

Wie Sie auf agile Software-Entwicklung umstellen – Übersicht

Ausgehend vom Beitrag 08A01 „Agile Konzepte“ wird in drei aufeinander aufbauenden Beiträgen die Umstellung auf eine agile Form der SW-Entwicklung beschrieben.

Der hier vorliegende Beitrag zeigt, dass diese Umstellung Veränderungen der Unternehmenskultur voraussetzt. Das sind langfristige und strategisch relevante Investitionen. Erst auf dieser Basis können die jeweils angemessenen und vergleichsweise kurzlebigen Methoden und Techniken der agilen (aber auch „traditionellen“) SW-Entwicklung sinnvoll genutzt werden.

Dieser Beitrag gibt einen zusammenfassenden Überblick über einige Methoden

und Techniken zur Kulturveränderung und schlägt danach einen agilen Umgang mit den Methoden und Techniken der agilen SW-Entwicklung vor. Das „craftsmanship“ und das „Clean Code Development“ haben dabei unabhängig von der Vorgehensweise eine zentrale Bedeutung.

In zwei weiteren Beiträgen (Kap. 08A03 und Kap. 08A04) werden einzelne Methoden und Techniken zur Kulturveränderung im Detail behandelt und kontextabhängige Vorgehensmuster zur Umstellung auf agile Entwicklungsmethoden ergänzt.

Autor: Hans-Peter Korn
E-Mail: hans-peter@korn.ch

1 Das agile Vorgehen als strategische Investition

Sie möchten Ihre Software-Entwicklung (SW-Entwicklung) auf ein „agiles“ Vorgehen umstellen? Und zwar – so nehme ich an – als eine langfristig wirksame strategische Investition? Dann lade ich Sie zu einer langfristigeren Betrachtung ein. Werfen wir doch zunächst einen Blick auf die historischen Meilensteine der agilen SW-Entwicklung:

Der Begriff *Agile Software-Entwicklung* entstand erst 2001. Allerdings wies Winston W. Royce bereits 1970 in seinem Artikel „Management of Large Software Systems“ darauf hin, dass ein „wasserfallartiges“ Vorgehen ungeeignet sei, und

empfahl stattdessen ein iteratives Vorgehen mit sehr vielen Rückkopplungsschleifen unter Einbeziehung auch des Benutzers [1].

„Agile“ Methoden wie RAD bereits ab 1970

Ab 1970 gab es auch schon Vorgehensmethoden und Praktiken wie Rapid Application Development (RAD) von Dan Gielan (in den 1980er-Jahren bis 1991 formalisiert von James Martin) und Evolutionary Systems Delivery (EVO) von Tom Gilb, heute von ihm „Evolutionary Project Management“ genannt.

1980er – 1990er: EVO, Crystal, Spiralmodell Scrum seit 1995

Das Spiralmodell von Barry W. Boehm folgte 1986 und die Crystal-Methodenfamilie von Alistair Cockburn ab 1990.

Jeff Sutherland, John Scumniotales und Jeff McKenna entwickelten 1993 ein schlankes teambasiertes inkrementell-adaptives Vorgehen bei Easel Corporation, das sie „Scrum“ nannten, inspiriert von dem Artikel „The New New Product Development Game“ der Autoren Hirotaka Takeuchi und Ikujiro Nonaka. Bei Ken Schwaber’s Firma Advanced Development Methods wurde in den frühen 1990er-Jahren ein ähnliches Vorgehen wie Sutherlands Scrum praktiziert und 1995 stellte Ken Schwaber „Scrum“ in einem Workshop der OOPSLA ’95 mit Jeff Sutherland als einem der Organisatoren vor.

XP seit 1996

1996 verwendeten Kent Beck, Don Wells, Ward Cunningham und Ron Jeffries bei Chrysler erstmals eXtreme Programming (XP). Nach seiner Publikation in 1999 fand es rasante Verbreitung.

Ab Mitte 1990er: PRINCE2, DSDM, FDD, ASD

Danach folgten Schlag auf Schlag: 1996 PRINCE2 (Nachfolger von PRINCE) als allgemeine inkrementell-adaptive Projektmanagement-Methode; 1997 DSDM; ebenfalls 1997 das Feature Driven Development (FDD) von Jeff De Luca und

	1999 das Adaptive Software Development (ASD), entwickelt von Jim Highsmith und Sam Bayer, ausgehend von RAD.
2001: Agiles Manifest	Am 12. Februar 2001 trafen sich 17 Vertreter solcher – damals „leichtgewichtig“ genannter – Methoden und Vorgehensweisen in einem Skiresort in den Rocky Mountains mit dem Ziel, das sie Verbindende kompakt zu formulieren und dafür einen auch für Topmanager attraktiven Titel zu finden. Das Ergebnis war das heute bestens bekannte „Agile Manifest“ [2]. Immer wieder kommt seitdem die Forderung auf, seine vier „Werte“ und zwölf „Prinzipien“ im Lichte der inzwischen gemachten Erfahrungen anzupassen. Andererseits wird das Manifest in seiner Urform oft als „der“ bleibende und einzige Maßstab dafür angesehen, was „Agilität“ auch heute noch bedeutet. In einem Vortrag am 16. September 2013 in Vorarlberg (bei OMICRON, organisiert von PIONIERBASIS) betonte Alistair Cockburn, einer der Verfasser des Manifests, dass dieses Agile Manifest von 2001 ein „Schnappschuss“ aus der damaligen Zeit heraus sei. Deshalb mache es keinen Sinn, diesen historischen Schnappschuss zu aktualisieren. Wenn sich diese 17 Unterzeichner des Manifests – oder andere Vertreter der aktuellen SW-Entwicklungsszene – heute träfen, um wiederum das sie Verbindende zu formulieren, dann wäre das Ergebnis ein anders.
Heute: Scrum mit XP-Techniken ist dominant	Bis ungefähr 2005 dominierte Extreme Programming (XP) die agile SW-Entwicklung. Daneben waren auch Scrum, die Crystal Family of Methodologies, Feature Driven Development (FDD), die Dynamic Systems Development Method (DSDM) und das Adaptive Software Development (nebst anderen) verbreitet. Seit 2005 entwickelte sich indessen Scrum (3.3.1 in [3]) rasch zur heute bekanntesten agilen Vorgehensweise der SW-Entwicklung auf Teamebene, jedoch mit Einschluss vieler Techniken aus XP.

Vier Grundsätze	Kennzeichen dieses Vorgangs <ul style="list-style-type: none"> • Ein einzelnes kleines verfügbares („interdisziplinäres“) Produkt. • In fix getakteten „Sprints“ (zwei bis vier Wochen) werden Produkte realisiert und geliefert. • Es gibt nur eine einzige Verantwortliche Person gegenüber dem Team und verbindlich entschlossen. • Es gibt bereits eine Endergebnisbeschreibung (den „Product Backlog“). <p>Mehr und mehr setzt sich durch, dass dieses Vorgehen ein idealisierendes und starres ist, das viele Problemzonen der Projektmanagements unbeherrschbar macht.</p>
Scrum oft zu vereinfachend und radikal	Zudem stellt sich immer wieder die Einführung von Scrum als eine Vereinfachung und Radikalisierung von Organisationsänderungen dar. In der Praxis werden von vielen Betrieben und in ihrer Radikalität als Vereinfachung und Radikalisierung angesehen. Zunehmend kristallisiert sich ein gewachsenes System heraus, das einen hohen Grad an Automatisierung, eine rasche Abfolge von Iterationen und eine zunehmende Vorgehensweise der SW-Entwicklung auf Teamebene – adaptiv und flexibel – ermöglicht.

2010: Kanban für SW Die Problematik starrer Sprints und der oft überfordernden Organisationsänderungen wird durch das erst seit 2010 breiter bekannte, von David J. Anderson für die SW-Entwicklung angepasste „Kanban“ entschärft (3.3.2 in [3]). Es verbreitet sich derzeit rasch und wird auch zusammen mit Scrum zu „Scrumban“ kombiniert.

SAFe gewinnt an Boden Zur team- und projektübergreifenden agilen Abstimmung der projektunabhängigen fortlaufenden Produktentwicklung auf Programm- und Portfolioebene tritt derzeit das „Scaled Agile Framework for Enterprises“ (SAFe) von Dean Leffingwell stark in Erscheinung (3.3.3 in [3]). Er hat es erstmals 2010 als Buch und als „big picture“ im Internet veröffentlicht. Und für das agile Management von Projekten gewinnen DSDM und PRINCE2 (3.2 in [3]) als Rahmen für die Arbeit einzelner agiler Teams an Bedeutung.

Methodenentscheidungen sind keine Langfriststrategie! Angesichts dessen, dass Scrum erst in den letzten Jahren so dominant wurde und heute in der agilen SW-Entwicklung zunehmend bekannte Vorgehensweisen wie Kanban und SAFe erst 2010 ins breitere Bewusstsein rückten, kann heute nicht vorausgesagt werden, was in fünf oder zehn Jahren die Szene der SW-Entwicklung prägen wird und inwieweit dann „agil“ (oder „lean“) immer noch dominante Begriffe sein werden.

Längerfristig ausgelegte strategische Investitionsentscheidungen im Bereich eines spezifischen SW-Entwicklungsvorgehens sind also mit einer hohen Unsicherheit verbunden.

Systemheterogenität und Architekturmängel als Hindernisse

Eines aber ist mit Sicherheit auch langfristig gültig: Heterogene Systemlandschaften mit vielen, eher undurchsichtigen und oft improvisierten Schnittstellen, mit unüberblickbaren oder gar nicht dokumentierten Systemarchitekturen und Praktiken, die dem „Clean Code Development“ widerspre-

chen, können mit keinem wie auch immer gearteten Vorgehen der SW-Entwicklung entschärft werden. Das bedeutet im Umkehrschluss: Wichtiger als das Optimieren des Vorgehens der SW-Entwicklung ist,

- historisch wild gewachsene Systemlandschaften und Architekturen transparent zu machen und dann zu bereinigen sowie
- die Praktiken der SW-Entwicklung, vom Requirements Engineering über die Testautomation und kontinuierliche Integration bis hin zu einem zuverlässigen Configuration Management im Sinne des „Clean Code Development“ zu professionalisieren. „Craftsmanship“, das handwerkliche Können, ist zentral!

Gefordert weiterhin: „Craftsmanship“

Angenommen, all das ist erreicht. Dann geht es jetzt um das Optimieren des Vorgehens in der SW-Entwicklung. Wie und wo beginnen Sie? Und was genau bedeutet „Optimieren des Vorgehens“? Wohin soll die Reise führen?

2 Wohin wollen Sie mit Ihrem Unternehmen?

Um das zu beantworten, sollten Sie – bevor Sie sich mit spezifischen Philosophien (wie „Lean“ oder „Agil“), Vorgehens-Frameworks oder Methodiken (wie Scrum, SAFe, PRINCE2, PMBOK, CMMI, RUP, V-Modell oder HERMES) beschäftigen – zusammen mit ihren „Peers“ im Management Folgendes klären:

Setzen Sie sich in die Zeitmaschine

Angenommen, Sie setzen sich mit Ihren Peers in eine Zeitmaschine . . . und „beamen“ sich fünf Jahre in die Zukunft . . . in eine Zukunft, in der Ihr Unternehmen ein optimales Vorgehen der SW-Entwicklung praktiziert . . . und Sie alle steigen aus der Zeitmaschine aus:

- *Woran werden Sie, als Manager, bereits in den ersten Stunden und Tagen anhand STRATEGISCH relevanter KONKRETER Beobachtungen dieses optimale Vorgehen erkennen?*
- *Was werden Ihnen die Kunden erzählen, was sich in den letzten fünf Jahren für sie markant verbessert hat – und was sie sogar begeistert?*
- *Was werden Ihnen die Mitarbeitenden erzählen, was sich in den letzten fünf Jahren für sie markant verbessert hat – und was sie stark motiviert?*

Notieren Sie all das, was Sie und Ihre Peers beobachten und von Ihren Kunden und Mitarbeitenden zu hören bekommen. Nehmen Sie diese Notizen mit in die Zeitmaschine . . . und beamen Sie sich wieder zurück ins Jetzt. Leiten Sie dann aus Ihren Notizen all das ab, was Ihre „besten Hoffnungen“ für ein in fünf Jahren optimales Vorgehen der SW-Entwicklung sind.

(Eine ausführliche Beschreibung der Technik solcher „Reisen in die Zukunft“ finden Sie in [4])

Formulieren Sie das möglichst spezifisch und mit konkret beobachtbaren Kriterien. Vermeiden Sie dabei Fachbegriffe aus der „agilen Szene“ (die in fünf Jahren vielleicht nicht mehr „in Mode“ ist), sondern umschreiben Sie es in allgemein verständlichen „5-Cent-Worten“.

Etwa so:

„Die Kunden unserer CRM4KMU-Lösung erhalten nur noch zwei Mal pro Jahr einen umfassend getesteten Release ohne kurz darauf folgende Updates.“

„Pro Release erhalten wir nur noch zwei Mal pro Jahr einen umfassend getesteten Release ohne kurz darauf folgende Updates.“

KONKRETE Ziele als beste Hoffnungen

Diese nicht als vage Visionen, sondern als konkrete Ziele formulieren. Ihre Messlatte für möglicherweise anzupassende Vorgehenskonzepte für die nächsten SW-Entwicklungszyklen.

3 Wie reisen Sie mit

Ihre Reise ins Land der Ideenfindung und der Identifizierung möglicher genereller Methodiken hängen ganz von den Umständen Ihres Unternehmens ab. Die komplex erscheinenden sozialen und technischen spezifischen Problemstellungen als Lösung sind hier ungenutzte Potenziale. Sie können aber nicht 1:1 kopiert werden.

Kein Kochbuch, keine „Best Practices“

Als solche Denkanstöße und direkt nutzbare Blaupausen sind unterschiedliche Vorgehensmodelle, die drei von den jeweiligen Problemstellungen angeht, zu behandeln:

1. Entwicklung und Umlösungen:

Plattformbasierte Lösungen

Zur Verfügung stellen (Ergänzung/Anpassung/Weiterentwicklung)

rianten von SW-Lösungen und Services auf der Basis einer kundenneutralen Plattform oder eines kundenneutralen Lösungsbaukastens. Typische Beispiele: Anbieter von ERP-Lösungen; Anbieter umfassender Web-Shops samt Backoffice-Logistik und B2B-Lösungen.

Vorgehens-Frameworks und Methodiken für diese Konstellation müssen insbesondere auch das strategische Portfoliomanagement für die verschiedenen langfristig existierenden Produktlinien umfassen. Von zentraler Bedeutung sind bei dieser Aufgabenkonstellation einerseits die fortlaufende Pflege der kundenneutralen Lösungsplattform und andererseits die rasche Realisierung auch sehr spezieller – später auch zu betreuender – Kundenlösungen, ohne dabei die generelle Lösungsplattform zu verkomplizieren oder mit separaten Speziallösungen zu unterlaufen. Die konsequente Einhaltung und fortlaufende Optimierung einer modularen, serviceorientierten Gesamtarchitektur ist hier unabdingbar.

Es müssen also drei Arten von Vorgehensweisen in Einklang gebracht werden:

Erstens das kontinuierliche, als „Arbeitsfluss mit unbestimmtem Ende“ gestaltete Pflegen und Weiterentwickeln der kundenneutralen Plattform. Zweitens das zeitlich begrenzte projektartige Entwickeln kundenspezifischer Lösungen unter weitestgehender Nutzung dieser Plattform. Und drittens das wiederum als „Arbeitsfluss mit unbestimmtem Ende“ gestaltete Pflegen und Weiterentwickeln der kundenspezifischen Lösungen.

Unternehmen dieser Aufgabenkonstellation umfassen mehrere Hundert oft bis viele Tausend (weltweit) kooperierende Informatikspezialisten in Hunderten verschiedenen Teams. Nicht das Vorgehen und die Methodik in einem kleinen Team,

das ein Produkt innerhalb eines Projekts realisiert, ist hier die Herausforderung, sondern die Koordination all dieser Spezialisten und Teams auf Programm- und Portfolioebene.

Für eine agile Form des Portfoliomanagements sind SAFe oder andere Varianten von Portfolio-Kanban denkbar. Das agile Management von Projekten ist mit PRINCE2 oder DSDM möglich. Die agile Kooperation vieler Teams innerhalb einer Produktlinie kann als „Release Train“, z. B. auf der Basis von SAFe auf der Programmebene erfolgen. Und für das agile Arbeiten im Team eignen sich Kanban (insbesondere bei fortlaufenden Wartungsarbeiten), Scrum, Scrumban oder DSDM. In jedem Fall müssen auf Teamebene Praktiken des Clean Code Development genutzt werden.

2. Entwicklung und Unterhalt umfassender Individuallösungen:

Umfassende Individuallösungen

Zur Verfügung stellen (Entwicklung/fortlaufende Betreuung/Anpassung/Weiterentwicklung) umfassender individueller SW-Lösungen und SW-Services für spezifische Kunden oder für den Eigenbedarf des Unternehmens. Typische Beispiele: Anbieter und Betreiber firmenspezifischer Spezialsoftware und -services; unternehmensinterner IT-Bereich.

Vorgehens-Frameworks und Methodiken auch dieser Konstellation müssen das strategische Portfoliomanagement für die verschiedenen langfristig zu unterstützenden individuellen SW-Lösungen und SW-Services umfassen.

In diesen Unternehmen müssen zwei Arten von Vorgehensweisen in Einklang gebracht werden:

Erstens das zeitlich begrenzte projektartige Neuentwickeln oder grundlegende Überarbeiten kundenspezifischer Indivi-

duallösungen. Und zweitens das als „Arbeitsfluss mit unbestimmtem Ende“ gestaltete Pflegen und Weiterentwickeln der kundenspezifischen Lösungen.

Bei Unternehmen dieser Aufgabenkonstellation, insbesondere bei unternehmensinternen IT-Bereichen, besteht die Herausforderung sehr oft im Umgang mit der ohne „Überbauplan“ historisch gewachsenen und sehr heterogenen System- und Anwendungslandschaft mit vielen, oft schlecht dokumentierten und undurchsichtigen Altsystemen.

Auch diese Art von Unternehmen kann mehrere tausend weltweit kooperierende Informatik-Spezialisten in vielen verschiedenen Teams umfassen. Auch hier ist die Herausforderung nicht primär das Vorgehen und die Methodik in einem kleinen Team, das ein Produkt innerhalb eines Projekts realisiert, sondern die Koordination all dieser Spezialisten und Teams auf Programm- und Portfolioebene.

Für eine agile Form des Portfoliomanagements sind – wie bereits vorher erwähnt – SAFe oder andere Varianten von Portfolio-Kanban denkbar. Das agile Management von Projekten ist mit PRINCE2 oder DSDM möglich. Die agile Kooperation vieler Teams innerhalb einer Produktlinie kann, wie schon erwähnt, als „Release Train“, z. B. auf der Basis von SAFe auf der Programmebene, erfolgen. Und für das agile Arbeiten im Team eignen sich Kanban (insbesondere bei fortlaufenden Wartungsarbeiten), Scrum, Scrumban oder DSDM. In jedem Fall müssen auf Teamebene auch hier Praktiken des Clean Code Developments genutzt werden.

Für das Bereinigen der historisch gewachsenen und sehr heterogenen System- und Anwendungslandschaften kann im SAFe all das nützlich sein, was sich auf „Architektur“ (z. B.

„Architectural Epics“ auf Runway“ auf Programme

3. SW-Entwicklung als I

SW-Entwicklung als Dienstleistung Anbieten von SW-Entwicklungsinfrastruktur Kundenprojekte bis zur I seltenen – separat beauftragte Wartungs-, Anpassungs- und Beispiele: Als Agenturen wareschmieden“ für Kunden im Bereich Web-Lösungen

Die Vorgehens-Frameworker der SW-Entwicklung, wenn mit dem Auftrag der SW, die Kosten und solche Aufträge typischer mit einem klar definierten Handvoll kooperierender Vorgehensweisen die die Arbeit im Team und dernde teamübergreifend werden. Oft aber arbeiten Agenturen innerhalb von größeren Projekten mit auf Vorgehen und die Methoden

Erfolgsentscheidend für die verschiedensten Vorgehensweisen professionell bewegen können über dem Auftraggeber die Bedienung“ bedienen und dahingehend sondern effektiv im Sinne

arbeiten. In jedem Fall – und kaum im Widerspruch zu vorgegebenen Frameworks – müssen auf Teamebene Praktiken des Clean Code Development als „die“ Kernkompetenz des Dienstleisters genutzt werden.

Herausfordernd für diese Anbieter ist, dass sie einerseits in der Regel an mehreren solcher Projekte gleichzeitig arbeiten und andererseits bestimmte Spezialisten wie z. B. GUI-Designer mit nur einem Projekt allein nicht ausgelastet sind. Das erfordert eine projektübergreifende mittelfristige Einsatzplanung auf Personenebene, was allerdings die kurzfristige Flexibilität erschwert und der Idealvorstellung eines personell stabilen interdisziplinären Teams mit Personen, die nur in einem Team arbeiten, widerspricht.

Wenn Vorgehens-Frameworks und Methodiken vom Anbieter der SW-Entwicklungskapazität frei wählbar sind, dann sind auf der Ebene des agilen Projektmanagements – auch bei sehr kleinen Projekten – mit „Tailoring“ angepasste Formen von PRINCE2 oder DSDM möglich und das agile Arbeiten auf Teamebene kann z. B. mit Kanban, Scrum, Scrumban oder DSDM erfolgen, kombiniert mit der Nutzung der Praktiken des Clean Code Development. Für die projektübergreifende mittelfristige Einsatzplanung auf Personenebene sind alle bekannten Methoden denkbar. Sie sollten im Interesse der Flexibilität und Transparenz jedoch möglichst einfach gestaltet sein und von den Betroffenen selbst (und nicht von einer zentralen Planungsinstanz) gepflegt werden können.

Je nach Konstellation eine andere Agilität

Es gibt also nicht „das eine“ Vorgehens-Framework und „die eine“ Methodik für alle diese unterschiedlichen Konstellationen. Und daher gibt es auch nicht „DIE“ Agilität und „DEN EINEN“ Weg in Richtung Agilität, sondern ganz verschie-

dene Ausprägungen von Agilität und der Wege dorthin. Selbst innerhalb eines Unternehmens können sich diese Ausprägungen und die Wege dorthin je nach aktueller Situation im und rund um das Unternehmen immer wieder ändern.

„Agilität“ für ein Unternehmen bedeutet also nicht, sich einer der als „agil“ bezeichneten Vorgehensweisen zu verpflichten, sondern je nach Situation das angemessene Vorgehen zu praktizieren, solange es passt, und rasch zu einem anderen Vorgehen zu wechseln, wenn es sich als passender anbietet.

Wichtig!

Erfolgsentscheidend ist eine agil genutzte Agilität, nicht das dogmatische und schmerzhaft Verfolgen irgendeines Frameworks.

„Agile Agilität“, keine normierte

Führt diese „agile Agilität“ nicht zu einer überfordernden Vielfalt? Sollte Agilität nicht eher zu einer markanten Vereinfachung führen?

Ja – Vereinfachung, aber im Sinn eines drastischen Abbaus des zentralistischen Mikromanagements überall dort, wo es nicht passt, zugunsten des Vertrauens in die Selbststeuerungsfähigkeit sozialer Systeme.

Keine Simplifizierungen!

Vereinfachung jedoch nicht in dem Sinn, dass die Gesamtheit der unterschiedlichsten internen Steuerungsformen insgesamt betrachtet einheitlicher und einfacher wird. Vereinfachung darf nicht als Simplifizierung verstanden werden, dass z. B. alle Organisationseinheiten eines Unternehmens die identischen und einfachen Arbeitsregeln anwenden. Das wäre unangemessen.

Dieses Dilemma beschreibt der Philosoph Paul Valéry gut mit dem Satz: „Alles Einfache ist falsch, alles Komplizierte unbrauchbar.“

4 Wie beginnt ein Unternehmen seine ganz individuelle Reise zu seiner eigenen Art von Agilität?

Zentrale Frage

Zu Beginn dieser Reise ist es wichtig zu klären, was denn diese „unternehmensindividuelle Art der Agilität“ bedeutet.

Mithilfe der „Zeitmaschine“ konnten Sie ja die „besten Hoffungen“ für ein an Ihr Unternehmen angepasstes Vorgehens-Framework für eine optimale SW-Entwicklung als konkrete Ziele formulieren. Um diese Ziele zu erreichen, wird die Reise ziemlich sicher durch diese zwei Kontinente führen:

- den Kontinent der Vorgehensweisen: Arbeitsergebnisse/Arbeitsschritte/Arbeitsmethoden
- den Kontinent der Unternehmenskultur: Kommunikations-/Kooperations-/Führungsmethoden

4.1 Der Kontinent der Vorgehensweisen: Arbeitsergebnisse/Arbeitsschritte/Arbeitsmethoden

Dieser Kontinent umfasst die „Mechanik“ der SW-Entwicklung und bildet den größten Teil dessen, was in den Beschreibungen der Tätigkeiten, Koordinationsgefäße (Meetings), Rollen und Ergebnisse der diversen Vorgehensmethoden dokumentiert ist.

Vorgehensweisen: polymorph und situativ

Dieser Kontinent ist höchst polymorph: Je nach Aufgabenstellung nimmt er eine andere Gestalt an. Wenn versucht wird,

seine Gestalt trotz geändertes Ziel beizubehalten, dann kommt die Entwicklung erheblich störender

Das sind die drei polymorphen Kontinente:

Arbeitsergebnisse

- Ergebnisse, die innerhalb des Arbeitsgebietes als Grundriss erzeugt werden und nachher relevant sind,
 - entweder vordefiniert
 - oder situativ in Abhängigkeit von den Umständen
- Software, die effektiv genutzt werden kann
- Weitere Ergebnisse (Normen, Anleitungen, Auslegungsrhythmen, Qualitätsnachweise, Realisierung der SW für die Veränderung unverzichtbar)

Arbeitsschritte

- Entweder im Detail für die Ausführung, soweit dies erforderlich ist, muss,
 - oder als so weit wie möglich vorgegeben, sodass sich der Prozess anpassen und empirisch validieren lässt

Arbeitsmethoden

- Modellierungsmethoden

- Designpatterns
- Programmieretechniken
- Clean Code Development und (einige) XP-Praktiken
- Testmethoden/Testwerkzeuge
- ...

4.2 Der Kontinent der Unternehmenskultur: Kommunikations-/Kooperations-/Führungsmethoden

Dieser Kontinent hat bei jedem Unternehmen seine eigene, unverwechselbare und recht stabile Gestalt. Sie ist das Ergebnis jahrelanger Prozesse der Organisations-, Personal- und Führungsentwicklung, geprägt von der permanenten Kommunikation und Kooperation aller am Unternehmen in irgendeiner Form beteiligten Menschen.

Kultur: längerfristig stabil

Seine drei augenfälligsten Landschaften sind:

- Kommunikationsmethoden
- Kooperationsmethoden
- Führungsmethoden

Es sind altbekannte Landschaften, oft jedoch unwirtlich und ungepflegt. Mangel an Methoden, um diese Landschaften lebensfreundlich zu machen, herrscht nicht. Es mangelt sehr oft jedoch daran, sie auch einzusetzen. Erfolgreich eingesetzt wurden sie u. a. bei SEMCO, der Morning Star Company und in Firmen, die „Beyond Budgeting“ bzw. den „Beta Codex“ verfolgen (3.4 in [3]). Hinzuzurechnen sind auch die Arbeit in sich selbst steuernden teilautonomen Gruppen (etwa bei Daimler-Benz in den 1920ern und bei Volvo und Saab in den 1980er- und 1990er-Jahren) und die verschiedenen Formen partizipativer Führung und Organisationskonzepte wie So-

ziokratie und Holacracy, auf die in Kap. 08A03 eingegangen wird.

Kulturveränderungen sind schwierig

Der Versuch, diese Methoden im Rahmen etablierter Führungsprinzipien und der heute üblichen Kommunikations- und Kooperationsweisen im Unternehmen einzuführen ist schwierig: Es erfordert ein grundsätzlich verändertes Denken und Handeln. Diese Forderung nach einem neuen Denken und Handeln wird jedoch bereits seit Jahrzehnten erhoben.

Neues Denken gefordert

Dieses veränderte Denken und Handeln setzt voraus, dass beginnend bei der Unternehmensspitze

- das „Alles-im-Griff-haben“ ersetzt ist durch ein „vertrauensvolles Loslassen“. Also: „Kontrolle ist gut – Vertrauen ist besser“;
- die Einsicht vorherrscht, dass ein Unternehmen erst durch die fortlaufende Kommunikation aller vom Unternehmen umfassten Mitarbeitenden und Stakeholder entsteht, nicht durch von oben verkündete Visionen oder verordnete Strukturen.

Vertrauen ist besser als Kontrolle

Mit Kommunikation zur Organisation

Die beiden Kontinente sind im Wesentlichen unabhängig voneinander – mit Ausnahme einer entscheidenden Verbindung – allerdings nur als Einbahnstraße:

Wichtig!

Der etablierte Kontinent der Unternehmenskultur bestimmt, wie polymorph der Kontinent der Vorgehensweisen sein kann oder muss, nicht aber umgekehrt.

So kann es sein, dass der Kontinent der Vorgehensweisen eben nicht oder nicht schnell und konsequent genug eine zur je-

weiligen Aufgabenstellung der SW-Entwicklung passende Gestalt annehmen kann, weil die Unternehmenskultur das nicht zulässt.

Es kann aber auch sein, dass der Kontinent einer sehr experimentierfreudigen Unternehmenskultur vom Kontinent der Vorgehensweisen immer wieder neue und modische Gestalten verlangt, obwohl sich die Aufgabenstellung der SW-Entwicklung nicht essentiell verändert hat.

In beiden Fällen führt das zu „erdbebenartigen Konflikten“ an diesen Berührungspunkten:

- Sind alle entwicklungsinternen und nur temporären Zwischenergebnisse im Detail vorzudefinieren oder werden sie situativ in Absprache mit den jeweils Beteiligten vereinbart?
- Ist die Abfolge der Ergebniserarbeitung im Detail festgelegt oder nur in Form so weit wie möglich Raum bietender Rahmenregeln, sodass sich der jeweilige Arbeitsprozess situativ passend und empirisch navigierend ergibt?

Diese zwei „Erdbebenzonen“ hängen direkt mit dem Erfüllungsgrad der o. e. zwei kulturellen Voraussetzungen zusammen:

- Inwieweit ist das „Alles-im-Griff-haben“ ersetzt durch ein „vertrauensvolles Loslassen“?
- Inwieweit ist bewusst, dass ein Unternehmen durch die fortlaufende Konversation und nicht durch definierte Strukturen entsteht?

Kultur prägt die Vorgehensweisen, nicht umgekehrt

Also: Die Unternehmenskultur prägt die Vorgehensweisen, nicht umgekehrt. Wenn in einem Unternehmen mit enger

operativer Steuerung, eine dem Zwang zu detaillierter Steuerung begonnen wird, M zu führen, die auf einer situativen Kommunikation und Kooperation von autonomen Teams und autonomen Teams und Kommunikation und Kooperation dann verändert das nicht gemeint: Diese neuen Methoden so lange „verbogen“, bis sie in die Unternehmenskultur passen – und verlie

5 Zuerst geht die Kultur, dann erst die Vorgehensweisen

Das heute am meisten verbreitete

- Flexibilität und Adaptivität
- plus „Lean Management“
- plus diverse post-tayloristische Überzeugungen (siehe

und findet seinen Ausdruck im Manifest“.

„Agilität“ in diesem Sinne irgendeiner der „agilen“ Methoden zu führen, sondern in der Kultur des Unternehmens „Organisational Loslassen“ und „Organisation“ verändert.

Agilität ist zuallererst eine Kulturveränderung

Erst dann kann der polymorphe Kontinent der Vorgehensweisen die zur jeweiligen Aufgabenstellung passende Gestalt annehmen. Das kann z. B. Scrum, SAFe, PRINCE2, PMBOK, CMMI, RUP, V-Modell oder HERMES sein. Oder in einigen Jahren etwas, das wir heute noch gar nicht kennen. „Passend“ kann nämlich je nach Situation etwas davon sein:

- adaptiv oder planbestimmt,
- inkrementell oder kontinuierlich liefernd,
- iterativ/empirisch oder linear/analytisch-konzeptionell.

Wichtig!

Investitionen in die Unternehmenskultur sind langfristig und strategisch. Sie bilden die tragfähige Basis für kurzlebige und stets anzupassende, jeweils PASSENDE, Vorgehensweisen. Investitionen nur in Vorgehensweisen sind Verschwendung.

Natürlich ist es einfacher, vor allem in Vorgehensweisen zu investieren:

- Man kann deren Beschreibungen kaufen und deren Verwendung vorschreiben.
- Man kann Schulungen, Coachings, Beratungen und Zertifizierungen von Experten kaufen.
- Man kann Organisationsstrukturen neu zeichnen (und hoffen, dass die befolgt werden).
- Man kann recht schnell als Erfolgsnachweis zeigen, dass es neue und veränderte Formen von Arbeitsergebnissen gibt (nicht aber deren nachhaltig positive Wirkung auf das Gesamtergebnis).

Investitionen in die Unternehmenskultur hingegen lassen sich nicht in diesem Sinn „machen“. Und deren Wirkung lässt sich nur bedingt „beweisen“ und ist langfristiger Natur.

Es sind unternehmerische Entscheidungen, keine „Management Issues“.

6 Reiseführer durch den Kontinent der Kultur

Dieser „Reiseführer“ beschreibt keine empfehlenswerten Routen, sondern für Ihr Unternehmen möglicherweise interessante Gegenden. Welche Sie wählen und wie Sie ihre Route zusammenstellen, hängt ganz von Ihrem Unternehmen ab. Und zwar von jenen mithilfe der „Zeitmaschine“ erkannten Zielen, die sich auf den „Kontinent der Kultur“ beziehen, also auf Kommunikations-, Kooperations- und Führungsmethoden, die Gegenstand der Organisations-, Personal- und Führungsentwicklung sind.

6.1 Das sind die Reiseziele:

Rufen Sie sich jetzt jene Ziele Ihres Unternehmens in Erinnerung, die sich auf den „Kontinent der Kultur“ beziehen, und notieren Sie hier Stichworte dazu:

Notiere die Ziele „Kontinent der Kultur“

6.2 Welche Ziele, welche Kultur sind „richtig“?

Es gibt keine generell „richtigen“ Ziele und nicht die „richtige“ Kultur. Entscheidend ist, dass sie zur aktuellen Situation des Unternehmens (seinem Markt und seinen vorhandenen Ressourcen) passen.

Vier Arten von Organisationskulturen William Schneider [5] unterscheidet vier Arten von Organisationskulturen in Abhängigkeit davon, was der Zweck der Organisation/des Unternehmens ist (siehe Abb. 1).

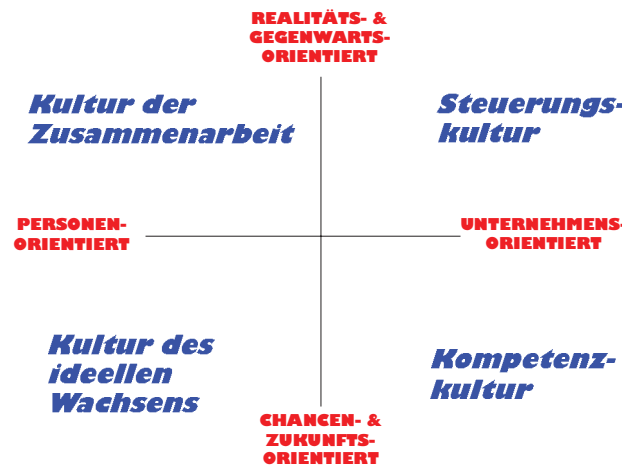


Abb. 1: Kulturdimensionen nach William Schneider [5]

Auch eine „Steuerungskultur“ kann passen So passt eine „Kultur der Steuerung“ zu Unternehmen für sehr sicherheitskritische technologische Produkte oder zu Unternehmen in sehr reifen Märkten („Cash-Cow“-Produkte). Zu Unternehmen in sehr spezifischen Marktnischen hingegen

passt eine „Kompetenz-Kultur“ Kunst und Kultur un reich Kunst und Kultur un eine „Kultur des ideeller Zusammenarbeit“ passt d wicklung gemeinsam mit etwa im Bereich der Ausl reich der experimenteller Services.

Verschiedene Kulturen im selben Unternehmen In größeren Unternehmen che Kulturen etwa in En chen geben. Aus dem gl auch ganz unterschiedlic bestimmte Vorgehensme weite Vorgehensweisen s wie der Versuch, „eine“ U

Ein im Sinn des Agilen M nur eine der möglichen F sche Kontexte. Angemes gemeinen ist, dass es über

Sechs Fähigkeiten **Robustheit:** die Fähigkeit gungsübergreifend effekti

Belastbarkeit: die Fäh Schäden oder einer destab zu erholen oder sich dara

Reaktionsfähigkeit: die l Umgebung rechtzeitig zu

Flexibilität: die Fähigke einzusetzen und nahtlos v

Innovationsfähigkeit: die Fähigkeit, neue Dinge zu tun, und die Fähigkeit, alte Dinge auf eine neue Art und Weise zu tun;

Anpassungsfähigkeit: die Fähigkeit, Arbeitsprozesse zu ändern, und die Fähigkeit, die Organisation zu ändern.

Im Folgenden werden einige Gegenden des Kultur-Kontinents vorgestellt und in Kap. 08A03 im Detail behandelt, deren Besiedelung sich lohnt. Ob und wie Ihr Unternehmen das nutzen möchte, ist Ihnen überlassen. Es sind Denk- und Handlungsoptionen.

6.3 Kommunikationsmethoden

Ein Unternehmen entsteht durch die fortlaufende Kommunikation und Kooperation aller vom Unternehmen umfassten Mitarbeitenden und Stakeholder, nicht durch von oben verkündeten Visionen oder verordnete Strukturen. Kommunikationsblockaden sind daher Organisationsblockaden. Deshalb haben die Methoden der Kommunikation eine zentrale Bedeutung für jedes Unternehmen.

Erprobte Methoden

Einige erprobte Methoden dazu finden Sie in Kap. 08A03 ausführlicher vorgestellt, und zwar:

- Grundannahmen gelingender Kommunikation
- „Ein-Weg-Kommunikation“ ist *keine* Kommunikation: Konversationsräume schaffen!
- Daily Stand-up
- Erfahrungs- und Praxisgemeinschaften bilden und fördern
- Transparente Informationen
- Unternehmensinterne Kommunikation via Social Media

6.4 Kooperationsmethoden

Sehr oft werden die Methoden und Techniken der Zusammenarbeit in einem Team oder einer Gruppe nur im Zusammenhang und vor dem Hintergrund eines spezifischen Vorgehensmodells dargestellt. Das erweckt den Eindruck, als wären Methoden und Techniken nutzbar nur in Verbindung mit und jeweils ganz spezifisch für ein bestimmtes Modell.

Im Kap. 08A03 wird beispielhaft gezeigt, dass es von Vorgehensmodellen unabhängige Kooperationsmethoden gibt, die die Kultur der Zusammenarbeit im Allgemeinen unterstützen. Es handelt sich dabei um Methoden in diesen Bereichen:

- Sichtbarkeit, Verfolgbarkeit, Analysierbarkeit der Zusammenarbeit.
- Arbeitsplanung im Team *durch das Team* – nicht durch den Teamleiter.
- Kontinuierliche Verbesserung der Kooperation im Team.
- Teamarbeit mit Elementen von Scrum.
- Optimierung der Teamstruktur:
 - Vor- und Nachteile von Feature-Teams.
 - Vor- und Nachteile von Component-Teams.
 - Kombination beider Strukturierungsprinzipien als pragmatische Lösung.

6.5 Führungsmethoden

Die diversen Vorgehensmodelle beschreiben spezifische Rollen der SW-Entwicklung und des Projektmanagements, lassen aber deren Einbindung in die Führungsstruktur des Gesamtunternehmens offen.

**Vorgehens-
modelle
lassen
Führungs-
methoden
offen**

So etwa kennt PRINCE2 einen „Teammanager“, der jedoch nur als Repräsentant und Schnittstelle eines der – projekt-spezifischen – Entwicklerteams gegenüber dem Projektmanager für die Erstellung, Prüfung und Lieferung der Produkte verantwortlich ist und der bei kleinen Projekten entfallen kann (dann nimmt der Projektmanager die Aufgaben des Teammanagers gegenüber den Teams wahr). Und Scrum kennt pro Scrum-Team einerseits einen Scrum Master als „dienende Führungskraft des Teams“, d. h. als Verantwortlichen dafür, dass Scrum verstanden und korrekt angewandt wird, und andererseits einen Product Owner als Verantwortlichen für die Maximierung des Werts des Produkts und der Arbeit, die das Entwicklungsteam verrichtet. Weiterhin heißt es im Scrum Guide [7]: *„Für unterschiedliche Organisationen, verschiedene Scrum-Teams und Einzelpersonen kann es viele Wege zur Erreichung dieses Ziels geben.“*

**Behindert der
Teamleiter
das Team?**

Das führt im Fall von Scrum zu Behauptungen, dass es insgesamt nur die Rollen Product Owner – Entwicklungsteam – Scrum Master gebe und dass ein Teamleiter im Widerspruch zu Scrum stehe.

Generell führt das oft zu Entwicklungsteams, deren Mitglieder im Rahmen einer Matrixorganisation aus diversen Organisationslinien zusammengesetzt sind. Die beurteilende und fördernde Führungskraft der jeweiligen Organisationseinheit ist dann aber recht weit entfernt vom Alltag des Teams. Auf der Basis welcher Beobachtungen und Informationen können dann diese – weit entfernten – Führungskräfte „ihre“ Mitarbeitenden beurteilen und fördern? Es entsteht quasi eine „Führungslücke“ und die Informationen über die Mitarbeiter kommen auf indirekten Wegen zur Führungskraft, z. B. – ein konkreter Fall – via schriftliche 360-Grad-Feedbacks von Personen, die der/die Mitarbeitende selbst ausgesucht hat.

Besser und transparenter bilden (siehe dazu Kap. C Teamstruktur“) mit je ein arbeitenden Teamleiter.

**Der Team-
leiter als
Scrum Master!**

Dieser kann dann, z. B. Teammanagers und bei S nehmen. Unter „Führung die Details der Arbeiten trollierenden „Hierarcher Teamarbeit setzende und individuell fördernde Füh

Bei Unternehmen, die de sungen und auf umfassende Individuallösungen setzen Teams gut möglich. Im G lung als Dienstleistung v jeweils pro Projekt unters kurzlebigen Teams gepräg nicht opportun. Firmen n Regel eher klein und ehr Die Problematik „formel weniger.

Im Kap. 08A03 wird vert

- Anpassbare Kreise sta
- Change Event: Nur d lassen.
- „Agile“ Art der Führu
- Führen von Teams
- Der Teamleiter als Füh

6.6 Wer ist der Reiseleiter durch die Landschaften des „Kontinents der Kultur“?

Die Reise durch den Kontinent der Kultur führt Sie durch diese Landschaften:

- Kommunikationsmethoden
- Kooperationsmethoden
- Führungsmethoden

Sie werden in Kap. 08A03 im Detail dargestellt.

Wer aber führt Ihr Unternehmen auf der „Individualreise“ durch diesen Kontinent der Kultur? Wer reist alles mit?

Die schlechte Nachricht: Es gibt keinen Reiseleiter, den Ihr Unternehmen dafür engagieren kann.

Die gute Nachricht: Sie haben bereits genug Reiseleiter. Alle Manager und Führungskräfte des Unternehmens – beginnend beim Topmanagement – sind die Reiseleiter. Das nämlich ist eine ihrer ganz zentralen Leitungsaufgaben.

Die Führungskräfte selbst sind das „Transition Team“

Und diese Aufgaben können die Führungskräfte nicht an ein „Transition Team“ oder an „Change Agents“ delegieren. Die Führungskräfte selbst sind das „Transition Team“, die „Change Agents“. Die Reise beginnt damit, dass alle Manager und Führungskräfte des Unternehmens, unterstützt von qualifizierten OE-/PE-Beratern, zu „Reiseleitern“ werden.

Bottom-Up nicht machbar

Nicht nur die bloße Unterstützung durch die Entscheidungsträger, beginnend beim Topmanagement auf Geschäftsleitungsebene, ist essentiell. Erfolgsentscheidend ist deren beispielhaftes Vorleben. Aktuelle Erkenntnisse der Organisa-

tionsentwicklung widersprechen klar der Machbarkeit einer Veränderung allein „von unten nach oben“ (bottom-up), oft verbunden mit dem Infragestellen hierarchischer Machtstrukturen. Dean Leffingwell stellt dazu fest, dass in der „traditionellen“ agilen Denkweise Manager nur als Unterstützer und Hindernisbeseitiger gesehen werden. Im Gegensatz dazu wirken die dem „Lean“-Gedanken folgenden Manager als Führungskräfte der Veränderung und als mit den Basisprinzipien vertraute „Lehrer“ [8].

Kein „Change Team“: Alle machen von Anfang an mit

Die Reisenden sind *alle* Mitarbeitenden des Unternehmens und kein temporär zusammengestelltes Spezialteam, das „das Unternehmen verändert“, damit die große Mehrheit der Mitarbeitenden weiterhin möglichst ungestört ihrer Arbeit nachgeht und irgendwann von diesem Spezialteam unterwiesen wird, bestimmte Dinge zu ändern. Ein Unternehmen zu verändern ist ein hoch komplexes Kunstwerk. Die Veränderungskünstler sind kein „Change Team“ sondern *alle* Mitarbeitenden, *diese* leisten die Veränderung. Unterstützt werden sie dabei von den Führungskräften.

Schrittweises Lernen und Adaptieren

Solche Veränderungen sollten nur bei einem erheblichen Handlungsdruck linear geplant und als „Big Bang“ in aller nötigen Tiefe und Breite umgesetzt werden. In allen anderen Fällen ist ein schrittweises Lernen und Adaptieren besser. Agilität als solche sollte mit einem adaptiven und inkrementellen Vorgehen schrittweise eingeführt werden.

Wichtig!

Die Veränderung des Unternehmens ist kein Sonderfall, kein spezielles Projekt mit einem definierten Ende. Sie ereignet sich zeitlich unbegrenzt und als schrittweise Adaptionen im Rahmen der tagtäglichen Arbeit.

Entscheidend dabei ist die unternehmensweite transparente Information über die Veränderungsvorhaben, insbesondere auch des mittleren und Topmanagements. Ermöglicht wird diese Transparenz z. B. mit offengelegten „Roadmaps“ der anstehenden Vorhaben und mit dem offengelegten aktuellen Bearbeitungsstand, dargestellt z. B. als Kanban-Tafel (siehe Kap. 08A03, Abschnitt „Sichtbarkeit, Verfolgbarkeit, Analysierbarkeit der Zusammenarbeit“).

Die Aufmerksamkeit jedoch nur auf die Veränderungen zu richten führt zur Überforderung. Betty Zucker nennt es das „Chronic Change Fatigue Syndrom“ [9]. Zusätzlich muss auch Stabilität, also all das Vorhandene und Funktionierende ausdrücklich anerkannt und gestärkt werden.

Wichtig!

Neben dem „Veränderungsmanagement“ ist auch ein „Stabilitätsmanagement“ unabdingbar.

6.7 Die Wirkungen dieser nie endenden Reise

Gestärkte Fähigkeiten

Mit der nie endenden Reise durch die Landschaften der Kommunikations-, Kooperations- und Führungsmethoden werden diese Fähigkeiten des Unternehmens gestärkt:

- Robustheit
- Belastbarkeit
- Reaktionsfähigkeit
- Flexibilität
- Innovationsfähigkeit
- Anpassungsfähigkeit

Und damit werden die jeweiligen Aufgabenstellungen passende Vorgehensweise davon, um welche agile (sich handelt.

7 Reiseführer durch Vorgehensweisen

Auch dieser Reiseführer Kontinent der Kultur, kei dern für Ihr Unternehmen Aufgabestellung der SW Welche Sie wählen und w hängt von der Aufgabens Kultur des die Aufgabe le ab.

7.1 Was sind die Reisez

Notiere die Ziele „Kontinent der Vorgehensweisen“

Rufen Sie sich jetzt jene nennung, die sich auf den (Arbeitsergebnisse, Arbeiten, die Sie mithilfe der , tieren Sie hier Stichworte

Betrachte die Ziele kritisch. Werfen Sie einen kritischen Blick auf diese Ziele:

- Sind es konkrete, spezifische Ziele, ausgehend von den heutigen Hauptaufgaben Ihres Unternehmens/Unternehmensbereichs?
- Wie sicher ist es, dass es in fünf Jahren dieselben Hauptaufgaben sein werden?
- Sind Ihre Ziele – wenn Ihr Unternehmen SW-Entwicklung als Dienstleistung im Rahmen unterschiedlichster Kundenprojekte anbietet – eine Extrapolation Ihrer heutigen Auftragssituation?
- Wie gut können Sie heute bereits die Auftragssituation und Kundenarten in fünf Jahren und die dazu passenden Ziele für die dann angemessene oder von den Kunden vorgeschriebene Vorgehensweise abschätzen?

Meine Vermutung ist, dass Ihre längerfristigen Reiseziele, also die in fünf Jahren jeweils spezifischen Vorgehensweisen, eher unbekannt als sicher sind. Ihre Reise führt also zu mittelfristig veränderlichen und heute noch unbekanntem Zielen. Bereiten Sie sich also darauf vor, immer wieder veränderte Vorgehensweisen einzusetzen. Und: Die jeweils optimale Vorgehensweise müssen Sie rasch erkennen und praktizieren können. Das ist die erforderliche Agilität, nicht die möglichst regeltreue Implementation eines bestimmten (agilen) Vorgehensmodells wie z. B. Scrum.

7.2 Agiler Umgang mit agilen Vorgehensmodellen

Der real existierende agile Umgang mit agilen Vorgehensmodellen wurde bereits 2010 im Forrester Report „Agile Development: Mainstream Adoption Has Changed Agility“ [10] als gängige Praxis in unterschiedlichsten Unternehmen

so beschrieben (auszugsweise zusammengefasst durch den Autor des vorliegenden Beitrags):

Agiler Umgang mit agilen Vorgehensmodellen die Regel

Sogar die am weitesten fortgeschrittenen Anwender agiler Methoden pflücken aus unterschiedlichen Vorgehensmodellen die für sie passenden Techniken heraus und kombinieren sie mit anderen. Die einzelnen Techniken nämlich (wie z. B. „daily standup“ und „continuous integration“) können unabhängig voneinander eingesetzt werden. Ein einzelnes agiles Vorgehensmodell kann nicht alles abdecken. Und für spezifische unterschiedliche Herausforderungen müssen agile Teams ihre Praktiken und Techniken angemessen ändern können.

Ebenso werden in der „traditionellen“ SW-Entwicklung agile Techniken und Praktiken wie z. B. kurze Iterationen und daily standup genutzt. Das zeigt, dass es den Teams mehr um die Verbesserung der Zusammenarbeit und um das Entwickeln qualitativ hochstehender Software geht und nicht darum, ihren etablierten Prozess der SW-Entwicklung komplett zu verändern. Die meisten Teams verstehen „Agilität“ eher als Ethos und Philosophie der Arbeit denn als bestimmte Entwicklungspraktiken wie z. B. eXtreme Programming oder Scrum.

Agilität eher als Ethos denn als Entwicklungspraktik

Darum stelle ich die kulturellen Aspekte (Kommunikations-, Kooperations- und Führungsmethoden) als langfristig wirksame Basisinvestition an den Anfang, und nicht die sich rasch ändernde „Mechanik“ der SW-Entwicklung als solche.

Weiterhin ist im Forrester Report zu lesen, dass *die Kombination unterschiedlicher agiler Praktiken sehr gut funktioniert, um auch innerhalb längerer Releasezyklen agil zu arbeiten, um den Anforderungen von Governance und Compliance Rechnung zu tragen und um mehr als nur die Lieferung*

funktionierender Software im Auge zu haben. Um die für die jeweilige Situation geeignete Mischung agiler Praktiken und Techniken zu finden, sollten die Teams von agilen Coaches unterstützt werden.

Interne Coaches statt permanenter externer Berater

Unternehmen und Unternehmensbereiche mit der SW-Entwicklung als Kerngeschäft sollten über solche agilen Coaches selbst verfügen und in diesem strategisch wichtigen Bereich nicht primär auf externe Berater angewiesen sein.

Es muss auch unterschieden werden zwischen Coaches und Beratern für die heute „agil“ genannten (jedoch nicht neuen) Praktiken und Methoden im Bereich der Organisations-, Personal- und Führungsentwicklung einerseits und Coaches und Beratern für die Praktiken und Methoden der eigentlichen SW-Entwicklung andererseits.

Kulturveränderung erfordert qualifizierte OE-/PE-Berater

Da „Agilität“, gemäß Forrester-Report, eher als Ethos und Philosophie der Arbeit denn als SW-Entwicklungspraktik gesehen wird, tendieren einige Berater agiler SW-Entwicklungspraktiken dazu, auch im Bereich der Organisations-, Personal- und Führungsentwicklung tätig zu sein, ohne jedoch über die dazu nötigen speziellen Kenntnisse und Erfahrungen zu verfügen. Nutzen Sie in diesem Bereich besser erfahrene Berater, die auf Organisations-, Personal- und Führungsentwicklung spezialisiert sind.

„Polychrome“ Coaches nötig

Bei der Auswahl eines Coaches für agile SW-Entwicklungspraktiken sollten Sie besonderen Wert auf seine breite Kenntnis und Praxiserfahrung sowohl in mehreren agilen (z. B. XP, FDD, Crystal, Kanban und Scrum) als auch in nicht unbedingt als agil geltenden Vorgehensmodellen und Entwicklungsmethoden (z. B. RUP, V-Modell, DSDM und PRINCE2) legen. Eine all diese Modelle und Methoden übergreifende Qualifikation wie z. B. die PMI-ACP-Zertifizierung

zung ist einer Zertifizierung in jedem Fall vorzuziehen, da die für einen Coach ein Jahr erforderlich ist.

7.3 Einladung zur chaos

Wenn die von den jeweiligen abhängigen freien Konzepte und Methoden durch die erfolgreich ist – gibt es da ein alle Teams umfassen eine und nur eine Komplexität? Oder geht es um Fähigkeiten, die jeweils passenden Praktiken und Methoden

Die Teams bestimmen ihre Praktiken selbst

Ja, es geht primär darum, passenden Praktiken und Methoden zu finden. Und zwar so, dass sie zu leiten können. Die Teams bestimmen und leisten Teamleiter (als „Agile Manager“ unterstützen. Teams sind durch selbstorganisierende Arbeit (siehe Guide 2011 zu lesen). Wie bestimmtes Vorgehensweisen Was gesetzten Rahmens von Widerspruch zur „agilen“ aus Unterstützung verlangen oder Elemente davon

Pilotprojekte sind kein Maßstab

Da innerhalb dieses geschäftlichen Vorgehens, Methoden

ist, von den Teams selbst bestimmt wird, macht es auch keinen Sinn, mit ein oder zwei Pilotprojekten spezielle agile Vorgehensweisen zu testen. Diese zeigen nicht die generelle Brauchbarkeit, sondern nur die Brauchbarkeit für den jeweiligen Kontext und die Pilotteams. Derartige Pilotprojekte haben zudem in der Regel

- wenig Abhängigkeiten,
- keine Wartung von Altlasten und
- ein motiviertes Team (weil es ein Pilot ist).

Unter solchen Voraussetzungen funktioniert nahezu jedes Vorgehen überdurchschnittlich gut.

Qualitätsanforderungen stellen, statt Vorgehen vorschreiben

Und wie kann dann sichergestellt werden, dass einzelne Teams Software mit der erforderlichen Qualität und unter Beachtung genereller Programmierstandards als Voraussetzung für die Wartbarkeit und Erweiterbarkeit entwickeln?

Damit, dass die erforderliche Qualität, Wartbarkeit und Erweiterbarkeit mit überprüfbaren Abnahmekriterien als Bestandteil des *Was*, etwa als „nichtfunktionale Anforderungen“, vorgegeben werden.

Trotz all dieser Vielfalt gibt es einige generelle Muster zum Vorgehen beim Umstellen auf agile SW-Entwicklung. Einige davon werden nun behandelt.

7.4 Generelle Muster zum Umstellen auf agile SW-Entwicklung

7.4.1 Agile Kultur

Das Schaffen einer agilen Kultur im Unternehmen ist die unabdingbare Voraussetzung für den agilen Umgang mit agilen und nicht agilen Vorgehensmodellen. Wie das gelingt, wurde im Abschnitt 6 „Reiseführer durch den Kontinent der Kultur“ beschrieben. Vieles davon, was unter „agile SW-Entwicklung“ subsumiert wird, wird dann als Ergebnis der Reise durch den „Kulturkontinent“ in Ihrem Unternehmen bereits praktiziert werden:

Praktizierte Kultur . . .

- Etablierte Konversationsräume
- Daily Stand-up
- Erfahrungs- und Praxisgemeinschaften
- Transparente Informationen
- Unternehmensinterne Kommunikation mittels Social Media
- Sichtbarkeit, Verfolgbarkeit, Analysierbarkeit der Zusammenarbeit
- Arbeitsplanung im Team durch das Team – nicht durch den Teamleiter
- Kontinuierliche Verbesserung der Kooperation im Team
- Teamarbeit mit Elementen von Scrum
- Optimierung der Teamstruktur
- Anpassbare Kreise (statt starrer Hierarchien) mit Führungsmeetings, wöchentlichen taktischen Meetings und Daily Stand-up Meetings
- Change Events: Nur den Rahmen setzen, das Bild malen lassen

**... führt zu
fortlaufender
Entwicklung**

- „Agile“ Art der Führung
- Führen von Teams
- Teamleiter – neu definiert

Das führt zu einem Unternehmen, das diese Fähigkeiten fortlaufend weiterentwickeln kann:

- Robustheit
- Belastbarkeit
- Reaktionsfähigkeit
- Flexibilität
- Innovationsfähigkeit
- Anpassungsfähigkeit

Das ist der erste und entscheidende Schritt auf dem Weg zur Umstellung in Richtung agile SW-Entwicklung.

Der nächste – von irgendwelchen agilen oder nicht agilen Vorgehensmodellen unabhängige – Schritt ist das Erreichen und Weiterentwickeln von „Craftsmanship“. Denn ohne SW-Entwicklung als professionelles Handwerk scheitern alle Vorgehensmodelle.

**7.4.2 Craftsmanship: SW-Entwicklung als
professionelles Handwerk****Menschen
machen
Software**

Gute Software mit der erwünschten Funktionalität innerhalb des festgelegten Zeit- und Kostenrahmens entsteht nicht primär wegen irgendeines Vorgehensmodells, sondern durch Menschen, die das Handwerk der SW-Entwicklung professionell ausüben. Das sind Menschen, die schlechte Software (funktional ungenügend, schlecht wartbar und erweiterbar, zu teuer) mit ihrer Berufsehre nicht vereinbaren können und

daher auch alle Ansinnen ablehnen, zurückweisen. Ebermann einwilligen wirklich schlechtem Holz oder nicht. Er wird solche Aufträge ablehnen. SW-Entwicklung der Nord

**Manifesto for
SW-Crafts-
manship**

Genau das fördert das „Manifest“ [11]:

„Als engagierte Softwareentwickler für professionelle Software zu arbeiten und anderen dabei zu helfen. Durch diese Tätigkeit lernen wir:“

- *Nicht nur funktionierende fertige Software*
- *Nicht nur auf Verändern Mehrwert zu schaffen*
- *Nicht nur Individuen und Gemeinschaft aus Erfahrung*
- *Nicht nur Zusammenarbeiten produktive Partnerschaften*

Das heißt, beim Streben nach Qualität halten wir die Werte aufrecht

**Clean Code
Development**

In engem Zusammenhang mit Clean Code Development sind Clean Code Developer auch

„In Clean Code geht es nicht um die Programmierparadigmen“

von .NET oder Java oder ASP.NET oder SVN oder OOP sein, um aus ihm Gewinn zu ziehen. Es dreht sich vielmehr um das unter all dem liegende Substrat: Code als Quelltext und Code als strukturierter Ausdruck von Funktionalität. Für Code als kleinsten gemeinsamen Nenner zwischen Softwareentwicklern aller Couleur beschreibt Clean Code eine Menge von Prinzipien und Best Practices als kleinsten gemeinsamen Nenner.“ [12]

Professionalität

Zum Verständnis von Professionalität heißt es außerdem:

- „Ein professioneller Softwareentwickler setzt sich mit dem Metier bewusst auseinander. D. h., er reflektiert über sein Produkt, seine Arbeitsweise, seine Materialien und Werkzeuge. Ein professioneller Softwareentwickler ist nicht einfach zufrieden, wenn sein Chef oder Kunde zufrieden ist. Er ist auch nicht einfach mit dem zufrieden, was ein Hersteller ihm empfiehlt. Stattdessen beobachtet und prüft er ständig seine Ergebnisse und bemüht sich um die Weiterentwicklung seiner selbst wie auch des Metiers.“
- Ein professioneller Softwareentwickler hat ein inneres Wertesystem. Gegen dieses Wertesystem prüft er seine Ergebnisse und Handlungen. Nur wenn seine Arbeit diesem Wertesystem entspricht, empfindet er sie als gut getan, als professionell. Sein Streben ist es daher, auch unter widrigen Umständen, unter Druck von Kunden oder Herstellern, diesem Wertesystem treu zu sein.“ [12]

Prinzipien und Best Practices

Zu den Prinzipien und Best Practices von Clean Code gehören:

- Quelltextformatierung (engl. code conventions),
- Entwurfsmuster (engl. design patterns),

- Konvention vor Konfiguration (engl. convention over configuration),
- eine reichhaltige Menge von Vorschlägen aus dem Buch „Clean Code“ von Robert C. Martin [13].

Zur Software Craftsmanship gehört auch das:

1. (Re-)Professionalisierung des Requirement Engineering

Professionelles RE

Listen (sogenannte Backlogs) von „Stories“ ohne erkennbaren Zusammenhang genügen nicht, um die Anforderungen in ihrem Gesamtzusammenhang zu verstehen und zu managen. Nach wie vor ist die Anwendung professioneller und toolunterstützter Methoden des Requirement Engineering unabdingbar. Word-Dokumente oder E-Mails oder nur persönliche Gespräche genügen nicht. Agiles Requirement Engineering ersetzt die zu Beginn der Realisierungsphase den Entwicklern vor die Füße geworfenen umfassenden Detailspezifikationen des „Big Design Up Front“ (BDUF) durch eine schrittweise Detaillierung von Release zu Release auf der Basis eines vor dem ersten Release erstellten „Just Enough Design Up Front“ (JEDUF).

2. TDD und ATDD und die Integration von Entwicklung und Test

TDD und ATDD

Um in kurzen Iterationen oder auf Basis eines kontinuierlichen Arbeitsflusses potenziell nutzbare Ergebnisse zu erzeugen, müssen sie kontinuierlich im Rahmen der Entwicklung und nicht erst danach getestet werden. Sonst ist das Risiko sehr groß, dass erhebliche Fehler zu spät entdeckt werden. Test-Driven Development (TDD), aber auch Acceptance Test-Driven Development (ATDD) unterstützen diese Integration von Entwicklung und Test.

CI, CD, Test-automatisierung**3. Continuous Integration, Continuous Delivery und Testautomation**

„Potenziell nutzbar“ bedeutet nicht, dass die Ergebnisse nur bereit zur Integration sind. Es bedeutet „in der Integrationsumgebung technisch und funktional getestet und bereit für die Produktionsumgebung“. Das erfordert eine möglichst frühzeitige, im Idealfall kontinuierliche Integration und Auslieferung mindestens in die der Produktionsumgebung unmittelbar vorgelagerte produktionsnahe Testumgebung. Das bedeutet auch, dass in dieser und den ihr vorgelagerten Umgebungen automatisierte Regressionstests mit einem hohen Abdeckungsgrad vorhanden sind.

4. Applikations- und Systemarchitektur modellieren und bereinigen**Architektur**

Rasche oder gar kontinuierliche Integration und Testautomation sind umso schwieriger zu realisieren, je heterogener die Technologie der Systemlandschaft ist. Auch die Teamstrukturierung und damit die Schnittstellen zwischen den Teams sind umso komplizierter, je heterogener die Applikations- und Systemlandschaft ist. Und diese Heterogenität erschwert insbesondere die Bildung von „Feature Teams“, die über alle nötigen Kompetenzen verfügen. Die Bereinigung und Vereinfachung der Applikations- und Systemarchitektur erleichtert das agile Arbeiten erheblich. Und ein stets aktuelles Modell der Applikations- und Systemarchitektur ermöglicht das koordinierte Arbeiten vieler Teams und die zügige Realisierung von Change Requests ohne vorangehende aufwändige „Systemarchäologie“.

... und noch mehr

Weitere Aktionsfelder einer professionellen SW-Entwicklung sind [14]:

- Entwicklung serviceorientierter
- Investitionen in hochskalierbare Lösungen
- Orientierung an etablierten Plattformen, die eine Integration mit verschiedenen Endgeräten und Netzwerken ermöglichen
- Unterstützung von flexibel konfigurierbaren Rules Engines für den Einsatz von intelligenten
- Data-Warehouse- und Analytics-Lösungen, die auch Ad-hoc-Analysen ermöglichen

Ergebnis = tragfähige Plattform

Agile Kultur + Softwareentwicklung ergeben die traditionellsten Varianten (agile Plattform ist, umso mehr Plattformen auf agile SW-Entwicklung und Kosten nichts verbessern – Preis ist ein Irritationen und unklar. Das zeigt auch die fehlende Flexibilität, die nach geblieben aus Sicht der agilen SW-Entwicklung (Agilität/Extensibilität) verändert oder haben sich verschlechtert.

Trotz aller unterschiedlichen Varianten agiler Entwicklung (Kap. 08A04) angeboten (aber nicht als Anleitung)

**Kontext-
abhängige
Vorgehens-
muster der
Umstellung**

- Vorgehensmuster zur Umstellung auf agile Entwicklung und Unterhalt umfassender Individuallösungen
- Vorgehensmuster zur Umstellung auf agile Entwicklung und Unterhalt plattformbasierter Lösungen
- Vorgehensmuster zur Umstellung auf agile SW-Entwicklung als Dienstleistung

Das Muster für agile Entwicklung und Unterhalt plattformbasierter Lösungen ist das umfassendste, bestehend aus diesen Schritten:

- Das ist bereits erreicht: die Voraussetzungen.
- Produktlinien/Applikationsfamilien als Value Streams.
- Value Streams als Agile Release Trains (ART) implementieren.
- Plattformgruppen als Sekundärstruktur.
- Überprüfen und anpassen der Teamstruktur.
- Das Was vom Wie trennen: Pro Team ein Team-Backlog-Owner.
- Die Teamleiter bleiben Rahmensetzer für das Wie.
- Pro ART die Zeitpunkte der externen und internen Releases bestimmen.
- Programm-Management agil gestalten.
- Planung des kommenden PSI (interner Release) eines ART.
- Agile Arbeit auf Teamebene.
- Projekte innerhalb eines ART.
- Agiles Portfoliomanagement.

8 Was ist der (wirtschaftliche) Nutzen von all dem?

In der „agilen Szene“ werden, insbesondere im Zusammenhang mit Scrum, geradezu phantastische Erhöhungen der Entwicklungsgeschwindigkeit berichtet.

Phantastische Erhöhung der Entwicklungsgeschwindigkeit?

Ein typisches Beispiel ist das Paper „Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft“ von Scott Downey und Jeff Sutherland [16]. Dort wird über Fälle mit um mehr als einige hundert Prozent (!) höheren Geschwindigkeiten von Scrum-Teams im Vergleich zu „Wasserfall-Teams“ berichtet. Alle diese Berichte beruhen jedoch auf einzelnen Fällen mit eher episodischem Charakter, die überdies von überzeugten Vertretern agiler Methoden publiziert wurden.

Wissenschaftliche Studie zeigt anderes Bild

Eine unabhängige Masterarbeit an der Universität Hagen [17] zeigte im Gegensatz dazu, dass Scrum zu keiner signifikanten Erhöhung der Effizienz führt. Insgesamt (bezogen auf ein gesamtes fertiges Produkt) wird es demnach mit Scrum nicht eindeutig schneller oder billiger. Die von Verfechtern des agilen Vorgehens behaupteten massiven Verbesserungen der Entwicklungsgeschwindigkeit und Kostenreduktionen bedürfen noch der Erhärtung durch unabhängige wissenschaftliche Studien, basierend auf einer Vielzahl von Fällen.

Gelegentlich wird auch das „CHAOS Manifesto“ der Standish Group 2012 zitiert, wonach agil abgewickelte Projekte nur in 9 % der Fälle scheiterten, „Wasserfallprojekte“ hingegen in 29 %. Unterschlagen wird dabei, dass es sich bei den agilen Projekten vor allem um eher kleine bis mittlere Projekte handelt. Und gemäß CHAOS-Report der Standish Group hängt die Erfolgsrate von Projekten, unabhängig von der Vorgehensweise, stark von ihrer Größe ab: Projekte unter

	750.000 US\$ sind zu 46 % erfolgreich, jene zwischen 6 und 10 Mio. US\$ nur zu 11 %.
Auch negative Auswirkungen des agilen Vorgehens	Dass sich die Umstellung auf agile SW-Entwicklung in jedem Fall rechnet, ist also umstritten. Im Gegenteil: Sie kann, gemäß der Swiss Agile Study 2012 [18], auch zu einer eher verschlechterten Wartbarkeit und Erweiterbarkeit und zu eher höheren Entwicklungskosten führen, wobei andererseits die Handhabbarkeit geänderter Prioritäten und „time to market“ markant besser werden.
Keine bessere SW-Qualität mit „agil“	Ein ernüchterndes Ergebnis zeigt auch die in Kooperation mit den Hochschulen Bremen und Bremerhaven, der Fachhochschule Köln, der ANECON Software Design und Beratung G.m.b.H., dem German Testing Board und dem Swiss Testing Board durchgeführten „Umfrage 2011: Softwaretest in der Praxis“ [19]. Rund 800 Personen antworteten, 1/3 davon aus Firmen zwischen 100 und 1'000 Mitarbeitenden. Ergebnis war, dass es bezüglich der SW-Fehler im Betrieb kaum einen Unterschied zwischen agil und nicht-agil erstellter SW gab.
Agil dann nützlich, wenn es zum Ziel passt	Ob die Umstellung auf ein agiles Vorgehen in der SW-Entwicklung wirtschaftlichen Nutzen bringt, hängt also davon ab, was damit erreicht werden soll: Geht es primär um die schnelle Reaktionsfähigkeit oder primär um die langfristige Wartbarkeit und um eine möglichst hohe Qualität?
Zielerreichung mit Metriken verfolgen!	Wenn das Vorgehen der SW-Entwicklung verändert wird, muss also klar sein, was der Effekt sein soll, und dieser Effekt muss im Verlauf der Umstellung kontinuierlich mit geeigneten und robusten Metriken beobachtet werden.

Kulturveränderung entzieht sich der Wirtschaftlichkeitsbetrachtung	Unabhängig von der reinen Nutzen einer veränderten Führung entziehen sich je Betrachtungen. Es sind unter wie ein Unternehmen vor men werden soll und wi züglich Robustheit, Belastbarkeit, Innovationsfähigkeit
---	--

9 Fazit

Die Umstellung auf agil mit einer langen Reise: I Kontinent der Kultur, um passende Art der (agilen) Reise durch den Kultur langfristige wirksame In Praktiken im Bereich I Führung sind – unabhängig agilen Vorgehensweisen d wenn einige davon – wie Sualisierung des Arbeitsfl bestimmter agiler Entwic

Danach (und nicht vorher) ment der Vorgehensweiser Craftsmanship für eine Entwicklung jenseits all Danach geht die Reise – mens, seinen Hauptaufgal ganz unterschiedliche Vo gerüstet für einen sehr ag (agilen) Vorgehensweisen

wenigen Jahren der Kontinent der Vorgehensweisen möglicherweise von anderen, heute noch nicht erahnten, Landschaften geprägt sein könnte. Legen Sie sich also nicht allzu stark auf bestimmte, heute aktuelle Vorgehensweisen fest. Bleiben Sie auch diesbezüglich agil!

Und formulieren Sie klar, was der wirtschaftliche Effekt der Umstellung sein soll, und beobachten Sie fortlaufend mit geeigneten Metriken, wie gut dieser Effekt erreicht wird.

Quellen

- [1] Royce, Winston W.: Management of Large Software Systems. The Institute of Electrical and Electronics Engineers, IEEE Wescon, August 1970.
- [2] Manifesto for Agile Software Development, 2001, agilemanifesto.org
- [3] Korn, Hans-Peter: Agile Konzepte. In: Bartsch, Oliver, Lindinger, Markus (Hrsg.): IT-Servicemanagement. TÜV Media, 2013; Kapitel 08A01.
- [4] Korn, Hans-Peter: Blick zurück in die Zukunft. In: „Solution Tools“, Peter Röhrig (Hrsg.), manager Seminare Verlag, 2008; S. 145 ff.
- [5] Schneider, William: Why Good Management Ideas Fail – Understanding Your Corporate Culture. www.parshift.com/Speakers/Speak016.htm
- [6] Alberts, David S.; Hayes, Richard E.; Honekamp, Wilfried (Übersetzer): Power to the Edge. Re Di Roma-Verlag, 2009.
- [7] Schwaber, Ken; Sutherland, Jeff: Scrum Guide. www.scrum.org/Scrum-Guides
- [8] Vgl. Leffingwell, Dean: Lean Abstract, Abschnitt „Foundation – Management Support: Lean Thinking Manager-Teachers“, scaledagileframework.com/lean/
- [9] Zucker, Betty: Chronic Change Fatigue Syndrom. In: GDI_IMPULS (ISSN 1422-0482), 20. Jahrgang, Nr. 2.02, S. 7–11, www.bettyzucker.ch/download/chronic_change_fatigue.pdf
- [10] West, Dave; Grant, Tom: Agile Development: Mainstream Adoption Has Changed Agility. Forrester Report January 20, 2010.
- [11] manifesto.softwarecraftsmanship.org/#/de
- [12] www.clean-code-developer.de
- [13] Martin, Robert C.: Clean Code – A Handbook of Agile Software Craftsmanship. Prentice Hall International, 2008.
- [14] Nissen, Volker: Einige Grundlagen zum Management von IT-Agilität. Arbeitsbericht Nr. 2008-03, Oktober 2008, Technische Universität Ilmenau, Fakultät für Wirtschaftswissenschaften, Institut für Wirtschaftsinformatik.
- [15] Kropp, M.; Meier, A.: Swiss Agile Study 2012. Fachhochschulen Zürich und Nordwestschweiz, 2012. Siehe: www.swissagilestudy.ch
- [16] Downey, Scott; Sutherland, Jeff: Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft. hicss, pp. 4870–4878, 46th Hawaii International Conference on System Sciences (HICSS), 2013. www.computer.org/csdl/proceedings/hicss/2013/4892/00/4892e870.pdf
- [17] Harwardt, Mark: Wasserfallmodell versus Scrum – Ist der gute Ruf der agilen Methode gerechtfertigt? Masterarbeit. Fernuniversität Hagen Lehrgebiet Programmiersysteme, 2008. www.fernuni-hagen.de/imperia/md/content/ps/masterarbeit-harwardt.pdf
- [18] Tabelle auf Seite 120 in: Korn, Hans-Peter: Das „agile“ Vorgehen: Neuer Wein in alte Schläuche – oder ein

„Déjà-vu“? In: Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.): Lecture Notes in Informatics (LNI) – Proceedings, Series of the Gesellschaft für Informatik (GI), Volume P-224.

www.korn.ch/archiv/publikationen/↔

Neuer-Wein-in-alte-Schlaeuche-oder-Deja-vu.pdf

[19] Haberl, Peter; Spillner, Andreas; Vosseberg, Karin; Winter, Mario; Umfrage 2011: Softwaretest in der Praxis. dpunkt.verlag; Heidelberg, 2012;

www.softwaretest-umfrage.de/↔

Broschuere_Softwaretest_in_der_Praxis.pdf