



BOOKLET

**GUIDELINES
FÜR EINE ERFOLGREICHE
SCRUM-EINFÜHRUNG**

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

Kontakt Deutschland

bbv Software Services GmbH
Agnes-Pockels-Bogen 1
80992 München
Telefon: +49 89 452 438 30
E-Mail: 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.	Grundsätze	7
2.1	Geschäftswertgetriebenes Vorgehen	8
2.2	Kontinuität	9
2.3	Inspektion und Adaption	9
3	Der Start mit Scrum	11
3.1	Grundsätze des Veränderungsmanagements beachten	12
3.2	Notwendigkeit kommunizieren	12
3.3	Mitstreiter identifizieren	13
3.4	Besuch im Holodeck	13
3.5	Eignet sich Scrum für Ihre Herausforderung?	14
3.6	Legen Sie los!	16
3.7	Passendes Team finden	16
3.8	Managementunterstützung	17
4	Das erste Scrum-Projekt	18
4.1	Scrum-Guide	19
4.2	Vision	19
4.3	Rollen besetzen	20
4.4	Konstanz bei der Teamzusammensetzung	22
4.5	One size fits all?	23
4.6	Initiales Backlog	24
4.7	Definition of Done	25
4.8	Infrastruktur und Umgebung	26
4.9	Scrum Board	27
4.10	Agile Gadgets	28
4.11	Software und Dokumente	29
4.12	Training	30

5	Planung	31
5.1	Aufwandschätzungen	32
5.2	Teamgeschwindigkeit messen	32
5.3	Budget und Zeitrahmen definieren	33
5.4	Geschäftswertgetriebenes Vorgehen	33
5.5	Sprint-Planung	34
5.6	Product Backlog Refinement	36
5.7	Release-Planung	36
6	Erfolgsfaktoren	38
6.1	Get a coach!	39
6.2	Einbezug von TDD-Gurus und agilen Architekten	39
6.3	Agiles Requirements Engineering	41
6.4	Investieren Sie in gute User Stories	42
6.5	Kombination mit Kanban	43
6.6	Messbarkeit der Fortschritte	43
6.7	Erfolge feiern	44
6.8	Einbindung des Kunden	44
6.9	Agiles Management	45
7	Fazit	47
8	Anhang	49
8.1	Fachleiter Agile	50
8.2	Literatur	51

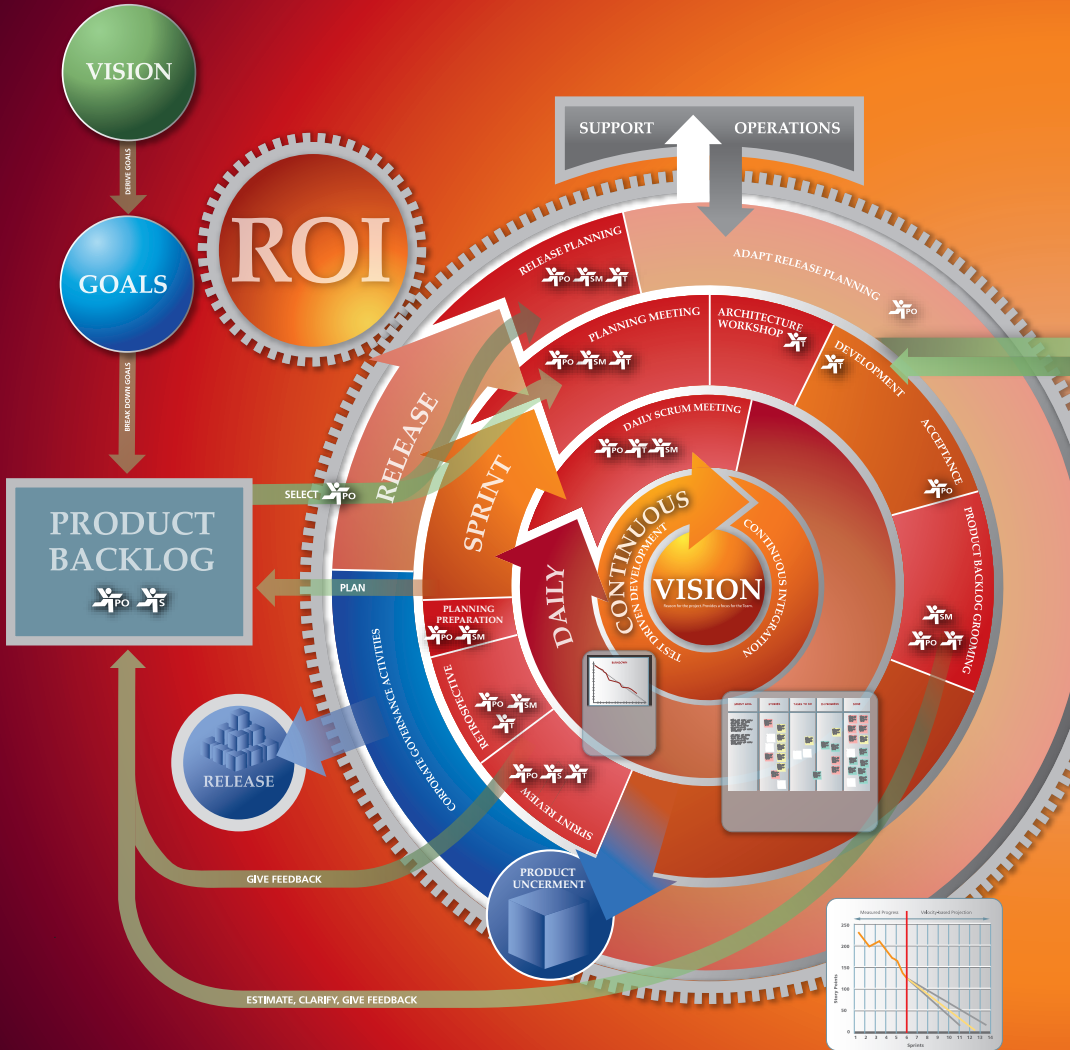
1 EINLEITUNG

Scrum ist en vogue. Gemäss aktuellen Studien sind agile Prozesse und Scrum bereits in der Mehrheit aller IT-Unternehmen eingeführt [1]. Die Einführung von Scrum gehört heute zum Standardrepertoire vieler IT-Berater und eine grosse Auswahl an Literatur beleuchtet die Thematik in ihren vielen Facetten.

Dieses Booklet fasst die Erfahrungen von bbv Software Services in der Einführung von Scrum zusammen und gibt eine Hilfestellung im Aufsetzen von Projekten mit Scrum. Im Booklet finden Sie Hinweise auf wichtige Aspekte in der Scrum-Einführung und konkrete Praxis-Tipps aus den Erfahrungen der letzten Jahre.

SCRUM THROUGHPUT DRIVEN

CYCLE-TIME DRIVEN KANBAN



BAN

2 GRUNDSÄTZE

Die Einführung von Scrum hat es in sich. Die Einfachheit der Regeln aus dem Scrum-Regelwerk «Scrum Guide» [2] täuscht über die Komplexität der tatsächlichen Umsetzung hinweg. Schnell ist man daher versucht, das Scrum-Regelwerk anzupassen, um es besser ans eigene Umfeld zu adaptieren. Damit umschifft man komplexes Änderungsmanagement im Unternehmen, profitiert aber meist auch nicht von den Vorteilen, die Scrum und Agilität versprechen. Man spricht bei halbherzig umgesetzten Agilitäts-Transformationen oft von «Water-Scrum-Fall» oder «Scrum-but»-Prozessen, die manchmal mehr neue Probleme schaffen, als sie zu lösen gedenken.

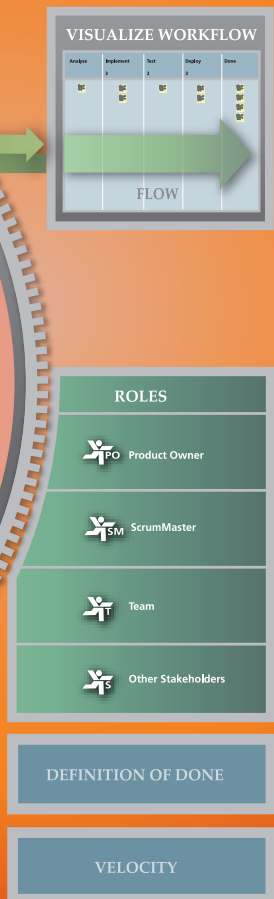


Abbildung 1: Überblick über den Scrum-Prozess

Grundsätzlich wird in agilen Projekten die gleiche Arbeit erledigt wie im klassischen Vorgehen. Gewisse Tätigkeiten fallen weg, andere werden in anderer Art und Weise umgesetzt. Wesentlicher Treiber der Agilität sind das geschäftswertgetriebene Vorgehen, die kontinuierliche Beteiligung aller Disziplinen im Produktentwicklungsprozess und die Inspektion und Adaption des eigenen Entwicklungsvorgehens zwecks stetiger Produktivitätssteigerung.

2.1. GESCHÄFTSWERTGETRIEBENES VORGEHEN

Im Wissen um konstant limitierte Ressourcen zur Produktentwicklung wird der Priorisierung in Scrum grosse Beachtung geschenkt. Die Tätigkeiten und Scrum-Regeln zielen auf eine Priorisierung der Tätigkeiten auf allen Ebenen hin. Meetings sind zeitlimitiert, was die Teilnehmer zur Fokussierung auf die wichtigsten Themen zwingt. Von den Stakeholdern wird eine kompromisslose Priorisierung der Funktionalitäten eingefordert. Die Limitierung der Entwicklungsabschnitte auf maximal vier Wochen zwingt die Projektverantwortlichen zu einer effektiven Priorisierung nach Geschäftswert, die über die bekannten Muss-, Soll-, Kann-Kriterien hinausgeht. Das damit verbundene Herunterbrechen von grossen Funktionalitätsblöcken in kleinere oder einfachere Teilfunktionen erzwingt auch Überlegungen zum möglichst schnellen Realisieren von Geschäftswerten.

Bei der geschäftswertgetriebenen Priorisierung kommen sowohl klassische Priorisierungstechniken zum Einsatz wie auch deren agile Ausprägungen (Priority Poker [3], «Buy a feature» [4]).

2.2 KONTINUITÄT

Der zweite Grundaspekt in agilen Prozessen ist die Kontinuität. Alle Tätigkeiten in agilen Projekten finden kontinuierlich statt. Es gibt keine abgeschlossenen Phasen, in denen nur ein Teilaspekt der Produktentwicklung stattfindet. Anforderungserhebung wird kontinuierlich betrieben, Architekturen werden laufend neu beurteilt und mittels Refactoring den sich ändernden Bedürfnissen angepasst, eingebettete und fortlaufende Qualitätssicherung vermeidet nachgelagerten Testaufwand. Besonders im Bereich Architektur und Anforderungserhebung wirkt sich die agile Vorgehensweise positiv aus. Aus vergangenen Projekten haben wir gelernt, dass man nie alles im Voraus wissen und vorhersehen kann. Insofern ist es auch nicht möglich, vorgängig eine Architektur zu definieren, die alle künftigen Anforderungen optimal bedient. Wir haben gelernt, dass man mehr weiss, nachdem man sich über Tage, Wochen, ja sogar Monate intensiv mit einem Thema oder einer Technologie beschäftigt hat. Agile Projekte nutzen diesen Effekt und gehen in kleinen Schritten vor, in denen an allen Bereichen stetig weitergearbeitet wird. Trotz dieser schrittweisen Vorgehensweise wird dank enger Zusammenarbeit mit dem Kunden und vorausschauender Releaseplanung das Gesamtbild nicht aus den Augen verloren.

Den Gesamtprozess in allen Bereichen kontinuierlich über längere Zeit weiterzutreiben und einzuhalten, bildet eine der grössten Herausforderungen in agilen Projekten und in Scrum. Gleichzeitig stellt die kontinuierliche Verbesserung aller Prozessaspekte auch eine der Kernstärken von Scrum dar.

2.3 INSPEKTION UND ADAPTION

Die Kontinuität findet sich auch im Verbesserungsprozess. Eine wiederkehrende periodische Inspektion ist im Grundkorsett von Scrum

als Retrospektive etabliert. Die Retrospektive findet nach jeder Entwicklungsiteration statt. Ziel der Retrospektive ist die Verbesserung des eigenen Entwicklungsvorgehens. In einem gezielten Meeting mit allen Teammitgliedern wird analysiert, welche Lehren aus der letzten Iteration gezogen werden können und welche Massnahmen zur Verbesserung des Vorgehens und des Umfelds eingeleitet werden müssen. Mit der Inspektion und der darauffolgenden Adaption an die Erkenntnisse wird ein kontinuierlicher Verbesserungsprozess etabliert. Je schneller das Feedback zu möglichen Verbesserungen vorliegt, umso besser. Im Sinne eines Zinseszins-effekts profitieren die künftigen Tätigkeiten und Aktivitäten im Projekt umso mehr, je früher die Verbesserung umgesetzt wird. Entsprechend bedeutungsvoll ist die Durchführung der Sprint-Retrospektive bereits ab dem ersten Sprint.

Als weiteres Beispiel eines bei Scrum integrierten schnellen Feedbackloops fungiert das agile Testen. Scrum propagiert, dass neue Funktionalität bereits während des Sprints Tests unterzogen wird. Die so gewonnenen Erkenntnisse fliessen sofort in die Entwicklung ein. Qualitätssicherung ist Aufgabe des ganzen Teams. Entwickler erstellen die Software testgetrieben, indem sie zuerst einen fehlschlagenden Test schreiben und erst danach die Implementation, die den Test erfolgreich macht. Tester sind ins Team integriert und komplettieren die Testingaktivitäten mittels explorativem Testen, End-to-end Tests und Tests für nicht-funktionale Anforderungen.

Um das Projekt zum Erfolg zu führen, ist einiges an Hartnäckigkeit gefordert. Nur ein kontinuierliches und konsequentes Umsetzen von Verbesserungsmassnahmen garantiert die Fortschritte, die nötig sind, um das Projekt und das Team weiterzubringen und grösstmöglichen Kundennutzen zu erzielen.

3 DER START MIT SCRUM

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

3.1 GRUNDSÄTZE DES VERÄNDERUNGSMANAGEMENTS BEACHTEN

Sie spielen mit dem Gedanken, Scrum in Ihrem Unternehmen einzuführen? Dann betrachten Sie Scrum als Change-Management-Vorhaben [5] und bereiten Sie sich entsprechend vor. Scrum beinhaltet in letzter Konsequenz mehr als nur einige neue Meetings, Rollenspiele und eine iterative Umsetzung. Scrum wird Sie dazu zwingen, tiefgreifende Veränderungen im Vorgehen, Verhalten und sogar in der Kultur in Ihrem Unternehmen umzusetzen.

Scrum führt zu mehrschichtigen Veränderungen in der Organisation. Neben dem Projektvorgehen an sich ändern sich die Rollen, deren Tätigkeiten und Verantwortung. Ihre Entwickler werden gefordert sein, neue Technologien, Tools, Patterns und Frameworks einzusetzen und ihre Programmiertechniken weiterzuentwickeln. Requirements Engineers sind gezwungen, bisherige Denkmuster über Bord zu werfen und Spezifikationen schrittweise ohne abschliessenden Charakter zu erstellen und zu pflegen. Tester werden ganz nah mit den anderen Rollen zusammenarbeiten und sind gefordert, sehr schnelles Feedback zu nicht vollständig abgeschlossenen Arbeiten abzugeben.

3.2 NOTWENDIGKEIT KOMMUNIZIEREN

Aus welchem Grund wollen Sie Scrum einführen? Mit dieser Frage werden Sie unweigerlich konfrontiert. Scrum als Selbstzweck wird keine Akzeptanz finden. Reflektieren Sie Ihre Motivation, eine Veränderung herbeizuführen, und sammeln Sie die Argumente, weshalb die aktuelle Situation nicht zufriedenstellend ist. Wo drückt der Schuh? Weshalb sind die eigenen Projekte eben nicht die Erfolgsgeschichten, die sie sein sollten? Was sind die Konsequenzen, wenn Sie mit dem bisherigen Vorgehen weiterarbeiten? Wenn

Sie diese und ähnliche Fragen für sich beantwortet haben, teilen Sie Ihre Sicht der Probleme mit den Kollegen und Kolleginnen im Unternehmen. Schaffen Sie ein Problembewusstsein und ein Gefühl von Handlungsbedarf.

3.3 MITSTREITER IDENTIFIZIEREN

Sie wollen Scrum im Alleingang einführen? Das führt in den seltensten Fällen zum Erfolg. Mit hoher Wahrscheinlichkeit stören sich nicht nur Sie an den heutigen Problemen und Unzulänglichkeiten in der Produktentwicklung Ihres Unternehmens. Mit hoher Wahrscheinlichkeit sind auch andere Mitarbeiter der Auffassung, agile Prozesse wie Scrum wären ein möglicher Weg zur Verbesserung der Situation. Finden Sie diese Mitarbeiter und gewinnen Sie Mitstreiter für das Vorhaben einer Transformation hin zur Agilität.

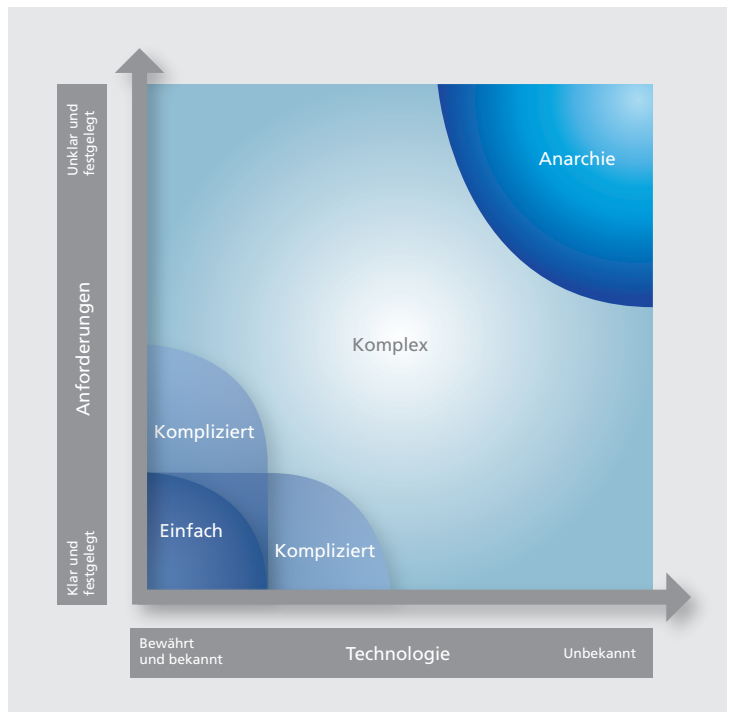
3.4 BESUCH IM HOLODECK

Scrum verkaufen Sie im Unternehmen nicht nur mit dem Aufzeigen der aktuellen Problemlandschaft. Sie müssen auch aufzeigen, wohin die Reise gehen soll und welches die Vorteile einer veränderten Arbeitsweise sind. Erfolgsgeschichten zu Scrum gibt es zur Genüge [6]. Es gibt grosse und kleine Unternehmen, die den Schritt zur Agilität vollzogen haben. Suchen Sie einen Erfahrungsbericht einer möglichst vergleichbaren Situation, zeigen Sie die Vorteile auf und skizzieren Sie die Vision, wie es auch in Ihrem Bereich funktionieren könnte. Damit gewinnen Sie weitere Fürsprecher und die Unterstützung im Management. Hilfreich bei der Gewinnung von Allianzpartnern erweist sich auch das Aufzeigen konkreter, schnell erreichbarer Ziele («Quick Wins»).

3.5 EIGNET SICH SCRUM FÜR IHRE HERAUSFORDERUNG?

Grundsätzlich stellt sich die Frage, ob Scrum als konkrete Ausprägung von Agilität für Ihr Vorhaben und Ihre Organisation überhaupt das geeignete Vorgehen ist. Als Voraussetzung sollte eine bestimmte Grösse und Komplexität Ihrer Projekte oder Ihrer Entwicklungsabteilung gegeben sein. Scrum eignet sich sehr gut für komplexe und vielschichtige Projekte. Für einfache Projekte mit bewährten Technologien und klaren Anforderungen können andere Vorgehensweisen effizienter sein. Für Projekte oder Teams, die aus nur

Abbildung 2:
Scrum eignet sich für
komplexe Projekte



zwei Entwicklern bestehen, ist Scrum ein Overkill. Kleine Teams funktionieren häufig sehr gut ohne formalen Prozess, skalieren aber auch nicht, wenn Engpässe entstehen.

Softwareprojekte fallen meist in die Kategorie «komplex». Innovative Produktentwicklungen vieler Organisationen sind geprägt von Abhängigkeiten innerhalb des Systems sowie zu verschiedenen Umsystemen, noch unklaren Anforderungen der Anspruchsgruppen oder neuen, unwägbareren Technologien. Softwareprojekte nehmen häufig einen Umfang ein, der nicht selten mehrere Dutzend Entwickler umfasst. Scrum adressiert just die Bedürfnisse solcher Entwicklungsvorhaben und gilt dank seiner Skalierbarkeit («Scrum of Scrums») auch als guter Kandidat für grosse Projekte. Dies gilt häufig auch für Vorhaben, bei denen bestehende Produkte abgelöst werden sollen. Nicht selten ist auch bei diesen Produkten durch versteckte Anforderungen nicht klar, was diese wirklich für Funktionalitäten in sich bergen, weil die Spezifikation nicht nachgeführt wurde oder die damaligen Know-how-Träger das Unternehmen verlassen haben. Ein weiteres Entscheidungsmerkmal für oder gegen Scrum ist die Planbarkeit der anfallenden Tätigkeiten. Scrum eignet sich hervorragend für strategische Entwicklungsvorhaben mit klarer Zielsetzung und Stossrichtung. Sind Ihre Tätigkeiten dagegen ereignisgetrieben, eignet sich Scrum wegen der Verzögerungszyklen durch fixe Sprints schlecht. Für betriebliche Entwicklungsaufgaben oder Wartungsaufgaben ist Kanban oder gegebenenfalls eine Mischform die passendere agile Vorgehensform.

3.6 LEGEN SIE LOS!

Zur Einführung von Scrum sind oft nicht alle Voraussetzungen optimal erfüllt. Eventuell ist das Team noch nicht vollständig, der Scrum-Einführungskurs noch nicht besucht worden oder es stehen noch keine optimalen Räumlichkeiten zur Verfügung. Dies soll Sie nicht daran hindern, mit Scrum zu starten. Viel wichtiger und prägender als optimale Voraussetzungen ist es, erste eigene Erfahrungen mit Scrum zu machen. Erfahrungen, ob positiv oder negativ, fließen durch die Retrospektive am Ende des Sprints in den Prozess ein. Durch die stetige Analyse der Rückflüsse aus der Retrospektive identifizieren Sie Verbesserungspotenziale. Während der erste Sprint läuft, besteht bereits die Gelegenheit, einige Hindernisse hin zum idealen Zustand beiseitezuräumen. Wesentlich ist dabei, dass die Beseitigung von Hindernissen auch tatsächlich und bewusst stattfindet. Die Aufnahme von Hindernissen als hoch priorisierte Einträge auf dem Product Backlog ist dabei eine mögliche Vorgehensweise, Hindernissen die nötige Beachtung zu schenken. Als Alternative bietet sich das Führen eines separaten Verbesserungsbacklogs, oft auch Impediment Backlog genannt, an.

3.7 PASSENDES TEAM FINDEN

Es hat sich bewährt, Scrum in einem ersten Team einzuführen, um auch schnell Erfolge ausweisen zu können und Erfahrungen zu sammeln. Am besten eignen sich Teams, die bereits «End-to-end»-Funktionalitäten erstellen, also weitgehend eigenständig vollständige Applikationen entwickeln können. Ist dies nicht gegeben, arbeiten Sie während der Entwicklung mit simulierten Software-Schnittstellen zu den Systemteilen anderer, noch nicht nach Scrum arbeitender Entwicklungsteams. In einem nächsten Schritt können Sie Projektteams bilden, die über mehrere Organisationseinheiten hinweg ganze Systeme entwickeln können.

3.8 MANAGEMENTUNTERSTÜTZUNG

Die Einführung von Scrum erfordert die Unterstützung des Managements. Dabei reicht es nicht, nur passiv «grünes Licht» für ein Pilotprojekt von einem Vorgesetzten zu erhalten. Mitglieder des höheren Managements müssen die Prozessveränderung aktiv unterstützen und öffentlich dafür Stellung beziehen. Nur so haben Sie den nötigen Rückhalt, die anstehenden Veränderungen durch Scrum auch umsetzen zu können.

4 DAS ERSTE SCRUM-PROJEKT

Sie haben ein erstes Projekt identifiziert, das Sie mit Scrum umsetzen wollen? Sie haben die aktive Unterstützung durch das Management? Dann steht dem Start nichts im Weg. Im Folgenden werden die Kernaspekte erläutert, die bei der Scrum-Einführung zum Erfolg führen.

4.1 SCRUM-GUIDE

Als Grundsatz bei der Scrum-Einführung gilt, die Kernelemente von Scrum, beschrieben im «Scrum Guide» [1], konsequent umzusetzen und nicht schon im Vorfeld durch Kompromisse aufzuweichen. Es sind vielmehr die Herausforderungen von den mit Scrum verbundenen Organisations-, Prozess- und Verhaltensanpassungen seriös in Angriff zu nehmen. Die Grundregeln von Scrum bilden die Essenz und dienen als klare Rahmenbedingungen für eine Transformation hin zur Agilität. Es ist nicht ratsam, eine Einführung von Scrum damit zu beginnen, die Grundregeln nach eigenem Gusto abzuwandeln und die eigene agile Geschmacksrichtung im Unternehmen zu etablieren, ohne zu wissen, ob diese experimentelle Abwandlung überhaupt den gewünschten Effekt erzielt. Nur im Ensemble entfalten die Scrum-Methodiken und -Prozeduren die gewünschte Wirkung. Ein Zuschneiden von Scrum ist allenfalls für erfahrene Scrum Master und Trainer ein mögliches Szenario. Diese sind auch mit anderen agilen Prozessen vertraut und wissen um die negativen Auswirkungen von Anpassungen und wie mögliche Gegenmassnahmen aussehen. Scrum-Novizen orientieren sich dagegen am Scrum Guide, der Best Practices und jahrelange Erfahrungen kondensiert in klaren Regeln beschreibt.

4.2 VISION

Scrum-Projekte fordern das Vorliegen einer Produktvision. Die Vision ist der Kompass für das Scrum-Team und dient dazu, die nötigen Grundlagen für das Projekt in der Hand zu haben, um während des Projekts selbstständig und selbstorganisiert Entscheidungen treffen zu können.

Die Vision beschreibt wesentliche Fragestellungen wie:

- Wieso baut man dieses Produkt?
- Welches Problem löst das Produkt?
- Welche Funktionen und Vorteile gegenüber existierenden Produkten soll das Produkt im Endausbau bieten?
- Wer benutzt das Produkt?
- Was hebt dieses Produkt besonders von den Mitbewerbern ab?
- Was ist nicht Teil des Produkts?

Eine Produktvision mit den Anspruchsgruppen zu erarbeiten, ist eine gute Einstimmung auf die kommende Arbeit mit Scrum. Bereits für die Erarbeitung der Vision werden die Akteure dazu angehalten, sich auf Kernaspekte zu fokussieren und zu priorisieren, zwei wesentliche Merkmale von Scrum. Die Vision kann in Form einer virtuellen Presseerklärung, einer Produktbox, eines Datenblatts oder eines Elevator Pitch's¹ erfasst werden. Wichtig ist, sich dabei kurz zu fassen und die Essenz des Projekts auf den Punkt zu bringen.

4.3 ROLLEN BESETZEN

Identifizieren Sie Mitarbeiter für die Rollen in Scrum. Als Product Owner bieten sich beispielsweise bisherige Gesamtprojektleiter, Requirements Engineers oder Product Manager an. Wichtig ist, dass diese Person Entscheidungskompetenz und ein hohes Mass an Verfügbarkeit für das Projekt vereint. Die Product-Owner-Rolle in Scrum ist eine Schlüsselstelle für den Erfolg von Scrum-Projekten und bedarf eines hohen Masses an Engagement für das Projekt während der ganzen Laufzeit. Je nach Komplexität des Geschäftsfeldes ist es auch zulässig, ein Product-Owner-Team zu bilden.

¹ 30 Sekunden dauerndes Verkaufsgespräch während einer Aufzugsfahrt

Vergessen Sie dabei nicht, einen Chief Product Owner (CPO) als Leiter der Gruppe zu definieren. Dieser übernimmt die Verantwortung für die übergeordnete Feature-Priorisierung im Team.

Das Entwicklungsteam ist der Kern bei Scrum. Die Anforderungen an das Team respektive die Teammitglieder sind hoch. Scrum-Projekte arbeiten mit «empowered teams». Das Team hat eine Managementrolle und wird ermächtigt, Entscheidungen zur Erreichung der Vision zu treffen und Verantwortung zu übernehmen. Dazu zählt der Qualitätsaspekt. Das Team ist für die Sicherstellung der vereinbarten Qualität verantwortlich. Qualitätssicherungsfachkräfte sind daher ins Team integriert und stellen sicher, dass die entwickelten Funktionen am Ende des Entwicklungssprints Produktionsreife erlangt haben.

Bei der Zusammensetzung des Teams ist es wichtig, nicht einfach eine Handvoll Einzelkämpfer zu einer Gruppe zu formieren. Wichtiger als das reine Fachwissen sind die Einstellung der Teammitglieder und die Bereitschaft, im Team Verantwortung zu übernehmen und agile Engineering-Praktiken anzuwenden. Das Team ist in Scrum für die Entwicklung der Funktionalität und die Qualität der Software verantwortlich. Damit es diese Verantwortung wahrnehmen kann, muss es funktionsübergreifende Fähigkeiten haben. Design, Architektur, Programmierung und Qualitätssicherung sind wesentliche Fähigkeiten, die ein Entwicklungsteam besitzen muss. Das Team organisiert und managt sich selbst. Es gibt keinen Manager, der die Aufgaben des Teams plant und für das Team Dinge organisiert. Insofern ist jedes Teammitglied gefordert, die anfallenden Themen anzupacken und gemeinsam mit den anderen Teammitgliedern nach Lösungen zu suchen. Die Scrum-Rolle des Scrum Masters kann dabei dem Team behilflich sein. Er sagt den Teammitgliedern

nicht, was sie zu tun haben, aber er weist sie ständig auf mögliche Risiken, Fehler oder Verbesserungsmöglichkeiten hin und hilft bei der Beseitigung von Hindernissen und Unzulänglichkeiten. Der Scrum Master agiert als Coach und Motivator des Teams, im Gegensatz zur heutigen Rolle eines Teamleiters, der häufig Arbeitspakete vorbereitet und verteilt. Der Scrum Master schützt das Team vor Eingriffen wie der direkten Auftragsvergabe von nicht eingeplanten Funktionalitäten. Er stellt sicher, dass die Grundregeln von Scrum eingehalten werden. Zudem unterstützt der Scrum Master das Team bei Themen, die über den Aktionsradius des Teams hinausgehen. In seiner Rolle kümmert er sich um teamübergreifende organisatorische Aspekte, bei Bedarf ersucht er um Hilfe beim Management. Beim Scrum Master ist es keineswegs so, dass diese Person durch einen erfahrenen Entwickler besetzt werden muss. Der Scrum Master ist ein Motivator, ein Change Agent, der nicht zwingend in der Entwicklungsmannschaft zu suchen ist. Der Scrum Master kennt die Scrum-Regeln und verfügt über ein hohes Mass an Sozialkompetenz sowie Erfahrung in der Etablierung von Prozess- und Verhaltensänderungen in Teams.

4.4 KONSTANZ BEI DER TEAMZUSAMMENSETZUNG

Ein gutes und leistungsfähiges Team durchläuft gemäss Tuckman [7] einen Entwicklungsprozess mit folgenden Phasen:

- Orientierungsphase **«Forming»**
- Konfrontationsphase **«Storming»**
- Kooperationsphase **«Norming»**
- Leistungsphase **«Performing»**

Das Team möglichst schnell in die Leistungsphase zu bringen, ist eine der zentralen Aufgaben des Scrum Masters.

Wenn man es einmal geschafft hat, ein leistungsfähiges Team zusammenzustellen, sollte man die einzelnen Teammitglieder möglichst nicht gleich wieder auswechseln. Wir haben die Erfahrung gemacht, dass die Produktivität des Teams stark reduziert wird, wenn in einem 6- bis 8-Personen-Team auch nur zwei Personen ausgewechselt werden. Der Produktivitätsverlust erklärt sich unter anderem durch teamspezifische Konventionen, die mit neuen Teammitgliedern erst wieder etabliert werden müssen. Erst nach mehreren Sprints entfaltet das Team wieder seine volle Leistung.

Dabei ist es keineswegs so, dass ein Entwicklungsteam nur aus erfahrenen Entwicklern bestehen darf. Erfahrungsgemäss harmonisiert eine gute Durchmischung aus erfahrenen und jungen Ingenieuren am besten.

4.5 ONE SIZE FITS ALL?

Legen Sie bei der Teambildung Ihr Augenmerk auf die Teamgrösse. Orientieren Sie sich bei der Teamgrösse am Richtwert 7 ± 2 Teammitglieder. Nicht berücksichtigt sind bei dieser Zählweise Scrum Master und Product Owner, sofern diese nicht auch eine Entwicklerrolle übernehmen. Scrum orientiert sich hier an den Teamgrössen typischer Mannschaftssportarten wie Handball (sieben Spieler) oder Basketball (fünf Spieler). Vermeiden Sie Entwicklungsteams, die aus weniger als drei Entwicklern bestehen. Bei solchen Mikroteams besteht die Gefahr, dass nicht mehr alle Fertigkeiten vorhanden sind, um sämtliche anfallenden Aufgaben zu erfüllen. Zudem benötigen Sie bei Mikroteams nicht die formalen Elemente von Scrum, um die Kommunikation sicherzustellen.

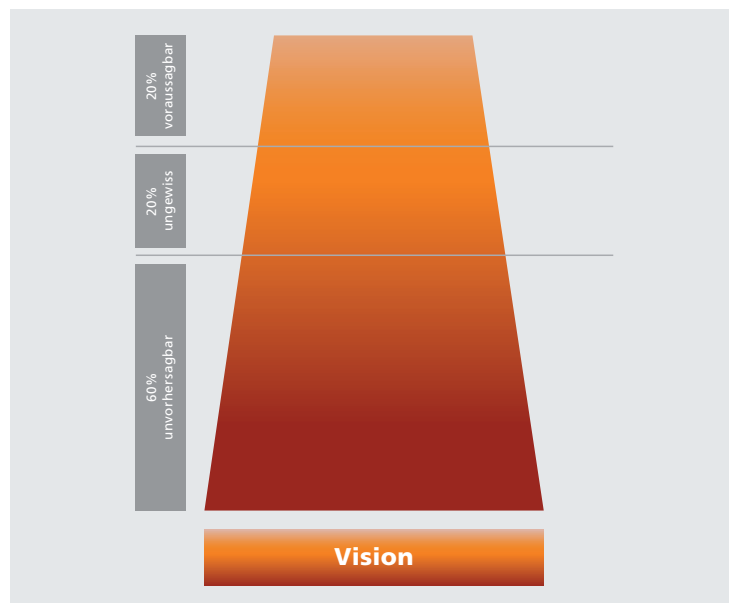
Erreicht Ihr Entwicklungsteam eine Grösse von neun Personen, sollten Sie eine Aufteilung in zwei Feature-Teams anstreben. Der

Koordinations- und Kommunikationsaufwand im Team steigt ansonsten überproportional. Bei Feature-Teams handelt es sich um Teams, die am gleichen Produkt über alle Technologieschichten hinweg Funktionalitäten entwickeln. Statt eines grossen Teams verfügen Sie über mehrere Teams, die Funktionalitäten für das ganze Produkt entwickeln können. Durch die durchgängige Umsetzung von Funktionalitäten minimieren Sie die Abhängigkeit zwischen den Teams und stellen sicher, dass jedes Team Produktfunktionen umsetzen kann.

4.6 INITIALES BACKLOG

Unter der Leitung des Product Owners wird im Rahmen von Workshops zusammen mit den Stakeholdern ein initiales Backlog erarbei-

Abbildung 3:
Zu Beginn sind nur die hoch priorisierten Features im Product Backlog detailliert beschrieben



tet. Das initiale Backlog ist eine Sammlung von Features, die für das erste Release gefordert werden. Unter Release versteht man dabei das effektive Ausrollen des Produkts an den Markt. Das erste Release umfasst die minimal nötigen Funktionen, um mit einem Produkt auf den Markt zu gehen. Mittels Priorisierungsmethoden fordern Sie eine klare Priorisierung der Features von den Stakeholdern ein. Diese Priorisierung ist die Grundlage zur weiteren Detaillierung jener Features mit der höchsten Priorität. Diese werden in weiteren Workshops vertieft, sodass eine erste Liste an Epics und User Stories resultiert, bereit für die Umsetzung in den ersten Sprints.

4.7 DEFINITION OF DONE

Vor der Umsetzung im Sprint müssen die Randbedingungen gesetzt werden, unter welchen das Produkt entwickelt werden soll. Gibt es regulatorische Einschränkungen? Wird dies ein mehrsprachiges Produkt? Gibt es Leistungskriterien, die erfüllt sein müssen? Im Speziellen muss definiert sein, wann eine Anforderung vom Entwicklungsteam als «erledigt» bezeichnet werden kann. Damit stellt man sicher, dass unbeliebte Arbeiten wie Dokumentation oder ungewohnte Tätigkeiten nicht auf später verschoben werden. Das beschreibende Regelwerk wird «Definition of Done» genannt. Die «Definition of Done» legt Anforderungen fest, die bei jeder Funktionalität gefordert sind.

Es ist wichtig, dass die «Definition of Done» jedem Entwickler bekannt ist und von Anfang an durchgesetzt wird. Die «Definition of Done» kann auch während des Projekts verfeinert werden, wenn sich zeigt, dass gewisse Aspekte zu wenig berücksichtigt worden sind. Im Folgenden ein Beispiel aus einem aktuellen Projekt, bei dem wir die verschiedenen «Done»-Kriterien auf Code-, Task- und User-Story-Ebene separat definierten:

Code-Commit/Push Definition of Done

- Code fulfills coding guidelines (automatically checked by tools).
- Local build successful.
- Unit and developer acceptance tests are successful.
- Code reviewed by team member.

Task Definition of Done

- TDD/ATDD developed.
- Design discussions have been made.
- Implementation pushed (git).
- UI is I18N² ready. (No hard coded strings; the localization service is used.)

User Story Definition of Done

- Story review passed (team demo on test environment).
- UI is I18N² ready. (No hard coded strings)
- Update deployment if needed.

4.8 INFRASTRUKTUR UND UMGEBUNG

Scrum-Teams arbeiten nachweislich am effektivsten, wenn sich alle Teammitglieder im gleichen Raum befinden. Schon kleine räumliche Distanzen wie die Verteilung der Arbeitsplätze auf unterschiedliche Ecken eines Grossraumbüros wirken sich negativ auf die Zusammenarbeit und Teamgeschwindigkeit aus. Effektive Kommunikation und eine gute Zusammenarbeit sind Grundwerte von Scrum und müssen bei der Teamzusammensetzung und der Bereitstellung der Räumlichkeiten entsprechend beachtet werden.

² Abkürzung für Internationalisierung

Organisieren Sie einen Projektraum für Ihr Scrum-Team, oft auch «War room» genannt. Gerade Tische für alle Teammitglieder sowie ein Arbeitsplatz für den Scrum Master und den Product Owner gehören in diesen Raum. Zur Unterstützung von Engineering-Praktiken wie Pair-Programming bieten sich höhenverstellbare Tische an, sodass zwei Entwickler auch stehend am Pult entwickeln können.

Unsere Erfahrung hat gezeigt, dass die Kommunikation im Team am besten funktioniert, wenn die Teammitglieder zusammen an einem 4er-, 6er- oder 8er-Block sitzen. Sobald ein Team auf mehrere Tischblöcke verteilt wird, nimmt die Kommunikationsaktivität bereits wieder ab, da gewisse Teammitglieder einander den Rücken zugewandt haben.

Es gilt zu beachten, dass effiziente Kommunikation und gute Zusammenarbeit auch Nachteile haben können. Scrum-Teams sind phasenweise laut. Scrum-Teams in Räumen zu platzieren, in denen andere Mitarbeiter in Ruhe konzentriert arbeiten müssen, ist suboptimal. Auch den Scrum-Teammitgliedern müssen Rückzugsmöglichkeiten zur Verfügung stehen, um bestimmte Aufgaben konzentriert erledigen zu können. Separate kleine Sitzungszimmer, sogenannte Fokusräume, bieten sich dazu an. Ein grösserer Meetingraum wird für Planungmeetings oder zur Durchführung der Team-Retrospektiven benötigt. Ein Flipchart und genügend Platz für Gruppendiskussionen sind weitere wichtige Arbeitsplatzmerkmale für Scrum-Teams.

4.9 SCRUM BOARD

Ein wesentlicher Bestandteil im Arbeitsumfeld von Scrum-Teams ist das Scrum Board. Das Scrum Board als zentrales Arbeitsinstrument für die tägliche Arbeit sollte für alle Teammitglieder vom Arbeitsplatz aus ersichtlich sein. Auch wenn mittlerweile viele elektroni-

sche Tools die Organisation der Arbeit nach Scrum erlauben, ist ein physisches Scrum Board gemäss unserer Erfahrung für lokale Teams das geeignetste Hilfsmittel zur Arbeit mit Scrum. Auf dem physischen Scrum Board werden User Stories und Entwicklungsaufgaben auf Zetteln festgehalten und je nach Zustand (to do, in progress, to verify, done) umgehängt. Der aktuelle Stand der Entwicklung ist dabei auf einen Blick ersichtlich. Das Scrum Board dient als kommunikationsunterstützendes Mittel. Während der täglichen Kurzbesprechungen kann direkt auf die jeweilige Anforderung oder Entwicklungsaufgabe Bezug genommen werden.

4.10 AGILE GADGETS

Folgendes wird beim Starten eines Projektes mit Scrum vom ersten Tag an helfen:

- Ein Stapel Karten (Grösse ca. 74 x 105 mm): weiss für Stories, gelb, grün, blau für Tasks und rot für ungeplante Tasks und Bugs
- Magnete (mind. 50): um Karten an die Wand zu heften (Stahlgestell, Trennwand oder Magnet-Board)
- Leere Wand (mind. 3 x 2 m): um Karten und Projektinformationen aufzuhängen – bevorzugt Stahlgestell, Trennwand oder Magnet-Board, an welchem Magnete funktionieren
- «bbv Planning-Poker»-Karten: zum Schätzen von Stories und Tasks (Karten: ?, 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, ∞)
- bbv Agile Scrum Poster³: damit man den Scrum-Prozess immer vor Augen hat
- bbv Agile Requirements Engineering Poster: visualisiert die wesentlichen Aspekte bei der Anforderungserhebung in Scrum
- bbv Agiles Testing Poster zur Visualisierung der Qualitätssicherungsaspekte in Scrum

³ Bestellmöglichkeit: <http://www.bbv.ch/de/posterbestellung.html>

4.11 SOFTWARE UND DOKUMENTE

Oft werden wir gefragt, welche Tools für Scrum eingesetzt werden sollen. Die Toolfrage ist bei der Einführung von Scrum aber sekundär. Am besten eignen sich einfache Tools wie ein physikalisches Scrum Board und eine manuelle Story Map an der Wand. Einfache Office-Artefakte wie ein elektronisch geführtes Backlog in einer Tabellenkalkulation erfüllen den Zweck für ein einzelnes Scrum-Team. Erst bei komplexen, verteilten Umgebungen über mehrere Standorte hinweg oder bei regulatorischen Zwängen drängt sich die Nutzung eines Cloud-basierten Tools auf.

Die folgenden Dokumente helfen beim Starten eines Projekts mit Scrum vom ersten Tag an, ohne Zeit mit der Installation und Konfiguration von zusätzlichen Software-Tools zu verlieren:

- Product Backlog in einer Tabellenkalkulation: priorisierte Liste mit Stories und Epics
- Sprint-Informationen: Sprintziele, Ausblick und Resultate der Retrospektive als Ausdruck an der Wand. Sprint-Burndown-Grafik von Hand auf eine Vorlage am Scrum Board eingezeichnet
- Release-Informationen: Produktvision, Release-Ziele, Release Burndown Chart als Ausdruck oder als manuelle Grafik am Scrum Board
- Team-Informationen in einer Tabellenkalkulation: kumulierte Verfügbarkeiten, Statistik der Teamgeschwindigkeit der vergangenen Sprints als Ausdruck an der Wand oder am Scrum Board
- Release-Plan: Meilensteine, Sprint-Ziele und Projektion
- Liste der Risiken in einer Tabellenkalkulation: Eintrittswahrscheinlichkeit, Ausmass und Kategorie, gemäss Best Practices aus dem klassischen Projektmanagement

4.12 TRAINING

Sie können nicht erwarten, dass alle Mitarbeiter Agilität in ihrer jeweiligen Disziplin einfach so können oder sich dies im Selbststudium aneignen. Um Widerstände zu überwinden und schnelle Erfolge erzielen zu können, müssen alle Projektbeteiligten proaktiv in ihrer Disziplin geschult werden. Neben einer allgemeinen Schulung in agilen Grundprinzipen, Verhaltensweisen und Scrum als Projektvorgehensmethodik sind gezielte Schulungen für die jeweiligen Spezialdisziplinen Software Engineering und Product Ownership unabdingbar.

Die Schulung kann in drei Hauptbereiche aufgeteilt werden:

- Training Agilität und Scrum (alle inkl. Management und Stakeholder)
- Training der Engineering-Praktiken (Entwickler)
- Training im agilen Requirements Engineering (Product Owner)

Idealerweise wird die Schulung mehrstufig durchgeführt, sodass die Mitarbeiter nach einer Einführung das Gelernte direkt anwenden können. Nachdem die ersten Erfahrungen gemacht worden sind, muss das Wissen mit gezielten Workshops vertieft werden. Nach einigen Monaten sollte das Wissen überprüft und nochmals vertieft werden, indem in den Folgekursen auch auf fortgeschrittene Themen eingegangen wird.

Die stetige Verbesserung als Kernidee hinter Scrum soll sich auch auf die Fähigkeiten der Mitarbeiter auswirken. Für jeden Sprint muss ein halber Tag (2 bis 4 Stunden in einem zweiwöchigen Sprint) für Weiterbildung reserviert bleiben. Dieser halbe Tag findet am besten nach dem Sprint Planning Meeting statt und kann Themen im Bereich Scrum, Engineering-Praktiken und System- oder Softwarearchitektur enthalten.

5 PLANUNG

Ein verbreitetes Vorurteil gegenüber Scrum ist die Ansicht, dass Scrum-Projekte unplanbar seien. Ganz im Gegenteil bietet Scrum machtvolle Planungswerkzeuge, die, richtig angewandt, klassischen Planungsinstrumenten überlegen sind.

5.1 AUFWANDSCHÄTZUNGEN

Voraussetzung für jegliche Planung ist die Verfügbarkeit von Aufwandschätzungen. In Scrum-Projekten werden Aufwandschätzungen grundsätzlich vom Entwicklungsteam vorgenommen, also jenen Experten, welche die Funktionalität erstellen werden. Mittels Planungspoker, einer Ausprägung einer Wideband-Delphi-Schätzmethode, werden Schätzungen in der «Story-Point»-Einheit abgegeben. Story Points sind Aussagen zur Komplexität einer Funktionalität und stellen eine relative Schätzgrösse dar. Die Aussagekraft bezieht sich auf den relativen Aufwand im Vergleich zu anderen User Stories. Rückschlüsse von Story Points auf Manntage oder -stunden sind zwar nach einer gewissen Zeit möglich, aber nicht das primäre Ziel des Schätzvorgehens.

Alle Einträge auf dem Product Backlog werden vom Scrum-Team geschätzt. Die relative Schätzgrösse in Story Points gibt dem Product Owner ausreichend Informationen, seine Priorisierung gegebenenfalls anzupassen. Zudem hat die relative und zeiteinheitunabhängige Schätzung des Aufwands den Vorteil, dass die Schätzungen nicht angepasst werden müssen, wenn das Entwicklungsteam über die Zeit produktiver und effektiver wird und für die gleiche Arbeit weniger Zeit benötigt.

Periodisch werden neue oder veränderte Backlog-Einträge vom Team neu geschätzt, sodass immer ein komplett geschätztes Backlog als Planungsgrundlage zur Verfügung steht.

5.2 TEAMGESCHWINDIGKEIT MESSEN

Parallel zu den Schätzungen wird die Geschwindigkeit des Entwicklungsteams gemessen. Unter Geschwindigkeit versteht man die Summe der geschätzten Story Points, die ein Entwicklungsteam in

einem Sprint erledigt. Anhand der gemessenen Geschwindigkeit und der Schätzwerte auf dem Backlog lässt sich projizieren, wie lange es mit dem gegebenen Team und der gegebenen Produktivität dauert, bis eine bestimmte Funktionalität gemäss der aktuellen Priorisierung umgesetzt ist.

5.3 BUDGET UND ZEITRAHMEN DEFINIEREN

Vor dem Abschluss des ersten Sprints sind die Schätzungen und die Teamgeschwindigkeit noch nicht verfügbar. Hier kann man wie beim klassischen Vorgehen Annahmen aufgrund von Erfahrungswerten treffen. Nach jedem Sprint werden die Annahmen durch die neuen Erfahrungswerte verfeinert und die noch vorhandene Unsicherheit in der Abschätzung reduziert.

Vor dem Projektstart werden Budget und Zeitrahmen für das Projekt fixiert, die Planungsprojektion basierend auf den Schätzwerten und der Teamgeschwindigkeit zeigt dem Product Owner dabei auf, wie viel Funktionalität mit gegebenem Budget und Zeitrahmen erzielt werden kann.

5.4 GESCHÄFTSWERTGETRIEBENES VORGEHEN

Bei der Planung zeigt sich der wesentliche Unterschied von agilen Projekten im Vergleich zum klassischen Projektansatz. Bei agilen Projekten werden die Kosten und der Zeitplan fixiert, die in diesem Rahmen umsetzbare Funktionalität wird geschätzt. Scrum propagiert das geschäftswertgetriebene Vorgehen. Alle Features werden nach Geschäftswert priorisiert (Return on Investment) und dann in dieser Reihenfolge umgesetzt. Dadurch wird sichergestellt, dass nach Ablauf von Zeit und Budget die wichtigsten Funktionalitäten umgesetzt sind. Dies bedeutet einen wesentlichen Kulturwandel bei der Transformation hin zur Agilität.

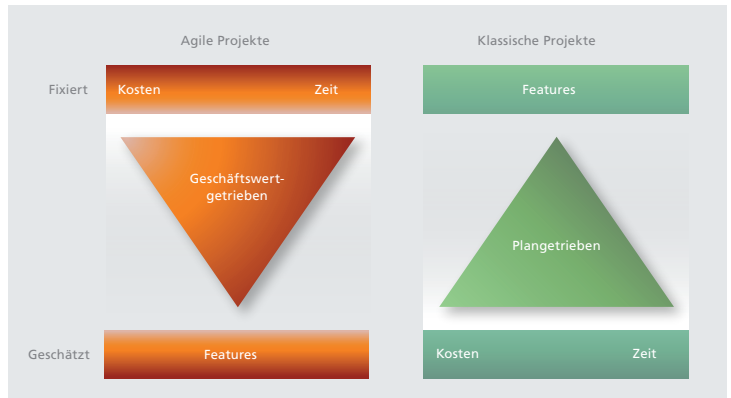


Abbildung 4:
Geschäftswert- statt
plangetrieben

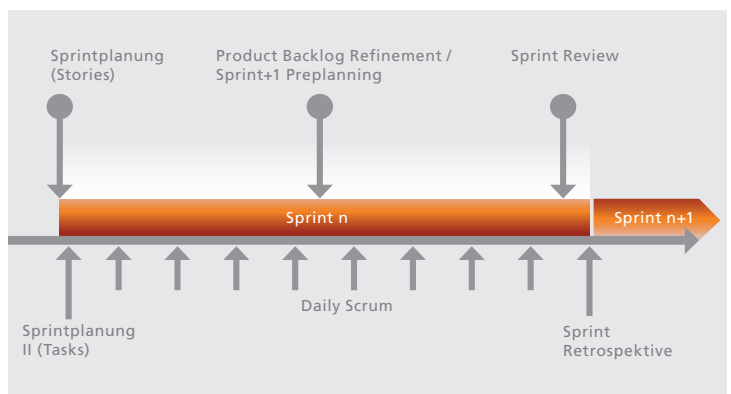
5.5 SPRINT-PLANUNG

Im ersten Teil des Planungs-Meetings (auch genannt Sprint-Planung I) findet die Festlegung der umzusetzenden Anforderungen für den nächsten Sprint statt. Das Meeting findet jeweils am ersten Tag des Sprints statt. Der Product Owner erklärt dem Team die User Stories in der Reihenfolge der Priorität. Das Scrum-Team versucht, die Stories zu verstehen, und fragt bei Unklarheiten nach. Es prüft die Akzeptanzkriterien, die definieren, was erfüllt sein muss, damit die Story den meist funktionalen Anforderungen aus Sicht des Product Owners genügt. Es prüft, ob die Stories die «Definition of Ready» erfüllen. Die «Definition of Ready» ist eine Sammlung an Kriterien, die Stories erfüllen sollen, bevor sie zur Umsetzung in einen Sprint gelangen können. Das Team schätzt diese Stories und anhand der Kapazität des Scrum-Teams teilt es dem Product Owner mit, welche Stories es in diesem Sprint voraussichtlich umsetzen kann. Bei dieser Aussage, auch «Forecast» genannt, handelt es sich um eine Abschätzung mit relativ hoher Genauigkeit, jedoch

nicht um eine vertragliche Zusage, die vom Team auf Biegen und Brechen unter Missachtung aller anderen Aspekte eingehalten werden muss.

In einem Folgemeeting (Sprint Planning II) ermittelt das Entwicklungsteam die nötigen Entwicklungsaufgaben zu jeder Story. Es betreibt in diesem Folgemeeting Softwaredesign und Softwarearchitektur und legt die notwendigen Arbeitsaufgaben so weit wie möglich fest. Der Product Owner ist nur im ersten Teil des Planungsmeetings aktiv involviert, in dem das «Was» abgesteckt wird. Die «Wie»-Diskussion im zweiten Teil verläuft unter der Führung des Entwicklungsteams.

Abbildung 5:
Sprint-Ablauf mit
Meetings während
des Sprints



5.6 PRODUCT BACKLOG REFINEMENT

Damit die Sprintplanung innerhalb der vorgesehenen Zeit effizient durchgeführt werden kann, wird rund eine Woche vor der Sprintplanung ein Product Backlog Refinement mit dem ganzen Team abgehalten. Im Product Backlog Refinement geht es um das gemeinsame Aufräumen und Querchecken der Anforderungen für das laufende Release. Primär geht der Product Owner die für den kommenden Sprint provisorisch vorgesehenen Stories mit dem Team durch. Diese prüfen, ob die Stories verständlich genug sind und vom Umfang her in den Sprint passen könnten. Das Feedback des Teams nutzt der Product Owner, um die Anforderungen in der verbleibenden Woche bis zum definitiven Planungsmeeting mittels Requirements Engineering zu verbessern. Das Product Backlog Refinement wird auch dazu genutzt, noch unklare Anforderungen und neue Features im Backlog grob zu schätzen und dem Product Owner mit den Schätzwerten die Grundlage für die Releaseplanung in die Hand zu geben. Das Ziel ist, das gesamte Backlog stets geschätzt zu haben.

5.7 RELEASE-PLANUNG

Für jedes Scrum-Projekt wird eine Release-Planung erstellt. Diese wird typischerweise alle zwei bis drei Sprints überprüft und an die Realitäten angepasst. Beim Projektstart basiert die Release-Planung für das erste Release auf Erfahrungswerten aus anderen, möglichst vergleichbaren Projekten. Nach den ersten Sprints wird diese an die Erfahrungswerte aus dem laufenden Release angepasst. Künftige Releases basieren auf den Werten von vorangegangenen Releases, um so stets die jeweils bestmögliche Planungsgenauigkeit zu erhalten.

Der Product Owner leitet aus den übergeordneten Zielen des Projekts die Ziele und Features für das kommende Release ab. Die Prioritäten

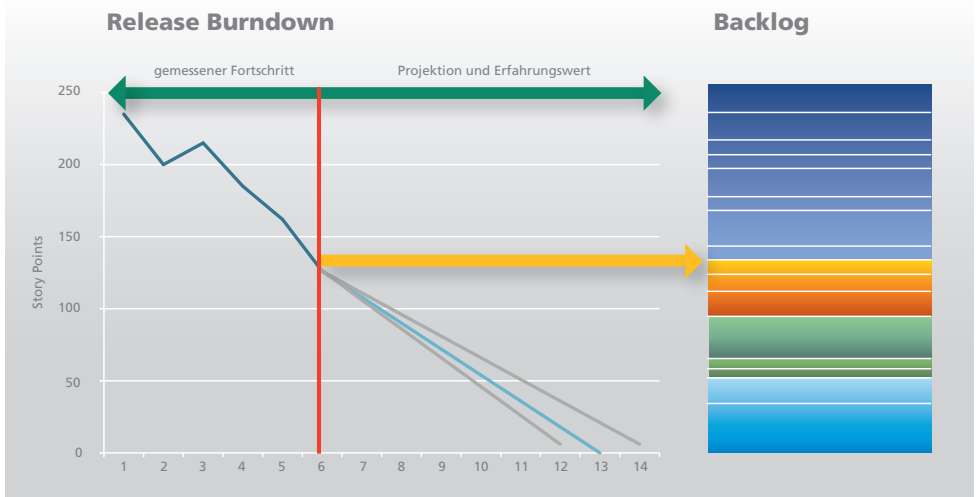


Abbildung 6: Die Release-Planung basiert auf Schätzwerten des Teams und auf der gemessenen Teamgeschwindigkeit

sind nach Kosten-Nutzen-Abwägung mit dem Kunden abgesprochen und klar definiert. Je nach Dauer eines Releases und der Sprint-Länge ergibt sich die Anzahl Sprints pro Release. Die Dauer des Releases hängt davon ab, wann der Kunde eine gewisse Funktionalität benötigt und wie viel Budget für ein Release zur Verfügung steht.

Der Product Owner wägt dabei insbesondere auch die Ausprägung eines Features so ab, dass sich mit der im Release umsetzbaren Funktionalität ein stimmiges Gesamtprodukt ergibt. Dieses Geschäftswertengineering ist eine spannende und herausfordernde Aufgabe und erfordert die Fähigkeit, Wichtiges von Unwichtigem zu trennen und aktiv Vereinfachungen zu suchen.

6 ERFOLGSFAKTOREN

Aus den Erfahrungen der letzten Jahre beim Aufgleisen von Scrum-Projekten in unterschiedlichen Sektoren und Firmen verschiedenster Grösse möchte ich einige der wichtigsten Erfolgsfaktoren hervorheben.

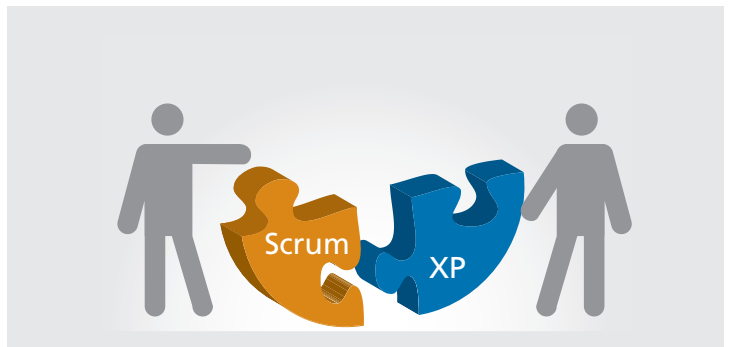
6.1 GET A COACH!

Machen Sie nicht die gleichen Fehler wie so viele andere Unternehmen, die sich zur Agilität transformiert haben. Zu oft verzichtet ein unerfahrener Scrum Master auf das Durchsetzen wichtiger Methodiken und Praktiken oder vernachlässigt wesentliche Aspekte im Change Management. Erfahrene Scrum Coaches haben bereits aus Fehlern gelernt und werden diese beim nächsten Einsatz tunlichst vermeiden. Ziehen Sie solche Leute bei und profitieren Sie davon, dass diese Scrum Coaches und Trainer auch jene Feinheiten beherrschen, die sich unter der einfachen Oberfläche von Scrum verbergen.

6.2 EINBEZUG VON TDD-GURUS & AGILEN ARCHITEKTEN

In Scrum wird die Qualität der Software nicht infrage gestellt. Zur Sicherstellung der Qualität sind fortschrittliche Engineering Practices eine Notwendigkeit. Die unter XP zusammengefassten Praktiken (TDD [7], Continuous Integration, Pair Programming, Collaborative Code Ownership), Clean-Code-Richtlinien [9] sowie der Fokus auf test- und wartbarer Softwarearchitektur bilden in Gesamtheit die Grundlage für erfolgreiche langfristige Scrum-Vorhaben.

Abbildung 7:
Scrum als Projektmanagement-Framework muss mit XP-Engineering-Praktiken kombiniert werden, um optimale Resultate zu erreichen



Für Aussenstehende und oftmals sogar für Product Owner ist die Bedeutung von professionellem Softwareengineering-Handwerk nicht ersichtlich. Qualität im Softwareengineering-Handwerk wird dabei nicht selten infrage gestellt. Während in der Bauwirtschaft niemand die sorgfältige, normengerechte Statikberechnung des Bauingenieurs hinterfragt, wird in der Softwareentwicklung die Arbeit gemäss XP-Grundsätzen nicht als Selbstverständlichkeit erachtet. Starre verzahnte Softwarearchitekturen und entsprechend geringe Änderungsflexibilität sind Beispiele für die im Lebenszyklus von Software sichtbaren Effekte einer mangelhaften Softwarequalität.

Legen Sie bei der Einführung von Scrum auch Wert auf XP, da die Engineering-Praktiken die handwerkliche Antwort auf die inkrementell-iterative Vorgehensweise darstellen. Scrum ohne XP funktioniert auf Dauer nicht. Die flexible Umsetzung sich stetig ändernder Anforderungen lässt sich nur erreichen, wenn dem Entwickler eine Phalanx an automatisierten Testfällen sowie eine flexible Softwarearchitektur zur Seite steht.

Agile Softwarearchitektur erfordert in der Entwicklungsdisziplin ein hohes Mass an Know-how im Bereich von Clean Code, test-getriebener Entwicklung und agiler Entwurfsmuster. Um Engineering Practices wie TDD [8] in einem Team einzuführen, brauchen Sie mindestens ein bis zwei Softwareingenieure mit Erfahrung in diesem Bereich. Ein unerfahrener Softwareentwickler benötigt rund ein halbes Jahr, bis er die Kniffe für TDD beherrscht. Es ist nicht ratsam, während dieser sechs Monate Software ohne fundierten TDD-Know-how-Code zu entwickeln. Die Kosten eines nachträglichen Umbaus der Software hin zu einer agilen Softwarearchitektur übersteigen in der Regel die Kosten für den Beizug eines TDD-Experten in der Anfangsphase des Projekts. Der Umgang mit Frameworks

und Entwurfsmustern ist dabei das eine, das Erkennen passender Anwendungsfälle im Code das andere. Das flexible Akzeptieren von Featureänderungen im Product Backlog erfordert zwingend die Etablierung von Clean Code, Testautomatisierung mit TDD und agiler Softwarearchitektur als flankierende Massnahmen. Andernfalls leidet mittelfristig die Entwicklungsgeschwindigkeit, und als Folge davon können die versprochenen Vorteile von Scrum nur unzureichend realisiert werden.

6.3 AGILES REQUIREMENTS ENGINEERING

Die Arbeit mit Scrum bedingt auch eine Adaption des Vorgehens bei der Anforderungserhebung und Anforderungsverwaltung. Das Agile Requirements Engineering [10] wird in Scrum vom Product Owner wahrgenommen. Dieser kann durch Fachspezialisten (User Interface, Fachteilbereiche) unterstützt werden und leitet das Product-Owner-Team mit derartigen Experten.

In der Startphase werden die zentralen Funktionalitäten des Produkts auf einer hohen Abstraktionsstufe erfasst. Diese sogenannten Epics werden untereinander priorisiert und für die ersten Epen erste Funktionalitäten in Form von User Stories extrahiert. All diese Elemente werden im Product Backlog verwaltet. Eine Tabellenkalkulation mit den Spalten «Id», «Beschreibung», «Status», «Akzeptanzkriterien», «Thema», «Priorität» und «Schätzung» ist ein ausreichendes Hilfsmittel für den Start eines Projekts.

Die Priorisierung, mehrstufige Gliederung und Detaillierung der Epen in Stories erfolgt dabei kontinuierlich. Sobald ausreichend Funktionalitäten spezifiziert sind, um ein Entwicklungsteam während zwei bis drei Sprints zu beschäftigen, startet bereits die Umsetzung. Parallel zur Umsetzung wird Requirements Engineering für

die folgenden Sprints betrieben und es werden weitere Stories aus den Epen abgeleitet und detaillierter erfasst.

6.4 INVESTIEREN SIE IN GUTE USER STORIES

Die User Stories [11] dienen dem Scrum-Team als seine einzige Anforderungsquelle. Die Qualität der User Stories ist dabei ein entscheidendes Kriterium für den Projekterfolg. Wichtig ist, dass das Scrum-Team in die Beurteilung der User Stories involviert wird. Das Scrum-Team schätzt die User Stories und beurteilt, ob diese detailliert genug sind, um sie in den Sprint aufnehmen zu können («Ready»). Dem Product Owner dient dabei die «Definition of Ready» als Hilfsmittel, um zu erkennen, welche Punkte erfüllt sein müssen, damit er die User Story dem Team vorstellen kann.

Eine User Story besteht aus drei wesentlichen Teilen: Beschreibung, Akzeptanzkriterien und Komplexitätsabschätzung. Dabei kann die Grösse der User Stories je nach Team und Komplexität der Funktionen variieren. Eine Faustregel für die Grösse einer Story: Wenn die Schätzung mehr als eine Entwicklerwoche beträgt, ist die User Story zu gross. In einen Entwicklungssprint sollten als Richtwert etwa zehn User Stories passen. Diese Menge gewährleistet, dass die Entwickler in Zweier- oder Dreiergruppen an verschiedenen Stories arbeiten können. Zudem können kleinere Stories verlässlicher geschätzt werden, was die Planungsgenauigkeit des Sprints verbessert.

Die User Stories müssen genaue Akzeptanzkriterien enthalten, damit klar ist, wann die User Story erfüllt, also «Done» ist. Akzeptanzkriterien können sowohl fachliche Anforderungen (Eingabevalidierung, Geschäftsregeln etc.) wie auch qualitative Anforderungen (Antwortzeit der Funktion) sein.

6.5 KOMBINATION MIT KANBAN

Zur Behandlung von Projektaufgaben (planbar) und Wartung (Bugfixing, Support) mit dem gleichen Entwicklungsteam bietet sich die Kombination von Scrum und Kanban [12] an. Bugs, die sofort gefixt werden müssen (Hotfixes), können nicht im Scrum-Prozess behandelt werden. Eine Zeitversetzung von mehreren Wochen ist nicht akzeptabel.

Parallel zu den geplanten Scrum-Aufgaben reserviert man in solchen Teams eine limitierte Kapazität für Ad-hoc-Aufgaben aus dem Betrieb. Diese werden über den Kanban-Teil eines hybriden Scrum-/Kanban-Boards ins Team eingespielen. Das Hybridboard führt eine «Priority»-Linie mit einer bis zwei möglichen Ad-hoc-Aufgaben, die, sollten sie auftreten, vom Entwicklungsteam priorisiert behandelt werden müssen. 10–20% der Entwicklungskapazität werden dabei für diese Art Aufgaben reserviert. Übersteigt die nötige Kapazität diesen Prozentwert, sind andere Massnahmen wie ein dediziertes Wartungsteam zu prüfen.

6.6 MESSBARKEIT DER FORTSCHRITTE

Scrum verspricht Prozessverbesserungen auf mehreren Ebenen. Etablieren Sie vorgängig Messgrössen, anhand derer Sie die Verbesserungen ausweisen können. Erheben Sie die Kennzahlen bereits vor dem Start mit Scrum. Denn Verbesserungen, die Sie vorgängig nicht mit Messgrössen hinterlegen, können Sie im Nachhinein nicht belegen. Mit klaren Fakten überzeugen Sie Kritiker und Zweifler vom positiven Effekt der Veränderung und Sie erkennen selbst noch vorhandenes Verbesserungspotenzial. Mögliche Messgrössen sind Durchlaufzeiten (Feature Requests, Bugfixes), Anzahl Releases pro Jahr, die Aufwände für System Testing oder die Anzahl Supportanrufe im Callcenter. Passende Kennzahlen erlauben Ihnen auch, verlässliche Rückschlüsse über den Erfolg oder Misserfolg von Produktionseinführungen zu ziehen.

6.7 ERFOLGE FEIERN

Feiern Sie Ihre Erfolge. Warten Sie nicht bis zum Projektabschluss. Erfolgserlebnisse motivieren und steigern damit auch die Leistungsfähigkeit des Teams. Scrum in einem kleineren Rahmen (ein Team) einzuführen, gibt Ihnen Gelegenheit, schnell erste Erfolge zu erzielen. Bei der Einführung von Scrum ist es wichtig, dass spätestens nach ein paar Sprints ein Erfolgserlebnis für das Team erreicht werden kann. Gelegenheiten zu feiern, gibt es viele. Mit Scrum liefern Sie alle zwei Wochen lauffähige Software, die Geschäftsnutzen erbringt. Nutzen Sie Scrum-Zeremonien wie den Sprint Review für einen anschließenden Apéro. Dieser kann zugleich dazu dienen, in ungezwungener Atmosphäre wichtige Informationen mit dem Kunden und den Stakeholdern auszutauschen.

Je nach Firmenkultur bergen die Scrum-Retrospektiven die Gefahr, dass in einem Projekt nur die negativen Aspekte erkannt werden. Geben Sie hier Gegensteuer und beleuchten Sie auch die positiven Aspekte Ihrer agilen Projektarbeit.

6.8 EINBINDUNG DES KUNDEN

Im Gegensatz zum Projektvorgehen ist der Vertreter des Kunden während des ganzen Entwicklungsprozesses stark in die Entstehung des Produkts involviert. Er ist dauerhaft bei der Klärung der Anforderungen engagiert, da diese nicht vorgängig vollständig definiert worden sind. Die enge Zusammenarbeit erfordert die Präsenz des Kunden oder dessen Vertretung bei den Sprint Reviews und Anforderungsklärungsmeetings.

Reviews sind wesentlicher Bestandteil der in Scrum integrierten Feedback-Kultur, über die der Kunde seine Bedürfnisse ins Projekt einfließen lassen kann und über den Stand des Projekts in-

formiert wird. Der Kunde hat so die Möglichkeit, Korrekturen sofort dem Team zu kommunizieren und so sicherzustellen, dass er am Ende wirklich das Produkt erhält, das seinen Erwartungen entspricht. Ein Sprint Review beinhaltet allgemeine Informationen zum aktuellen Stand der Arbeiten sowie eine Demonstration der in diesem Sprint umgesetzten Funktionalitäten.

Schaffen Sie ein Bewusstsein für dieses wichtige Kundenengagement im Scrum-Projekt. Andernfalls besteht die Gefahr, dass Sie nicht alle Vorteile von Scrum realisieren können und Sie auf wertvolles Kundenfeedback verzichten müssen.

6.9 AGILES MANAGEMENT

Scrum bedeutet für viele Firmen auch einen Kulturwandel. Scrum propagiert die Arbeit mit selbstorganisierten, eigenverantwortlichen Teams. Dies bedingt eine bewusste Zurückhaltung des Managements beim Eingriff in die Umsetzungsentscheide der Projektteams. Selbstorganisation funktioniert nur, wenn dies aktiv eingefordert wird und nicht bei der erstbesten Gelegenheit durch fremdgesteuerte Entscheide unterwandert wird. Selbstorganisation darf dabei aber auch nicht missverstanden werden. Es ist durchaus legitim, ein Projektteam auf Unzulänglichkeiten oder Unterlassungen hinzuweisen oder gegebenenfalls durch Rahmenbedingungen die Leitplanken klarer zu definieren. Der Umgang mit Selbstorganisation erfordert von den Führungskräften viel Disziplin und Fingerspitzengefühl sowie ein gutes Verständnis agiler Werte und Methoden.

Die Rolle des Managements ist in einer agilen Umgebung auf die Bereitstellung einer agilen Unternehmenskultur fokussiert, in der Vertrauen und Respekt ermöglicht und gefördert werden. Als wesentlicher Bestandteil gehört ein offenes Arbeitsklima mit transpa-

renter Kommunikation dazu. Die Beseitigung unternehmensinterner Hindernisse wird dabei zu einer Hauptaufgabe von Führungskräften in agilen Organisationen.

Zur Verbesserung der Teamleistung sind teaminterne Massnahmen meist unzureichend. Eine Firma muss entsprechende Bereitschaft zeigen, organisationsweite Veränderungen aktiv zu adressieren. Entsprechend wichtig ist es, dass Scrum oder andere agile Methoden vom Management der Firma mitgetragen und propagiert werden. Das Management wird eher früher als später über schwierige unternehmensweite Veränderungen entscheiden und seinen Beitrag an die Agilität der Firma leisten müssen. Ein Ignorieren der Verbesserungsbedürfnisse aus Scrum-Teams führt zu Frustration bei den Teammitgliedern und dazu, dass der Verbesserungsprozess als integraler Bestandteil von Scrum abgewürgt wird. Selbst wenn Scrum in einem Bottom-up-Ansatz eingeführt wird, ist relativ schnell die Unterstützung aus der Führung des Unternehmens erforderlich, um die Agilität nachhaltig im Unternehmen verankern und verbreitern zu können.

7 FAZIT

Scrum in seiner Essenz ist so einfach, dass es sich auf einem Bierdeckel erklären lässt. Die effektive Umsetzung mit allen Ecken und Kanten zu beherrschen, ist hingegen eine hochkomplexe Aufgabe, die sich nur mit viel Erfahrung und dem nötigen Fingerspitzengefühl bewerkstelligen lässt. Die Transformation hin zur Agilität ist für jedes Unternehmen ein komplexer Veränderungsprozess, der sehr schnell die ganze Firma erfassen kann. Nur eine konsequente und durchgängige Umsetzung führt zu nachhaltigem Erfolg und dazu, die von der Agilität versprochenen Vorteile realisieren zu können.

Scrum ist mehr als nur ein Umsetzungsprozess. Die konsequente Umsetzung erfordert die Adaption des Projektmanagements, der Anforderungserhebung, der Qualitätssicherung und der Engineering-Praktiken Ihrer Entwicklungsabteilung. Richtig eingeführt und im Projekt gelebt führt Scrum dazu, dass alle Projektbeteiligten mit Spass und hochmotiviert am Projekt arbeiten. Daraus resultierende hyperproduktive Teams vervielfachen die Fortschritte Ihrer Projekte und zahlen die Investitionen in Prozessverbesserungen mehrfach zurück. Erhöhte Reaktionsfähigkeit, Transparenz, Qualität und schnellere Markteinführungen sind die positiven Resultate einer nachhaltig umgesetzten Agilitätsstrategie.

Einige der genannten Aspekte in diesem Booklet tragen sicher dazu bei, dass Sie Fehler bei der Einführung von Scrum vermeiden können und schneller die gewünschten Erfolge erzielen. Gerne helfen wir bei der konkreten Erarbeitung und Anwendung der entsprechenden Massnahmen. bbv Software Services AG ist ausgewiesener Experte in der Produktentwicklung mit agilen Prozessen. Als Agilitätspionier in der Schweiz kann bbv Software Services AG auf einen grossen Fundus an Erfahrungen aus Scrum-Transformationen zurückgreifen. Die Scrum Coaches, Experten und Softwareingenieure von bbv Software Services unterstützen Sie gerne bei Ihrem Wandel zum agilen Unternehmen. Über das Schulungsprogramm «bbv Academy» geben wir unser Wissen weiter. Unsere Kurse unterstützen Ihre Mitarbeiter bei der Aneignung der nötigen Fähigkeiten.

8 ANHANG



8.1 FACHLEITER AGILE

Urs Enzler

Urs Enzler hat an der ETH Zürich Informatik studiert. Neben seiner Haupttätigkeit als Softwarearchitekt bei bbv Software Services AG unterstützt er Unternehmen bei der Einführung agiler Entwicklungsmethoden wie Scrum oder Test-driven Development. Er referiert auf Konferenzen und Tagungen in der Schweiz und in Deutschland über agile Softwareentwicklung und -architektur.

Er bloggt auf:

www.planetgeek.ch

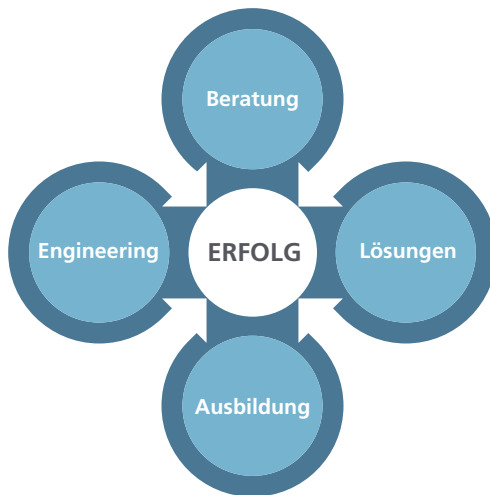
www.bbv.ch/blog

8.2 LITERATUR

- [1] Agile Trends und Benchmarks Schweiz 2012
www.swissq.it/images/uploads/SwissQ_Agile_Trends_2012.pdf
- [2] Scrum Guide, Ken Schwaber, Jeff Sutherland
www.scrum.org/scrumguides/, Oktober 2011
- [3] A guide to priority poker
www.uxforthemasses.com/priority-poker/
- [4] Buy a feature design game
www.uxforthemasses.com/buy-the-feature/
- [5] Leading Change, John P. Kotter, Harvard Business School Press
- [6] Hans-Jürg Kleine, Identitas, Erfahrungsbericht Scrum-Einführung
www.bbv.ch/images/bbv/pdf/downloads/system_event/2011_erfahrungsberichtscrum-einfuehrungbbv.pdf
- [7] Bruce Tuckman, Development Sequence in small groups
aneesha.ceit.uq.edu.au/drupal/sites/default/files/Tuckman%201965.pdf
- [8] Test-Driven Development by Example, Kent Beck,
 Addison-Wesley 2003
- [9] Clean Code Cheat Sheets, UrsENZler
www.planetgeek.ch/2011/01/04/clean-code-and-clean-tdd-cheat-sheets/
http://www.bbv.ch/images/bbv/pdf/downloads/Clean_Architecture.pdf
- [10]Agiles Requirements Engineering, Raphael Auf der Maur,
www.bbv.ch/images/bbv/pdf/downloads/booklets/Agiles_Requirements_Engineering.pdf
- [11]User Stories Applied: For Agile Software Development,
 Mike Cohn, Addison-Wesley 2004
- [12] Kanban, Successful Evolutionary Change for your Technology
 Business, David J. Anderson, Blue Hole Press, ISBN 978-0984521401



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

MAKING VISIONS WORK.

www.bbv.ch · info@bbv.ch