



BOOKLET

# ALM APPLICATION LIFE CYCLE MANAGEMENT

# PROFITIEREN SIE VON UNSERER ERFAHRUNG!

## Kontakt Schweiz

bbv Software Services AG  
Blumenrain 10  
6002 Luzern  
Telefon: +41 41 429 01 11  
E-Mail: [info@bbv.ch](mailto:info@bbv.ch)

## Kontakt Deutschland

bbv Software Services GmbH  
Agnes-Pockels-Bogen 1  
80992 München  
Telefon: +49 89 452 438 30  
E-Mail: [info@bbv.eu](mailto:info@bbv.eu)

Der Inhalt dieses Booklets wurde mit Sorgfalt und nach bestem Gewissen erstellt. Eine Gewähr für die Aktualität, Vollständigkeit und Richtigkeit des Inhalts kann jedoch nicht übernommen werden. Eine Haftung (einschliesslich Fahrlässigkeit) für Schäden oder Folgeschäden, die sich aus der Anwendung des Inhalts dieses Booklets ergeben, wird nicht übernommen.

# INHALT

1	Einleitung	5
2	Definition ALM	6
3	Welchen Nutzen bringt ein ALM-Tool	9
4	Rollen im ALM	11
4.1	Rolle «User»	12
4.2	Rolle «Management»	12
4.3	Rolle «Development»	14
4.4	Rolle «Operations»	18
5	ALM-Tools	19
5.1	Point Solutions	20
5.1.1	Bekannte Vertreter der Point Solutions	21
5.2	All-in-one-Solutions	23
5.2.1	Bekannte Vertreter der All-in-one-Solutions	23
5.3	ALM mit Open-Source-Produkten	27
5.3.1	Individueller Baukasten	27
5.3.2	Grenzen	29
5.3.3	Lösungsbeispiel mit Open-Source-Produkten	29
5.3.4	TeamCity	30
6	ALM mit Microsoft	31
6.1	Lebenszyklus laut Microsoft	32
6.2	Tools von Microsoft	34
6.2.1	Visual Studio	34
6.2.2	Team Foundation Server	35
6.2.3	Team Explorer	37
6.2.4	Team Web Access	37
6.2.5	Team Web Access – Work Item View Only	37
6.2.6	Office	37

6.2.7	Visual Studio Team Foundation Server Reports	37
6.2.8	Microsoft Test Manager	38
6.2.9	Interaktion für Work Items	38
6.3	Best-Practice «Agile mit Microsoft»	39
6.4	ALM in Ihrer Firma einführen	42
7	Persönliche Erfahrungen	43
8	Autoren	47
9	Quellenverzeichnis	48
9.1	Literaturverzeichnis	48
9.2	Abbildungsverzeichnis	49
10	Glossar	50

# 1 EINLEITUNG

Im Zuge der Verbreitung der agilen Entwicklung nimmt die Popularität von Application Life Cycle Management (ALM) Tools zu. Diese unterstützen alle Stakeholder eines Softwareprojekts während des gesamten Lebenszyklus.

Die Tools können unabhängig vom gewählten Vorgehensmodell eingesetzt werden. Da das Aufzeigen der vorgehensmodellspezifischen Unterschiede den Rahmen dieses Booklets sprengen würde, haben sich die Autoren entschieden, ALM anhand von Scrum vorzustellen.

Das Booklet ist in drei Teile gegliedert. Im ersten Teil gehen wir der Frage nach, was ALM überhaupt ist. Der zweite Teil beinhaltet eine Marktübersicht von ALM-Tools mit deren Vor- und Nachteilen. Im letzten Teil wird noch detaillierter auf den Microsoft Team Foundation Server eingegangen.

## 2 DEFINITION ALM

Der Begriff Application Life Cycle Management wird oft im Zusammenhang mit Tools verwendet. Diese bieten Unterstützung bei diversen Softwareentwicklungstätigkeiten. Dabei kommt die Frage auf, welche Tätigkeiten überhaupt in einem Softwareprojekt anstehen; und zwar nicht nur für die eigene Rolle, sondern für alle involvierten Stakeholder.

Grob gegliedert gibt es folgende Zeitabschnitte im Leben einer Applikation:

**Vision/Start:**

Zuerst werden Visionen von möglichen Projekten gesammelt. Diese werden so weit ausgearbeitet, dass es möglich ist, sie zu priorisieren. Anhand der Priorisierung wird entschieden, welche Softwareprojekte gestartet werden sollen.

**Entwicklung:**

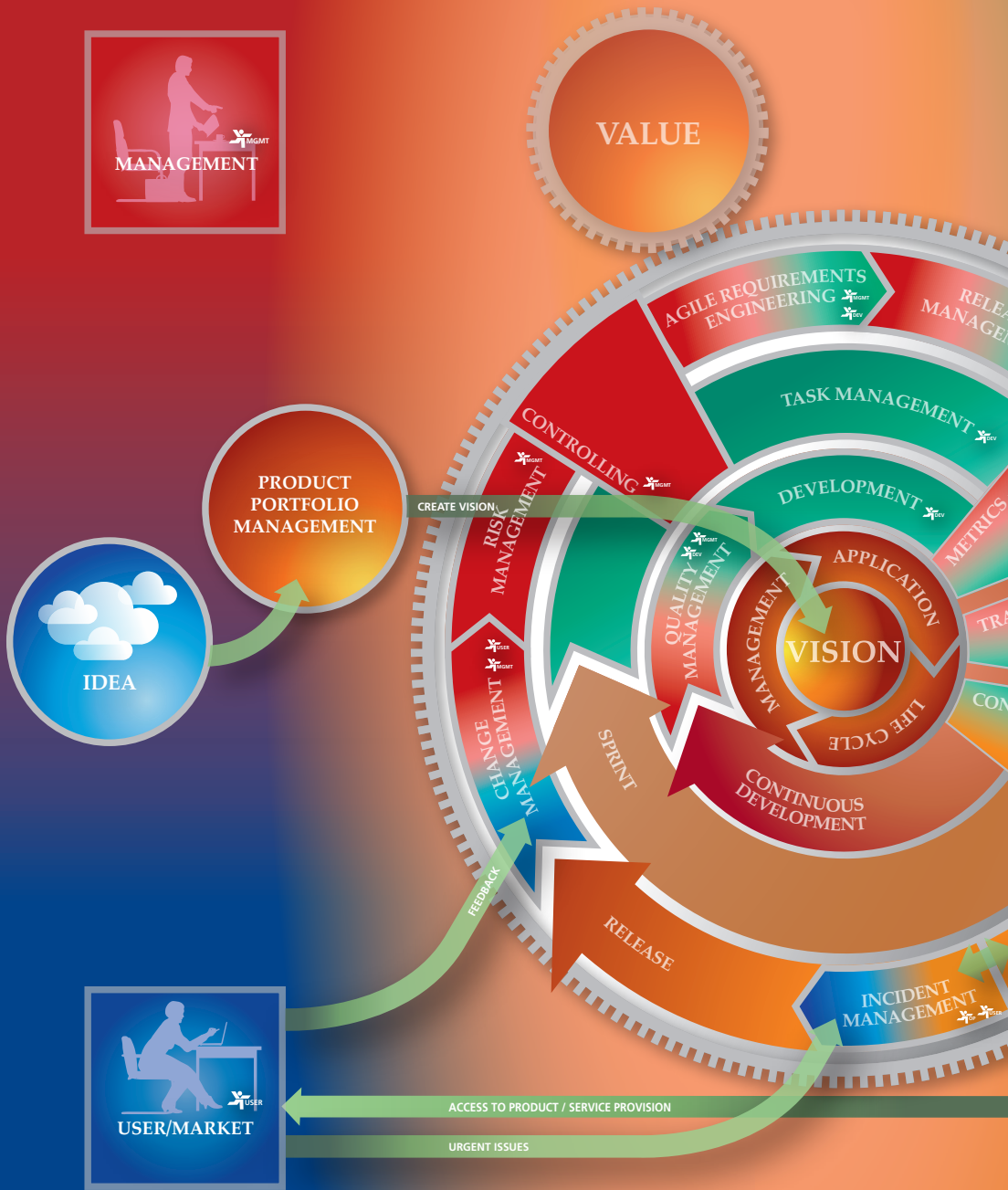
Während der Entwicklung wird die Vision in eine lauffähige Applikation überführt. Die verschiedenen Tätigkeiten der Softwareentwicklung werden vielfach innerhalb von Sprints wiederholt. Dabei gibt es einen fließenden Übergang in die Produktion.

**Produktion:**

Die Applikation steht dem Benutzer zur Verfügung und kann produktiv verwendet werden. Da sich Entwicklung und Produktion in der Regel überlagern, werden laufend neue Releases installiert.

**Ende/Ablösung:**

Die Applikation erreicht das Ende ihrer Einsatzzeit und wird eingestellt oder durch eine neue Applikation abgelöst.





# 3 WELCHEN NUTZEN BRINGT EIN ALM-TOOL

Ein grosser Teil der Softwareprojekte wird nicht termingerecht fertiggestellt und/oder die Budgetvorgaben können nicht eingehalten werden. Ein ALM-Tool erlaubt im Idealfall allen Stakeholdern, eine transparente Übersicht des Projektstandes zu erhalten. Dies ermöglicht ihnen, frühestmögliche Anpassungen im Projekt anzugehen, um den Projekterfolg zu garantieren.



Feature	Integrated Development Environment (IDE)	Source Code Management (SCM)	Build Server	Work Tracking	Reporting
Product Portfolio Management				●	●
Metrics		●	●		●
Controlling					●
Risk Management				●	●
Change Management				●	
Quality Management	●	●	●	●	●
Development	●	●	●		
Agile Requirements Engineering				●	
Release Management		●	●	●	
Task Management				●	
Traceability		●		●	
Continuous Deployment		●	●		
Operations Management				●	
Incident Management				●	

Ein gutes ALM-Tool unterstützt die Zusammenarbeit aller Projektbeteiligten. Über die zentrale Datensammlung können die Stakeholder alle aktuellen Informationen austauschen. Beispielsweise muss bei der Anpassung von Requirements die neue gültige Version sofort allen Stakeholdern zur Verfügung stehen. Insbesondere im Fall von verteilten Teams, wie beispielsweise bei Offshoring, kommt diesem Punkt eine essenzielle Bedeutung zu.

Jede Stakeholdergruppe ist für unterschiedliche Themenbereiche verantwortlich. Dadurch beurteilt sie den Projektstand anhand unterschiedlicher Metriken. Ein gutes ALM-Tool bietet eine Vielzahl von Metriken, die auf der Basis eines zentralen Datenbestandes ermittelt werden.

Scrum als agiler Entwicklungsprozess setzt auf sich wiederholende Zyklen. Dadurch entstehen diverse sich wiederholende Aufgaben. Diese repetitiven Aufgaben sind unter anderem: Überprüfen von Coderichtlinien, Ausführen von Testsuiten, Deployment etc. ALM-Tools ermöglichen die Automatisierung dieser Tasks. So können der Arbeitsaufwand und dadurch generierte Kosten minimiert werden, und gleichzeitig kann die Qualität gesteigert werden.

## 4 ROLLEN IM ALM

In den Lebenszyklus einer Applikation sind diverse Stakeholder involviert. Im Folgenden werden diese in vier Rollen zusammengefasst: User, Management, Development und Operations.

#### **4.1 ROLLE «USER»**

Für den User wird die Applikation entwickelt. Den User und seine Bedürfnisse zu verstehen, ist essenziell. Darum ist es wertvoll, ihn über den ganzen Lebenszyklus der Applikation mit einzubeziehen.

##### **Idea**

Aus einem Bedürfnis des Users entsteht eine Idee für eine Applikation. Der User in diesem Kontext kann auch generell der Markt sein. Die Idee bildet die Basis für die Vision des Projekts.

##### **Requirements**

Während der Entwicklung wird der Auftraggeber im Zuge des Requirement Engineerings zusammen mit der Entwicklung Requirements ausarbeiten. Ist kein Enduser verfügbar, wird diese Tätigkeit vom Requirement Engineer übernommen, der das gleiche Verständnis wie der User aufbringt.

##### **Acceptance-Test**

Ist ein Release fertiggestellt, wird dieser getestet, abgenommen und User-Feedback eingeholt. In Scrum werden diese Schritte vielfach wiederholt.

##### **Feedback/Urgent Issues**

Wenn die Software in einen Produktivbetrieb übergeht, kann der User Feedback in Form von Feature Requests liefern, die ins Change Management einfließen. Störungen während des Betriebs werden durch das Incident Management behandelt.

#### **4.2 ROLLE «MANAGEMENT»**

Der eigentlichen Softwareentwicklung übergeordnet finden sich diverse Managementtätigkeiten. Ein ALM-Tool soll auch auf dieser

Ebene Informationen liefern und eine Einflussnahme auf das Projekt ermöglichen. Im Folgenden werden einige dieser Aufgaben kurz erläutert.

### **Vision**

Jedem Projekt sollte eine Vision zugrunde liegen, die dem Team den Weg weist. Sie beschreibt die Essenz und die Motivation des Produkts in einer verständlichen Weise.

### **IT-Portfolio-Management**

IT-Verantwortliche müssen die Geschäftsstrategie im Auge behalten, das heisst alle Anwendungen und laufenden Projekte als Ganzes betrachten. Sie müssen über den Einsatz von Anwendungen und die Priorisierung von Projekten entscheiden. Ein hilfreiches Werkzeug dazu können Projektmetriken sein, wie z. B. Projektgeschwindigkeit oder Anforderungsabdeckungsgrad.

### **Risk Management**

Beim Risk Management gilt es, einen Mitigationsplan zu erstellen und entsprechende Gegenmassnahmen zu treffen. Dazu können Metriken hilfreich sein sowie ein funktionierendes Projektcontrolling. Dabei kann z. B. die Fehlerrate gemessen werden oder wie vielen Storys ein hohes Risiko beigemessen wird.

### **Project Planning**

Auch die Projektplanung ist ein Teil des ALM. Terminplanung, Ressourceplanung oder Releaseplanung sollen idealerweise in einem hoch integrierten System stattfinden. So kann erreicht werden, dass die Planung automatisch aktuell bleibt; auch wenn beispielsweise eine unvorhergesehene Ressourcenverschiebung stattfindet.

### Project Controlling

Während der Laufzeit der Entwicklung gilt es, verschiedene Kennzahlen laufend im Auge zu behalten. Dazu gehören unter anderem der Projektfortschritt aufgrund der abgeschlossenen Requirements sowie der finanzielle Aufwand. Durch präzise Metriken und ein integriertes Echtzeit-Reporting kann ein ALM-Tool in diesem Bereich grosse Transparenz schaffen.

### Metrics

Das Ziel von Metriken soll letztlich sein, Optimierungen im Projekt vorzunehmen. Zu Beginn ist es wichtig zu definieren, was man messen will. Die Ergebnisse sollen in einen Verbesserungsprozess einfließen. Eine integrierte ALM-Lösung kann dabei jederzeit Daten zu unterschiedlichsten Aspekten des Projekts liefern, die mit geeigneten Werkzeugen analysiert werden können. Beispiele sind Projektgeschwindigkeit, verbleibende Arbeit oder Fehlerbehebungszeit.

### Traceability

Unter Traceability versteht man die Rückverfolgbarkeit von Requirements im Laufe des Projekts. Es ist von Interesse, welche Requirements bereits umgesetzt worden sind oder wie weit ihre Entwicklung fortgeschritten ist. Dabei sind auch die Einhaltung von Standards und die Qualität von Interesse. Ein ATDD-Ansatz (Acceptance Test Driven Development), bei dem man Tests für die Spezifikation automatisiert, kann in diesem Punkt dienlich sein. Dazu ist ein hoher Integrationsgrad der ALM-Toolchain hilfreich.

## 4.3 ROLLE «DEVELOPMENT»

Die Softwareentwicklung geschieht in Scrum mittels Sprints. Jeder Sprint entspricht einer Iteration und resultiert in einem Inkrement (eine lauffähige, potenziell auslieferbare Version) der Software. Da-

durch werden pro Sprint alle Tätigkeiten des Softwareentwicklungs-Lebenszyklus (Software Development Life Cycle SDLC) durchlaufen. Bei einer agilen Arbeitsweise werden die einzelnen SDLC-Tätigkeiten nicht sequenziell, sondern parallel ausgeführt. Diese Tätigkeiten bilden zusammen einen essenziellen Aspekt des ALM, nämlich den der Entwicklung.

### Agile Requirement Engineering

Als Ausgangslage für das Development dienen Requirements. Diese werden oft in Zusammenarbeit mit dem User erarbeitet und in einer adäquaten Form den Entwicklern zur Verfügung gestellt. In Scrum werden dazu User Stories erstellt, wobei jede Story einen messbaren Kundennutzen aufweisen sollte. Die Stories werden in einem priorisierten Projekt-Backlog abgefüllt und über ein Tracking Tool allen Stakeholdern zur Verfügung gestellt.

Ein integriertes Dokument-Management-System kann weitere Dokumente und Artefakte allen Stakeholdern bereitstellen.

### Task Management

Unter Task Management versteht man das Speichern, Bearbeiten und Finden von Tasks. Ein Task in diesem Zusammenhang ist ein Stück Arbeit: eine User Story, ein Test Case, ein Defect etc. Entwickler müssen ihren nächsten Job im Task-Management-Prozess finden. Scrum bietet dazu das Scrum Board.

### Development

Die eigentliche Umsetzung besteht hauptsächlich aus dem Handwerk des Programmierens. Eine moderne Entwicklungsumgebung (Integrated Development Environment IDE) hilft dem Entwickler massgeblich dabei, effizient zu arbeiten. Als ein Teil des Developments verstehen sich hier auch das Design der Architektur und das Testen der Software.

Artefakte dieser Phase sind der Source Code sowie die ausführbare Software. Es versteht sich von selbst, dass der Source Code vernünftig verwaltet werden muss. Ein Source-Code-Management (SCM)-Tool übernimmt die Aufgaben der Verwaltung, Versionierung und Rückverfolgbarkeit (Traceability) des Source Codes.

Wann immer eine Anpassung am Source Code ins Source-Code-Management-Tool übernommen wird, soll dies über einen Continuous-Integration-Prozess geschehen. Dabei wird durch das Kompilieren der Software auf einem Build Server und durch Ausführen der Tests die Integrität und Qualität der Software laufend sichergestellt.

### Quality Management

Quality Management stellt sicher, dass die ausgelieferte Software eine hohe Qualität aufweist. Einerseits kann davon ausgegangen werden, dass der Entwickler bereits automatisierte Tests (z. B. in Form von TDD/ATDD) geschrieben hat. Man kommt aber andererseits trotz guter Testabdeckung nicht darum herum, die Software auch manuell in einer realistischen Testumgebung zu testen.

Durch Testen kann gezielt nach Fehlern gesucht werden. Quality Management enthält auch das Defect Management. Ebenso sollen hier nicht funktionale Aspekte wie beispielsweise Usability und Performance untersucht werden.

Resultate all dieser Tests sollen in die Entwicklung zurückfließen. Idealerweise existiert, wie bereits bei der Requirements-Management-Phase erwähnt, ein Backlog, in dem auch Fehler oder Schwachstellen erfasst werden und welches über ein Issue-Tracking-Tool sämtlichen Stakeholdern zur Verfügung steht.



Ein weiterer interessanter Punkt im Quality Management sind Metriken für das Entwicklungsprojekt. Metriken können Aussagen über bestimmte Qualitätsaspekte (z. B. Anzahl ausgelieferter Fehler pro Release, Testabdeckung) des Projekts machen. Tools können dabei helfen, diese Metriken laufend im Auge zu behalten, um ggf. Massnahmen zu ergreifen.

Im erweiterten Sinn ist auch eine Qualitätssicherung der Prozesse anzustreben, beispielsweise durch einen standardisierten und klar definierten Entwicklungsprozess.

### **Release Management**

Release Management kümmert sich schliesslich um die Erstellung und das Deployment eines Softwarepakets auf ein Test- oder das Produktivsystem.

Ein Releasepaket soll dabei möglichst automatisch und schnell erstellt werden können. Ein Build Server stellt die dazu nötige Infrastruktur bereit. Ebenso soll es möglich sein, das Paket unkompliziert auf dem Zielsystem installieren zu können.

Zu den weiteren Managementaspekten gehört es, zu bestimmen, welche Features und Bug Fixes in einen bestimmten Release hineinfliesen sollen. Als Entscheidungsgrundlage dienen die Priorität und das Risiko eines Features oder Bug Fixes.

#### **4.4 ROLLE «OPERATIONS»**

Ist ein Release des Produkts bereit, kann dieser dem User zur Verfügung gestellt werden. Die Operations-Rolle beinhaltet alle Tätigkeiten der Bereitstellung und Betreuung der laufenden Applikation.

##### **Continuous Deployment**

Ein vom Development erstellter Release wird auf dem Test- bzw. dem Produktivsystem installiert und so den Usern zur Verfügung gestellt. In einer agilen Umgebung sind diese Releasezyklen sehr klein. Entsprechend wichtig ist ein möglichst automatisches, schnelles und sicheres Continuous Deployment.

##### **Operations Management**

Während des Betriebs nimmt das Operations Management vor allem Monitoringaufgaben wahr. Es ist besorgt, dass die Applikation ordnungsgemäss läuft und wird gegebenenfalls eingreifen. Für die Verwaltung von Aufgaben und Problemen existiert auch für das Operations ein Work Tracking.

##### **Incident Management**

User haben die Möglichkeit, Probleme des produktiven Betriebs zu melden. Bei dringenden Problemen, wie Ausfall, wird der Operator sofort eingreifen. Bei produktspezifischem Feedback kann dies auch ins Change Management einfließen.

## 5 ALM-TOOLS

Über den gesamten Lebenszyklus verteilt bedarf es für jedes spezifische Aufgabengebiet entsprechender unterstützender Tools. Hierbei ergeben sich zwei unterschiedliche Ansätze, wie man zu den für einen selbst optimalen Hilfsmitteln kommt: Zum einen lassen sich für jedes Gebiet stets einzelne voneinander unabhängige Tools einsetzen. Hierbei spricht man von Point Solutions. Zum anderen finden sich sogenannte All-in-one-Solutions, die für den gesamten ALM-Zyklus bereits ein vollständiges Toolset anbieten, das jeden Bereich vollständig abdecken soll.

## 5.1 POINT SOLUTIONS

Point Solutions konzentrieren sich darauf, ein spezifisches Gebiet des ALM bestmöglich abzudecken. Auf Veränderungen am Markt oder Feature Requests kann schnell mit neuen Versionen reagiert werden. Dadurch soll stets sichergestellt werden, dass speziell im Application Development Management (ADM) die Entwickler optimal bei ihrer Arbeit durch effizienzsteigernde Tools unterstützt werden.

Häufig wird von den Unternehmen aus den Point Solutions für ihren Prozess die aus ihrer Sicht jeweils optimale Lösung selektiert. Dadurch entsteht eine Best-of-Breed-Infrastruktur. Eine Herausforderung bei der Auswahl der entsprechenden Tools stellt die jeweilige Integration in die gesamte ALM-Infrastruktur dar. Neben der Integration stellt die eher flache Lernkurve eine weitere Herausforderung dar, die durch die unterschiedlichen User Interfaces oder Datenaustauschformate resultieren kann. Bei der Evaluation von Point Solution wird sehr häufig der Fokus auf Tools gelegt, die den Entwickler bei seiner Tätigkeit optimal unterstützen sollen. Die Erfahrung hat jedoch gezeigt, dass durch Nichtbeachtung auch nur eines Stakeholders bei der Evaluation eine spätere Akzeptanz der neu eingeführten Tools gefährdet ist.

Es ist somit auch keine Überraschung, dass die populärsten Point Solutions aus Open-Source-Projekten hervorgehen. Im Bereich des ADM werden diese Tools von vielen Entwicklern den All-in-one-Solutions vorgezogen. Eines der besten Beispiele ist hierbei sicherlich das Versionskontrollsystem Git. In Kapitel 5.3 finden Sie eine Auswahl weiterer Open-Source-Tools.

### 5.1.1 BEKANNTE VERTRETER DER POINT SOLUTIONS

Zu den jeweiligen Disziplinen im ALM werden nachfolgend einzelne Lösungen von kommerziellen Anbietern aufgeführt.

#### Requirements Management

Visure Solutions bietet mit Visure Requirements (Visure Solutions, 2012) ein eigenständiges Requirements-Management-Tool an, welches den gesamten Lebenszyklus einer Anforderung vom Erfassen und der Analyse bis hin zum Abnahmetestmanagement abdeckt. Visure Requirements lässt sich unter anderem mit JIRA oder Quality Center von Hewlett Packard integrieren.

IBM® Rational® DOORS® (IBM, 2012) ist eine Requirements-Management-Softwarelösung, optimiert für die Bedürfnisse von komplexen und embedded Systementwicklungen.

#### Projektmanagement

TeamPulse von Telerik (Telerik, 2012) ist eine Software für agiles Projektmanagement und bietet neben Planungs- und Projektverfolgungsfunktionalitäten noch Möglichkeiten zum Anforderungsmanagement, Tools zur verbesserten Zusammenarbeit unter den Stakeholdern sowie zum Fehlermanagement in einem Paket an. Daneben kann eine Zwei-Wege-Synchronisation mit dem Team Foundation Server von Microsoft sehr einfach eingerichtet werden. Der Funktionsumfang reicht weit über die klassischen Projektmanagement-Aktivitäten hinaus.

#### Fehlermanagement

Mit YouTrack von JetBrains (JetBrains, 2012) kann mit einer kostenfreien Zehn-Benutzer-Lizenz gestartet werden. JetBrains legt seinen Fokus beim Fehlermanagement-Tool auf einfaches Benutzerhand-

ling und die Möglichkeit einer vollständigen Tastaturbedienung im Web. Eine machtvolle und dennoch benutzerfreundliche Suche optimiert die tägliche Arbeit mit YouTrack. Zudem wird mit YouTrack ein vollständiges Projektmanagement- und Projektverfolgungs-Tool mitgeliefert, um im agilen Umfeld die Teams webbasiert bei ihrer Arbeit optimal zu unterstützen.

### **Build-Management**

JetBrains bietet neben seinem Tool YouTrack noch einen Continuous Integration (CI) Server an. Mit TeamCity lässt sich ein überaus machtvolles Build-Management-System webbasiert betreiben, um alle relevanten Build-Szenarien abzudecken. Zu seinen Stärken zählt die Plugin-Vielfalt; hier finden sich Erweiterungen aus allen erdenklichen Bereichen, die sich bei einem Build-Management-System ergeben können und zu integrieren sind. So zum Beispiel zur Anbindung verschiedener Versionskontrollsysteme wie Mercurial, Git oder TFS; Unterstützung von weiteren Testing Frameworks wie C++ Unit Test oder FitNesse Tests Runner bis hin zu Plugins für Virtualisierungssysteme wie VMWare.

### **Source Code Management**

SourceGear Vault (SourceGear LLC, 2012) ist ein Versionskontrollsystem und kann zusätzlich als Bug-Tracking-Lösung für professionelle Entwicklungsteams eingesetzt werden. Vault basiert auf einer Client-/Server-Architektur und verwendet Technologien wie Microsoft SQL Server und IIS Web Services für bessere Performance, Skalierbarkeit und Sicherheit. Vault ist kostengünstig, einfach zu installieren und intuitiv zu bedienen.

Vault bietet eine Integration in populäre Entwicklungsumgebungen wie Visual Studio und Eclipse an. Durch die Bereitstellung

eines Vault-Kommandozeilen-Tools werden weitere Plattformen wie Mac OS X, Linux und Solaris unterstützt.

Weitere sehr verbreitete Systeme wie Git oder svn werden in Kapitel 5.3 unter dem Aspekt Open Source betrachtet.

## **5.2 ALL-IN-ONE-SOLUTIONS**

Bei der Wahl einer Komplettlösung sollte Wert darauf gelegt werden, dass sich die Toollandschaft nicht nur auf den Bereich des Application Development Management (ADM) konzentriert, sondern möglichst alle Disziplinen des ALM durch die Lösung abgedeckt werden. Das ADM stellt eine entwicklungspezifische Untermenge von ALM dar und wird sehr häufig durch den Einsatz von Open-Source-Tools abgedeckt.

### **5.2.1 BEKANNTE VERTRETER DER ALL-IN-ONE-SOLUTIONS**

Die Reihenfolge der Vertreter stellt keine Rangliste oder Priorisierung durch die Autoren dar.

Die folgende Matrix soll einen groben Feature-Überblick über die bekanntesten Vertreter von All-in-one-Solutions geben. Aus ihnen haben wir einige Anbieter ausgewählt, die detaillierter vorgestellt werden.

Feature-Überblick All-in-one-Solutions									
	Atlassian	IBM	Micro Focus	HP	Rally Software	CollabNet	Version One	Polarion	Microsoft*
Requirements Management	●	●	●	●	●	●	●	●	●
Projektmanagement	●	●		●	●	●	●	●	●
Fehlermanagement	●	●	●	●	●	●	●	●	●
Build Management	●	●			●	●		●	●
Release Management	●	●		●	●	●		●	●
Qualitätsmanagement	●	●	●	●	●	●	●	●	●
Task Management	●	●	●	●	●	●	●	●	●
Reporting	●	●	●	●	●	●	●	●	●
Labor Management				●		●			●
Modeling		●							●
SCCM		●	●		●	●		●	●
Collaboration	●	●			●	●			●
SaaS	●	●			●		●		●

SCCM: Software Change and Configuration Management

SaaS: Software as a Service

\*Microsoft : Kapitel 6 geht im Detail auf Microsoft ALM ein.



### Atlassian

Basierend auf dem Projektverfolgungstool JIRA aus eigenem Hause bietet Atlassian (Atlassian, 2012) eine ganze Reihe von Zusatztools, um alle Bereiche des ALM abdecken zu können. Lediglich der Bereich Labor Management wird hierbei nicht abgedeckt. Atlassian bindet ein breites Spektrum an Open-Source- und kommerziellen Versionsverwaltungssystemen über Plugins an. Zudem stellt Atlassian die Onlineplattform bitbucket zur Verfügung. Diese fungiert als Hostingplattform für die verteilten Versionskontrollsysteme Git und Mercurial. Der angebotene Service stellt zusätzlich einen Issue Tracker und ein Wiki zur Verfügung.

Alle angebotenen Tools integrieren direkt untereinander von Hause aus. Sie können aber auch eigenständig genutzt werden. Des Weiteren bietet Atlassian eine vollständige Cloud-Lösung an. Das Testtool Bonfire stellt eine Browsererweiterung dar, die sich in die aktuellen Browser integriert.

### IBM Rational Software

Mit den Rational Services und Lösungen bietet IBM ein komplexes Toolset auf Basis der Jazz-Plattform für den gesamten ALM-Bereich an. Mit Rational Team Concert kann in den ALM-Bereichen agile Planung, Aufgabenverfolgung, Versionsverwaltung, Continuous Integration und Reporting begonnen werden. Vorhandene Builds auf Basis Ant, Maven und jeder auf Kommandozeilen basierende Build-Aufruf können integriert werden.

Unter dem Titel «Rational solution for Collaborative Lifecycle Management» wird eine vollständig integrierte Lösung angeboten. Sie beinhaltet neben Team Concert auch Quality Manager, Requirements Composer und die Erweiterung Rational Software Architect Design Manager.

IBM bietet zudem jedem Interessenten die Möglichkeit, eine Sandbox für einen beschränkten Zeitraum zu Testzwecken zu nutzen. Es können somit weitgehend alle Funktionalitäten ohne Installationsaufwand genutzt werden. Weitere Informationen zu Tools im Umfeld agile Softwareentwicklung von IBM Rational finden Sie unter (IBM, 2012).

### **Micro Focus Borland ALM**

Micro Focus bedient sich beim ALM-Angebot der Produkte aus der Borland-Produktpalette, ausgehend von Caliber für das Requirements Management, DevPartner für die Entwicklung, über Silk für die automatisierte Softwarequalitätssicherung bis zu StarTeam im Bereich des Software Change und Configuration Management. Die Integration mit bekannten Entwicklungsumgebungen und weit verbreiteten anderen Life-Cycle-Produkten wird ebenfalls angeboten. Das Angebot als SaaS-Lösung wird von Micro Focus weiter forciert und soll zum Standard werden (Borland Software Corporation, 2012).

### **Rally Software**

Rally Software stellt seine ALM-Lösung primär als SaaS-Variante zur Verfügung. Es besteht jedoch auch die Möglichkeit, eine eigenständige Installation innerhalb des eigenen Unternehmens durchzuführen. In der SaaS-Variante werden drei Editionen angeboten: Es kann mit einer Community Edition gestartet werden, den vollständigen Funktionsumfang erhält man jedoch erst mit der Unlimited Edition. Diese bietet dann nahezu alle Tools für den gesamten ALM-Bereich. Auf Prozessebene bieten die Tools sehr gute Unterstützung im agilen Umfeld. Traditionelle Vorgehensmodelle werden nicht direkt unterstützt und werden mit Hilfe von Kanban Boards realisiert. Rally bietet die Lösung aktuell lediglich in englischer Sprache an (Rally Software Development Corp., 2012).

### 5.3 ALM MIT OPEN-SOURCE-PRODUKTEN

Neben den All-in-one-Solutions der grossen Anbieter stehen seit einiger Zeit auch leistungsfähige Lösungen aus dem Open-Source-Bereich zur Verfügung.

Da die traditionellen ALM-Lösungen häufig mit grösseren Kosten für Lizenzen und Wissensaufbau zu Buche schlagen, kamen über die letzten Jahre immer mehr Open-Source-Point-Solutions in den Entwicklungsabteilungen zum Einsatz. Diese Produkte gingen häufig aus grossen Open-Source-Projekten hervor. Nach und nach verbreiteten sich Bugzilla, Trac und andere, vornehmlich Issue-Tracking-Produkte, sowohl in kleinen wie auch in grösseren Firmen. Dieser Trend wird sich, insbesondere mit der Entwicklung der Source-Code-Management (SCM)-Tools hin zu Distributed Version Control Systems (DVCS) wohl noch verstärken, da viele Entwickler heute bereits einen gewissen Standard von GitHub, Sorceforge oder Google Code gewohnt sind und die grossen Anbieter diesen Trend derzeit noch nicht abdecken.

#### 5.3.1 INDIVIDUELLER BAUKASTEN

Will man sein ALM mit Open-Source-Produkten abdecken, steht für die verschiedenen Disziplinen jeweils eine Vielzahl an möglichen Produkten zur Verfügung.

Dank der breiten Auswahl kann das jeweils geeignetste Produkt eingesetzt werden, um eine Aufgabe zu erfüllen. Diese Produkte sind teilweise auf gewisse Programmiersprachen oder Umgebungen spezialisiert. Man muss sich also nicht mit einem Allrounder begnügen, sondern kann ein spezialisiertes Produkt einsetzen. Hier eine Übersicht über die gängigsten Produkte aus den verschiedenen ALM-Disziplinen:

### Source-Code-Management:

- Git (<http://git-scm.com/>)
- Mercurial (<http://mercurial.selenic.com/>)
- SVN (<http://subversion.apache.org/>)

### Issue Tracking/Project Management:

- Bugzilla (<http://www.bugzilla.org/>)
- Mantis (<http://www.mantisbt.org/>)
- Trac (<http://trac.edgewall.org/>)
- The Bug Genie (<http://www.thebuggenie.com/>)

### Continuous Integration:

- Hudson (<http://hudson-ci.org/>)
- Jenkins (<http://jenkins-ci.org/>)
- Cruise Control (<http://cruisecontrol.sourceforge.net/>)
- Cruise Control .NET (<http://www.cruisecontrolnet.org/>)
- TeamCity<sup>1</sup> (<http://www.jetbrains.com/teamcity/>)

### Code Quality:

- Sonar (<http://www.sonarsource.org/>)
- StyleCop (<http://stylecop.codeplex.com/>)
- FxCop<sup>2</sup> ([http://msdn.microsoft.com/en-US/library/bb429476\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/bb429476(v=vs.80).aspx))

---

<sup>1</sup> TeamCity ist nicht Open Source, allerdings bietet der Hersteller JetBrains eine kostenlose Version, mit beschränktem Funktionsumfang an, die auch in kommerziellen Projekten eingesetzt werden kann.

<sup>2</sup> Ist nicht Open Source, aber kostenlos in vollem Funktionsumfang erhältlich.

### 5.3.2 GRENZEN

Eine im Baukastensystem aus Open-Source-Produkten zusammengesetzte Lösung stösst dann an ihre Grenzen, wenn ein hohes Mass an Integration gefordert ist. Zwar ist eine Integration der einzelnen ALM-Komponenten möglich, aber meistens müssen manuell Integrationen erstellt werden. Beispielsweise existiert häufig keine gemeinsame Benutzerverwaltung für die einzelnen Komponenten. Ein Benutzer besitzt also verschiedene Benutzerkonten mit eigenem Passwort.

Nebst der Integration muss auch der Aspekt des Reporting erwähnt werden. Da die verschiedenen Produkte häufig keine einheitliche Schnittstelle bieten, wird auch das Reporting über den ALM-Lebenszykluserschwert. Soll also ein übergeordnetes Reporting erstellt werden, müssen die eingesetzten Produkte einzeln angebunden werden.

### 5.3.3 LÖSUNGSBEISPIEL MIT OPEN-SOURCE-PRODUKTEN

Als Beispiel soll hier ein agil arbeitendes Softwareteam von zehn Personen genommen werden (ein Scrum Master, ein Product Owner, zwei Tester, sechs Entwickler), das in einem C#, ASP.NET-Umfeld Software entwickelt.

#### Folgende Produkte kommen zum Einsatz:

- Versionskontrolle: Git
- Issue Tracking: Trac
- Build Server: TeamCity
- Code Quality: StyleCop, FxCop

### **Beschreibung des Arbeitsablaufs:**

Requirements werden als User Stories in Trac erfasst, priorisiert und einem Sprint zugeordnet. Jede Story wird von Trac mit einem eindeutigen Key versehen. Um die Nachverfolgbarkeit der Requirements zu gewährleisten, wird von den Entwicklern beim «Commit» einer Änderung am Source Code der von Trac erstellte Key als Kommentar angegeben.

Bei jedem Commit wird der Source Code von TeamCity automatisch kompiliert, Unit- und Integrationstest werden ausgeführt und der Code wird durch StyleCop und FxCop überprüft.

Das Management der Projekte, Sprints und Releases wird in Trac erfasst. Innerhalb des Entwicklerteams übernimmt ein Teammitglied bei Bedarf die Rolle des Release Managers. Um am Ende eines Sprints eine installierbare Software zu erhalten und den Stand des Releases festzuhalten, wird durch den Release Manager ein spezieller Build-Prozess in TeamCity gestartet. Mittels eines Build Scripts/ TeamCity wird der aktuelle Stand der Software kompiliert, ein Setup erstellt und als Artefakt in TeamCity abgelegt. Zudem wird die aktuelle Revision als Tag in Git markiert.

Für Reporting-Zwecke können die bei den Commits in Git eingegebenen Kommentare und die Einträge und Statuswechsel in Trac ausgewertet werden.

### **5.3.4 TEAMCITY**

Obwohl TeamCity an sich nicht Open Source ist, wird es häufig eingesetzt, da es standardmässig mit einer freien Lizenz vertrieben wird. Damit kann eine begrenzte Anzahl an Projektkonfigurationen verwaltet werden, was normalerweise für kleinere bis mittlere Projekte ausreicht.

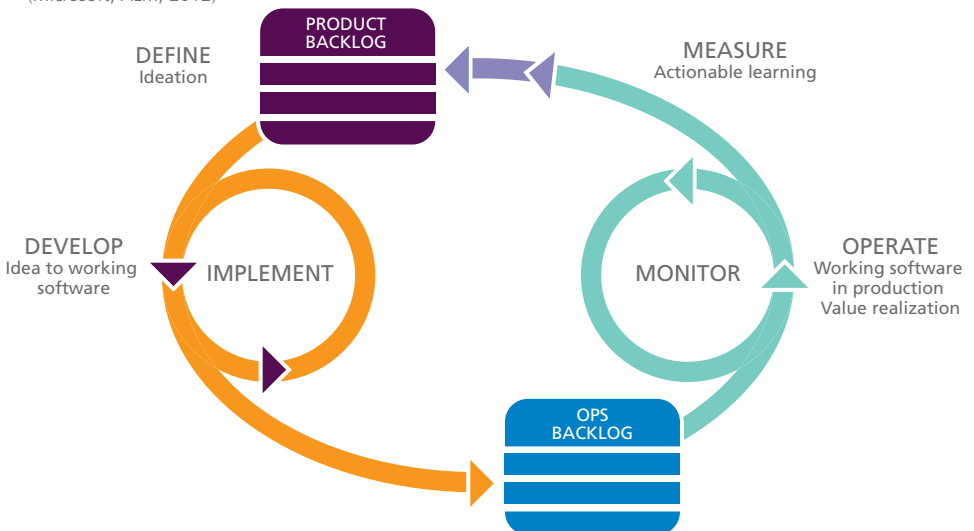
## 6 ALM MIT MICROSOFT

Der Microsoft Visual Studio Team Foundation Server definiert sich als eine Sammlung von verschiedenen Tools, die eine gemeinsame Interaktion mit dem ALM ermöglichen.

### 6.1 LEBENSZYKLUS LAUT MICROSOFT

Bekanntlich existieren unterschiedliche Auffassungen bei der Definition von ALM, so werden je nach Produkt und dessen Herkunft andere Schwerpunkte hervorgehoben. Dies zeigt sich beispielsweise bei Borland, dessen Fokus auf den Metriken liegt, anders bei Polarion, hier steht die Nachverfolgbarkeit im Vordergrund. Der bisher definierte Lebenszyklus von ALM weicht folglich auch bei der Definition von Microsoft leicht ab. Dies erklärt sich dadurch, dass Microsoft vermehrt Wert auf ineinandergreifende Aktivitäten und die Unterstützung unterschiedlicher Prozessmodelle legt. Folgender Zyklus zeigt den von Microsoft definierten Lebenslauf einer Software:

Abbildung 1:  
Application Life Cycle  
(Microsoft, ALM, 2012)





**Define**

Definiert ein gemeinsames Verständnis der Software zwischen Auftraggebern und Auftragnehmer. Daraus entsteht ein Backlog, das nach Priorität umgesetzt werden kann.

**Develop**

Develop beschreibt den eigentlichen Implementierungsvorgang einer Software.

**Operate**

Hier befindet sich die Software im Einsatz und erbringt den angestrebten «Return on Investment».

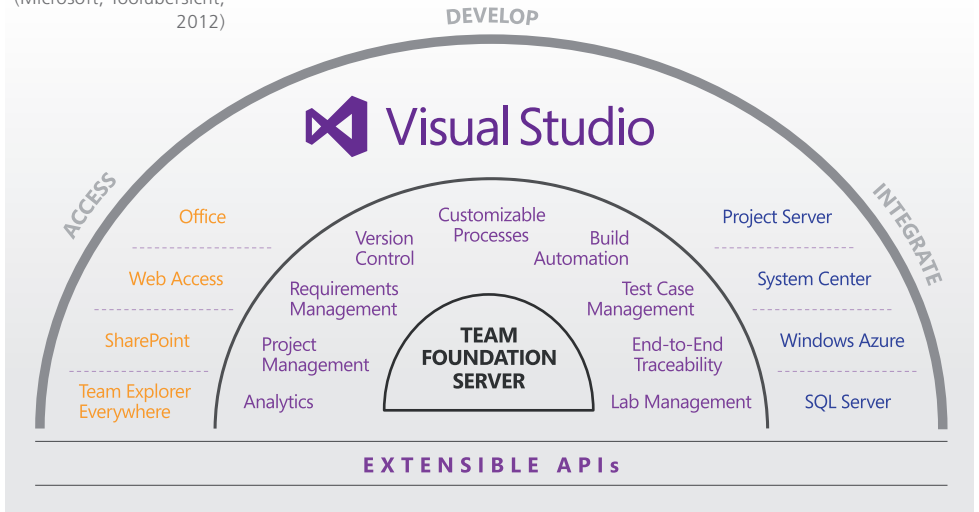
**Measure**

Durch das Feedback aus dem produktiven Einsatz können neue Erkenntnisse ins Backlog einfließen.

## 6.2 TOOLS VON MICROSOFT

Mit den nachfolgenden Tools von Microsoft lässt sich der Lebenszyklus einer Applikation steuern. Die einzelnen Tools bilden sich um den Team Foundation Server:

Abbildung 2:  
Toolübersicht ALM  
(Microsoft, Toolübersicht,  
2012)



### 6.2.1 VISUAL STUDIO

Das Microsoft Visual Studio (VS) ermöglicht einem professionellen Softwareentwicklungsteam eine komplette Plattform, mit der ein durchgängiger Software Life Cycle betrieben werden kann. Die Entwicklungsumgebung enthält sämtliche Kernfunktionen und vereint diverse Aspekte für das Life Cycle Management. Das reicht von den Analyseanforderungen über Architektur bis hin zum Design oder

dem Deployment. Abgerundet wird das VS mit unzähligen Plugins, die zur freien Verfügung stehen. Nennenswert ist hierbei StyleCop, der während der Entwicklung dafür sorgt, dass im Zusammenspiel mit Resharper sämtliche definierte Code-Richtlinien eingehalten werden. Besonders beim Refactoring ist ReSharper von JetBrains hilfreich. Dieser unterstützt massgeblich das stetige Verbessern des Quellcodes.

Mit der neuesten Version von Visual Studio 2012 hat der Entwickler alles zur Hand, um eine Applikation im «Windows 8 Style UI» zu entwickeln. Folgend ist ein kleiner Einblick in die Neuerung von Visual Studio 2012 aufgelistet:

- Anpassung der Optik (der Farben, Icons etc.)
- Möglichkeit zur Entwicklung von Windows-Apps
- Beinhaltet Light Switch mit verbesserter Performance
- Unterstützung für Windows RunTime
- Neuer Solution Explorer
- Verbesserte SQL-Interaktion durch den neuen SQL Server
- Object Explorer
- Quellcode ist während der Laufzeit editierbar

### **6.2.2 TEAM FOUNDATION SERVER**

Der Team Foundation Server (TFS) ermöglicht durch Integration der ALM-Tools eine leistungsfähige Plattform für professionelle Softwareentwicklung. Der TFS bietet unzählige prozessunterstützende Funktionen an, beispielsweise das zentrale Portal der Versionskontrolle, Build-Management oder das Verfolgen sogenannter Work Items.

### Versionskontrolle

Um durchgängig einen hohen Softwarequalitätsstandard zu gewährleisten, ist es unerlässlich, auf bereitgestellte Hilfsmittel wie Continuous Integration oder Check-in-Policies zu setzen. Solche Richtlinien können dazu genutzt werden, dass der Entwickler gezwungen wird, Unit-Test zu schreiben, den Quellcode mit einem Work Item zu assoziieren oder dass eine kontinuierliche statische Codeanalyse durchgeführt wurde.

### Shelves

Ein häufig verwendetes Feature im TFS sind die sogenannten Shelve-Sets. Diese ermöglichen dem Entwickler, seinen aktuell veränderten Quellcode temporär auf dem TFS auszulagern. Dabei gelten keine Check-in-Policies, und beim Quellcode darf es sich durchaus noch um unfertigen Code handeln. Der besondere Nutzen hierbei ist, dass einerseits auf dem zentralen TFS zwischengespeichert wird, was die Datensicherheit erhöht. Andererseits ist es dienlich, falls ein Entwickler kurzfristig nicht an seinem Arbeitsplatz erscheinen kann und ein Mitentwickler auf seinen Code angewiesen ist. Mit wenig Aufwand kann sein Shelve ausgecheckt und mit diesem weitergearbeitet werden.

### Interaktionen mit TFS

Es existieren mehrere Möglichkeiten, wie die Informationen, zum Beispiel einzelne Work Items, vom TFS abgerufen und verwaltet werden können. Der Grund für diese Variationen ist, dass es verschiedene Rollen in einem Projekt gibt, die folglich auch unterschiedliche Tools präferieren.

### **6.2.3 TEAM EXPLORER**

Der Team Explorer wird automatisch mit jeder kostenpflichtigen Version von Microsoft Visual Studio mit installiert. Dieses Tool ermöglicht durch die Verwendung von Visual Studio den Zugriff auf Teamprojekte, Versionskontrolle, Builds und Projektmanagement (z. B. Work Items).

### **6.2.4 TEAM WEB ACCESS**

Web Access ist ein modifizierbares Webinterface, das beinahe sämtliche obigen Features abdeckt. Der Web Access wird automatisch mit dem Team Foundation Server mit installiert.

### **6.2.5 TEAM WEB ACCESS – WORK ITEM VIEW ONLY**

Diese Schnittstelle ist besonders für teamexterne Mitarbeiter gedacht, denn hier können Work Items auch ohne zusätzliche Lizenzkosten verwaltet werden. Die einzige Einschränkung dabei ist, dass nur die selber erstellten Work Items sichtbar sind und bearbeitet werden können.

### **6.2.6 OFFICE**

Microsoft Excel und Microsoft Project erlauben einfache bidirektionale Verwaltung und Organisation von Work Items. Diese Möglichkeit ist direkt nach der Installation verfügbar und benötigt keinen weiteren Installationsaufwand.

### **6.2.7 VISUAL STUDIO TEAM FOUNDATION SERVER REPORTS**

Visual Studio Team Foundation Server Reports ermöglichen jederzeit einen Einblick in den aktuellen Stand des Projekts. So können neben Projektstatus, Testabdeckung, Fehlerrückmeldung auch die Codequalität bis hin zur Ressourcenausnutzung als Report eingesehen werden.

### **6.2.8 MICROSOFT TEST MANAGER**

Der Testmanager von Microsoft dient dem Tester als eine eigenständige Applikation, mit der er die Handhabung von Testfällen ausserhalb des Visual Studios erlangt. Darin verwaltet werden sämtliche Testarten. Nebst den automatisierten Tests erweist sich insbesondere bei den manuellen Testfällen der Testmanager als grosse Hilfe. So werden nicht nur die einzelnen Schritte aufgelistet, sondern es kann auch sogleich ein Video aufgenommen werden, was zusätzlich die Reproduzierbarkeit verstärkt. Dank dem Vorteil, dass der Testmanager direkt mit der TFS-Datenbank integriert, können auch direkt im Testmanager auftretende Bugs erfasst werden.

### **6.2.9 INTERAKTION FÜR WORK ITEMS**

Die Verwaltung der Work Items nimmt eine zentrale Rolle ein, da diese als Kommunikation und Schnittstelle zwischen den einzelnen Rollen dient. Aus diesem Grund wird die Handhabung aus unterschiedlichen Tools ermöglicht. Verdeutlicht wird dies an der folgenden Darstellung:

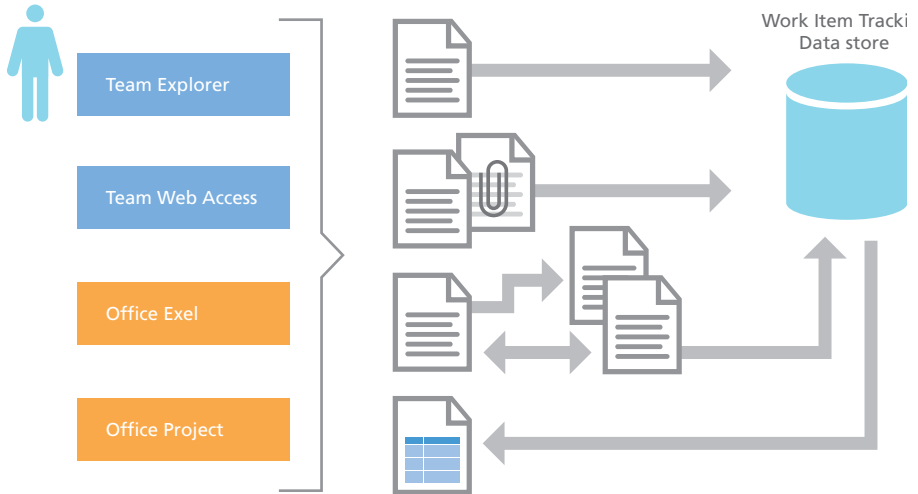


Abbildung 3:  
Möglichkeiten zur  
Verwaltung von Work Items

### 6.3 BEST-PRACTICE «AGILE MIT MICROSOFT»

Mithilfe der Prozessvorlage «MSF for Agile Software Development» können agile Verfahren leichter angewendet werden. Dadurch können Scrum und andere agile Entwicklungsverfahren unterstützt werden.

Microsoft bietet selber drei vorgefertigte Templates an, die umfangreiche Einstellungen in diversen Prozessbereichen beschreiben. Besonders im agilen Umfeld ist «Microsoft Solution Framework for Agile» (MSF for Agile) oder das «Scrum 2.0» vorzufinden. Das Template «Microsoft Solution Framework for Capability Maturity Model Integration» (MSF for CMMI) findet bei strikten formalen Prozessen seinen Einsatz. Sämtliche Templates dienen nur als Grundlage und können beliebig individualisiert oder erweitert werden.

- MSF für Agile beinhaltet zum Beispiel die Workflow-zustände: «Aktiv», «Gelöst» und «Geschlossen», während in der Produktplanung User Story, Produktplanung als Option zur Verfügung stehen. Das Agile Template wird dann verwendet, wenn sich das Team für einen agilen Prozess entscheidet.
- Microsoft Visual Studio Scrum 2.0 (2012) ist ein Template, das für das Team den Prozess im Scrum unterstützt. Dies beinhaltet zum Beispiel Work Items, Reports wie Burndown-Chart und Team Queries, die das Team nutzen kann, um Informationen zu aggregieren und zu analysieren.
- Beim MSF für CMMI gibt es auch vordefinierte Workflow-zustände. Bei der Produktplanung zeigen sich Unterschiede, denn hier werden Anforderung und Änderungsanforderung verwendet. CMMI wird eingesetzt, falls das Team einen strengen Überwachungspfad verlangt und einen strikten und formalen Prozess für sein Änderungsmanagement benötigt.

Sämtliche Templates können direkt im VS unter dem Menüpunkt «Team» heruntergeladen werden. Dort ist der Process Template Manager zu finden, mit dem sich sämtliche Templates verwalten lassen. Die folgende Auflistung gibt einen Einblick, welche agilen Praktiken durch welche Lösung von Microsoft realisiert wurden.

### **Plan a Sprint (Planning Game)**

Das «Planning Game» ist eine der fundamentalen Praktiken beim agilen Verfahren. Alternativ zum VS2012 kann hier das Excel mit der Auflistung des Backlogs als zentrale Informationsquelle genutzt werden. Von diesem priorisierten Excel-Dokument aus können direkt die vom Team geschätzten Aufwände erfasst werden.



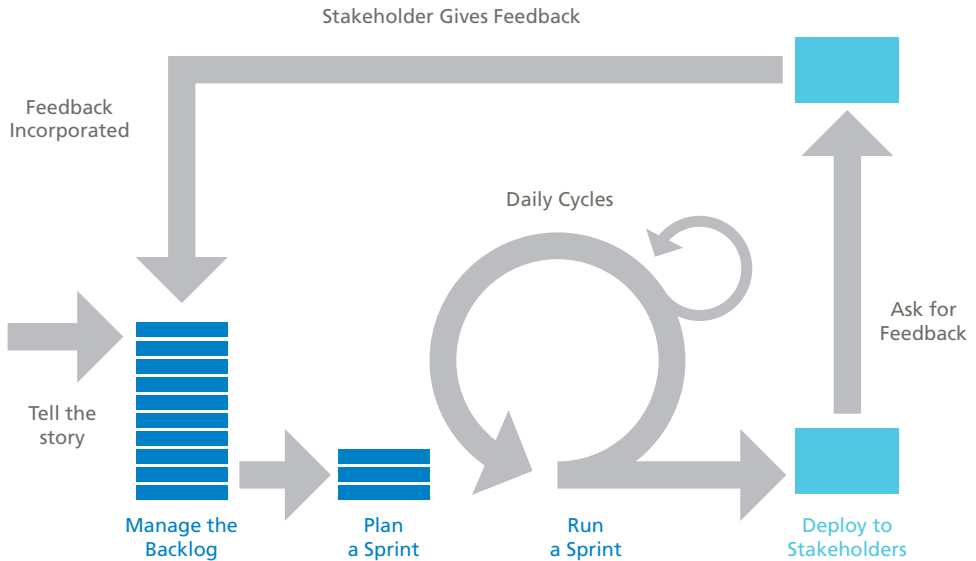


Abbildung 4:  
Visualisierung eines Sprints  
von Microsoft (2012)

### Testing

Test Driven Development (TDD) ist auch bekannt unter «redfirst» und definiert, dass beim Entwickeln von Features der zugehörige Test gleichzeitig mit der Implementierung geschrieben wird. Durch das von Visual Studio integrierte Unittesting ist TDD einfach umsetzbar und unterstützt die Codequalität um ein Vielfaches. Die daraus entstandenen Tests können dann anschliessend mit der User-Story (als Work Item) assoziiert werden.

### Refactoring

Das Umbauen und Verbessern von Quellcode zeigt sich als tägliche Pflicht eines guten Entwicklers. Microsoft unterstützt mit dem Toolset von Visual Studio unzählige Hilfestellungen beim Umbauen von Quellcode. So können zum Beispiel direkt ganze Klassen umbenannt oder verschoben werden.

### Continuous Integration

Team Foundation Build und das Source Control ermöglichen automatische Buildprozesse, die auf Wunsch angepasst und mit Code Coverage oder anderswertigen Anpassungen verwendet werden können.

### Deploy to Stakeholders

Nach der Definition von Agile wird der Fokus auf kleine Releases gelegt. Nach einem Sprint von 2 bis 3 Wochen soll eine potenziell releasefähige Software zur Verfügung stehen. Die Planung dafür kann übersichtlich mittels Work Item als Stories erfasst werden. Während des Sprints kann der Progress durch grafische Reports jederzeit eingesehen werden. So erhalten die Beteiligten Informationen über den aktuellen Projektstand und können Entscheidungen zeitnah fällen.

## 6.4 ALM IN IHRER FIRMA EINFÜHREN

### bbv Academy

Die bbv Software Services möchte ihr reiches Expertenwissen und ihre langjährige Erfahrung in den verschiedensten Softwarebereichen weitergeben und ihren Kunden oder anderen Interessierten zur Verfügung stellen. Aus diesem Grund wurde die bbv Academy ins Leben gerufen, die ein vielfältiges Angebot beinhaltet. Dieses reicht von einer professionellen TFS-Beratung bis hin zu einzelnen Kurstagen oder einer Kurswoche inklusive Zertifikat.

## 7 PERSÖNLICHE ERFAHRUNGEN

Bei der Einführung eines ALM-Prozesses konzentriert man sich vielfach auf die Auswahl der Tools anhand ihres Funktionsumfangs. Das Tool ist aber nur ein Puzzleteil für die erfolgreiche Einführung eines neuen ALM-Prozesses. Vielfach wird der Aufwand für die Anpassung der Arbeitsweise an die neu einzuführenden Tools unterschätzt. Um ein Tool effizient einsetzen zu können, muss der Prozess auf dieses optimiert werden. Für diese Änderungen müssen die Mitarbeiter motiviert, ausgebildet und eventueller Widerstand überwunden werden.

Der Aufwand für diese Standardisierung der Arbeitsweise zahlt sich aus. Die individuellen und unterschiedlichen Vorgehen werden durch ein einheitliches abgelöst. Die Arbeitsweise wird für alle Beteiligten transparenter. Die Teammitglieder können die geleistete Arbeit ihrer Kollegen besser selbständig nachvollziehen und sich gegenseitig besser vertreten.

In einem agilen Entwicklungsumfeld ist es sogar ein Muss, einen gemeinsamen Prozess anzuwenden, um eine gute und effektive Teamzusammenarbeit zu ermöglichen. Der Erfolg der Umstellung der Arbeitsweise ist stark von der Erfahrung der betrauten Person abhängig. Je mehr verschiedene Projekte sie schon begleitet hat, desto besser kann sie eine auf das Projektumfeld abgestimmte Arbeitsweise einführen und passende ALM-Tools identifizieren.

Bei der Auswahl der ALM-Tools stellt sich die Frage, ob Point Solutions oder eine All-in-one-Solution gewählt werden soll. Beide Ansätze haben ihre Stärken und Schwächen. Bei den Point Solutions müssen verschiedene Tools kombiniert werden, um den geforderten Funktionsumfang zu erreichen.

Das Toolset muss so zusammengestellt werden, dass die einzelnen Tools einwandfrei zusammenarbeiten können. Ist Verfolgbarkeit gefordert, müssen sie untereinander verlinkt werden können, z. B. das Referenzieren von Items aus dem Item Tracking Tool beim Speichern von Source Code in das Source-Code-Management-Tool. Point Solutions können Schritt für Schritt, anhand der benötigten Funktionen, installiert werden. Dadurch ist der Initialaufwand für die Einführung der ersten ALM-Funktion klein.

Bei einer All-in-one-Solution muss das gesamte Tool von Beginn weg installiert werden, die Funktionen können aber trotzdem schrittweise in Betrieb genommen werden. Der Wartungsaufwand ist bei den All-in-one-Solutions häufig kleiner, da nur ein Tool unterhalten werden muss und nicht mehrere eventuell von verschiedenen Herstellern mit unterschiedlichen Releasezyklen. Im Gegensatz dazu können bei Best-of-Breed-Lösungen einzelne Tools aktualisiert oder sogar gänzlich ausgetauscht werden. Dies verringert die Downzeit während des Updateprozesses des gesamten Systems und reduziert zudem das Gesamtrisiko.

Bei der Evaluation von ALM-Tools wird vielfach der Fokus auf die Bedürfnisse der Softwareentwickler gerichtet. Die Anforderungen der restlichen Stakeholder sind aber ebenso wichtig. Bei der Festlegung der Evaluationsfaktoren müssen auch Funktionen wie Agile Requirements Engineering, Projekt Controlling, Continuous Deployment, Incident Management etc. einbezogen werden.

Nur die Akzeptanz bei allen Stakeholdern stellt sicher, dass die neuen ALM-Prozesse gelebt werden und somit die Früchte der Einführung eines ALM-Tools geerntet werden können.

# 8 ANHANG



### **8.1 AUTOREN**

#### **Thorsten Ruf ist Diplomingenieur (FH)**

für technische Informatik, seit über 10 Jahren in der Informatikbranche tätig und hält verschiedene Microsoft-Zertifizierungen. Seit 2009 arbeitet er als Senior Software Engineer bei der bbv Software Services und ist verantwortlich für die Einführung und den reibungslosen Betrieb des TFS bei Inhouse-Softwareprojekten. Er unterstützt zahlreiche Kunden sowohl bei der Planung und Einführung als auch bei Migrationsprojekten, bis hin zu individuellen Anpassungen des Visual Studio Team Foundation Servers.



#### **Stefan Wermelinger, MCPD, MCTS TFS**

Stefan hat mehrjährige Erfahrung im Bereich der Softwareentwicklung mit verschiedenen Technologien. Seit 2010 arbeitet er als Software Engineer für .NET bei der bbv Software Services. In dieser Zeit hat er Erfahrungen in der Administration und Benutzung des Team Foundation Servers gesammelt.

Weiter haben folgende Personen massgeblich zur Entstehung dieses Booklets beigetragen:

**Dominik Fretz, MCPD**

**Simon Inderbitzin, MCPD, MCTS TFS**

**Roland Simon, MCPD, MCTS TFS**

## 8.2 QUELLENVERZEICHNIS

### Literaturverzeichnis

- Atlassian. (2012). Products. Abgerufen am 15. 11. 2012 von <http://www.atlassian.com/software>
- Borland Software Corporation. (2012). Abgerufen am 15. 11. 2012 von <http://www.borland.com/>
- IBM. (2012). Agile software development. Abgerufen am 15. 11. 2012 von <http://www-01.ibm.com/software/rational/agile/solutions/>
- JetBrains. (2012). Abgerufen am 15. 11. 2012 von <http://www.jetbrains.com/youtrack/index.jsp>
- Microsoft. (2012). ALM. Schweiz.
- Microsoft. (2012). Toolübersicht. Schweiz.
- Microsoft. (2012). Visual Studio 2012 Product Guide. Schweiz.
- Murphy, T., & Duggan, J. (2012). Magic Quadrant for Application Life Cycle. Gartner Inc.
- Rally Software Development Corp. (2012). Abgerufen am 15. 11. 2012 von <http://www.rallydev.com/>
- SourceGear LLC. (2012). Abgerufen am 15. 11. 2012 von <http://www.sourcegear.com/vault/>
- Telerik. (2012). Abgerufen am 15. 11. 2012 von <http://www.telerik.com/agile-project-management-tools/>
- Visure Solutions. (2012). Abgerufen am 15. 11. 2012 von <http://www.visuresolutions.com/visure-requirementssoftware>



### Abbildungsverzeichnis

Abbildung 1: bbv-Poster Application Life Cycle Management

Abbildung 2: Application Life Cycle (Microsoft, ALM, 2012)

Abbildung 3: Toolübersicht ALM (Microsoft, Toolübersicht, 2012)

Abbildung 4: Möglichkeiten zur Verwaltung von Work Items

Abbildung 5: Visualisierung eines Sprints von Microsoft (2012)

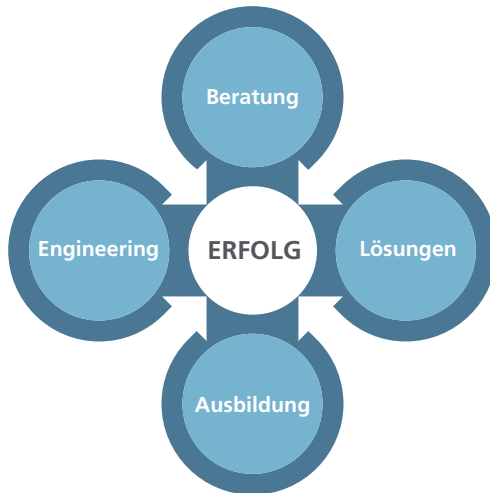
## 9 GLOSSAR

Begriff	Erklärung
SCCM	Das Software Change and Configuration Management erlaubt es, Konfigurationen (eine Menge von Änderungsanforderungen inkl. aller betroffenen Artefakte für einen neuen Release) und alle ihre Änderungen im Zeitverlauf zu kontrollieren. SCCM garantiert die Integrität und Nachverfolgbarkeit aller Konfigurationen und die Möglichkeit der Wiederherstellbarkeit einer älteren, lauffähigen Version.
SaaS	Software as a Service ist ein Teilbereich des Cloud Computings. Das SaaS-Modell basiert auf dem Grundsatz, dass die Software und die IT-Infrastruktur bei einem externen IT-Dienstleister betrieben und vom Kunden als Service genutzt werden.
Best-of-Breed	Steht als Synonym für «das Beste für mein Anwendungsgebiet»
Work Items	Ein Work Item dient zum Tracken von Tasks oder Bugs, das während des Entwicklungszyklus an das Team adressiert werden kann.
Team Queries	Team Queries dienen dazu, dass teamweit vordefinierte Abfragen auf Work Items abgefragt werden können.
Continuous Integration	Definiert eine fortlaufende oder permanente Integration einer Anwendung.
Check-in-Policies	Check-in-Policy ermöglicht Richtlinien, die vor dem Einchecken eingehalten werden müssen. Bsp. positive Codeanalyse.
Return on Investment (ROI)	Beschreibt eine Performancemessung, welche den Profit in Relation zu der Investition bringt.
TDD	Test Driven Development beschreibt eine Praktik, bei der Tests vor den zu implementierenden Komponenten geschrieben werden.
ATDD	Acceptance Test Driven Development ist eine Erweiterung von TDD, bei dem man Akzeptanzkriterien so definiert, dass daraus automatisierte Akzeptanztests erstellt werden.

Begriff	Erklärung
SDLC	Software Development Life Cycle beschreibt den Prozess der Erstellung von Software und der dazugehörigen Tätigkeiten.
IDE	Integrated Development Environment ist eine Entwicklungsumgebung mit einer Sammlung von Werkzeugen, die eine effiziente Entwicklung mit möglichst wenigen Medienbrüchen ermöglichen.
DVCS	Distributed Version Control System – Bezeichnet die neueste Generation von SCM-Tools, die auf einem dezentralen, Peer-to-Peer-Ansatz basieren.
SCM	Source-Code-Management-Tools sind Programme, die es ermöglichen, dass mehrere Softwareentwickler gemeinsam an einer Codebasis arbeiten. Sie speichern den aktuellen Stand des Codes und bieten die Möglichkeit, auf alte Versionen zurückzugreifen. Sie helfen zudem die Frage zu beantworten: Wer hat wann was geändert?



bbv Software Services AG ist ein Schweizer Software- und Beratungsunternehmen, das Kunden bei der Realisierung ihrer Visionen und Projekte unterstützt. Wir entwickeln individuelle Softwarelösungen und begleiten Kunden mit fundierter Beratung, erstklassigem Software Engineering und langjähriger Branchenerfahrung auf dem Weg zur erfolgreichen Lösung.



Unsere Booklets und vieles mehr finden Sie unter  
[www.bbv.ch/publikationen](http://www.bbv.ch/publikationen)

**MAKING VISIONS WORK.**

[www.bbv.ch](http://www.bbv.ch) · [info@bbv.ch](mailto:info@bbv.ch)