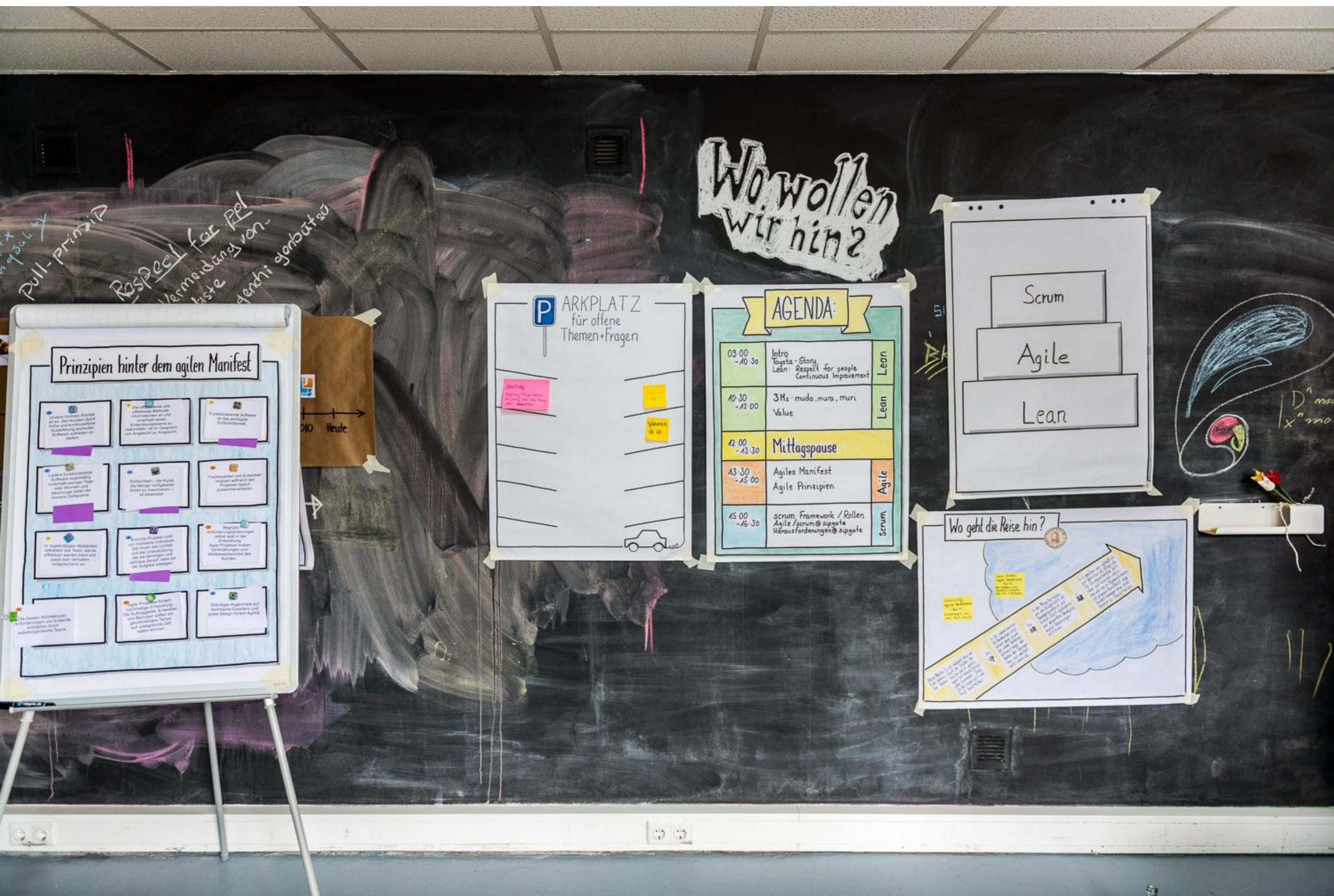


lean@



sipgate

Über die Grundlagen von Lean, der agilen Arbeitsweise und Scrum



einleitung

1

Hier bei sipgate arbeiten wir *lean* und sind *agil*. Außerdem machen wir *Scrum*, und *Kanban* gibt es auch. Wenn wir nun zusätzlich noch von *Standups*, *Retros* und *Plannings* sprechen, erwecken wir bei Dir möglicherweise schnell den Eindruck von **Buzzword-Bingo**. Diese Wahrnehmung wird der Sache aber nicht gerecht, denn hinter diesem Vokabular verbirgt sich der **Kern unserer Philosophie** und unserer **gesamten Arbeitsweise**. Dementsprechend liegt es uns sehr am Herzen, den Inhalt dieser Begriffe mit Leben zu füllen und Dir deren Sinn und Sinnhaftigkeit zu vermitteln.

Was genau sich hinter Lean, Agile und Scrum verbirgt, kannst Du auf den folgenden Seiten herausfinden. Für das Erste erklären wir die *Zusammenhänge* zwischen Lean, Agile und Scrum.

lean? agile? scrum?

1

Lean, Agile und Scrum lassen sich als die Stufen einer Pyramide darstellen: **Lean** bildet dabei die Basis, das Fundament. Lean (oder *Lean Production*) hat seinen Ursprung in der japanischen Automobilindustrie. Dabei handelt es sich sowohl um ein konkretes Produktions-System, als auch um eine Philosophie, die vor allem die Bedeutung von **wertorientierter Arbeit, kontinuierlicher Verbesserung** und **respektvollem Umgang miteinander** in den Vordergrund stellt.

Mehr über den Hintergrund von Lean und die darin enthaltenen Prinzipien erfährst du in **Abschnitt 2**.

SCRUM

AGILE

LEAN

lean? agile? scrum?

1

Bei **Agile** handelt es sich um die konsequente Anpassung der Lean Production auf die **Softwareindustrie**. Agile liefert, ebenso wie Lean, eine Reihe von Prinzipien, die die Arbeit und das Miteinander bei sippgate maßgeblich mitgestalten.

Agile ersetzt aber nicht etwa Lean - das Gegenteil ist der Fall! Agile formuliert einen Leitfaden, der ganz wesentlich auf den *lean principles* beruht und diese weiter konkretisiert.

Mehr über Agile erfährst du in **Abschnitt 3**.

SCRUM

AGILE

LEAN

lean? agile? scrum?

1

Bei **Scrum** handelt es sich schließlich um *ein* Framework der agilen Softwareentwicklung (es gibt noch weitere, z.B. **Kanban**). Scrum beschreibt, wie sich Softwareentwicklung im **Rahmen von Lean und Agile** vernünftig organisieren lässt.

Auch hier gilt es noch einmal, den Zusammenhang zwischen Lean, Agile und Scrum hervorzuheben: Um Scrum sinnvoll praktizieren zu können, müssen auch die Grundsätze von Lean und Agile im Unternehmen gelebt werden!

Mehr zum Thema Scrum erfährst du in **Abschnitt 4**.

SCRUM

AGILE

LEAN

Lean - Die Toyota Story

2

Die Geschichte von Lean beginnt mit der Geschichte der Familie Toyoda. Sakichi Toyoda wächst als einfacher Tischler gegen Ende des 19. Jhrd. in Japan auf. Für die heimische Manufaktur entwickelt er in Eigenarbeit Webstühle, die er später mit Hilfe von Dampfmotoren und viel Bastelei antreibt. Dabei lernt er schon früh, dass man gewissen Dinge einfach selber ausprobieren muss, um Resultate zu erzielen (**Trial and Error Prinzip**).

1926 gründete er die Toyoda Automatic Loom Works (die Mutterfirma des heutigen Toyota-Konzerns). Nun hatte er einen Webrahmen, der den Arbeitern viel Arbeit abnahm, aber auch das reichte ihm nicht.



Es kam immer mal wieder vor, dass beim automatischen Weben ein Faden riss, was Stoffbahnen minderwertig oder sogar wertlos machte. Also baute Opa Toyoda eine **Qualitätssicherung** ein: Wenn der Faden riss, stoppte die Maschine die Arbeit. So konnten Fehler sofort behoben werden und eine viel hochwertigere Stoffbahn gefertigt werden. Außerdem konnte ein Arbeiter mehrere Webmaschinen bedienen.

Opa Toyoda trug später den Namen "Japans Erfinder-König", aber eigentlich war sein Beitrag zum Erfolg der Familie Toyoda ein anderer, nämlich seine Philosophie, dass man **immer etwas verbessern** kann, und dass man sich dafür die Hände schmutzig machen muss.

Da seine Firma wirklich gut lief, könnte man erwarten, dass Opa Toyoda sie einfach seinem Sohn, Papa Toyoda (Kiichiro Toyoda), übergeben würde. Aber Opa Toyoda wusste, dass Webstühle eine Technologie der Vergangenheit waren und dass die Zukunft im Automobilgeschäft lag.

Lean - Die Toyota Story

2

Außerdem war er der Meinung, dass jeder seine Chance bekommen sollte, etwas Großes zu dieser Welt beizutragen. Also ließ er seinen Sohn studieren und dieser fing ca. 1930 an, eine Automobilsparte aufzubauen.

Nach dem Ersten Weltkrieg hatten Henry Ford und Alfred Sloan von GM die Welt in das Zeitalter der Massenproduktion geführt, deshalb besuchte Papa Toyoda in den 1930ern Ford und die neuen Supermärkte, um von den Großen zu lernen. Er übernahm die Philosophie seines Vaters, fügte aber seine eigene Innovation hinzu. Er übernahm von den Supermärkten das **Pull-System**, in dem Vorräte immer erst dann nachgefüllt wurden, wenn man sie tatsächlich brauchte, so dass man Lagerkosten sparen konnte. Außerdem testete er im väterlichen Webstuhl-Unternehmen Fließbänder und die Idee der Economies of Scale (also Einsparungen durch Massenproduktion). Ihm war klar, dass der japanische Markt zu klein und zu fragmentiert war, um die Techniken der Massenproduktion eins zu eins umzusetzen. Man musste sie anpassen, aber wie? Damals fand er keine Antwort.

1937 gründete er dann die Toyota Motor Company. Die Firma stellte eigentlich PKWs her, aber die Regierung zwang sie, LKWs für den bevorstehenden Pazifikkrieg zu bauen. Nach dem Krieg und dem Wiederaufbau der Firma (nach den zwei Atombomben fehlte es an allem: Zulieferer, Arbeitskräfte, Kunden) fielen die Verkäufe 1949 so stark ab, dass Toyota gezwungen war, ein Viertel der Arbeitskräfte (ca. 1.600) zu entlassen.

Das löste massive Streiks aus, so dass Papa Toyoda schließlich die Konsequenzen zog und zurücktrat und damit die volle Verantwortung für das Scheitern der Firma übernahm. Nicht nur das war ungewöhnlich, sondern auch die Tatsache, dass er mit den Gewerkschaften verhandelte. Danach wurde ein Viertel der Arbeiter zwar entlassen, aber die verbleibenden Dreiviertel erhielten einen lebenslangen unkündbaren Vertrag von Toyota.

Lean - Die Toyota Story

2

Das sicherte Toyota das Vertrauen und das Engagement seiner Mitarbeiter. Sie wurden nicht wie austauschbare Ware, sondern **mit Respekt** behandelt.

Den Firmenvorsitz übernahm Eiji Toyoda, der jüngere Cousin von Papa Toyoda. In den 13 Jahren seiner Firmengeschichte hatte Toyota insgesamt ca. 2.700 Autos gebaut. Verglichen mit den 7.000 Autos, die das Ford-Werk "The Rouge" täglich ausspuckte, war das nicht der Rede wert. Der Markt war voller Automobilfirmen, die bereit waren, nach Japan zu expandieren und den japanischen Markt zu übernehmen. Also fuhr auch Eiji mit seinen Managern in die USA, um zu sehen, welche Fortschritte die Massenproduktion mittlerweile gemacht hatte und wie man vom Automobil- und Wirtschaftsriesen Ford lernen konnte.

Sie erwarteten große Erkenntnisse, die die Erleuchtung für die Zukunft von Toyota bringen sollten. Als sie die Werke besuchten, waren sie alles andere als überzeugt: Sie fanden einige Makel an dem System, das damals als das Beste der Welt galt.

Wieder in Japan erteilte Eiji seinem Werksleiter Taiichi Ohno die Aufgabe, die Produktivität der Toyota-Werke auf das Niveau von Ford zu steigern. Das schien vor dem Hintergrund der Economies of Scale fast aussichtslos, aber Ohno nahm die Herausforderung an. Er fuhr mehrfach in die USA und las Fords Buch "Today and Tomorrow".

Ford predigte darin den gleichmäßigen Fluss der Materialien durch die Produktionsstraße, standardisierte Prozesse und die Vermeidung von unnützem Aufwand. Aber die Ford-Werke setzten diese Werte nicht um. Darin sah Ohno seine Chance. Er wusste, wenn er all die Makel des Systems ausmerzen würde und es auf den Bedarf des japanischen Marktes anpassen konnte, würde das die Produktivität von Toyota massiv steigern.

Die Hauptprobleme, die er bei Ford sah, waren u.a.:

- Ford hatte große und teure Maschinen, die sie auslasten mussten, damit es sich lohnte. Das führte zu Überproduktion von Teilen. Diese Teile mussten gelagert werden, was wiederum Geld kostete.
- Ford brauchte Leute, die die teuren Maschinen bedienten, und da die Maschinen immer ausgelastet sein mussten, waren auch die Arbeiter immer ausgelastet - oder sogar überlastet.
- Da für jedes Bauteil eine eigene Maschine benötigt wurde, war die Anzahl der Modelle, die Ford produzieren konnte, stark begrenzt.
- Da Ford von allen Teilen große Chargen herstellte, die dann zunächst gelagert wurden, wurden Fehler immer erst spät entdeckt.
- Beim Zusammensetzen der Autos mussten die Arbeiter in erster Linie darauf achten, dass das Fließband weiterlief. Für fehlerhafte Fahrzeuge gab es am Ende der Fertigungsstraße ein separates Team, das die Fehler im Nachhinein behob. Das brauchte Platz, zusätzliche Arbeitskräfte, die Fehlerbehebung dauerte wesentlich länger und Fehler wiederholten sich eventuell, so dass ganze Chargen von Autos fehlerhaft waren. Außerdem übernahmen die Arbeiter keine Verantwortung für Fehler, weil sie ja wussten, dass es am Ende jemanden geben würde, der sie behebt.

All diese Fehler im System resultierten daraus, dass auf Auslastung der Maschinen und Arbeitskräfte anstatt auf den Ausstoß optimiert wurde. Da Ohno nicht über unendliches Budget und Massenaufträge verfügte, musste er Wege finden, wie die Durchlaufzeit der Autoproduktion reduziert werden konnte:

- Toyota hatte weder Lagerplatz noch Geld. Deshalb war klar, dass immer nur so viele Teile produziert werden würden, wie gerade gebraucht wurden = **Pull-Prinzip**.
- bei den kleinen Chargen fielen Fehler direkt auf und konnten sofort behoben werden.

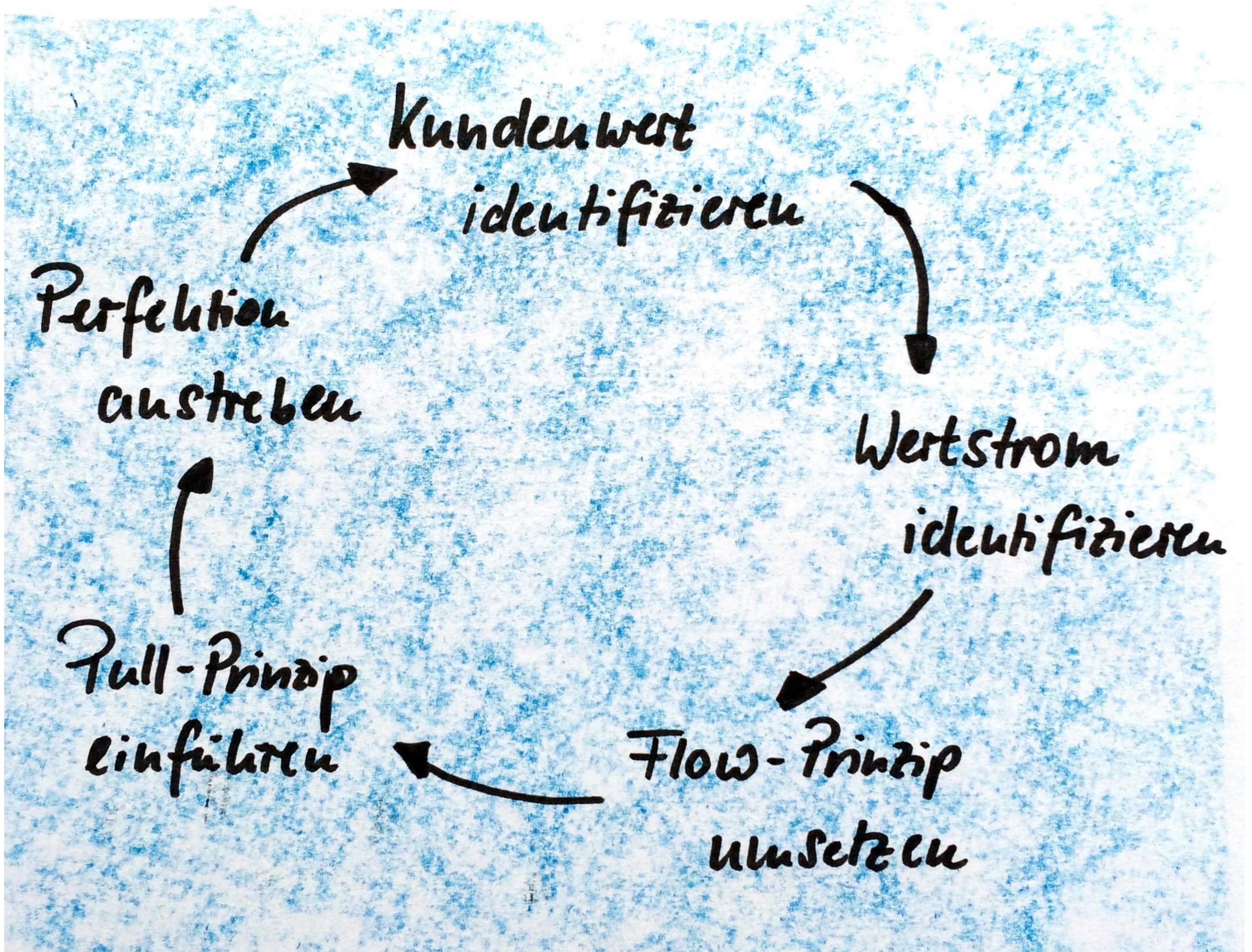
- Das Pull-Prinzip erlaubte auch, in kleinen Chargen zu arbeiten, weil immer nur das produziert oder eingekauft wurde, was gerade benötigt wurde. Da Ohno nicht das Budget hatte, für jedes Modell eine eigene Maschine zu kaufen, arbeitete er mit austauschbaren Stanzen, die jeweils für ein paar Stunden am Tag liefen, um die Bestände aufzufüllen und nicht wie bei Ford über Monate, um die Maschine sinnvoll auszulasten.
- In der Autostraße installierte Ohno über jedem Arbeitsplatz eine Kordel, die das Fließband stoppte. So konnte jeder Arbeiter das Band stoppen, wenn er einen Fehler entdeckte, den er nicht alleine beheben konnte. Dann kam das ganze Team zusammen und behob den Fehler gemeinsam, bevor das Band weiterlief = **Stop and fix und Respect for people**
- Außerdem wollte Ohno verhindern, dass Fehler sich wiederholten. Er brachte allen Arbeitern bei, jeden Fehler zu seinem Ursprung zurückzuverfolgen und ihn an der Quelle zu beheben = **genchi genbutsu und Continuous Improvement**
- Als Konsequenz brauchte es fast keine Nachbearbeitung der Autos, sobald sie das Band verließen (zum Vergleich: Auch heute noch gehen in Unternehmen ohne Lean Production ca. 25% des Platzes und der Zeit für Nacharbeiten drauf).

Die USA unterschätzten Toyota bis in die 60er Jahre. In den 70ern nach der Ölkrise stieg die Nachfrage nach kleinen, kraftstoffsparenden Autos, die Toyota gut bedienen konnte. Außerdem hatten sie durch Lean Production auch die Nase vorn was Qualität und Produktivität betraf, so dass sie die Autos weiterhin günstig anbieten konnten. Erst in den 80ern fingen dann andere Automobilfirmen an, die Produktionsmethoden von Toyota zu kopieren und konnten deshalb Toyotas Preisvorsprung verringern. 1990 war Toyota halb so groß wie General Motors und zwei Drittel so groß wie Ford. Mittlerweile ist Toyota das größte und langfristig erfolgreichste Industrieunternehmen der Welt.

Lean - Kernprinzipien

2

Es werden 5 übergeordnete Kernprinzipien in Lean Management definiert.



Lean - respect for people

2

Teamentwicklung

- effektive Teams, die funktionieren
- Teamwork
- Teams entwickeln eine eigene Kultur

Management agiert als Vorbild

- Management versteht und verhält sich gemäß dem Ziel, Verschwendung zu vermeiden und sich kontinuierlich zu verbessern
- Mitarbeiter erkennen dies in den Entscheidungen und dem Verhalten

Teams und Individuen schaffen ihre eigene Arbeitskultur

- Management fördert bei Mitarbeitern das Bestreben, sich kontinuierlich zu verbessern
- Mitarbeiter lernen Probleme zu lösen und sich selbst und ihre Fähigkeiten zu reflektieren
- Mitarbeiter entscheiden selber, wie (in welchen Bereichen und mit welchen Herangehensweisen) sie sich verbessern

RESPECT FOR PEOPLE

Beziehung zu Partnern

- nachhaltige und vertrauensvolle Kundenbeziehungen schaffen
- Partner bei Verbesserungen unterstützen

zuerst Weiterentwicklung der Mitarbeiter, dann Produktentwicklung

- Manager agieren als Coaches, nicht als Vorgesetzte
- begleite Mitarbeiter eng beim Ausbau ihrer fachlichen und Problemlösungskompetenzen
- bringe Mitarbeitern bei, Probleme sichtbar zu machen, ihre tatsächlichen Ursachen zu identifizieren und danach Lösungen zu entwickeln

Hilf deinen Kunden ihren Fokus zu behalten

- Kunde = jeder, der deine Arbeit und Entscheidungen nutzt
- lass sie nicht warten
- überlade sie nicht
- verführe sie nicht zu unrealistischen Erwartungen
- liefere keine fehlerhaften Produktinkremente
- analysiere und verändere dich und deine Arbeit stetig, um dem Kunden Freiraum zu ermöglichen

Kaizen

= kontinuierliche
Verbesserung

Go See

= genchi genbutsu = Fehler
zu ihren Ursprüngen zurück
verfolgen und sie an der
Quelle ihres Ursprungs
beheben

= gemba walk = an den Ort
des Geschehens gehen (z.B.
Manager gehen in die
Produktion)

challenge

= stetiges Hinterfragen von
Dingen, die noch besser
gemacht werden können

= hohe Ansprüche an sich
selbst, die Teams und
Partner zu haben

Lean - Muda, Muri, Mura

2

Ein Kernbestandteil von Lean ist die strenge Fokussierung auf **wertschöpfende Prozesse**. Der Wertschöpfung gegenüber steht die **Verschwendung (Waste)**, die wir möglichst vermeiden wollen. Im Rahmen von Lean unterscheiden wir zwischen verschiedenen Arten der Verschwendung, den **3Ms**:

Muda: Umfasst verschiedene Arten allgemeiner Verschwendung

Muri: Überlastung von Mensch und Maschine

Mura: Unausgeglichenheit

Innerhalb von *Muda* lassen sich verschiedene Arten der Verschwendung erkennen. Wir nennen zuerst die ursprüngliche Formulierung aus der Produktion und in Klammern dahinter deren Entsprechung im Rahmen der Software-Entwicklung:

- **Herstellung fehlerhafter Teile (Bugs / Defects)**
- **Überhöhte Lagerhaltung (Partially done work)**
- **Überproduktion (Extra features)**
- **Wartezeit (Delays)**
- **Unangemessener Herstellungsprozess (Extra processing, Relearning, Reinventing)**
- **Unnötige Bewegungen (Task switching)**
- **Überflüssiger Transport (Handoffs)**

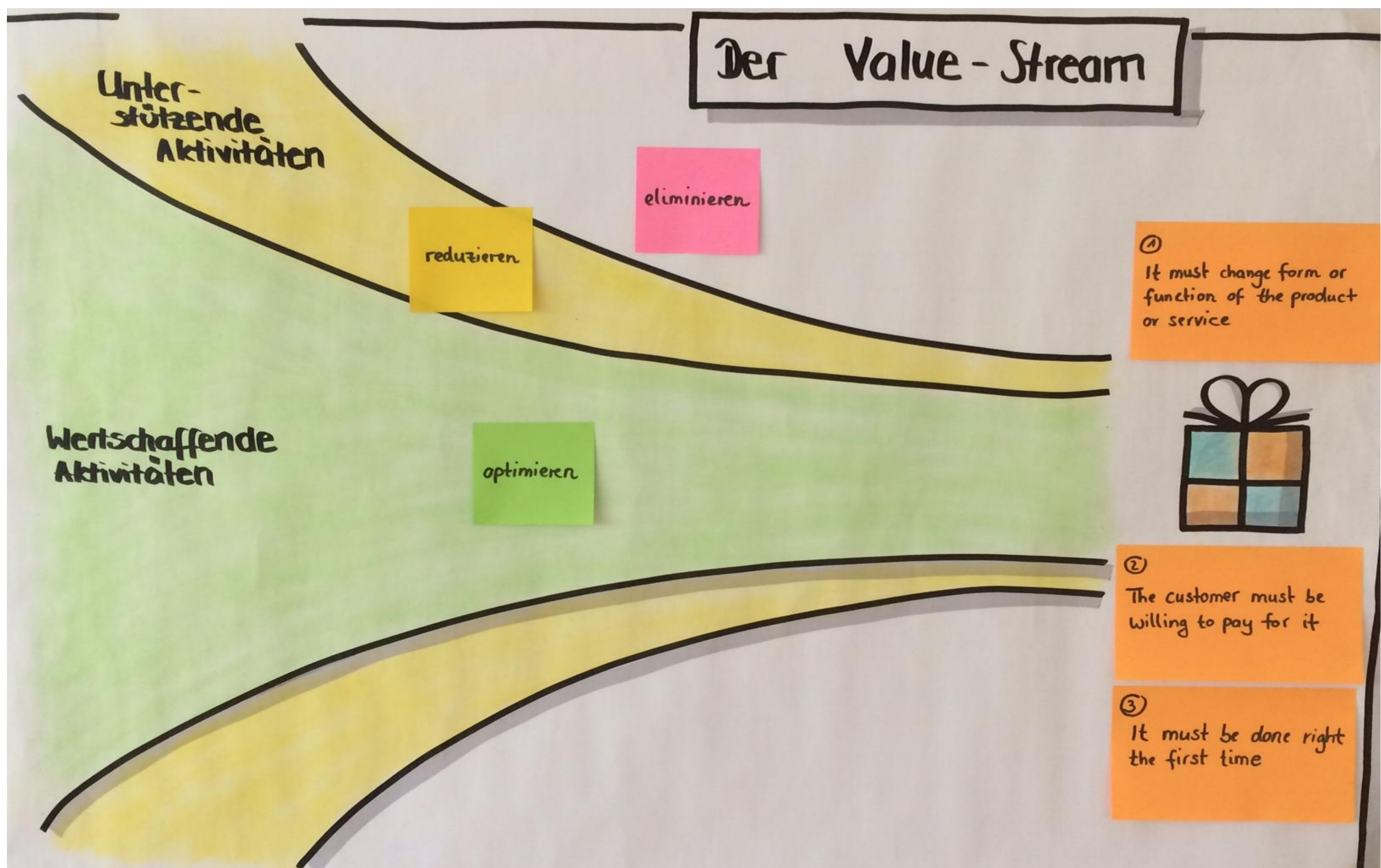
Lean - value

2

Damit im Lean Management etwas (Arbeit, Prozesse...) als **Mehrwert** oder **wertschöpfend** für den Kunden betrachtet wird, müssen folgende Kriterien erfüllt sein:

- Die Form oder Funktion des Produktes oder des Services muss geändert werden.
- Der Kunde ist bereit für die Änderung zu bezahlen.
- Es muss von Anfang an richtig gemacht werden.

Die Konzentration auf wertschöpfende Prozesse unterstützt damit die Ausrichtung auf Kundenbedürfnisse und hilft Verschwendung (waste) zu vermeiden.



Die agilen Werte bilden das Fundament agiler Softwareentwicklung. Die agilen Prinzipien basieren auf den agilen Werten und bilden Handlungsgrundsätze. Im Jahr 2001 wurden Werte und Prinzipien als **Agiles Manifest** (englisch *Manifesto for Agile Software Development* oder kurz *Agile Manifesto*) von mehreren Software-Entwicklern formuliert und unterzeichnet.

“Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen.

Durch diese Tätigkeit haben wir diese **Werte** zu schätzen gelernt:

- **Individuen und Interaktionen** mehr als Prozesse und Werkzeuge
- **Funktionierende Software** mehr als umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung
- **Reagieren auf Veränderung** mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.”

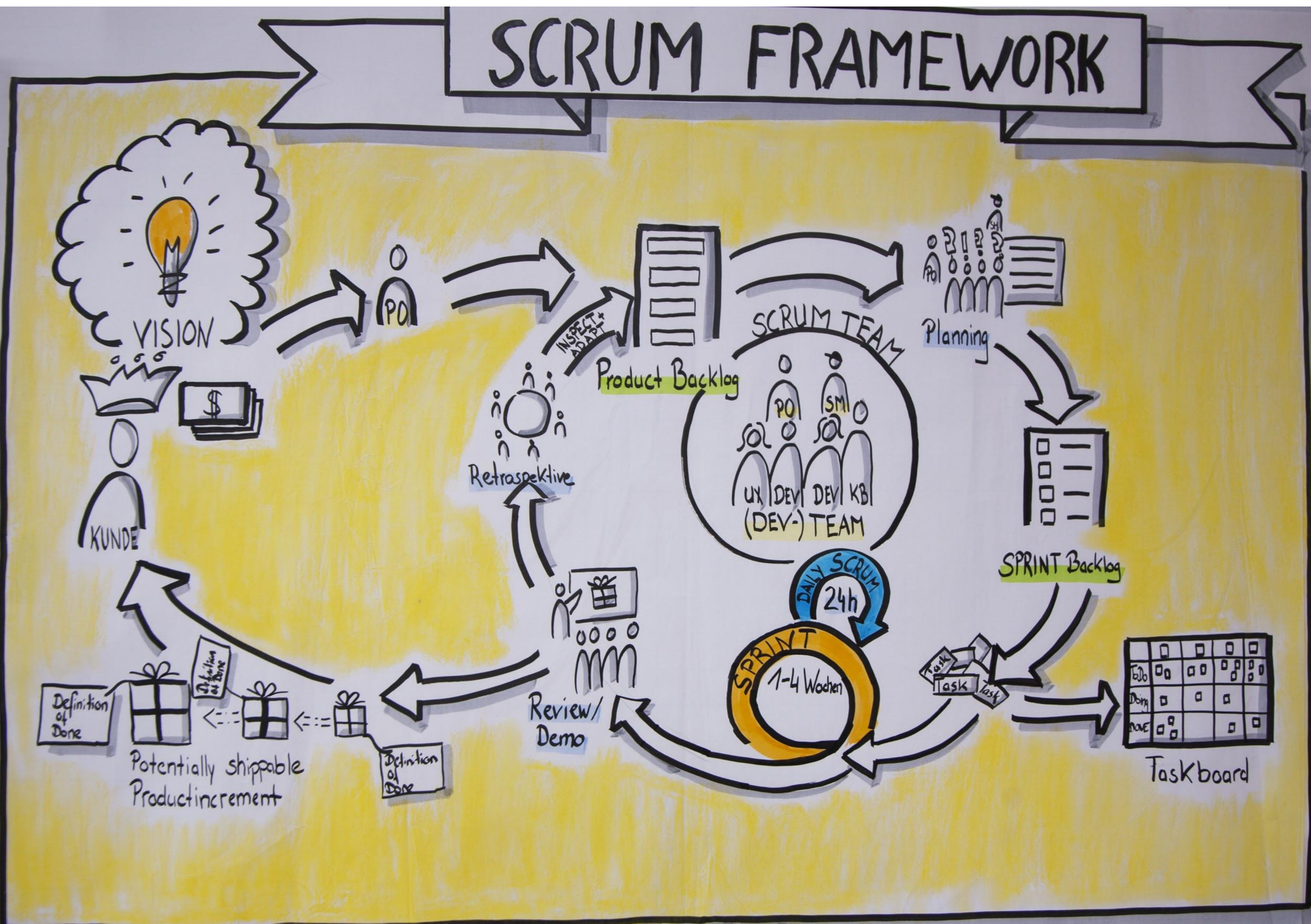
Hinter dem agilen Manifest stehen **12 Prinzipien** der Softwareentwicklung, die Du auf der folgenden Seiten findest.

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit -- die Kunst, die Menge nicht getaner Arbeit zu maximieren -- ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

Scrum - Prozess

4

Jedes Teammitglied ist in den **Scrum Prozess (Sprint)** involviert. "Das Herz von Scrum ist der Sprint, eine Time Box von maximal einem Monat, innerhalb dessen ein fertiges ("Done"), nutzbares und potenziell auslieferbares Produkt-Inkrement hergestellt wird. Alle Sprints innerhalb eines Entwicklungsvorhabens sollten die gleiche Dauer haben. Der neue Sprint startet sofort nach dem Abschluss des vorherigen Sprints (The Scrum Guide, 2013)."



Das sich selbstorganisierende Scrum Team besteht i. d. R. aus 5 bis 9 Personen. Es ist interdisziplinär aufgestellt, d. h. jede Kompetenz, welche zur Erreichung des Sprint-Ziels notwendig ist, ist im Team vorhanden (z. B. Software-Entwickler, User Experience Designer, Texter, Kundenbetreuer). Jedem Teammitglied wird eine eigene **Rolle** zugeschrieben:

Product Owner

Er verantwortet den wirtschaftlichen Erfolg des Produkts. Er erstellt, beschreibt und priorisiert die zu entwickelnden Produkteigenschaften/-anforderungen in Form von sogenannten User Stories, welche sich im Product Backlog als entsprechende Einträge wiederfinden.

Entwicklungsteam

Das interdisziplinär aufgestellte Entwicklungsteam erstellt das Produkt-Inkrement, in dem sie die User Stories aus dem Sprint Backlog umsetzen. Wie das Team dies tut - sprich welche Methode, Tools, Prozesse, etc. es dafür einsetzt - organisiert (entscheidet) es selbst.

Scrum Master

Er sorgt zunächst dafür, dass der Scrum Prozess und seine Regeln eingehalten werden. Er betreut das Team, moderiert die Scrum Meetings ergebnisorientiert und beseitigt Hindernisse (Impediments), welche das Sprint-Ziel gefährden.

Zu den **Scrum Artefakten** zählen das Product Backlog, Sprint Backlog sowie das Product Increment, welche Arbeit oder Wert repräsentieren. Sie dienen dazu, "Transparenz sowie Möglichkeiten zur Überprüfung zu schaffen". (The Scrum Guide, 2013)

Product Backlog

Das Product Backlog ist eine Liste, welche die geordneten Anforderungen an das Produkt enthält (z. B. Features, Funktionalitäten, Verbesserungen und Fehlerbehebungen). Es ist dynamisch und entwickelt sich mit dem Produkt. Für Inhalt, Zugriff und Reihenfolge der Einträge ist der Product Owner verantwortlich.

Sprint Backlog

Das Sprint Backlog ergibt sich aus der Menge der Product Backlog-Einträge, welche als Aufgaben im nächsten Sprint umgesetzt werden sollen.

Product Increment

Ein Inkrement (*auch: Produkt-Inkrement*) ist das Ergebnis aus allen in einem Sprint fertiggestellten Product Backlog-Einträgen und dem Resultat der Inkremente aller früheren Sprints. Am Ende des Sprints muss das neue Produkt-Inkrement nutzbar sein und somit die definition of done erfüllen.

Während des Sprints finden vorgeschriebene **Meetings** statt, um eine Regelmäßigkeit herzustellen. Alle Meetings sind zeitlich auf eine Maximaldauer begrenzt (Time Box).

Sprint Planning

Dient dazu, die Aufgaben gemeinschaftlich im Scrum Team für den nächsten Sprint zu planen.

- *Was kann im kommenden Sprint fertig gestellt werden?*
- *Wie sollen die ausgewählten Aufgaben erledigt werden?*

Daily Scrum (auch: Daily StandUp)

Das Daily Scrum findet täglich statt und dauert 15 Minuten. Das Team synchronisiert mit Hilfe folgender Fragen seine Aktivitäten:

- *Was habe ich seit dem letzten Daily StandUp getan?*
- *Was möchte ich bis zum nächsten Daily StandUp tun?*
- *Mit welchen Hindernissen (Impediments) wurde ich konfrontiert?*

Sprint Review (auch: Demo)

Das Sprint Review erfolgt am Ende eines Sprints, in dem das Team dem Product Owner sowie weiteren Stakeholdern die entwickelten Funktionalitäten präsentiert.

Sprint Retrospektive (auch: Retro)

Diese findet ebenfalls nach jedem Sprint statt und bietet dem Team die Möglichkeit sich zu überprüfen und sich im kommenden Sprint zu verbessern.

- *Was ist gut gelaufen (was haben wir gelernt)?*
- *Was können/ müssen wir verbessern?*

Quellen und Literaturtipps

Quellen

Lean - Die Toyota Story

- Jeffrey K. Liker: The Toyota Way

Lean - Kernprinzipien

- wikipedia.org - Lean Management

Lean - respect for people und Lean - continuous improvement

- Craig Larman und Bas Vodde: Scaling Lean & Agile

Lean - Muda, Muri, Mura

- Mary und Tom Poppendieck: Lean Software Development: An Agile Toolkit

Agiles Manifest und Agile Prinzipien

- wikipedia.org
- www.agilemanifesto.org

Scrum

- The Scrum Guide, 2013

zusätzliche Literaturtipps

- Damir Prusac: On the Trace of Agile Leadership
- Kenneth S. Rubin: Essential Scrum: Umfassendes Scrum-Wissen aus der Praxis