sich ausrechnen lassen, was ein konkretes Set-up kostet.[5] Hier kann man auch auswählen, was für weitere Ressourcen, zum Beispiel Build-Agenten, man gerne nutzen möchte.

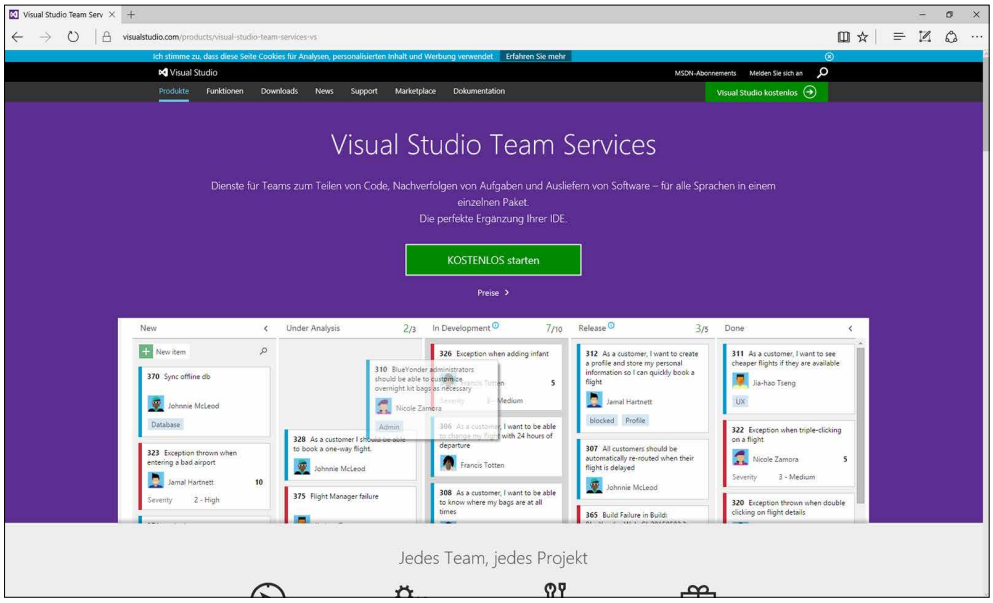

Abbildung 2.3: Die Startseite der Visual Studio Team Services

Zur Anmeldung klickt man auf Kostenlos starten, gibt eine Subdomain an, unter der man seinen TFS erreichen möchte, und schon ist man fertig. Diese Prozedur dauert nur wenige Sekunden. Danach gelangt man auf die Startseite seines Accounts und kann mit der Arbeit beginnen.

Alle Funktionen des TFS Express sind auch in dieser Version zu finden und sogar noch mehr. So sind in der Cloud-Variante des TFS, auch im kostenlosen Paket, die agilen Pla-

5 *https://azure.microsoft.com/de-de/pricing/calculator/?service=visual-studio-online*

nungstools wie das Taskboard und das Kanban-Board enthalten, die Microsoft aus der Express-Variante gestrichen hat. Auch Team Build ist in der Cloud möglich, hierfür ist also auch keine lokale Installation erforderlich. Die Build-Dienste werden nach Minuten abgerechnet. Der auf Workflow Foundation basierende XAML-Build-Dienst wird jedoch im Laufe des Jahres 2017 eingestellt.[6]

2.1.5 Versionsvergleich

Die On-Premise-Version des TFS beinhaltet natürlich sämtliche Produktfeatures, ist dafür aber auch die Installationsvariante, die die größten Kosten verursacht und die meiste Infrastruktur erfordert.

Die Express-Variante verursacht keine Lizenzkosten und für den Anfang minimale Kosten für Hardware. Ein normales Desktopbetriebssystem ist völlig ausreichend. Allerdings ist diese Kostenersparnis natürlich auch mit gewissen Einschränkungen verbunden. So darf der kleine TFS nur von fünf Nutzern parallel verwendet werden. Für jeden weiteren Nutzer muss eine CAL (Client Access License) erworben werden. Die Preisgestaltung dabei sorgt dafür, dass diese Variante schon nach sehr wenigen Nutzern nicht mehr sehr attraktiv ist. Außerdem ist diese TFS-Variante als Datenbank an seinen Namensvetter, den SQL Express, gebunden. Damit unterliegen die TFS-Datenbanken den Beschränkungen des kleinen SQL Server, also zum Beispiel 4 GB maximale Datenbankgröße, nur eine CPU und eine Beschränkung bezüglich des angesprochenen Hauptspeichers. An Features fehlen neben den Enterprise-Funktionen wie SharePoint-Integration und das Data Warehouse insbesondere die agilen Planungstools auf der TFS-Webseite. Hierzu zählen die Backlog-Verwaltung, die agile Sprintplanung und das Taskboard. Allerdings gibt es Drittanbieter, die ein entsprechendes Tooling anbieten, wie etwa die AIT GmbH. Auch auf dem Blog des Autors ist eine einfache Implementierung eines Taskboards zu finden, so dass sich diese Lücke noch recht einfach schließen lässt.

Die Visual Studio Team Services umfassen auch in ihrem kostenlosen Modell diese sehr komfortablen Tools, was sie auch funktional zu einer echten Alternative macht. Allerdings besteht auch hier, wenn man nichts bezahlen möchte, eine Beschränkung auf fünf parallele User. Auch Builds in der Cloud werden unterstützt, 240 Minuten darf man von diesem Dienst pro Monat kostenlos nutzen. Auf das Data Warehouse und den SharePoint muss man auch in der Onlinevariante des TFS verzichten.

Viele Features des neuen TFS entfalten sich erst im Zusammenspiel mit Visual Studio. Einige Features sind sogar an dessen verwendete Edition gebunden. Einen genauen Featurevergleich liefert Microsoft gleich selbst.[7] Durch die Reduktion der Produktlinien sind viele Features, die früher Premium- oder gar Ultimate-Kunden vorbehalten waren, nun Teil der Professional-Version geworden. Dazu gehören Features wie PowerPoint-Storyboarding oder das Feedbacktool.

6 *https://www.visualstudio.com/get-started/setup/get-more-build-or-load-testing-vs*
7 *http://www.microsoft.com/visualstudio/deu/products/compare*

2.2 Der TFS im Schnelldurchlauf

2.2.1 Mit dem TFS verbinden

Nach einer erfolgreichen Installation steht nun ein erster Rundgang durch das neu erworbene Produkt an. Für einen Entwickler bedeutet das: auf ins Visual Studio. Dort soll die Rundreise starten, im Zuge derer ein kompletter Durchlauf durch den TFS vorgestellt wird. Der folgende Abschnitt dient als Schnellstart, in dem querschnittsartig durch alle Bereiche dieses Buchs geschnitten wird. Dabei wird auf tiefergehende Erläuterungen weitgehend verzichtet – diese sind in den folgenden Kapiteln zu finden.

Der Ausgangspunkt für das Arbeiten mit dem TFS aus dem Visual Studio ist der Team Explorer. Im Laufe der Zeit hat dieser immer wieder, zum Teil umfangreiche, Überarbeitungen erfahren und hat sich dadurch zu einem zentralen Arbeitsmedium entwickelt. Falls der Team Explorer beim Starten von Visual Studio noch nicht sichtbar ist, kann er über das Menü Ansicht | Team Explorer oder die Tastenkombination STRG + ^, M eingeblendet werden. Zunächst sieht der Team Explorer noch ziemlich leer aus. Um mit der Arbeit zu beginnen, muss man sich mit einem Teamprojekt verbinden. Dazu klickt man auf das Steckersymbol in der oberen Symbolleiste und dann auf den Link Verbindungen verwalten, wo man aus dem Kontextmenü Mit Teamprojekt verbinden auswählt (Abbildung 2.4)

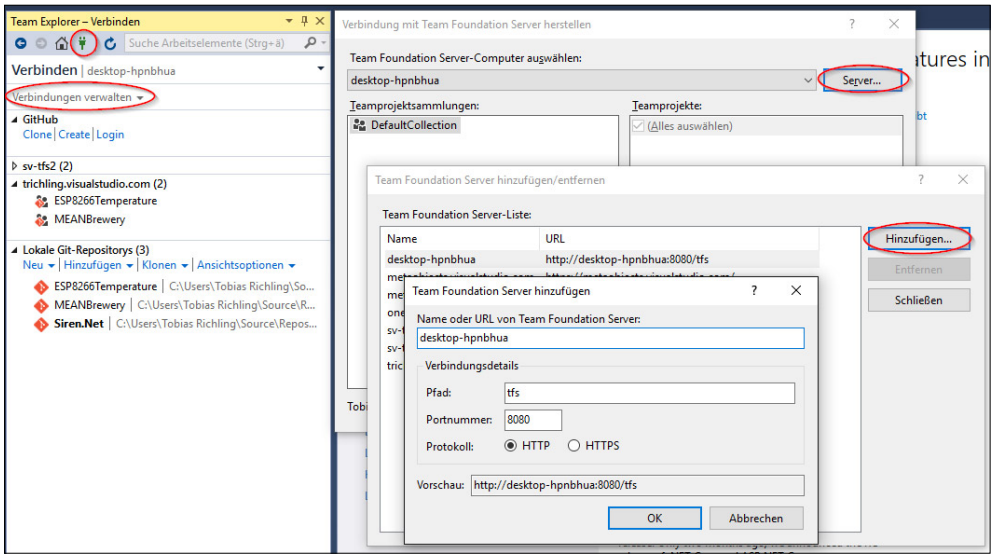

Abbildung 2.4: Der Team Explorer und der Dialog zum Verbinden mit dem TFS

Ähnlich wie ein Browser verfügt der Team Explorer über Vor- und Zurück-, einen Home- sowie einen Refresh-Button am oberen Fensterrand. Letzterer ist bereits ein Hinweis darauf, dass im Team Explorer alles asynchron vonstattengeht. Darunter ist der aktuell ausgewählte Bereich (Verbinden) zu erkennen sowie der Server, mit dem man verbunden ist. Im Hauptbereich werden später die Funktionsbereiche eingeblendet sein.

Um eine neue Verbindung zu einem TFS anzulegen, klickt man in dem Dialog Verbindung mit dem Team Foundation Server herstellen auf den Button Server… In dem neuen Dialog trägt man im Feld Name oder URL von Team Foundation Server den URL zu dem Server ein, mit dem man sich verbinden möchte. Im Falle eines TFS-Serviceaccounts ist dies *<Ihr Account Name>.visualstudio.com*. Bei einer lokalen Installation können Sie einfach den Name des Computers, auf dem der TFS installiert ist, verwenden. Die übrigen Felder kann man zunächst außer Acht lassen. Im Feld Vorschau wird der vollständige URL dargestellt. Der Dialog wird mit Ok bestätigt, daraufhin erscheint der neue Server in der Liste.

Zurück in dem ersten Dialog aus Abbildung 2.4 taucht der soeben erstellte Server in dem Kombinationslistenfeld auf. Links werden alle Teamprojektkollektionen des Servers aufgelistet, rechts daneben die Teamprojekte, die sich in der gewählten Kollektion befinden. Mit einem klickt auf Verbinden (in der Abbildung verdeckt) verbindet man sich mit dem gewählten Server und der gewählten Teamprojektkollektion.

2.2.2 Ein Teamprojekt anlegen

Nachdem eine Verbindung zum Server hergestellt ist, muss ein neues Teamprojekt angelegt werden. Dies bewerkstelligt man über den dreieckigen Knopf rechts neben Verbinden und dem Namen des Servers, mit dem man verbunden ist. Im Kontextmenü wählt man Projekte und Meine Teams | Neues Teamprojekt (Abbildung 2.5).

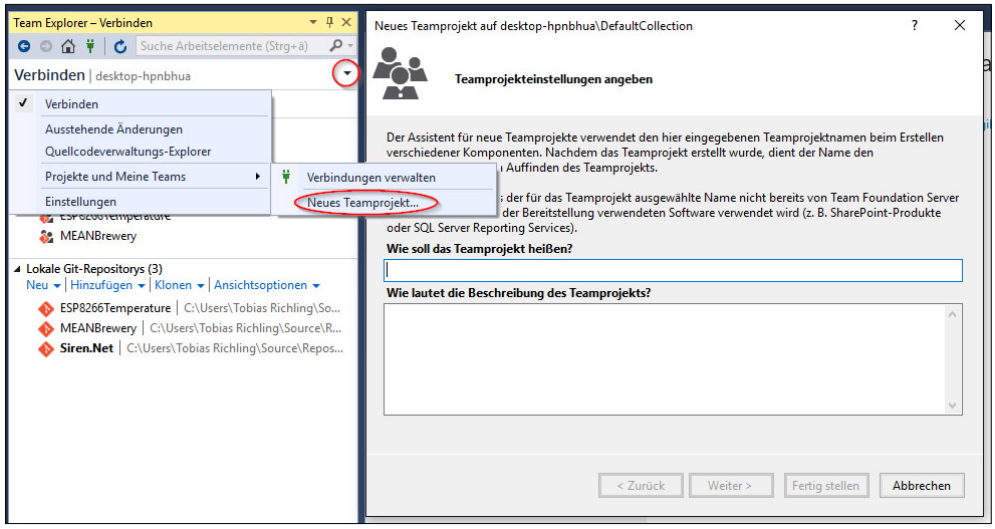

Abbildung 2.5: Ein neues Teamprojekt anlegen

Im folgenden Assistenten muss dem Teamprojekt zunächst ein Name und optional eine Beschreibung zugewiesen werden (Abbildung 2.5 rechts). Der Name taucht später im Team Explorer wieder auf. Er kann später nicht mehr geändert werden, wie immer in der Softwareentwicklung sollte man den Namen also mit Bedacht wählen. Der nächste

Schritt ist von entscheidender Bedeutung für das Arbeiten mit dem TFS. Hier wird die Prozessvorlage ausgewählt (Abbildung 2.6 links). Diese beeinflusst, welche Arten von Arbeitsaufgaben definiert sind, wie der Arbeitsablauf mit diesen Aufgaben definiert ist und welche Abfragen und Berichte vordefiniert werden. Microsoft liefert drei Vorlagen aus, die in Tabelle 2.1 zusammengestellt sind.

Prozessvorlage	Beschreibung
Scrum	Ein Prozesstemplate, das den Prozess von Scrum abbildet. Es werden die für Scrum üblichen Begriffe wie Sprint, Backlog Item und Task verwendet.
Agile	Diese Vorlage ist für Teams gedacht, die agil entwickeln, ohne jedoch Scrum zu verwenden.
CMMI	Stellt ein Prozesstemplate zur Nutzung des CMMI Prozesses bereit.

Tabelle 2.1: Die Microsoft-Prozessvorlagen

Die folgenden Ausführungen beziehen sich alle auf die Scrum-Vorlage, weshalb diese zur Auswahl empfohlen wird.

Der nächste Schritt ist ebenfalls von grundlegender Bedeutung: Es muss festgelegt werden, welches Versionskontrollsystem verwendet werden soll (Abbildung 2.6 rechts). Zur Auswahl stehen Git und die Team-Foundation-Server-Versionskontrolle (TFVC). Bei Git handelt es sich um ein dezentrales System, in dem es technisch gesehen keinen zentralen Server gibt. Die TFVC ist hingegen ein zentralisiertes Versionskontrollsystem, in dem der TFS an allen wesentlichen Aktionen beteiligt ist. Eine erschöpfende Behandlung von Git und der Unterschiede sowie Vor- und Nachteile der jeweiligen Systeme ist nicht Gegenstand dieses Buchs. Da Git als Versionskontrollsystem jedoch eine sehr hohe Popularität besitzt, und auch Microsoft für seine Open-Source-Entwicklungen Git einsetzt, wird in diesem Walkthrough Git als Versionskontrolle verwendet.

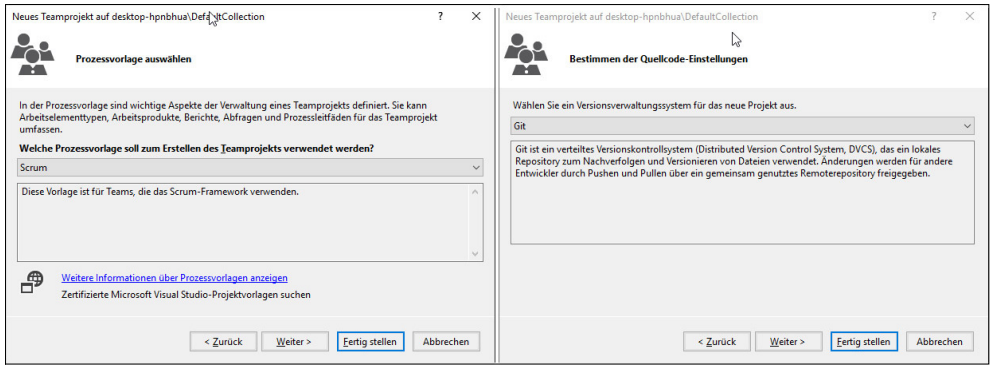

Abbildung 2.6: Wichtige Schritte im Assistenten zum Anlegen eines Teamprojekts

Die folgende Seite zeigt eine Zusammenfassung der getroffenen Einstellungen. Durch den Button Fertig stellen wird die Erstellung des Projekts gestartet, was einige Minuten in Anspruch nehmen kann. Im Anschluss erscheint ein Dialog, der die erfolgreiche Anlage des Teamprojekts vermeldet.

2.2.3 Arbeiten mit der Quellcodeverwaltung

Der zentrale Aspekt der Arbeit von Entwicklern ist natürlich der Quellcode. Bevor man also weitere Funktionen des TFS erschließt, braucht man zunächst ein Projekt, an dem man arbeiten kann.

In der Versionsverwaltung wird mit so genannten Codelinien gearbeitet. Eine Codelinie ist ein kompletter Stand des Quellcodes einer Applikation. Im einfachsten Fall arbeitet man mit einer Codelinie. Wenn die Applikation umfangreicher und der Entwicklungsprozess dementsprechend angepasst wird, führt man weitere Codelinien ein. Ein Beispiel hierfür ist jeweils eine Codelinie für jede ausgelieferte Version der Software. Ein weiteres Beispiel ist eine eigene Codelinie für neue Features oder für zu behebende Bugs. Der Vorteil dieses Ansatzes ist, dass bei späteren Fehlerbehebungen der entsprechende Quellcode noch vorliegt. In der Terminologie der Versionskontrollsysteme werden Codelinien als Verzweigung (Branch) bezeichnet.

Repository klonen

Ein zentrales Element bei der Nutzung von Git sind die so genannten Repositories. Diese beinhalten die entwickelten Artefakte und eine oder mehrere Codelinien. Bei der Verwendung des TFS in Kombination mit Git hält der TFS das zentrale Repository, von dem alle Entwickler ihren Code abholen und ihn irgendwann auch wieder hierhin befördern. Aus der Sicht von Git ist dies jedoch technisch nicht erforderlich, es ist eine rein organisatorische Festlegung. Jeder Git-Nutzer hat ein komplettes Versionskontrollsystem auf seinem Arbeitsrechner, mit dem alle Aktionen durchgeführt werden können. Eine Verbindung zum TFS ist dazu nicht erforderlich. Um mit der Arbeit zu beginnen, muss man jedoch das Repository vom Server *klonen*. Das bedeutet, dass eine lokale Kopie des Server-Repositorys in die lokale Versionskontrolle übernommen wird (Abbildung 2.7).

Auf der Startseite des Team Explorer wird beim erstmaligen Verbinden mit einem Git-Repository aus dem TFS darauf hingewiesen, dass man das Repository zunächst klonen muss (Abbildung 2.7 links). Durch das Klicken auf den entsprechenden Link muss man einen Pfad auf der lokalen Festplatte auswählen, zu dem das Repository geklont werden soll (Abbildung 2.7 Mitte, unteres Textfeld). Darüber bekommt man noch den Server-URL zu dem Repository eingeblendet. Über den Button KLONEN wird der Prozess gestartet. Danach kehrt man zur Startseite zurück, der Hinweis ist verschwunden (Abbildung 2.7 rechts).

HINWEIS: Es ist über den Team Explorer auch möglich, ein Repository anzulegen, ohne ein Teamprojekt dafür zu erstellen. Auch das direkte Arbeiten mit Repositories von GitHub oder einem anderen Git-Server ist möglich. Dieser Überblick beschränkt sich jedoch auf die Kombination eines TFS-Teamprojekts mit Git-Versionskontrolle.

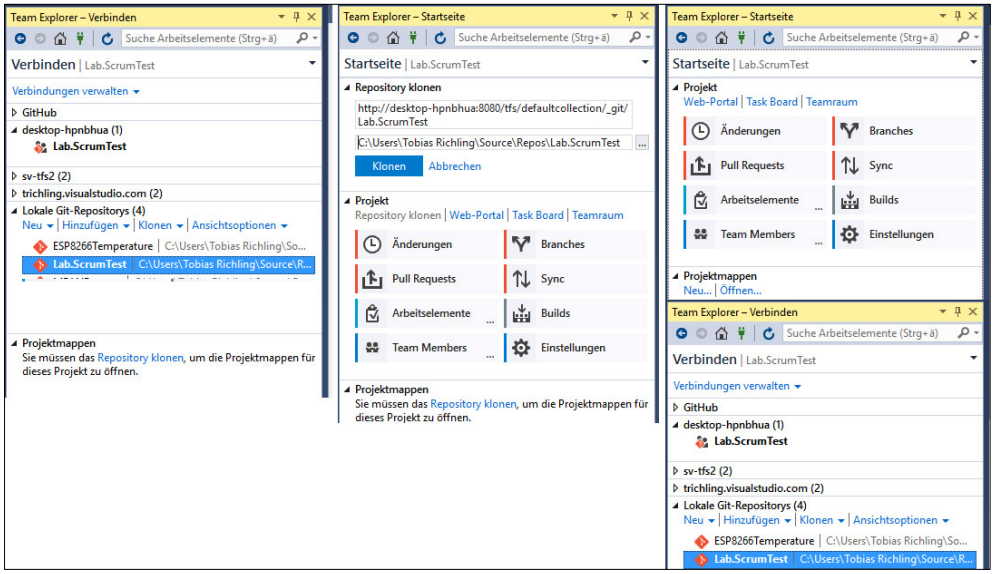

Abbildung 2.7: Ein Git-Repository klonen

Im Bereich Verbinden, erreichbar über das Steckersymbol in der oberen Leiste des Team Explorer, taucht das geklonte Repository nun im Bereich Lokale Git-Repositories auf. Über einen Rechtsklick auf den entsprechenden Eintrag kann man einen Dateiexplorer an dieser Position öffnen. Wer gerne mit der Kommandozeile arbeitet, kann auch eine Eingabeaufforderung in diesem Repository öffnen. Wer Visual Studio installiert hat, bekommt einen Git-Client für die Kommandozeile gleich mitgeliefert und kann diesen parallel zum Visual Studio verwenden.

MEINUNG: Der visuelle Git-Client, der in den Team Explorer des Visual Studio integriert ist, bietet nicht den vollen Funktionsumfang und die Konfigurierbarkeit der Git-Kommandozeile. Wer sich mit Git also schon gut auskennt und volle Flexibilität benötigt, ist mit dem Konsolenwerkzeug besser bedient. Wer gerade mit Git beginnt und einen an TFVC angelehnten Workflow verwenden möchte, für den ist der in den Team Explorer integrierte Client eine gute Wahl. Dieser verwendet allerdings zum Teil auch Begrifflichkeiten, die von den in Git üblichen Begriffen abweichen bzw. die es dort so nicht gibt (z. B. Sync).

Code in die Versionsverwaltung einfügen

Jetzt wird es Zeit ein bisschen Substanz in das noch leere Repository zu bringen. Um ein neues Projekt zu erstellen, kann man den Link Neu (Abbildung 2.7 rechts in der Mitte) verwenden. Er öffnet den Visual-Studio-Dialog zum Erstellen eines neuen Projekts. Zum Üben wird ein neues Konsolenprojekt mit dem vielversprechenden Namen *HelloWorld* angelegt. Man beachte, dass der Pfad, in dem das neue Projekt erstellt wird, der lokale Pfad des geklonten Repositorys ist. Die beiden Häkchen Projektmappenverzeichnis erstellen und Zu Quellcodeverwaltung hinzufügen sollen aktiviert sein.

Nach der Erstellung des Projekts erscheint die Projektmappe im Team Explorer unterhalb des Links Neu. Der Projektmappenexplorer zeigt die Projektstruktur, die Codedateien sind mit dem *Plus*-Overlay markiert, zum Zeichen, dass diese neu in die Quellcodeverwaltung eingefügt worden sind.

> **HINWEIS:** Nutzern, die mit dem TFS und TFVC vertraut sind, wird auffallen, dass es bei der Arbeit mit Git keinen Source Code Explorer gibt. Um die Struktur eines Repositorys zu untersuchen, verwendet man den normalen Dateiexplorer.

Um ein bereits bestehendes Projekt mit in das Repository aufzunehmen, genügt es zunächst die entsprechenden Dateien in ein Verzeichnis unterhalb des Stammverzeichnisses zu kopieren. Im Bereich Änderungen des Team Explorer erscheinen diese Dateien dann zunächst unter der Rubrik Nicht verfolgte Dateien. Durch die Aktion Alle hinzufügen lassen sich diese Dateien dann in das Repository übernehmen (Abbildung 2.8).

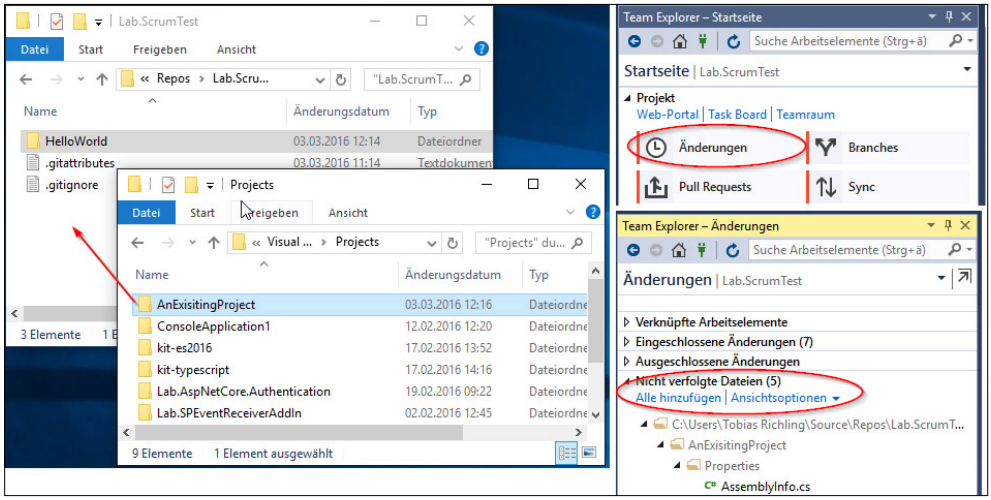

Abbildung 2.8: Bestehende Dateien in die Versionskontrolle einfügen

Es kann sinnvoll sein, beim Hinzufügen bereits bestehender Dateien bestimmte Ordner oder Dateitypen nicht in die Versionskontrolle zu übernehmen. Es ist zum Beispiel wünschenswert, alle DLLs, automatisch generierten Code und andere Artefakte auszuschließen, die sich aus dem eingecheckten Code ableiten. Auf diese Weise wird vermieden, dass Dinge quasi doppelt in der Versionskontrolle sind, denn das Kompilierungsergebnis eines Projekts lässt sich ja durch die Kompilierung erzeugen und muss nicht als solches in der Versionskontrolle liegen. Um dies zu steuern, kommt bei Git eine Datei Namens *.gitignore* ins Spiel. Diese liegt im Stammverzeichnis des Repositorys und kann über den Team Explorer aufgerufen werden. Hierzu klickt man auf Einstellungen | Git | Repositoryeinstellungen. Es öffnet sich ein Team-Explorer-Bereich, in dem verschiedene Einstellungen vorgenommen werden können. Im Bereich Ignorieren und Attributda-

teien wird auf die Konfigurationsdatei für den Ausschluss von Dateien verwiesen, über den Link Bearbeiten wird die Datei im Editor geöffnet und kann dort editiert werden.[8]

> **PROFITIPP:** Komponenten von Drittanbietern, auch in Form von kompilierten DLLs, sollten mit in die Versionsverwaltung, da sie nur in einer bestimmten Version mit dem Quellcode zusammen funktionieren.

Check-in und Change Sets

Nachdem man Dateien in das Repository aufgenommen hat, wechselt man im Team Explorer auf die Seite Änderungen. Dort werden alle derzeit noch lokal getätigten Änderungen angezeigt. Um diese Änderungen zu bestätigen, muss zunächst ein Kommentar eingetragen werden, der die Summe der Änderungen kurz beschreibt (Abbildung 2.9 links). Dieser ist später in einer Historienansicht zu finden, die das Identifizieren bestimmter Änderungen erleichtert. Die Menge der Änderungen sollte daher nicht zu groß sein und inhaltlich zu einer Anforderung gehören. In der Baumansicht der *Eingeschlossenen Änderungen* kann man über das Kontextmenü diesen Eintrag ausschließen. Dies funktioniert sowohl für Ordner als auch für einzelne Dateien. Über den Button Commit lassen sich die Änderungen in das lokale Repository übertragen.

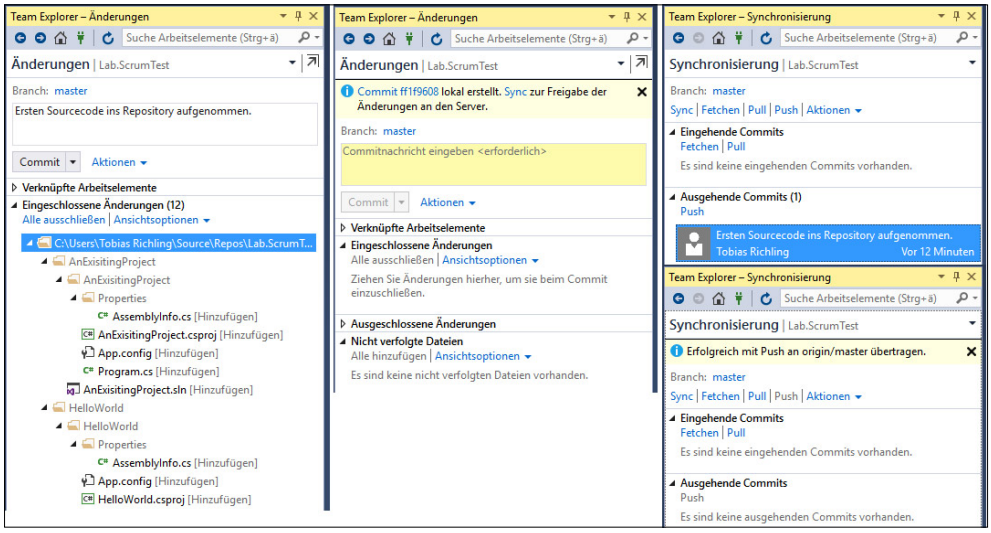

Abbildung 2.9: Check-in in die Versionskontrolle

Jeder Commit erhält eine eindeutige ID (Abbildung 2.9 Mitte oben). Im Commit sind alle Änderungen gebündelt, darüber hinaus werden noch Informationen über den ausführenden Benutzer sowie Datum und Uhrzeit des Vorgangs gespeichert. Über den Link in dem gelben Feld oben im Team Explorer können die Details zu dem Commit abgerufen wer-

8 Unter *https://git-scm.com/docs/gitignore* gibt es Details zum Format der Datei.