ULRICH BODE

DAS TRIPLE A FRAMEWORK

WIE SCRUM MASTER ALS TURBO FÜR DEN AKTIENKURS DURCHSTARTEN

Ulrich Bode

DAS TRIPLE-A FRAMEWORK

Wie Scrum Master als Turbo für den Aktienkurs durchstarten

Das Triple-A Framework
Wie Scrum Master als Turbo für den Aktienkurs durchstarten

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

Verlag: Trochos GmbH, Eichenau, Deutschland
Druck: Libri Plureos GmbH, Friedensallee 273, 22763 Hamburg
Mitarbeit: Barbara Niedner
Porträtfotos: Christoph Vohler

ISBN 978 3 938277 10 2 » eBook
ISBN 978 3 938277 11 9 » Hardcover
ISBN 978 3 938277 12 6 » Paperback

www.3xA.org

Inhaltsverzeichnis

Vorbemerkungen

Zitate aus dem Scrum Guide[1] sind der Version 2020 entnommen.

Die deutsche Fassung des Scrum Guides verwendet für zentrale Begriffe wie Scrum Team, Product Owner, Increment und Scrum Master meist die englische Version. Dem wird in diesem Buch gefolgt. Auch darüber hinaus werden branchenübliche englische Begriffe verwendet.

In Anlehnung an den Scrum Guide wird der Begriff „Developer" stellvertretend für alle umsetzenden Tätigkeiten verwendet.

[1] https://scrumguides.org/

Executive Summary

Die Kunst der Softwareentwicklung ist ein anspruchsvoller **Triathlon** in den drei **Disziplinen** Automatisierung, Agilität und Acting.

Triple-A vereint die drei Disziplinen zu einem kraftvollen Framework, das sowohl in der Softwareentwicklung als auch darüber hinaus wirkmächtig ist.

- **Automatisierung** schafft stabile Abläufe, die eine hochwertige Entwicklung sowie einen reibungslosen Betrieb gewährleisten.
- **Agilität** orientiert sich an den Kundenbedürfnissen und stärkt die Wettbewerbsfähigkeit in etablierten Märkten.
- **Acting** (Agieren) ist technologiegetrieben, disruptiv und erschließt neue Märkte.

Alle drei Disziplinen haben ihr eigenes Framework mit individuellen Mindsets, Methoden und Metriken. Triple-A führt diese erstmalig zu einem neuen, umfassenden **Gesamtframework** zusammen.

Agile Frameworks sind für eine Welt der Jahrhundertwende konzipiert. Teams sollen autark agieren, obwohl heute die Zusammenarbeit mit Dutzenden von Teilzeit-Teammitgliedern notwendig ist. Abhängigkeiten werden vermieden, obwohl sie einen großen Mehrwert bieten. Es ist an der Zeit, die agilen Konzepte weiter voranzutreiben.

Wenn **Agilität** in einer Organisation scheitert, liegt es nicht am „Mindset" oder an dem agilen Framework. Grundlegend für den Erfolg von Agilität ist eine leistungsstarke **Automatisierung** als Fundament. Automatisierung erfordert höchste Disziplin, die rasche Beseitigung von Fehlern und Schwachstellen sowie ein konsequentes Inkasso technischer Schulden.

Acting ist das dritte Framework. Während agile Frameworks kundenorientiert und mit schrittweisen Anpassungen in bestehenden Märkten arbeiten, ist Acting technologiegetrieben und disruptiv auf der Suche nach neuen

Märkten. Der **Descaling**-Ansatz von Acting vereinfacht die Organisation und ihre Arbeitsweise, anstatt agile Praktiken und Methoden auf eine größere, komplexere Ebene zu skalieren.

Seit rund 25 Jahren ist Software in **Umfang und Komplexität** gewachsen, die Abhängigkeiten haben zugenommen und die Zahl der Beteiligten ist gestiegen. Gleichzeitig sind die Anforderungen an Compliance, Sicherheit, Bedienbarkeit und vieles mehr angestiegen. Die Zuständigkeiten der Entwicklungsteams wurden auf den Anwendungsbetrieb ausgeweitet. Scrum Master organisieren um das Kernteam herum die virtuelle Organisation mit einem Heer von Experten und Partnern.

Je mehr die Daten und nicht mehr die Kunden laufen, desto intensiver wird die Zusammenarbeit zwischen den Teams und ihrer Software. Scrum Master wirken als **Netzwerk für die Dynamik** einer Organisation.

Methoden der Softwareentwicklung dienen zunehmend als **Blaupause** für die gesamte Organisationen. Software wird ausschließlich von Menschen gemacht. Die Anforderungen einer dienstleistungsorientierten Gesellschaft spiegeln sich in verdichteter Form in der Softwareentwicklung wider.

Umsetzungsstarke Scrum Master sind die treibenden Führungskräfte des permanenten Wandels auf operativer Ebene. Die Bezeichnung Scrum Master ist jung, ihre Rolle gab es bereits in der Antike und ist als Quartiermeister bekannt.

Scrum Master führen ein Team zum High Performer. Damit beeinflussen sie direkt das wirtschaftliche Ergebnis. **Umsätze und Gewinne, Aktienkurse und Renditen** hängen signifikant von der Leistungsfähigkeit der Softwareentwicklung ab.

High Performer haben ein **50 Prozent höheres Wachstum der Marktkapitalisierung** im Lauf von drei Jahren im Vergleich zu Low Performern.

Das Triple-A Framework befähigt Organisationen, ihre Produktivität zu stärken, Innovationen zu entfachen und Kosten zu senken.

Navigation durch die Kapitel

Onboarding

Kapitel 1: **Gewinne statt Business-Esoterik** analysiert die Ausgangslage und zeigt die Zielsetzung von Triple-A auf.

Kapitel 2: **Die Essenz der Agilität** ist eine Einführung in die agilen Kernelemente und Frameworks. Dabei werden Missverständnisse aufgeklärt.

Framework Automatisierung

Kapitel 3: **Automation First** fokussiert auf menschliche Routinen und technische Automatismen.

Kapitel 4: **Stärke durch Qualität** demonstriert den Nutzen von Qualität für schnelle und kostengünstige Softwareentwicklung.

Framework Agilität

Kapitel 5: **Agilität 2.0** entwickelt agile Methoden im Kontext zunehmender Komplexität weiter.

Kapitel 6: **Erfolgsfaktor Projekt** stellt diese in den Mittelpunkt der Organisationsstruktur.

Kapitel 7: **Der pragmatische Scrum Master** beleuchtet wichtige Aufgaben im Tagesgeschäft und erweitert die Rolle.

Kapitel 8: **Von der Last zur Leistung** wendet die in den vorherigen Kapiteln aufgezeigten Methoden gezielt auf das Management von schwierigen Situationen an.

Framework Acting

Kapitel 9: **Acting Organisation** entfaltet die Prinzipien für das dritte Framework Acting und erweitert Scrum zu Acting Scrum.

Gesamtframework Triple-A

Kapitel 10: **Triple-A** vereinigt alle drei Frameworks und definiert das Gesamtframework.

1 Gewinne statt Business-Esoterik

Dies ist keine Einführung in Scrum. Sie erfahren auch nicht viel über Motivation, Purpose oder Kreativitätstechniken. Sie können diese Themen an anderer Stelle lesen. Hier geht es um Veränderung in einem schwierigen Umfeld, um Scrum Master, die die Realität meistern.

Als Scrum Master gibt Ihnen dieses Buch das nötige Rüstzeug, um Ihr Projekt trotz aller Herausforderungen und Widerständen aktiv zum Erfolg zu führen.

Als Organisation lernen sie, wie Softwareentwicklung die rasant steigende Komplexität in drei Disziplinen professionell meistert und zum Umsatz- und Gewinntreiber wird.

1.1. Agile Wolkenschlösser

Mit dem Agilen Manifest von 2001 wurden Kundennutzen, Flexibilität und Zusammenarbeit in den Fokus gerückt. Weit über die Softwareentwicklung hinaus hat die Idee der **Agilität** Eingang in den Kanon der Geschäftswelt gefunden. Agilität soll alte Hierarchien, Silos und Autoritäten abschaffen. Ohne Angst würden die Teams für eine glückliche und prosperierende Zukunft arbeiten. Selbstorganisiert und selbstverwirklicht erstrahlen die beste IT-Architektur und die unglaublichste Software.

Agilität ist allzu oft nur ein wohlklingendes Buzzword, mit dem das altbekannte Chaos kaschiert wird. Man schmückt sich damit, hat aber den Kern nicht verstanden. Product Owner verstehen sich als Projektleiter alter Schule. Software-Qualität wird weiterhin als lästiges Übel betrachtet. Die Developer sind brave Befehlsempfänger. Scrum Master stehen am Rand und geben den Motivationsguru. Es blüht mehr Business-Esoterik als echte Agilität.

1.2. Führungskraft Scrum Master

Scrum Master sind Führungskräfte für die Organisation. Scrum Master stehen im Zentrum der organisatorischen Transformation. Sie sind die **operativen Schlüsselakteure**, die Teams als auch die gesamte Organisation befähigen, den kontinuierlichen Wandel zu steuern.

Leider springen viele Scrum Master zu kurz. Sie sind häufig als folgende Typen anzutreffen:

- **Animateur**: Versteht sich als Feel-Good-Manager und sorgt für fancy Meetings.
- **Bürokrat**: Erledigt die Scrum-Formalitäten, füllt Scrum aber nicht mit Leben.
- **Click Master**: Moderiert die Meetings und klickt durch die Tickets.
- **Zaungast**: Beobachtet das Team und gibt gelegentlich Tipps.

Scrum Master führen ein Projekt auch in einem herausfordernden und schwierigen Umfeld zum Erfolg. Sie beseitigen Hindernisse (impediments) für den Fortschritt des Scrum Teams. Sie gestalten aktiv und kraftvoll die Prozesse im Team und Organisation.

Dieses Buch vermittelt Scrum Master die erforderlichen Kenntnisse und passgenauen Werkzeuge für ihre Arbeit im Team und in der gesamten Organisation.

1.3. High Performer für Wachstum

Organisationen sind oft näher am Zusammenbruch als an der Perfektion. Pragmatismus und Engagement der Mitarbeiter verhindern ihren Untergang. Diese dunkle Seite einer Organisation ist das Betätigungsfeld der Scrum Master. Sie sind besonders gefragt, wenn es hart auf hart kommt. **Scrum Master führen ein Team zum High Performer**. Damit beeinflussen sie direkt das wirtschaftliche Ergebnis. Umsätze und Gewinne, Aktienkurse und Renditen hängen signifikant von der Leistungsfähigkeit der Softwareentwicklung ab.

Nicole Forsgren, Jez Humble und Gene Kim untersuchten mehrere Unternehmen[2] und stellten fest:

- **High Performer in der Softwarebereitstellung erreichen 46-mal mehr Deployments als Low Performer, sind 440-mal schneller im Prozess von der Auftragsvergabe (commit) bis zur Fertigstellung und haben eine fünffach geringere Fehlerquote bei Änderungen.**
- **High Performer haben ein 50 Prozent höheres Wachstum der Marktkapitalisierung im Lauf von drei Jahren im Vergleich zu Low Performern.**

Wenn ein Unternehmen immer mehr auf eigenentwickelte Software angewiesen ist, kann es durch eine leistungsstarke Softwareentwicklung Umsatz und Gewinn steigern und gleichzeitig die Kosten senken. Gelingt dies nicht, ist das Verschwinden vom Markt unvermeidlich.

1.4. Triple-A: Automation, Agility, Acting

Agile Frameworks sind für eine Welt der Jahrhundertwende konzipiert. Teams sollen autark agieren, obwohl heute die Zusammenarbeit mit Dutzenden von Teilzeit-Teammitgliedern notwendig ist. Abhängigkeiten werden vermieden, obwohl sie einen großen Mehrwert bieten. Es ist an der Zeit die agilen Konzepte weiter voranzutreiben.

Neben Agilität braucht Softwareentwicklung ein starkes Fundament: **Automatisierung**. Durch die Automatisierung der gesamten Pipeline, von der Integration aller Softwaremodule über die Testautomatisierung bis hin zur vollautomatischen Installation, wird eine hohe Qualität und Produktivität erzielt.

Agile Methoden sind kundenzentriert. **Disruptive** Organisationen arbeiten jedoch nicht kundenzentriert, sondern technologiegetrieben, nicht reagierend, sondern agierend. Sie entwickeln keine verbesserten Produkte, sondern völlig neue. **Agieren** (Acting) statt nur reagieren ist die nächste Herausforderung für eine Organisation.

[2] Nicole Forsgren, Jez Humble, Gene Kim: Das Mindset von DevOps. Accelerate 2019

Das **Triple-A-Framework** basiert auf der Automatisierung, einer weiterentwickelten Agilität für die Kundenorientierung und der agierenden (acting) Organisation für die disruptive Schaffung neuer Märkte.

Triple-A verbindet Automatisierung, Agilität und Acting zu einem integrierten Framework für IT-Projekte und mehr.

2 Die Essenz der Agilität

Software bildet eine wesentliche Grundlage für hohe Flexibilität und **Anpassungsfähigkeit**. Software kann viel leichter geändert werden als Hardware. Das bedeutet aber, dass industrielle Strukturen, die für die Entwicklung von Hardware konzipiert wurden, nicht für die Entwicklung von Software angewendet werden sollten. Andernfalls verspielt man einen entscheidenden Vorteil von Software: die Fähigkeit, sich schnell und flexibel an Marktveränderungen anzupassen.

Als Reaktion darauf entstanden um das Jahr 2000 agile Methoden. Sie sind effektiver und effizienter in der Softwareentwicklung als traditionelle Vorgehensweisen. Dieses Kapitel beleuchtet die grundlegenden Elemente der Agilität. Sie bilden den Ausgangspunkt für die weiteren Überlegungen.

2.1. Agile Beschleuniger

Software Engineering (1968) etablierte das Phasenmodell als Fundament der Softwareentwicklung. Agile Konzepte bauen auf dieser Grundlage auf und legen den Schwerpunkt auf kurze Iterationen, die durch eine Reihe ergänzender Elemente verstärkt werden.

2.1.1. Offen für die Zukunft

Wir können nur begrenzt in die Zukunft schauen. Agilität geht davon aus, dass eine vollständige Planung nicht machbar ist und Veränderungen jederzeit möglich sind. Vom preußischen Generalfeldmarschall Graf von Moltke ist der Satz überliefert: *„Kein Plan überlebt die erste Feindberührung."*

Das Agile Manifest formuliert als **vierten Wert:**

„Reagieren auf Veränderung mehr als das Befolgen eines Plans"

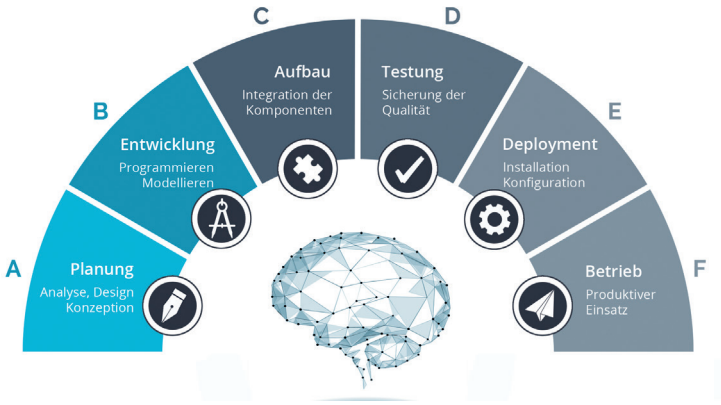
Und weiter als **2. Prinzip:**

„Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden."

Agilität bietet Organisationen einen strukturierten Rahmen, um **neue Anforderungen** jederzeit professionell zu meistern. Anpassungsfähigkeit ist das Können, um auf Veränderungen zu reagieren. Agile Organisationen blicken daher offen in die Zukunft.

2.1.2. Phasen

Jede Entwicklung folgt dem **Phasenmodell**, auch Wasserfallmodell genannt. Natürlich sind Fachwissen und Anforderungen Voraussetzungen, bevor die Entwicklung beginnen kann.



Phasenmodell: Nach der Planung folgt die Entwicklung, dann die Zusammenführung und Integration der Komponenten (Build), anschließend die Qualitätssicherung (Testung), zuletzt die Verteilung (Deployment) und der Betrieb (Operation).

Die Softwareentwicklung durchläuft in der Regel **mehrere Phasen** von der ersten Idee bis zum fertigen Produkt. Die Planungsphase wird bestimmt durch die Anforderungsanalyse, in der die Bedürfnisse des Kunden ermittelt werden, und die Konzepterstellung.

In der **Entwicklungsphase** findet die eigentliche Programmierung statt. Der Zusammenbau (Build-Phase) schließt sich in der Regel direkt daran an. In dieser Phase werden die einzelnen Module zusammengeführt und der gesamte Quellcode in eine ausführbare Datei übersetzt, beispielsweise eine exe-Datei bei Windows-Anwendungen. Es folgt eine intensive Testphase, um sicherzustellen, dass die Software fehlerfrei funktioniert. Schließlich wird die Software installiert und in Betrieb genommen.

Dieser Prozess kann je nach Komplexität des Projekts und der gewählten Methodik variieren. Reibungslose Übergänge zwischen den Phasen erfordern strukturierte und gut koordinierte **Übergaben** (Handover).

2.1.3. Iteration

Einer Anekdote zufolge soll Albert Einstein seinen Studenten die gleiche Abschlussprüfung wie im Vorjahr gestellt haben. Ein Assistent wies ihn darauf hin. Die Antwort: *„Es sind die gleichen Fragen, aber die Antworten haben sich geändert."*

Der ständige Wandel zwingt Organisationen, sich immer wieder anzupassen. Deshalb arbeiten wir in der Softwareentwicklung iterativ und reagieren in jeder Iteration auf die Veränderungen. Das war schon im sogenannten **Wasserfallmodell** so. Der Hauptunterschied zu heute bestand in den deutlich längeren Zeitabständen zwischen zwei Iterationen. In der Vergangenheit war die Automatisierung in der Softwareentwicklung weit weniger verbreitet. Dies erforderte langwierige manuelle Test- und Stabilisierungsphasen, die nur vierteljährliche oder halbjährliche Releases zuließen.

Heute arbeiten die Teams dank hoher Automatisierung in Iterationen von wenigen Wochen. Im agilen Framework Scrum wird eine **Iteration als Sprint** bezeichnet. Neben der schnelleren Anpassung ermöglicht die kurze Taktung auch ein früheres Feedback, sodass notwendige Korrekturen schneller erkannt werden.

In agilen Organisationen werden nicht nur Produkte iterativ entwickelt, um so früh wie möglich Wert zu schaffen, sondern auch die Organisation entwickelt sich mit jeder Iteration weiter. Entwicklung folgt generell einem

iterativen Muster, sei es in der Natur (Evolution), in der Kunst oder in der Technik.

Jede von Menschen gestaltete Iteration umfasst Planung, Umsetzung und Lieferung. Dies gilt auch für agile Vorgehensweisen. Die Bildung eines Gegensatzes von traditionellem Wasserfallmodell und moderner Agilität ist nicht zielführend. Der entscheidende Unterschied zwischen der Softwareentwicklung früher und heute ist die wesentlich kürzere Zeitspanne für eine Iteration.

Der immer schneller werdende **Wettbewerb** erfordert kürzere Iterationen, und dank der Automatisierung sind sie auch möglich. Während früher eine Iteration mehrere Monate dauerte, sind es heute nur noch wenige Wochen. Die viel kürzeren Iterationen ermöglichen eine bessere Anpassungsfähigkeit. **Dies ist der Schlüssel zur Agilität**.

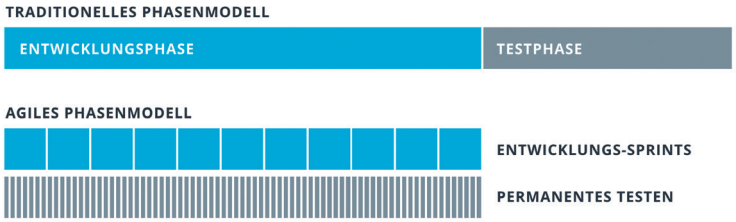
2.1.4. Zyklus

Der **Software-Zyklus** umfasst alle Phasen einer Iteration von der Planung bis zum Betrieb. Der Software-**Lebenszyklus** erstreckt sich über den gesamten Lebenszyklus der Software von der ersten Planung bis zur Außerbetriebnahme. Die Weiterentwicklung wird klassischerweise als Wartung bezeichnet. Der Entwicklungszyklus wurde bei manchen Projekten nur einmal durchlaufen, daher war ein Unterschied zwischen Zyklus und Lebenszyklus kaum erkennbar.

Heute wird Software viel intensiver und in kürzeren Abständen an Marktveränderungen angepasst. Während in der traditionellen Softwareentwicklung eine Iteration mehrere Monate dauert, arbeiten agile Teams typischerweise mit einer zweiwöchigen Iteration. Die Software befindet sich in einer kontinuierlichen Weiterentwicklung (Continuous Development).

Leistungsstarke Teams führen die Qualitätssicherung nicht erst am Ende einer Iteration durch, sondern bei jeder Änderung. Sobald ein Developer eine Änderung am Code freigibt, dem sogenannten **Commit**, wird eine Testautomatisierung ausgeführt, um das gesamte Softwaresystem zu überprüfen. Ein Testdurchlauf dauert typischerweise rund 15-60 Minuten.

Dies stellt sicher, dass die Änderung korrekt funktioniert und keine unbeabsichtigten Nebeneffekte verursacht.



In der traditionellen Arbeitsweise folgt auf eine lange Entwicklungsphase die Testphase. Im agilen Prozessmodell wird die Entwicklung in kurze Iterationen (Sprints) unterteilt und fehlerfrei ausgeliefert. High Performer führen Tests nach jeder Änderung (Commit) durch.

Ein zentraler Treiber zur Beschleunigung liegt in der Automatisierung der Phasen. Die Abfolge der Phasen wird üblicherweise als Pipeline bezeichnet. Heute können die Phasen Aufbau, Testung, Verteilung und Betrieb vollständig automatisiert werden. Statt mehrerer Wochen oder gar Monate beanspruchen diese Phasen nun nur noch etwa eine Stunde. Dadurch verkürzt sich die Dauer der einzelnen Iterationen drastisch. Dies führt zu einer enormen Beschleunigung, einem **Boost für den Time-to-Market.**

Automatisierung beschleunigt die Phasen drastisch.

Innerhalb einer zweiwöchigen Iteration werden die **Phasen von der Entwicklung bis zum Test** wiederholt durchlaufen. Falls erforderlich, werden Änderungen sofort in der Produktion bereitgestellt, sodass ein Go-Live während eines Sprints häufiger erfolgen kann.

Wer Tausende von Änderungen sammelt und erst dann mit dem Testen beginnt, **schafft Komplexität**. Wer direkt nach jeder Änderung testet, zerlegt die Arbeit in eine Vielzahl **einfacher Aufgaben.**

Zur Erinnerung sei noch einmal die Studie von Nicole Forsgren, Jez Humble und Gene Kim zitiert: *„High Performer im Bereich der Softwarebereitstellung schaffen 46-mal mehr Code-Fertigstellungen als Low Performer, sind 440-mal schneller im Prozess von der Auftragserteilung bis zur Fertigstellung."*

Autor

Der Autor Ulrich Bode ist Diplom-Informatiker und seit vielen Jahren als selbständiger Berater in Branchen wie Automotive, Handel, Finanzdienstleister und öffentliche Verwaltung tätig. Er wurde 2003 zum Fellow der Gesellschaft für Informatik e.V. ernannt.

www.ulrich-bode.de

Ulrich Bode ist auch Autor folgender Bücher:

Die Informationsrevolution (1997)

Das Werk liefert einen fundierten Überblick über die Grundlagen der Informationsgesellschaft, ihre Auswirkungen auf die Unternehmensorganisationen und die Gesellschaft sowie der Prognose des Smartphones.

The Making of Digital (2018) www.themakingof.digital

Die Menschheit erschafft sich eine neue, digitalisierte Welt. Das Buch bietet einen kompakten Überblick, zeigt wegweisende Trends auf und erläutert die zentralen Herausforderungen.

Sozial 4.0 statt Hartz IV (2021) www.www.s4h4.de

„Sozial 4.0" ist ein grundlegendes Reformkonzept, das sich am Leitbild eines Grundeinkommens orientiert und mehr als überfällige Erneuerungen staatlicher Strukturen umfasst. Dank digitaler Transformation vereinfachen wir das Grundeinkommen nicht nur radikal, sondern rechnen das optimale System für ein Grundeinkommen einfach aus.

Masterplan Digitales Bayern (2023) www.masterplan.bayern

Warum die Digitalisierung der öffentlichen Verwaltung so schwierig ist und wie sie trotzdem gelingen kann.

ULRICH BODE

DAS TRIPLE-A FRAMEWORK

Wie Scrum Master als Turbo für den Aktienkurs durchstarten

Aufbauend auf den klassischen Säulen des IT-Projektmanagements, allen voran dem Agilen Manifest von 2001 und Software Engineering von 1968, stellt das Triple-A Framework einen weiteren Meilenstein in dessen Entwicklung dar.

Software-Anforderungen sind seit der Jahrhundertwende rapide gewachsen und damit auch der Umfang und die Komplexität ihrer Entwicklung. Scrum Master werden zur treibenden Kraft hinter prosperierenden Marktführern und disruptiven Start-ups. Ob Sie eine erfolgreiche Führungskraft oder ein ambitionierter Neuling sind – dieses Buch ist Ihr Schlüssel zu Spitzenleistungen. Mit dem Triple-A Framework steigern Sie die Produktivität, akzelerieren Innovationen und senken die Kosten.

Die Softwareentwicklung gleicht heute einem Triathlon, der Kompetenzen in drei Disziplinen erfordert. Das Triple-A Framework vereint diese Elemente und bietet eine einheitliche und effektive Strategie für moderne IT-Landschaften:

Automation: Beherrschung solider Prozesse, die Qualität und Produktivität auf ein neues Niveau heben.
Agility: Entwicklung kundenorientierter Produktideen, um sich auf wechselnde Marktanforderungen einzustellen.
Acting: Beschleunigung disruptiver Innovationen, um technologiegetrieben neue Märkte zu erschließen.

Jede Disziplin erfordert eine spezifische Kombination von Mindset, Methoden und Metriken. Triple-A vereint diese zu einem innovativen, ganzheitlichen Ansatz, der Unternehmen zu neuen Leistungsniveaus antreibt. Er befähigt Softwareteams, in einer zunehmend komplexeren Welt erfolgreich zu sein und die Agilitätskrise in eine Chance für den Fortschritt zu verwandeln. Herausforderungen werden als Katalysatoren für bahnbrechende Innovationen, dynamisches Wachstum und nachhaltigen Erfolg neu definiert.

ISBN 978 3 938277 12 6

9 783938 277126