

Guten Morgen :-)

Hallo zusammen und guten Morgen :-)

Buzzword-Bingo

Beim nachträglichen Durchgucken des Talks habe ich festgestellt, dass jeder hier bequem praktisch jedes IT-Buzzword-Bingo gewinnen kann. Es gibt unendlich viele Begriffe, die man auch im Harvard Business Manager oder im CIO-Inflight-Magazin der Wahl findet.

Chief Tailwind Officer

Aber ich darf das auch, weil ich bin ja inzwischen in der Managementebene nicht nur angekommen, sondern auch wieder abgehauen. Warum mein offizieller Titel Chief Tailwind Officer - also Hauptverantwortlicher für Rückenwind auf deutsch - ist und nicht mehr Chief Technology Officer, das ergibt sich im Talk. Ich arbeite bei Mayflower, das sind etwa 80 Leute in der Softwareentwicklung, und wir haben noch 3 Windows-Rechner in der Firma, das bekommen wir noch in den Griff.

Unix-Admin Security OpenSource

Mir ist völlig klar, dass ich damit maximal inkompetent wirke. Ich möchte zu meiner Verteidigung anbringen, dass ich tatsächlich auch schon als Unix-Admin gearbeitet habe - wenn auch im Studium - dass ich ein paar Sachen im Bereich Security gemacht habe (und es hier in Köln noch eine Firma aus der Zeit gibt), und auch Sourcecode und Patches in ein paar OpenSource-Tools zu finden ist.

Wer von euch ist Unix-Admin oder in dem Bereich tätig? Wer macht Security? Wer hat schon zu OpenSource contributed? Uni? Firma? Selbstständig?

Bei uns gibt es ebenfalls eine zentrale Systemadministration wie einen Haufen System Engineers in den Teams. Der meiste Kram bei uns ist Puppet-basiert, wenn auch Puppet 3 - und wir migrieren im Moment einen Stapel Infrastruktur nach NixOs, und das macht tatsächlich viel Spass

Was passiert, wenn man
**„The Cathedral and the
Bazaar“**
auf Firmen anwendet.

Und auch wenn endlos Buzzwords und Management-Begriffe passieren werden, der Talk könnte auch so heißen: was passiert, wenn man die Einsichten aus Eric S Raymonds Talk „The cathedral and the bazaar“ auf Firmen anwendet. Das ist ganz spannend, denn offensichtlich sorgt genau der Opensource- und Unix-Background dafür, dass einem Organisationsdesign-Themen in der vernetzten Welt logisch und plausibel vorkommen. Tatsächlich kommen die meisten von denen von Software-Guys.



Er heisst aber: „Von Kutschern, Schriftsetzern, Managern und SysAdmins“. Das ist eine chronologische Reihenfolge. Und eine billige Provokation :-).



Bundesarchiv, Bild 183-1990-1128-500
Foto: o. Ang. | 1904

Auf dem Bild im Hintergrund von 1904 ist eine Kutsche und eine sogenannte Kraftdroschke zu sehen. Und man kann sich vorstellen, was der Kutscher dachte: „Einige von uns werden jetzt vermutlich mit diesen modernen Kutschen ohne Pferde unterwegs sein, aber ich glaube ja nicht, dass sich das durchsetzen wird.“

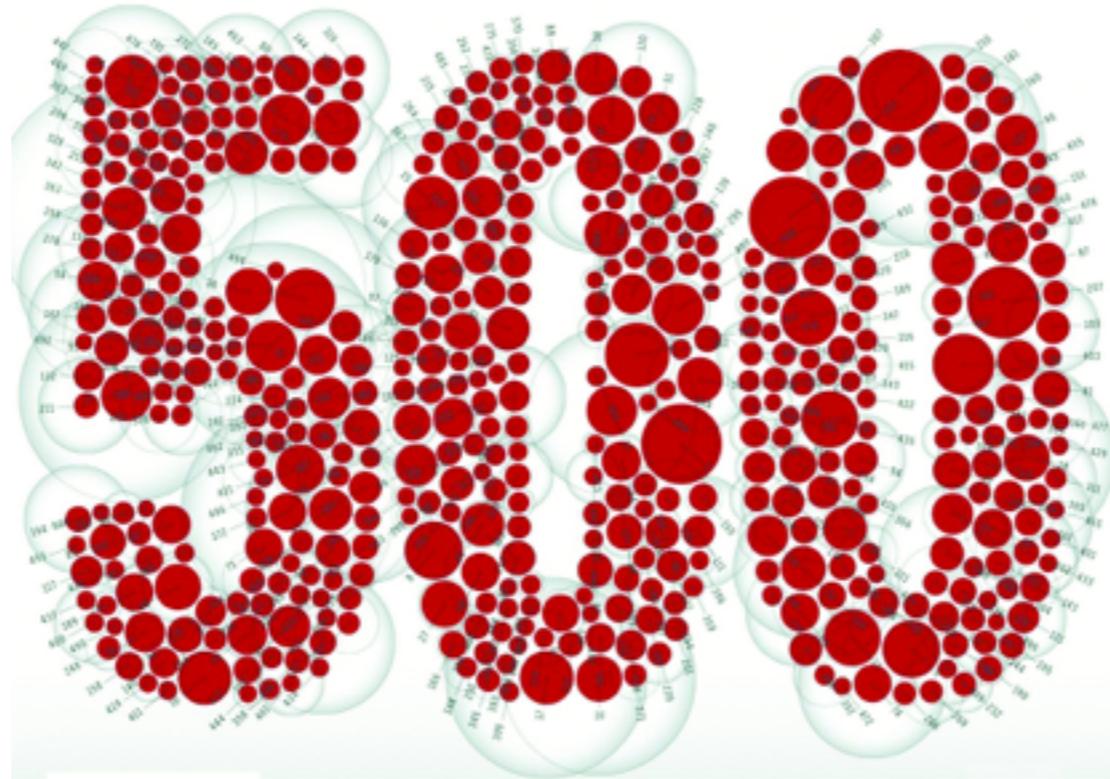


Heraklit von Ephesus, 500 v. Christus

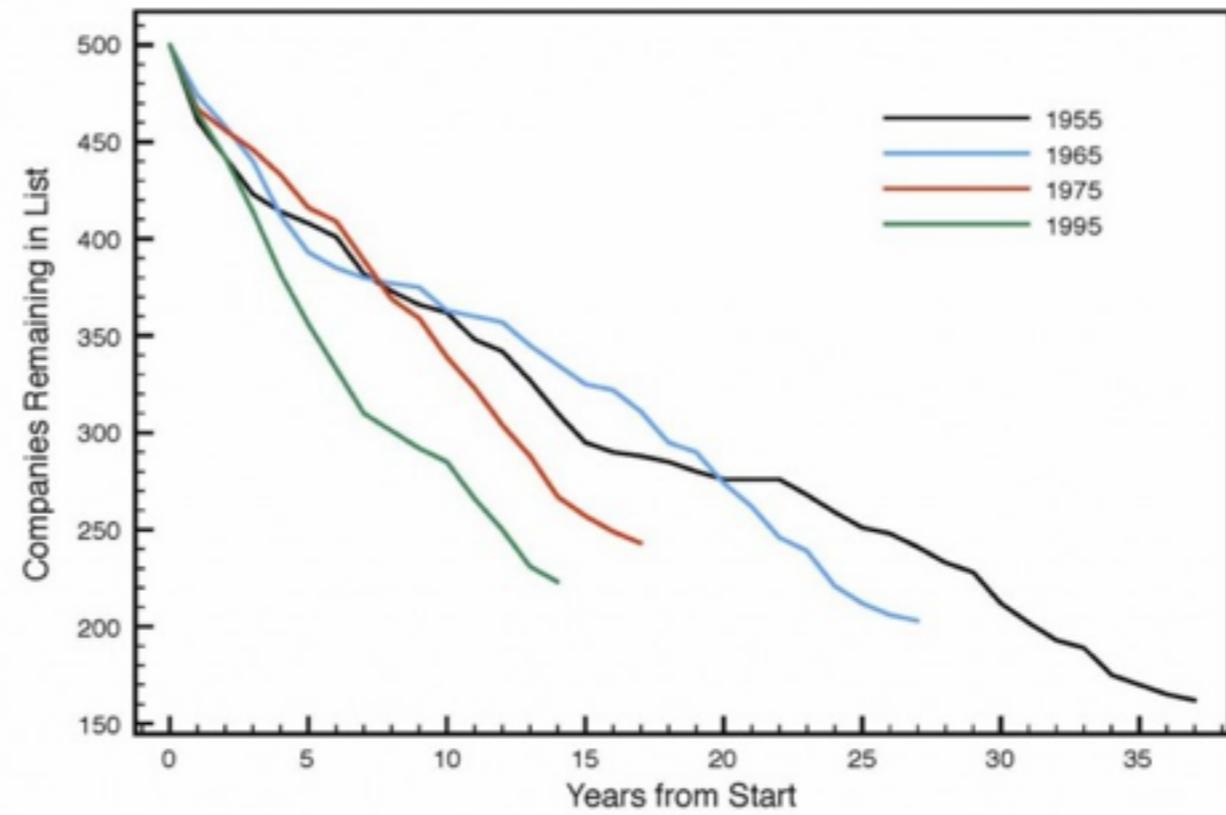
„Die
einzigste Konstante
im Universum ist
die **Veränderung**“

Klar, werden jetzt einige sagen, natürlich bewegen sich die Dinge. Das ist nicht neu. Das ist sogar mehr als 2500 Jahre alt, dieser Philosoph oben rechts - Heraklit von Ephesus - sagte das, als hätte er gewusst, was da noch so alles kommt. Klar, Veränderung gibt es immer, und Veränderung wird es immer geben. Aber mit der Veränderung selbst passiert etwas spannendes.

FORTUNE

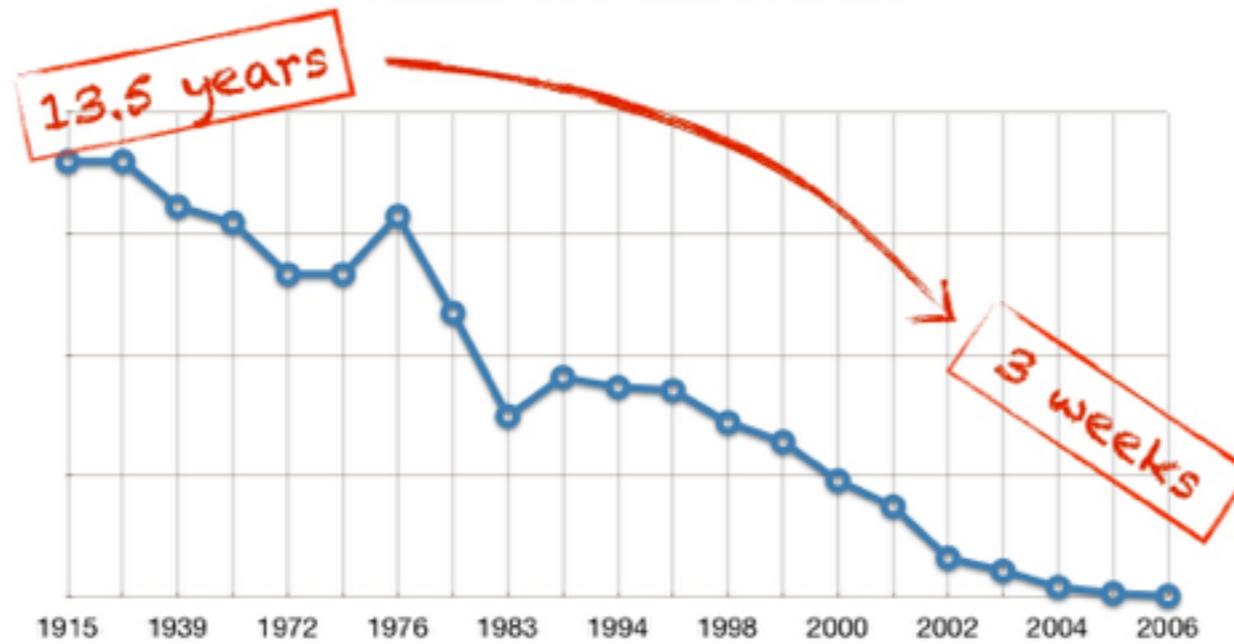


Ich denke, die meisten kennen die Fortune 500. Das ist die jährliche Liste von Forbes Magazin, die global Fortune 500. Die Liste der 500 umsatzstärksten Unternehmen der Welt. Zu dieser Liste gibt es den Fortune 500 Turnover, die Rate der Unternehmen, die in jedem Jahr dazukommen bzw. wieder verschwinden. Und da ist natürlich auch Bewegung drin, denn logischerweise waren dort Google, Facebook oder Ebay vor 20 Jahren noch nicht zu finden.



Das eigentlich spannende ist aber der Fortune 500 Turnover-Rate, also: wie viele Unternehmen kommen im Jahr dazu, und wie viele Unternehmen fliegen im Jahr raus, und wie lange waren sie dabei. Diese Grafik zeigt die Turnover-Rate - wie viele Unternehmen, die im Startjahr noch dabei waren, sind heute noch dabei. 1955 hat es noch mehr als 25 Jahre gedauert, bis die Hälfte verschwunden war - 1995 waren es nur noch 10 Jahre.

Time to Market



Deepa Chandrasekaran, Gerard J. Tellis - Marshall School of Business, University of Southern California, Los Angeles, California

Die Kollegen von agile42 haben das mal recherchiert und einen schönen Slide dazu - die Zeit von Produkt zu Markt ist von 13,5 Jahren vor 100 Jahren auf 3 Wochen runter.



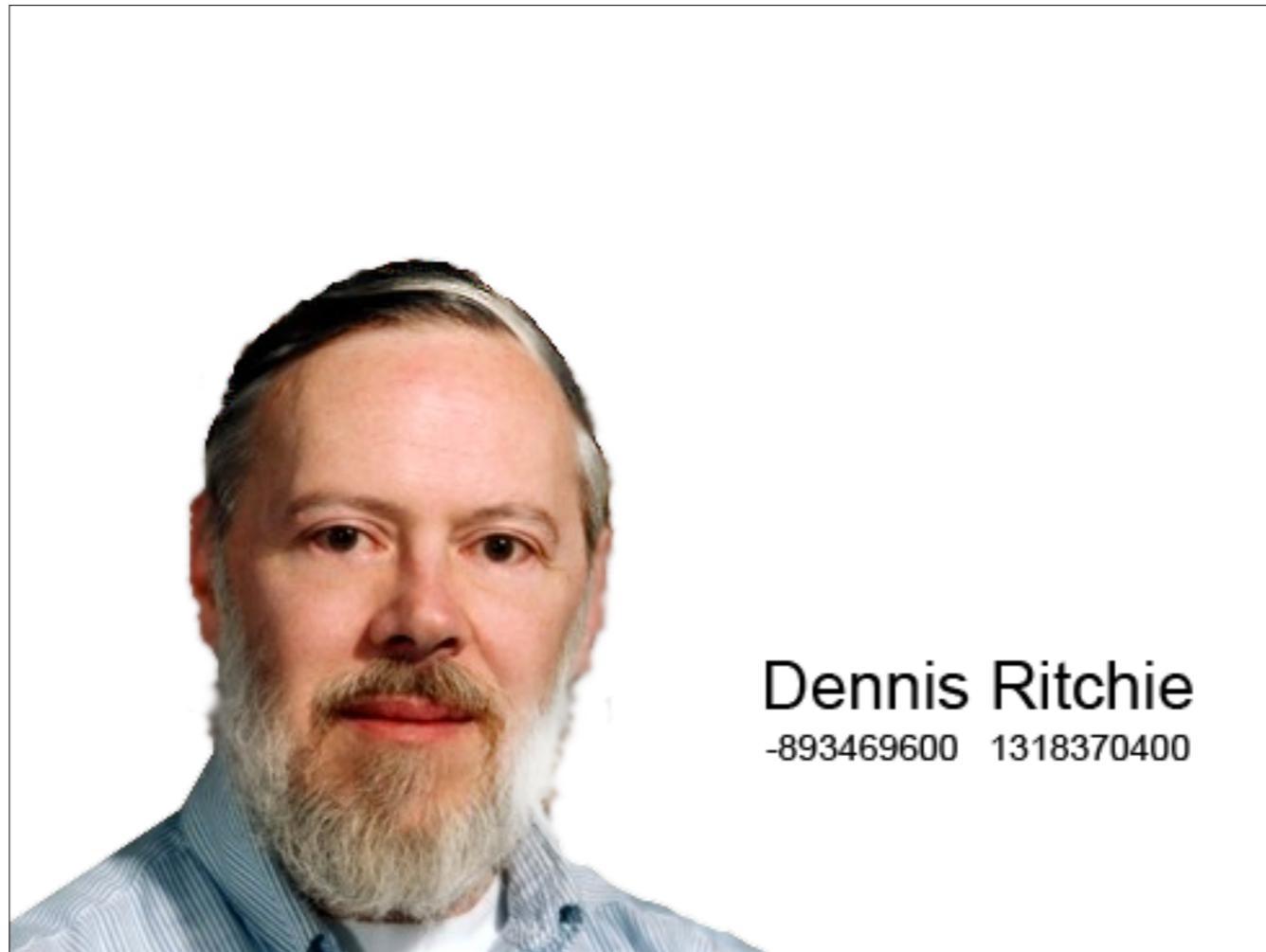
Bei Rocket Internet ist ein Startup nach 2 Monaten online, und es kann nach 3 Monaten schon die Entscheidung gefällt werden, ob das Startup eingestellt wird.

Und wer ist
schuld daran,
dass es jetzt
immer schneller
wird?

Warum beschleunigt sich das immer mehr?



Steve Jobs? Wenn wir hier auf einer Was-mit-Medien oder auf einer Vordenker-und-andere-Blogger-Veranstaltung wären, dann würde man ihm vermutlich die Schuld geben.



Da sind wir aber nicht, sondern bei der GUUG - und die meisten der Leute von hier waren mit dabei, als es wirklich passierte. Dieser Herr hat zunächst mit Ken Thompson zusammen Unix in der ersten Version gebaut, und später haben die beiden dann mit Brian Wilson Kernighan zusammen die passende Programmiersprache entwickelt, damit man die Entwicklung von Unix-Software auch humanoiden zumuten kann.

Reklame für Internetmarketing 1996:

24 Stunden
7 Tage
Weltweit

Wer, wie vermutlich die meisten Leute hier, 1996 schon dabei war, der kann sich noch an den Slogan erinnern. Das war damals ein grosses Ding, und damit hat man Internet an Firmen verkauft.

„Wir können uns **die**
Verkäufer sparen“

Die geheime Hoffnung der Firmen war natürlich, dass man damit Vertrieb spart, und man mehr und preiswerter verkaufen kann.

„Ich **kenne** die **Preise**
aller Deiner
Konkurrenten.“

Faktisch ist es dann aber doch noch deutlich anders gekommen als man dachte. Man konnte nämlich nicht nur selbst 24 Stunden am Tag und 7 Tage die Woche online sein, alle anderen konnten das auch.



Das gilt natürlich auch für die Konkurrenten in China - jetzt kann man, bevor man sich an den Zwischenhändler wendet, auch direkt an den Hersteller wenden.



Das ganze erzeugt deutlich schneller Geschäftsprozesse - Markttransparenz, Vergleichbarkeit und Alternativen sind auf einmal für jeden da. Das gilt für alle Produkte, die man verkaufen kann - aber auch für Geld oder Dienstleistungen und einfache Aufgaben bishin zum Crowdsourcing von grossen Aufgaben.

Es bleibt **nicht
mehr geheim,**
wenn jemand das
**gleiche besser/
billiger macht**

Das hat auf die Gesamtökonomie eine ganze Reihe von Folgen. Während ich mich vorher mit langfristigen Kundenbeziehungen, guten Vertrieblern und einer gute Marke halten konnte, bin ich deutlich substituierbarer geworden.

Schlimmer:
sie nutzen dafür
Computer!



Und nicht nur das - viele Dinge lassen sich besser digital machen. Deshalb fallen ganze Branchen weg.

Digitale Transformation

Vernetzt
Automatisiert
Global

Wir leben gerade in interessanten Zeiten, die grossen Unternehmensberatungen haben es inzwischen auch bemerkt und ihm einen Namen gegeben - nämlich digitale Transformation. Während jahrhundertlang alles ziemlich gut funktioniert hat, kommen wir jetzt um die Ecke und ändern es alles - in dem wir es vernetzen, automatisieren, und natürlich gleich weltweit und rund um die Uhr, ihr kennt das.

Software eats the world

„Meine eigene Theorie ist, dass wir in der Mitte einer **dramatischen** und **breiten technologischen** und **ökonomischen** Veränderung sind, in der **Softwarefirmen** bereit sind, grosse Teile der Wirtschaft zu **übernehmen.**“



Marc Andressen - Gründer von Netscape - hat das ganze 2011 in seinem Text „Software eats the world“ anders ausgedrückt - Meine eigene Theorie ist, dass wir in der Mitte einer dramatischen und breiten technologischen und ökonomischen Veränderung sind, in der Softwarefirmen bereits sind, grosse Teile der Wirtschaft zu übernehmen.

„**1** Software eats the world,
2 Every company becomes a
software company, and
3 Software people run every
company“



2014 hat er das noch einmal weitergeführt. Stimmt das?



Eines der Beispiele ist die Reisebranche - ein Markt, der kontinuierlich wächst, schliesslich haben wir immer mehr Freizeit, und parallel sinkt die Zahl der Reisebüros kontinuierlich - weil ein immer grösserer Teil des Geschäftes über das Internet läuft.

The logo consists of the text 'buchungsmaschine.de' in a white, lowercase, sans-serif font, underlined. The text is centered within a solid orange rectangular background.

Kennt jemand hier noch buchungsmaschine.de? Das war tatsächlich mal Deutschlands größter Anbieter von Reisen im Jahr 2000. 2006 wurde die Seite final geschlossen.

Was mache ich
da als **Unternehmen?**

Schnellere **Innovation**
Schnellere **Adaption**

Damit ich in so einer kontinuierlich schneller werdenden Welt weiter existieren kann brauche ich offensichtlich mehr - und vor allem schnellere - Innovation und Adaption, damit ich in der Lage bin auf die Innovation der anderen zu reagieren.



Der erste Versuch ist dann meistens schnellere Leute - es wird von der Unternehmensleitung kommuniziert, dass man mehr Innovation und mehr Adaption braucht, und es werden dementsprechende Metriken aufgestellt. In der Praxis bleiben die Menschen aber gleich schnell, da ist nicht so wirklich viel zu holen.

The fact is that the **system** that people work in and the **interaction** with people may **account** for **90** or **95 percent** of **performance**.

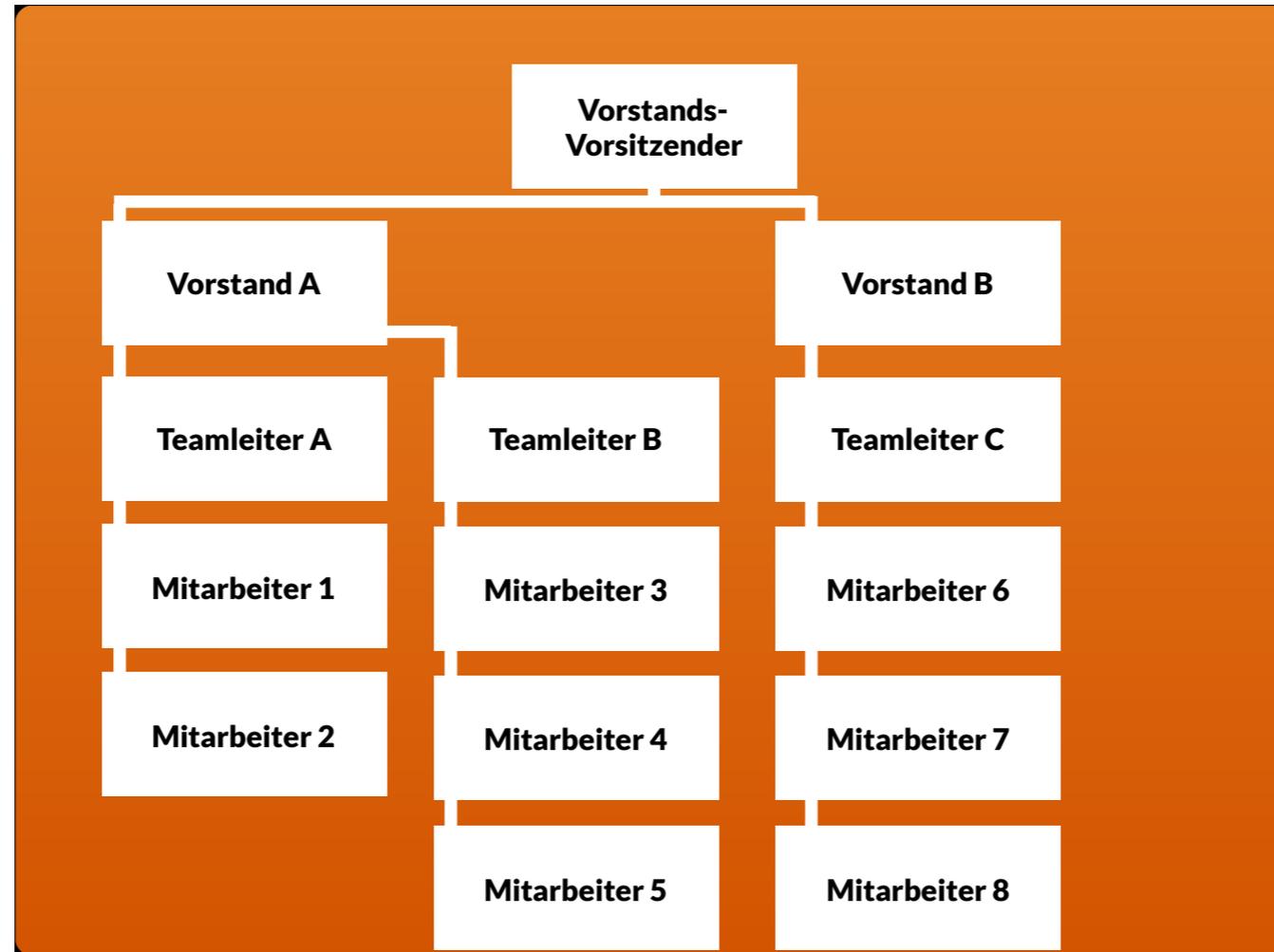
W. Edwards Deming



Traut man einem anderen Mitarbeiter des Bell Labs, William Edwards Deming, dann ist das ohnehin nicht der Ort, an dem die Geschwindigkeit wächst. Die entsteht vor allem im System selbst - also in der Interaktion innerhalb des Unternehmens selbst.

Ok, und wo
passiert da die
Performance?

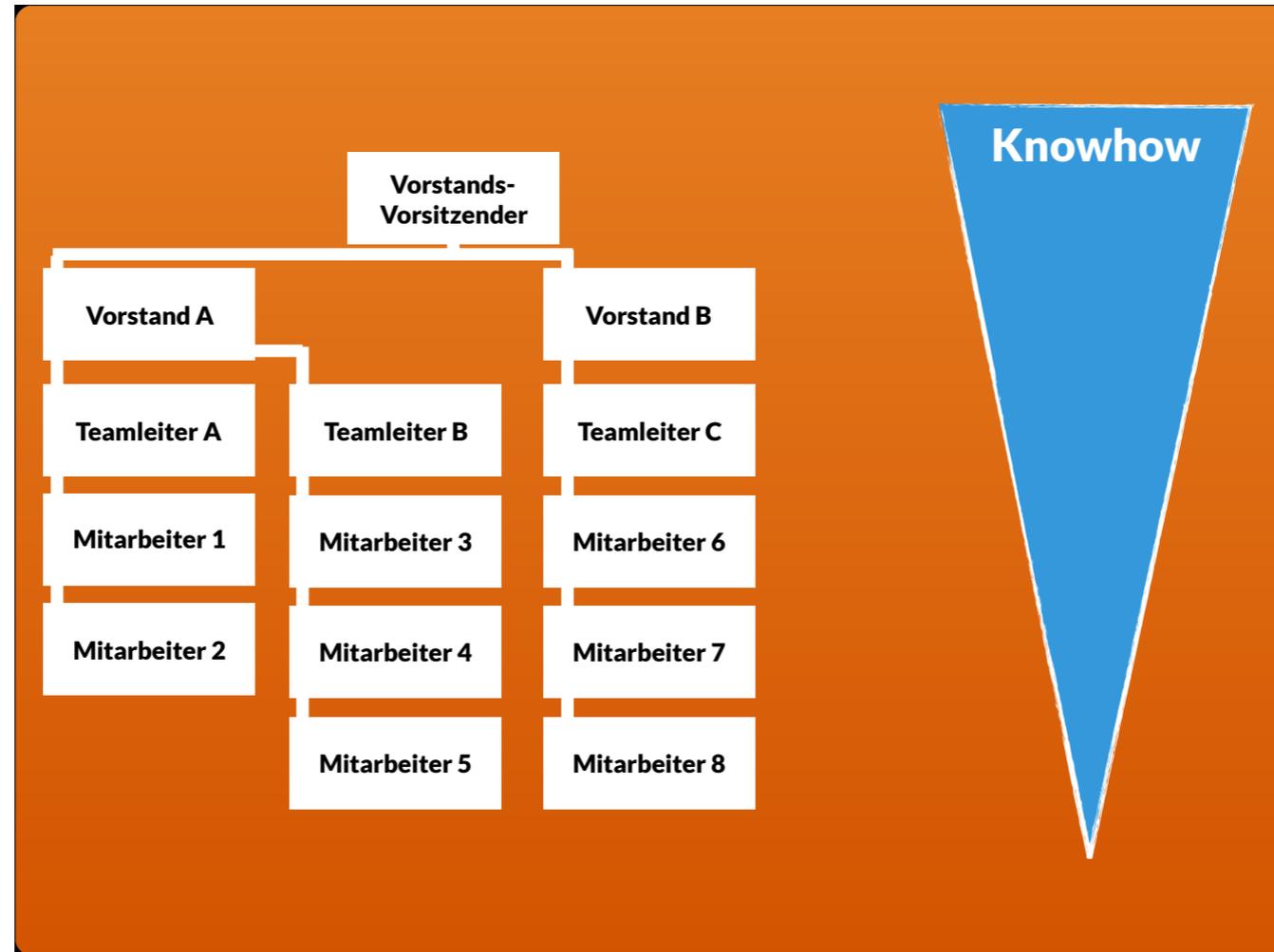
Ok, dann bin ich offensichtlich nicht als Individuum Schuld, wenn etwas katastrophal langsam läuft. Aber wo geht die Geschwindigkeit dann hin?



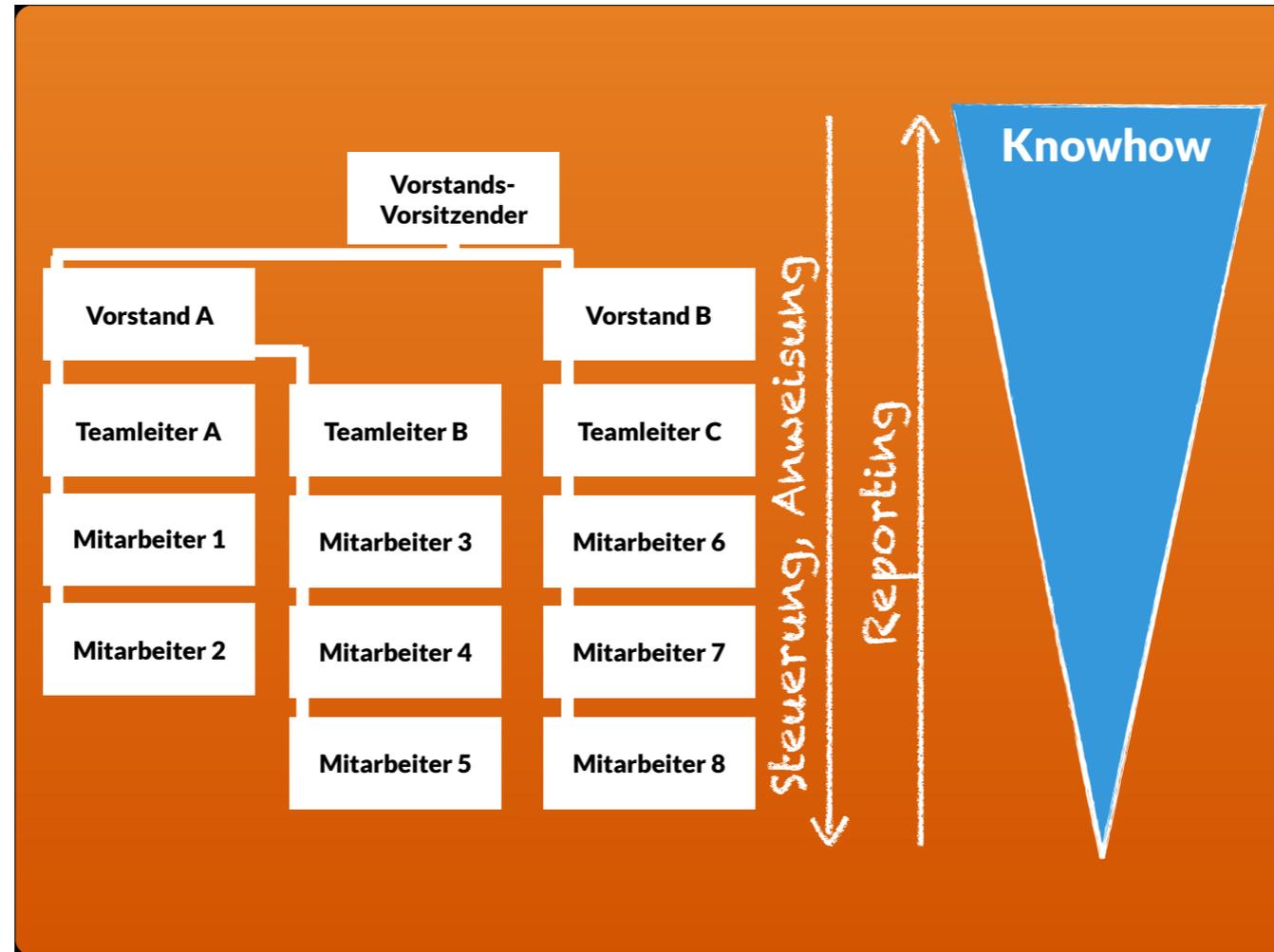
Wenn man nach der Gestalt von Organisationen fragt landet man meist in einem derartigem Diagramm - einem Organigramm.



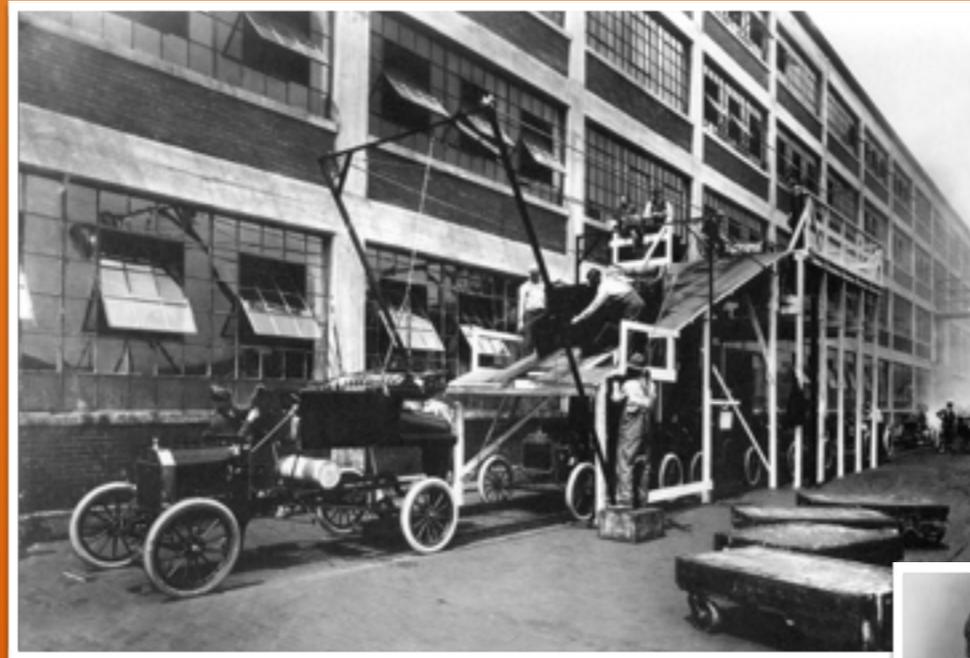
Die Grundidee ist schon 2000 Jahre alt und kommt vom Militär.



Die Grundidee dahinter ist, dass oben an der Spitze die Leute sitzen, die am meisten von der ganzen Sache verstehen. Sie kennen die meisten Hintergründe, haben am meisten Erfahrung und die nützlichste Form von Intelligenz.



Die Steuerung geht deshalb von oben nach unten - jede Hierarchiestufe weißt die darunter an. Sie weiß alles, was relevant ist, und gibt der Stufe darunter alle relevanten Informationen. Diese Stufe jeweils darunter reportet nach oben, so dass oben immer der Ort mit dem Größten Knowhow bleibt.

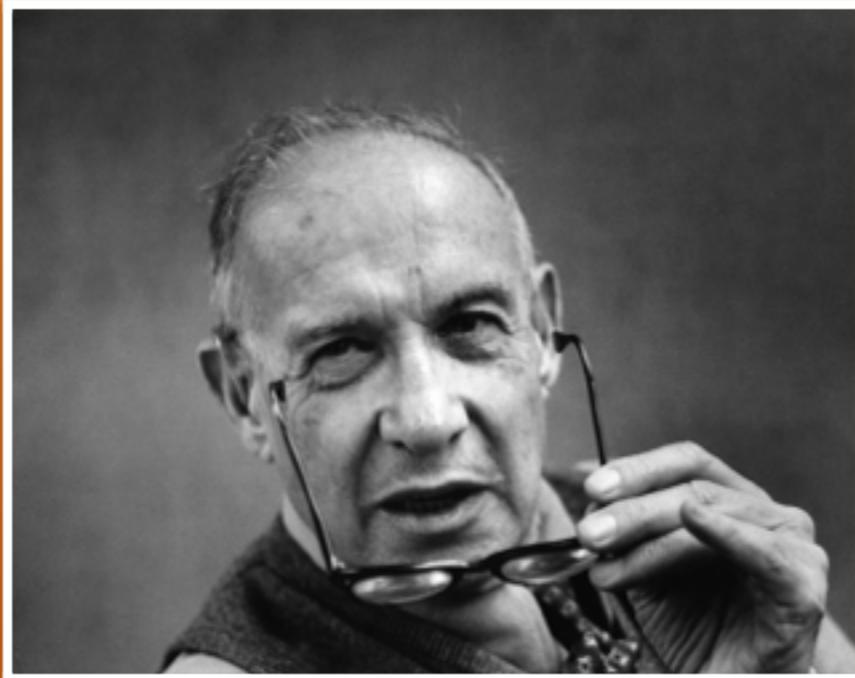


Das Prinzip funktioniert ziemlich gut, und war im Kontext der Industrialisierung das Erfolgskonzept schlechthin. Fließbandfertigung, wie damals bei Ford, ist quasi das Lehrbuchbeispiel dafür. Henry Ford hat selbst das Auto konstruiert - er hatte schon als 15jähriger seinen ersten Verbrennungsmotor konstruiert und hatte das Geld für die eigene Firma aus seiner Zeit als Chefindingenieur von Thomas Edison.

Bildquelle: https://de.wikipedia.org/wiki/Henry_Ford

Wessen Vorstand weiß noch alles?

Heute ist das nicht immer der Fall. Wer arbeitet bei einer Firma, bei der der Vorstandsvorsitzende noch ein Gesamtbild hat, mit dem er selbst das Produkt erzeugt?



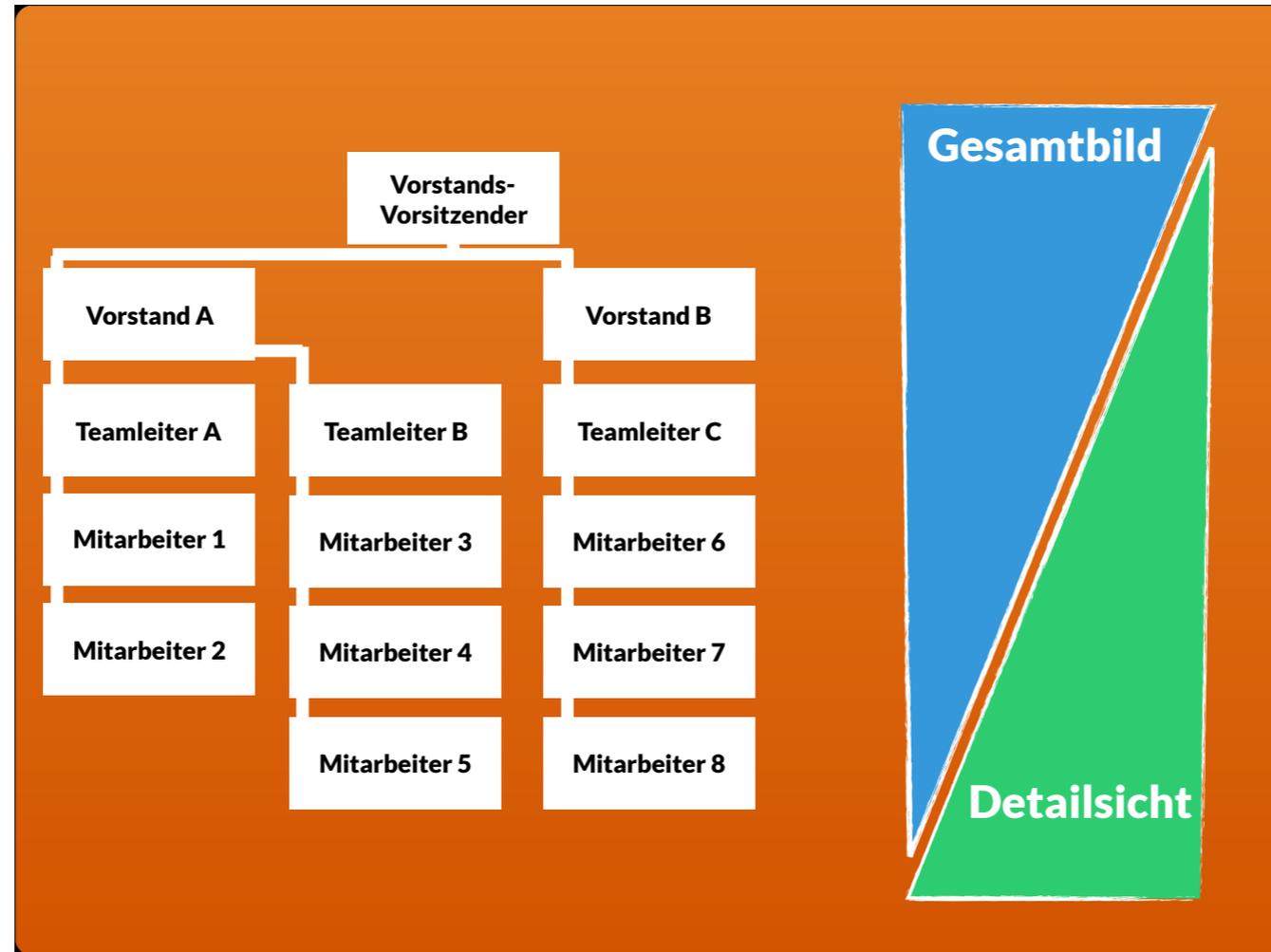
„Wissensarbeiter“

Genau, heute ist das nicht mehr der Normalfall - die Komplexität der Dinge ist deutlich zu hoch geworden, als dass ein Mensch alleine noch in der Lage wäre das vollständige Unternehmen zu erfassen, zumindest ab einer bestimmten Unternehmensgröße.

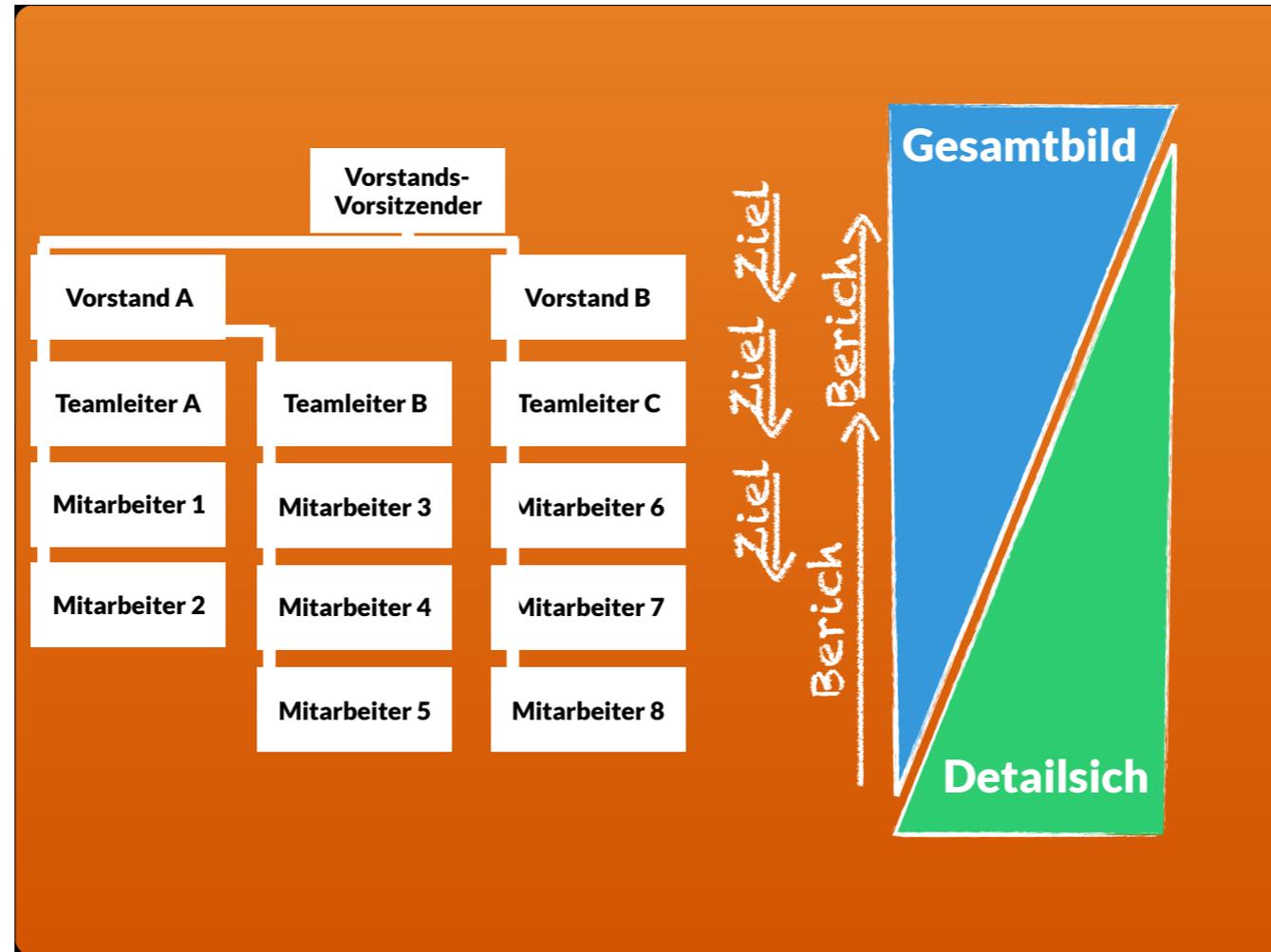
Der hier abgebildete Austroamerikaner Peter Drucker hat das analysiert - und das ganze Wissensarbeiter genannt. Der Wissensarbeiter arbeitet nicht mehr mit seiner körperlichen Kraft, sondern mit seinen geistigen Fähigkeiten.



Wir sind das, wir gehören dazu. Unser Schicksal ist es, dass unsere Chefs nicht mehr alle mehr wissen als wir. Und weder er noch wir wissen alles, was im Unternehmen relevant ist.



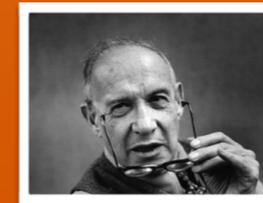
Dementsprechend kann unser Vorgesetzter gar nicht mehr alles steuern was wir machen. Das Detailwissen, das spezialisierte Knowhow müssen wir selbst einbringen. Das Gesamtbild passiert an der Konzernspitze, das Detailbild in den Abteilungen selbst - und bringt etwa die gleiche Menge an Komplexität mit.



Die Steuerung passiert dementsprechend nicht mehr in detaillierten Arbeitsanweisungen und Prozessen, sondern in mehr oder weniger komplexen Prozessen, die gemeinsam erstellt werden, und Ziele, die vorgegeben werden.

Management by Objectives

Zielvereinbarungsgespräche
SMARTe Ziele vereinbaren
Rahmenbedingungen schaffen
Bonus / Prämien



Die typische Form von Steuerung von Unternehmen mit Wissensarbeitern läuft heute über Management by Objectives, das ebenfalls der Herr Drucker gleich für sie entwickelt hat.

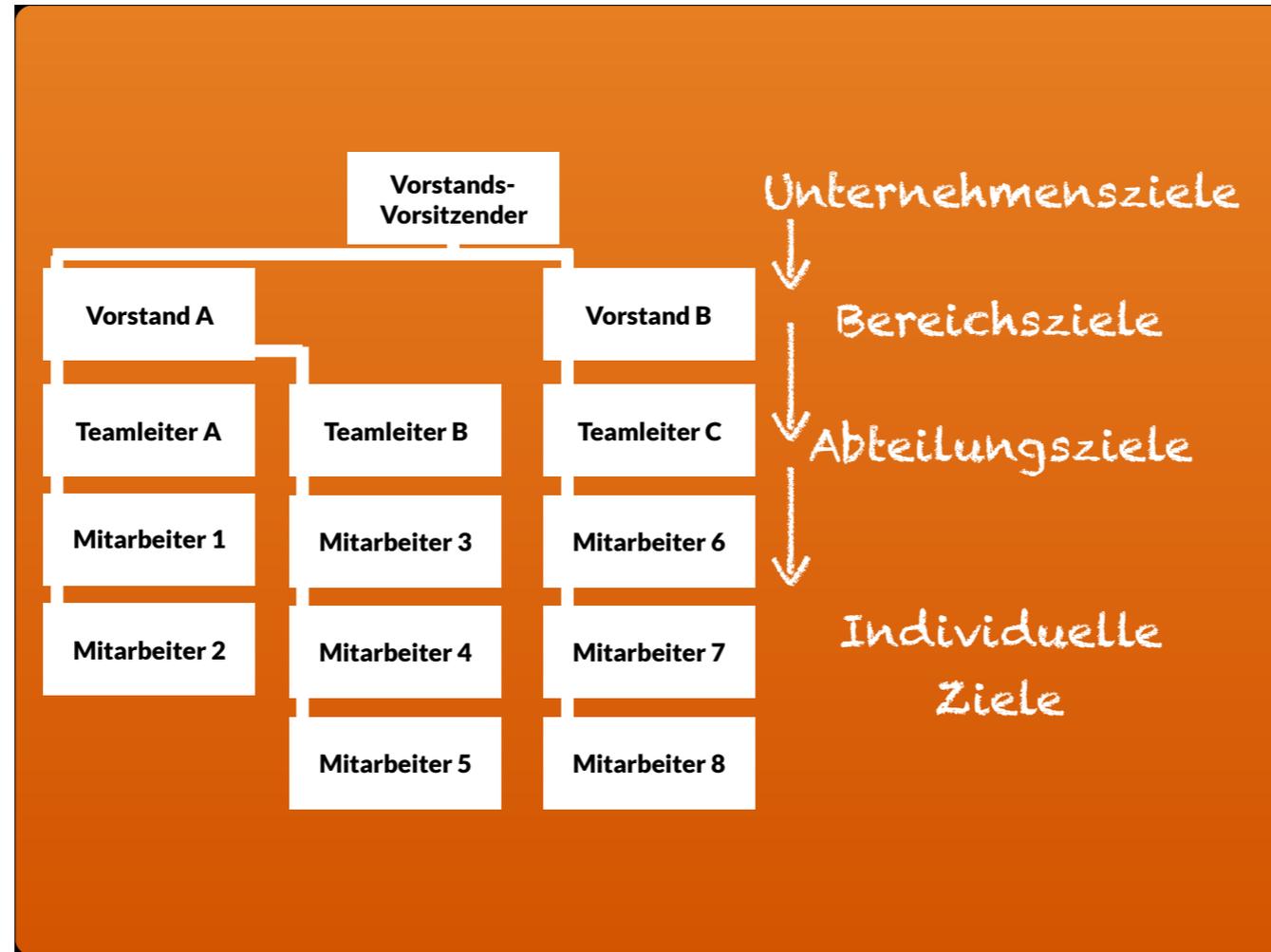
S – spezifisch (zur jeweiligen Abteilung),

M – messbar (klare Vorgaben),

A – aktiv beeinflussbar (erreichbar; auch: angemessen, attraktiv, akzeptiert oder anspruchsvoll),

R – realistisch (umsetzbar) und

T – terminiert (klares Zeitlimit).



Die Steuerung passiert dementsprechend nicht mehr in detaillierten Arbeitsanweisungen und Prozessen, sondern in mehr oder weniger komplexen Prozessen, die gemeinsam erstellt werden, und Ziele, die vorgegeben werden.
 Der Vorstand gibt die Unternehmensziele vor, aus denen ergeben sich die Ziele der Bereiche, die die der Abteilungen ergeben, die sich wiederum aus den individuellen Zielen zusammensetzen.

Leitungsaufgaben

Setzen der **Unternehmensziele**

Klärung von **Unterzielen**

Delegation von Verantwortung

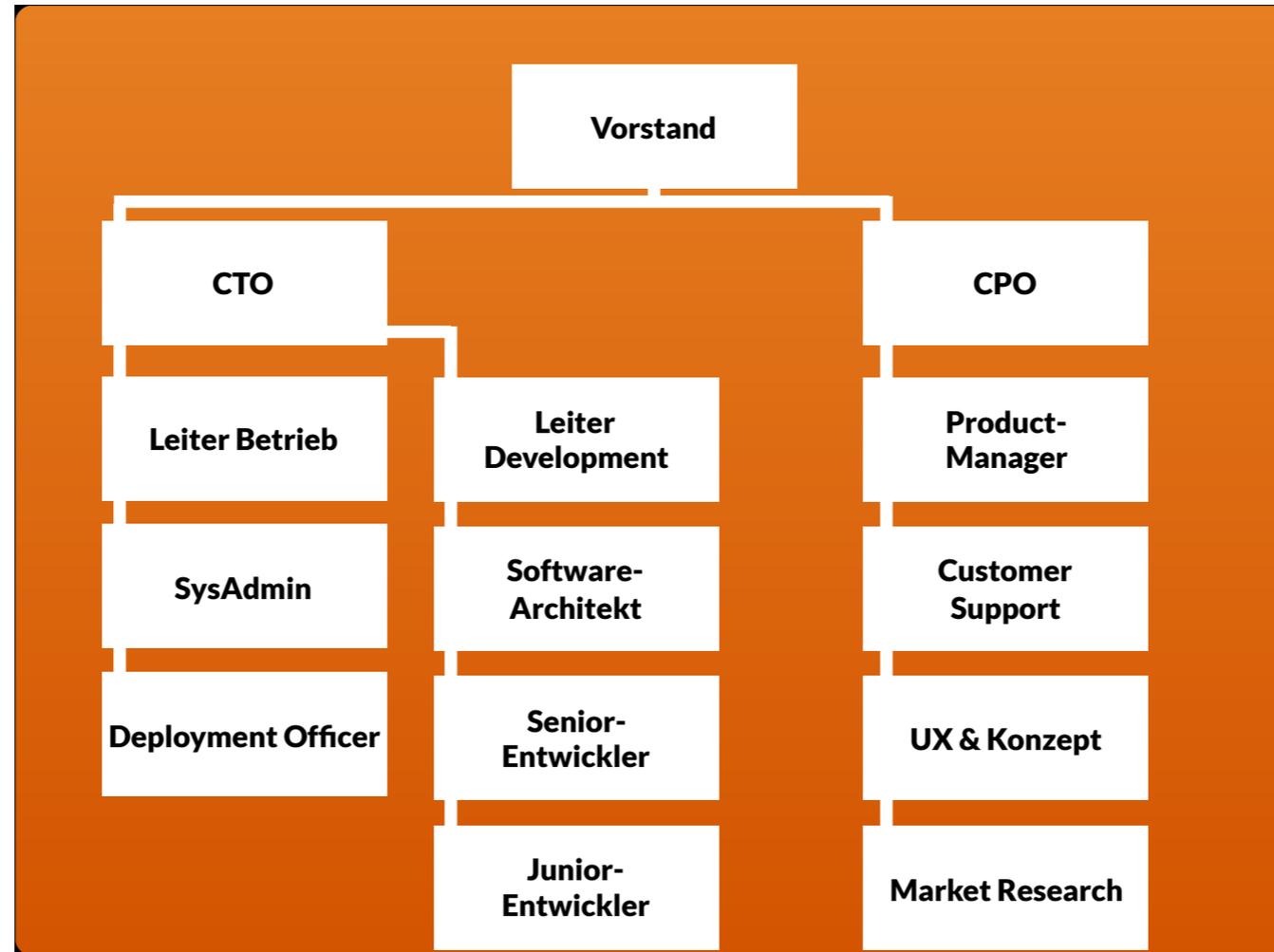
Incentivierung von Erfolg

Korrektur durch **Feedback & Kritik**

Klingt ja auch
logisch.

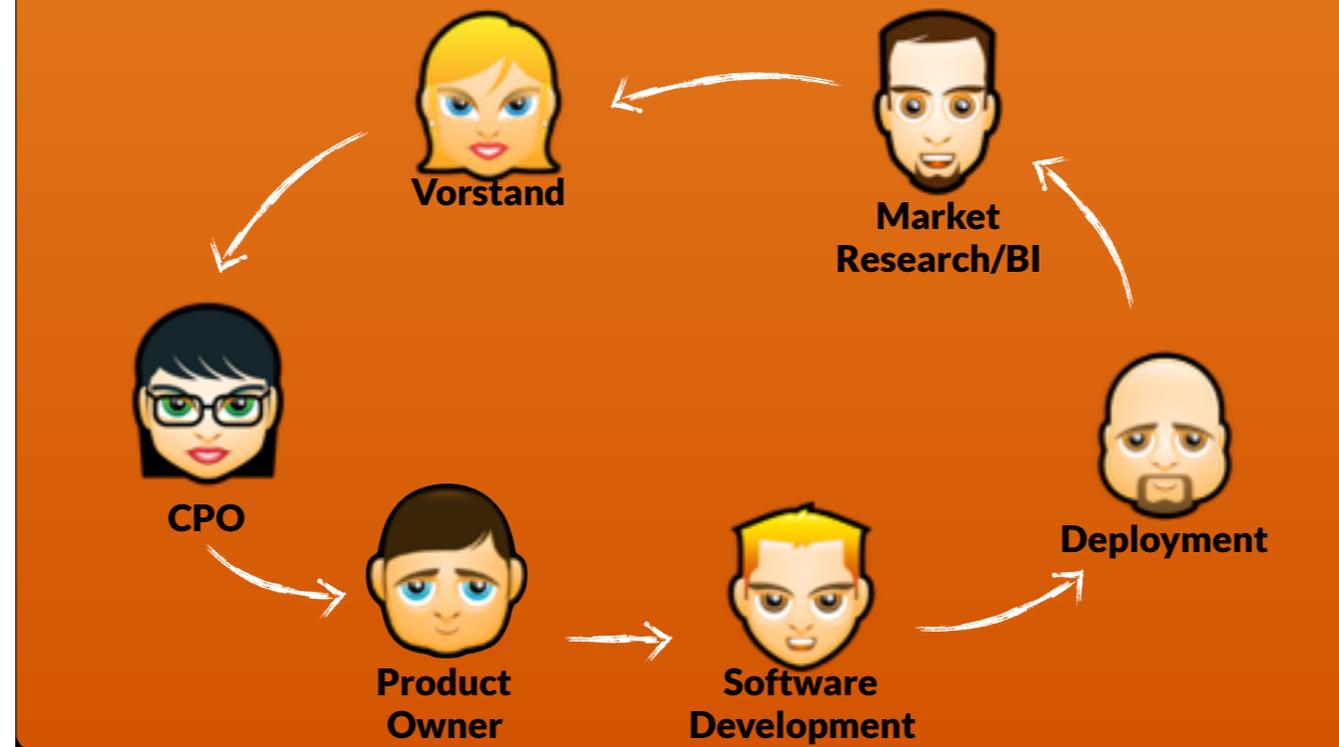
Wo ist das
Problem?

So weit, so gut - diese Struktur hat seit 1950 genug Zeit gehabt sich zu etablieren, und ist ja auch offensichtlich nicht gescheitert. Warum sollte ich das also ändern?

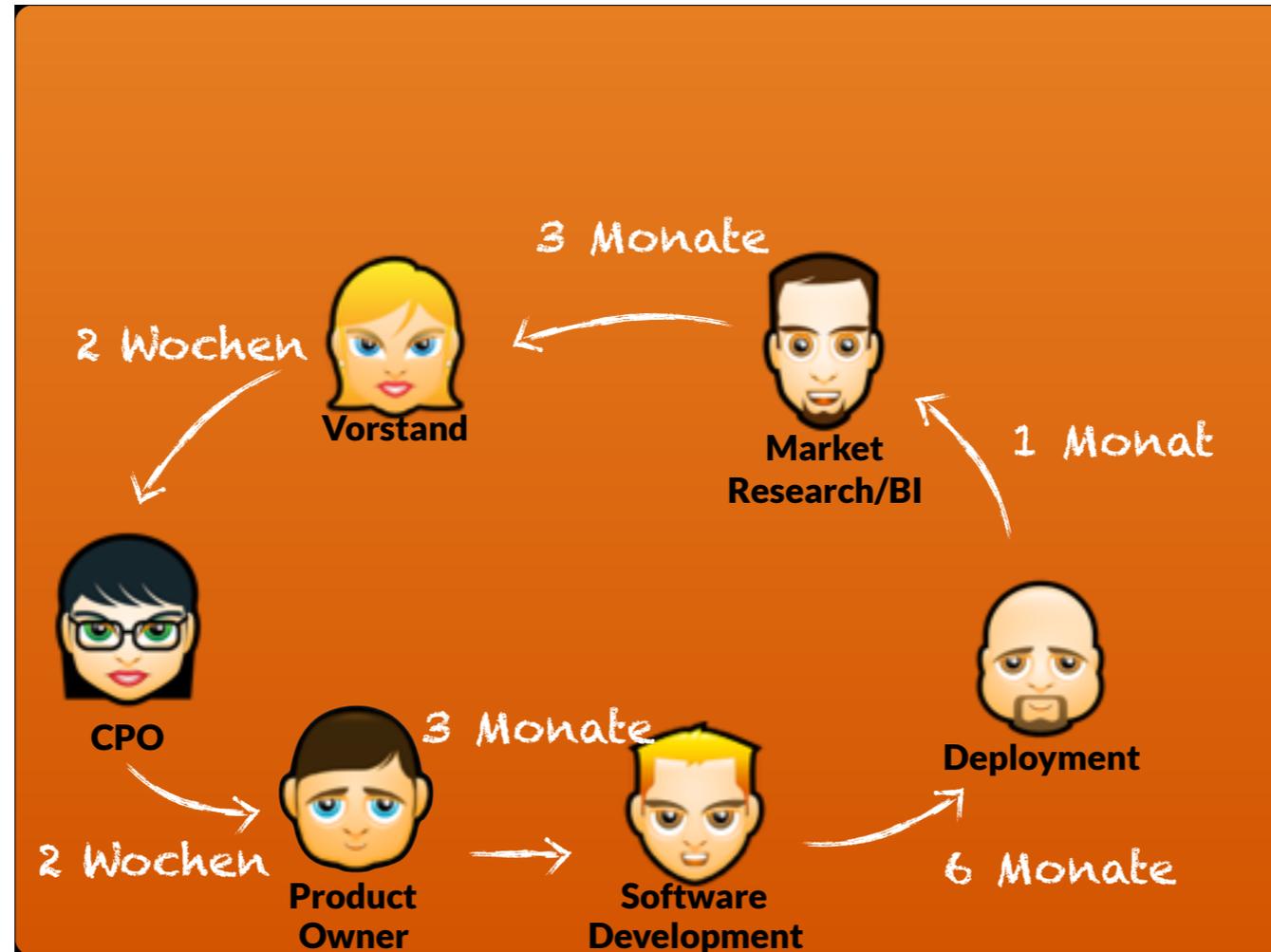


Nehmen wir doch mal eine vereinfachte Unternehmensstruktur, in der es nur IT und Produktentwicklung gibt. Und der Vorstand hat das Unternehmensziel, einen neuen Markt mit einem neuen Produkt zu beglücken.

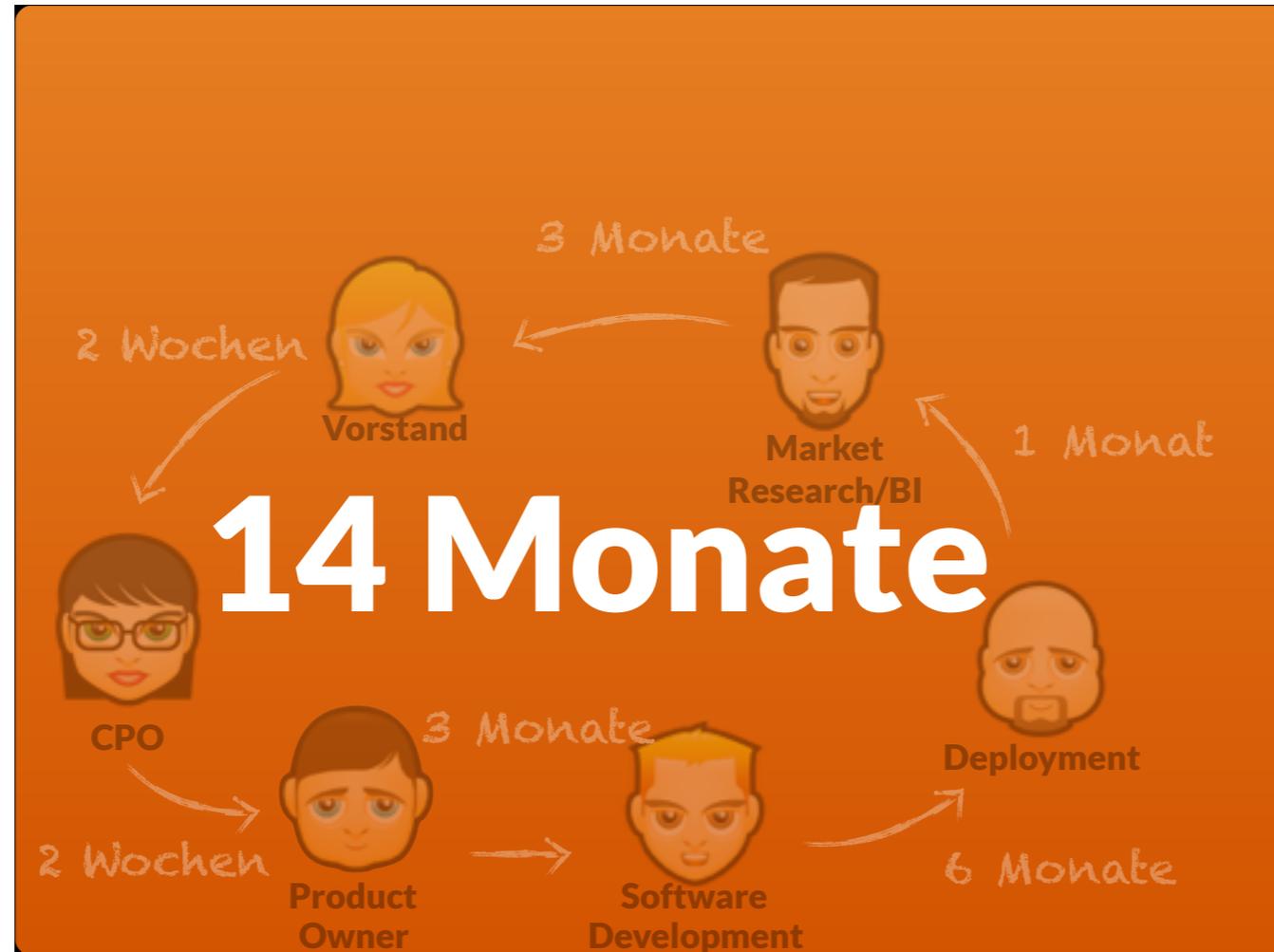
Neues Produkt in 3 Wochen!



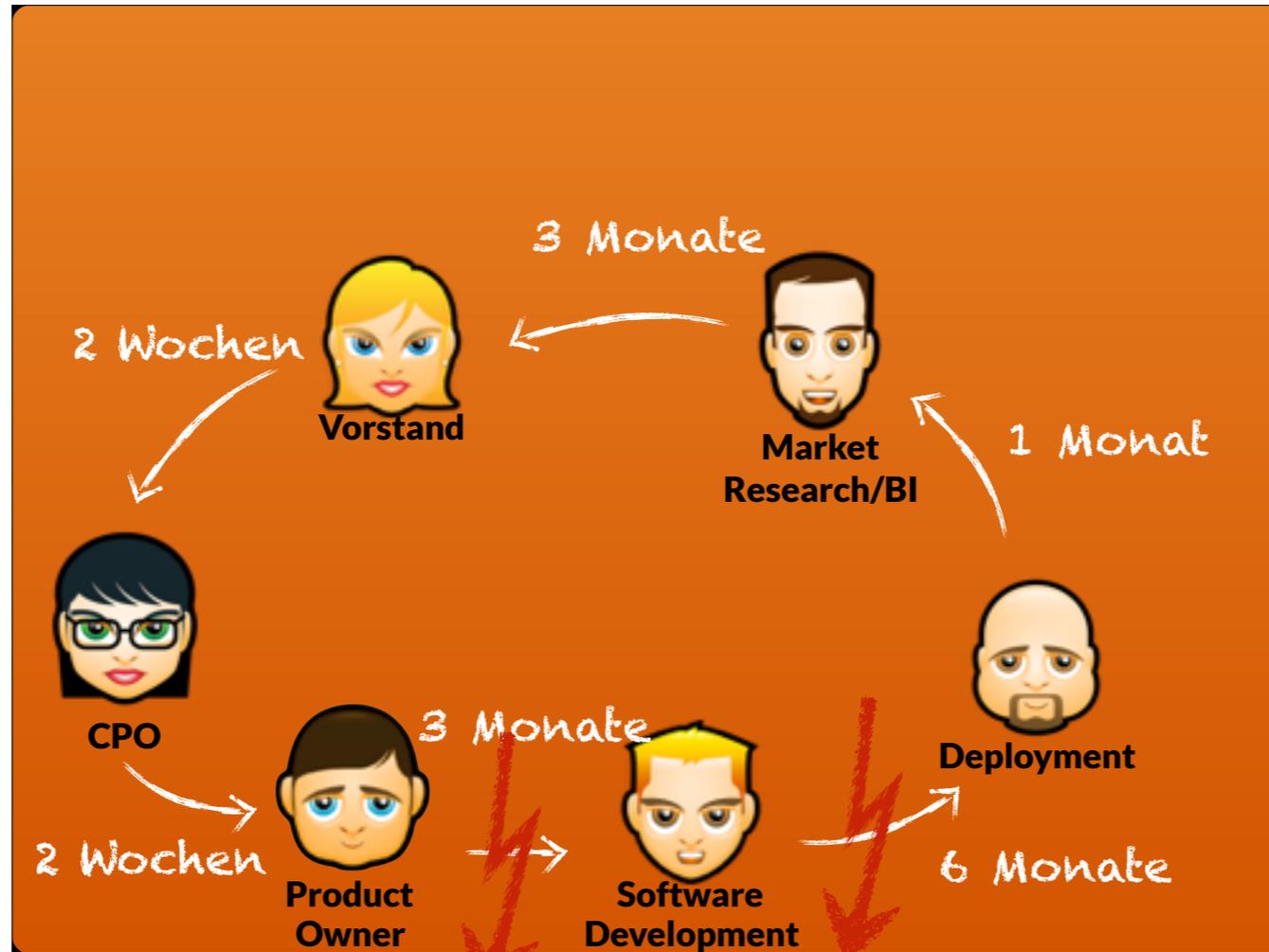
Damit das neue Produkt herauskommt definiert der Vorstand ein Ziel für den CPO, den Chief Product Officer, der formuliert die Vision und beauftragt mit dieser den Product Owner. Der bricht das ganze runter, und gibt es der Software Development Abteilung. Die entwickelt das, es geht in Produktion, und man sammelt Daten über die Nutzung.



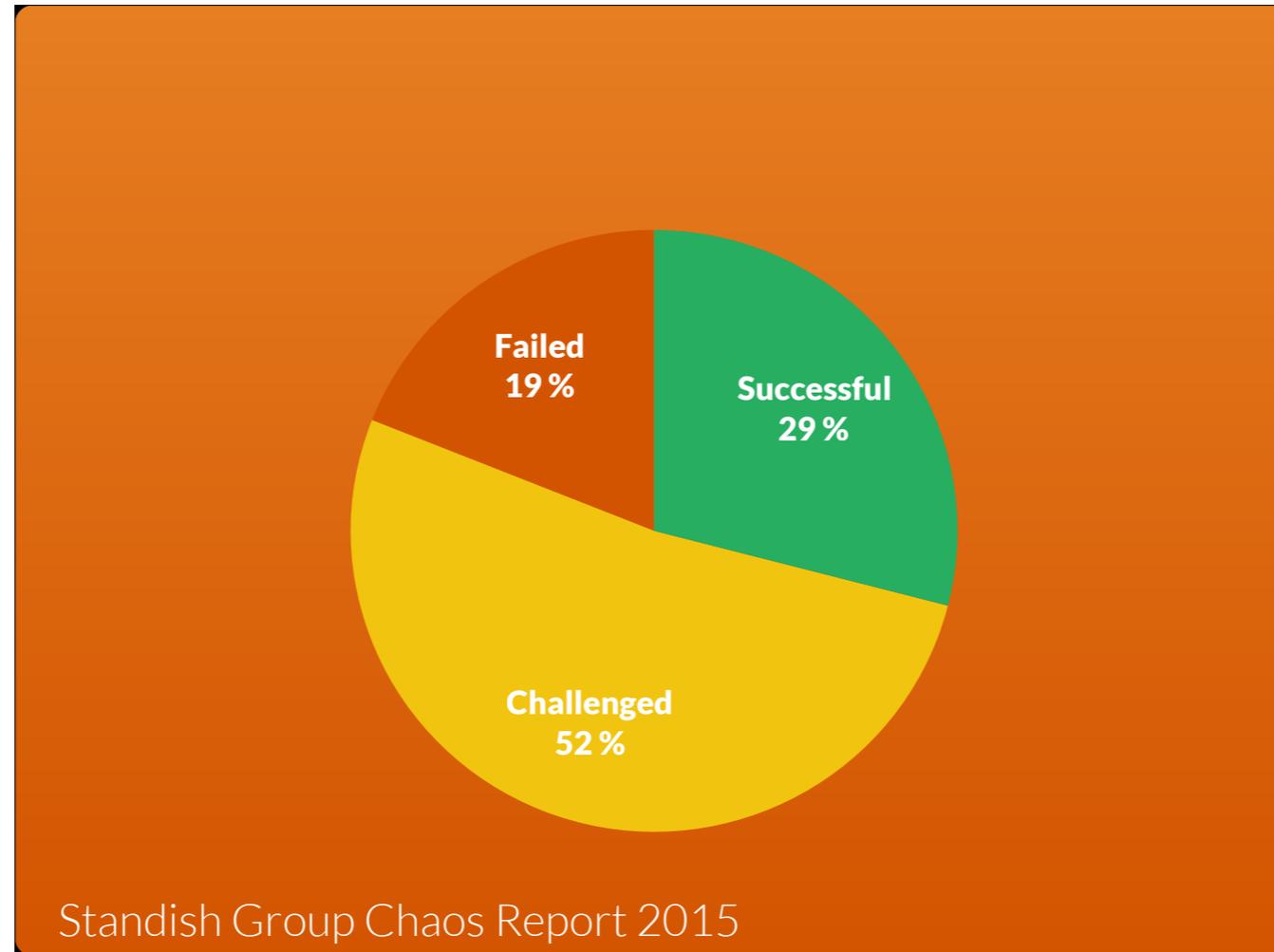
Ok, aber wo ist das Problem? Der Vorstand trifft sich nur einmal im Monat zur geschlossenen Sitzung. Da wird das ganze beantragt, abgestimmt, und es gibt den Auftrag. Beim nächsten Meeting mit dem Chief Product Officer wird das ganze dann besprochen, und die tütet das ganze bei dem Product Owner ein, der gerade Zeit hat. Der macht diverse Workshops, bis er das Konzept fertig hat, und übergibt das ganze dann an den Leiter Development, der gibt es dann an ein Team weiter. Die sind ziemlich schnell, deshalb sind sie nach 6 Monaten - inklusive Test-Abteilungen, Quality-Gates etc - damit durch, und geben es dem Betrieb, der es im Verlauf des nächsten Monats auch online bringt. Dann läuft es 3 Monate in Produktion, die BI wertet das Feedback aus, und gibt einen Report an den Vorstand. Der trifft sich einmal monatlich, wie gehabt.



Wenn man sich das in der Summe anschaut, weiß der Vorstand schon nach 15 Monaten, ob das neue Produkt eine gute Idee war oder nicht.



Und das auch nur dann, wenn die Strecke klappt und wir ITler nicht grossen Unfug erzeugen.



Und wir erzeugen erwiesenermassen regelmässig Unfug. Schaut man sich die letzten Zahlen der größten Studie zum Projekterfolg von IT-Projekten an, dann liefern wir in weniger als einem Drittel der Fälle tatsächlich erfolgreich.

~~On Time, On Budget,
Scope Done~~

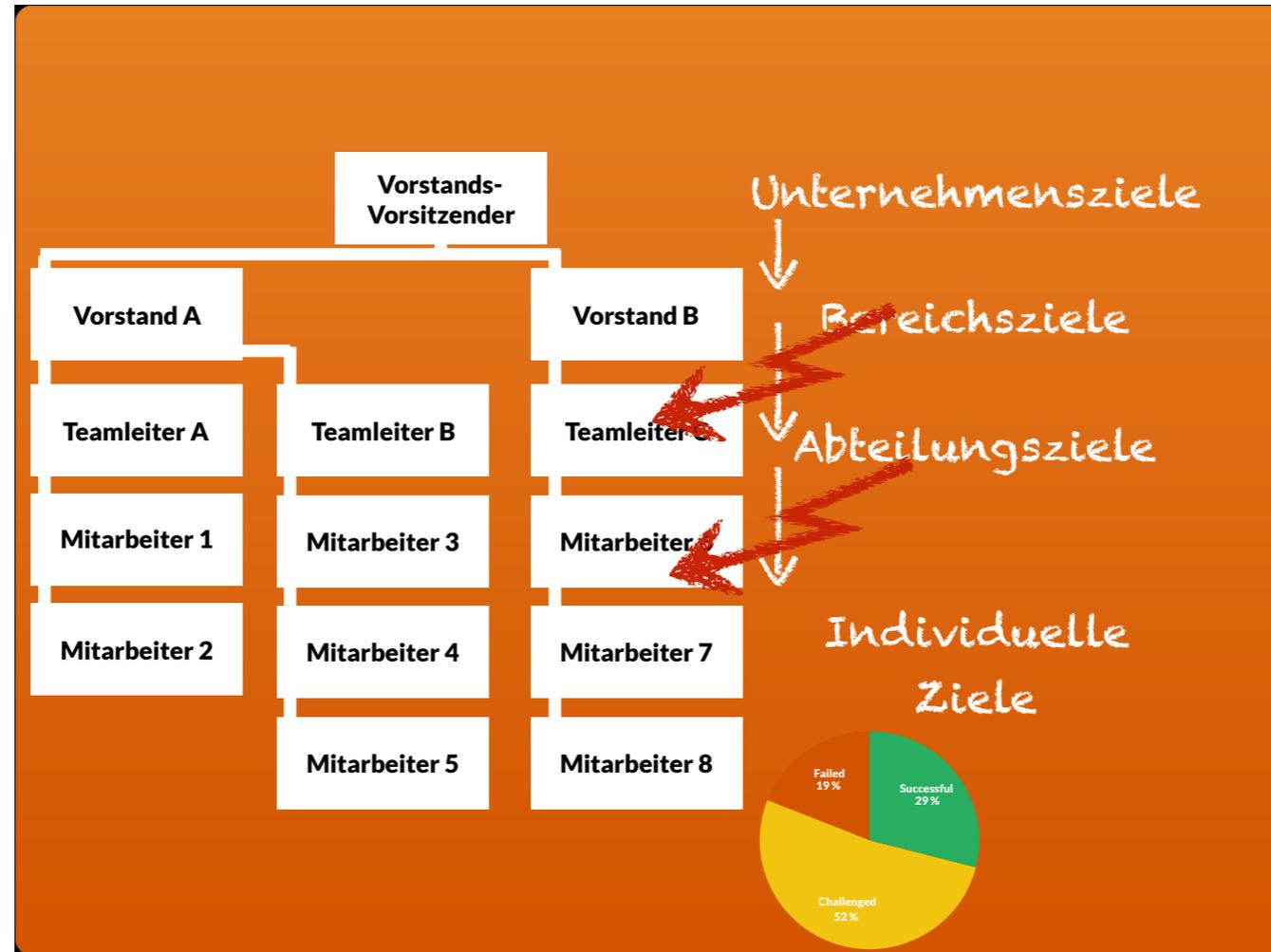
**On Time, On Budget,
Satisfactory Result**

Und die 29 Prozent erreichen wir auch nur deshalb, weil wir nicht mehr festen Scope liefern müssen, sondern nur etwas was „zufriedenstellt“.

- „**1** Software eats the world,
- 2** Every company becomes a software company, and
- 3** Software people run every company“



Und jetzt haben wir ein Dilemma - denn auf einmal wird jede Firma eine Softwarefirma.



In dieser Welt scheitert die Grundidee von Management by Objectives zwangsläufig, weil wir gar nicht in der Lage sind, diese Ziele zu liefern. Wir versuchen es zwar immer wieder, aber wir verfehlen die Ziele zu 70%.

Was mache ich,
wenn ich schon
weiß, dass ich
wahrscheinlich
scheitern werde?

Und was macht man, wenn man merkt, dass man deutlich häufiger daneben liegt als das man trifft? Man versucht, den Misserfolg zu professionalisieren. Und wie professionalisiert man Fehler?

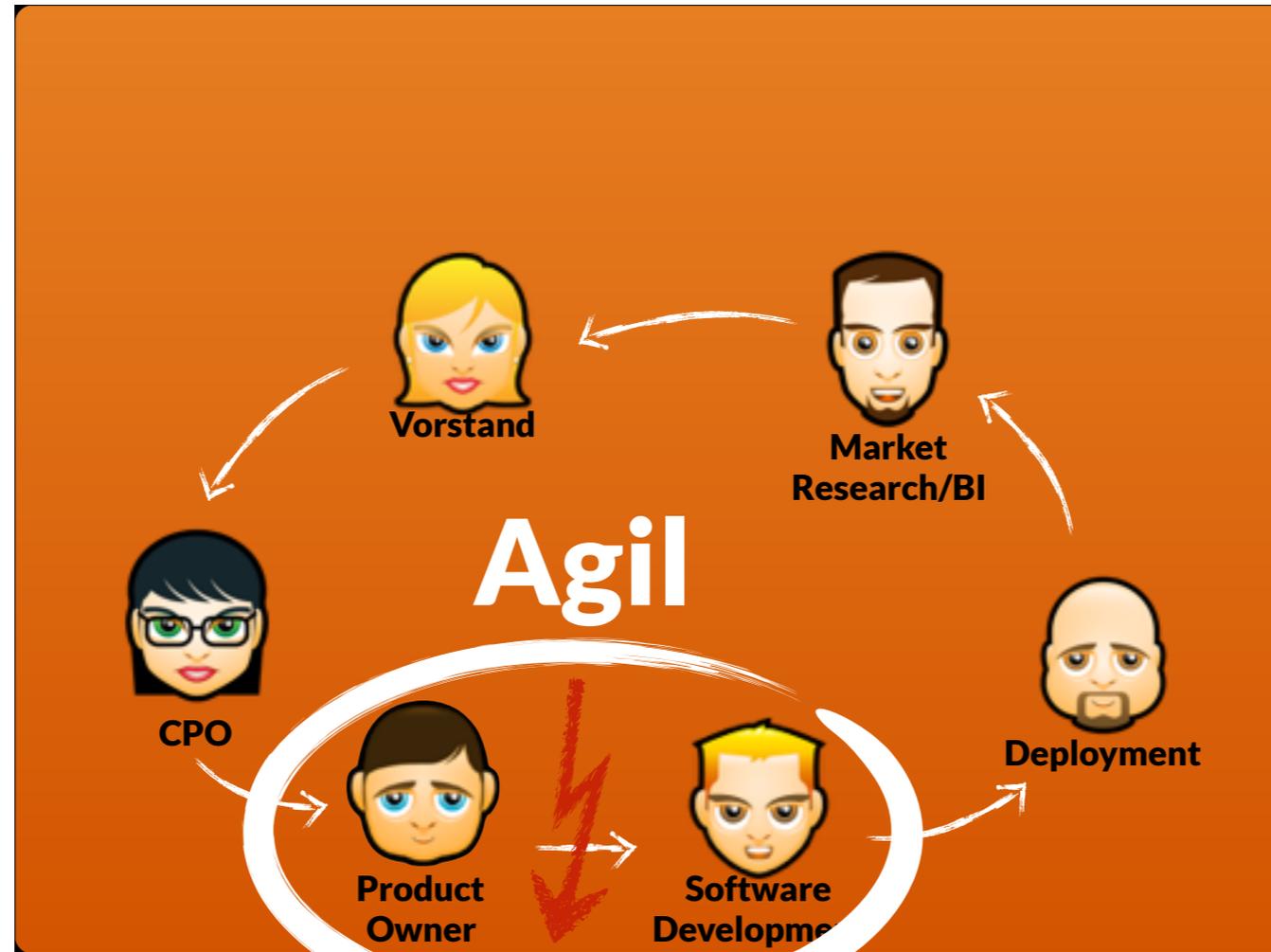
Ich versuche, es
schnell zu merken
und dann zu
reparieren.

Man richtet sich danach aus, es möglichst schnell und preiswert zu merken.

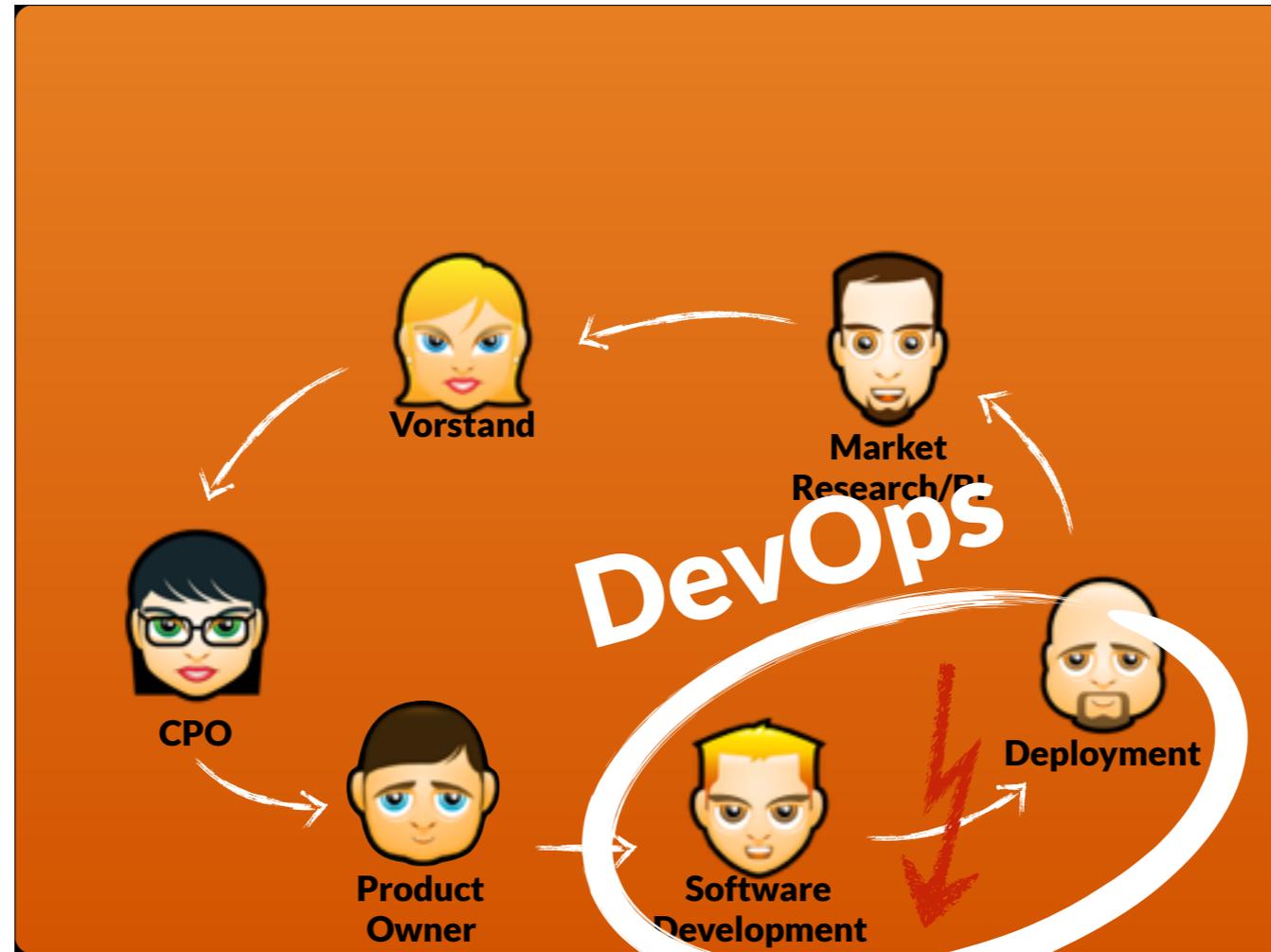


Fail **O**ften
Fail **F**ast
Fail **C**heap

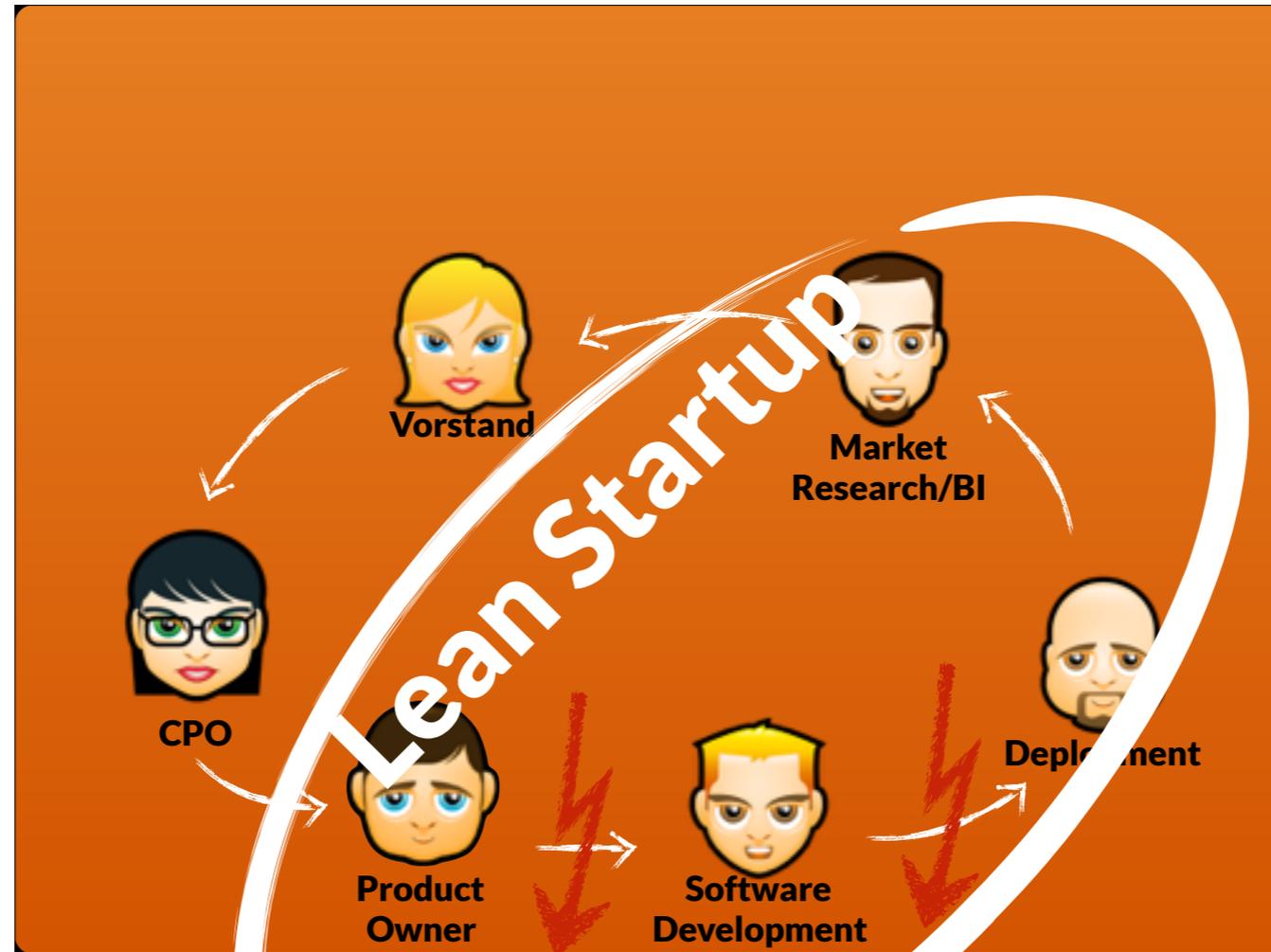
Kristian ist in seiner Keynote im letzten Jahr genau darauf eingegangen - und eigentlich kennen wir ITler das schon seit 1975, das Konzept wird schon von Fred Brooks im mythischen Mannmonat erwähnt. Und wie sieht das ganze in der Praxis aus?



Das haben natürlich auch wir ganzen Leute an der Praxis gemerkt, und unsere Methoden dafür entwickelt. Agile Softwareentwicklung setzt unmittelbar da an, wo der Schmerz beim Standish Group Chaos Report herkommt - zwischen Business und IT. Das hat gut funktioniert, und die Erfolgchancen deutlich erhöht - und in Folge ist es Mainstream geworden.



Nachdem das Development jetzt ungefähr das richtige gemacht hat, konnte man das nicht dem Kunden gegenüber auf die Strasse bringen - und DevOps ist entstanden - dort hat man die Strecke von Development zur Produktion endlich mit Werkzeugen wie Automatisierung, Continuous Deployment und Kooperation zum fliegen gebracht.

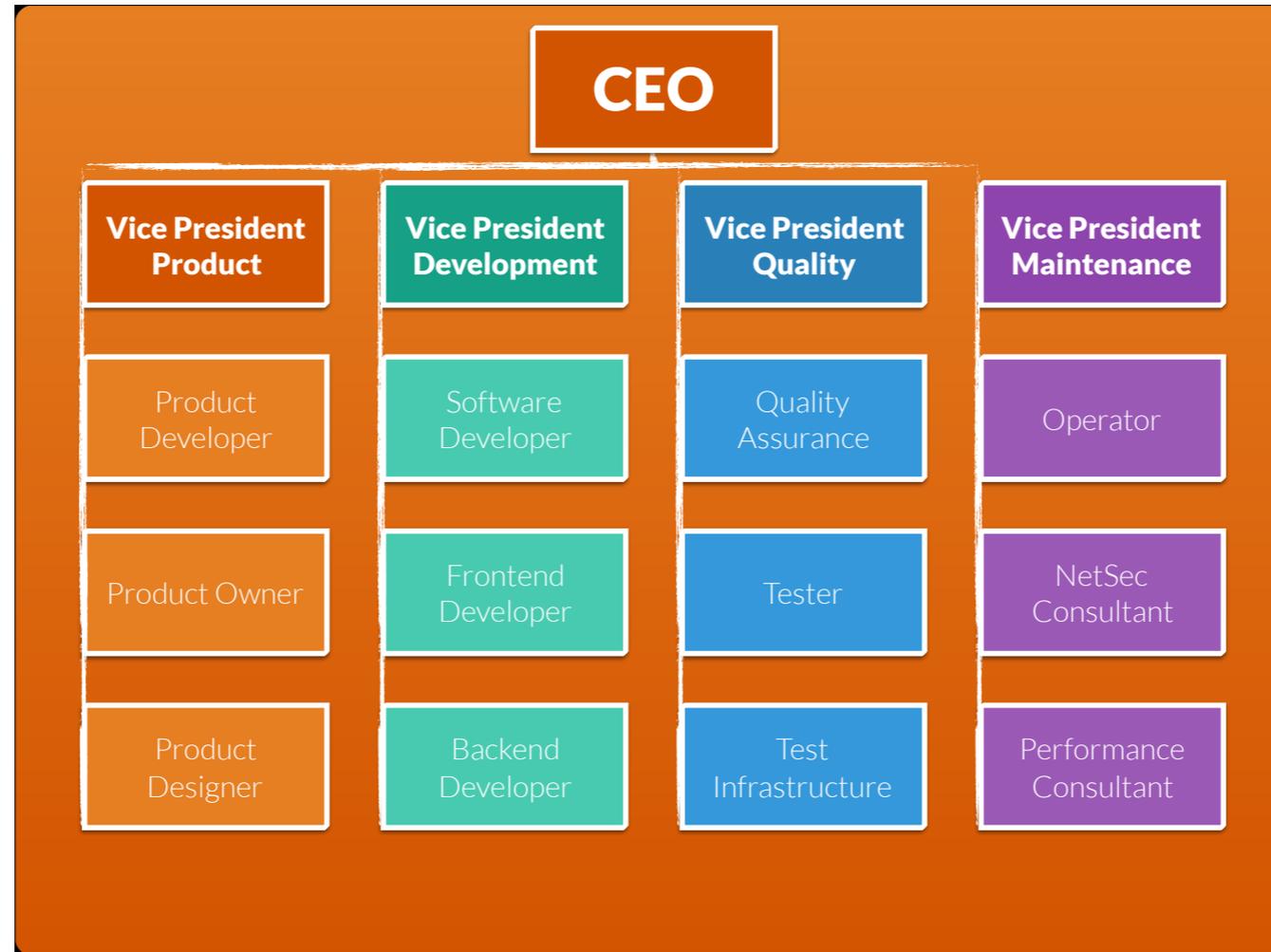


Lean Startup hat sich dann in Folge die komplette Strecke von Product Development über Development, Produktion und Business Intelligence geschnappt, und das gemeinsam in eine schnelle Feedbackschleife gebracht.

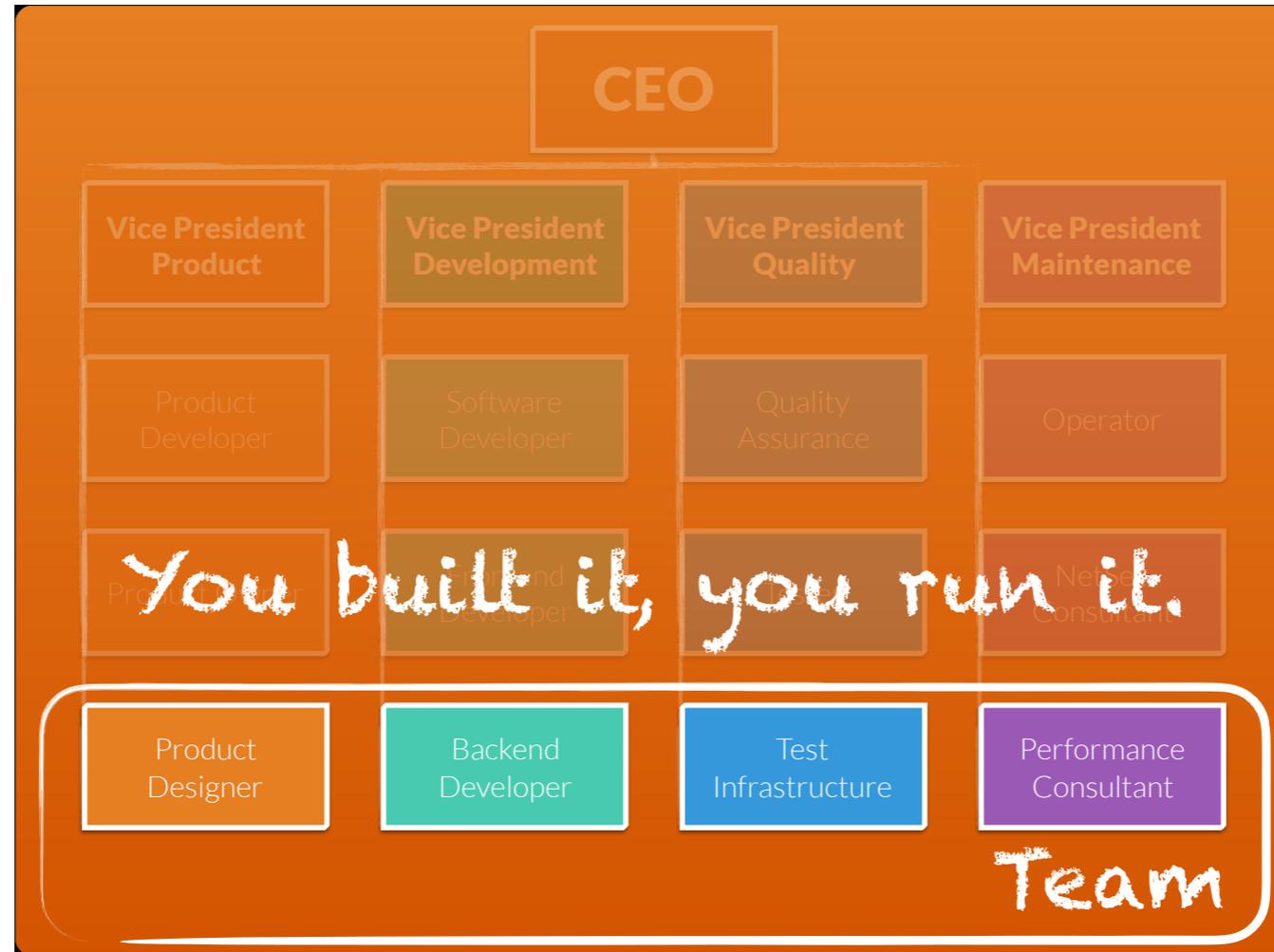
Lean Startup

Alles sind **Hypothesen**.
Sogar das **Unternehmensziel**.
Erst **Daten validieren** es.

Doch Lean Startup treibt den Ansatz der Invalidierung/Validierung noch weiter - denn hier wird praktisch alles als Hypothese betrachtet, und das schliesst sogar das Unternehmensziel ein. Dort herrscht die Logik: ich messe, ob es funktioniert, und wenn es funktioniert, dann machen wir weiter -sonst machen wir einen Pivot, sprich: wir haben einen Plan B und wechseln auf diesen. Wie man sieht: die Adaptionfähigkeit von solchen Methoden ist hoch.

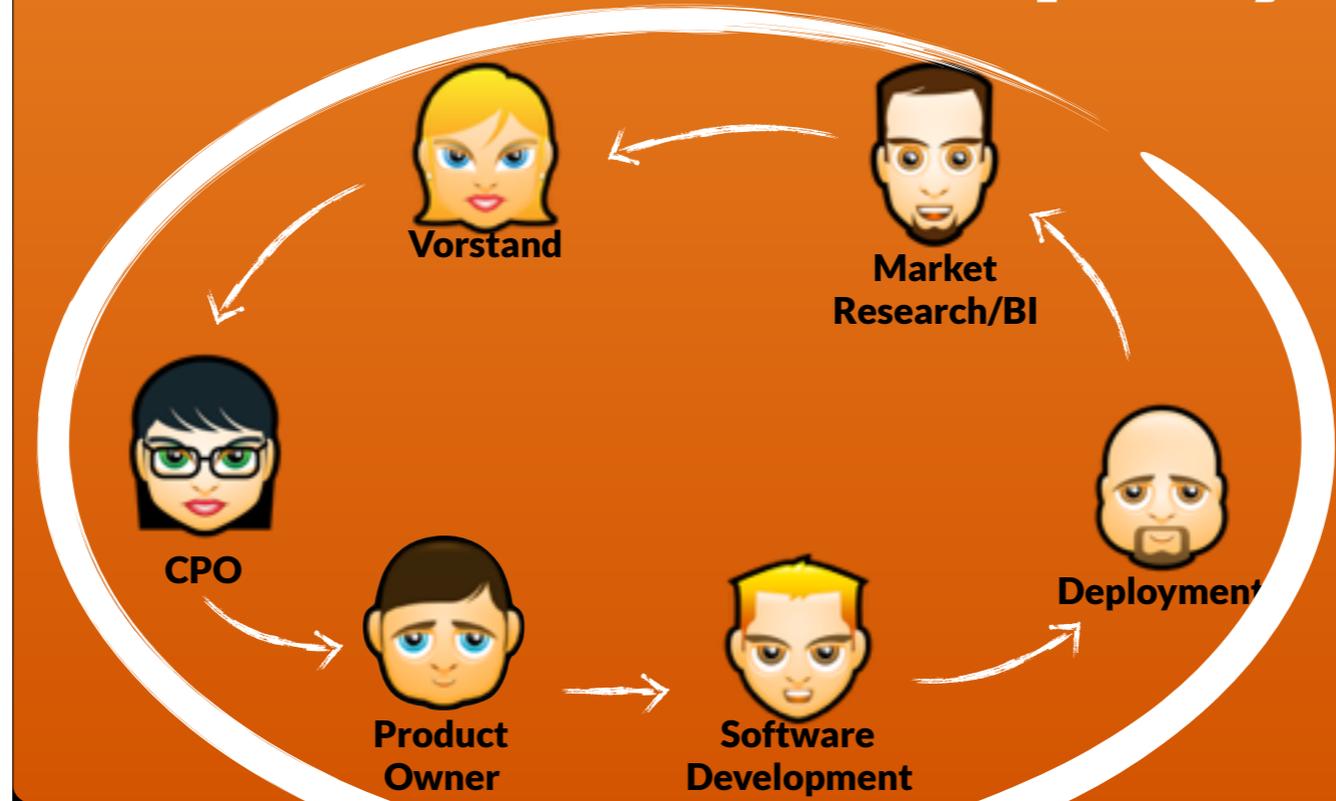


Damit man schnelles Feedback bekommen kann muss man Silo-Effekte brechen - konkret bedeutet das im Fall von Agil, DevOps und Lean Startup das Brechen der bisherigen spezialisierten Abteilungsgrenzen.



Damit wechselt die Organisation von einer funktionalen Organisation - jede Abteilung ist auf ein fachliches Thema spezialisiert - zu einer crossfunktionalen Organisation, bei der die Teams alles enthalten, was wie für die Arbeit brauchen.

Data Driven Company



Im grösseren Massstab heisst das ganze dann Data Driven Companies. Dort richtet sich das komplette Business weitgehend nach Daten, und es wird von Messungen getrieben gearbeitet.

Data Driven Company

Jeder hat **alle Daten**
Jeder soll & kann sie **nutzen**
Datensammlung immer & überall

Bei datengetriebenen Firmen hat jeder Zugriff auf alle Daten, und alles ist quasi in Echtzeit vorhanden. Alle Daten sind breit vorhanden, jeder soll und kann sie nutzen, und es wird praktisch alles an Daten gesammelt, was man potentiell einmal in Zukunft brauchen kann. Beispiele für solche Firmen sind natürlich Amazon, aber auch das 100-Millionen-Startup „Project Collins“ von der Otto Gruppe oder Kristian Köhntopps Ex-Arbeitgeber [booking.com](https://www.booking.com). Auch Facebook oder Google haben viele Bereiche, die hart datengetrieben arbeiten.



In solchen Firmen kann der komplette Lernzyklus - dann zwangsläufig eher über Features als ein ganzes Produkt - oft schon in einer Reihen von Tagen durchgeführt werden, je nachdem, wie lange man braucht, um genug relevante Daten vom Endkunden zu bekommen.

Check.



Schnellere **Innovation**
Schnellere **Adaption**

Netterweise haben wir damit nicht nur unsere Erfolgschancen erhöht - wir haben auch gleichzeitig einen hervorragenden Mechanismus zur Innovation und Adaption geschaffen. Solche Unternehmen sind hervorragend in der Lage neues zu schaffen als auch auf geänderte Umstände zu reagieren, denn durch die hohe Transparenz und das schnelle Lernen können sie schnell adaptieren.

Leitungsaufgaben

Das braucht niemand mehr..



Setzen der **Unternehmensziele**

Klärung von **Unterzielen**

Delegation von Verantwortung

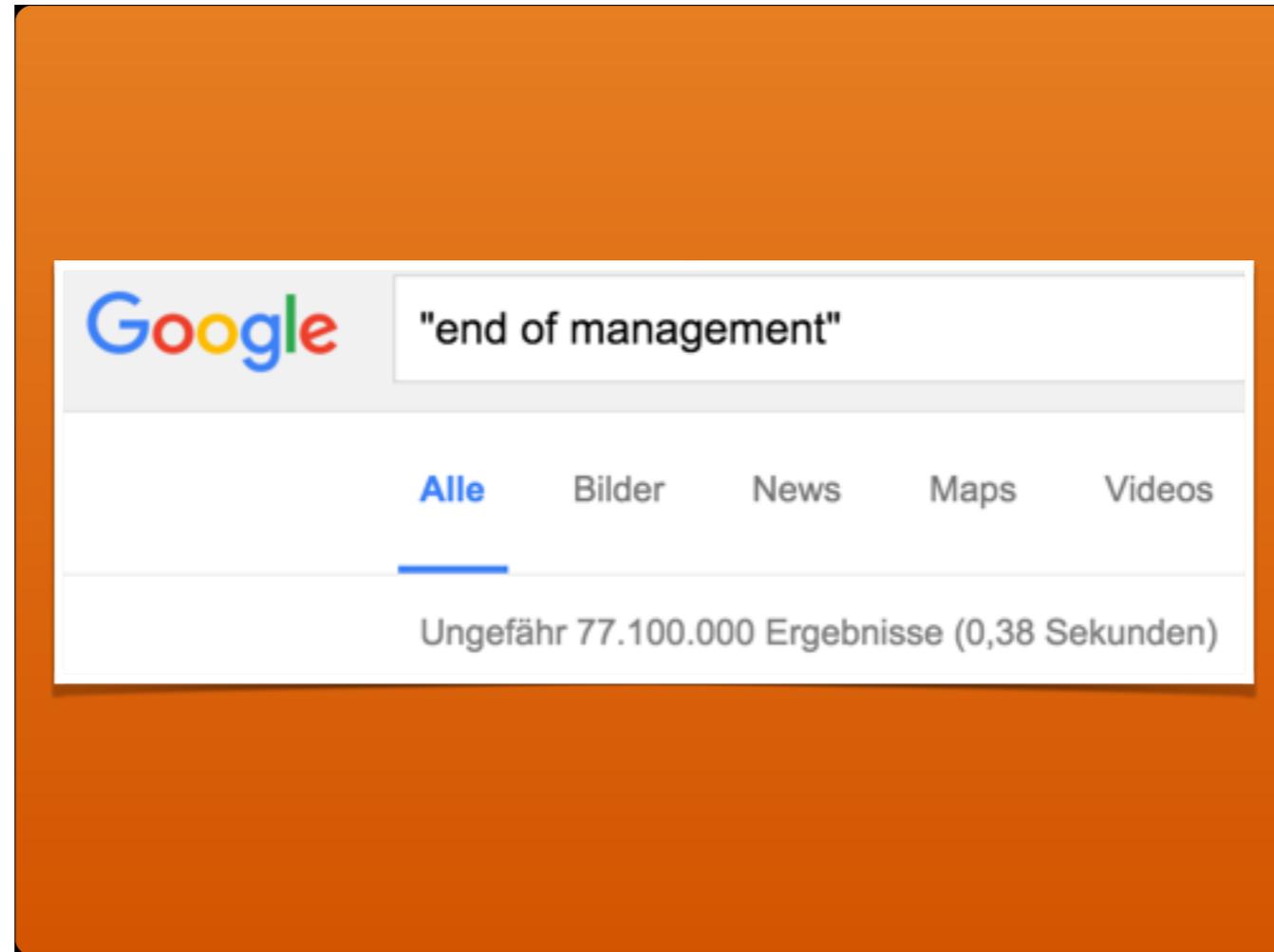
Incentivierung von Erfolg

Korrektur durch **Feedback &**

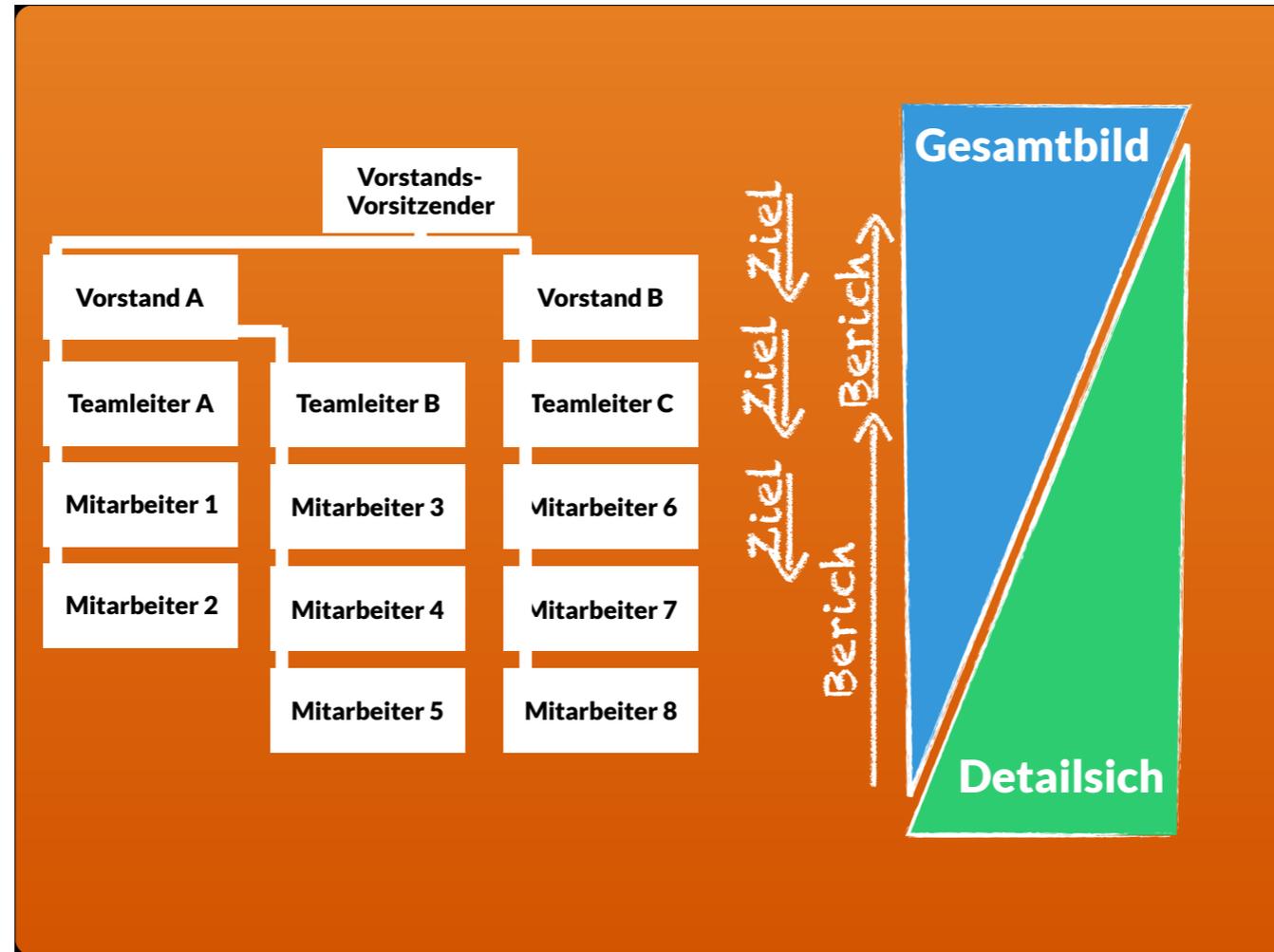
Kritik

Aber - was macht das Management in einer lernenden, datengetriebenen Organisation?

Die klassischen Aufgaben braucht niemand mehr: die Ziele setzen, die Klärung der Unterziele, die Delegation der Verantwortung, die Intensivierung von Erfolg, die Korrektur - das alles ist auf der Basis von Probieren - und Messen-Zyklen deutlich besser hinzubekommen.



Das haben natürlich auch viele andere gemerkt, und wenn man heute nach End of Management googled, dann kommt man bei 77 Millionen Treffern heraus. Ein Bekannter von mir hat sogar „Management Exorzist“ als Berufsbezeichnung auf der Visitenkarte stehen.



Und damit sind diese Formen von Management gemeint.

Leitungsaufgaben

Agile Softwareentwicklung:

Servant Leadership

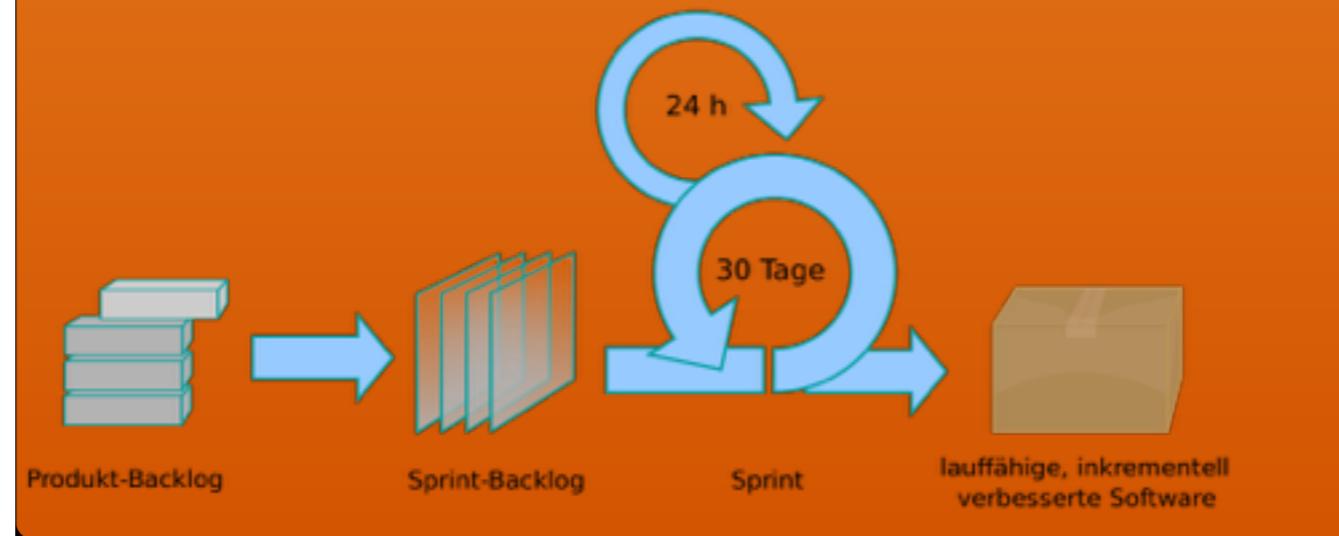
Coaching

Stewardship

Transformationale Führung

Agile Softwareentwicklung gibt es inzwischen seit 25 Jahren, deshalb hatte man da genug Zeit zu diskutieren welche Leitungsfunktionen noch gebraucht werden.

SCRUM



Aller Statistik nach am weitesten verbreitet ist Scrum als Methode für Softwareentwicklung. Wer hier hat davon schon einmal gehört?
Wer hat das schon einmal selbst gemacht?

Servant Leadership Coaching

Scrum-Master:
Show-Stopper beseitigen
Coaching von Personen &
Teams

Bei Scrum gibt es keine klassische Abteilungs- oder Teamleiterrolle mehr, sondern zwei Führungsrollen, die statt dessen eintreten. Die eine Rolle ist der Scrum-Master, der als Servant Leader agiert. Er bewegt sich irgendwo zwischen Projektassistenz und Coach, und sorgt nicht nur für Meetings, sondern hat auch explizit die Aufgabe Probleme als Dienstleister des Teams zu beseitigen.

Transformationale Führung Stewardship

Scrum-Product-Owner:
Produktvision pflegen
Aufgaben priorisieren

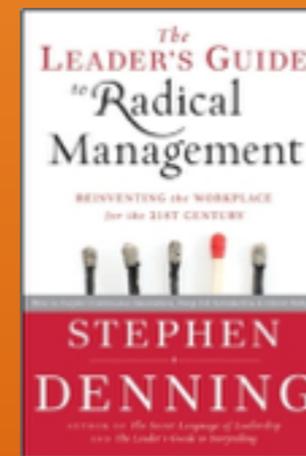
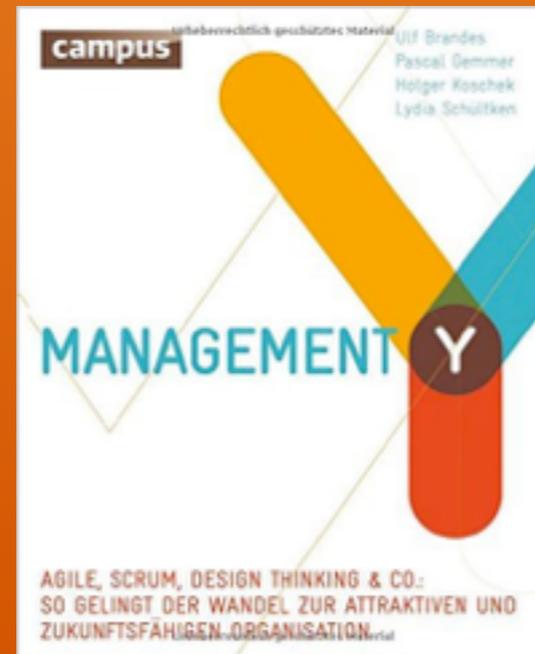
Die zweite Führungsrolle bei Scrum ist der Product Owner. Der Product Owner definiert das Produkt, indem er als Dienstleister die Produktvision pflegt und kontinuierlich adaptiert.

Der Product Owner darf keine direkten Aufgaben verteilen, und er weist auch nicht die Entwickler an. Aber er vergibt die Prioritäten der Aufgaben, und die Entwickler müssen diese - wer auch immer es dann konkret tut - in dieser Reihenfolge bearbeiten.

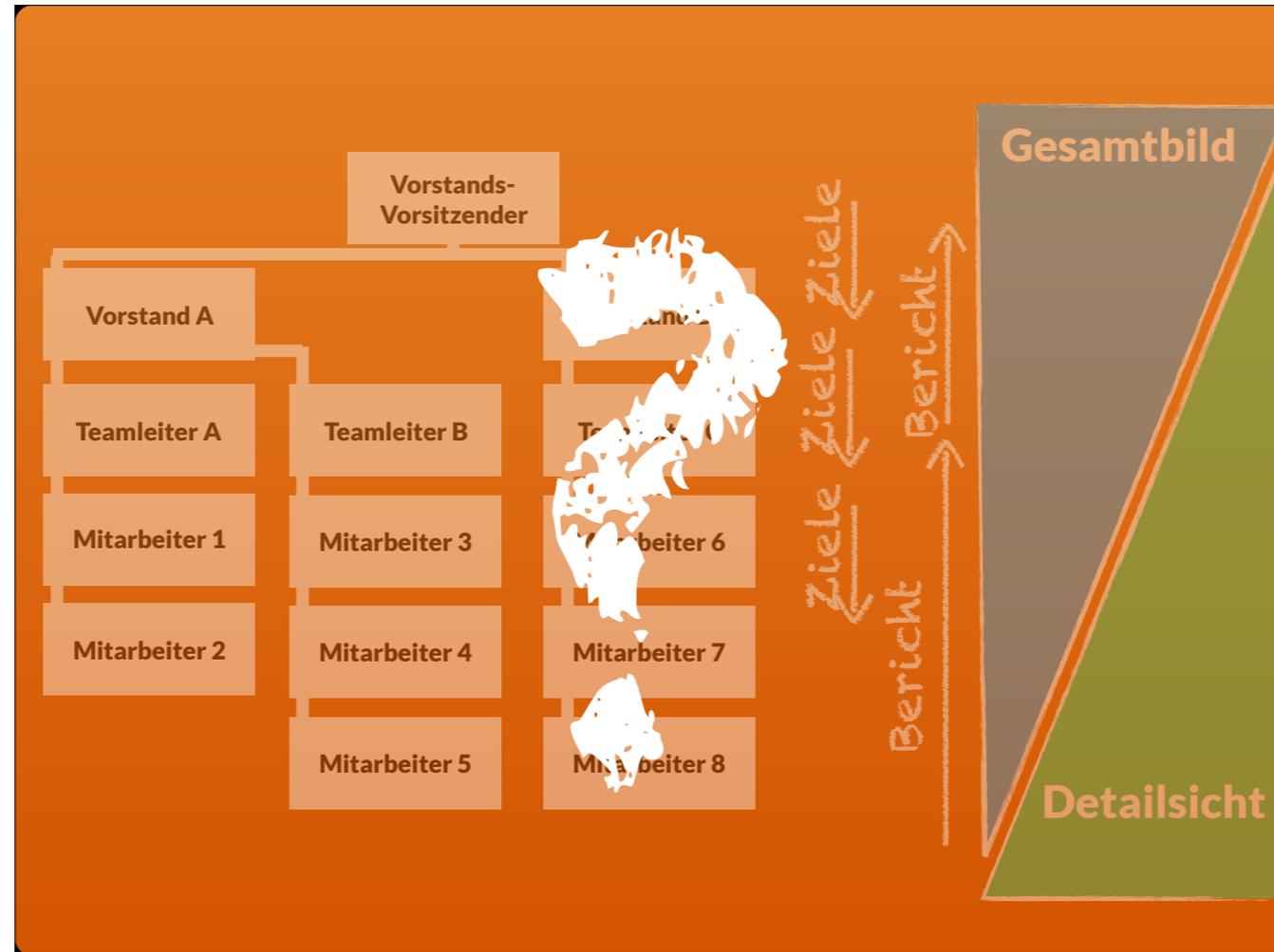
Da steht **weder**
Teamleiter noch
Abteilungsleiter.

Für viele Firmen ist das schwer - denn sie haben noch die Strukturen aus dem Organigramm, die Funktionen werden aber gar nicht mehr gebraucht.

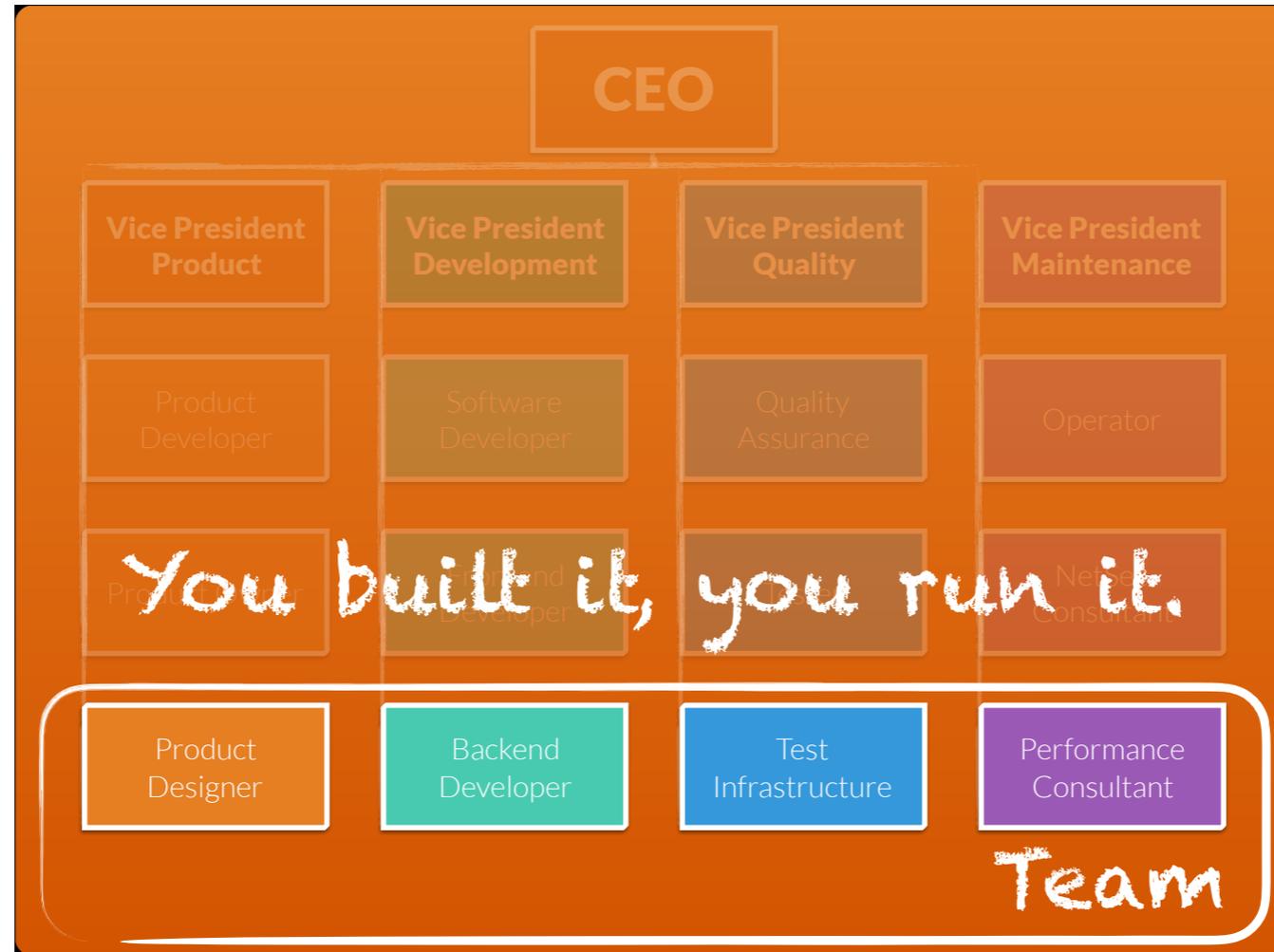
MANAGEMENT 3.0 Agile Leadership Practices



Zum einen gibt es einen ganzen Haufen Literatur, wie man die Leitungsfunktion jetzt macht - insbesondere in den Etagen, am Organigramm lang betrachtet, oberhalb der Teamebene.



Diese Ratlosigkeit, was denn die richtige Führung ist, hat aber noch einen weiteren Grund - die bisherige Struktur passt nicht mehr zu der Art zu arbeiten - denn nicht nur die konventionellen Leitungsfunktionen finden nicht mehr statt, die Strukturen ändern sich auch. Was muss ich machen, damit diese Struktur möglichst performant sind?



Wir hatten vorhin schon die crossfunktionalen Teams, die in der Lage sind, die vollständige Strecke selbst zu bedienen. Das heisst, sie können selbst Experimente fahren und selbst anhand der Daten messen, was funktioniert und was nicht funktioniert.

Organisationsstruktur nach **Deployment** statt nach **Spezialisierung**



Und das hat einen ganz interessanten Effekt gebracht. Auf einmal ist nicht mehr die Spezialkompetenz die Schnittkante der Abteilungen, sondern das Deployment. Der altmodische Admin-Task ist auf einmal massgeblich dafür, wie die Firma selbst geschnitten ist.

Portal-Team

Client-basiertes Team

MicroServices

Product
Designer

Backend
Developer

Test
Infrastructure

Performance
Consultant

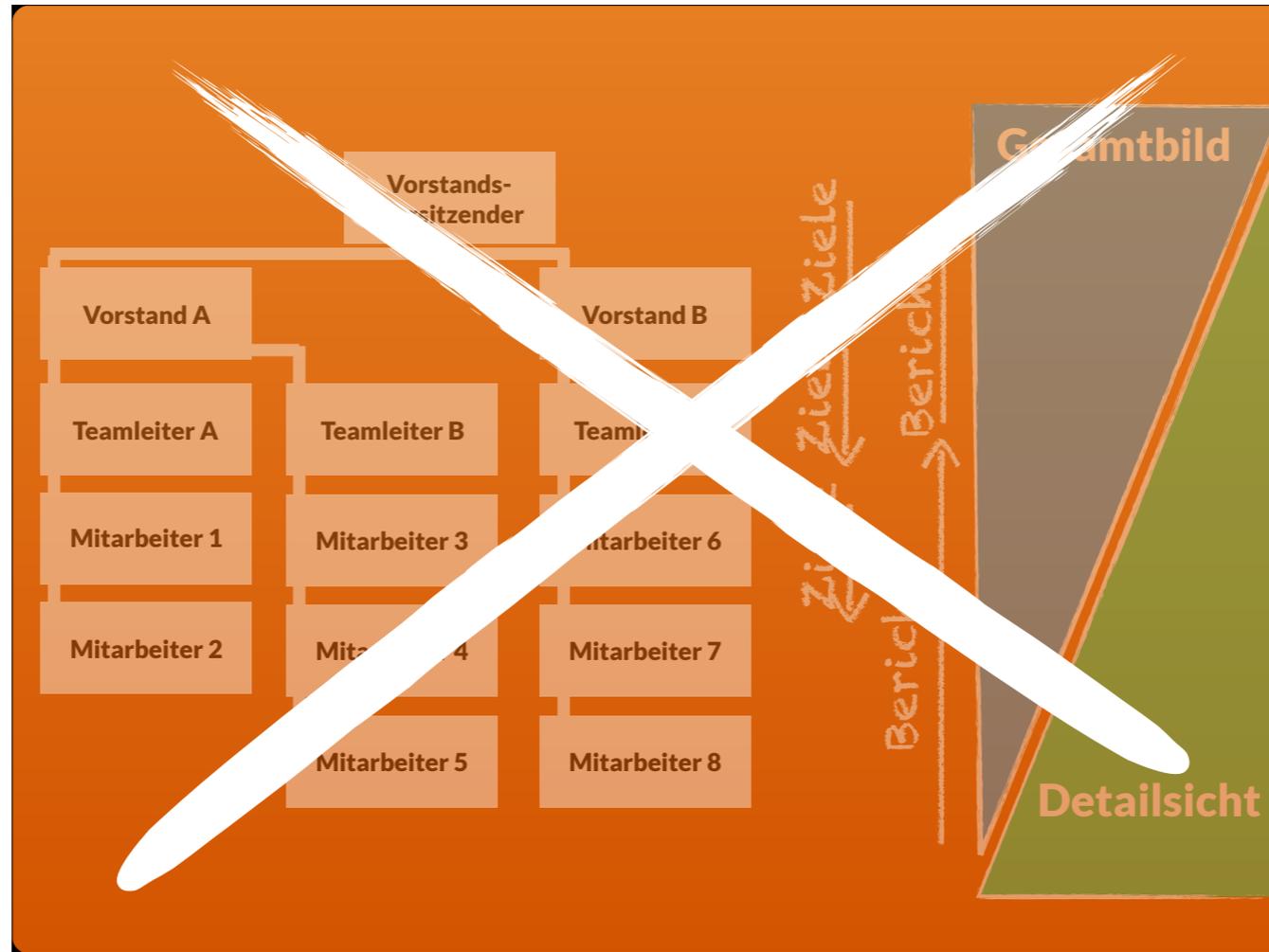
Team

Bei Ottos Projekt Collins sind das zB die Portale, bei Spotify gibt es pro Client ein Team. Bei uns im Software-Development ist die letzten Jahre MicroServices populär geworden, bei denen eine Applikation nach Business-Domänen zerlegt wird, die jeweils autark deploybar sind. Das sind die neuen Abteilungsgrenzen.

Portal-Team
Client-basiertes Team
MicroServices

Deshalb haben die auf
einmal auch alle so
„DevOps“-Leute, die
Docker-Container bauen.

Und weil sie crossfunktional sind, haben die auch alles mit dabei - inklusive der Developer, die auf einmal System-Engineering-Aufgaben übernehmen, und sich deshalb „DevOps“ nennen. Und schlimme Docker-Container bauen.



Und weil diese Form - bei der weder die Steuerung von oben nach unten perlt noch die Trennung nach Funktion passiert - nicht mehr stimmt, passt auch dieses Bild nicht mehr dazu.

**Ok,
aber was dann?**

Welche Organisationsform nutzt man dann, wenn diese nicht mehr funktioniert?

HOLACRACY

CORPORATE STRUCTURE



HOLACRACY STRUCTURE



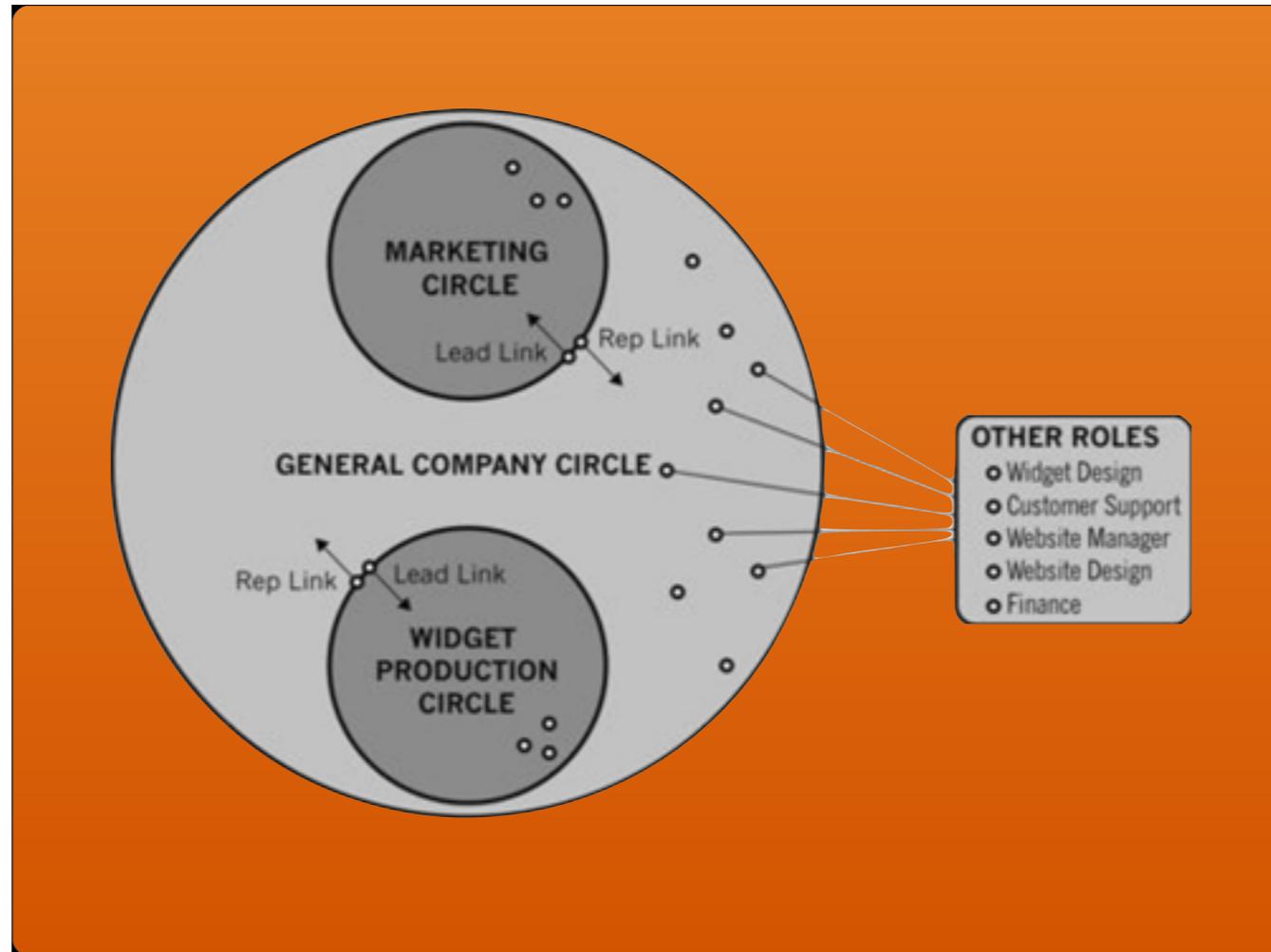
RODMAN'S APOLOGY OVER OUTBURST • GRAND JURY IN NEW YORK 3:50 CT AMERICA

Eine der Organisationsformen mit einem sehr guten Marketing-Department ist Holacracy - hier in einem Screenshot eines Berichtes auf Aljazeera. Diese Form wird also wirklich ernst genommen. Holacracy - oder auf deutsch Holokratie - ist eine der Organisationsformen, die auf Basis von agiler Arbeit entstanden sind.

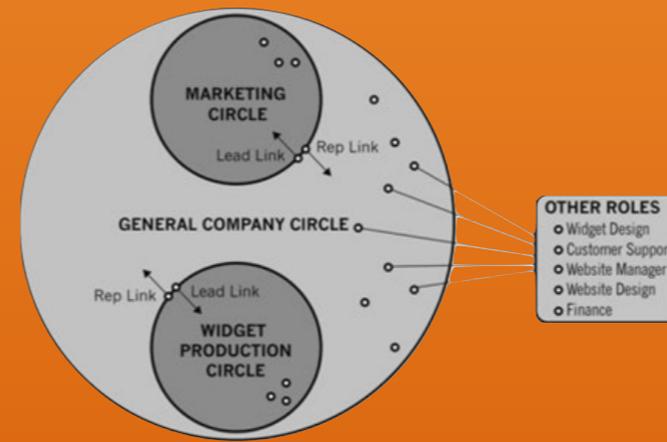
Dynamische Unternehmensstruktur
Rollen statt Positionen
Dezentrale Steuerung
Transparente, universelle Regeln



Holacracy ist vor allem durch die Einführung beim 1500 Personen starken Zappos bekannt geworden. Zappos ist das Vorbild von Zalando und wurde von Amazon gekauft. Aber auch andere Unternehmen wie Medium machen das, inzwischen sollen es an die 500 sein. Auch in Holacracy gibt es Leitungsfunktionen, aber sie sind nicht mehr statisch - sondern dynamisch.

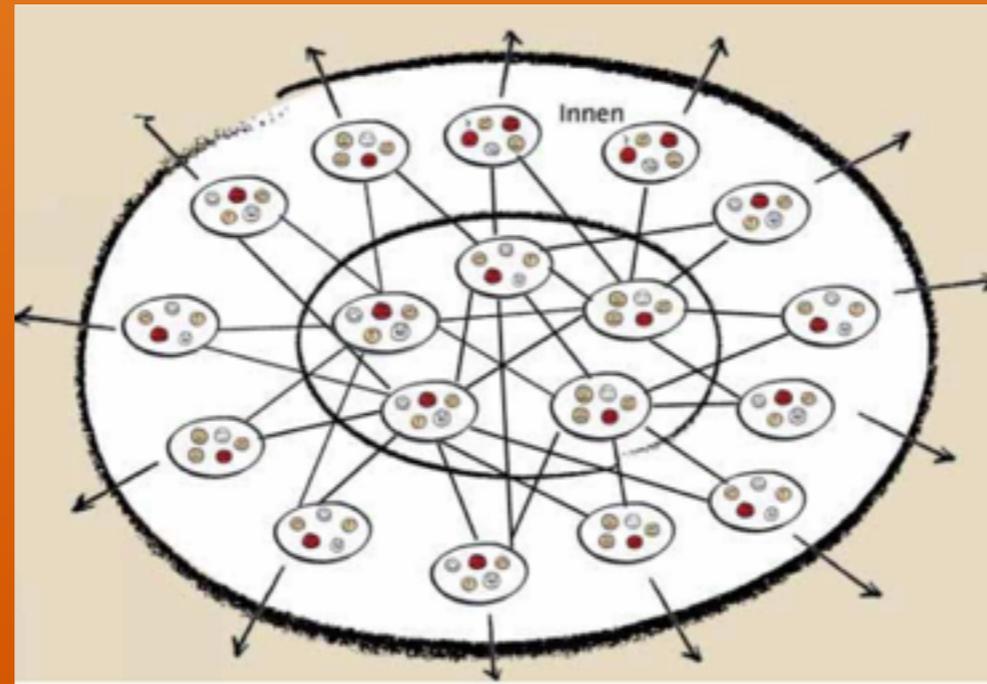


An die Stelle von Abteilungen treten hier Teams, die mit dem Unternehmen selbst in zwei Richtungen kommunizieren - es gibt einmal den Lead Link, der den „Gesamtkreis“ des Unternehmens vertritt - und dann den Rep Link, der keineswegs Reporting, sondern Represent heisst - der vom Team selbst als Vertretung zur Aussenwelt gewählt wird. Die Kreise ergeben sich durch die Beobachtung von Tensions, also von Spannungen - auf die in den Circles reagiert wird.



Wer soll die Docker-Container machen? **Das Team oder zentral?**

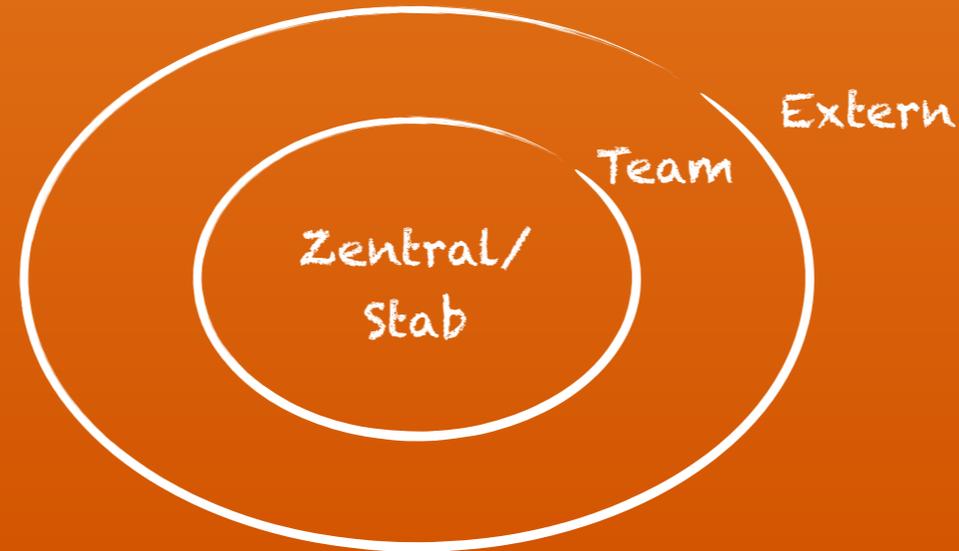
Am bekanntesten Beispiel - Zappos nämlich - ist zu sehen, dass es gar nicht so einfach ist, Aufgaben im Unternehmen zu lokalisieren, sondern erheblich Schmerzen verursacht.



In der deutschsprachigen Welt gibt es quasi als Konkurrenzveranstaltung aus dem sogenannten BetaCodex-Umfeld, heute in den Büchern „Organisation für Komplexität“ und „Komplexithoden“.

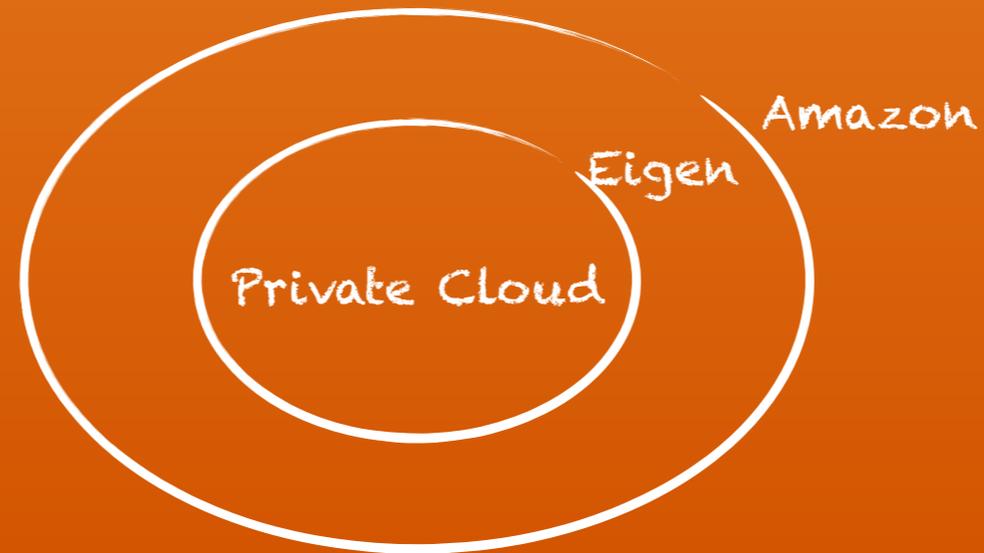
Interner Markt

Wer liefert eine **Leistung**
mittelfristig am **besten & billigsten?**



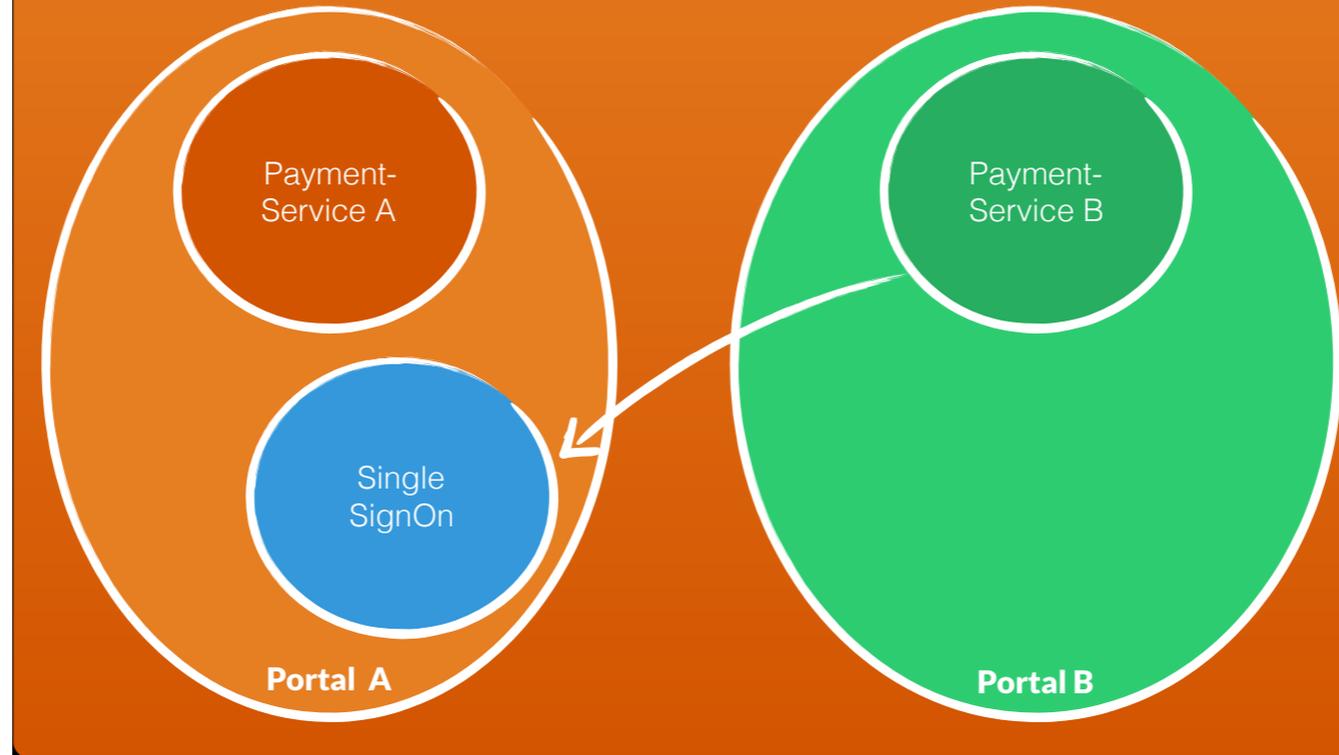
In den Komplexithoden nennt sich das Prinzip „Wertschöpfungsrechnung“, im Original kommt es aber aus dem Hause DM Drogeriemarkt - und hat einige esoterische Anteile, die Version von Betacodex ist tatsächlich deutlich mehr betriebswirtschaftlich orientiert und besser anzuwenden.

Jährlicher Abgleich Experimente



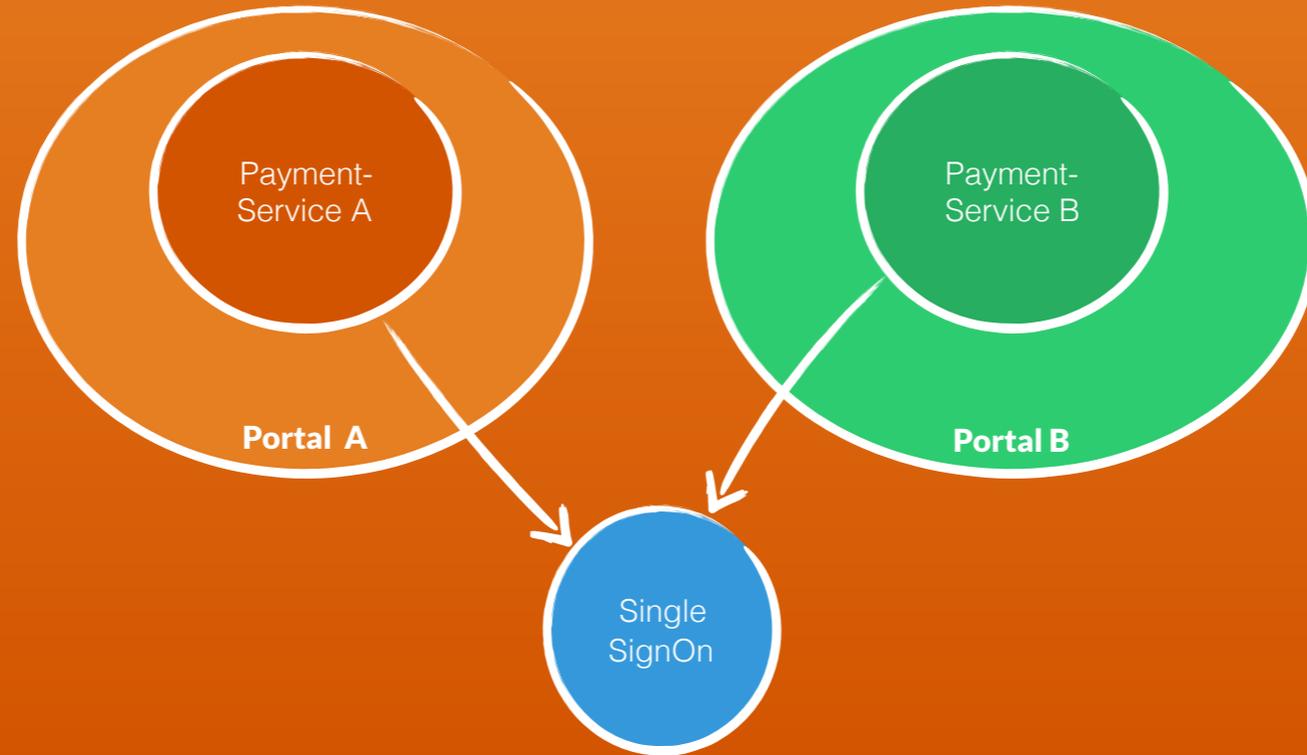
Das ganze kann kontinuierlich passieren, jährlich oder Quartalsweise. Meistens wird das von den Teams getriggert - sie können sich selbst entscheiden, woher sie welche Leistung einkaufen. Im konkreten Beispiel hat so ein Team zB die Wahl, ob sie selbst ein Hosting von Containern machen, ob sie dafür die zentrale Administration in Anspruch nehmen oder ob sich schlicht einen externen Service dafür nutzen.

Portale / MicroServices



Bei uns in der IT finden sich insbesondere im MicroService-Umfeld auch solche Konstrukte - jeder Client bzw. jedes Portal hat zwar die Wahl, Services selbst zu entwickeln, er hat aber jederzeit die Möglichkeit, auf einen Service des anderen Portals / des anderen Clients zuzugreifen.

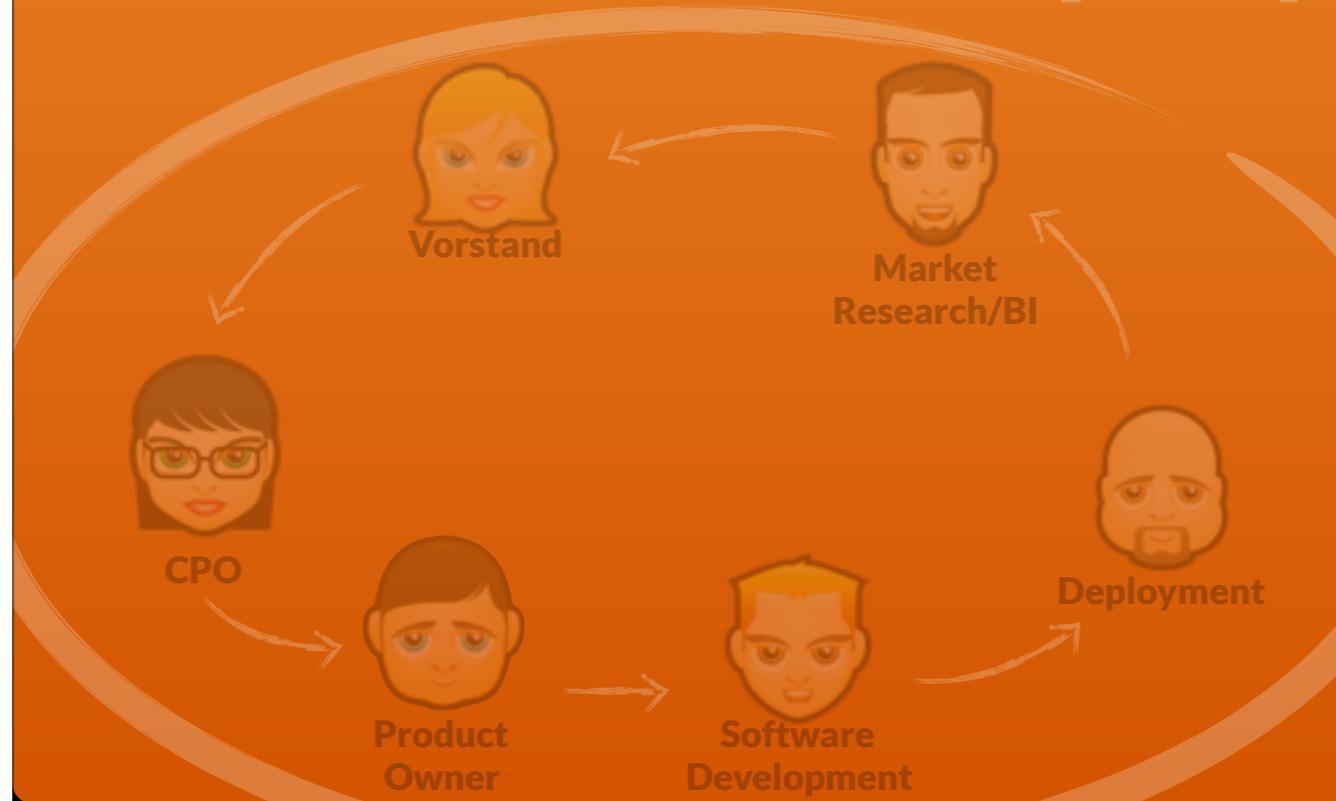
Portale / MicroServices



Und in dem Fall gibt es die Wahl - ob man das ganze bei der Organisationsstruktur lassen möchte, die ihn bisher betreibt, oder ob es einfacher ist, das ganze als zentralen Service auszulagern.

Also alles **ohne**
Governance und
voll-dynamisch?

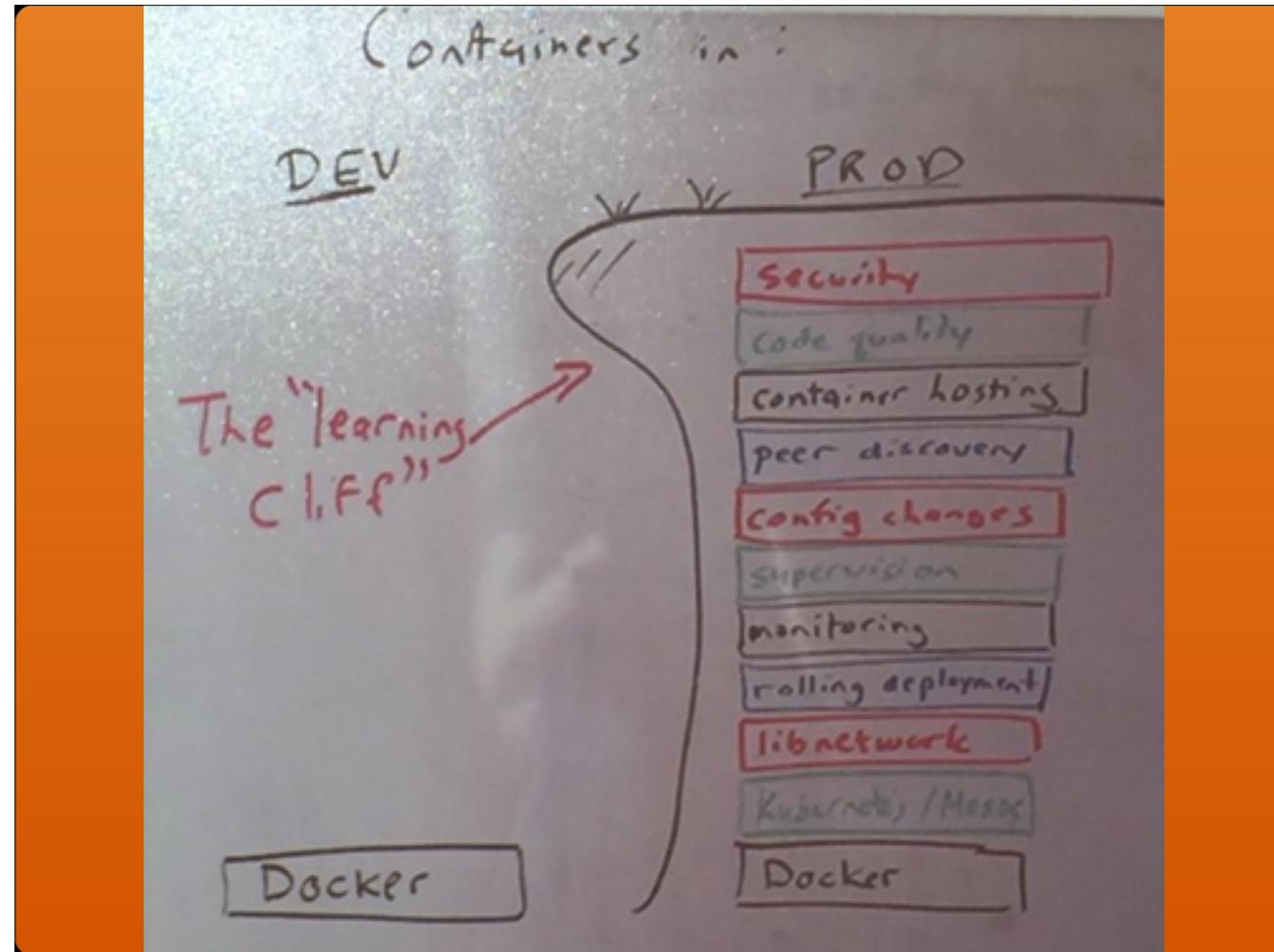
Data Driven Company



Auch wenn ich möglichst grosse Adaptions- und Innovationsfähigkeit möchte kann ich nicht in jeder Richtung Freiheitsgrade haben. Damit ich das nicht im Blindflug mache brauche ich Daten - und weil ich noch nicht immer weiß, welche Daten das in Zukunft sein sollen, erst mal möglichst viele.

Ok, also **Datensammlung**
für **Business Intelligence.**

Noch was?

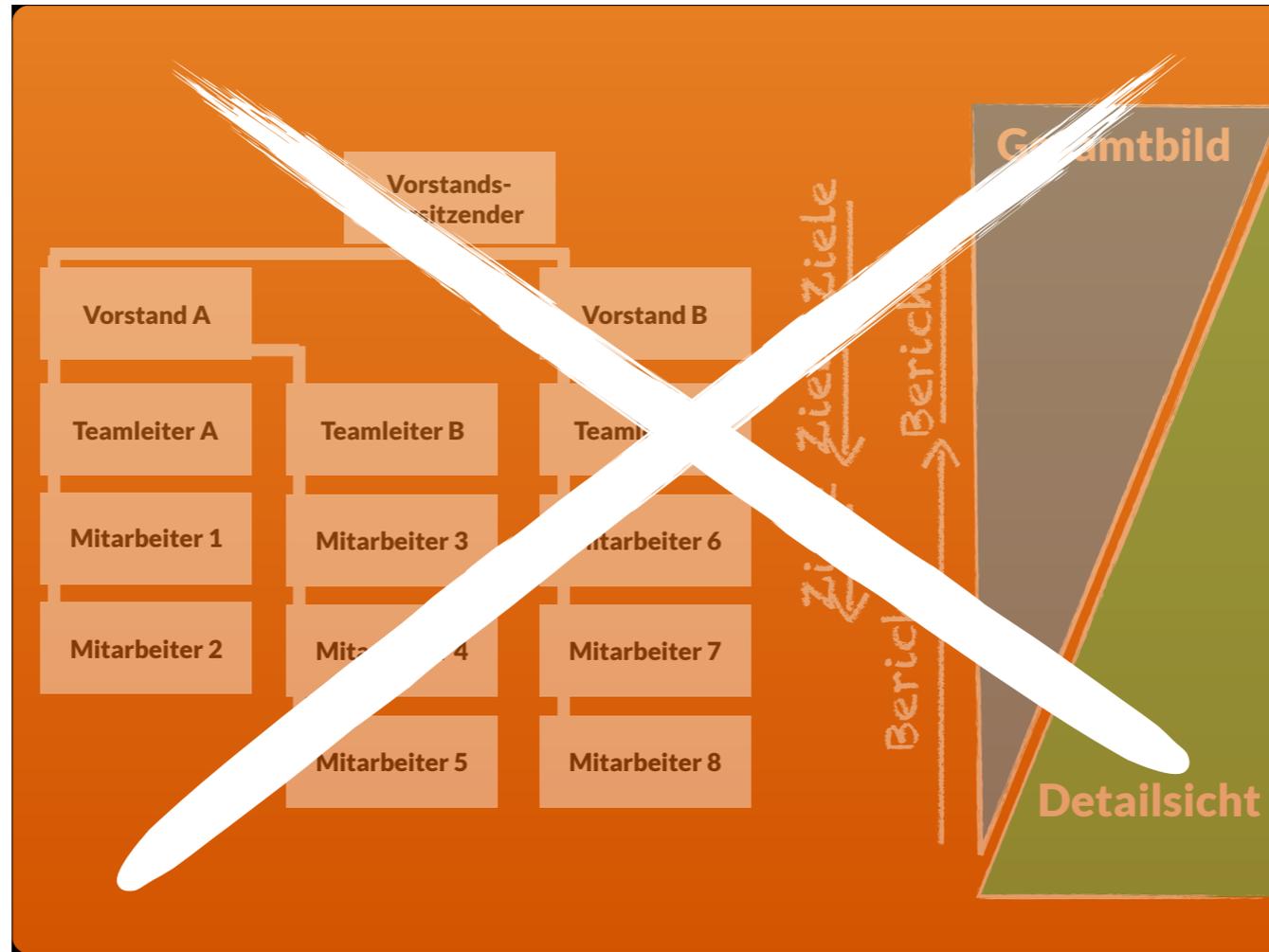


Vielleicht haben einige diesen Tweet auf Twitter gesehen - der Unterschied in der Wahrnehmung von Docker in Development und Produktion. Während wir in Development zwar Docker-Container bauen, sind in der Produktion auch Dinge wie Security, Container Hosting, Peer Discovery, Config Changes, Supervision, Monitoring, Rolling Deployments, Networksetup, Kubernetes oder Mesos zu beachten.

Ok, **das Knowhow** habe
ich, und die **Teams nicht**.

Dann **mache ich**
Ihnen **am besten**
Vorgaben?

Das Knowhow haben die dezentralen Teams nicht. Schliesslich müssen sie sich um viele Themen kümmern, nicht nur um das Setup von Services. Diese Kompetenz haben wir eher zentral. Also mache ich ihnen als Experte am besten Vorgaben, nach denen sie arbeiten müssen?



Leider gilt das gleiche, was für die Leitung des Unternehmens gilt auch für mich. Ich kann nur für die Fälle vorgeben, mit denen ich rechne. Was zukünftig neu dazukommen wird kann ich nicht beurteilen.



Erfreulicherweise kann man an dieser Stelle wieder von diesen Kollegen hier lernen. Konkret General von Moltke, in diesem Fall zitiert aus dem Art of Action von Stephen Bungay.

Alignment ↔ Autonomy

In militärischen Kreisen dachte man - wie in vielen Firmen auch - dass man entweder hohes Alignment oder hohe Autonomie haben kann, denn hohes Alignment braucht klaren und detaillierte Ansagen über das, was zu tun ist. Das hilft einem Einsatztrupp beim Militär aber nicht viel, wenn es nach dem Feindkontakt erst mal beim General anrufen muss, was denn als nächstes zu tun ist.

- Kontextknowhow
- Nutzungsknowhow
- Entscheidungskompetenz
- „Enabling“



General von Moltke sah das anders - wenn ich hohe Autonomie will, brauche ich hohes Alignment. Um im Docker-Beispiel zu bleiben - ich muss bei den Kollegen in den Teams das Knowhow aufbauen, um mit Docker gut umzugehen und gute autonome Entscheidungen treffen zu können, was ich zu beachten habe, was ich selbst machen möchte und was ich lieber extern oder zentral erledigen lasse.

- Kontextknowhow
- Nutzungsknowhow
- Entscheidungskompetenz
- „Enabling“



An der Stelle geht das den zentralen Admin-Abteilungen auch nicht besser als den Managern sonst - autonome Teams weist man nicht an, man gibt ihnen nur die Chance, das richtige zu tun.

Wird die **zentrale Administration** abgeschafft?



Bundesarchiv, Bild 103-1990-1129-500
Foto: o. Ang. | 1904

Also - hat die zentrale Administration noch eine Zukunft, oder geht es ihr mittelfristig wie dem Kutscher hinten auf der Kraftdroschke?

Natürlich nicht,
das war **nur Clickbait.**

Aber in **vielen Märkten**
wandeln sich die **Aufgaben**
mit **steigender Dynamik.**
DevOps ist der erste Schritt.

Das ist natürlich nicht der Fall. Aber überall dort, wo die Dynamik des Marktes einen bestimmten Grad überschreitet, ändert sich die Organisation - und damit auch die Rolle und die Aufgaben der zentralen Administration. DevOps ist der erste Schritt auf diesem Weg gewesen.

Steuerung- -
Dezentral++
Kooperation++
Enabling++
Data/Monitoring++
Hausmeister - -



Und dort passiert etwas, was der Änderung der Leitungs/Managementaufgaben im Unternehmen nicht ganz unähnlich ist. Man kann weniger Ansagen und Festlegen, wie die Dinge im Unternehmen zu laufen haben, welche Plattformen zu nutzen sind etc. Die Tätigkeiten werden an den Stellen, an denen mehr Dynamik gefragt ist dezentraler. Aus der klassischen Siloaufgabe wird im DevOps-Sinne eine Kooperative Aufgabe, die Enabling im Vordergrund hat.

Und Unternehmen, bei denen die Systemadministration als nebensächliche Hausmeisteraufgabe angesehen wird, werden in Summa weniger - weil sie nicht dynamiktauglich sind :-)

Mit genügend guter Kommunikation, wie über das Internet, und Führung ohne Zwang sind viele Köpfe immer besser als einer.

Mit einer hinreichend großen Gruppe von Betatestern und Mitentwicklern wird fast jedes Problem schnell erkannt und die Lösung von jemandem gefunden.

Fast so gut wie eigene gute Ideen zu haben, ist es, gute Ideen von den Benutzern zu erkennen. Manchmal ist letzteres besser.

Fragen: **@johannhartmann**
hartmann@mayflower.de

Slides mit Tonspur:
<http://slideshare.net/johannhartmann>

Wer Fragen oder Kritik hat, oder mich einfach nur mal trollen möchte - bitte per Mail oder Twitter Bescheid sagen. Praxiserfahrungen kann ich in Mengen beisteuern.

Diese Folien finden sich wie alle anderen auch mit Volltext auf dem Slideshare.

Die drei Sätze oben sind direkt aus den 19 Thesen der Kathedrale und dem Bazaar.