



## Beende den Spuk!

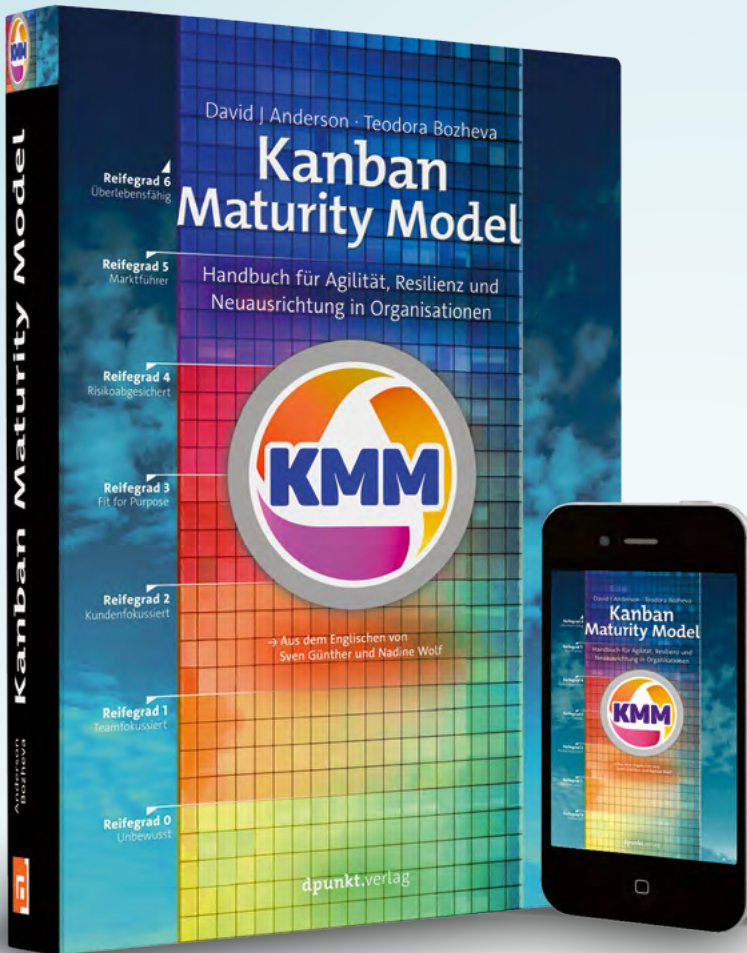
Velocity-Betrug

A Fool with a Tool ...

Lokale Optimierung

8 Fallstricke





# Leistungsstarkes Werkzeug für Kanban-Initiativen

David J Anderson · Teodora Bozheva

## Kanban Maturity Model

Handbuch für Agilität, Resilienz und  
Neuorientierung in Organisationen

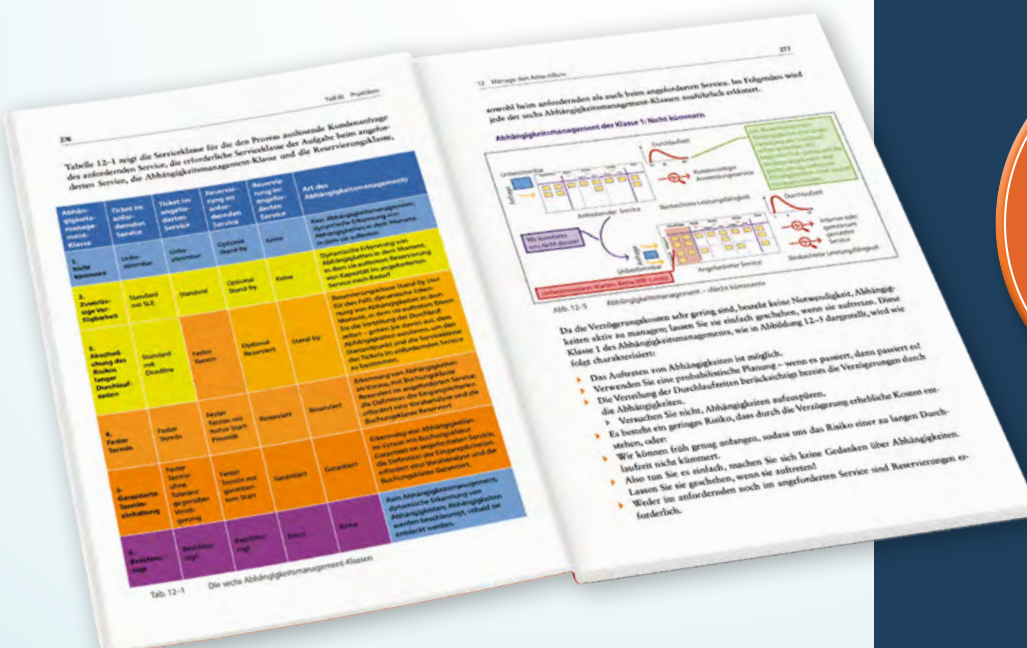
Aus dem Englischen von Sven Günther  
und Nadine Wolf

2021 · 648 Seiten · € 59,90 (D)  
komplett in Farbe · Festeinband  
ISBN 978-3-86490-608-4

- Roadmap zur schrittweisen kontinuierlichen Prozessoptimierung
- Spezifische Praktiken, um Organisationen beweglicher und anpassungsfähiger zu machen
- Pragmatisches Vorgehen, angelehnt an bekannte Reifegradmodelle wie CMMI®

»The Kanban Maturity Model helps you to understand where you are, shows what is possible and how to get there. It does so by explaining the observed behavior for each level and by what practices such behavior is encouraged. The Kanban Maturity Model has the potential to be the Rosetta Stone for organizational change with Kanban.«

Wolfgang Wiedenroth, Accredited Kanban Trainer and Coach, it-agile GmbH



**Bundle up!**  
Print & E-Book  
[www.dpunkt.de](http://www.dpunkt.de)



Liebe Leser:innen,

immer wieder diskutiere ich mit meinen Kolleg:innen über gruselige Verwendung von agilen Praktiken. Dieses Jahr habe ich es geschafft, davon ein paar Beispiele zusammen zu tragen – ganz im Geiste einer Halloween-Ausgabe.

In „Die Neuentwicklung und der Velocity-Betrug“ berichte ich von gut gemeinten Überstunden und dem Schaden, den zuviel Fokus auf Velocity anrichten kann (Seite 6). Mein Kollege Markus Gärtner wirft in „A Fool with a Tool ...“ einen Blick auf das vielfach zu beobachtende Nachahmen agiler Praktiken, ohne deren Sinn zu verstehen (Seite 10). Mit „Reise durch Europa“ beschreibe ich eine Organisation, die durch Incentivierung einer Abteilung negative Effekte in einer anderen Abteilung ausgelöst hat (Seite 16). In „Ceterum Censeo ...“ erzählt Markus Gärtner die Geschichte, wie Veränderungen Unsicherheit erzeugen und Beharrlichkeit brauchen (Seite 20). Und Ralf Lethmate zeigt in seinem Text „8 Fallstricke“ typische Problemfelder, die bei Veränderungsprozessen in Organisationen auftreten (Seite 36).

Ulrich Rücker untersucht in „Bin ich eine Rolle und wenn ja, wie viele?“ (ab Seite 28) welchen Nutzen das Rollen-Konzept hat und wie wir es heutzutage in der Arbeitswelt anwenden können. Und mit einem technischen Schwerpunkt schreiben meine Kollegen Andreas Havenstein und Steven Collins ihren Erfahrungsbericht „Von der eigenen Maschine in die Cloud“ (Seite 56).

Die Buchtipps (Seite 50) widmen sich den Werken „Team Topologies - Organizing Business and Technology Teams for Fast Flow“, „Radical Focus - Achieving Your Most Important Goals with Objectives and Key Results“ und „Kanban Maturity Model: Handbuch für Agilität, Resilienz und Neuausrichtung in Organisationen“. Die „Agile Führungsetage“ findet sich auf Seite 26 mit den Aufforderungen: „Schluss mit Tolerieren“ und „Sich Unangenehmem stellen“. In unserer Rubrik „agil kurz und knapp“ (ab Seite 43) sind diesmal die Themen „Empirische Prozesskontrolle“, „Delivering Fluency“ und „Product Backlog Refinement“.

Ich hoffe, es schaudert Sie schön, beim Lesen unserer gruseligen Beispiele und Sie können darüber vielleicht den einen oder anderen Spuk bei sich früher beenden. Ich freue mich über Ihr Feedback!

Wolf-Gideon Bleek, Chefredakteur der agile review

Treten Sie gern mit mir in Verbindung:  
wgb@it-agile.de

# INHALT

6

## Die Neuentwicklung und der Velocity-Betrug

Wolf-Gideon Bleek erzählt von Schaden durch gut gemeintes Team-Verhalten



Markus Gärtner zeigt auf, wie wichtig es ist, dass wir den Sinn hinter den agilen Praktiken verstehen, um sie im Alltag umsetzen zu können. Und er beschreibt die Auswirkungen, die entstehen, wenn wir darauf nicht achten.

10

## A fool with a tool is still a fool

Wie Cargo Cult die Wirkung agiler Praktiken zunichte macht



Veränderungsaktivitäten lösen nicht immer direkt Verbesserungen aus. Markus Gärtner argumentiert, dass es in dieser kritischen Phase wichtig ist, beharrlich zu bleiben, damit erfolgreiche Praktiken in den Alltag übernommen werden.

16

## Reise durch Europa

Wolf-Gideon Bleek berichtet über lokale Optimierung und falsche Incentivierung

In der Rubrik „agile Führungsetage“ fordert Henning Wolf zum Schluss mit Tolerieren auf und Nadine Wolf ermutigt uns, sich Unangenehmem zu stellen.

20

## Ceterum censeo ...

Was Veränderungsprozesse von 2000 Jahre alten Römern lernen können

26

## agile Führungsetage

Schluss mit Tolerieren - Sich Unangenehmem stellen



Ulrich Rücker schaut genauer auf die Rollenkonzepte, die in vielen Methoden Einzug gefunden haben. Insbesondere geht er auf die Probleme ein, die damit in unsere Arbeitswelt kommen.

**Bin ich eine Rolle und wenn ja, wie viele?**  
Rollenbasierte Konzepte sind aktuell überall zu sehen, aber was ist eine Rolle?

28



**8 Fallstricke**  
Die wesentlichen Fallstricke aus dem Buch von Dona Meadows „Thinking in Systems“

36

Empirische Prozesskontrolle  
Delivering Fluency  
Product Backlog Refinement

**kurz und knapp**  
Unsere Kolleg:innen beschreiben Grundlagen der Agilität

43



**Buchtipps**  
Unsere Einschätzung zu „Team Topologies“, „Radical Focus“ und „Kanban Maturity Model: Handbuch für Agilität, Resilienz und Neuausrichtung in Organisationen“

50



**Von der eigenen Maschine in die Cloud**  
Andreas Havenstein und Steven Collins liefern einen Erfahrungsbericht

56

**Impressum** 14



# Die Neuentwicklung und der Velocity-Betrug

Wie bitte? Mein Team ist gar nicht so schnell, wie ich dachte und das Budget ist schon aufgebraucht?

Von Wolf-Gideon Bleek

## Die Vorgeschichte

In diesem Projekt hat ein großes Unternehmen den Schritt gewagt, von einer klassischen Organisation zu einer agilen zu gehen, um das Kernprodukt von Grund auf neu zu entwickeln.

Das Projektmanagement war gewohnt Teams zu monitoren, um dem Management regelmäßig einen Status zu geben.

Die agilen Teams haben in diesem Projekt regelmäßig ihr Backlog gezeigt und die Velocity berichtet. Mithilfe des Durchschnitts ihrer Velocity aus den letzten Sprints konnte jedes Team abschätzen, ob es alle wichtigen Features bis zum Go-Live umsetzt und darüber berichten.

Die ehemaligen Projektmanager:innen sind umgeschult worden, um nun als Scrum-Master:innen die Teams anzuleiten, agil zu arbeiten. Außerdem haben

sie die Schnittstelle zwischen den Teams und dem Projektmanagement hergestellt.

Alle Teams wurden mit externen Mitarbeiter:innen verstärkt, für die ein Budget bereitgestellt wurde, das bis zum Go-Live ausreichte.

## Das Maß „Velocity“

In agilen Vorgehensmodellen wurden bereits früh feste Zeitblöcke für die Strukturierung von Arbeit bevorzugt. In Scrum kennen wir z. B. die immer gleich langen Sprints. Um die Leistungsfähigkeit des Teams zu bestimmen, wurde die Anzahl der fertiggestellten Storys oder die Summe der Storypoints aller Storys pro Zeitblock herangezogen. Diese Zahl wird Velocity (Team-Geschwindigkeit) genannt.

Die Velocity hat sich in den frühen 2000ern im Rahmen der extreme Programming (XP) Bewegung

# Mit deinem Latein am Ende?



***Damit dein Team abliefert –  
it-agile zeigt wie.***

direkt zum Angebot





## A fool with a tool is still a fool

Von Markus Gärtner

Als Jugendlicher habe ich mich eine Zeit in meinen Hobbies mit Tischtennis angefreundet. Meine Spielweise war häufig durchwachsen, so dass ich nie ein Spieler auf einem höher besetzten Turnier geworden bin. In meinem Ehrgeiz ergriff mich damals aber folgende Idee: „Mein Schläger ist nicht gut genug. Mit einem besseren Tischtennisschläger könnte ich da oben mitspielen.“ Also sparte ich Geld, bis ich mir einen besonders guten Tischtennisschläger erlauben konnte, ging in den nächsten Sportmaterialladen und ließ mich vom Fachpersonal beraten, bevor ich mir ein neues Holz und zwei neue Beläge für den Tischtennisschläger meiner Träume zusammenstellte. Eine Änderung an meinem Erfolg – oder Misserfolg – als Tischtennisprofi kam durch diesen Kauf jedoch nicht zustande. Einige Jahre später im Arbeitsleben treffe ich immer wieder auf dieses Muster in der Arbeitswelt und wundere mich, ob ich der einzige Jugendliche war, der so eine Lektion glaubte gelernt zu haben.



Team-Scale Anarcho-Syndicalism; Brian Marick; 2009;  
<http://arxta.net/>  
[Weinberg1985] The Secrets of Consulting, A Guide  
to Getting and Giving Advice Successfully; Gerald M.  
Weinberg; Dorset House Publishing; 978-0-932633-  
01-9

Treten Sie gern mit den Autoren in Verbindung:  
[markus.gaertner@it-agile.de](mailto:markus.gaertner@it-agile.de)



MARKUS  
GÄRTNER

Markus Gärtner arbeitet als Organizational Design Consultant, Certified Scrum Trainer (CST) und Agile Coach für it-agile GmbH, Hamburg. Markus schrieb unter anderem ATDD by Example – A Practical Guide to Acceptance Test-Driven Development, erhielt den Most Influential Agile Testing Professional Person Award in 2013 und steuert zur Softwarekammer, der deutschen Software Craftsmanship Bewegung, bei. Er bloggt auf Englisch unter <http://www.shino.de/blog>.

## IMPRESSUM

Chefredaktion (verantwortlich):  
Dr. Wolf-Gideon Bleek

Redaktionsadresse:  
it-agile GmbH, Willy-Brandt-Straße 1,  
20457 Hamburg  
[www.it-agile.de](http://www.it-agile.de), [info@it-agile.de](mailto:info@it-agile.de)

Gerichtsstand und Erfüllungsort: Hamburg

Layout: Alex Lötsch Design  
[www.loetsch-design.de](http://www.loetsch-design.de)

Gestaltung des Titelfotos: Wolf-Gideon Bleek

Autoren: Wolf-Gideon Bleek, Laura Austermann,  
Stefan Rook, Nadine Wolf, Henning Wolf, Stefan  
Zumbrägel, Michael Grenz, Sven Günther, Ralf  
Lethmate, Claudia Reitenbach

Druck: Drucktechnik Altona  
Große Rainstraße 87, 22765 Hamburg  
[www.drucktechnik-altona.de](http://www.drucktechnik-altona.de)

Bildnachweise: [stock.adobe.com](http://stock.adobe.com)

Alle anderen Fotos und Grafiken: it-agile GmbH

Digitales Abo, digitale Einzelausgabe und einzelne  
Artikel gibt es unter: <https://to.it-agile.eu/digital>

Gedrucktes Abo, Nachbestellung von Einzelausga-  
ben gibt es unter: <https://to.it-agile.eu/gedruckt>

**Alles verstanden? Kinderleicht?**



direkt zum Angebot



**Wo setzen Sie jetzt an,  
um wirkungsvoll agil zu arbeiten?**

**Training**  
**Agile Fluency**  
**Coaching**  
**User Storys**

**Business Storys**

**Workshops**

**Beratung**  
**Ergebnisse** **Wirkung**  
**Assessment**  
**Team-Coaching**

Verlassen Sie sich auf unsere Erfahrung, rufen Sie uns für ein unverbindliches Gespräch an:

☎ **0800 / IT AGILE**  
0 8 0 0 / 4 8 2 4 4 5 3

Anpacken. Begeistern.





# Reise durch Europa

## Unser Vertrieb hat WAS gemacht?

Von Wolf-Gideon Bleek

### Lokale Optimierung

Der Kontext, in dem ich handel, bestimmt meine Handlungsoptionen. Organisationseinheiten wie Abteilungen setzen den Kontext. Die dadurch gezogenen Grenzen in einer Organisation sind demnach wesentlich für das Handeln. Innerhalb der Abteilungsgrenzen wird das Handeln optimiert. Wirkungen, die außerhalb entstehen, werden ausgeblendet.

Fachabteilungen wie z. B. Marketing, Vertrieb etc.

bündeln nicht nur Fachwissen. Sie führen zu lokaler Optimierung. Der Vertrieb optimiert dann auf sein Ziel ohne Rücksicht auf das gute Funktionieren einer anderen Abteilung zu nehmen („jeder für sich selbst“).

In diesem Beispiel wird gezeigt, wie die Zielsetzung für eine Vertriebsabteilung zu einer lokalen Optimierung führte, die im Ergebnis Schäden in der Software-Entwicklungsabteilung verursachte.

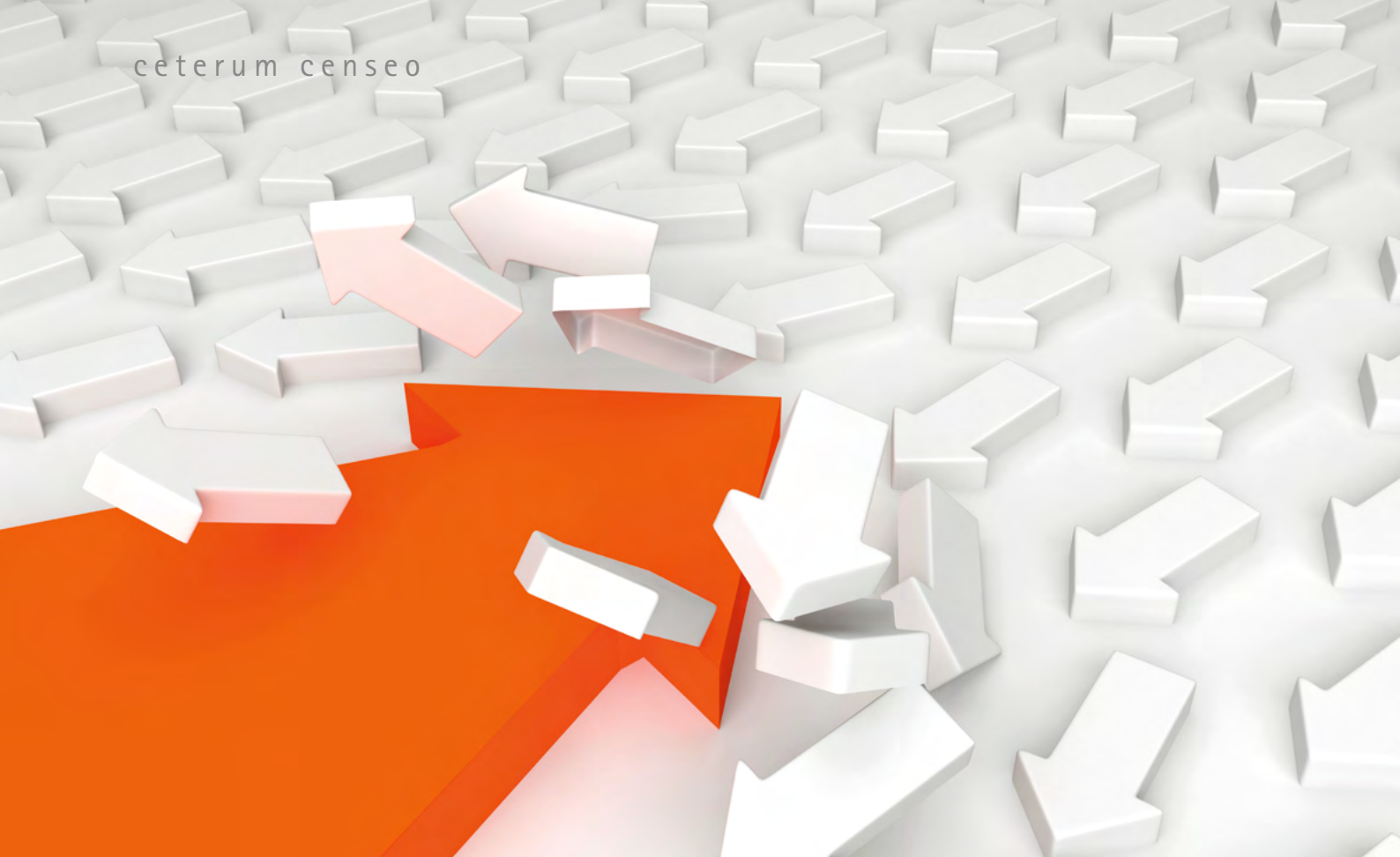
# Certified Agile Leadership



## Aus dem Inhalt:

- *Agile Leadership: Welche Interventionen stehen agilen Führungskräften zur Verfügung?*
- *Agile Organisation: Welche Organisationsformen sind für agile Entwicklung geeignet?*
- *Agiler Change: Mit welchen Mitteln kann eine Organisation agiler werden?*

*Mehr Infos und Termine unter [www.it-agile.de/cal](http://www.it-agile.de/cal)*



## Ceterum censeo ...

### Was Veränderungsprozesse von 2000 Jahre alten Römern lernen können

Hätten im März 2020 die Konzernleitungen so mancher Großkonzerne das Management um die ergriffenen Hygiene-Maßnahmen übernommen, wäre vermutlich nach einer 1-stündigen Fernsehansprache der Bundespräsidentin die Order erlassen worden, dass ab sofort alle zu Hause bleiben, Kurzarbeit machen, falls sie keinen wesentlichen Beitrag leisten und in einem Nebensatz wäre erwähnt worden, dass das alles auf Basis einer neuen Krankheit geschieht, die gerade grassiert. Die Umstände seien aber viel zu kompliziert zu erläutern, also habe man:frau sich den Versuch geschenkt. Natürlich kann ich knapp 1,5 Jahre später nur spekulieren, wie die Gesamtbevölkerung damals reagiert hätte – insgeheim bin ich aber froh, dass die Verantwortlichen in Zusammenarbeit mit den Medienanstalten regelmäßig Transparenz über die derzeitige Situation und deren Einschätzung geschaffen haben, obwohl ich der Berichterstattung irgendwann fast schon überdrüssig geworden bin. Worin besteht der Unterschied zwischen erfolgreichen Verhaltensänderungen, wie ein Großteil der Bevölkerung sie gerade durchlebt hat, und dem, was wir bei Veränderungen rund um agile Produktentwicklung häufig wiederfinden?

Von Markus Gärtner



**Reporting  
zum obersten Chef**

**Schnelle pragmatische  
Lösungen für die  
Fachabteilung**

**Die Entwickler wollen  
mehr Selbstorganisation**

**Der Betrieb soll laufen**



**Wie das gehen soll?  
Antworten, Ideen, Anregungen unter**

**<http://www.it-agile.de>**

# Schluss mit Tolerieren

Von Henning Wolf

Ich beobachte in fast allen Organisationen – in agilen wie klassischen – ein gemeinsames Problem auf der Leadership-Ebene: Es wird zu viel toleriert. Man gibt sich Regeln, beschreibt Rollen und ihre Aufgaben, trifft individuelle Vereinbarungen und wenn davon abgewichen wird, dann passiert meist gar nichts. Und ich gebe zu: Ich war viele Jahre genau so ein Leader, der meistens toleriert hat.



# Sich Unangenehmem stellen

Von Nadine Wolf

Zu The Responsibility Process® gehört das Werkzeug der „Drei Schlüssel zu Verantwortung“. Einer davon ist SICH STELLEN. Mich hat dieser Schlüssel zunächst abgeschreckt, weil er so konfrontativ klang und ich mir unsicher war, ob mir die Anwendung gelingt. Heute weiß ich: Es lohnt sich, SICH zu STELLEN, und der SICH-STEELLEN-Muskel wächst schnell und stetig, wenn wir nur anfangen, ihn bewusst zu benutzen. Nadine Wolf



## Mehr lernen über The Responsibility Process®

- ▶ Kostenloser Einstieg über „Mini-Kurs Responsibility“

- ▶ Online-Workshop „Führen und Coachen mit The Responsibility Process®“
- ▶ 20-Wochen-Online-Programm „Intensivkurs Responsibility“

Mehr Info unter [www.selbstfuehren.de](http://www.selbstfuehren.de)



## Bin ich eine Rolle und wenn ja, wie viele?

Rollenbasierte Konzepte sind aktuell überall zu sehen, aber was ist eine Rolle? Schauen wir auf Methoden im Bereich Lean/Agile wie bspw. Scrum, gibt es den Scrum Master und den Product Owner (PO). Schauen wir auf SAFe wimmelt es nur so von Rollen (selbst wenn es komplizierter scheint als es ist, denn die Rollen über die Ebenen hinweg ähneln sich stark). Und in Methoden im Bereich Teal wie bspw. Holacracy und S3 bilden Rollen das wesentliche Element, entweder vordefiniert (bspw. Circle Lead oder Repräsentant) oder eben durch die Mitglieder selbst definiert.

Von Ulrich Rücker



# *Erfolgslieferanten*

**Agile  
Organisations-  
entwicklung**

**Scrum und  
Kanban  
einführen**

**Coaching,  
Scrum Master,  
Product Owner,  
Schulungen**

**Agile  
Entwicklungspraktiken**

**Schulungen  
und technisches  
Coaching**



## 8 Fallstricke in Veränderungsvorhaben

Bei Veränderungsvorhaben finde ich persönlich die Momente am spannendsten, wo es nicht rund läuft, nicht vorangeht oder unerwünschte Begleiteffekte entstehen. Wir wissen, dass es bei Change keine Blaupause gibt. Aber was für Hilfestellungen kann man geben? Hierfür hilft der Blick von oben auf das Gesamtsystem. Donella Meadows skizziert in ihrem Buch „Thinking in Systems. A Primer“ 8 Fallstricke, über die man bei der Arbeit an komplexen dynamischen Systemen stolpern kann, die hier mit Anwendungsbeispielen in Unternehmen ergänzt werden.

Von Ralf Lethmate



### Falle 1: Widerstand gegen bestehende Regeln – Lösungen, die keine sind

1969 beschloss die rumänische Regierung die Bevölkerungszahlen anzuheben, weswegen Abtreibungen und Verhütungsmittel für unter 45-jährige Frauen verboten wurden. Kurz darauf verdreifachte sich wunschgemäß die Geburtenrate. Dann aber setzte der Widerstand der rumänischen Bevölkerung ein. Obwohl die Abtreibungen und Verhütungsmittel weiterhin illegal blieben, ging die Geburtenrate nahezu bis zu ihrem vorherigen Wert zurück. Was war geschehen? Es gibt dafür zwei tragische Gründe.

Zum einen verdreifachte sich die Müttersterblichkeit durch gefährliche, illegal unternommene Abtreibungen. Zum anderen wurden viele der ausgetra-

genen Kinder an Waisenhäuser übergeben, da sich die Familien aufgrund ihrer Armut nicht im Stande sahen, diese großzuziehen. Die rumänische Bevölkerung zahlte für diesen Widerstand einen hohen Preis: das Leben der Mütter und das Schicksal einer entwurzelten Folgegeneration.

Die Falle besteht darin, einen gewünschten Systemzustand herbeiführen zu wollen, der auf einen natürlichen Widerstand stößt. Insbesondere wenn die Maßnahmen wirkungsvoll sind, kann dies zu noch mehr Widerstand und zu Ergebnissen sowie Nachwirkungen führen, die niemandem gefallen können. Nicht nur, dass das Ergebnis niemandem gefällt, jede:r Akteur:in muss zudem beträchtlichen Einsatz für seine Aufrechterhaltung aufwenden. ►

### Der Ausweg

Loslassen. Das System laufen lassen und die Energie darauf verwenden, ein für alle Akteur:innen zufriedenstellendes Ziel zu formulieren, das einen Anreiz für alle schafft. Eine Alternative ist, eine Neudefinition von größeren und wichtigeren Zielen zu finden, auf die alle gemeinsam hinarbeiten können.

### Beispiel aus dem Unternehmenskontext

In den meisten Unternehmen gibt es „kurze Dienstwege“, weil sie die langsamen, offiziellen Prozesse abkürzen. Das kann soweit gehen, dass sich neue Mitarbeiter:innen darüber beschwerten, mit ihrer Arbeit nicht voranzukommen, weil sie diese direkten Kontakte ins Unternehmen noch nicht aufgebaut haben. Absurd oder?

Von Seiten des Managements wird dagegen erheblicher Aufwand betrieben, die offiziellen Regeln einzufordern, ohne sich aber der Wurzel des Problems anzunehmen. Im Ergebnis gibt es dann zwei Prozesslandschaften, die langsame offizielle und die schnelleren Schattenprozesse. Je besser die schnellen inoffiziellen laufen, desto langsamer werden die offiziellen. Das ist Impulserhaltung.

Für eine systemische Lösung wäre zu eruieren, warum die offiziellen Prozesse ursächlich so lange dauern. Hierzu helfen eine Analyse und kritischer Blick auf die Delivery Journeys z. B. unter Berücksichtigung der 8 Verschwendungen der Software-Entwicklung [Emmons2015].

### Falle 2: Tragik der Allmende

Es waren einmal jene Hirten, die jeder so viele Schafe hüteten, wie sie wollten. Weideland war Allmeindgut. Ein einziges weiteres Schaf stellte kein Problem dar; wenn jedoch jeder einzelne ein weiteres hütete schon. Diese Effekte exponentiellen Wachstums kennen wir alle sehr gut aus der Corona-Krise. Demonstrationen gegen das Masketragen sind ok, solange mit Maske demonstriert wird. Ohne Maske ist es ein Eingriff in die Allmende<sup>1</sup> der Gesundheit aller.

Wenn es eine gemeinschaftlich genutzte Ressource

gibt, profitieren alle, wir teilen aber die Kosten des Missbrauchs mit allen anderen. In solchen Fällen liegt das Problem in der zu schwachen oder zeitverzögerten Rückkopplung vom Zustand der Ressource auf die Entscheidungen der Ressourcennutzer:innen. Die Folge ist Raubbau, bis sie für niemanden mehr verfügbar ist.

### Der Ausweg

Ein möglicher Ausweg ist Bildung: die Nutzer:innen darin schulen und darin ermahnen, welche Konsequenzen die Ausbeutung geteilter Ressource bedeutet. In jedem Fall sollte eine direkte Rückkopplung geschaffen werden, so dass jede Person die Konsequenzen ihres Missbrauchs direkt spürt.

Eine weitere Lösung besteht in der Privatisierung der Ressourcen. Im eigenen Hause spuckt niemand den Kaugummi aus oder lässt den Köttel seines Hundes liegen. Eine dritte Lösung ist, den Zugriff auf die Ressource von öffentlicher Hand zu regulieren.

### Beispiel aus dem Unternehmenskontext

In Unternehmen gibt es mindestens drei beschränkte Ressourcen: Zeit, Budget und Arbeitskraft. Ein häufiger Fall ist, dass der Vertrieb schneller verkauft (und verspricht), als die Entwicklung entwickeln kann. Dies führt zur körperlichen Ausbeutung von Mitarbeiterinnen und Mitarbeitern in Form von Stress. Im Sinne eines schonenden Umgangs mit der „Ressource“ Arbeitskraft, sollte die Rückkopplung zum Vertrieb also enger, häufiger und fachlich kleinschrittiger sein als bisher. „Zu gute“ Vertriebspersonen sollten z. B. mit in die Product-Demos und Sprint-Plannings/Replenishment Meetings eingeladen werden, um direkt mitzubekommen wie und warum priorisiert und entwickelt wird.

### Falle 3: Drift zu schlechter Leistung

Es gibt Systeme, die sind nicht nur resistent gegen Regulierung; sie verharren auch nicht einfach in einem akzeptabel schlechten Zustand. Nein, sie werden kontinuierlich schlechter und schlechter.

Die Logik folgt der „Broken-Windows-Theorie“ [KellingWilson1982], nach der das Kriminalitätsaufkom-

<sup>1</sup> Das gemeinschaftliche Eigentum

men eines Stadtviertels mit seinem Verfall korreliert. Und auch das Einbrechen von Neujahrsvorsätzen gehört dazu. Wenn ich im Januar noch 3x die Woche laufen war, sieht das im März schon anders aus. Man kann statistisch recht solide Aussagen darüber treffen, mit welcher Wahrscheinlichkeit so etwas passiert.

Es handelt sich hierbei um sogenannte Hawkes Prozesse (nach Alan Hawkes, Professor Emeritus in Statistik der Swansea University), die man bei lose gekoppelten Systemen antrifft. Die Statistik dieser Prozesse wird z. B. für Predictive Policing [Frey2016] eingesetzt, das heißt für die Prognostizierung von Verbrechen auf Basis von Daten. Es heißt, man habe damit in den USA lokal Diebstahl bis zu 32% reduzieren können.<sup>2</sup>

### Beispiel aus dem Unternehmenskontext

In der Softwareentwicklung wissen wir: Schlechter Code bleibt schlecht oder wird noch schlechter. Das liegt daran, dass niemand mehr den Code warten (refactoren) mag oder kann. Hier handelt es sich ebenfalls um den Hawkes Prozess. Die Leistungsstandards können per se immer durch vergangene Leistungen beeinflusst werden, insbesondere durch schlechte.

### Der Ausweg

Wenn Leistungsstandards abfallen, sollten absolute Leistungsstandards eingebaut oder sogar erhöht werden, um eine Drift zu höherer Leistung zu erzeugen. Hoppla, wird da jetzt jemand rufen: höhere Leistungsstandards? Darf man Teams denn z. B. feste Velocities vorschreiben? Nein, Team-Velocities sind eine Metrik aber kein Leistungsstandard. Gängige Leistungsstandards des Software-Craftmanship sind z. B. die Clean-Code-Prinzipien und SOLID [Martin2008, WintersteigerMathis2012].

<sup>2</sup> Das Vorgehen trifft nachvollziehbarerweise auf Widerstand und Kritik, insbesondere wenn die Statistik personenbezogen ist. Selbst, wenn eine personenbezogene Auswertung z. B. auf Gangs statthaft wäre, kann die Methode nur dann signifikante Ergebnisse liefern, wenn die Stichproben groß genug sind. Und das ist selbst in den USA glücklicherweise nur in wenigen Districts der Fall.

### Falle 4: Eskalationen

Mit den Flottengesetzen ab 1898 und späteren systematischen Anpassungen bis 1912 startete Admiral von Tirpitz unter Kaiser Wilhelm II eine Reihe von Maßnahmen, die der britischen Flotte Paroli bieten sollten. Dies führte zu einem Wettrüsten mit Großbritannien, das aus heutiger Sicht als eine der Ursachen des 1. Weltkriegs angesehen wird.

Im kalten Krieg wiederholte sich diese Logik zwischen den USA und der UdSSR und führte mit der Kubakrise im Oktober 1962 beinahe zum 3. Weltkrieg. Erst seit 2002 weiß man, wie knapp die Menschheit der Gefahr eines Atomkriegs entrann. Die Sowjetunion hatte auf Kuba bereits 80 Nuklearsprengköpfe installiert, die für den Abschuss autorisiert waren. Gelöst wurde die Krise durch das Einlenken der Sowjets. Chruschtschow erklärte sich bereit, die Raketen zu entfernen.

Wenn der Zustand eines Systemteils dadurch bestimmt wird, dass man versucht, den Zustand eines anderen Systemteils zu übertreffen – und umgekehrt –, dann gibt es eine verstärkende Rückkopplung, die das System in ein Wettrüsten führt.

Dies können politische Krisen, eskalierende Gewalt, aber auch eskalierende Lautstärke sein. Die Eskalation ist exponentiell und kann erstaunlich schnell zu Extremen führen. Die negative Spirale wird dann nur durch einen Zusammenbruch einer Partei gestoppt – denn exponentielles Wachstum kann nicht ewig weitergehen. Andernfalls verlieren beide Parteien.

### Der Ausweg

Der beste Weg aus dieser Falle ist, nicht in sie hinein zu geraten. Wenn man jedoch in ihr gefangen ist, gibt es die Möglichkeit sich zu weigern mitzuspielen, z. B. durch einseitiges Abrüsten oder Aufgabe. Oder man handelt mit der anderen Partei eine ausgleichende Rückkopplungsschleife aus, die Eskalationen verhindert.

### Beispiel aus dem Unternehmenskontext

In Firmenkontexten werden Eskalationen als Priorisierungswerkzeug verwendet, wenn eine neue Aufgabe als dringlicher erachtet wird, als die geplante Aufgabe. Diese landet dann auf einer ►

Überholspur, zum Missfallen der anderen Stakeholder, denn auch sie hätten gerne ihre Features umgesetzt.

Eskalation im Sinne eines Wettrüstens findet hier darüber statt, indem der Stakeholder gewinnt, der am lautesten schreit. Der Streit wird auf dem Rücken der Teammitglieder ausgetragen. Der Aufwand vergangener Deadlines wurde umsonst geleistet und das Team muss wegen der Umpriorisierungen neu planen. Was hier helfen würde, ist eine ausgleichende Rückkopplungsschleife über alle Aufgaben aller Stakeholder und ihre Bewertung nach Geschäftswert auf globaler Ebene. Dies lässt sich mit firmenweit transparenten Taskboards erreichen, wie man sie bei Flight-Levels-Kanban [Leopold2017] antrifft.

### Falle 5: The winner takes it all – Erfolg den Erfolgreichen

Wenn der Gewinner eines Systems systematisch mit Mitteln belohnt wird, um damit wiederum Gewinne zu erwirtschaften, entsteht ebenfalls eine verstärkende Rückkopplungsschleife. Sie führt in eine Sackgasse, bei der die Gewinner am Ende alles haben und die Verlierer nichts.

Eindrucksvoll zu sehen ist diese Art der Wettbewerbsverzerrung zu Coronazeiten am Beispiel Amazon. Während alle Händler ihre lokalen Geschäfte schließen müssen, führte das bei Amazon zu sprudelnden Kassen.

### Der Ausweg

Sofern umsetzbar, so ist das Gegenmittel die Diversifizierung, die es den Verlierer:innen des Wettbewerbs ermöglicht, aus dem Spiel auszusteigen und ein neues zu beginnen. Das ist im Amazon-Beispiel für die meisten Geschäfte nicht so ohne Weiteres möglich. Von daher braucht es politische Hilfe, z. B. in Form von Kartellgesetzen, die den Teil des Kuchens beschränkt, den ein einzelnes Unternehmen gewinnen kann. Eine weitere Möglichkeit ist, Richtlinien zu formulieren, die das Spiel ausgleichen, indem sie Vorteile der stärksten Spieler:innen beseitigen oder den Vorteil der schwächsten Spieler:innen verstärken.

### Beispiel aus dem Unternehmenskontext

Es lohnt hierbei eine kurze Selbstreflexion. Das Phänomen beschreibt nämlich allgemein die Ungerechtigkeit von Privilegien. Im Allgemeinen schreiben wir unseren Erfolg unserem Können und unserer harten Arbeit zu. Weitaus relevanter ist jedoch der Beitrag des Zufalls<sup>3</sup>. Sei es der, für das Vorstellungsgespräch zur rechten Zeit am rechten Ort gewesen zu sein, die richtige Hautfarbe zu besitzen, gesund zu sein oder einfach, weil im Reisepass Deutschland steht und nicht etwa Honduras.

Die Effekte kontinuierlicher Benachteiligung sind nur aus der Sicht der Benachteiligten erlebbar; es fehlt die Rückkopplung ins System, die aber der Logik der Mehrheit folgt. Treffen wir auf die Ausnutzung von Privilegien, so sind empathische und zentrale Instanzen im Unternehmen nötig, die mit den Mitarbeiter:innen zusammen für ihre Minderheiten regulierende Maßnahmen entwickeln.

### Falle 6: Lastverschiebung und Abhängigkeit durch Symptombekämpfung

Wenn ich Frust habe esse ich Schokolade. Gelegentlicher Frust ist kein Problem. Es wird aber zu einem Problem, wenn ich zu häufig Frust habe und ich das dahinterliegende Problem ausblende. Mit dieser Strategie bleibt das erste Problem bestehen und ich gewinne ein neues: Gewichtszunahme mit anderen gesundheitlichen Folgebeschwerden. Man spricht dann von Lastverschiebungen (hier im Wortsinne).

Lastverschiebungen und Abhängigkeiten dieser Art entstehen, wenn die Lösung eines systemischen Problems zwar die Symptome reduziert, aber nicht das ursächliche Problem löst. Schokolade ist eine Brücke, sollte aber nicht zum Weg werden.

Wenn diese destruktive, sich verstärkende Rückkopplungsschleife zuschlägt, ist mehr von der Pseudolösung nötig. Das System wird abhängiger und ist weniger in der Lage, seinen eigenen gewünschten Zustand aufrechtzuerhalten.

<sup>3</sup> Vergleiche „Outliers: The Story of Success“ von Malcolm Gladwell [Gladwell2011].

## Der Ausweg

Auch hier ist der beste Weg aus der Falle, nicht hineinzugeraten. Hüten sollte man sich vor symptomlindernden Maßnahmen oder Praktiken, die das Problem nicht wirklich angehen sondern verschleiern.

Die systemische Lösung ist leider ein „low brainer“: den Fokus weg von der kurzfristigen Erleichterung hin zu einer langfristigen Kurierung des Problems zu legen. Leicht gesagt, schwer umzusetzen. Äußere Hilfe sollte willkommen geheißen werden.

## Beispiel aus dem Unternehmenskontext

Als offensichtliches Beispiel für Unternehmen sehe ich die häufige Abhängigkeit zu externen Dienstleistern (Entwickler:innen, Business Analysten, Coaches etc.), statt interne Mitarbeiterinnen und Mitarbeiter kontinuierlich in den eingekauften Kompetenzen auszubilden. Man könnte zwar, aber der interne Kollege bzw. die interne Kollegin muss ja das Holz hacken, statt die Axt zu schärfen.

In diesen Situationen sollten wir dem Prinzip folgen:

*„Brauchst du's häufig, tue es selbst;  
brauchst du's selten, du bestellst.“*

In vielen Unternehmen wird die Diskussion aber nicht mit dieser Unterscheidung geführt, erst recht nicht in Cost-Centern, denn dort fehlt die Rückkopplung zum Aufwandsursprung. Profit-Center sind da im Vorteil. Es ist gut mit beschränkten Ressourcen auskommen zu müssen. Die Rückkopplung zu Gewinn und Verlust schafft Fokus aufs Wesentliche.

## Falle 7 Das System ausspielen – Gaming

Regeln zur Steuerung eines Systems können zu Regelübertretungen führen. Verhalten, das den Anschein erweckt, die Regeln zu befolgen oder die Ziele zu erreichen, aber in Wirklichkeit das System verzerrt.

Ein bekanntes Beispiel sind Stechuhren, die pünktlich betätigt werden, um dann erstmal eine rauchen zu gehen. Ein weiteres Beispiel sind die Endjahres-Rallies, die dazu führen Budgets zu verballern, damit man zur Budgetvergabe im neuen Jahr ein ähnlich hohes bekommt.

## Der Ausweg

Neue Metriken definieren, die zum Ziel haben, den dahinter liegenden Zweck der Metriken zu erreichen.

## Beispiel aus dem Unternehmenskontext

Die Testabdeckung beschreibt in der Software-Entwicklung die Anzahl von Quelltextzeilen, die durch automatisierte Tests gestreift werden. Eine hohe Testabdeckung (Ziel) bedeutet Aufwand, erlaubt aber keine direkten Rückschlüsse auf die Quelltextqualität (Zweck). Wenn Teams in Product Demos, ihre Testabdeckung zeigen, weil sie den Stakeholdern damit „beweisen“ wollen, dass ihre Quelltextqualität mit der Zeit besser wird, werden sie damit dem Ziel, die Testabdeckung zu maximieren immer näher kommen, jedoch muss das nicht für den Zweck gelten. Nun gibt es in Software kritische Klassen, die häufig durchlaufen werden und solche, die eher seltenere Use-Cases abdecken. Sofern die Testabdeckung < 100% ist, ist es naheliegend, dass nicht die Masse der Testabdeckung entscheidend ist, sondern ihre Auswahl.

Wenn das Team sich als Ziel hingegen vornehmen würde, testgetrieben zu entwickeln [TDD2021, Beck2002] - also vor dem Feature zunächst den in einen Fehler laufenden Test zu schreiben – so fokussiert man auf den gewünschten Zweck; ergo Qualität.

## Falle 8: falsche Zielsetzungen.

Jedes System ist besonders empfindlich gegenüber den Zielen seiner Rückkopplungsschleifen. Wenn die Ziele – die Metriken für die Erfüllung der Regeln – ungenau oder unvollständig definiert sind, kann das System folgsam auf ein Ergebnis hinarbeiten, das aber eigentlich nicht beabsichtigt oder sogar unerwünscht ist.

Es kommt zu diesen Effekten immer dann, wenn Ergebnisse mit Zielen verwechselt werden.

## Der Ausweg

Ziele definieren, die den tatsächlichen Nutzen des Systems widerspiegeln. ►

### Beispiel aus dem Unternehmenskontext

Ein von der Geschäftsführung verkündetes Unternehmensziel von 6% Wachstum ist kein Ziel, sondern ein mögliches Ergebnis eines dahinterliegenden Unternehmenszwecks. Bessere Ziele bzw. Metriken sind Konversionsraten, Ertrag pro Benutzer, Kundengewinnungskosten, Kundenabwanderungsrate, Abo-Zuwachsraten, Abo-Verlängerungen etc., weil sie das Ziel bzw. die Metrik mit dem Zweck koppeln.

### Reflexionsfragen:

Fühlten Sie sich bei einer oder mehreren der beschriebenen Fallen an Ihr Unternehmen erinnert?

Welche konkrete Falle ist es? Wie sind Sie ihr damals oder zuletzt begegnet? Welche Schlüsse ziehen Sie daraus?

An welchen Stellen Ihrer Organisation müssen Sie aufpassen, nicht in eine dieser Fallen zu laufen? ■

### Referenzen

[Beck2002] Kent Beck: Test Driven Development by Example, Addison Wesley 2002

[Emmons2015] Patrick Emmons: Get Lean: Avoid 8 Wastes in Software Development, <https://www.dragonspears.com/blog/get-lean-avoid-8-wastes-in-software-development>, zuletzt besucht 7.4.2021.

[Frey2016] Predictive Policing, Hannah Fry: The Mathematics of Crime and Terrorism, Numberphile, <https://www.youtube.com/watch?v=ICjspXB5F4A>, zuletzt besucht 17.6.2021.

[Gladwell2011] Malcom Gladwell: Outliers: The Story of Success, Penguin Books 2011

[KellingWilson1982] Broken Windows Theorie, Wikipedia, <https://www.wikiwand.com/de/Broken-Windows-Theorie>, zuletzt besucht 7.4.2021.

[Leopold2017] Klaus Leopold: Flight Levels: Die Verbesserungsebenen der Organisation <https://2020.leanability.com/de/blog/2017/04/flight-levels-die-verbesserungsebenen-der-organisation/>, zuletzt besucht 17.6.2021.

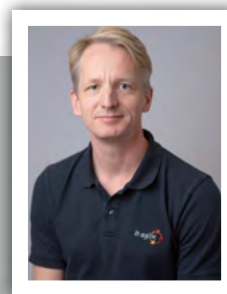
[Martin2008] Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall 2008.

[Meadows2015] Donella H. Meadows, Diana Wright (Hrsg.): „Thinking in Systems: A Primer“, 218 Seiten, Chelsea Green Publishing Co., 2015.

[TDD2021] Was ist Testgetriebene Entwicklung? <https://www.it-agile.de/wissen/agiles-engineering/testgetriebene-entwicklung-tdd/>, zuletzt besucht 17.6.2021.

[WintersteigerMathis2012] Andreas Wintersteiger, Christoph Mathis: Clean Code, Prinzipien bei der Entwicklung von sauberem Code, <https://entwickler.de/online/agile/clean-code-134128.html>, zuletzt besucht 17.6.2021.

Treten Sie gern mit dem Autor in Verbindung: [ralf.lethmate@it-agile.de](mailto:ralf.lethmate@it-agile.de)



RALF  
LETHMATE

Ralf Lethmate wurde 1972 geboren, ist promovierter Physiker und hat einen Master of Business Administration (MBA). Als Projektleiter lernte er seit 2002 zunächst klassisches Projektmanagement in KMUs und internationalen Konzernen kennen. Mit der Zeit entfachten Scrum und Kanban seine Begeisterung für agile Praktiken. Da er selbst schon in die eine oder andere beschriebene Falle lief, schließt er seither mit Herzblut Feedback-Schleifen.



# Empirische Prozesskontrolle

## Was ist es?

Agile Produktentwicklung und insbesondere Scrum baut auf die Idee empirischer Prozesskontrolle.

Empirie bedeutet, Erkenntnisse aus Vorwissen und Erfahrung zu gewinnen, und Entscheidungen aus diesen Erkenntnissen abzuleiten, anstatt aus Schlussfolgerungen.

Die Empirische Prozesskontrolle bedient sich dazu dreier „Säulen“: Transparenz, Überprüfung und Anpassung, oft schlagwortartig verkürzt zu „Inspect & Adapt“.

Eine charakteristische Ausprägung empirischer Prozesskontrolle in agilen Methoden ist das Vorgehen in kurzen Zyklen mit regelmäßiger Anpassung des Plans.

Der Begriff der empirischen Prozesskontrolle steht im Kontrast zur statistischen Prozesskontrolle nach Walter Shewart und W. E. Deming, die auf Spezifikation, Überwachung und Eliminierung von Abweichungen setzt.

## Warum?

Empirische Prozesskontrolle ist insbesondere in komplexen Umfeldern nützlich, die durch schnelle Veränderung der wirtschaftlichen, sozialen oder technischen Umgebung gekennzeichnet ist.

In solchen Umfeldern, also z. B. bei der Entwicklung

eines innovativen neuen Produkts, ist die Veränderung nicht langfristig vorhersehbar. Vorausplanung der Schritte der Entwicklung oder Produktion ist daher nur eingeschränkt oder gar nicht möglich.

Empirische Prozesskontrolle ermöglicht Unternehmen, diesem Problem zu begegnen, indem sie Werkzeuge und Methoden zur regelmäßigen Analyse und Anpassung des Plans bereitstellt.

## Wie?

Unternehmen implementieren empirische Prozesskontrolle, indem sie Datensammlung über den Fortschritt der Arbeit und die Entwicklung des Umfelds (Transparenz) und Zeitpunkte zur Überprüfung und Anpassung in regelmäßigen Intervallen in ihre Prozesse integrieren.

Viele Agile Methoden bieten Ansätze dazu; im Falle Scrum sind es (insbesondere) die vier regelmäßigen Scrum-Ereignisse.

Die Verbindung empirischer Prozesskontrolle mit statistischer Prozesskontrolle zum Zwecke kontinuierlicher Verbesserung hat Deming im Plan-Do-Check-Act-Zyklus (PDCA) beschrieben (vgl. Abb. 1). In den 4 Phasen des PDCA wird eine Veränderung erst vom Ziel ausgehend geplant (Plan), dann von Überwachung begleitet umgesetzt (Do), anschließend die Ergebnisse ausgewertet (Check) und die

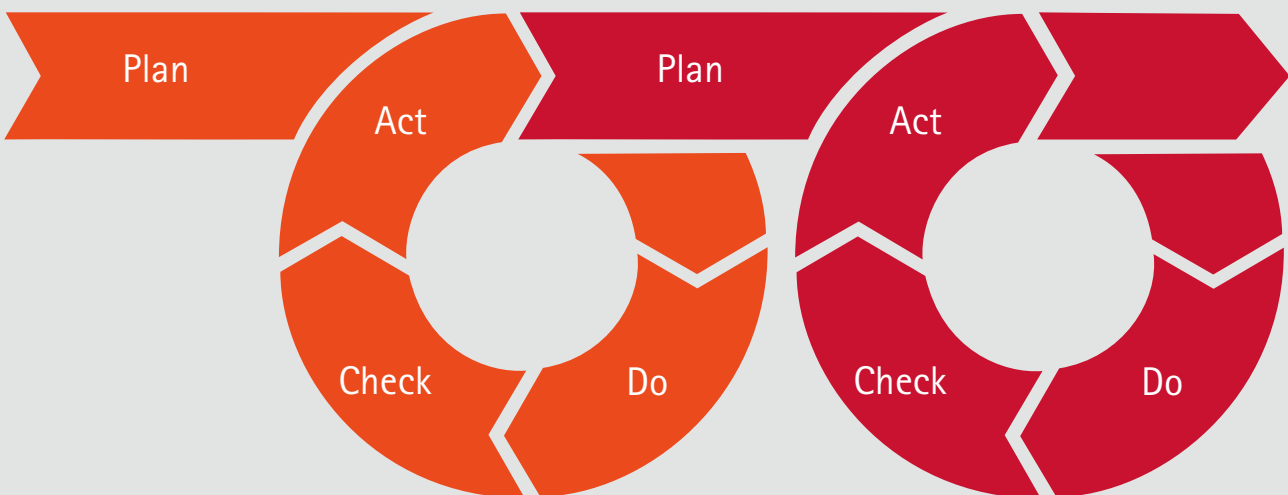


Abb. 1: Plan-Do-Check-Act mit zwei Durchläufen

# Delivering Fluency

## Was ist es?

Delivering Fluency ist die zweite Zone des Agile Fluency® Models von Diana Larsen und James Shore. Es gibt vier Zonen: „Focusing“, „Delivering“, „Optimizing“ und „Strengthening“. Mit jeder Zone ist eine Reihe von Fähigkeiten verknüpft, die ein Team beherrschen muss. Hat das Team die Fähigkeiten ausgebildet, kann es eine Reihe von Vorteilen nutzen. Die Zonen helfen mir zu bestimmen, welchen Grad an Agilität ich benötige. Dafür betrachte ich die jeweiligen Vorteile, die mir eine Zone bietet. Gleichzeitig kann ich mit dem Modell untersuchen – unabhängig von der gewählten agilen Vorgehensweise –, ob ich die notwendigen Fähigkeiten erreicht habe.

## Vorteile

Delivering Fluency bietet die folgenden Vorteile:

1. Das Team kann immer dann ein Release herausgeben, wenn es der Product Owner benötigt.
2. Die durch Feature-Entwicklung geschaffenen Werte können regelmäßig erschlossen werden.
3. Hindernisse werden durch häufige Produktivstellung aufgedeckt.

Immer dann ein Release herauszugeben, wenn es der Product Owner benötigt bedeutet, dass alle Features, die bis gestern als „fertig“ benannt wurden, in dieses Release aufgenommen werden können. Es bedeutet ferner, dass die Release-Entscheidung vom Product Owner und nach Markt-Erfordernissen getroffen wird.

Der zweite Vorteil beruht darauf, dass die Entwicklungsarbeiten als Investition gesehen werden, die sich erst dann auszahlen, wenn die Features von Anwender:innen genutzt werden können. Mit diesem Vorteil wird der wirtschaftliche Aspekt erschlossen.

Der dritte und für das Entwicklungsteam wesentliche Vorteil ist, dass durch den Einsatz der Software in der Produktionsumgebung Unsicherheiten bzgl. unbekannter Hindernissen ausgeräumt werden.

## Verlagerung zu Team-Fähigkeiten

Die entscheidende Veränderung hin zur Delivering Zone betrifft die Team-Fähigkeiten (Team Skills Shift). Damit ist gemeint, dass für das Erreichen die-

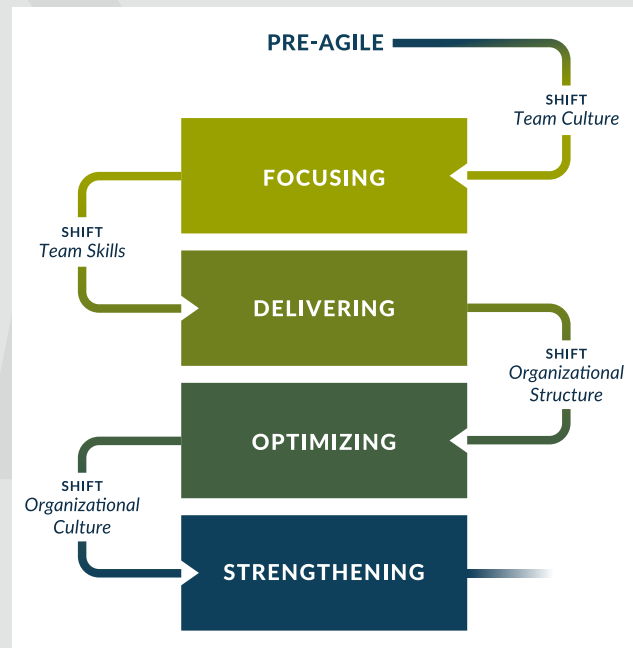


Abbildung 1: Das Agile Fluency Model

ser Zone substantielle handwerkliche Fähigkeiten aufgebaut werden müssen. Dazu gehören Programmierfähigkeiten und die Fähigkeiten, die Software in die Produktionsumgebung zu bringen und dort zu betreiben (DevOps). Alle Fähigkeiten müssen so erlernt werden, dass sie in Fleisch und Blut übergehen und im Alltag selbst unter Stress weiter zuverlässig abgerufen werden können.

## Investitionen

Um die Delivering Zone zu erreichen, sind eine Reihe von Investitionen notwendig:

1. Technisches Training  
Um in kurzen Zyklen zuverlässig robuste Software auszuliefern, sind handwerkliche Fähigkeiten der Softwarekonstruktion notwendig. Dazu kann das Team Extreme Programming o. ä. praktizieren.
2. Automatisierte Tests  
Wesentlich für die häufige Auslieferung in Produktion sind automatisierte Tests. Alle Ebenen des Testens lassen sich automatisieren – erfordern aber unterschiedliche technische Fähigkeiten. Diese frühzeitig aufzubauen ist entscheidend, um im späteren Verlauf des Projektes nicht ausgebremst zu werden.



# Scrum: Product Backlog Refinement

## Was ist es?

Product Backlog Refinement ist jeder Vorgang, in dem das Product Backlog um Details oder Schätzungen angereichert oder neu geordnet wird. Damit gehört es zu den Hauptaufgaben von Product Ownern in Scrum.

Spezieller bezeichnen viele Praktiker:innen als Product Backlog Refinement spontane oder geplante Treffen eines ganzen Scrum Teams, in dem es diese Tätigkeiten vollzieht.

Teil des Product Backlog Refinement ist ausdrücklich auch, Fragen über Einträge im Product Backlog aufzubringen oder einen Eintrag in mehrere aufzuteilen, denn beides fügt auf unterschiedliche Arten Verständnis oder Details hinzu.

In älterer Literatur wird das Product Backlog Refinement auch als „(Product Backlog) Grooming“ bezeichnet.

## Warum?

**Für Product Owner** ist es unerlässlich, das Product Backlog so aufzubereiten, dass die Entwickler:innen die Einträge darin selbständig umsetzen können. Den nötigen Detailgrad und eventuelle Fragen zu antizipieren, gelingt nur den wenigsten.

**Für Entwickler:innen** ist es unerlässlich, die Einträge im Product Backlog zu verstehen. Erfahrene Gruppen, die mit Produkt und Markt sehr vertraut sind, können dieses Verständnis während des Sprints im Dialog mit den Stakeholder:innen oder sogar aus sich heraus generieren.

Daher ist es im Interesse beider Parteien, im Vorfeld der Umsetzung gemeinsam Details zu klären und Unklarheit bis zu dem Punkt abzubauen, an dem die Entwickler:innen den Eintrag umsetzen können. So vermeiden Scrum Teams Missverständnisse und ermöglichen den

Entwickler:innen, das Richtige zu bauen.

## Wie?

Scrum gibt die Form des Product Backlog Refinement nicht vor. Verbreitet ist, Product Backlog Refinement mit dem ganzen Scrum Team in einem regelmäßigen Treffen durchzuführen.

Je nach Vorliebe der Beteiligten finden dazu kürzere Treffen in jeder Sprintwoche oder längere einige Tage vor dem Sprintabschluss statt.

Während der Treffen obliegt es dem Product Owner oder einem eingeladenen Fachexperten, neue Einträge im Product Backlog vorzustellen und Fragen der Entwickler:innen dazu zu beantworten.

Damit Product Owner im Falle fehlender Antworten diese noch rechtzeitig zum Sprintstart nachliefern können, sollten das Treffen ausreichend weit vor dem Sprintstart liegen.

Die „Things That Matter“-Matrix (siehe Abb. 2) ist ein nützliches Werkzeug, um den Dialog im Product Backlog Refinement zu strukturieren. Mit ihr klärt das Scrum Team zunächst für jedes involvierte Gewerk im Team die Aspekte und Fragen zum

## Product Backlog

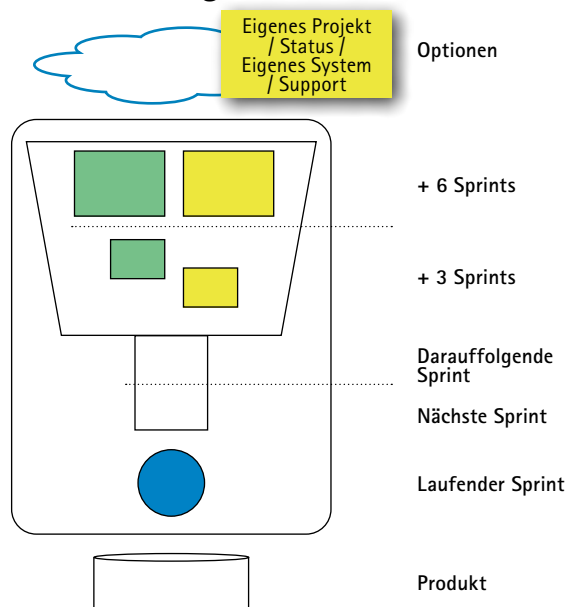


Abb. 1: Der Product-Backlog-Trichter



# Die agile review bequem im Abo!

ab 20 € pro Jahr (3 Ausgaben)



Hier Abo bestellen: [www.agilereview.de](http://www.agilereview.de)



Abb 1: Hier sehen Sie einen Ausschnitt unseres Bücherregals im Büro

## Team Topologies - Organizing Business and Technology Teams for Fast Flow

Matthew Skelton, Manuel Pais, 216 Seiten, IT-Revolution, Portland Oregon, ISBN 978-1-942788-81-2, 2019.

Erfolgreich agieren in komplexen Herausforderungen benötigt immer Kooperation. Ein Mensch wird eine komplexe Aufgabe nicht alleine bewältigen. Auch ein Spitzenathlet in einer Individualsportart wie Tennis oder Golf ist zum größten Teil seines Arbeitsalltags in einem Team eingebunden.

Für die Entwicklung von Software ist das schon lange bekannt, agile Vorgehensweisen sind dementsprechend deutlich auf Teamarbeit ausgelegt. Dabei ist die Digitalisierung unseres Lebens mittlerweile so weit fortgeschritten, dass kaum noch analoge Prozesse existieren. Diese massive Einwirkung von Software auf unser Leben macht einen Punkt deutlich: Ein Team alleine reicht in der Regel nicht, um moderne Software-Anwendungen gleichermaßen schnell und von hoher Qualität zu liefern. Die dafür benötigten Fähigkeiten sprengen entweder eine angemessene Teamgröße oder die kognitive Belastbarkeit eines Teams. Den immer noch üblichen Kompromiss Schnelligkeit zulasten der Qualität einzugehen, können sich Start-ups leisten, deren Scheitern eine valide Option ist. Was nicht mehr geht: zugunsten

hoher Qualität lange Releasezyklen in Kauf zu nehmen. Man denke nur an die Dynamik, die durch die Pandemie an allen Fronten der Digitalisierung entstanden ist.

Ein Beispiel sind die Schwierigkeiten ein fälschungssicheres Impfzertifikat zu erstellen. Hierzu kam die Nachricht, dass die Apotheken vorerst keine Zertifikate mehr ausstellen dürfen, weil das System leicht zu manipulieren ist.

An dieser Stelle setzt Team Topologies an und nimmt seine Leser mit auf eine theoretische aber anwendungsfreundliche Lesereise über die Verbindung von Teamkultur und Organisations-Design.

Eine der zentralen Thesen des Buches ist: Conway's Law sei trotz seines Ursprungs in den 60er Jahren höchst aktuell. Die Architektur eines Systems korreliert stark mit der Kommunikationsstruktur des Unternehmens, in dem das System entwickelt wurde. Jeder Versuch, die Architektur des Systems zu ändern, ohne dabei die Kommunikationsstruktur entsprechend anzupassen, würde daher scheitern. Und das sei ein Dilemma, denn eine zeitgemäße Architektur sei Grundpfeiler für die benötigte Flexibilität in der Entwicklung. Gleichzeitig seien die meisten Organisationen weiterhin in funktionalen Silos strukturiert und nicht anhand ihrer Wertströme. Die Autoren



# Von der eigenen Maschine in die Cloud – ein Erfahrungsbericht

Von Andreas Havenstein und Steven Collins

Von dem wohlbehüteten, selbstverwalteten Umfeld in die Cloud zu migrieren ist für viele Unternehmen ein abenteuerlicher Schritt. Es stellen sich technische und organisatorische Fragen:

- Wie kann eine sanfte Migration von einem selbstbetrieblenen Rechenzentrum zur Cloud gelingen?
- Wie frei können Teams über die Cloud-Ressourcen verfügen?
- Braucht man noch Operations in der Cloud?
- Führt die Cloud zu neuen Team-Schnitten?
- Welchen Beitrag zur Agilität kann eine Cloud-Infrastruktur leisten?

Wir haben im Rahmen eines großen Onlineshop-Systems die Migration in die Cloud begleitet und werden diese Fragen im Folgenden beleuchten.

## Ausgangssituation

Der Kontext, in dem wir die Migration begleitet haben, ist ein großes Onlineshop-System für Endkunden. Der gesamte Shop ist in einem agilen Umfeld entwickelt worden. Teams organisieren sich nach verschiedenen, jeweils für sie am besten passenden agilen Paradigmen wie Kanban oder Scrum. Als Leitbild für die Makro-Architektur wurde eine über REST eingebettete Vertikalen-Architektur gewählt: Statt eines monolithischen Gesamtsystems werden fachliche Teilprozesse und -Domänen in eigenen vertikal (sowohl das Front- als auch das Backend umfassend) geschnittenen Systemen entwickelt. Diese unabhängigen Systeme werden als Self Contained Systems bezeichnet [SCS2020]. Abb. 1 stellt beispielhaft drei Vertikalen des Gesamtsystems dar. Die Vertikalen sind unterschiedlichen Teams zugeordnet und können ohne großen Abstimmungsaufwand entwickelt werden.

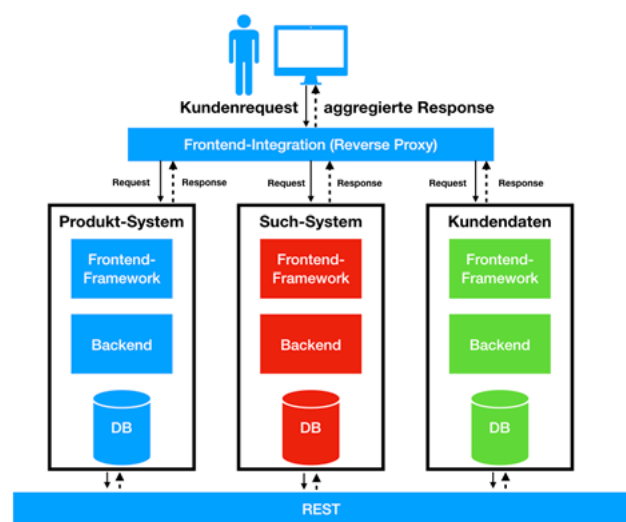


Abb. 1: SCS-Vertikalen

Die von den Teams verantworteten Vertikalen kommunizieren über REST und werden über ein Reverse-Proxy zu einem Gesamt-Shop kombiniert. Der Request des Kunden wird zu den Servern der Vertikalen geleitet und dann im Reverse-Proxy zu einer Ergebnisseite zusammengefügt. Dieses Leitbild ermöglicht unabhängiges, agiles Arbeiten:

- Teams können innerhalb ihres SCS weitgehend autonom handeln. Nach außen muss ein SCS einen an der REST-Schnittstelle auf Makro-Ebene definierten Vertrag erfüllen. Wie aber eine Vertikale in der Mikro-Architektur definiert ist, welche Programmiersprachen, Rahmenwerke oder Serverkomponenten zum Einsatz kommen, kann von jedem Team individuell bestimmt werden.
- Die Shop-Bestandteile der Teams können unabhängig voneinander beliebig häufig ausgeliefert ►

# Immer noch nicht genug?

Dann abonnieren Sie doch den **Newsletter** von it-agile!

it-agile Newsletter Oktober 2021



**WiP-Limits, Team Building, Tools4AgileTeams, Scrum-Buch, Agiler Tipp, Leadership des Monats**

Guten Tag,

große Veränderungen lassen sich grundsätzlich agil gedacht in vielen kleinen Schritten durchführen. Manchmal aber braucht es den großen Schritt, die große und krasse Veränderung, um ein Ziel zu erreichen. Dem großen Schritt können dann viele kleine agile Schritte der Anpassung folgen. Voller groß unser gemeinsamer Sprung als Gesellschaft beim Thema Klima wird, werden wir bald erf... Oder wir gestalten mit. Denn längst beginnen Unternehmen sich anzupassen. Bei manchen w... agile reichen dafür kleine Schritte, bei anderen stehen große Umwälzungen an. Was es in jedem Fall braucht für solche Veränderungen, sind Geduld, langer Atem und Leader, die dranbleiben.

In diesem Newsletter findest du folgenden Themen:

- Artikel-PDF für dich: „Warum WiP-Limits schwierig sind“
- Flawless Team Building: 8.-10. Dezember nachmittags (online)
- Tools4AgileTeams am 2./3. Dezember 2021: kostenlos für dich
- 3. Auflage Scrum-Buch: Reviewer und Feedback gesucht
- Agiler Tipp: Gleich und Verschieden (Same and Different)
- Schulungsempfehlungen für dich
- Leadership des Monats: Erklären

Ich wünsche dir viel Spaß beim Lesen und ausreichend Geduld für deine Vorhaben,  
Henning.

P.S.: Die XP Days Germany finden am 4. und 5. November in Hamburg als Präsenzveranstaltung (mit Hamburger 2G-Regel) statt. Hier kannst du dich über das spannende Programm informieren und dich anmelden.

P.P.S.: Voraussichtlich ab November wird der Newsletter nur noch halb so lang sein, dafür aber zweimal im Monat erscheinen. Damit reagieren wir auf häufiges Feedback und können noch aktueller über Neues aus der (it-)agilen Welt berichten.

informativ

aktuell

auf den Punkt

Leadership des Monats

Einfach anmelden auf: [www.it-agile.de](http://www.it-agile.de)

direkt zum Angebot





Entwicklung, Beratung, Schulung



*Was wollen  
Sie erreichen  
und wofür?*



direkt zum Angebot



***Nehmen Sie jetzt Kontakt mit uns auf, um den nächsten Schritt zu wirkungsvoller Agilität zu gehen:***

**☎ 0800 / 482 4453 ✉ [wirkungsvoll@it-agile.de](mailto:wirkungsvoll@it-agile.de)**