



OST
Ostscheizer
Fachhochschule

Vorlesung PMQM

8 – Testing and Configuration

Werner Staub

Informatik

Ziele der heutigen Vorlesung

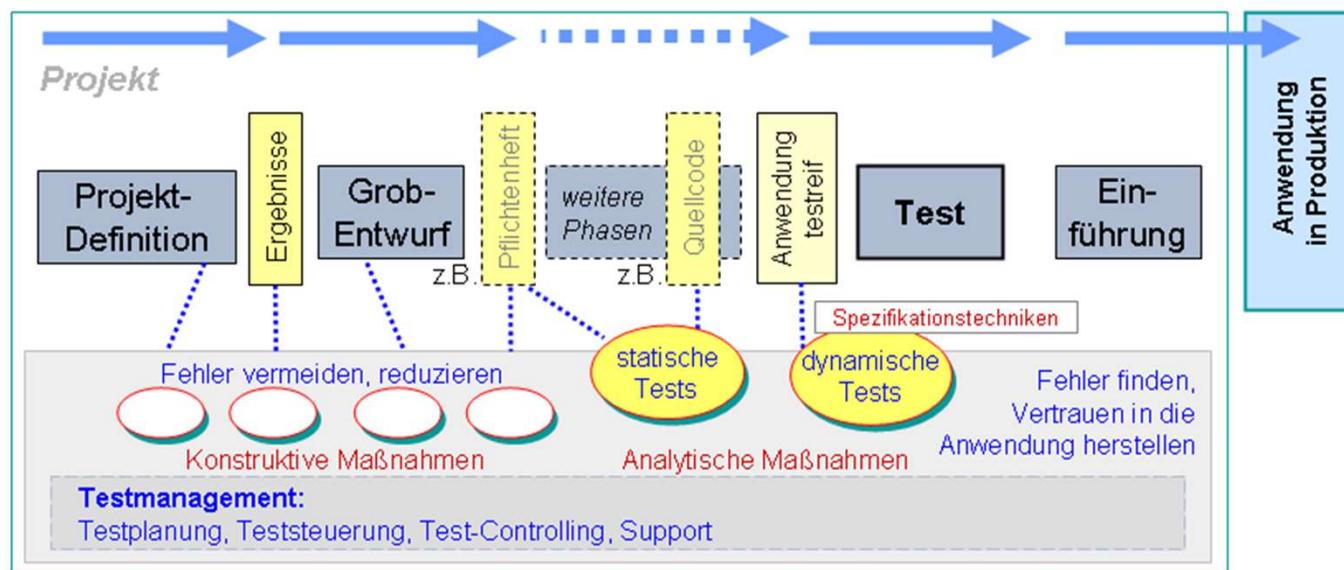
- Lerninhalte
 - Teststrategie
 - Testfälle
 - Testdaten
 - Testorganisation
 - Konfigurationsmanagement
- Ziele: Die Studierenden...
 - kennen Verfahren, um Software möglichst effektiv zu testen
 - unterscheiden die Teststufen und ihre Möglichkeiten
 - verbessern mit dem Wissen ihre zukünftigen Projekte
 - verstehen wie Konfigurationsmanagement funktioniert und für was es wichtig ist

Software Qualität

Randbemerkung:

Im Prinzip kann man dieses Konzept im agilen Projekt Management auf jeden Sprint anwenden.

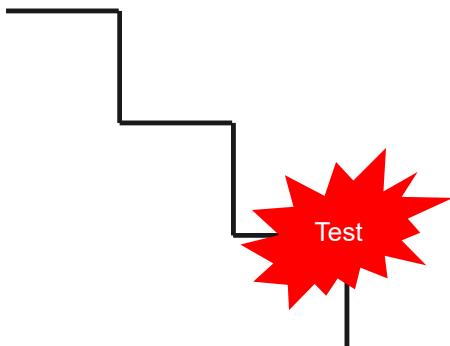
Testen als ganzheitlicher Ansatz im Software-Qualitätsmanagement



<https://de.wikipedia.org/wiki/Softwaretest>

Der Testprozess im Überblick

Klassisch nach Wasserfall



Agil oder nach DevOps



- Unabhängig ob klassisch oder agil:
getestet werden muss immer
- Substantiell ist die Methodik:
 - Test-Planung, -Organisation, -Durchführung
- Notwendig sind:
 - Fachwissen in der Materie
 - passende Testfälle
- Unterstützt durch:
 - effiziente Werkzeuge für das Testmanagement
 - Testautomatisierung

Werkzeuge: Siehe auch
Konfigurationsmanagement

Was beinhaltet der Testprozess



IT-Consultants mit Autismus gesucht

Das Potential von Autisten wird stark unterschätzt, findet ein Berliner IT-Unternehmer. Er bietet Fehler-Checks für Computerprogramme an und stellt dafür gezielt Menschen mit Asperger-Syndrom ein. Sie graben sich wie Archäologen durch das Buchstaben- und Zahlengewirr - mit Erfolg.



Fotos

Quelle: spiegel.de

Tests müssen frühzeitig eingeplant und die Strategie muss so festgelegt sein, dass möglichst viele Anforderungen überprüft werden können. Die Testfälle werden im Vorfeld festgelegt und decken das Testfeld möglichst optimal ab (weniger ist mehr).

Es kann nie 100% getestet werden!

Die Durchführung und Protokollierung der Tests ist eine Aufgabe, welche eine hohe Konzentration und Disziplin verlangt.

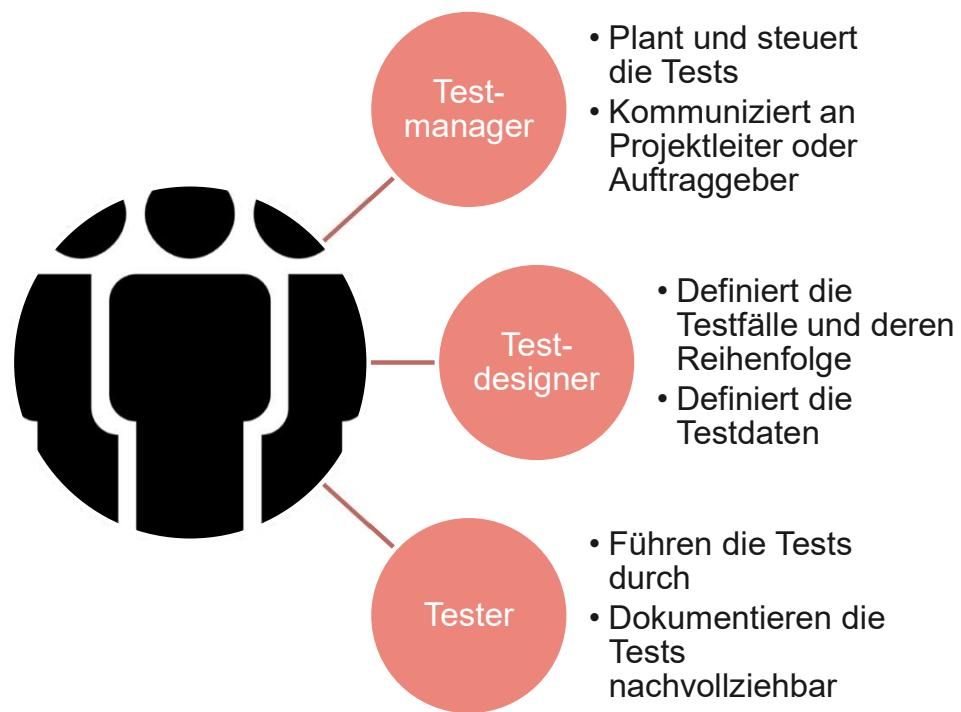
Nutzen: Qualitätssicherung sowie Abnahme von Werken

Es kann nie 100% getestet werden

- Das heisst:
- Mein Produkt geht nie fehlerfrei in die Produktion!
- Je grösser der Release,
umso kritischen ist die Einführung
und die erste Phase danach
- Incident Management, Pickett
- Endlich ist der Release fertig
jetzt kann ich ausruhen
funktioniert nicht



Testteilnehmer



In der Literatur erscheinen weitere Teilnehmer wie Testkoordinator, Testautomatisierer, Testsystem Administrator, etc.

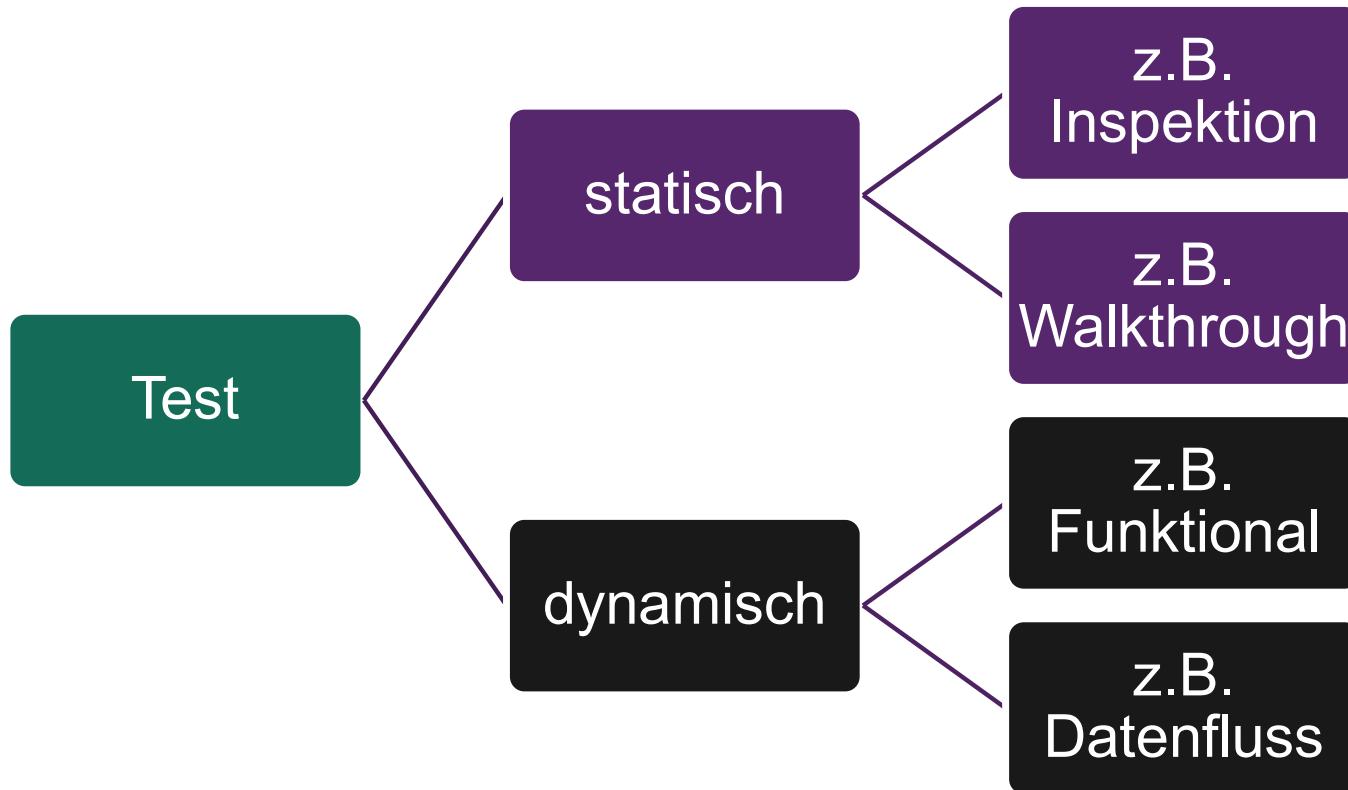
Folgende Eigenschaften müssen Tests erfüllen

- Gemäss Frank Witte, Autor des Buches «Testmanagement und Softwaretest», welches für die Vorbereitung der Zertifizierung zum Testmanager nach ISTQB empfohlen wird, müssen Tests folgende Eigenschaften erfüllen:

Reproduzierbarkeit	Planbarkeit	Wirtschaftlichkeit	Risiko- und Haftungsreduktion
<ul style="list-style-type: none">• Die Ergebnisse entstehen systematisch und nachvollziehbar und sind nicht Zufällen oder unbekannten Randbedingungen unterworfen.	<ul style="list-style-type: none">• Es ist möglich, den Testaufwand und den Testnutzen (gefundene Fehler) vorherzusagen.	<ul style="list-style-type: none">• Das Verhältnis von Testaufwand und Testnutzen wird stetig optimiert.	<ul style="list-style-type: none">• Der systematische Test gibt Sicherheit, dass kritische Fehler nicht auftreten. Vertragsstrafen oder Rückrufe vermieden werden

Testfälle müssen darauf ausgelegt werden

Typen von Testfälle

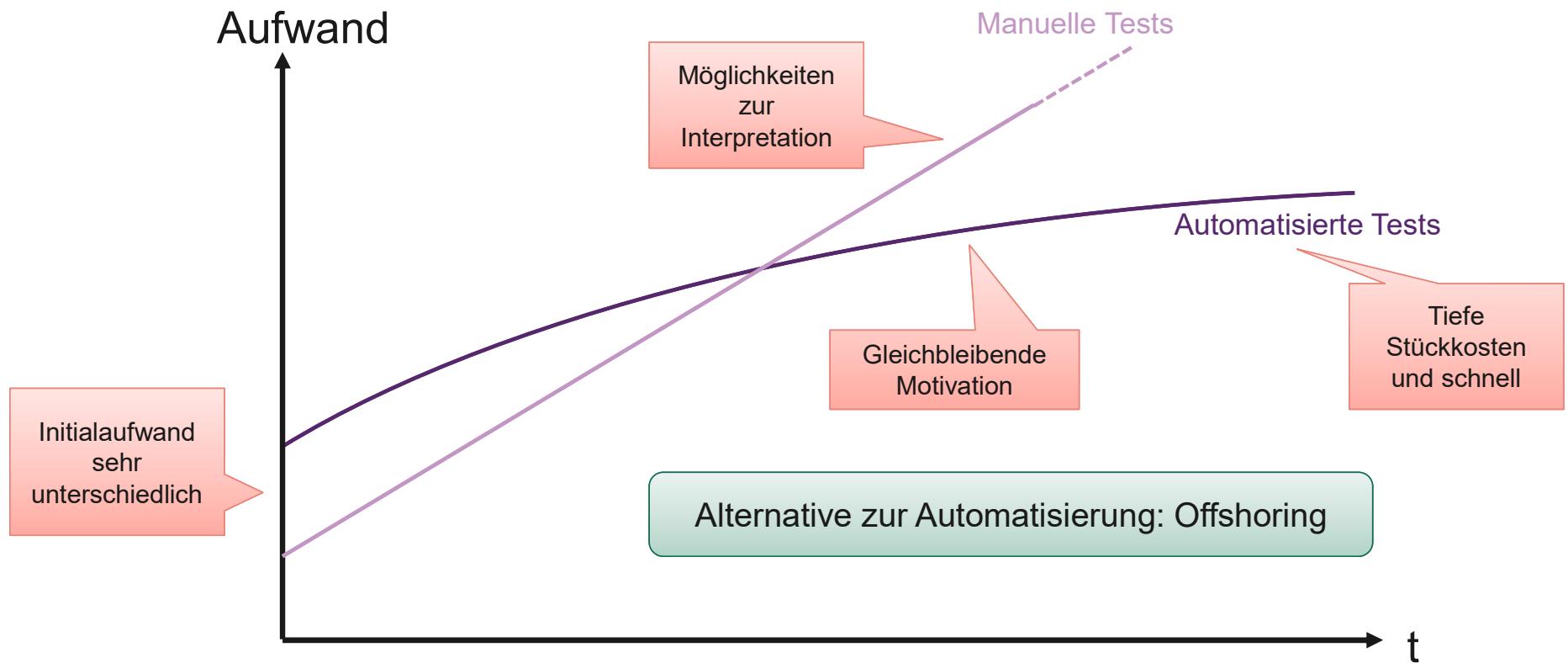


Spezialgebiet «nicht-funktionale» Tests

- Last- und Performancetests dynamisch (mit Tools wie NeoLoad, HP Loadrunner, usw)
- Sicherheitstests
Pennetrationstests statisch und dynamisch (ev. durch spezialisierte Firmen)
- Desaster Recovery Tests
Business Continuity Tests statisch und dynamisch (aufwändig und teuer)
- Usability Tests statisch und dynamisch
(im agilen Projekt permanent durch die User getestet)

Achtung:
Bedingt, dass «nicht-funktionale»
Anforderungen vereinbart sind.

Testautomatisierung

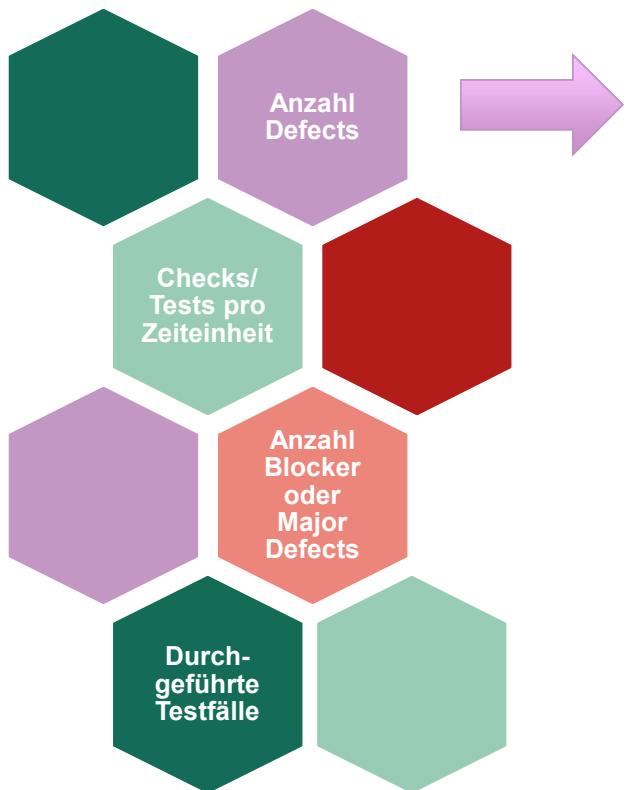


Achtung: Wer misst, misst Mist....



- Verwenden Sie beim Testen stabile Bedingungen, sonst erhalten Sie gerade bei Performance Tests grosse Schwankungen.
- Weiter müssen Sie berücksichtigen, dass Testumgebung und das Vorhandensein von Messungen (z.B. durch intensives Logging) die Resultate ebenfalls beeinflussen, auch sogenannter «probe effect» (**wer misst beeinflusst das Systemverhalten**).
- Versuchen Sie das Testsystem so gleich wie möglich zum produktiven System zu halten.

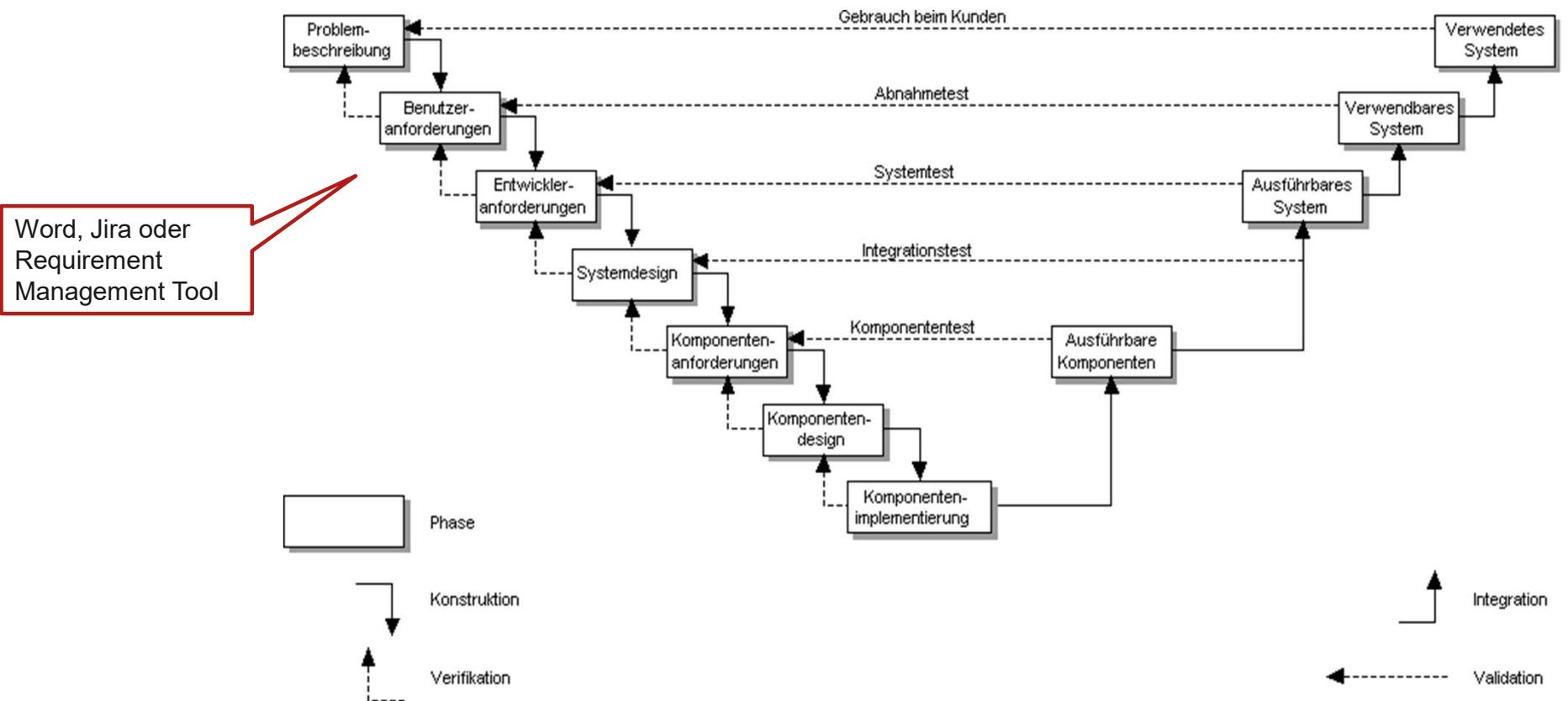
Kennzahlen



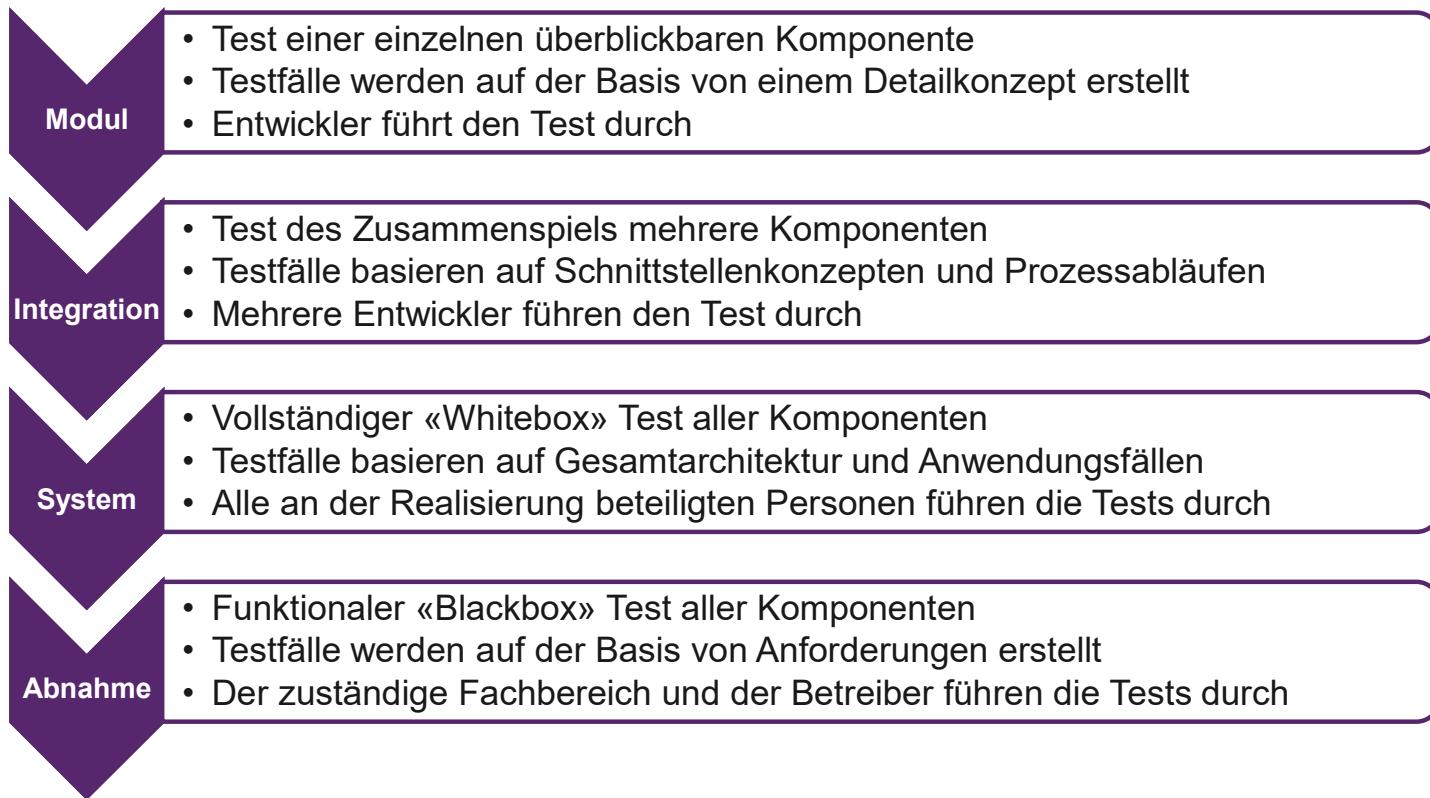
☐ Hard Facts – Defects Historie und Details



Das V-Modell im Bestreben einer Früherkennung



Teststufen im V-Modell



Positiv- oder Negativtest

- **Positivtest:**
 - Der korrekte Fall wird getestet. Das heisst, wenn die Inputs und die Einstellungen richtig sind, kommt das richtige Ergebnis heraus.
- **Negativtests** (Provokationstest, Robustheitstest, Falsifizierender Test oder Schlecht-Test):
 - Der Negativtest prüft, ob die Anwendung auf eine (falsche) Eingabe oder Bedienung, die nicht den Anforderungen an die Anwendung entspricht, erwartungsgemäss reagiert, z. B. durch eine Fehlermeldung.
 - Es werden absichtlich ungültige Werte eingegeben, Masken werden nicht oder nur unvollständig ausgefüllt, Schnittstellen werden mit falschen Werten beliefert oder die Datenbank wird abgeklemmt

Das Projektteam weiss wie es funktionieren soll und bedient es daher instinktiv richtig.
Der User kennt das System nicht und macht daher unvorhergesehene Fehler.

Inhalte nach dem Standard IEEE829

- Übersicht
 - Testkonzept (**test plan**): Das Testkonzept bestimmt Abgrenzung, Vorgehensweise, Mittel und Ablaufplan der Testaktivitäten. Es bestimmt die Elemente und Produktfunktionen, die getestet werden sollen, die Testaufgaben, die durchgeführt werden müssen, das verantwortliche Personal für jede Aufgabe und das Risiko, das mit dem Konzept verbunden ist.
- **Test-Spezifikation** (test specification)
 - Testentwurfspezifikation (**test design specification**): Die Testentwurfspezifikation verfeinert die Beschreibung der Vorgehensweise für das Testen der Software. Sie identifiziert die Produktfunktionen, die von den Tests abgedeckt werden müssen. Sie beschreibt weiterhin die Testfälle und Testabläufe, die benötigt werden, um Tests zu bestehen und spezifiziert die Bestehens- oder Verfehlenskriterien der einzelnen Produktfunktionen.
 - Testfallspezifikation (**test case specification**): Die Testfallspezifikation dokumentiert die zu benutzenden Eingabewerte und erwarteten Ausgabewerte. Testfälle sind vom Test-Design getrennt. Dies erlaubt die Verwendung der Testfälle in mehreren Designs und die Wiederverwendung in anderen Situationen.
 - Testablaufspezifikation (**test procedure specification**): Beschreibung aller Schritte zur Durchführung der spezifizierten Testfälle und Implementierung des zugehörigen Test-Designs.

Inhalte nach dem Standard IEEE829

- **Testbericht** (**test reporting**)
 - Testobjektübergabebericht (**test item transmittal report**): Der Testobjektübergabebericht beschreibt die Übergabe der Testfälle für den Fall, dass getrennte Entwicklungs- und Testteams eingebunden sind oder für den Fall, dass ein offizieller Zeitpunkt für den Beginn einer Testausführung erwünscht ist.
 - Testprotokoll (**test log**): Das Testprotokoll dient zur Aufzeichnung der Ereignisse während einer Testausführung.
 - Testabweichungsbericht (**test incident report**): Es beschreibt alle Ereignisse, die während einer Testausführung auftreten und weitere Nachprüfungen erfordern.
 - Testabschlussbericht (**test summary report**): Fasst die Testaktivitäten zusammen, die mit einer oder mehreren Testentwurfsspezifikationen zusammenhängen.

Selbst HERMES verfolgt einen minimalen Ansatz resp. minimale Vorgaben

Testen

- Das Testen konzipieren, vorbereiten, durchführen und dokumentieren

Initialisierung	Konzept	Realisierung	Einführung
	Testkonzept erarbeiten	Testinfrastruktur realisieren Test durchführen	Test durchführen Testkonzept und -infrastruktur überführen

Quelle: hermes.admin.ch



Inhaltsverzeichnis	
1	Testziele.....2
2	Testobjekte
3	Testarten.....2
4	Testabdeckung
5	Testrahmen.....2
5.1	Testvoraussetzungen.....2
5.2	Fehlerklassen.....2
5.3	Start- und Abbruchbedingungen.....3
6	Testinfrastruktur.....3
6.1	Testsystem.....3
6.2	Testdaten
6.3	Testhilfsmittel.....3
7	Testfallbeschreibungen.....3
8	Testplan

Hinweis: In HERMES Projekten befinden sich die Testfallspezifikationen direkt im Testkonzept

Hinweis: Üblicherweise verlinkt man die Anforderungen mit den Testfällen

Beispiel eines Testfalls

Nummer	Voraussetzung	Testschritte	Testdaten	erwartetes Resultat
1 Eintritt intern, Eintritt Heute im IAM-System wird eine Identität mit allen Attributen angelegt, wobei Extern gleich nein und das Eintrittsdatum ist Heute				
		1. Neue Identität im HR anlegen (alternativ direkt HR-File bearbeiten)	1. Vorname, Nachname, Intern, Vorgesetzter, Eintrittsdatum = Heute (...) und Rollen von 4	
		2. HR Export anstoßen		
		3. HR-File kontrollieren	3. HR-EXPORT.csv in LOCATION_PATH	3. Datensatz mit allen Attributen vorhanden
		4. HR Synchronisation Zeitplan (<ZEITPLAN-HR-SYNC>) im Designer starten	4. Rollen: (...)	4. HR Export wird archiviert
		5. Die JobQueueInfo überprüfen und abwarten bis der Synchronization Job und weiteren Jobs verarbeitet wurden		
		6. Im Manager in der Übersicht die zugewiesenen Rollen überprüfen		6. alle Rollen sind zugewiesen
		7. Im Manager in der Stammdatenbearbeitung die Attribute überprüfen	7. vgl. 1, nicht zeitweilig deaktiviert	7. Identität mit allen Attributen angelegt.
		8. Im OWA einloggen als Vorgesetzter von erstellter Identität einloggen	8. OWA-Adresse: <ADRESSE-OWA> 8. E-mail: EXAMPEL@DOMAIN	8. E-Mail an Linievorgesetzter mit Zuweisung neuer Person; E-Mail an Linievorgesetzter mit Username, Passwort und Aktivierungsdatum von neuer Person
		9. Im Webportal als Vorgesetzten von erstellter Identität einloggen	9. Adresse-Webportal: (...)	9. Möglichkeit für Person eine Bestellungen auszulösen ist vorhanden; Person ist unter "Meine Mitarbeiter" ersichtlich
		10. In allen entsprechend angebundenen Zielsystemen die Existenz vom Account überprüfen	10. (...)	10. zugehöriger Account existiert in allen Zielsystemen

Aufbau einer Fehlermeldung

Die schematisch festgelegte Fehlermeldung unterteilt sich in 3 Bereiche:

- **Problembeschreibung:** Schilderung der aufgetretenen Errors im Verlauf des Tests.
- **Identifikation:** Testperson, Software, Testumgebung (Rahmendaten), Fehlerstatus.
- **Klassifikation:** Fehlerklasse und Priorisierung.

Von besonderer Relevanz ist neben der Problembeschreibung die Klassifikation bzw. die Einstufung der Meldung, da hier zum einen der erkannte Fehler priorisiert wird zur Darstellung der Wichtigkeit und der damit verbundenen Lösung und zum anderen eine Kritikalität in Form der Fehlerklasse festgehalten wird.

Fehler, die Rückfragen vom Entwickler auslösen, deuten auf ein Defizit bei der Fehlerbeschreibung hin. Diese Fehler sollten ebenfalls gezählt und klassifiziert werden.

Messungen, die der Entwickler benötigt um die Fehlerquelle aufzuspüren (Logfiles, Traces, aber auch Bilder die das Fehlverhalten erklären), sind dem Fehler hinzuzufügen.

Ein einzelner Fehler erzeugt Prozesskosten von ca. 500 bis 2000,- CHF, je nach Phase in der er entdeckt wird und je nach Qualität der Fehlerbeschreibung. Das Fehlermanagement ist ein weiterer Bereich, in dem eine Vereinheitlichung und Standardisierung gewaltige Potenziale freilegen würde. Auch dem Management sollte dabei bewusst sein, wie hoch die Kosten und die Lebensdauer eines Fehlers sind. Es gibt Fehler, die mehrfach behoben, geschlossen, wieder eröffnet werden, teilweise weil die Beschreibung von Anfang an nicht vollständig war oder bei der Behebung des Fehlers nicht nach rechts und links geschaut wurde. Einige Fehler geistern monatelang durchs System und werden immer wieder zwischen Testern und Entwicklern hin- und hergeschoben.

Beispiel einer Fehlerklassifikation aus einem Vertrag

15 Abnahmebestimmungen

15.1 Die bei der Abnahme festgestellten Mängel werden nach folgendem Schema klassifiziert:
Schweregrad Beschreibung

- 1 Erheblicher Mangel, d.h. die in Frage stehende Zielsetzung wird nicht erreicht oder die Aufnahme des produktiven Betriebs ist nicht möglich.
- 2 Mittlerer Mangel, d.h. die Fortsetzung des Projekts und/oder die Aufnahme des produktiven Betriebes ist nur mit wesentlicher Behinderung möglich.
- 3 Leichter Mangel, Betrieb ist ohne Behinderung möglich: Ein Workaround ist bis zur Behebung des Fehlers möglich.
- 4 Unerheblicher Mangel, d.h. die Fortsetzung des Projekts und die Aufnahme des produktiven Betriebes ist ohne Behinderung möglich.

15.2 Ein erheblicher Mangel! liegt auch vor, wenn:

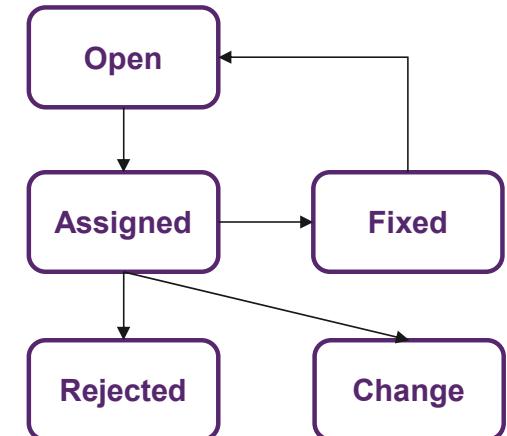
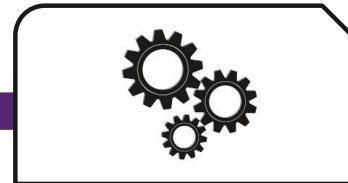
- a) 3 Mängel des Schweregrads 2 vorliegen oder
- b) 10 Mängel des Schweregrads 3 vorliegen.

15.3 Die Abnahme ist bestanden, wenn keine erheblichen Mängel vorliegen. Die Abnahme ist zu protokollieren.

Fehler-Status



Aktuelles Test-Set



und im agilen Projekt?

Agile Testkonzepte

ORGANISATION UND ORGANISATORISCHES / 8/05/2016

Bild: Public Domain Images/Amanda Mills - CC0 1.0

Zu den wohl unagilsten Dingen die mir auf verschiedenen Projekten begegnet sind gehörten die Testkonzepte mitsamt der damit verbundenen Testmanager. Die gesamten Dokumente waren durchzogen von Wasserfall, V-Modell, Command & Control, Dokumentationszwängen und ähnlichen Management-Relikten des zwanzigsten Jahrhunderts.

<https://www.lean-agility.de/2016/08/agile-testkonzepte.html>

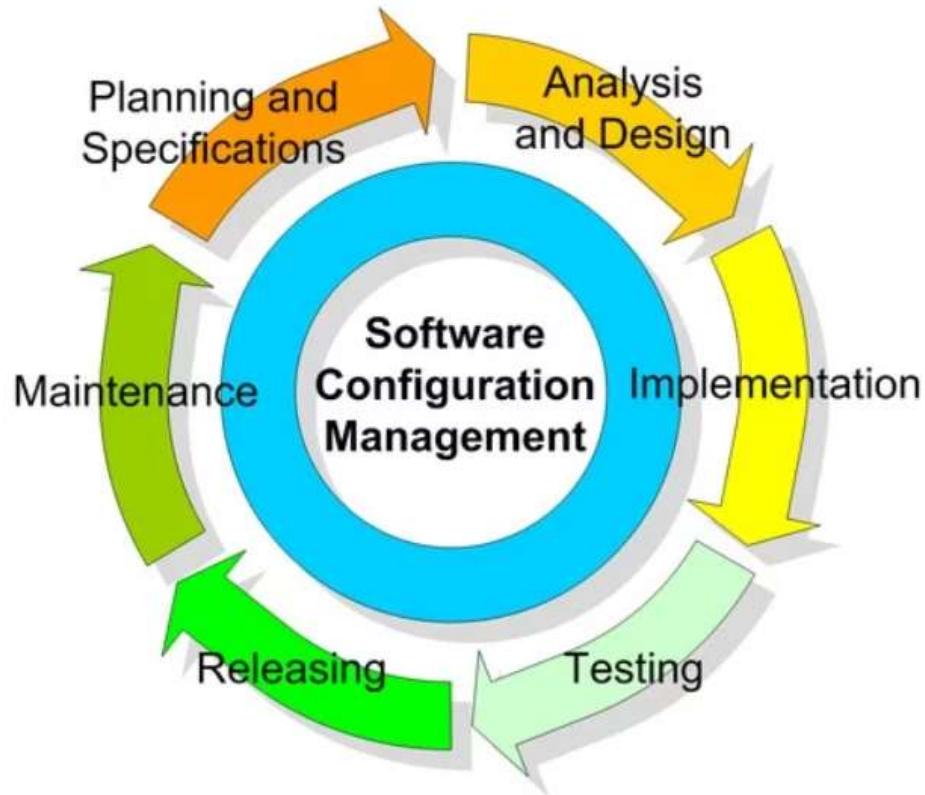
Agiles Testen

Im Grundsatz ist es dasselbe. Es muss geklärt sein, wer was wann macht.

- Was bedeutet Testen im agilen Umfeld? → so früh und so umfassend wie möglich
- Wer trägt QA-Verantwortung? → alle
- Was wird getestet? → sowohl die neuen Features als auch regressiv alle alten
- Wie wird getestet? → möglichst automatisiert
- Wann beginnt die Testphase? → ganz am Anfang, bzw. es gibt sie nicht gesondert
- Welche Tools werden eingesetzt? → z.B. Jenkins, Fitnesse, usw.
- Auf welchen Umgebungen wird getestet? → möglichst produktionsnah
- Wie lang sind die Testzyklen → idealerweise maximal ein Tag, daher automatisch
- Gibt es spezielle Test-Arten? → z.B. Lasttests, Penetrationstests, A/B-Tests, usw.
- Wie werden die Ergebnisse dokumentiert? → im Idealfall durch das Test-Tool selbst

<https://www.lean-agility.de/2016/08/agile-testkonzepte.html>

Konfigurationsmanagement



39

Werner Staub, PmQm, Rapperswil, Herbst 2022

Wozu braucht es Konfigurationsmanagement?

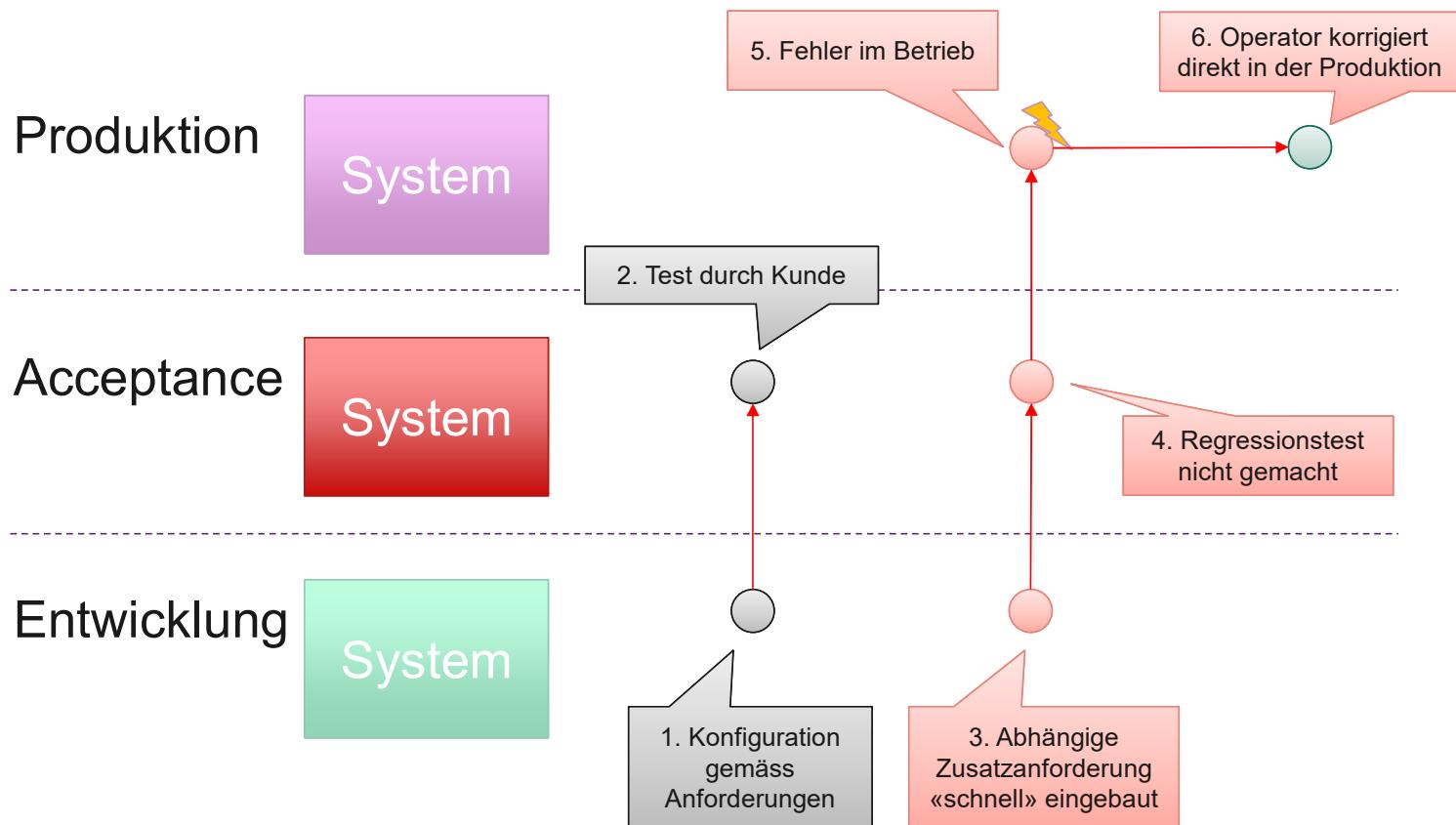
- Disziplin übergreifende Überwachung der Arbeitsschritte in der Entwicklung und in der Produktpflege
- Sicherstellung der qualitativ richtigen Lieferobjekte
- Qualitativ richtige Auslieferung der Lieferobjekte
- Aufzeichnung der Entwicklungsgeschichte
- Strukturierung des Änderungswesens
- Überwachung der Änderungen und der Änderungsprozesse
- Fehlerkorrekturen und Festlegung der Auswirkungen auf einzelnen Versionen
- Konfigurationsmanagement ist ein wichtiger Pfeiler des Qualitätsmanagement

Bitte mach doch mal schnell....

Typische Fehler in der Software-Entwicklung, welche mit einem sauberen Konfigurationsmanagement verhindert werden können:

- Software lässt sich sehr leicht ändern, dies verleitet zu nicht dokumentierten und schnellen Reparaturen, teilweise sogar direkt auf dem Produktionssystem
- Artefakte werden stetig geändert und weiterentwickelt, aus Unachtsamkeit könnten alte Artefakte verwendet werden
- Mehrere Personen arbeiten am gleichen Code und überschreiben sich gegenseitig

Beispiel aus der Praxis: Versäumnisse beim Konfigurieren



Releases und ihre Versionsbezeichnung

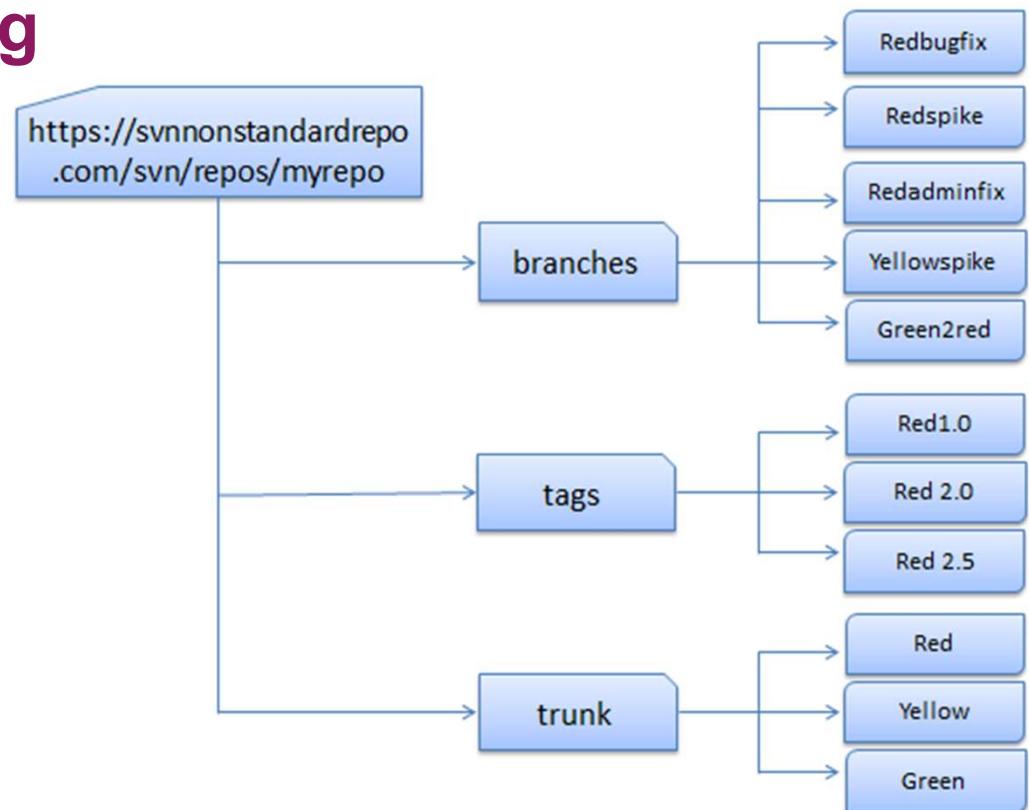
Es lohnt sich, zu Beginn des Projekts festzulegen, wie die Releases und Versionen bezeichnet werden. Es gibt keinen gängigen Standard dazu. Bei www.semver.org gibt es eine Empfehlung für die Softwareentwicklung.

MAJOR.MINOR.PATCH z.B. 1.2.3

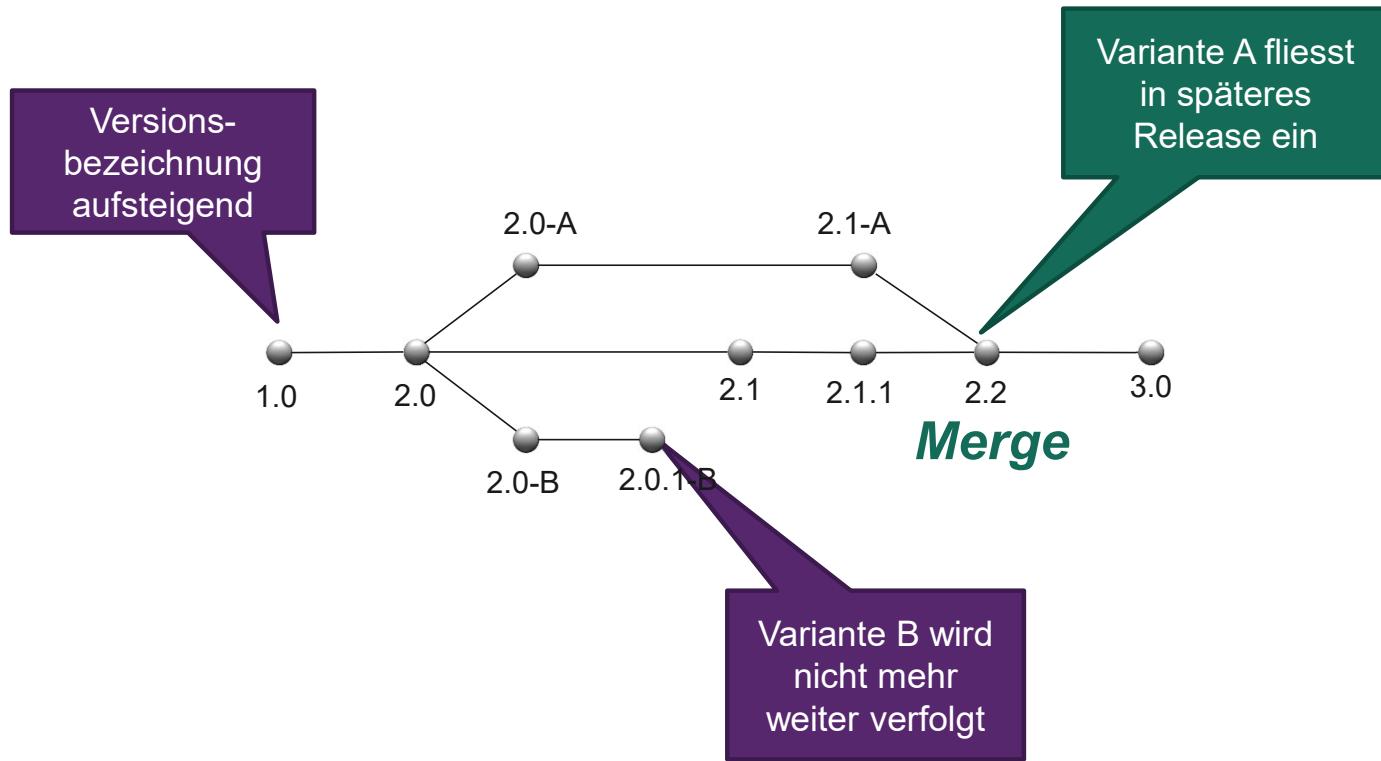
1. MAJOR version when you make incompatible API changes
2. MINOR version when you add functionality in a backwards-compatible manner
3. PATCH version when you make backwards-compatible bug fixes

GIT: Trunk, Branch und Tag

- Der Trunk (Stamm) enthält die aktuelle Entwicklungslinie, hier wird in der Regel ausgecheckt und weitergearbeitet
- Der Branch (Ast) enthält alternative Entwicklungszweige oder Varianten
- Ein Tag enthält ein unveränderlicher (eingefrorener) Stand des Produkts



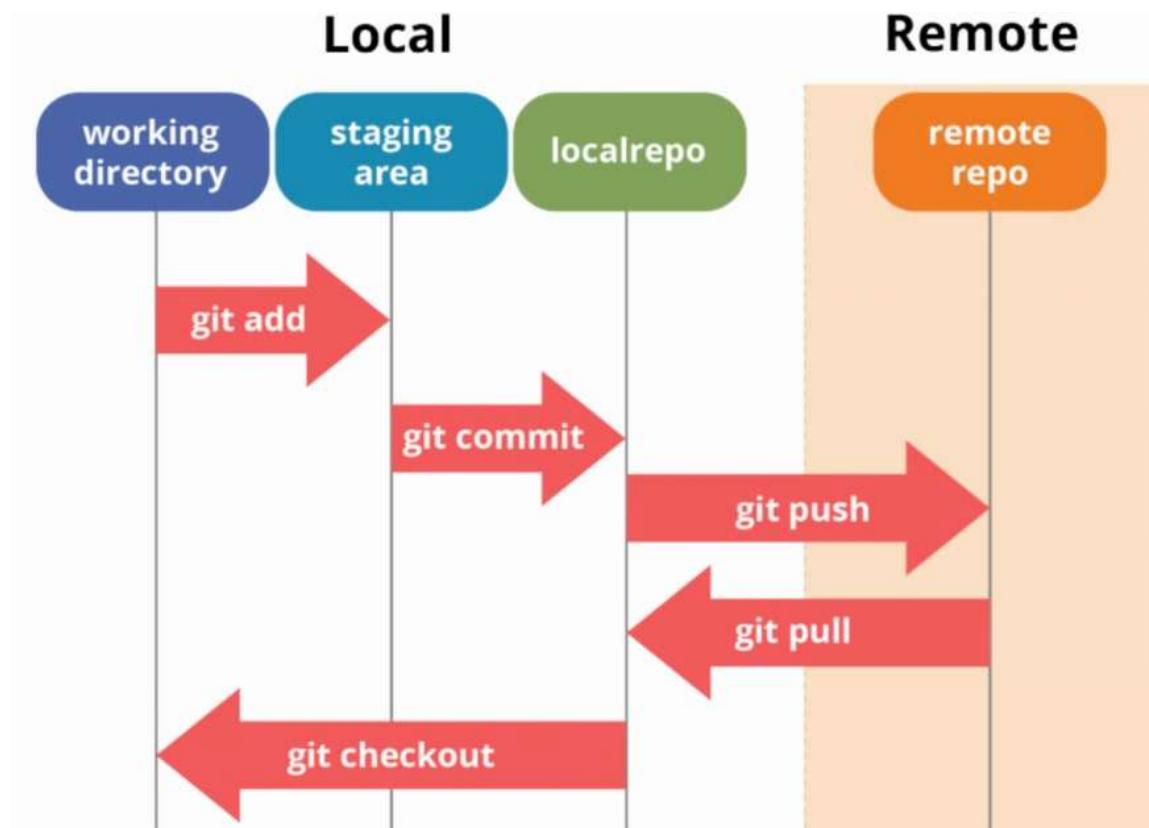
Varianten in Form von Branches



Slang im Konfigurationsmanagement mit GIT

Commit	Records a bunch of file changes in the local repository
Push	Transfer the local repository to the remote repository
Pull	Transfer the remote repository to the local repository
Force Push	Transfer the local repository to the remote repository without handle conflicts gently (override)
Pull Request	Requesting the target repository to grab your changes
Merge	Combine two trees

GIT Commands - Basis



Merge Strategien

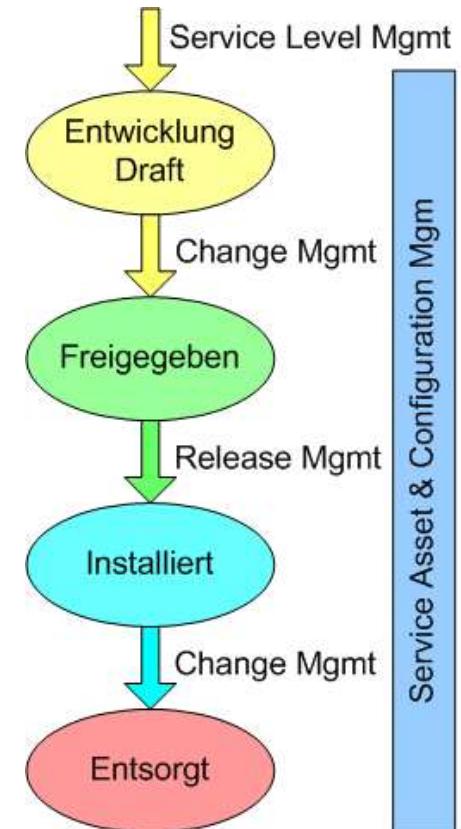
- Sperren «Check-Out»
 - Die Artefakte werden solange gesperrt, wie sie in Bearbeitung sind
 - Paralleles Arbeiten nicht wirklich möglich
 - Sicher und immer möglich
 - Keine Vermischung innerhalb eines Artefakts, nur ausserhalb
- Mischen
 - Automatischer Merge des Inhaltes
 - Dezentrale unabhängige Entwicklung möglich
 - Steckt voller Unsicherheiten, nur möglich wenn Source vorhanden
 - Vermischung benötigt Absprache zwischen Entwickler
- Manuelles Mischen
 - Zuständige Stelle mischt manuell und verwendet die «guten» Teile
 - Bei kompetenter Stelle nur Vorteile ausser Zeitfaktor

Vorteile eines guten Tools

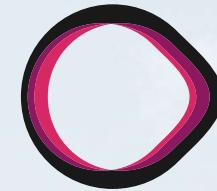
- Verwaltet alle Komponenten an einem Ort, z.B. mit einem virtuellen Dateisystem über mehrere Standorte hinweg
- Änderungen können identifiziert und beurteilt werden
 - Annehmen
 - Verwerfen
- Macht Unterschiede zwischen Versionen sichtbar, mittels Compare-Funktion oder automatischem Merging
- Wiederherstellung früherer Zustände
- Mit Tags können besondere Versionen, die einen speziellen Zwischenstand darstellen, definiert werden
- Mittels Workflow können Change- und Release-Prozesse elektronisch abgebildet werden
- Teams können einfacher zusammen Code entwickeln

ITIL gibt als Best Practice Framework

- ITIL gibt als Best Practice Framework einen Prozess vor, welcher von Konfigurationsmanagement flankiert wird
- Das **Service Level Management** gibt die Anforderungen, diese werden umgesetzt und durch das **Change Management** freigegeben, so dass diese vom **Release Management** eingeplant werden
- Das Change Management unter ITIL hat mit dem Change Management aus der Projektwelt nur am Rande zu tun
- Changes in ITIL beziehen sich auf den produktiven Betrieb. Changes in Projekten beziehen sich auf Scope, Budgets oder Zeit für die Entwicklung.



Quelle: AXA-Tech, Werner Staub



OST
Ostscheizer
Fachhochschule

Vorlesung PMQM

09 - Agiles Projektmanagement I

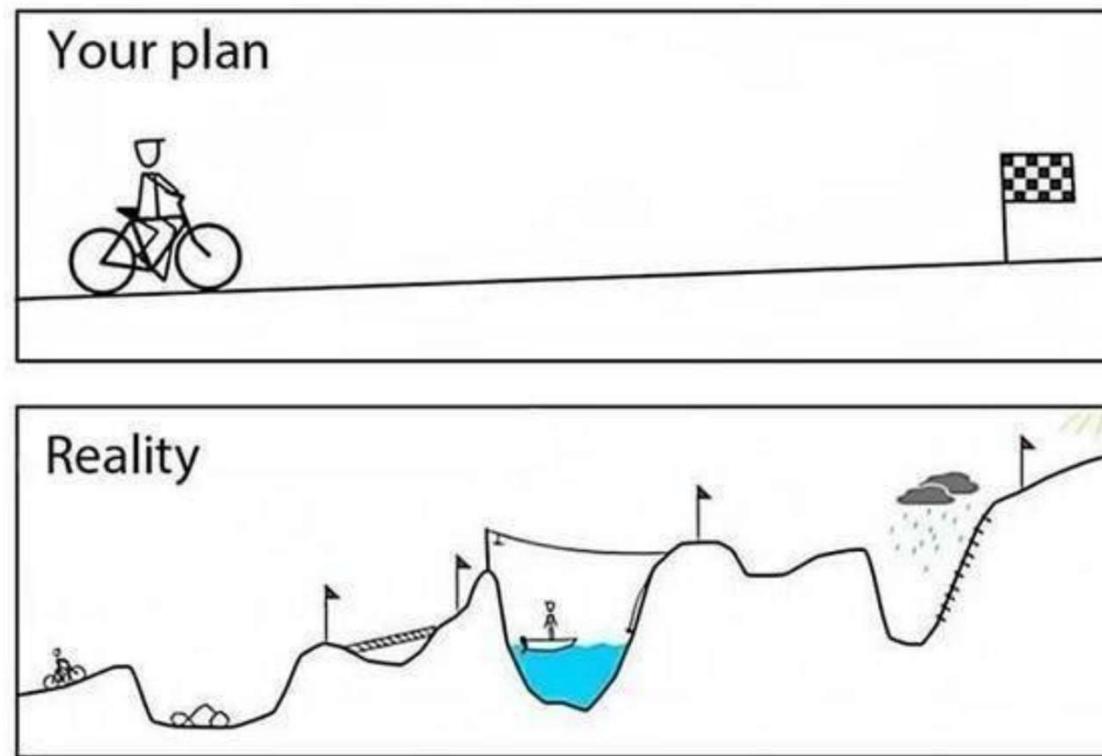
Werner Staub

Informatik

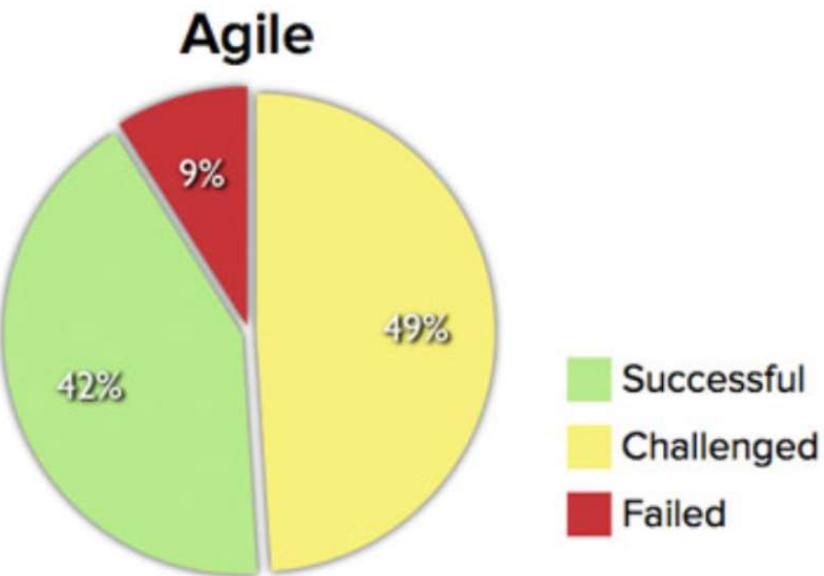
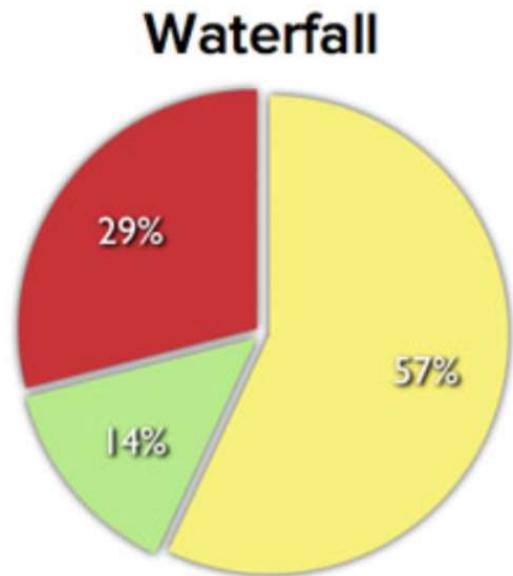
Ziele der heutigen Vorlesung

- Lerninhalte
 - Motivation für Agilität
 - Agiles Manifest
 - Verwendete Methoden
- Ziele: Die Studierenden...
 - lernen das Agile Manifest kennen
 - verstehen die Motivation hinter einem agilen Vorgehen
 - Verschaffen sich einen Überblick über agile Methoden

Planung und Realität



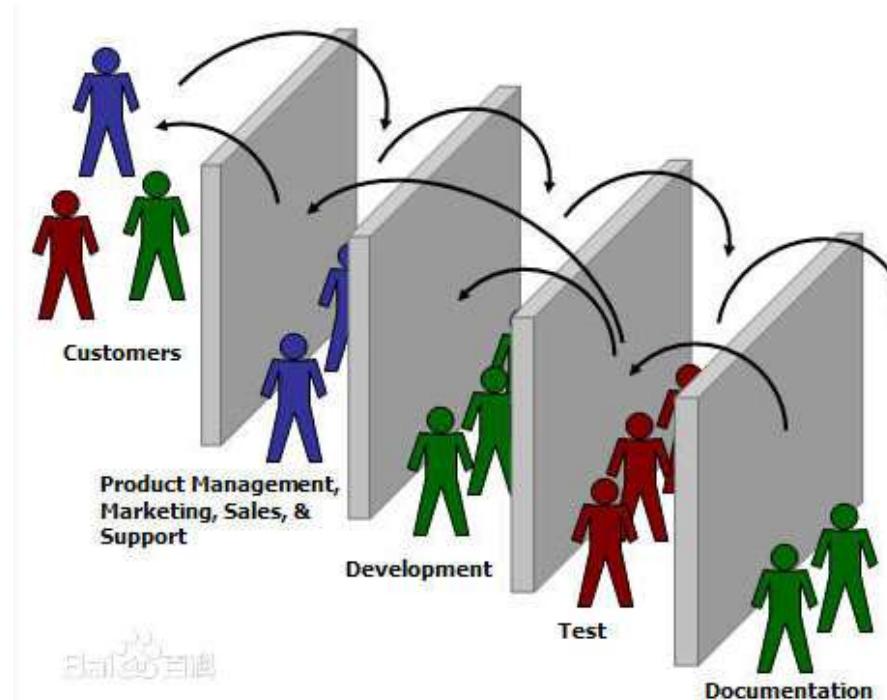
Projekt Erfolge



Successful
Challenged
Failed

Quelle: *The CHAOS Manifesto*, The Standish Group, 2012

Vom Wasserfall zum «Mauerfall»



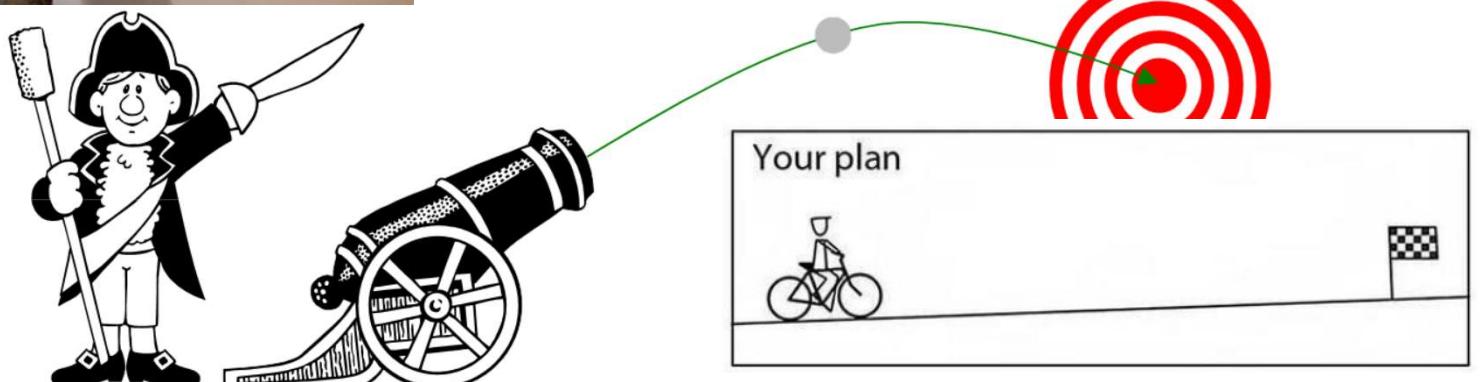
Quelle: blog.inf.ed.ac.uk

Probleme des Wasserfalls



Voraussetzungen:

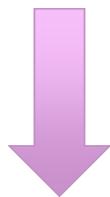
- Der Kunde weiss genau, was er will
- Das Projekt Team weiss, wie man es umsetzen muss
- Nichts ändert sich während der Projektdauer



Warum scheitern Projekte?

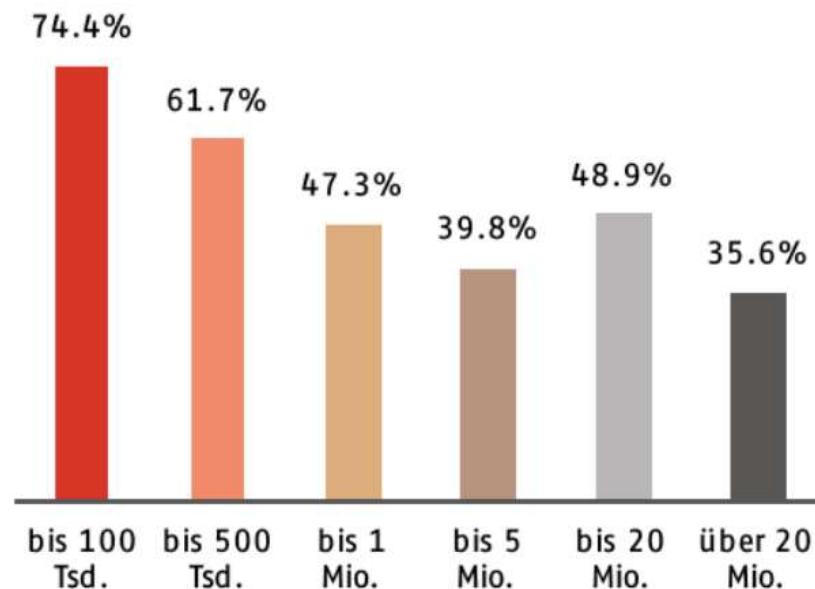
Je grösser das Projekt, desto höher die Wahrscheinlichkeit zu scheitern

- Viele Abhängigkeiten
- Ursache – Wirkung nicht erkannt
- Fachwissen zu tief
- Anforderungen zu unklar
-



**Iteration statt
unpräzise Analyse**

Projekterfolg nach Grösse
Unabhängig vom Vorgehen



Quelle: Trend Report swiss Q

Werte die wir schätzen



12 Prinzipien

Since 2001

1) Den Kunden zufrieden stellen

Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen

- Vorteile
 - Kunde profitiert schneller vom Projekt
 - Die Umsetzung findet in kleinen Schritten statt und reduziert damit auch das Risiko
 - Missverständnisse in den Anforderungen werden schneller erkannt und können früher korrigiert werden
- Nachteile
 - Betrieb wird regelmässig durch neue Releases unterbrochen
 - Projektumfang ist nicht klar abgegrenzt
 - Frühes Anwenden des Produkts weckt Wünsche für neue Anforderungen (steigende Kosten, ...)

2) Änderungen willkommen heissen

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

- Änderungen willkommen heissen
 - Anforderungen können jederzeit ändern, da auch das Umfeld des Auftraggebers nicht stabil sondern immer im Fluss ist.
- Wettbewerbsvorteil des Kunden
 - Das oberste Ziel ist die Zufriedenheit des Kunden. Er bezahlt uns dafür, dass wir ihm zu seinem Erfolg verhelfen.
 - Mit agilen Prozessen können wir ihm dazu einen Wettbewerbsvorteil liefern.
Aber Achtung: Die Konkurrenz schläft auch nicht

3) Regelmässige Lieferungen

Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.

- funktionierende Software
 - In jedem Sprint soll ein vollständiges SW-Paket (getestet und dokumentiert) erstellt und geliefert werden. Das heisst, das Produkt ist jederzeit betriebsbereit und verbessert ständig den Kundennutzen.
- regelmässig
 - Die Sprints sollen regelmässig und so kurz wie möglich, aber so lang wie nötig sein, um weiteren Kundennutzen zu erzeugen.
Daraus entsteht ein kontinuierlicher Fluss (im Vergleich zum Wasserfall)

4) Bereichsübergreifende Zusammenarbeit

Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.

- Fachexperten und Entwickler
 - Im Team sind nicht nur die Entwickler, sondern auch die Fachexperten (Security, Architekten, Service Management, System Engineers, ...)
- täglich
 - Der Austausch basiert auf einen DSTAM (Daily Standup Meeting). Dieses Meeting ist moderiert und dauert nur wenige Minuten. Dabei werden alle wichtigen Informationen zu den Tätigkeiten von gestern und heute ausgetauscht.

5) Unterstützung und Vertrauen

Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.

- Motivierte Individuen
 - In einen Projekt (klassisch oder agil) sollen nur motivierte Personen mitarbeiten. Alle anderen sind schädlich.
- Umfeld, Unterstützung und Vertrauen
 - Es ist die Aufgabe der Projektleitung, dem Team das optimale Umfeld zur Verfügung zu stellen und es zu unterstützen.
 - Er kann dafür dem Team vertrauen, dass sie ihre Aufgaben im Sinne des Projekts selbstständig ausführen.

6) Konversation von Angesicht zu Angesicht

Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.

- Wichtige Informationen werden im Gespräch «mündlich» ausgetauscht und nicht in Dokumenten irgendwo festgehalten.
- Ich denke, mit den heutigen Mitteln (Zoom, Teams, ...) ist das auch in einer Videokonferenz möglich.
- Jedoch ist E-Mail oder Chat ist nicht genügend.

7) Funktionierende Software

Funktionierende Software ist das wichtigste Fortschrittsmaß.

- Es ist nicht relevant, wie viele Module, Klassen und Methoden das Team erstellt hat. Nur voll funktionsfähige und getestete Teile des Produkts zählen.
- Da in jedem Sprint ein Stück funktionierende Software an den Kunden geliefert wird, können diese Sprint ideal als Maß für den Fortschritt verwendet werden.
- Oft werden die Sprints mit Story-Points gewichtet. Diese eignen sich sehr gut für die Fortschrittsbestimmung.

8) Gleichmässiges Tempo

Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmässiges Tempo auf unbegrenzte Zeit halten können.

- Nachhaltige Entwicklung
 - Die Arbeit in diesen Prozessen verändert die Arbeitskultur innerhalb einer Firma.
 - Ein Kulturwandel braucht Zeit und ist keine «Hau Ruck» Übung
- gleichmässiges Tempo auf unbegrenzte Zeit
 - Auch ein agiles Projekt ist nicht unbegrenzt. Aber hier geht es darum, dass das Team nicht mit Vollgas in ein Burnout läuft.
 - Im Sport würde man sagen, das Team muss im «anaeroben» Bereich bleiben.
 - Für Auftraggeber und Benutzer bedeutet das aber, dass sich über eine lange Zeit das Produkt entwickelt (und immer wieder ändert).

9) Technische Excellenz

Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.

- Ständiges Augenmerk
 - Routine und Ermüdung im Alltag ist Gift und führt zu Nachlässigkeit.
 - Wenn nötig zwingend Verschnaufpausen einlegen.
- technische Excellence
 - Die richtigen Leute mit den richtigen Skills (Fachexperten und Entwickler)
 - Immer wieder bewährtes in Frage stellen und noch besser werden
Das Produkt darf nicht nur aussen schon aussehen, sondern auch innen sauber umgesetzt sein.
 - Lebenslanges Lernen etablieren und sich im Team regelmässig herausfordern und unterstützen

10) Einfachheit

Einfachheit - die Kunst, die Menge nicht getaner Arbeit zu maximieren - ist essenziell.

- Alle Aufgaben danach analysieren, ob sie etwas zum Erfolg des Projekts beitragen.
- Unnötige Arbeiten in der ToDo Liste weglassen
- Was muss nicht umgesetzt werden, weil es niemand braucht?
- Auch Methoden, Klassen und Software Design im allgemeinen in Frage stellen und falls unnötig, weglassen
- ➔ Unser Auftraggeber bezahlt uns nicht für unnötigen technischen Schnick-Schnack

11) Selbstorganisation

Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.

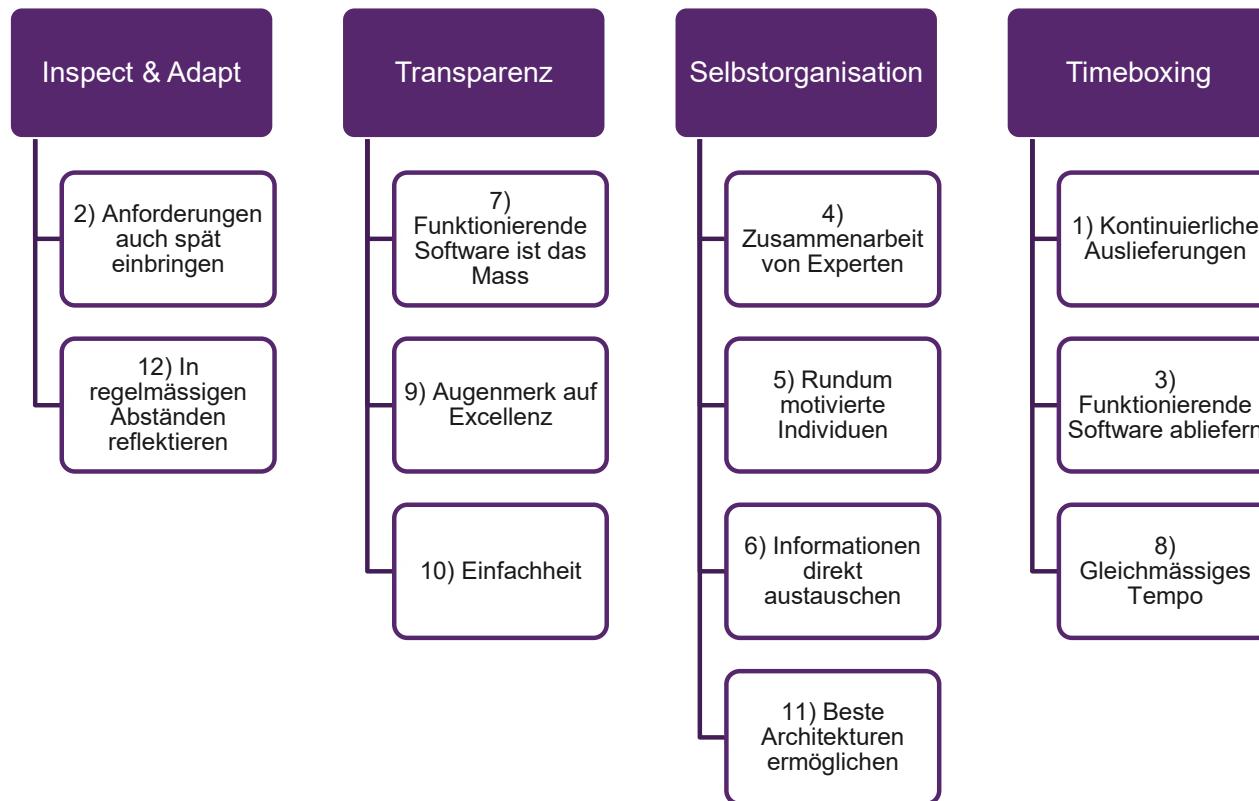
- Beste Architekturen, Anforderungen und Entwürfe
 - Es sollen Lösungen zu Produktideen, technische und inhaltliche Herausforderungen im Team gefunden werden.
 - Diese dürfen nicht einzeln entwickelt werden. Sie gehören zusammen zu einem gemeinsamen Ganzen.
- entstehen
 - Die Resultate können nicht geplant werden.
 - Sie werden im Laufe des Projekt entdeckt und evolutionär weiter entwickelt.
 - Sie sollen dabei immer wieder hinterfragt und verbessert werden.
- Selbstorganisierte Teams
 - Die Teams brauchen keinen Chef, um zu entscheiden, was sie wie tun müssen
 - Wenn sie die nötigen Rollen und Fähigkeiten haben, dann können sie sich selbst organisieren.

12) Inspektion und Adaption

In regelmässigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

- In regelmässigen Abständen
 - Erkannte Fehler müssen im laufenden Projekt ausgewertet und korrigiert werden können.
 - Daher ist ein «Lessons Learn» nicht nur am Ende des Projekt, sondern laufend nötig.
- reflektiert das Team, wie es effektiver werden kann
 - Das Team selbst kann die eigenen Gewohnheiten am Besten beurteilen und entscheiden, was geändert werden muss.
- und passt sein Verhalten entsprechend an
 - Resultate einer Retrospektive / Reflexion müssen umgesetzt werden (können)
 - Dazu braucht das Team die entsprechende Einsicht

Welche Bestandteile ergeben sich aus den 12 Prinzipien

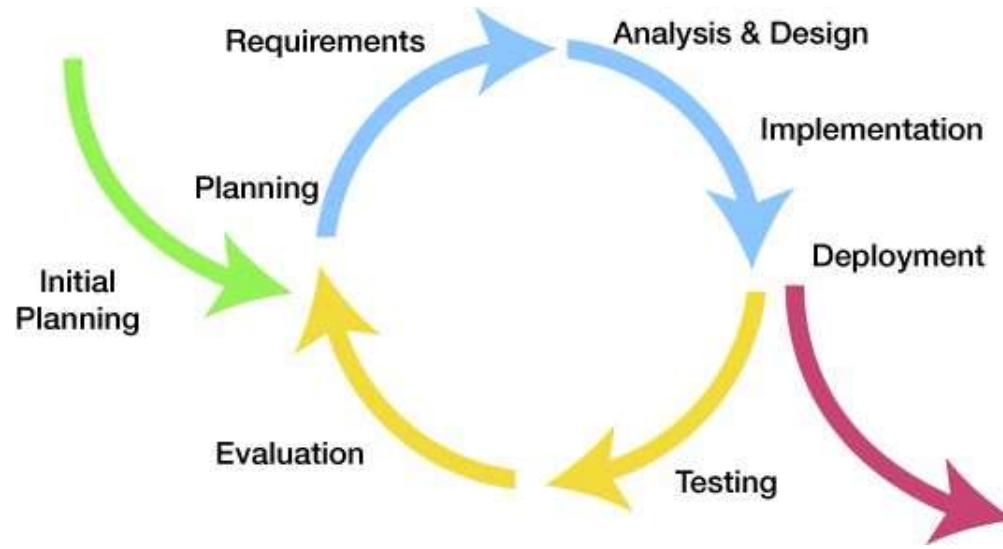


Wann ist ein Projekt agil?

- Zu allererst ist Agilität eine Denkweise und eine Einstellung der Personen im Projekt. Nur wenn das **Projektteam** und die **Stakeholder** ihre geistige Flexibilität nutzen, kann tatsächlich von agil gesprochen werden.
- Wann ist aber eine Projektmethodik agil?
Grundsätzlich sobald die vier Bestandteile im Projekt erkennbar sind:
 - Inspect & Adapt
 - Transparenz
 - Selbstorganisation
 - Timeboxing
- Erfahrene Projektleiter vermögen auf Basis der Projektcharakteristik situativ eine Methode zu entwickeln, welche von Beginn bis Ende verwendet werden kann. Dazu kennen Sie die Vorteile und Nachteile aus der agilen und klassischen Welt.

Inspect & Adapt

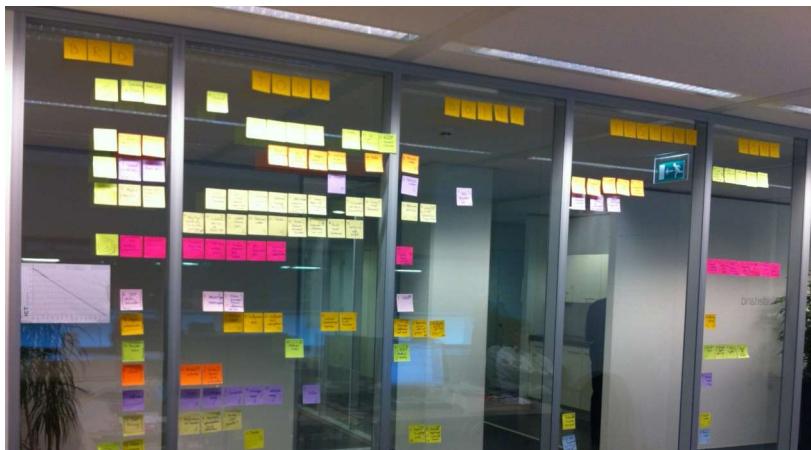
- Sämtliche Bestandteile im Projekt werden regelmässig überprüft und wenn notwendig angepasst, d.h. auch wichtige Anforderungen dürfen jederzeit einfließen



Quelle: curtisbacon

Transparenz

- Informationen stehen jedem zur Verfügung und werden «einfach» gehalten
- Qualität und funktionierende Software gelten als Fortschrittsindikator
- Keine versteckte Agenda oder alternative Fortschrittskontrollen
- Visualisierung der Daten

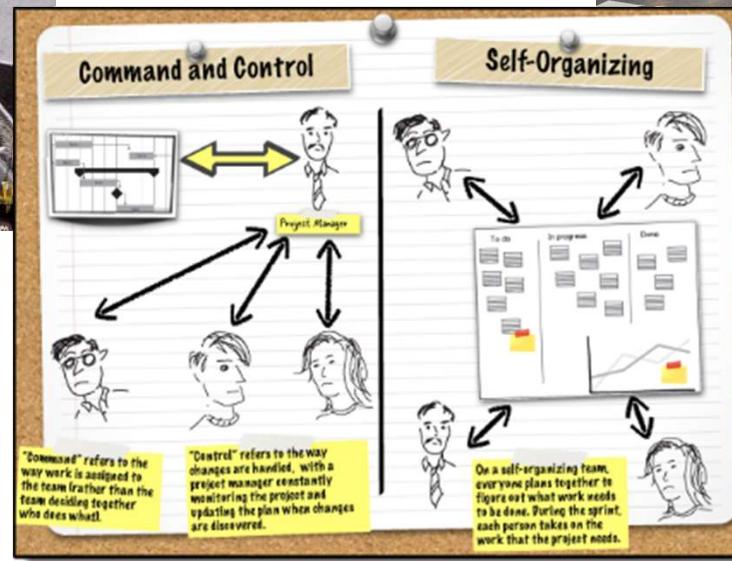


Quelle: pinterest.com



Quelle: youtube.com

Selbstorganisation



Timeboxing vs. Functionboxing

Zeit vor Inhalt



- Fixe Zeitdauer, nicht länger und nicht kürzer
- Nicht erledigte Arbeiten werden in die nächste Box eingeplant
- Mitarbeitende gegen Einfluss von aussen abgeschottet

Inhalt vor Zeit



- Fixer Scope, darf nicht gekürzt oder erweitert werden
- Arbeiten werden immer erledigt
- Mitarbeitende gegen Einfluss von aussen abgeschottet

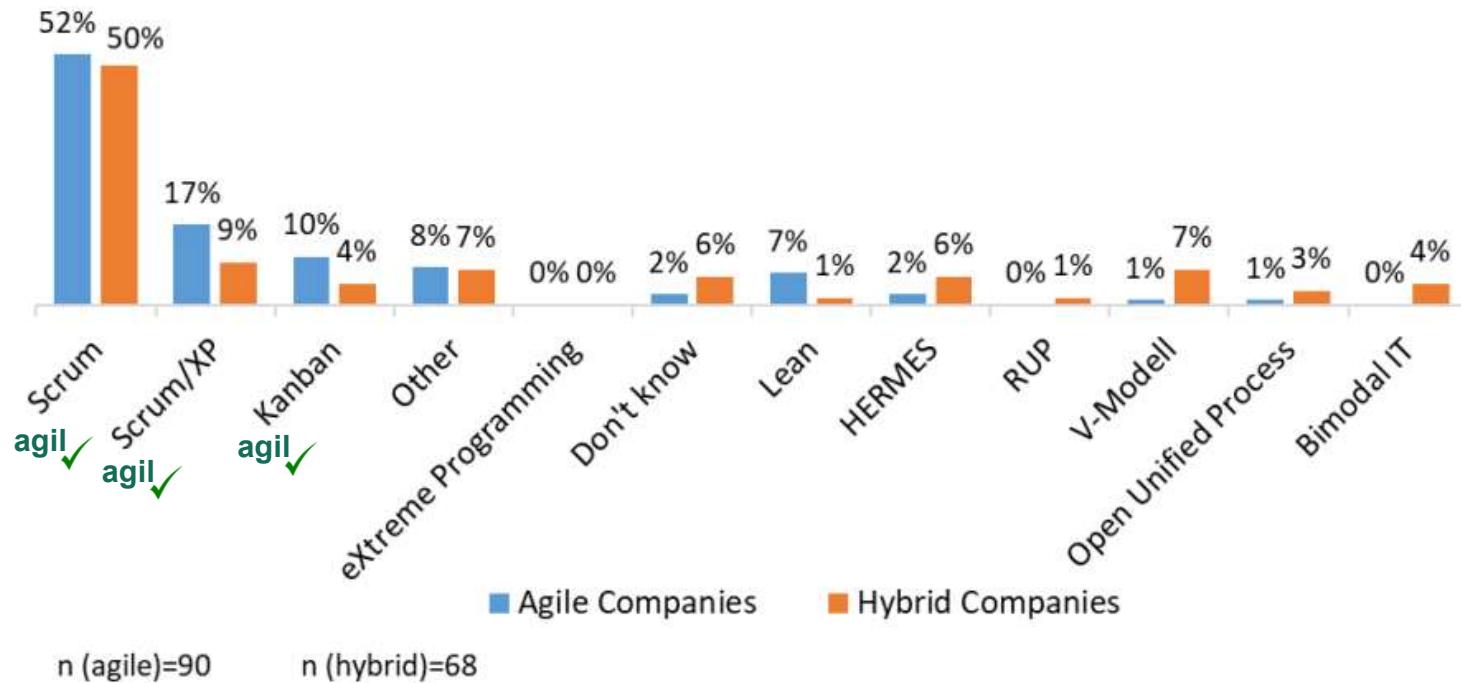
User Stories

- User Storys sind Nutzergeschichten und Grundlage für ein Scrum-Projekt.
- Der Product Owner beschreibt in einer User Story die Anforderungen aus Nutzersicht: Wie wünschen sich Anwender eine Software oder Kunden einen Liefergegenstand, um ihre Geschäftsziele zu erreichen.
- User Story besteht aus wenigen, leicht verständlichen Sätzen. Sie sind kurz, aus Kundensicht jedoch spezifisch und detailliert.
- INVEST-Kriterien (für gute User Stories):
 - Independent (unabhängig von anderen User Stories)
 - Negotiable (verhandelbar ; diskutierbar)
 - Valuable (nützlich ; muss Wert stiften)
 - Estimable (Aufwände müssen abschätzbar sein)
 - Small (klein, damit in einem Sprint umsetzbar)
 - Testable (testbar ; Definition of Done)

User Stories

- Eine User Story wird im agilen Projektmanagement anstelle von Anforderungen erhoben und in einem Backlog abgelegt
- „Als [Rolle] möchte ich [Funktion] um [Nutzen].“
- Schema: “Wer” möchte “was” [und “warum”]
- Beispiele:
 - *Ich als Projektleiter möchte die Buchungen meiner Mitarbeitenden freigeben können, damit diese Gültigkeit erlangen*
 - *Als Student möchte ich mich im E-Learning Portal registrieren, um mich auf die IPMA-Zertifizierung vorzubereiten.*

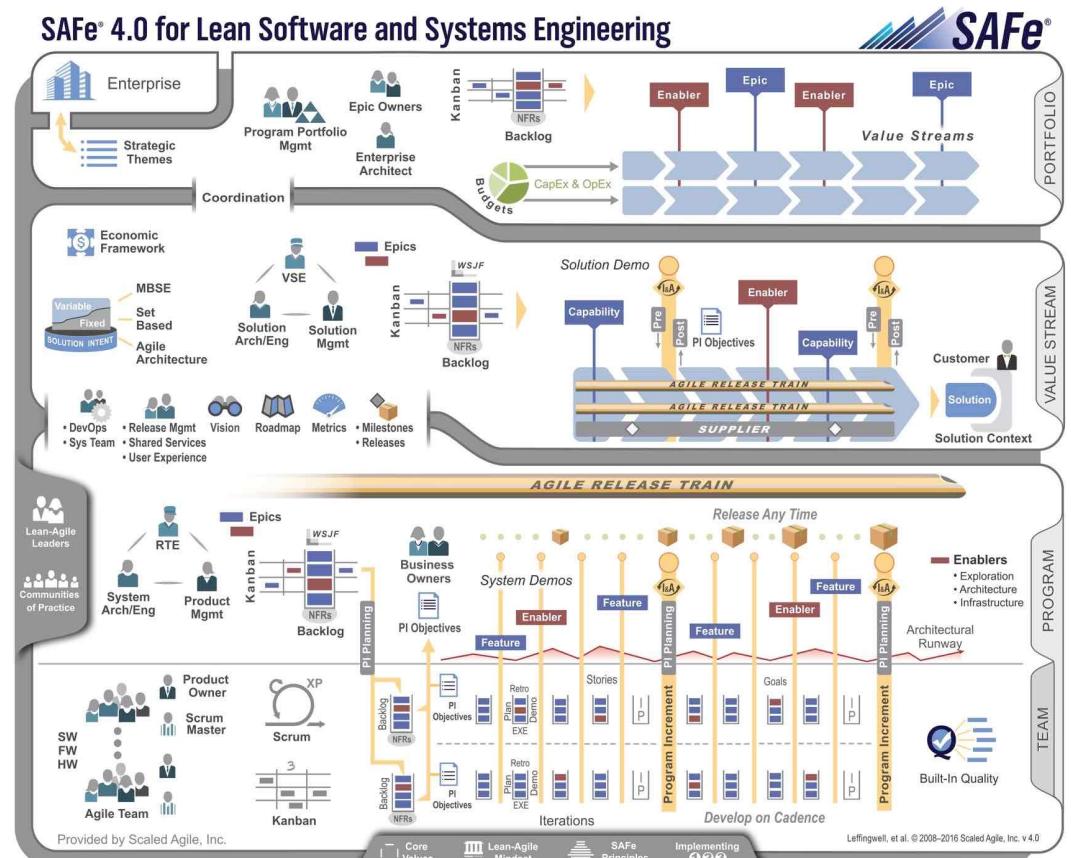
Welche agile Methoden werden in der Schweiz verwendet



Quelle: swissq.it (2016)

Scaled Agile Framework

Das Scaled Agile Framework (SAFe) ist das verbreitetste Rahmenwerk, um Scrum bzw. Agil zu skalieren – sprich, im „Großen“ zu leben. SAFe bietet agile Lösungen für alle Ebenen einer Organisation: von Teams über Bereiche bis hin zur Unternehmenssteuerung.



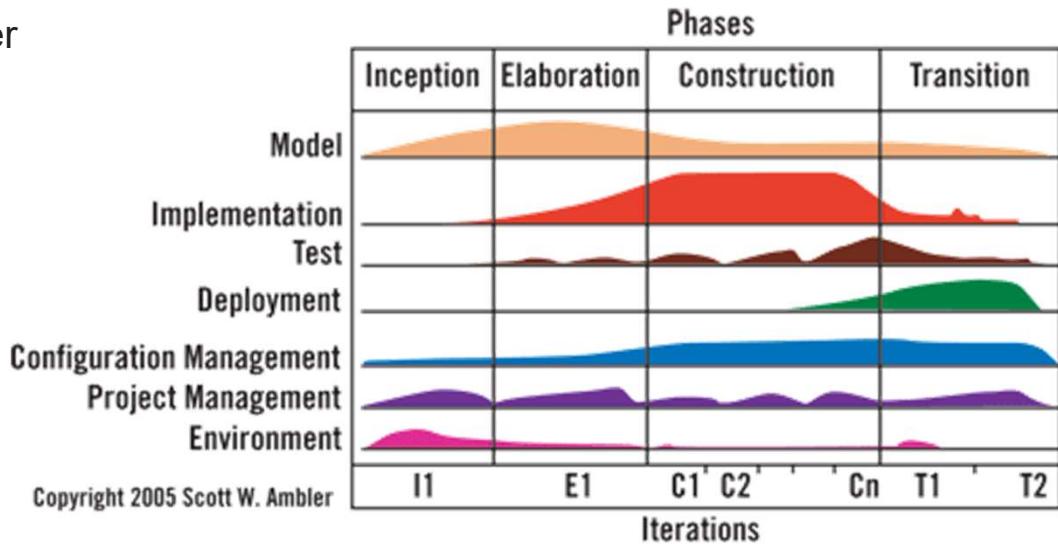
Agile Unified Process

Der **Agile Unified Process (AUP)** ist ein hybrider Modellierungsansatz, der den Rational Unified Process (RUP) mit agiler Softwareentwicklung verbindet.

Der AUP bietet einen iterativ-inkrementellen Zugang zur Softwareentwicklung.

Philosophie

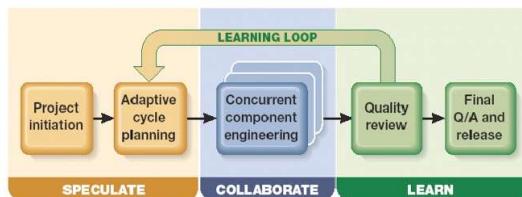
- *Ihre Leute wissen, was sie machen*
- *Alles ist auf wenigen Seiten prägnant beschrieben*
- *Auf die hochwertigen Aktivitäten konzentrieren*
- *Tool-Unabhängigkeit*
- *Sie werden dieses Produkt womöglich anpassen müssen*



Quelle: drdubbs.com

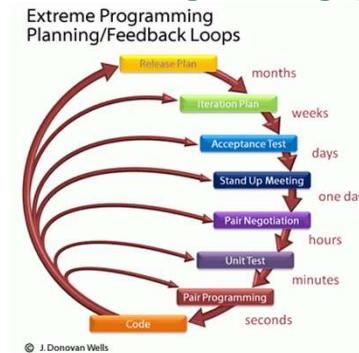
Weitere Agile Methoden #3

Adaptive Software Development (ASD)



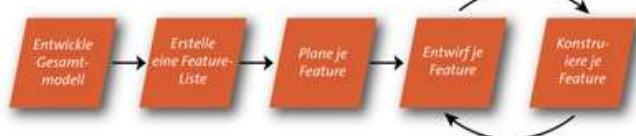
Kontinuierliche Anpassung an immer neue Anforderungen

Extreme Programming (XP)



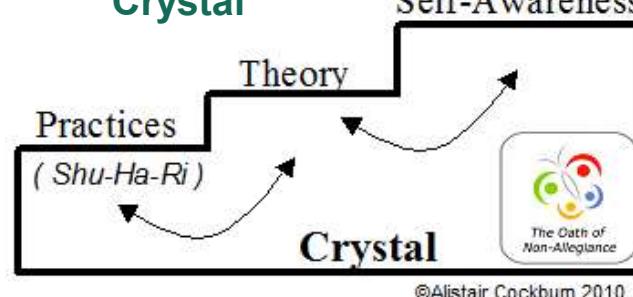
Methode, die das Lösen einer Programmieraufgabe in den Vordergrund stellt

Feature Driven Development (FDD)



Sammlung von Arbeitstechniken, Strukturen, Rollen und Methoden

Crystal



Familie mit Methoden benannt nach Farben:

- Crystal Clear
- Crystal Yellow
- Crystal Orange
- Crystal Red
- usw.



Warum werden agile Methoden verwendet

- **Schneller Projektstart**
da zu dessen Beginn nicht alle Details bereits festgelegt und entschieden sein müssen.
- **Direkter Einfluss auf den Projektverlauf**
durch die Teilnahme an Meetings, Usability-Tests, Sprint-Planungen und persönliche Abnahme von Sprint-Ergebnissen.
- **Effektivere Arbeitsabläufe**
- **Hohe Flexibilität**
gegenüber geänderten Wettbewerbsanforderungen, Kunden- und Nutzerbedürfnissen.
- **Fehler werden früh erkannt** und behoben
- **Effektive Ergebnisse**
die auf Kunden- und Nutzerbedürfnisse zugeschnitten sind.
- **Schnelle Ergebnisse**
und geringe Zeit bis zur Markteinführung (Time-to-Market)

Quelle: interactive-tools.de



Vorlesung PMQM

10 - Agiles Projektmanagement II

Werner Staub

Informatik

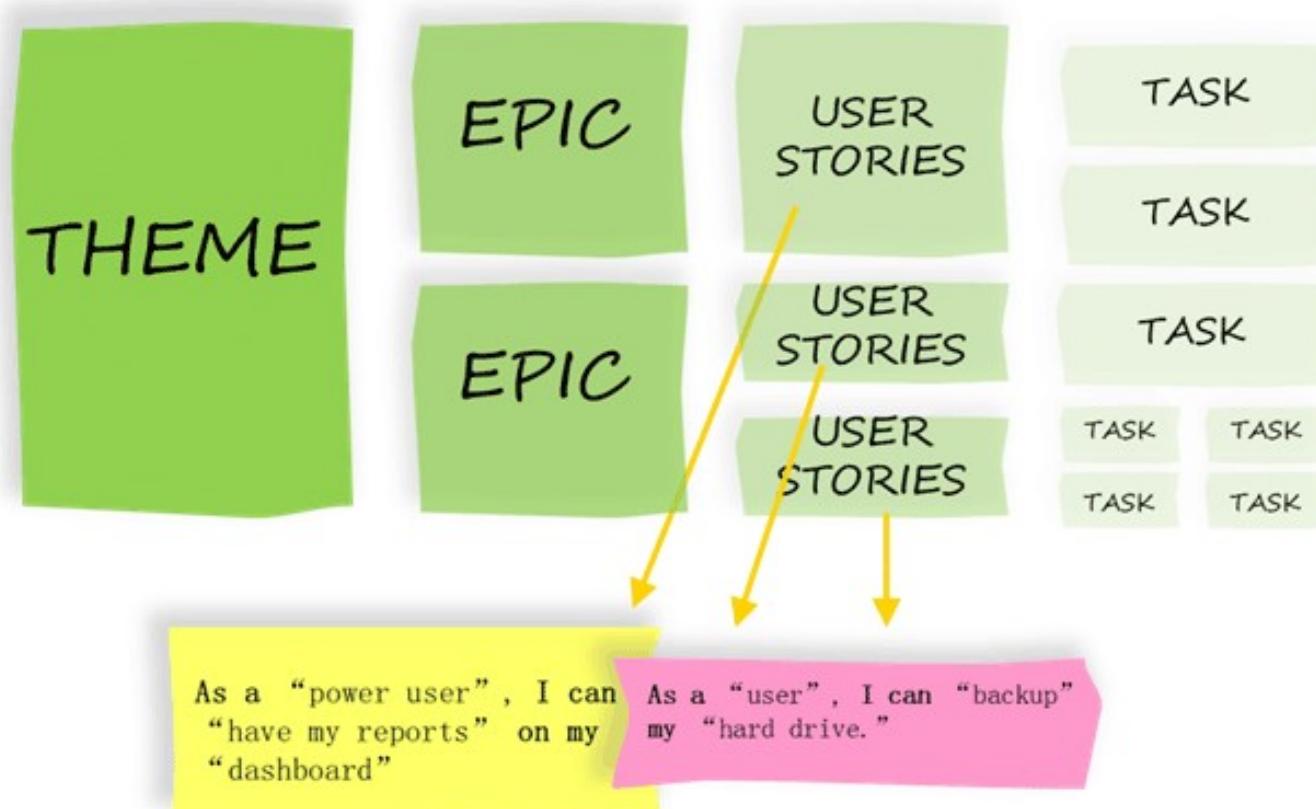
Ziele der heutigen Vorlesung

- Lerninhalte
 - Scrum
 - Kanban
 - Weitere agile Methoden
- Ziele: Die Studierenden...
 - lernen Scrum noch besser kennen oder repetieren die Methode
 - verstehen das Vorgehen von Kanban
 - wissen um weitere Ausbaumöglichkeiten der agilen Methoden

SCRUM

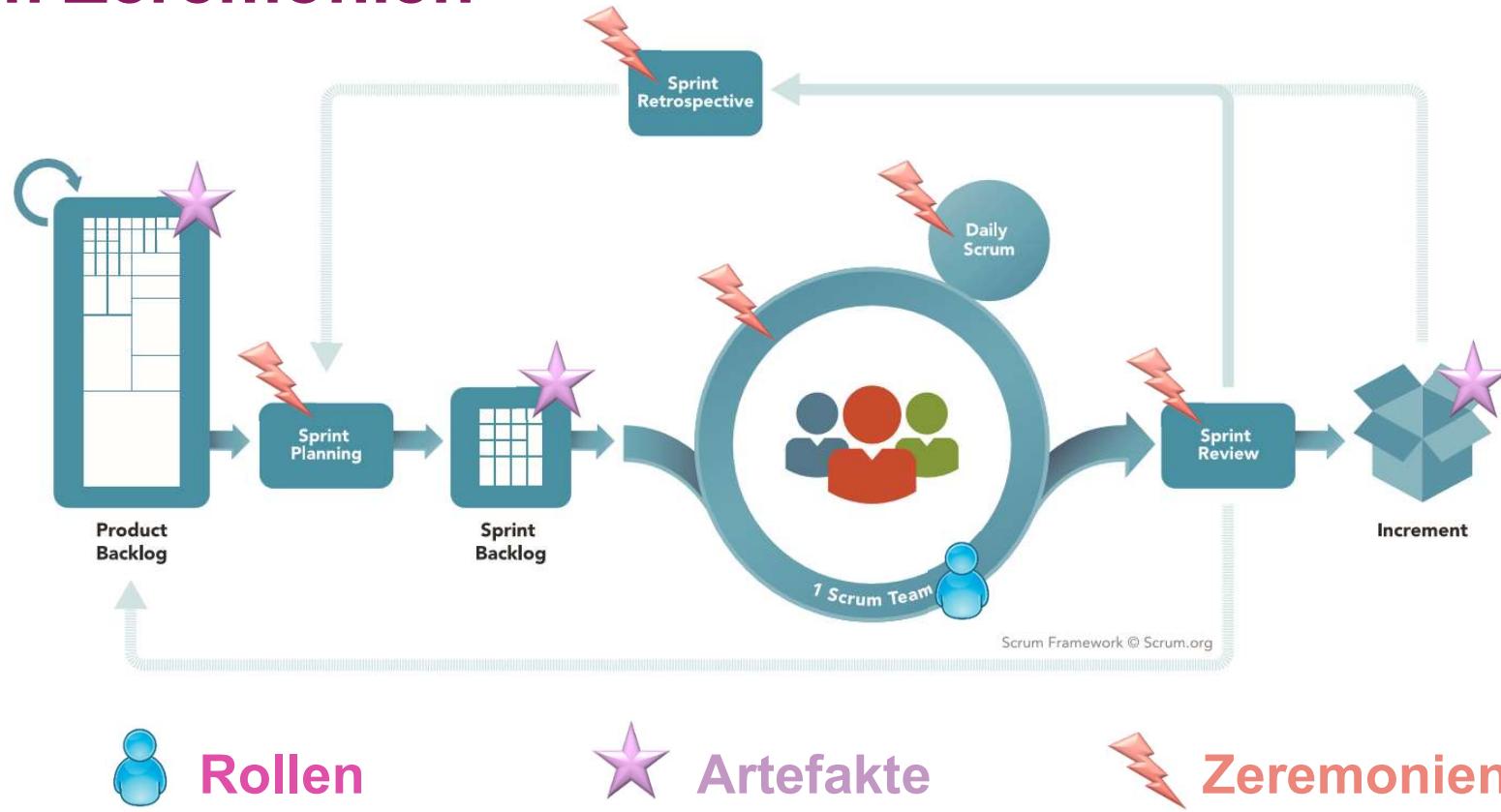
- **Scrum** ist eine der vielen agilen Methoden des Projektmanagements
- Es soll die Teamproduktivität und -geschwindigkeit steigern und den Produktwert optimieren, indem es sich auf regelmäßiges Feedback von Endbenutzern stützt
- Der Name dieser Methode, „Scrum“, ist eigentlich der Rugby-Welt entlehnt:
Genau wie Rugby-Spieler versammeln sich agile Scrum-Teammitglieder so oft wie möglich
(in ... Scrums), um das Entwicklungsprojekt zu überprüfen, immer bereit für Veränderungen des ursprünglichen Plans
- Scrum ist ein ziemlich dynamischer und immer in Bewegung befindlicher Ansatz, der ein kollaboratives Projektmanagement fördert und dem Kunden die richtige Mischung zwischen der geplanten Investition und dem gelieferten Endprodukt garantiert
- Er fördert:
 - mehr Zusammenarbeit mit dem Kunden
 - Überwindung der Angst vor Veränderungen
 - Interaktion mit Menschen in den Mittelpunkt jedes Softwareprojektmanagements stellen
 - Fokussierung bei der Lieferung von Betriebssoftware

Theme, Epic und USER STORIES



www.agile-scrum.be

Scrum Zeremonien



Die Elemente

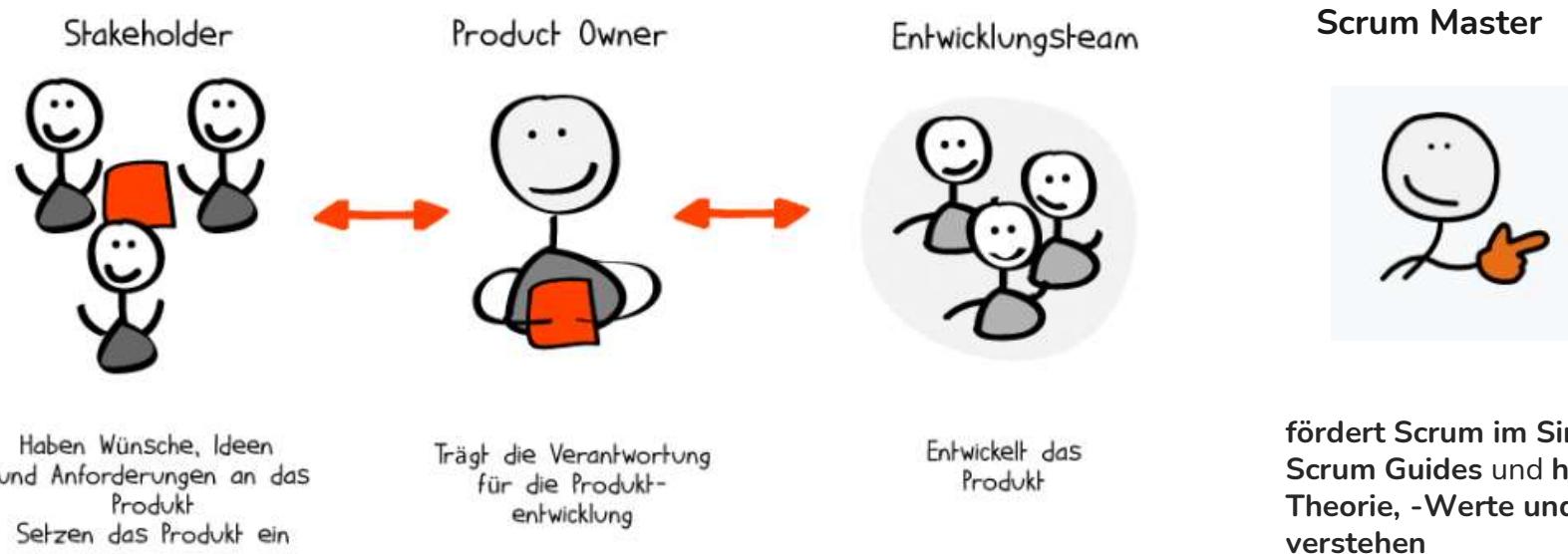
Rollen
Product Owner
Development Team
Scrum Master

Artefakte
Product Backlog
Sprint Backlog
Increment

Zeremonie	Dauer
Sprint	2 bis 4 Wochen
Sprint Planning	< 4 bis 8 Stunden*
Sprint Review	2 bis 4 Stunden*
Sprint Retrospektive	1.5 bis 3 Stunden*
Daily Scrum	<15 Minuten*

* Streng ausgelegt keine Timebox

Scrum - Rollen



Quelle: <https://projekte-leicht-gemacht.de/projektmanagement/agiles-projektmanagement/scrum>

Fundament von Scrum



Transparenz

- Wesentliche Aspekte sind sichtbar
- Gemeinsamer Standard



Überprüfung

- Fortschritt ständig überprüfen
- Zu gegebenen Zeiten überprüfen



Anpassung

- Anpassung so schnell wie möglich
- Anpassung ist stetig möglich

Unterscheidung Produkt Backlog und Sprint Backlog

	Produkt Backlog	Sprint Backlog	
Der Product Owner priorisiert	✓		
Das Development Team priorisiert/nimmt entgegen		✓	
Enthält nur eine bestimmte Anzahl Stories		✓	
Kann so viele Stories wie möglich beinhalten	✓		
Besteht ausschliesslich aus Testfällen			✓
Das Development Team passt es während Sprint an		✓	
Der Scrum Master priorisiert			✓
Einträge daraus werden täglich besprochen		✓	
Einträge daraus werden alle 2-4 Wochen besprochen	✓		

«Verunreinigungen» von Scrum – was so gemacht wird..

Falsch / (Richtig)



Zero-Sprint
(Existiert nicht)



Sprints verlängern/
verkürzen
(immer gleich lang)



Irgendwer bricht
Sprints ab
(Product Owner)



Burn Down der
Teams vergleichen
(No Benchmarks)



Teamzusammen-
setzung bestimmen
(Selbstorganisation)



Daily Standup wird
durchgeführt
(Daily Scrum)



Der Product Owner
arbeitet mit
(Arbeitet nie mit)



Arbeit wird verteilt
(Pull)

Was ist Kanban (in der IT)?

«Kanban ist ein Vorgehensmodell zur Softwareentwicklung, bei dem die Anzahl paralleler Arbeiten, der Work in Progress (WiP) reduziert und somit schnellere Durchlaufzeiten erreicht und Probleme – insbesondere Engpässe – schnell sichtbar gemacht werden sollen.»

[wikipedia.org](https://de.wikipedia.org)

Simples Kanban-Board



Nutzen

- Arbeit visualisieren
- Ist-Zustand abbilden
- Workflow beobachten
- Beschränkungen einbauen (Fokus auf aktive Aufgaben durch WiP-Limits)
- Gesamtarbeit wird in gut überschaubare Teilaufgaben unterteilt
 - Zu grosse Aufgaben fliessen schlecht durch das System (> 3 Tage)
 - Gute Unterstützung bei der Aufwandsschätzung
- Prognosen ableiten (und reagieren)



Woher kommt Kanban?

kan·ban

看板

a Japanese manufacturing system in which the supply of components is regulated through the use of an instruction card sent along the production line.

Hinweis: Im Japanischen werden auch heute noch oft chinesische Zeichen für feste Ausdrücke verwendet

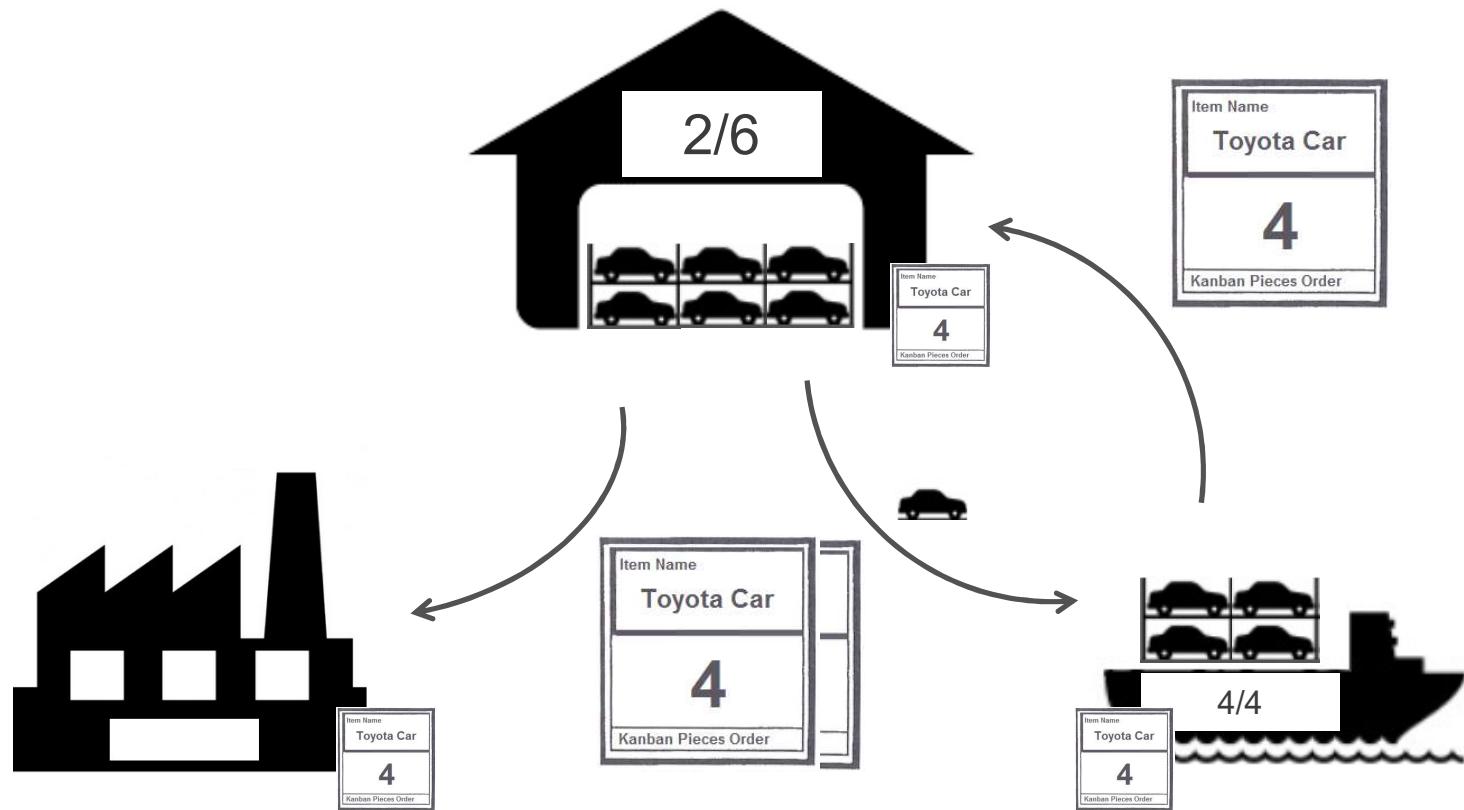
kan = (an-)sehen

ban = Tafel

Taiichi Ohnos Idee

- «Es müsste doch möglich sein, den Materialfluss in der Produktion nach dem Supermarkt-Prinzip zu organisieren, das heißt, ein Verbraucher entnimmt aus dem Regal eine Ware bestimmter Spezifikation und Menge; die Lücke wird bemerkt und wieder aufgefüllt.»



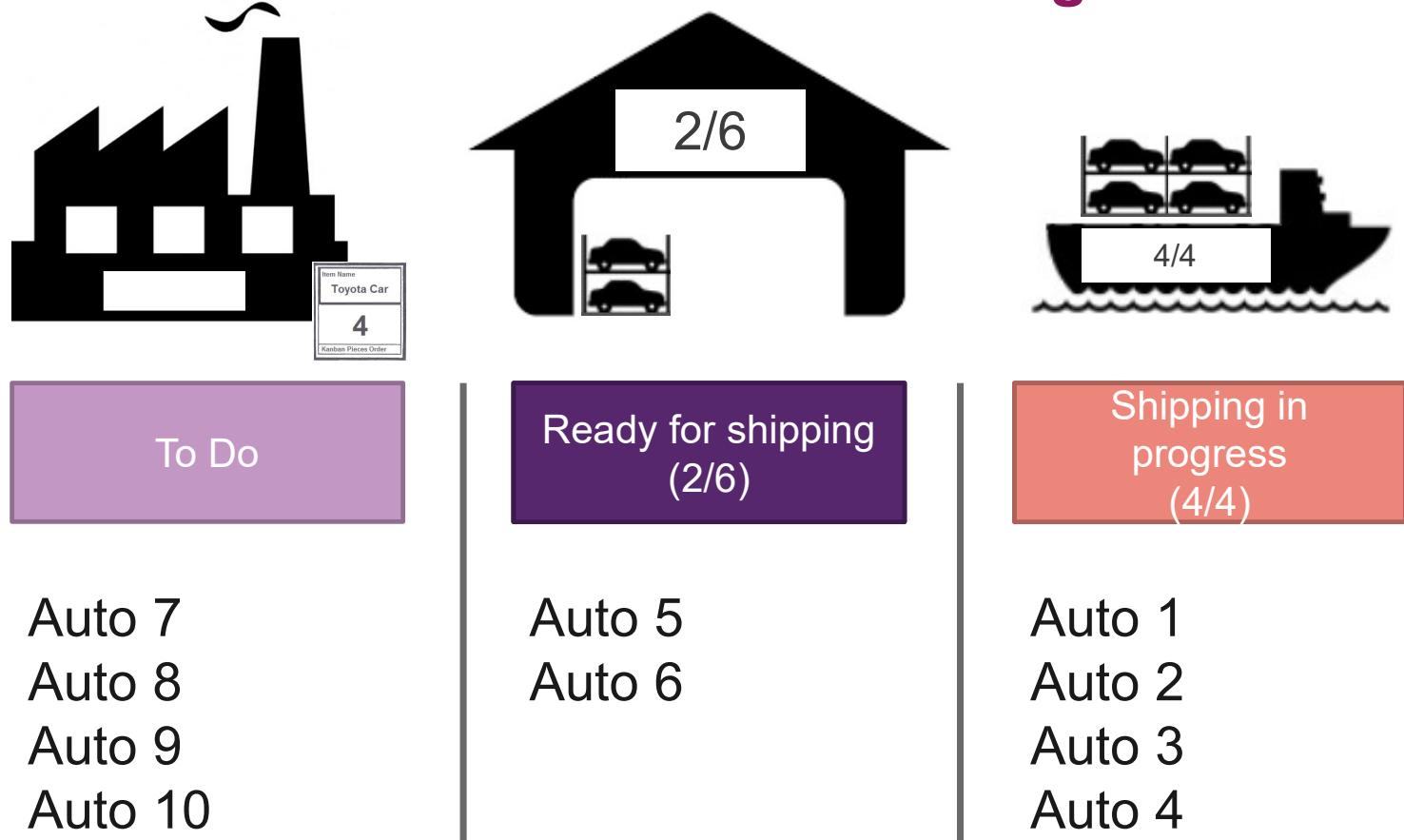


Folgen des Einsatzes von Kanban bei Toyota

- Innerhalb der Logistikkette erhält jede Arbeitseinheit nur die Materialmenge, die sie effektiv zum Produzieren benötigt.
- So wird sehr flexibel immer genau die richtige Menge in der richtigen Qualität produziert.
- Effizienzsteigerung in der Produktion
- Verschwendungen durch Lagerhaltung wird weitgehend vermieden
- Kürzere Reaktionszeiten auf sich ändernde Rahmenbedingungen
 - Marktsegmente, Gesetze
- Senkung der Kosten -> Steigerung des Gewinnes
Aber Risiko bei Problem mit den Lieferketten !



Von der Produktion zur Software Entwicklung – Transformation



Von der Produktion zur Software Entwicklung – Transformation

Backlog	Ready for development (2/6)	Work in progress (4/4)	Done
Item 10	Item 9	Item 7	Item 1
Item 11	Item 8	Item 6	Item 2
Item 12			Item 3
Item 13			Item 4
Item 14			Item 5
Item 15			
Item 16			
Item 17			
Item 18			
Item 19			

Jira Board SurflIT

DEV - IT Service Excellence
Kanban board

QUICK FILTERS: Only My Issues TO WS DB MM AR MW FK MC Others Recently Updated SurflIT Items ChargeIT Items

PLANNED 4 IN PROGRESS 6 IMPLEMENTED 12 TESTED 2 OF 8 CLOSED / REJECTED / CANCELLED 3 OF 29 Release...

ITSE-1029 Auto correct "Lemende" Cost Centers
ITSE-987 OpenShift SurflIT Prototype
ITSE-1085 Details automatisch und manuell laden
ITSE-1053 DB Struktur einlesen rekursiv und auslagern
ITSE-978 Filetransfer in SurflIT statistisch auswerten
ITSE-965 Oracle Shared DB Split aus Silva

ITSE-1063 ICP Flag in Staff nach AKK unterscheiden
ITSE-1063 ICP Flag in Staff nach AKK unterscheiden
ITSE-1074 Mockup von Product VisitCard
ITSE-1060 Ausbau Automatisierung Dashboard
ITSE-784 Cloud Cost Forecast

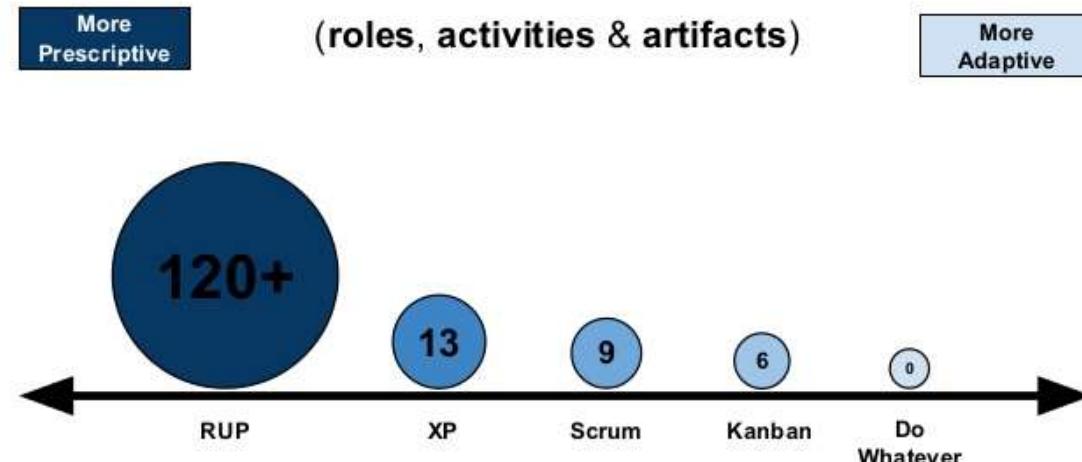
ITSE-994 Staff View with filter for "Distribution List"
ITSE-994 Staff View with filter for "Distribution List"

Vorneweg

- Kanban gilt offiziell zu Recht nicht als agile Methode
- Kanban enthält aber so gute Elemente, dass diese immer wieder in der Praxis sichtbar sind, auch im Zusammenhang mit Scrum

Übersicht: Von verordnend zum adaptiv

Prescriptive vs. Adaptive



Source: Kanban and Scrum, making the most of both. Henrik Kniberg & Mattias Skarin



www.torak.com

Kann ich Kanban agil machen?

- Wenn wir Kanban gegen Agilität prüfen, dann:

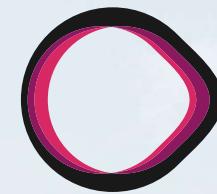
- Transparenz → Ja
- Inspect & Adapt → nur gering
- Selbstorganisation → Ja
- Timeboxing → Nein



- Die Lösung ist **Scrumban**, z.B. mit
 - Sprints einbauen (Timeboxing), pro Sprint ein Board
 - Sprint Review einbauen



➔ Wir ergänzen also Kanban um agile Elemente aus Scrum



OST
Ostschweizer
Fachhochschule

Vorlesung PMQM

11 – Kommunikation und Dokumentation

Werner Staub

Informatik

Ziele der heutigen Vorlesung

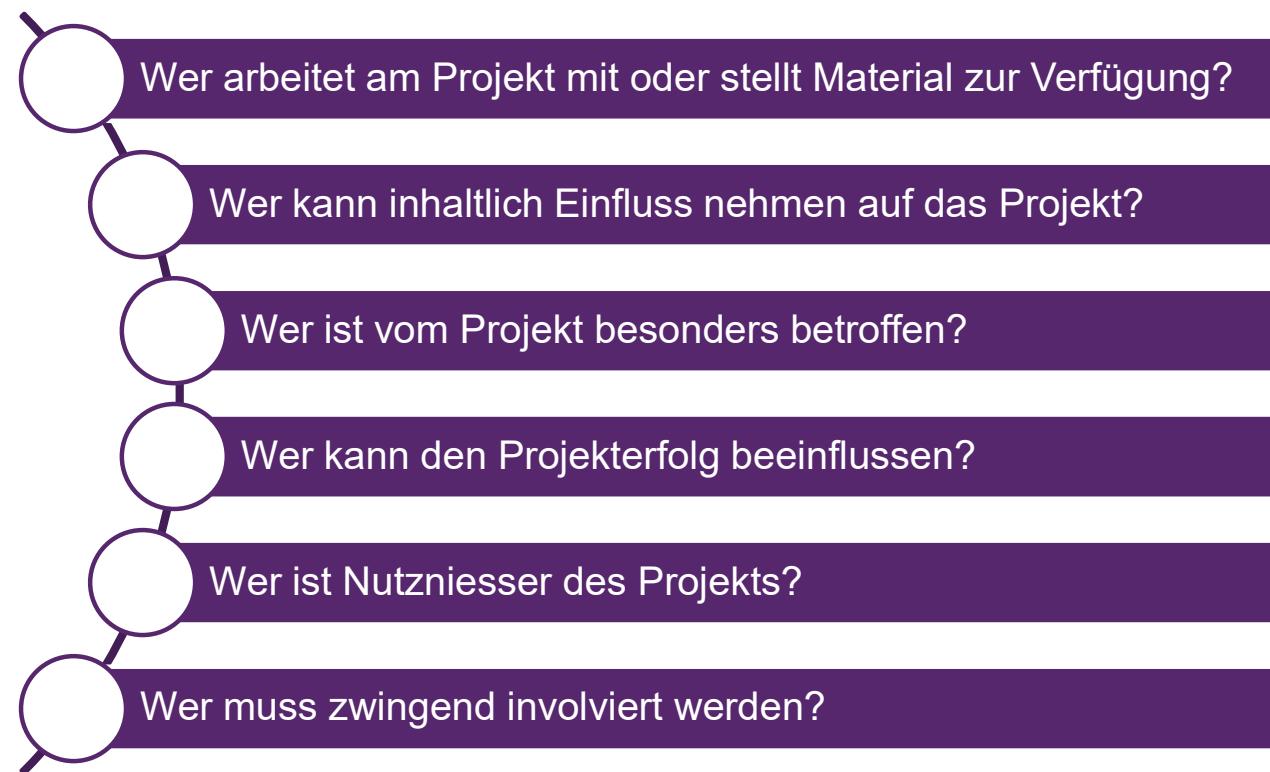
- Lerninhalte
 - Stakeholder
 - Führung / Ressourcenmanagement
 - Kommunikation
 - Projektmarketing
 - Abschlussdokumentation und laufende Dokumentation
- Ziele: Die Studierenden...
 - wissen um die Bedeutung der Stakeholder und lernen deren systematische Bearbeitung
 - kennen die Führungsansprüche in einem Projekt
 - erkennen die Wichtigkeit von Kommunikation und Projektmarketing
 - erweitern ihr Wissen hinsichtlich Projektdokumentation und kennen mehrere Beispiele

Stakeholder

- Als Stakeholder gelten **alle** am Projekt interessierten Parteien
- Die Stakeholder bilden verschiedene Anspruchsgruppen und beeinflussen das Projekt
- Projektleiter tun gut daran, die verschiedenen Stakeholder von Beginn weg zu identifizieren und die Kontakte zu pflegen
- Eine stetige Aktualisierung der Stakeholder Informationen ist unerlässlich
- Typische Stakeholder sind die Mitglieder des Lenkungsausschusses, die Kunden, die Lieferanten, die verschiedenen Endbenutzer, die Leistungserbringer aus IT oder Facilitymanagement und generell alle Personen die durch irgendeine Art vom Projekt betroffen sind
- Beim sogenannten Stakeholder Management gibt es unterschiedliche Möglichkeiten diese zu visualisieren



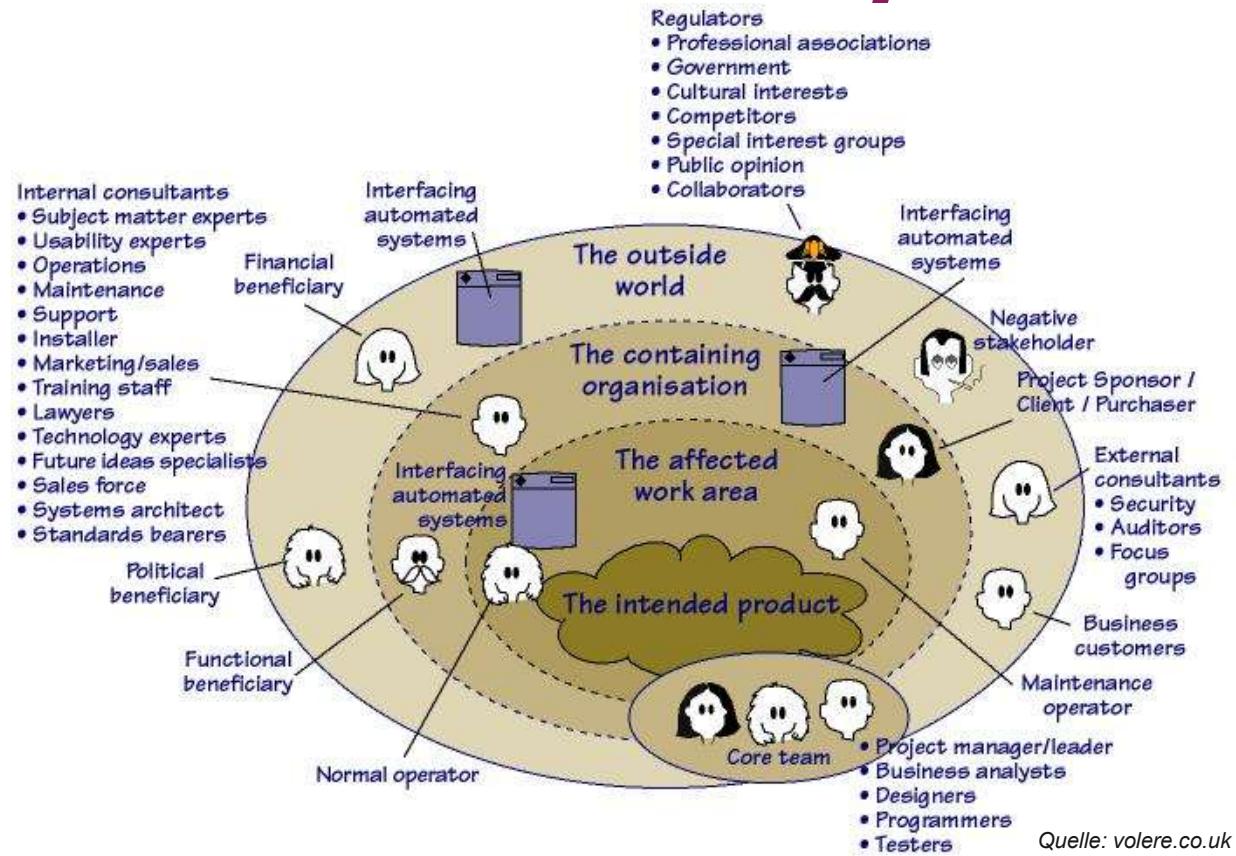
Sechs Fragen um Stakeholder zu finden



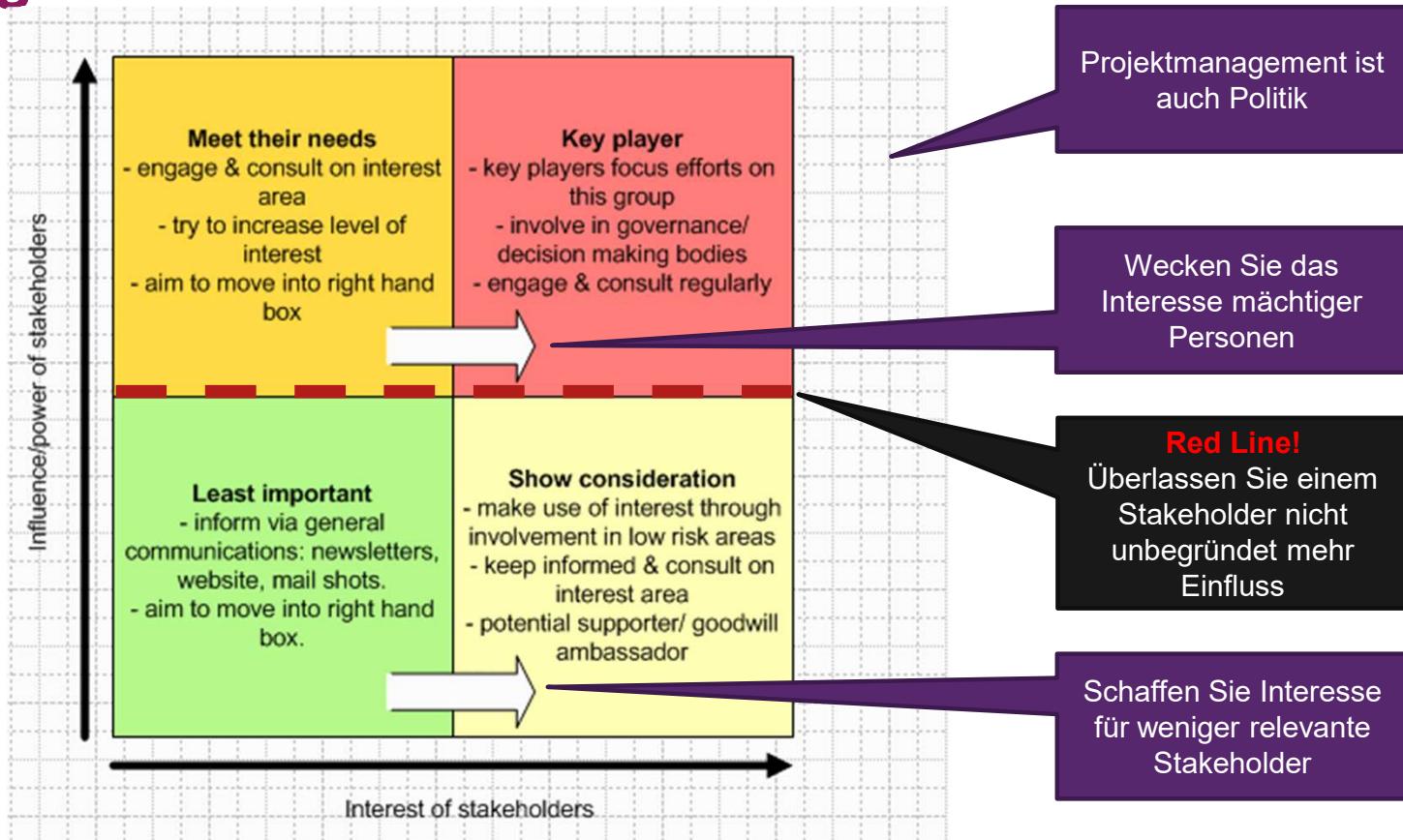
Diskurs: Gehört das Projektteam zu den Stakeholdern?

- In der Fachliteratur existiert keine einheitliche Meinung, ob das Projektteam selber zu den Stakeholdern gehört. Je nach Betrachtungsweise gehört das Team daher zu den Stakeholdern oder auch nicht.
 - Pro: Die sechs Fragen auf den Folien vorher passen auf die Mitglieder des Projektteams
 - Contra: Stakeholder sind Anspruchsgruppen die es zu identifizieren und analysieren gilt, für das Projektteam ist dies nicht notwendig
- Handhabung im Modul PmQm:
 - Projektteam ist in einer Stakeholder Map nicht aufzuführen
 - Wenn explizit danach gefragt wird, kann das Team ebenfalls zu den Stakeholdern gezählt werden.

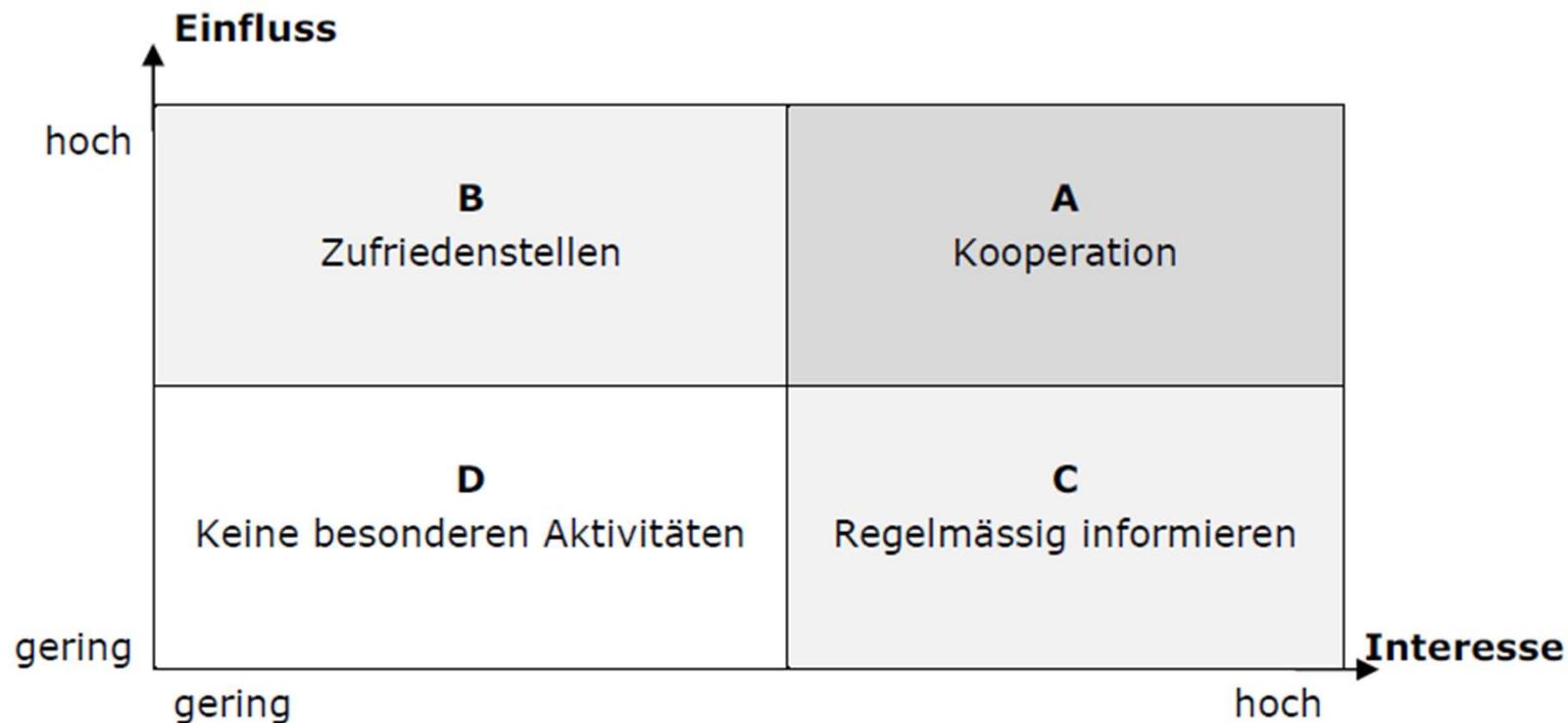
Wer ist alles Stakeholder eines Projekts?



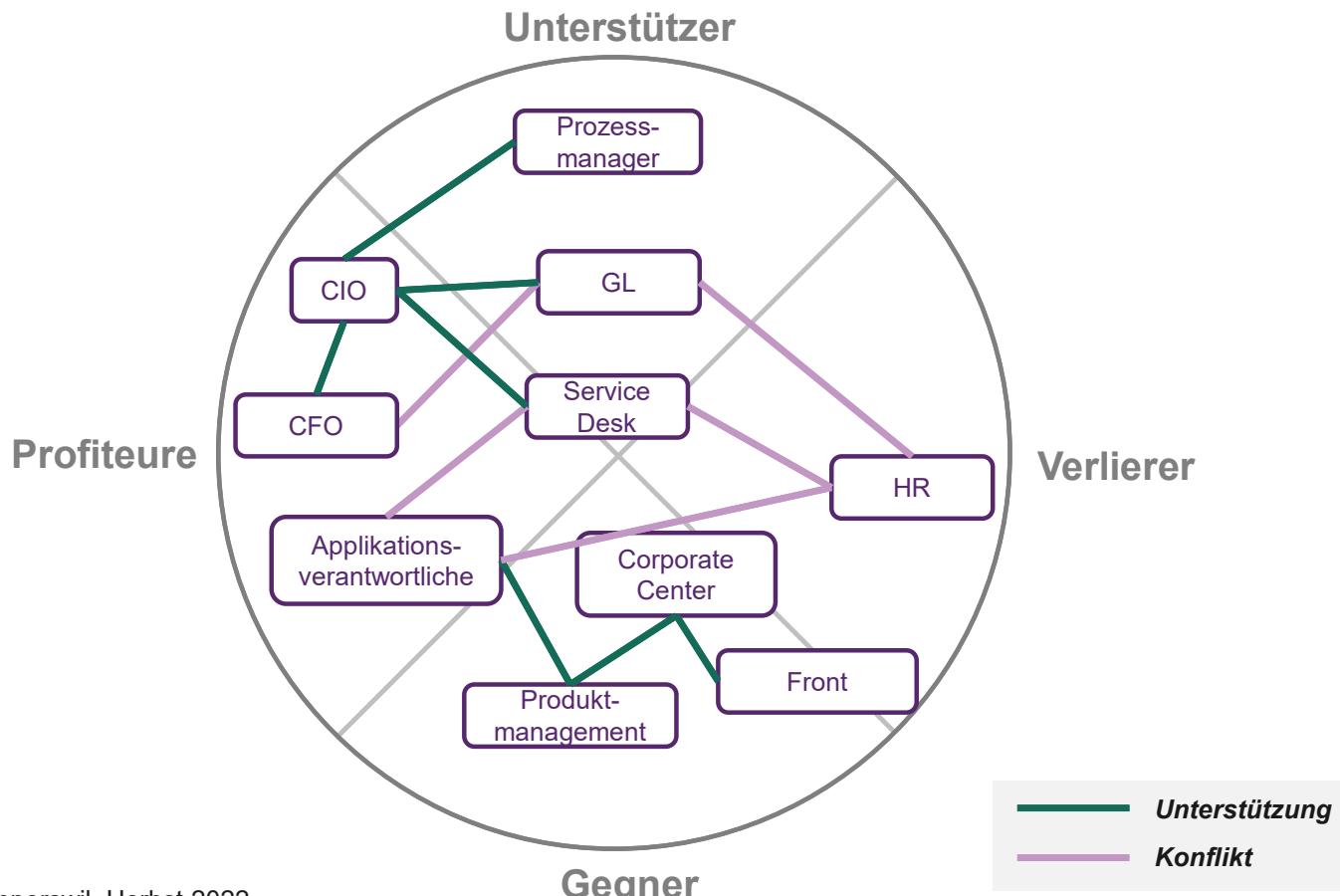
Abwägen von Einfluss und Interessen der Stakeholder



Klassische Einordnung nach IPMA



Beispiel: Einteilung in verschiedene Kategorien



Was ist Führung ?

Führung ist: Leute dazu bewegen, Dinge zu tun

Voraussetzungen:

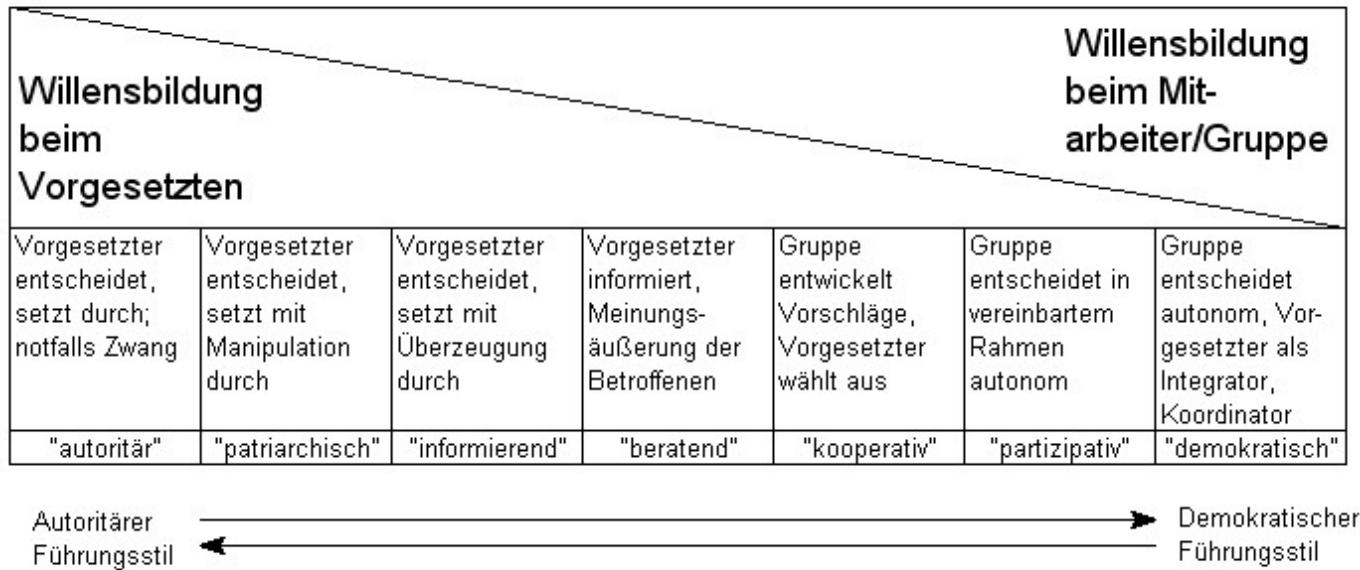
- Beziehung
- Sinn
- Möglichkeit

Wenn eine der Voraussetzungen nicht gegeben ist,
sind Probleme wahrscheinlich

Projektführung

- «Unter Führung (Leadership) versteht man die Anleitung und Motivation anderer bei der Erfüllung ihrer Funktionen oder Aufgaben im Dienst der Projektziele. Es handelt sich um eine für Projektmanager ausschlaggebende Kompetenz» NCB 4.1
- Sobald mehr als eine Person am gleichen Auftrag oder gleichen Produkt arbeiten, braucht es Führung oder mindestens Koordination
- Das «Ein-Mann-Vorhaben» benötigt Arbeitstechnik und Disziplin
 - Projektführung ist somit die Anleitung der Arbeitstechnik für eine Gruppe
- Heutige IT-Projekte sind keine «Ein-Mann-Vorhaben» mehr und benötigen daher zwingend Führung
- Wer selber eine gute Arbeitstechnik hat, beweist Methodenkompetenz und besitzt eine wichtige Eigenschaft eines guten Projektleiters

Führungsstile nach Tannenbaum

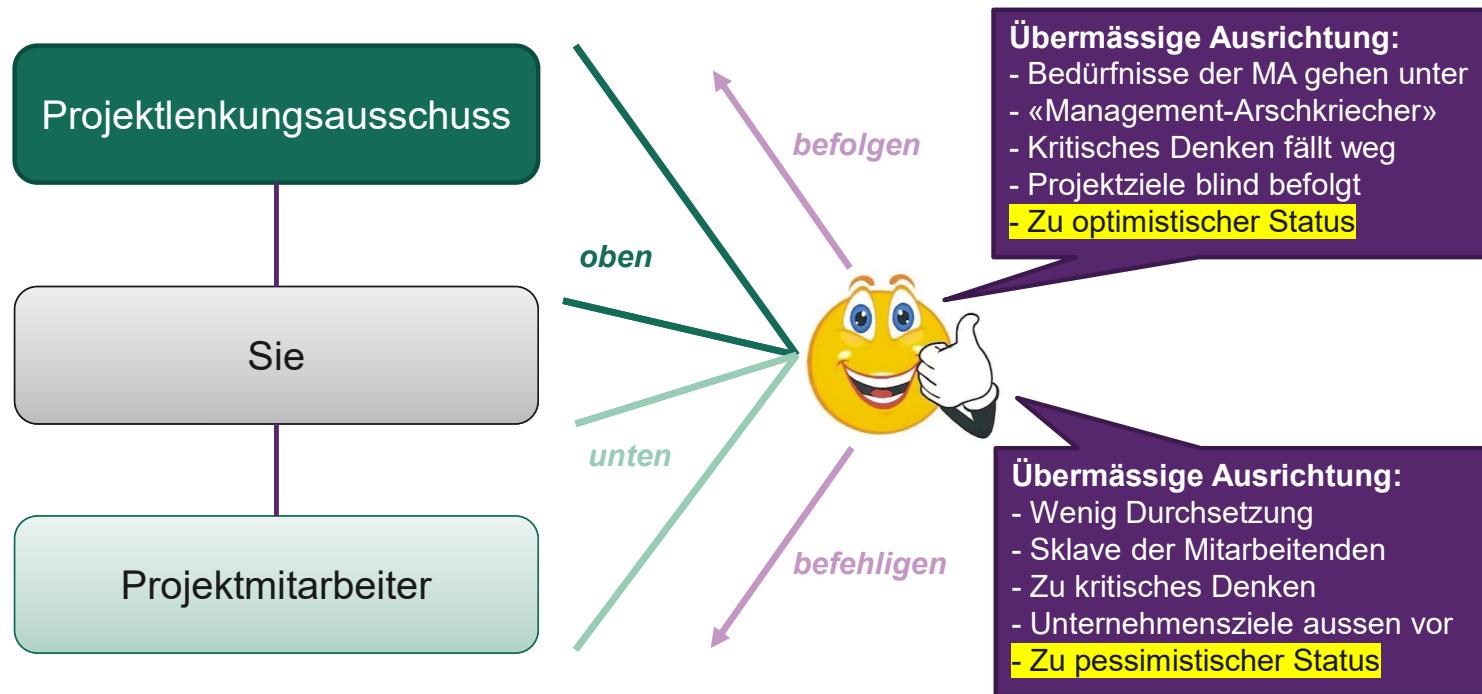


Sie müssen verschiedene Führungsstile kennen und entscheiden, welcher Stil für Ihre Persönlichkeit, das Projekt, das Team und die anderen Beteiligten in verschiedenen Situationen angemessen ist.

Quelle: Frank Tannenbaum, 1893 bis 1969

Diskussion: Welchen Führungsstil erachten Sie als bewährt?

Führungs-(Nahrungs-)kette – Gleichgewicht der Interessen



Führungsrythmus

- Der Führungsrythmus beschreibt die Häufigkeit mit welcher eine Führungskraft Informationen aufnimmt, beurteilt, entscheidet und Resultate kontrolliert



- Gestalten Sie den Rhythmus so selten wie möglich, aber so häufig wie nötig
- Ein hoher Rhythmus benötigt viel Zeit, gibt aber viel Einfluss
- Organisieren Sie strukturiert und strategisch, also z.B. wöchentliche Projektsitzung, monatliche Einzelgespräche, etc.
- Erhöhen Sie den Rhythmus bei anstehenden Herausforderungen

Ressourcenmanagement

- Das Ressourcenmanagement im Projekt bezeichnet die Kapazitäts- und teilweise auch die Einsatzplanung von Personen und Systemen
- In den Stammesorganisationen müssen Sie frühzeitig die Ressourcenanfragen tätigen
- Berücksichtigen Sie das Wissen der entsprechenden Mitarbeitenden, lassen sich Sie nicht mit dem «B-Kader» abspeisen
- Planen Sie so grob wie möglich, aber so detailliert wie nötig
- Führen Sie Änderungen stetig nach



Kommunikation im Projekt

- In den Folien vorher wurde implizit von Kommunikation gesprochen, da Stakeholder Management und Führung einen sehr grossen Zusammenhang mit Kommunikation haben
- Die Folgen Folien behandeln das Thema projektorientiert und tiefen das Thema Kommunikation somit ab
- Gute Projekt haben ein Kommunikationskonzept
- Im einem Kommunikationskonzept legen Sie fest, wer im Projekt was, wie, wann und warum kommuniziert

Projekt- und Organisationsanalyse

- Es gilt, die Projektorganisation zu analysieren und deren Aufbau darzustellen. Hierzu ist das **Organigramm des Projektes** notwendig, um die folgenden Fragen zu beantworten.
 - Wer ist für die Gesamtkommunikation innerhalb des Projektes verantwortlich?
 - Welchen Einfluss habe ich als Projektleiter auf die Kommunikation ausserhalb des Projektes?
- Wie stellen sich die Kommunikationswege zwischen den einzelnen Einheiten dar?
 - Wer sind die „Entscheider“ für ein Projekt?
 - Welches sind die Ansprechpersonen innerhalb der Projektorganisation?
 - Zudem sollen die wichtigen Termine und Meilensteine aufgelistet werden, die für die Kommunikation im Projekt relevant sind.

Beispiel einer Kommunikationsmatrix aus der Praxis

Stakeholder (Kategorie) / Adressat	Instrument(e)	Ziel / Zweck	Inhalt	Zeitpunkt / Periodizität	Verantwort- lich	Bemerkungen
Stakeholder	Welches Instrument wird eingesetzt?	Welche Ziele/Absichten werden mit der Massnahme verfolgt?	Welche Informationen sind für den jeweiligen Stakeholder relevant?	Zeitpunkt und Periodizität der Kommunikation massnahmen	Wer ist verantwortlich für die Kommunikation?	Allfällige Bemerkungen (z.B. Eigenschaften des Stakeholders, besondere Bedingungen etc.)
Bsp: Leiter HR	Bsp: E-Mail, Persönliches Gespräch, Projektnewsletter	Bsp: Mit regelmässigen Informationen versorgen, um Stakeholder als Befürworter zu halten.	Bsp: Informationen zum Projektfortschritt, Nutzen und Konsequenzen für HR.	Bsp: Projektnewsleter alle zwei Wochen, Persönliches Gespräch je nach Bedürfnis.	Bsp: PL	Bsp: Stakeholder bevorzugt E-Mail-Kommunikation da schwierige Erreichbarkeit.

Haben Sie eine Sitzungslandkarte?

Stakeholder (Kategorie) / Anlass	Ziel / Zweck	Zeitpunkt / Periodizität	Teilnehmer	Verantwortlich	Bemerkungen
Sitzung	Ziele und Zweck des Meetings	Wann und wie oft findet das Meeting statt?	Wer nimmt am Meeting teil?	Wer ist verantwortlich?	Kurzbeschreibung des Meeting (z.B. Traktanden, die regelmässig behandelt werden müssen).
Bsp: Teamsitzung Projektteam	Bsp: Update Projektstatus, Austausch allfälliger Schwierigkeiten, Abgleich Informationsstand	Bsp: Die Teamsitzung findet jeweils jede zweite Woche am Montag statt.	Bsp: PL, MA, MA Kunde	Bsp: PL	Bsp: Allgemeine Informationen zum Projektstatus, Schwierigkeiten, Personelles, Meilensteine etc.

Qualität vor Quantität

Schlechte Kommunikation

- Alle wissen nichts oder jeder weiss alles
- Viele Sitzungen behindern das Arbeiten
- Relevante Stakeholder sind nicht informiert

Gute Kommunikation

- Jeder weiss fundiert was er benötigt und ein wenig darüber hinaus
- Sitzungen sind eine Nebensache aber notwendig
- Die Stakeholder fühlen sich «abgeholt»

Schulz von Thun – Vier Ohren



Die Botschaft wird beim Empfänger gemacht

Kommunikation - Schulz von Thun

In der Kommunikation



Jede Äußerung ist wie ein
Klang aus verschiedenen Tönen.
Ein selektives Gehör verändert den Klang.

Die Botschaft wird bei
Empfänger gemacht

Projektmarketing

- Marketing bedeutet **zum Markte tragen**.
- Es ist weit mehr als reine «Werbung».
- Wobei es im Zusammenhang mit Projekten auf die «Werbemassnahmen» reduziert wird



www.inknowaction.com

Beispiele von Marketingaktionen

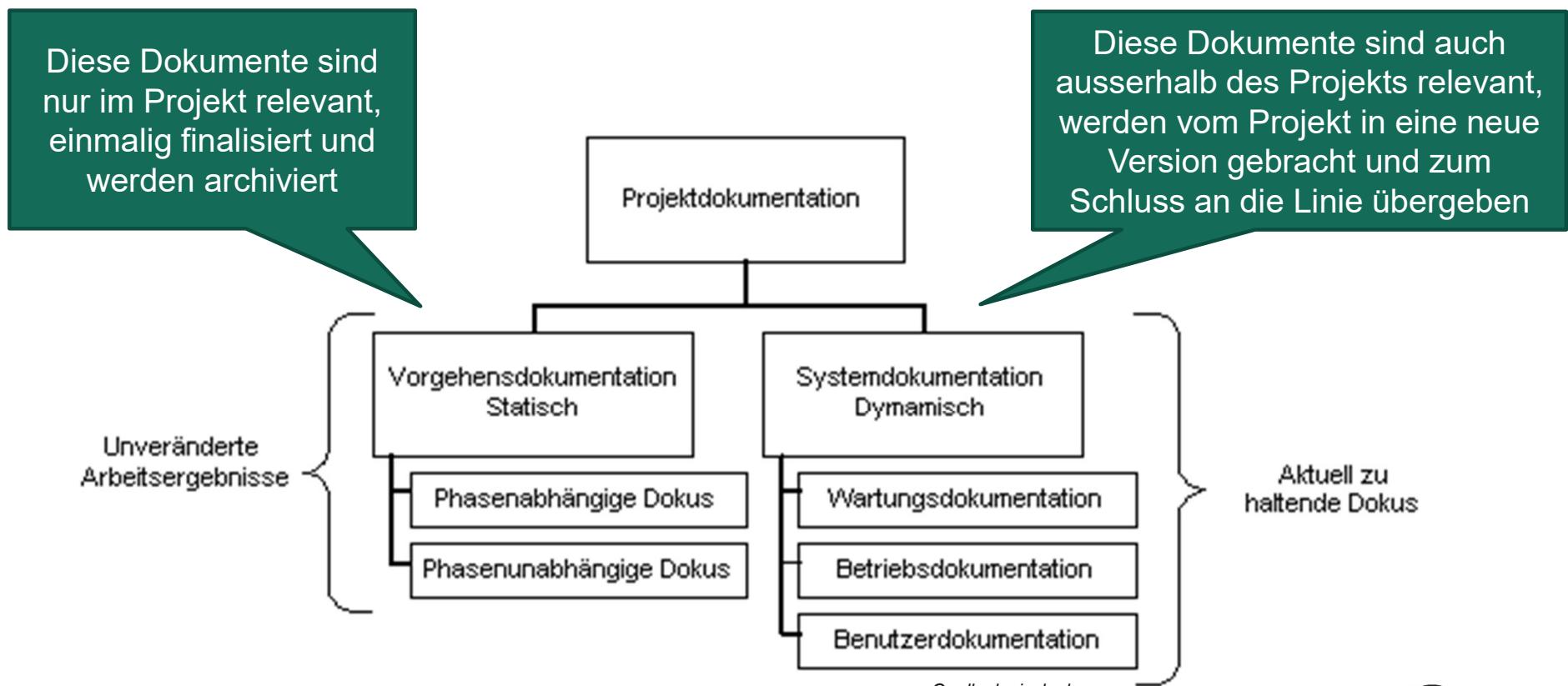
- Roadshows bei den Endbenutzer
- Film über das Projekt
- Infos an der Sitzung der Führungskräfte
- Tischset in der Kantine des Kunden
- Meldungen im Intranet des Kunden
- Bericht in der Kundenzeitung
- Erstellung einer Projektbroschüre
- Master-Slides in PowerPoint über das Projekt
- Events



Der Projektleiter als Botschafter

- Die **Präsentationsstärke eines Projektleiters** ist einer der massgeblichsten Faktoren im Projekt
 - Zwischenergebnisse müssen souverän präsentiert und Fragen kompetent beantwortet werden
 - Die Präsentation ist die wichtigste Visitenkarte des Projektleiters, viele Führungskräfte achten darauf und sind «Bauchentscheider»
 - Mit Präsentationen bauen Sie als Projektleiter Beziehungen zu den Stakeholdern auf
- Der Projektleiter vertritt hauptsächlich das Projekt gegen aussen und somit gegenüber den Kunden, Geldgebern sowie anderen Stakeholdern
- Bei wichtigen Ereignissen wird erwartet, dass der Projektleiter persönlich teilnimmt
- Wichtige Auftritte des Projektleiters mit dem Auftraggeber vorbesprechen

Projektdokumentation



Beispiele von Dokumenten

Produkt
Backlog

Vorgehen: Projektspezifische Dokumente

- Projektauftrag
- Projektplan
- Definition der Arbeitspakete
- Ressourcenplanung
- Business Case
- Statusberichte
- Projektorganisation
- Risikoliste
- Dokumentenplan

Sprint
Backlog

- Use Case Modell
- Grobkonzept
- Detailkonzept
- Datenmigrationskonzept
- Einführungsdrehbuch
- Testdrehbuch
- Befundliste
- Testabschlussbericht

System: Unternehmensspezifische Dokumente

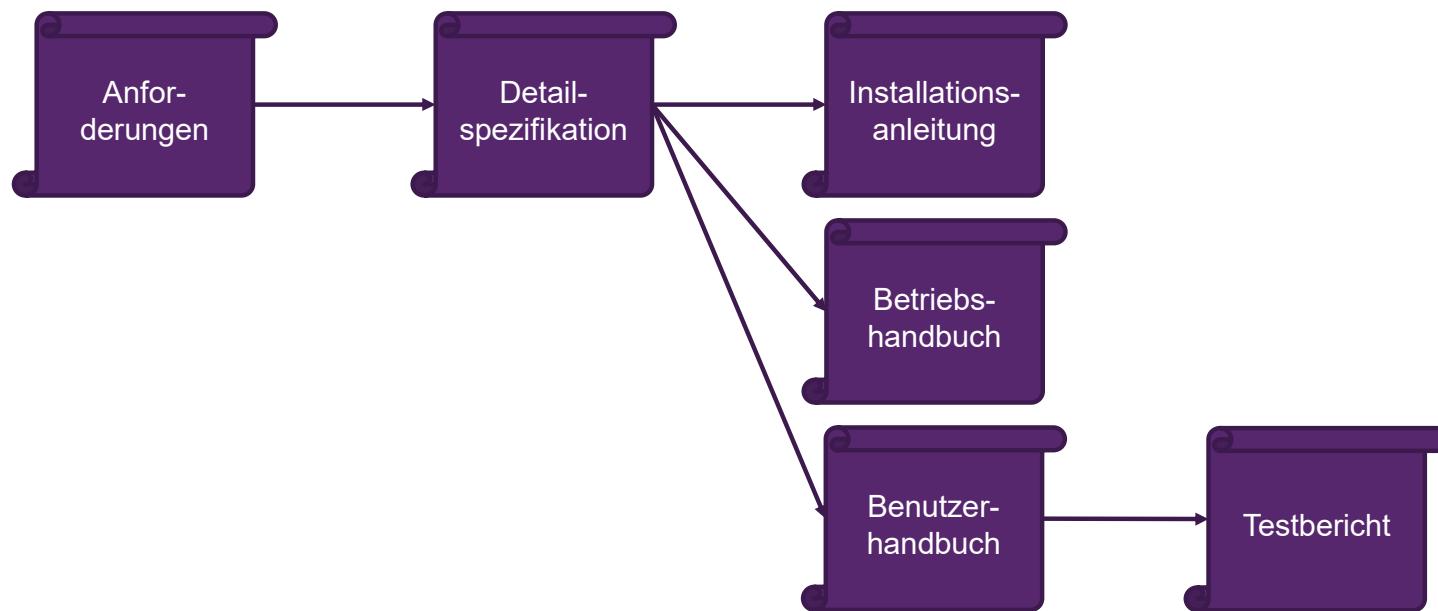
- Qualitätshandbuch
- Konfigurations-Management
- Geschäftsprozesse
- Vorschrift für GUI (CI)
- Betriebshandbuch
- Sicherheitskonzept

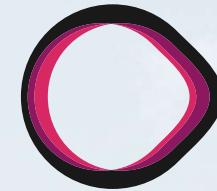
Phasenunabhängig

Lieferergebnisse der Phasen

Empfehlung Dokumentenplan

- Tipp: Definieren Sie gleich zu Beginn eines Projekts, welche Dokumente sie wem schuldig sind, zum Beispiel:





OST
Ostschweizer
Fachhochschule

Vorlesung - PMQM

12 – Lösungsfindung Teamarbeit

Werner Staub

Informatik

Ziele der heutigen Vorlesung

- Lerninhalte
 - Verschiedene Methoden zur Lösungsfindung
 - Reviewtechnik
 - Sitzungstechnik
 - Verhandlungstechnik
- Ziele: Die Studierenden...
 - kennen mehrere Methoden zur Lösungsfindung
 - verstehen als Vorbereitung auf die Übungsstunde den Ablauf eines Reviews
 - wenden die Sitzungsplanung an
 - erweitern ihr Wissen hinsichtlich Verhandlungstechnik

Methoden zur Lösungsfindung

Das methodische Vorgehen zur Findung von Lösungen mag auf den ersten Blick als übertriebenen scheinen (mit Kanonen auf Spatzen schießen)



Es gibt aber durchaus Situationen, wo eine geordnete Findung der einzige Ausweg ist oder die Bewertung nachvollziehbar sein muss. Auszug einiger Methoden:

Finden	Bewerten
<ul style="list-style-type: none">• Brainstorming• Methode 635• Morphologischer Kasten	<ul style="list-style-type: none">• Nutzwerkanalyse• Entscheidungsbaum

Brainstorming - Vorbereitung

- Zusammensetzung der Gruppe
 - breite Streuung des Fachwissens
 - keine zu grossen hierarchischen Unterschiede
- Bestimmung eines Protokollführers
 - geeignet zur schriftlichen Fixierung der geäusserten Ideen
- Bekanntgabe des Problems
 - besonders bei komplexen Problemen etwa zwei Tage vor der Sitzung
- Vorbereitung des Sitzungsraumes
 - Isolierung von äusseren Einflüssen
 - Vorbereitung von Hilfsmitteln wie Wandtafel oder Flipchart

Brainstorming - Regeln

- Diskussionsleiter
 - formuliert das Problem nochmals möglichst exakt (paraphrasieren)
 - überwacht den Ablauf der Sitzung
 - greift ein, wenn die Regeln nicht beachtet werden oder der Ideenfluss zu versiegen droht
- Protokollführer
 - schreibt alle Ideen auf
- Regeln für die übrigen Teilnehmer
 - Kritik ist verboten
 - Der Phantasie soll freien Lauf gelassen werden, um möglichst kühne Ideen zu erhalten
 - Je mehr Ideen, desto besser: **Quantität geht vor Qualität**
 - Die Ideen anderer sollen aufgenommen, kombiniert und weiterentwickelt werden

Brainstorming - Nachteile

Brainstorming ist als Kreativ-Methode sehr schlecht dafür geeignet, Teilnehmer dazu zu bringen, die Perspektive zu wechseln und lateral (intuitiv) zu denken. Die originellsten und oft nützlichsten Ideen entstehen aber gerade dann, wenn wir als Gruppe lateral denken und unsere Denkmuster verlassen. So braucht man für ein gutes Brainstorming einen sehr guten Moderator. Ohne richtige Moderation, die das Denken der Gruppe steuert, fällt ein produktives Brainstorming schwer.

Das größte Problem jedoch ist das sogenannte Production Blocking : Einer redet, alle hören zu und sind geistig beansprucht. Ein Vortragender blockiert das Denken der gesamten Gruppe. Das ist der größte Nachteil, aber eben auch ein wichtiger Bestandteil eines Brainstormings: Wenn nicht **alle** von **allen anderen alle** Ideen zu hören bekommen, können diese Ideen keine neuen Ideen triggern.

Quelle: <https://www.foerderland.de/organisieren/news/brainstorming>

Brainstorming eignet sich nicht, wenn nicht **Kreativität**
sondern **Fachwissen** gefragt ist.

Methode 635

- Vorgehen
 - Jedes Mitglied der aus genau 6 Teilnehmern bestehenden Gruppe schreibt 3 Ideen auf ein Blatt Papier, das 5 mal weitergereicht wird
 - Jeder Teilnehmer ergänzt das Blatt um 3 weitere Ideen und lehnt sich möglichst an das vorhandene an
- Folgende Zeitvorgaben haben sich als geeignet erwiesen:
 - 1. Runde: 5min
 - 2. Runde: 6min
 - 3. Runde: 7min
 - 4. Runde: 8min
 - 5. Runde: 9min
 - 6. Runde: 10min

Methode 635 bedeutet also:

- 6 Teilnehmer
- 3 Ideen
- 5 mal weitergeben

Methode 635 - Beispiel

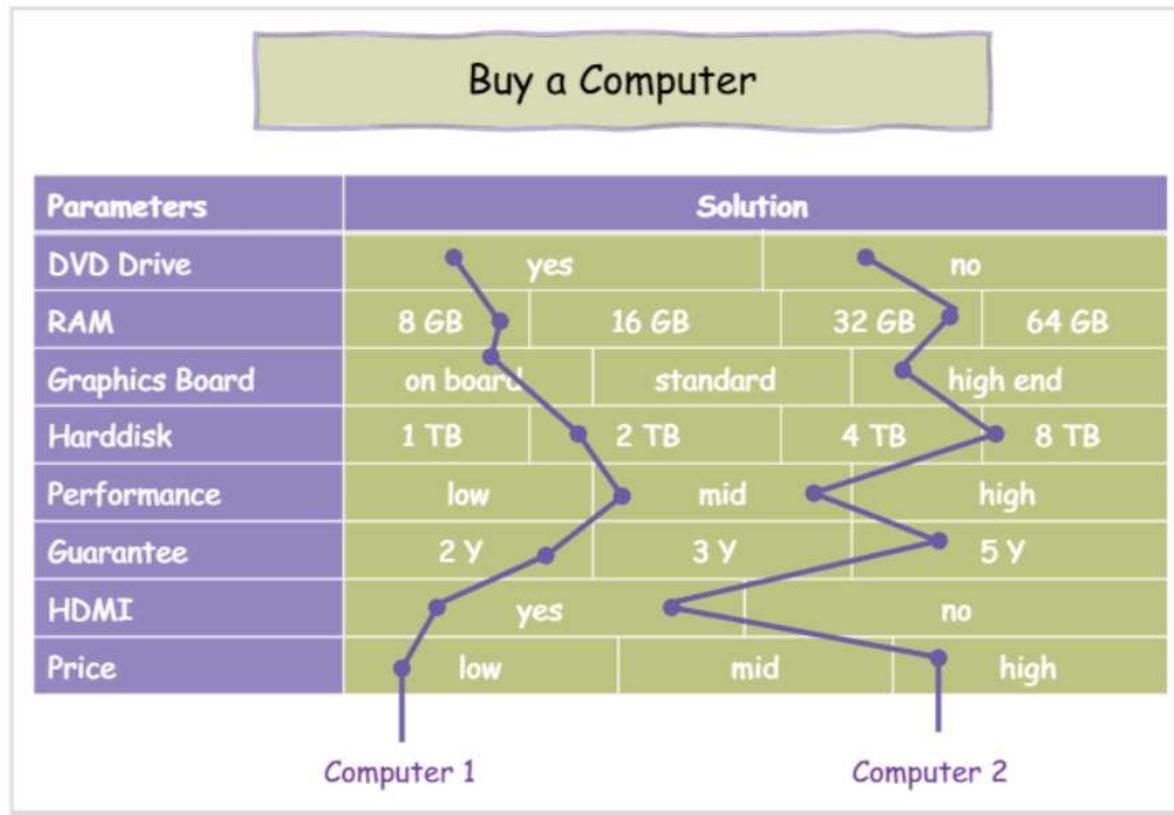
Problem: Wie können wir unseren Absatz für Schokopudding steigern?			
Teilnehmer: Herr L, Frau S, Frau G, Herr K, Frau E, Herr M			
Name	Idee	Idee	Idee
Herr L	Neue Verpackung	Treuepunkte	Weniger Zucker
Frau S	In Schokobohnenform	Treuepasse	Gesundheitsaspekt betonen
Frau G	Pudding in Schokobohnen	Treuepunkte und Treuepass für ganze Produktlinie von Puddings	Gesundheit steigert Wohlbefinden, Schoko macht glücklich
Herr K	Schockobohnen müssen nicht gekühlt werden --> Verkauf am Kassenregal	Prämien, wenn Punktemengen erreicht werden	Aufklärungskampagne über Gesundheitsaspekte von Schokolade
Frau E	Sammelalbum zu Schockopudding-Bohnen an Kasse den Kindern mitgeben	Über Punkte können Sammelsticker erworben werden	Ausgeglichene glückliche und gesunde Familien durch Konsum von Schokopudding
Herr M	Ausgefülltes Sammelalbum gibt freien Eintritt in Zoo, Museum, etc.	Die ersten Fünf, die tausend Punkte erreichen, werden ins Werk eingeladen	Familienausflüge in die Natur, ermöglicht durch unseren Schokopudding

Quelle: Buch «Projektmanagement (IPMA)
von Karen Dittmann / Konstantin Dirbanis

Morphologischer Kasten

- Die Morphologie wurde vom Schweizer Forscher F. Zwicky entwickelt (er spricht dabei auch von "diskursiver Problemlösung"). Mit dieser Methode soll durch das systematische Zusammenstellen aller logisch denkbaren Möglichkeiten über einen bestimmten Bereich in Form einer Tabelle das unvollständige, eingefahrene, fixierte Denken vermieden werden.
- Hauptinstrument der Morphologie ist ein "morphologischer Kasten". Dabei wird in vier Schritten vorgegangen:
 1. Problem als Überschrift des Kastens notieren
 2. Zerlegung in Teilprobleme oder Handlungsfeldern
 3. Lösungsvarianten pro Teilproblem oder Handlungsfeld ermitteln
 4. Kombination herausfinden, zum Beispiel jede Person kriegt einen andersfarbigen Stift

Morphologischer Kasten - Beispiel



Quelle: Buch «Projektmanagement (IPMA)
von Karen Dittmann / Konstantin Dirbanis

Nutzwertanalyse

- Die Nutzwertanalyse weist den gewichteten Nutzen pro Variante aus
- Als Bewertung findet man in der Schweiz oft das Schulnotensystem (1 = sehr schlecht, 6 = sehr gut)
- Folgende vier Schritte benötigen Sie:
 1. Kriterien (oder Anforderungen) auflisten
 2. Kriterien gewichten
 3. Jedes Kriterium für jede Variante bewerten
 4. Punktzahl zusammenzählen
- Bei knappen Unterscheidungen empfiehlt es sich die Sensitivität der Gewichtung mittels kleinen Verschiebungen zu überprüfen

Wird oft bei öffentlichen
Ausschreibungen
angewendet.
(Submissionsverfahren)

Bewerten

- Nutzwerkanalyse
- Entscheidungsbaum

Beispiel einer Nutzwertanalyse

Kriterien
auflisten

Noten setzen

Exceldatei

Kriterien	Gewicht	Anbieter 1		Anbieter 2		Anbieter 3	
		Note	Punkte	Note	Punkte	Note	Punkte
Bewertung Hersteller	10%	4	0.4	6	0.6	5	0.5
Funktionalität	25%	3	0.75	6	1.5	5	1.25
Look & Feel	15%	6	0.9	5	0.75	5	0.75
Wartbarkeit	20%	2	0.4	6	1.2	5	1
Initialkosten	10%	6	0.6	2	0.2	5	0.5
Betriebskosten	20%	6	1.2	2	0.4	5	1
	100%			4.25		4.65	

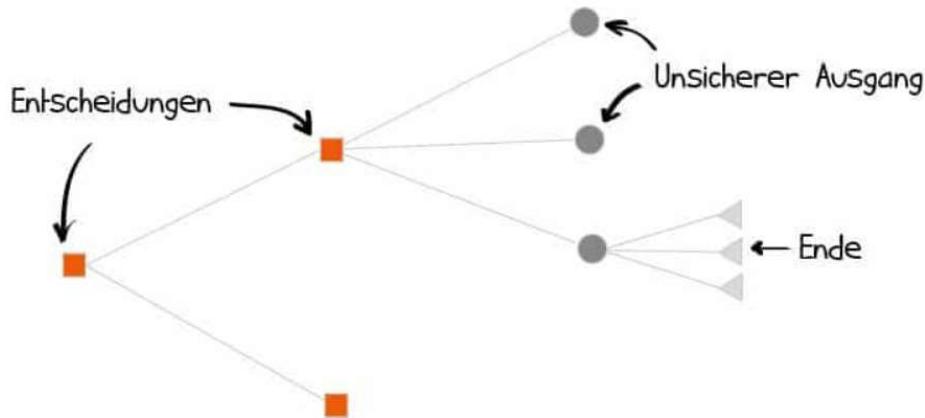
Gesamtgewicht
beachten

Gesamtbewertung
ablesen

Entscheidungsbaum

Graphische Darstellung von Entscheidungsoptionen

- Im Entscheidungsbaum wird ein Problem systematisch dargestellt



- Der Baum ist immer von links nach rechts aufgebaut.
- Die Alternativen zweigen den Baum nach rechts auf.
- Rechtecke stehen für Entscheidungen.
- Kreise stehen für Möglichkeiten/Unsicherheiten.
- Dreiecke stehen für das Ende des Zweigs.

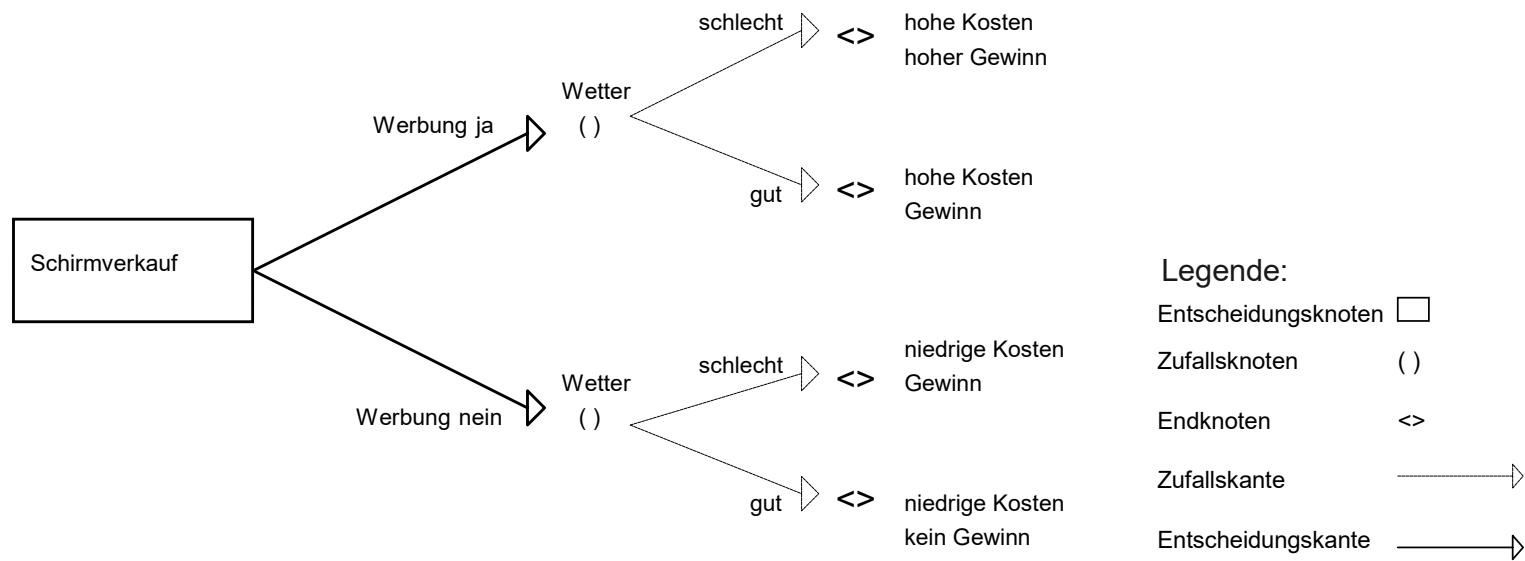
Quelle: <https://projekte-leicht-gemacht.de/>

Bewerten

- Nutzwerkanalyse
- Entscheidungsbaum

Beispiel eines Entscheidungsbaums

Ein Schirmgeschäft überlegt sich, ob es eine Werbeaktion starten soll. Dabei wird das Wetter (unbeeinflussbares Ereignis) mit in die Überlegung einbezogen. Die Entscheidung lässt sich wie folgt darstellen (ohne Angaben von Wahrscheinlichkeiten):



Reviewtechnik

- Ein Review ist ein systematischer Check durch einen Stakeholder oder durch unabhängige Personen
- Beispiele von Review-Objekten:
 - Dokumente
 - Code
 - Konzept
 - System Architektur
 - Software Architektur
- Ein Review kann als Sitzung oder schriftlich als Protokoll durchgeführt werden
- Man darf auch loben und muss nicht nur bemängeln
- Das Objekt wird nach dem Review abgenommen, unter Vorbehalt abgenommen oder zurückgewiesen

Beispiel Reviewprotokoll (vereinfachte Version)

Reviewprotokoll		Objekt	Testdokument	Abnahme	Ja, mit Vorbehalt	
Nr.	Datum	Wer	Seite/Kapitel	Bemerkung	Gewichtung	Bemerkung
1	01.10.2015	C. Fuchs	5 / 3.2	Die Unterscheidung zwischen branch und trunk ist nicht korrekt formuliert, bitte entsprechenden den Definition übernehmen	Critical	
2				oder: «...» ist eine sehr gute Beschreibung und muss unbedingt so beibehalten werden.		
3						
4						
5						
6						
7						
8						
9						
10						

Review-Objekt benennen

Ob abgenommen oder nicht

Excelfile

Eintrag eines Review-Befunds

Befund verständlich beschreiben und vom Autor eine Aktion fordern

Gewicht des Befunds hinterlegen

Sitzungstechnik

- Der Sitzungsleiter bereitet die Sitzung vor und eröffnet. In Projekten ist dies in der Regel der Projektleiter
- Folgende Fragen können Sie sich als Vorbereitung stellen:
 - Ist die Notwendigkeit der Sitzung gegeben?
 - Sind die Ziele bekannt?
 - Welche Teilnehmer sind notwendig?
 - Haben die Teilnehmer auch Zeit für die Sitzung? Brauchen sie eine spezielle Vorbereitung?
 - Was muss am Ende für jeden heraus kommen?
 - Wie sind die Abhängigkeiten zum Umfeld?
 - Welche Traktanden und in welcher Reihenfolge werden besprochen?
- Frühzeitige Festlegung des Termins und persönliche Einladung versenden mit Traktandenliste und Zielen

So agieren Sie als Sitzungsleiter

- Protokollschreiber bestimmen
- Wo notwendig, Protokoll der letzten Sitzung verabschieden
- Einleiten und Traktanden mit den Zielen nochmals präsentieren
- Zeitrahmen einhalten
- Sitzung moderieren und nicht bestimmen
- Bei Abschweifungen zurück zum Thema finden
- Aufmerksame Beobachtung der Teilnehmer, jeder Teilnehmer muss zu Wort kommen
- Störungen unterbinden
- Festhalten der Entscheide, Termine und Tätigkeiten
- Nie persönlich werden

Die Teilnehmer an einer Sitzung



Bulldoge

- Widerspricht aggressiv
- Kritisiert destruktiv



Reh

- Hält sich zurück
- Schweigt am liebsten



Pferd

- Sanftmütig und selbstsicher
- Zielorientiert



Igel

- Opponiert möglichst immer
- Will sich nicht integrieren



Affe

- Weiss alles
- Unterbricht mit Einwänden



Flusspferd

- Uninteressiert, wortkarg
- Unterbricht mit Einwänden



Frosch

- Redet und redet und redet
- Redet um des Reden willens



Giraffe

- Eingebildet
- Nicht kritikfähig

Quelle: binetsch-consulting.ch / frox.ch

Führungsrythmus

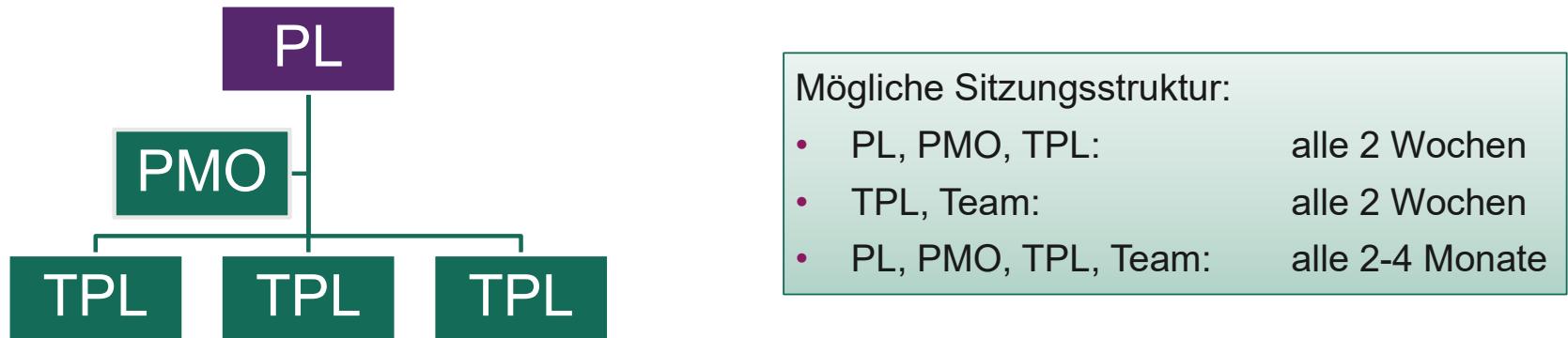
- Projektteams werden in der Regel mit Sitzungen geführt
- z.B. in Scrum das Daily Meeting
- Sie als Projektleiter definieren die «Sitzungslandkarte» im Projekt



- Legen Sie einen passenden Rhythmus fest, ändern Sie diesen bei Bedarf wie ein Dirigent in einem Orchester

Sitzungslandkarte - Beispiel

- Sie leiten ein grosses Projekt. Drei Teilprojektleiter (TPL) und ein Project Management Office (PMO) unterstützen Sie hierbei. Pro Teilprojekt sind etwa 8 Personen im Team.



In Scrum

- Sie sind nicht Projektleiter, aber vielleicht Produkt Owner
- Sie müssen den Sitzungsrhythmus nicht festlegen – ist durch Scrum vorgegeben
- Der Scrum Master kümmert sich um die Durchführung
- Das Team ist selbstverantwortlich und managt die Sitzung selbst

Verhandlungstechnik

Wir erinnern uns an Schulz von Thun



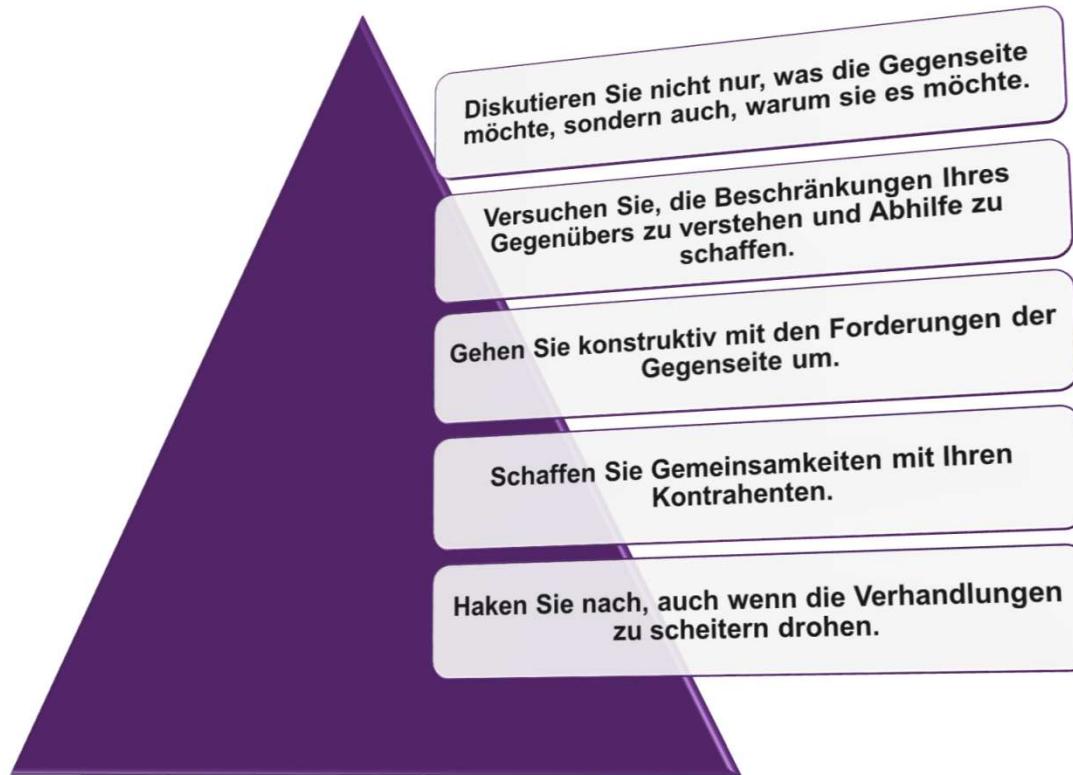
Definieren Sie die gewünschten Ziele vorgängig und legen Sie die Mindestziele fest, welche Sie aber verdeckt halten

Verhandlungskommunikation



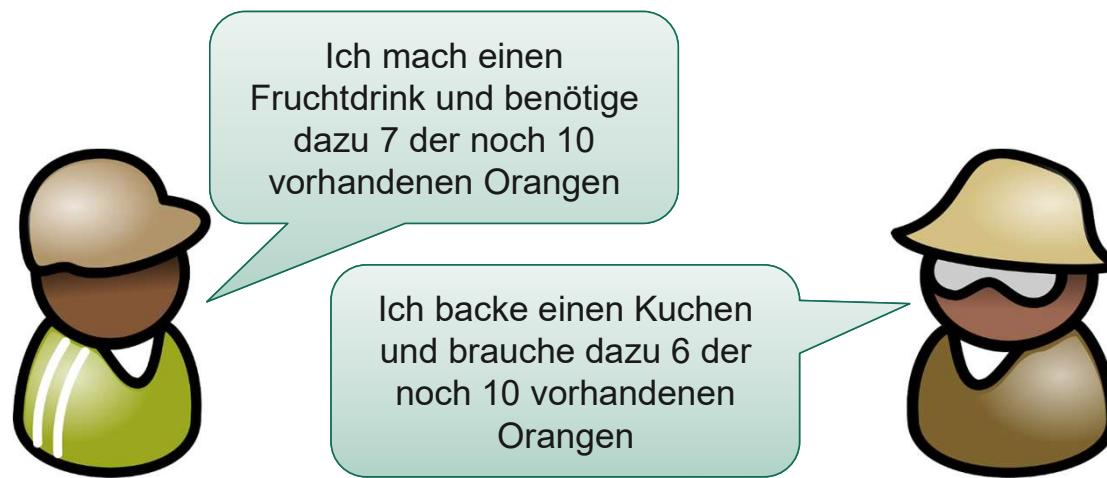
- Sorgen Sie schon im Vorfeld für eine gute Beziehungsebene
- Verhandeln Sie möglichst auf der Sachebene
- Vermeiden Sie Selbstoffenbarungen
- Verzichten Sie auf Appelle

Die Harvard-Methode – 5 Grundsätze



Kompromiss vs. Konsens

- Suchen Sie den Konsens, je stärker die Win-Win-Situation, desto stabiler wird die Abmachung sein



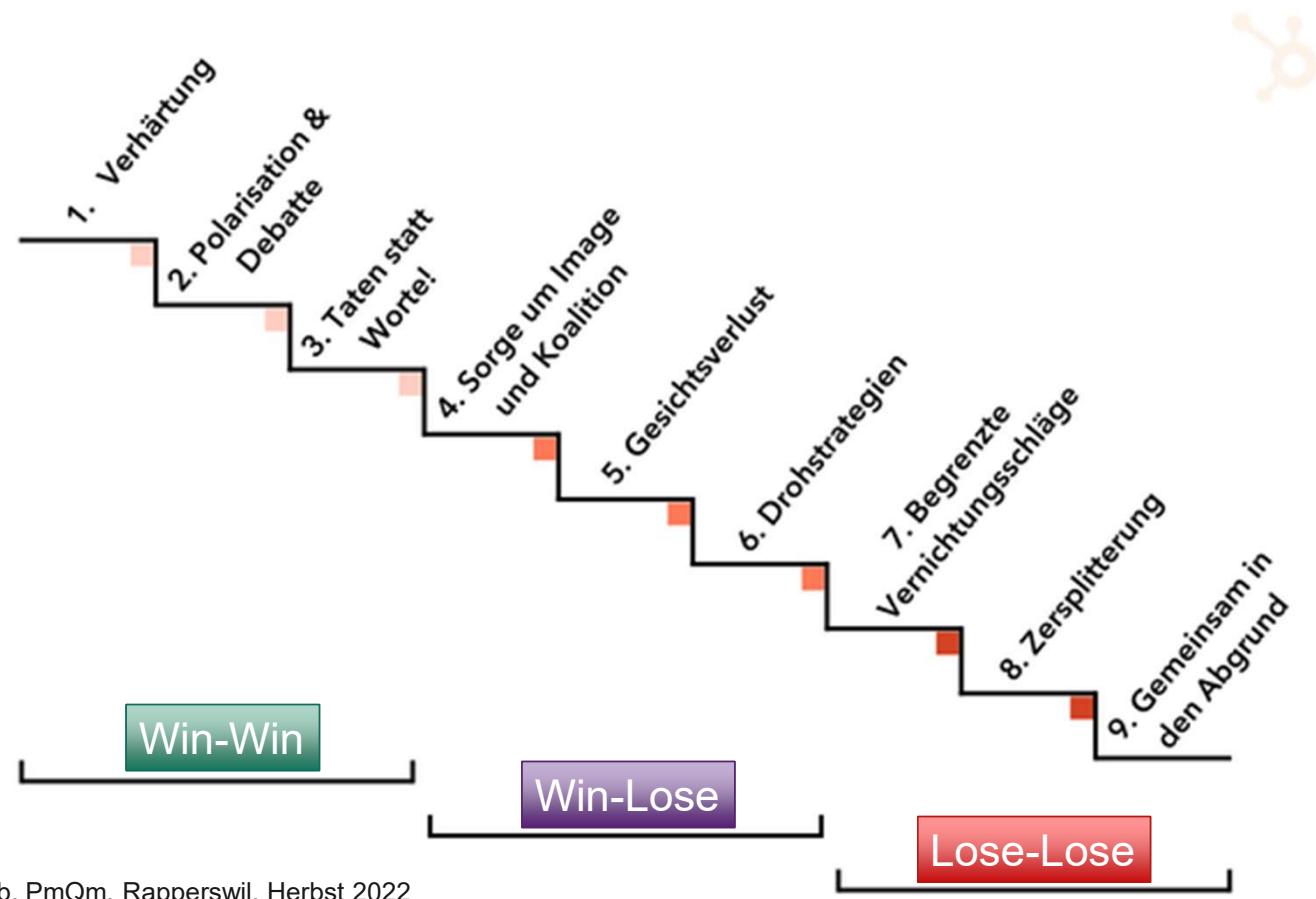
Ein paar Tipps

- Verkaufen Sie dem Anderen seine Vorteile und nicht Ihre
- Zeigen Sie ihm seine Konsequenzen auf, wenn er diese Vorteile nicht hat



Verzichten wir nun auf den Apero oder auf den Kuchen oder finden zusammen eine andere Lösung? z.B. Fruchtdrink aus Äpfeln oder einen Apfelkuchen

Konflikt-Eskalation nach Friedrich Glasl



Konflikt-Eskalation nach Friedrich Glasl



1. Ebene

- Stufe 1 – Verhärtung: Konflikt beginnt mit Spannungen
- Stufe 2 – Debatte: Meinungsverschiedenheit, Druck aufsetzen
- Stufe 3 – Taten statt Worte: Mitgefühl geht verloren, nonverbal

Win-Win



2. Ebene

- Stufe 4 – Koalitionen: Gegner angreifen/denunzieren
- Stufe 5 – Gesichtsverlust: Unterstellungen anbringen
- Stufe 6 – Drohstrategien: Eigene Macht veranschaulichen

Win-Lose

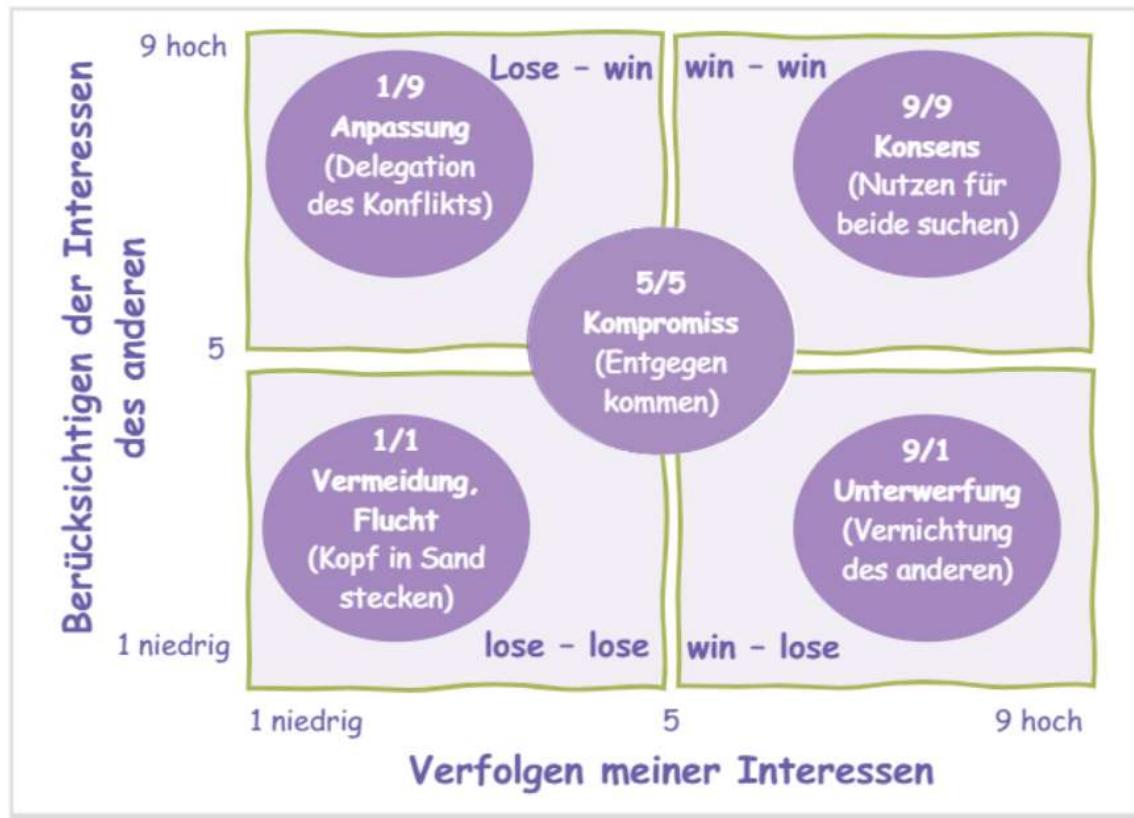


3. Ebene

- Stufe 7 – Begrenzte Vernichtung: Gegner Schaden zufügen
- Stufe 8 – Zersplitterung: System des Gegners vernichten
- Stufe 9 – Gemeinsam in den Abgrund: Eigene Vernichtung kalkulieren

Lose-Lose

Konflikt Grid nach Virginia Satir



Kommt keine Lösung zustande, fallen die Konfliktparteien auf einen der drei grundlegenden biologischen Stile zurück:

- Flucht – Anpassung
- Unterwerfung – Durchsetzung
- Kampf - Rückzug

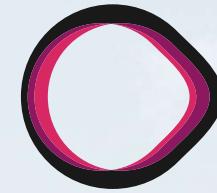
Quelle: Buch «Projektmanagement (IPMA)
von Karen Dittmann / Konstantin Dirbanis

«Business Case» für Konflikte

- Ist es sinnvoll und nötig, jeden Konflikt auszutragen?
- Wann lohnt sich der Aufwand im Verhältnis zu Nutzen nicht mehr?
- Kann ich in einem solchen Fall einen anderen Weg gehen, so dass ich auf die Lösung nicht angewiesen bin, oder gar den Konflikt umgehen kann?

Gelassenheitsgebet vom Theologen Reinhold Niebuhr:

**Gott, gib mir die Gelassenheit, Dinge hinzunehmen, die ich nicht ändern kann,
den Mut, Dinge zu ändern, die ich ändern kann,
und die Weisheit, das eine vom anderen zu unterscheiden.**



OST
Ostschweizer
Fachhochschule

Vorlesung - PMQM

13 – IT-Servicemanagement nach ITIL

Werner Staub

Informatik

Ziele der heutigen Vorlesung

- Lerninhalte
 - IT-Servicemanagement
 - IT-Betriebsprozesse / Praktiken kennen
 - ITIL als Prozess Framework und Quasi-Standard
- Ziele: Die Studierenden...
 - Zweck und Nutzen von IT Servicemanagement kennen
 - kennen das ITIL als Framework
 - verstehen, welche Eigenschaften im Betrieb eines IT Systems wichtig sind

Motivation für ITIL

- Unternehmen sind zunehmend von der IT abhängig
→ wachsender Bedarf an IT Services
- Information sind das wertvollste Gut einer Unternehmung
→ Sicherheit (Vertraulichkeit, Verfügbarkeit, Richtigkeit)
→ Cyber Crime wird heute als grösstes Risiko eingeschätzt
- Trend weg von der Entwicklung der IT Anwendungen
hin zum Management von IT Services (SaaS)
 - Eine Anwendung unterstützt erst dann die Businessziele, wenn sie voll einsetzbar ist (fit for purpose ; fit for use) → Betrieb, Support, Wartung, Anpassungen
- ITIL (Information Technology Infrastructure Library)
- defacto Standard
- Good Practice

Ziele von IT Servicemanagement

- zuverlässige und konsistente IT Services
 - mit hoher Qualität
 - zu tragbaren Kosten
- Abstimmung der IT Services auf die Geschäftsanforderungen
 - Kostenoptimierung
 - Qualitätssicherung (steuerbares Qualitätsniveau)
 - Betriebsoptimierung
- Trennung von Entwicklung und Betrieb
 - neuer Trend: **Dev-Ops**
- Definierte Schnittstelle zum Business
(Kunde / Benutzer)
 - → neuer Trend: **Agile IT ; Produkte-Teams**

Guiding Prinzipiels

ITIL beschreibt eine Liste von Prinzipien, die zur erfolgreichen Umsetzung von Services und Produkten angewandt werden können. Diese sind generisch und können universell eingesetzt werden.

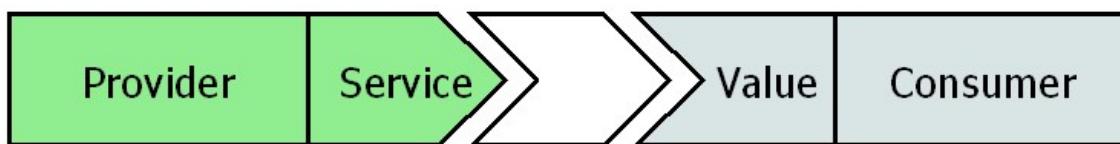
- Fokussierung auf den Nutzen
- Starte an dem Punkt, an dem du bist
- Mache kleine Schritte und prüfe sie regelmässig
- arbeite zusammen und fördere die Visibilität
- Denke und handle ganzheitlich
- Halte es einfach und praktikabel
- Optimiere und automatisiere

Kann auch gut auf Projekte angewandt werden

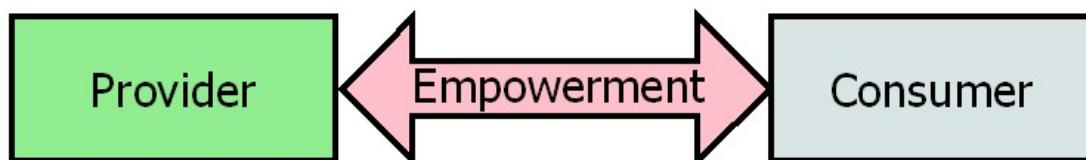
Business Nutzen

Return on Investment (ROI) aus Sicht des Kunden:

Der Kunde will seine Kosten möglichst bald durch den zusätzlichen Nutzen kompensieren können



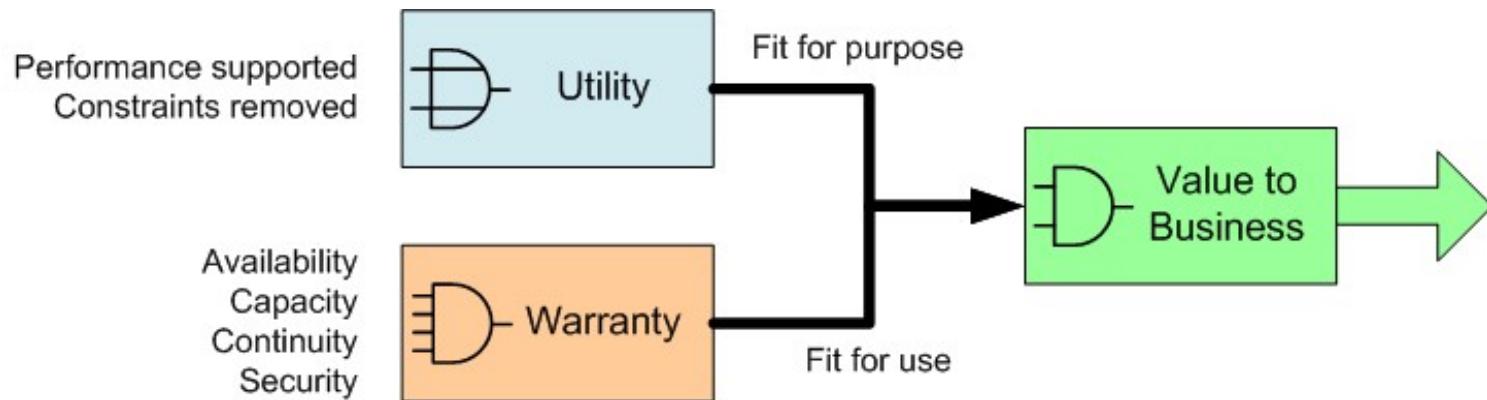
In einer sehr einfachen Provider-Consumer-Beziehung stellt ein Provider einen Service zur Verfügung, der von einem Consumer angewendet wird.



Grössere Organisationen setzen eine partnerschaftliche Beziehung zwischen Provider und Consumer voraus.

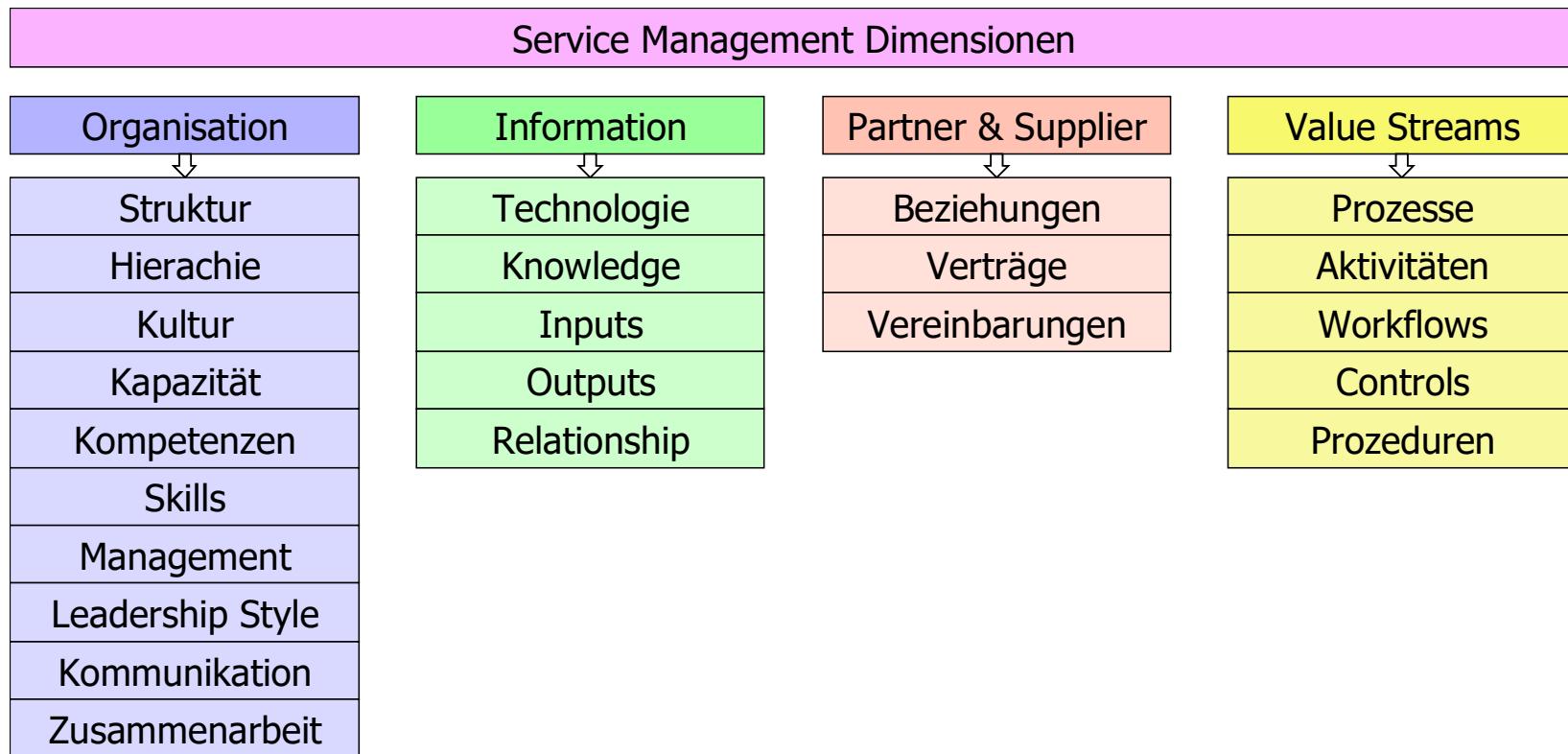
Utility & Warranty

Utility → Funktionalität (fit for purpose)
Warranty → Garantie (fit for use)

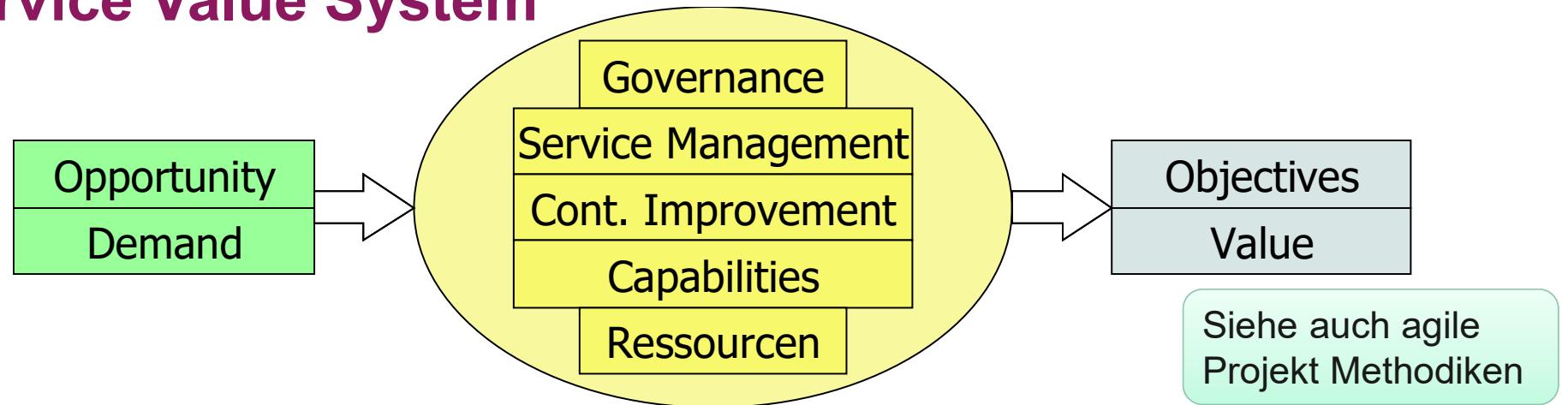


Die Utility wie auch die Warranty müssen erfüllt sein, damit der gewünschte Nutzen entstehen kann. Die Utility wird oft sehr genau in einem Pflichtenheft oder einer Service Vereinbarung festgelegt. Dabei kommt der Warranty oft zu wenig Beachtung zu.

Vier Service Management Dimensionen



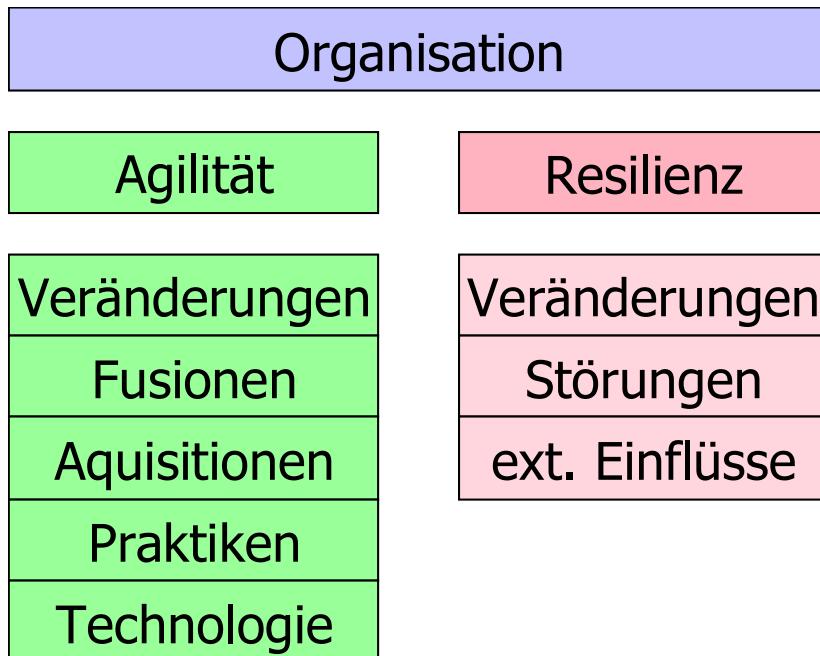
Service Value System



Die grösste Herausforderung einer Organisation in einem agilen Umfeld sind Silos. Diese sind oft change resistant und verhindern den schnellen Zugriff auf Informationen und Spezialisten. Solche Silos können auch durch Prozesse oder Praktiken (Practices) entstehen, wenn diese nicht genügend flexible Schnittstellen aufweisen. Silos reduzieren die Effizienz und erhöhen die Kosten und die Risiken.

ITIL gibt kein fixes System von Value Streams vor. Organisationen müssen in der Lage sein, diese selbst zu gestalten und regelmässig und schnell den aktuellen Anforderungen anzupassen. Die umfassenden Operationen sind in ITIL definiert durch leitende Prinzipien (guiding Principles) beschrieben.

Agilität und Resilienz einer Organisation



Agilität und Resilienz sind zwei wichtige Eigenschaften einer Organisation.

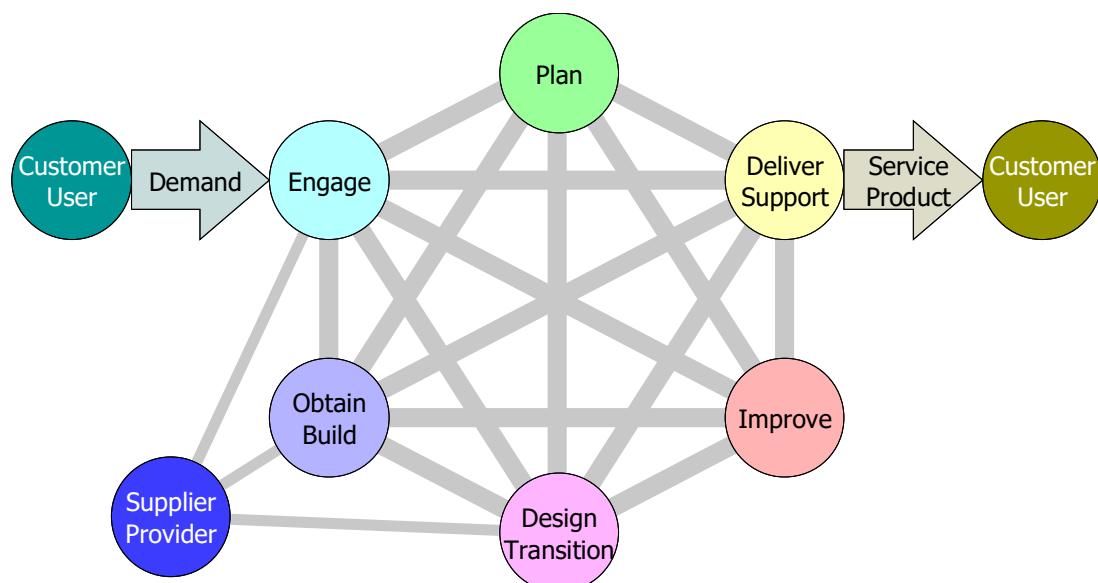
Agilität ist nötig, um schnell und flexibel interne Veränderungen umsetzen zu können.

Resilient ist wichtig, um robust auf externe Veränderungen reagieren zu können.

Das gilt im Übrigen auch für Menschen und Teams (z.B. Projekt-Teams)

Service Value Chain

Von den Kundenanforderungen
zum Kundennutzen



Plan: Gemeinsames Verständnis, aktueller Status und Richtung

Improve: Ständige Verbesserungen über alle Aktivitäten der Value Chain

Engage: Verständnis der Anforderungen, Transparenz und Engagement

Design & Transition: Qualität, Kosten und «Time to Market» erfüllen

Obtain oder Build: Komponenten zur richtigen Zeit und am richtigen Ort

Deliver und Support: Services liefern und supporten

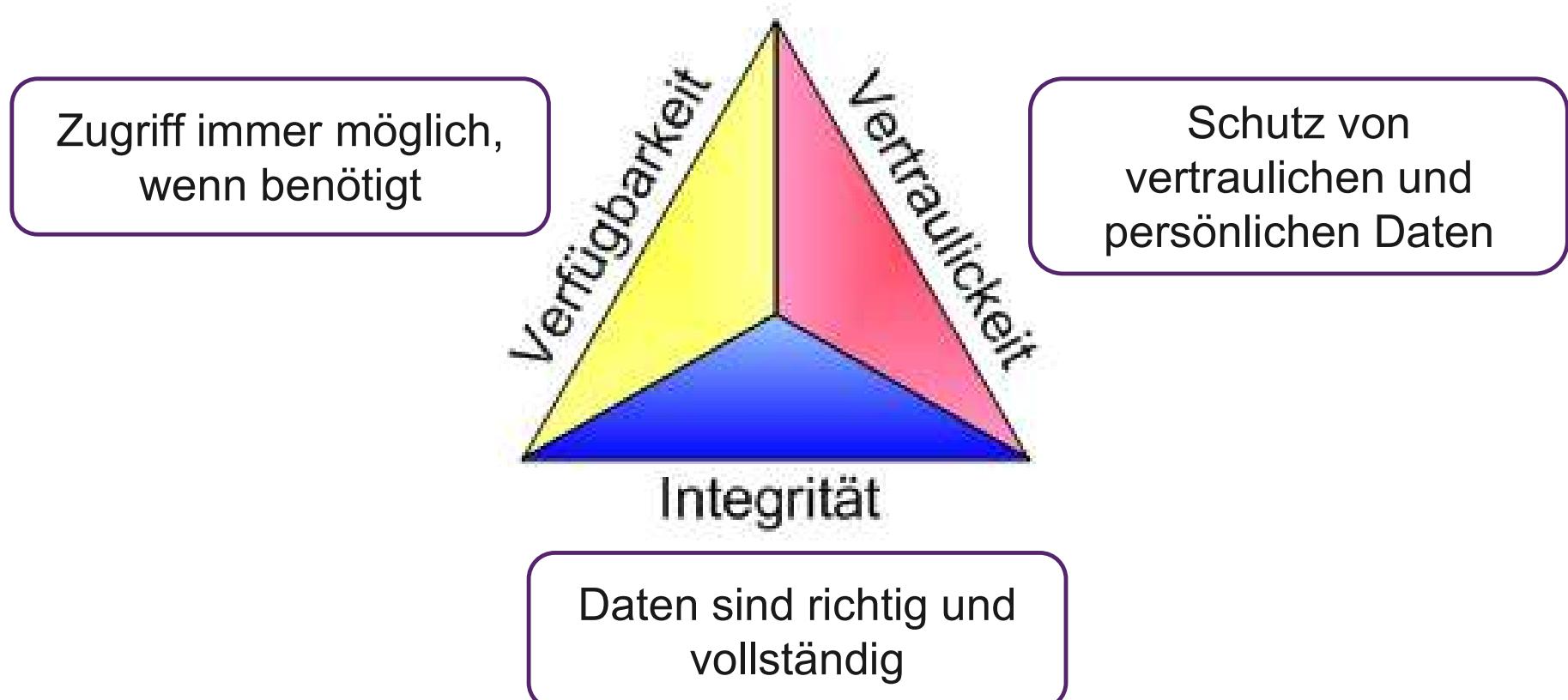
General Management Practices #1

Architecture Mgmt	liefert ein Verständnis über alle Elemente einer Organisation
Continual Improvement	gleicht die Praktiken und Services einer Organisation mit den stetig ändernden Business Anforderungen ab
Information Security Mgmt	schützt die Informationen der Organisation, die für das Business benötigt werden
Knowledge Mgmt	unterhält und verbessert die Effizienz, die Effektivität und die Anwendbarkeit von Information und Wissen über die ganze Organisation
Measurement und Reporting	unterstützt das Fällen von guten Entscheidungen und kontinuierliche Verbessern durch die Reduktion von Unsicherheiten
Organisational Change Mgmt	stellt eine sorgfältige und erfolgreiche Implementation von Veränderungen in einer Organisation sicher und sorgt für einen nachhaltigen Nutzen in dem es die menschlichen Aspekte einer Veränderung berücksichtigt
Portfolio Mgmt	stellt mit den vorhandenen Mittel sicher, dass die Organisation den richtigen Mix aus Programmen, Produkten und Services hat, um die Strategie zu erfüllen
Project Mgmt	stellt sicher, dass alle Projekte in der Organisation erfolgreich durchgeführt werden

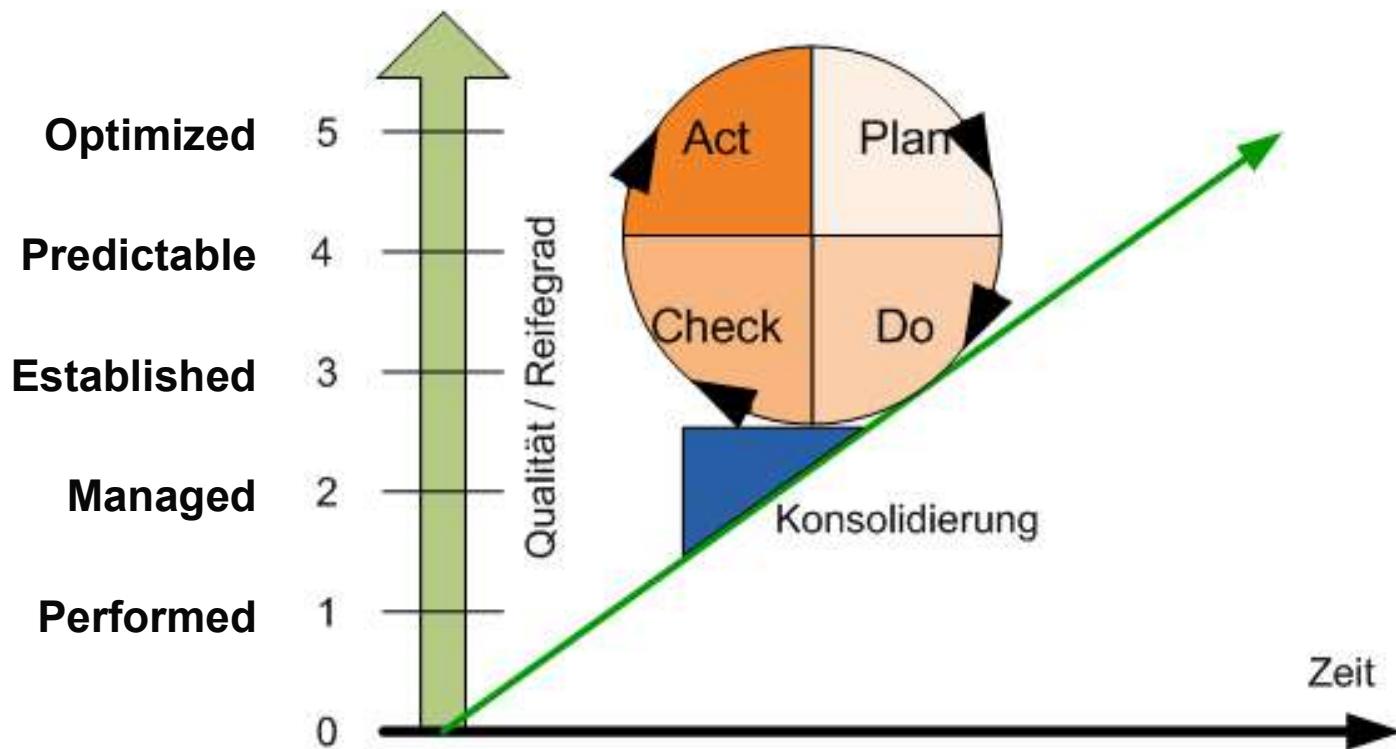
General Management Practices #2

Relationship Mgmt	etabliert und pflegt die Beziehung zwischen der Organisation und ihren Stakeholdern auf einem strategischen und taktischen Level
Risk Mgmt	stellt sicher, dass die Organisation die Risiken versteht und effektiv damit umgeht
Service Financial Mgmt	Budgetierung und Hochrechnung der Voll-Kosten für Produkte und Services sowie Accounting (Kostenzuweisung) und Charging (Verrechnung)
Strategy Mgmt	formuliert die Ziele einer Organisation und steuert Aktivitäten und Ressourcen um diese zu erreichen
Supplier Mgmt	stellt sicher, dass die Leistung der Supplier nahtlos auf die Qualität der Produkte und Services abgestimmt ist
Workforce and Talent Mgmt	stellt sicher, dass die Organisation die richtigen Leute mit dem richtigen Wissen und Können in den richtigen Rollen hat, um Businessziele zu unterstützen

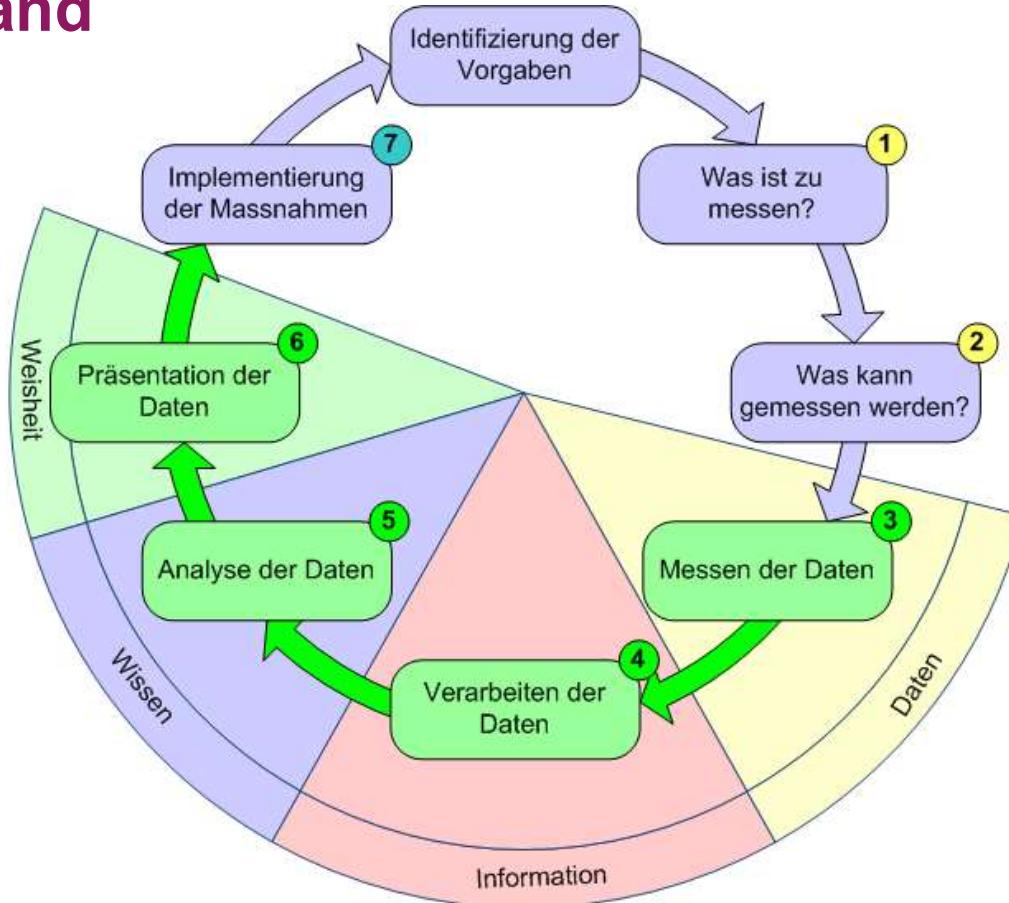
Information Security Management



Continual Improvement



Measurement and Reporting



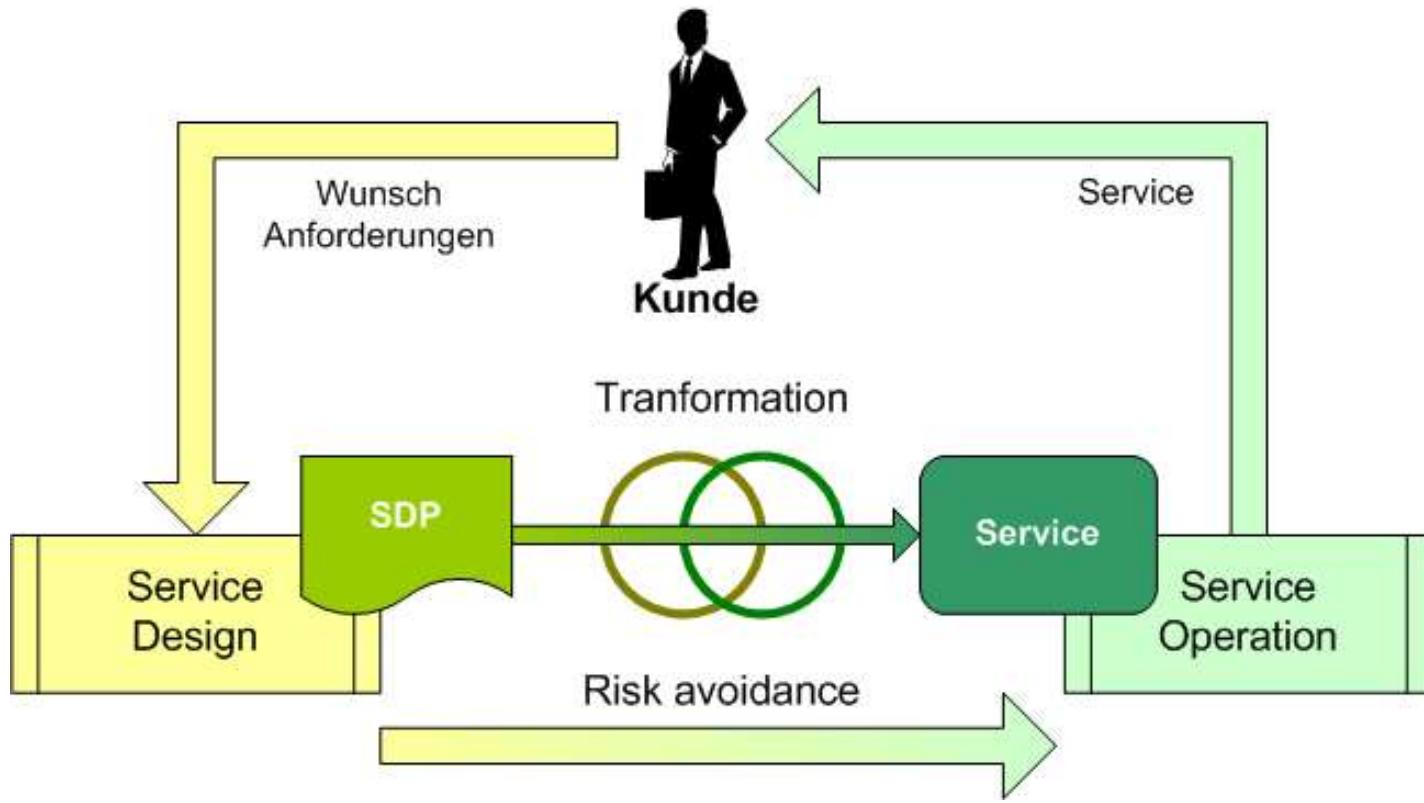
Service Management Practices #1

Availability Mgmt	stellt sicher, dass die Verfügbarkeit der Produkte und Services den Vereinbarungen entsprechen
Business Analysis	analysiert das Business oder Elemente davon, definiert die Anforderungen und schlägt Lösungen vor, diese zu erfüllen
Capacity and Performance Mgmt	stellt sicher, dass Services die vereinbarte und erwartete Performance sowie aktuelle wie zukünftige Anforderungen erfüllt
Change enablement	maximiert die Anzahl der erfolgreichen Produkt- und Service Changes durch sauberes Risk Management, autorisierte Changes und eine Change Planung
Incident Mgmt	reduziert die negativen Einflüsse einer Störung in dem es den normalen Betriebszustand so schnell als Möglich wiederherstellt
IT Asset Mgmt	plant und bewirtschaftet alle IT Assets über den gesamten Lifecycle
Monitoring and Event Mgmt	überwacht systematisch Services und Service Komponenten und zeichnet Statusänderungen als Events auf
Problem Mgmt	reduziert die Wahrscheinlichkeit von negativen Einflüssen von Störungen durch die Identifizierung von aktuellen und potenziellen Ursachen und verwaltet Workarounds und Known Errors
Release Mgmt	macht neue und geänderte Services und Funktionen für den Kunden verfügbar

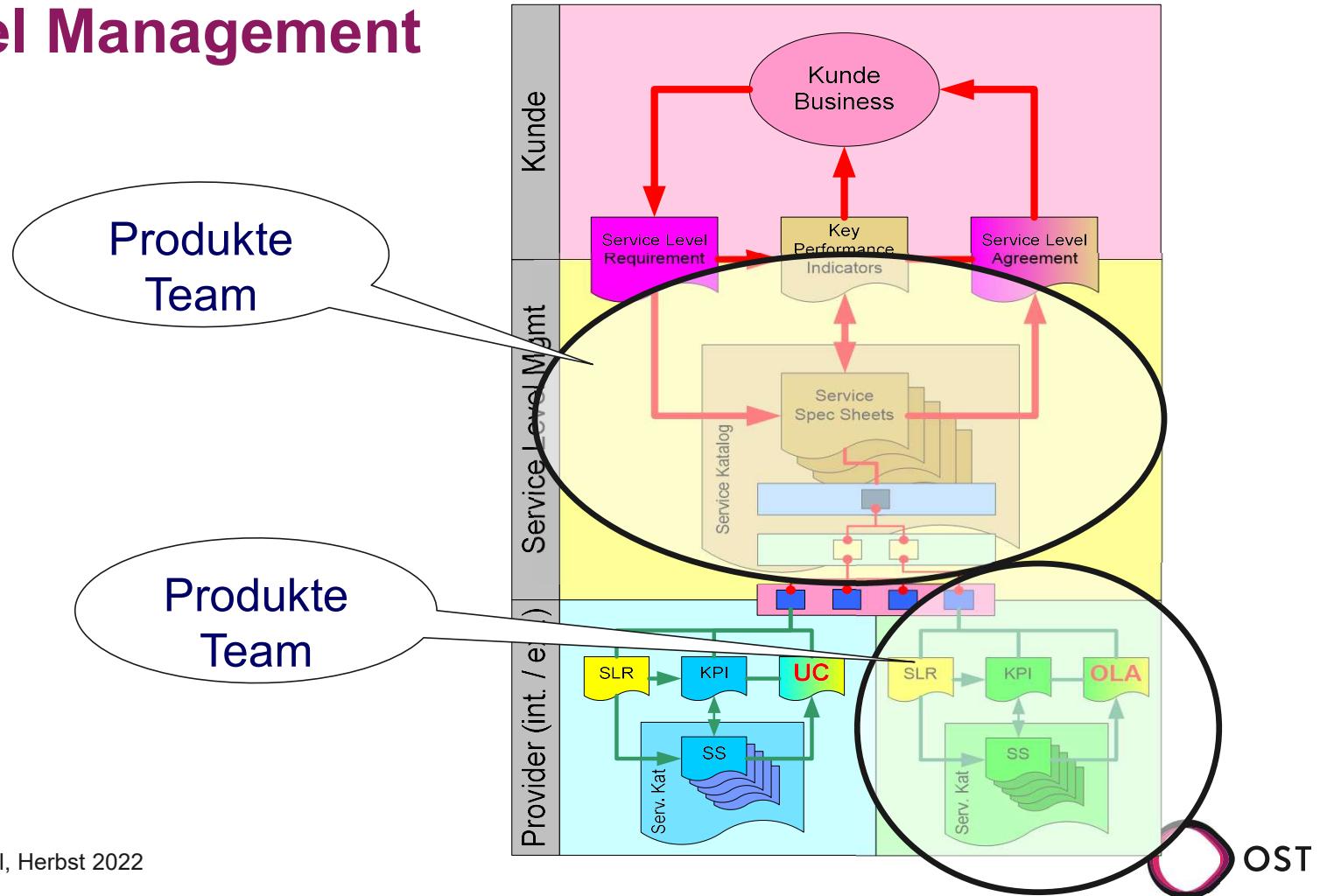
Service Management Practices #2

Service Catalog Mgmt	liefert eine einheitliche Sicht auf konsistente Informationen aller Services und stellt diese den richtigen Personen zur Verfügung
Service Configuration Mgmt	stellt zuverlässige Informationen über die Konfiguration von Services und deren Komponenten (CI's) sowie der Beziehungen zur Verfügung
Service Continuity Mgmt	stellt sicher, dass die Verfügbarkeit und die Performance eines Service in einem genügenden Mass während eines Desasters vorhanden ist
Service Design	definiert Produkte und Services, die «fit for purpose», «fit for use» und die Möglichkeiten der Organisation geliefert werden können
Service Desk	erfasst Störungsmeldungen und Service Requests
Service Level Mgmt	setzt klare business basierende Ziele für die Service Levels und stellt sicher, dass die Lieferung der Services sauber überprüft, überwacht und entsprechend diesen Zielen betrieben wird
Service Request Mgmt	unterstützt die vereinbarte Qualität eines Service durch die Abwicklung aller vorbestimmten und durch den User initiierten Service Requests in einer effektiven und userfreundlichen Art
Service Validation and Testing	stellt sicher, dass neue oder geänderte Produkte und Services definierte Anforderungen erfüllen

Service Design

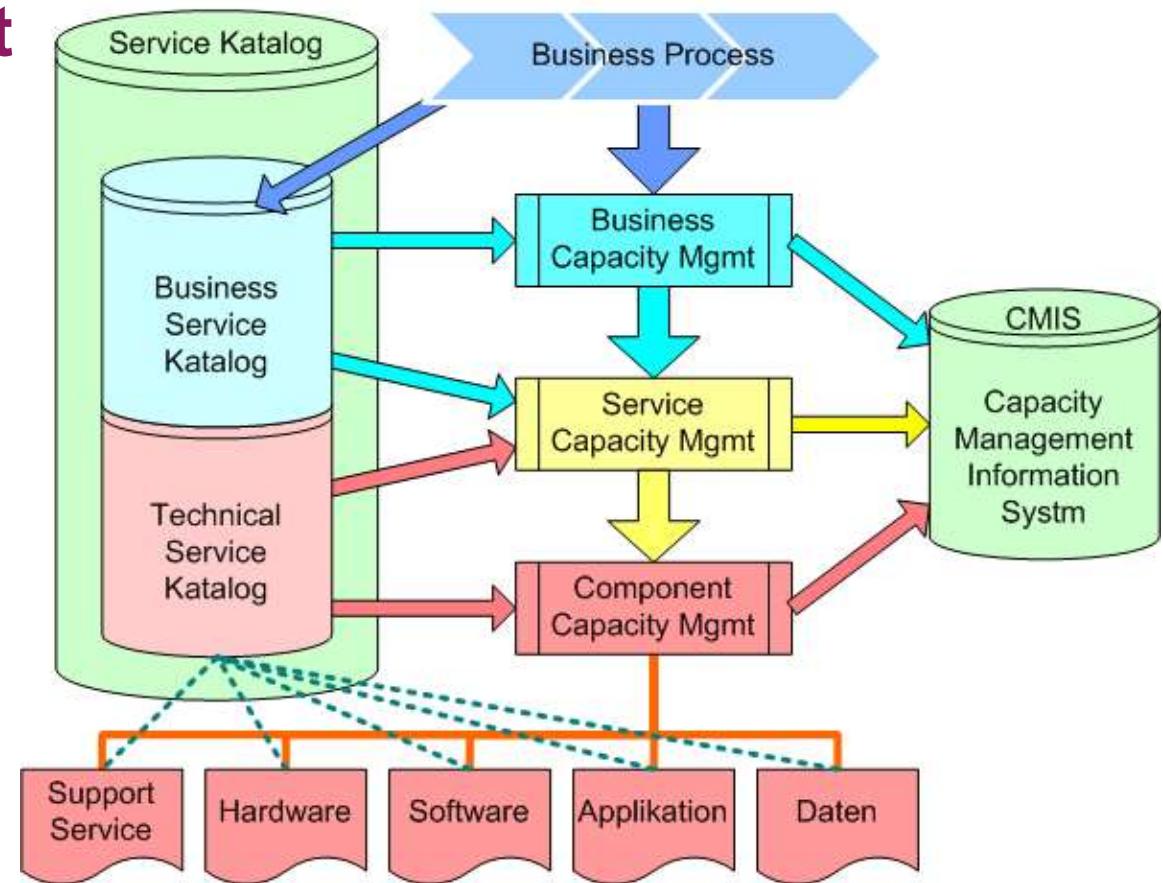


Service Level Management



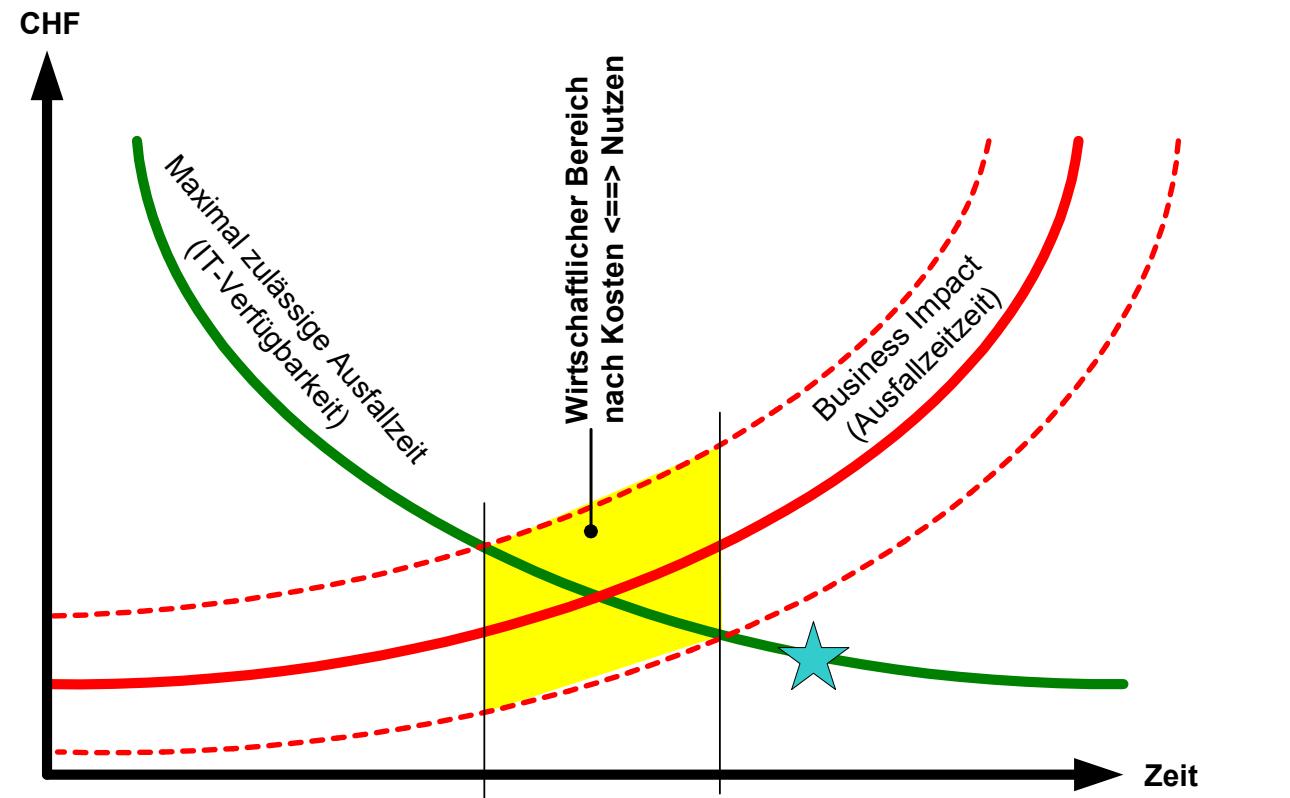
Capacity Management

Welchen Einfluss hat mein Projekt auf den aktuellen Betrieb?

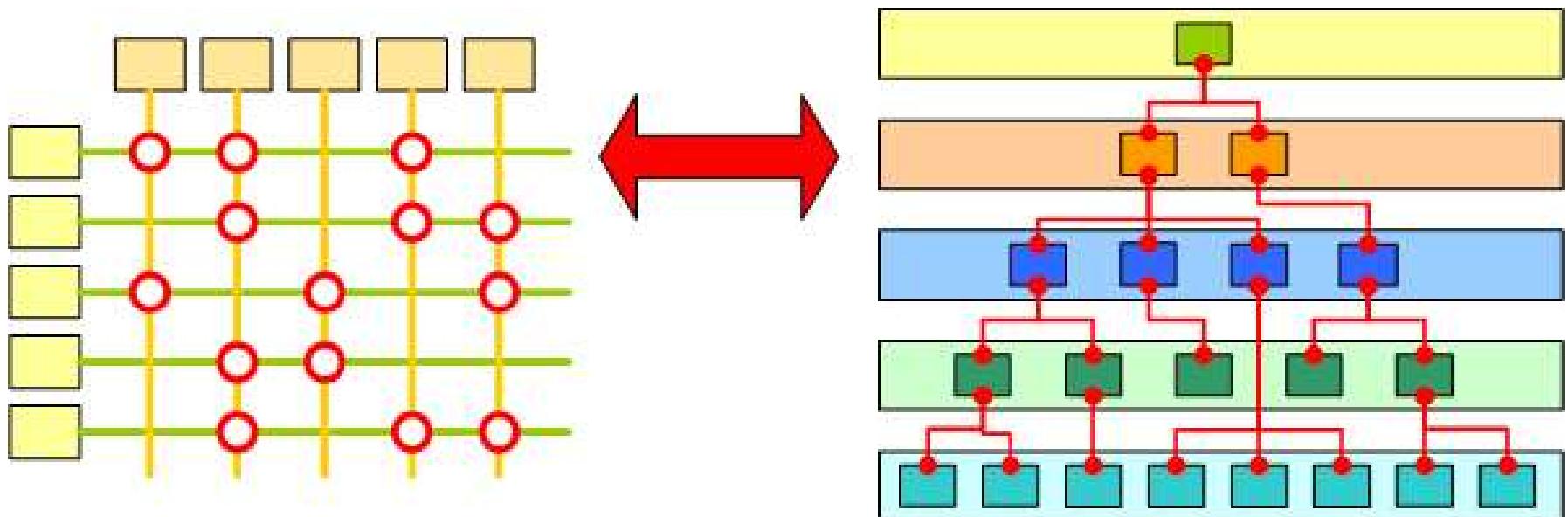


Availability Management

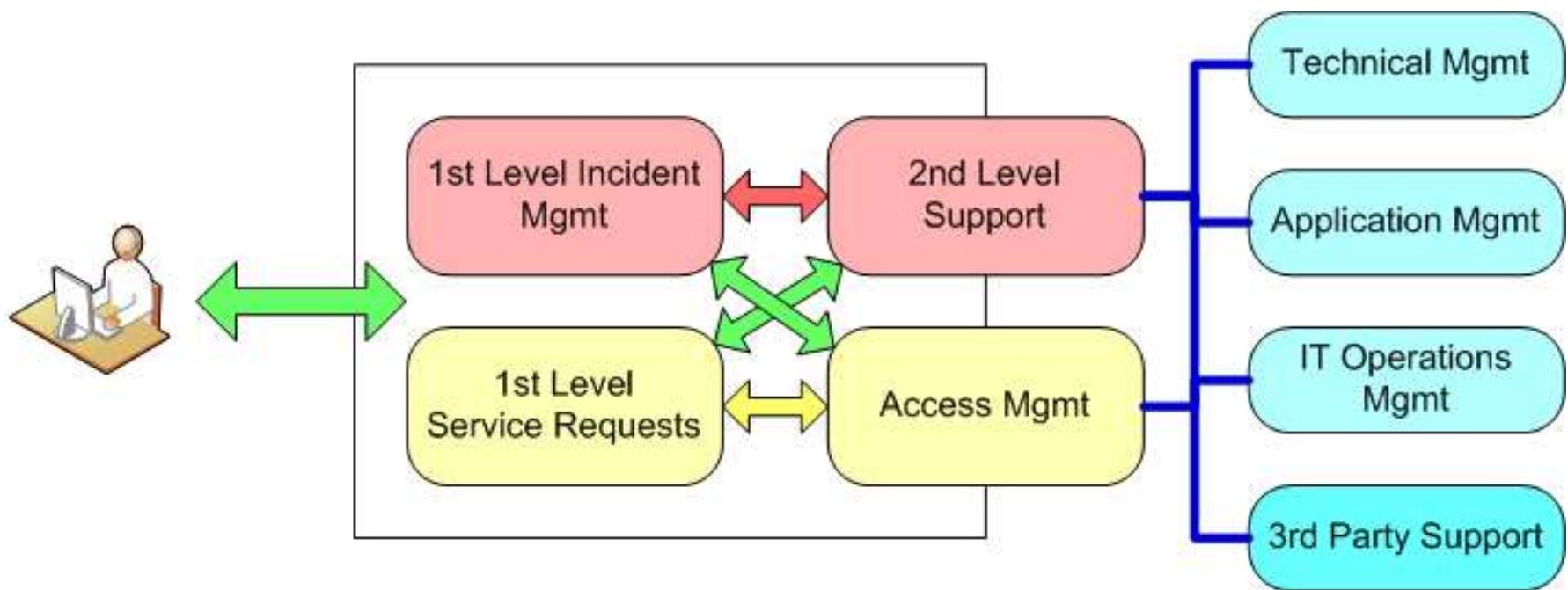
Wann habe ich das optimale Preis-Leistungs-Verhältnis erreicht?



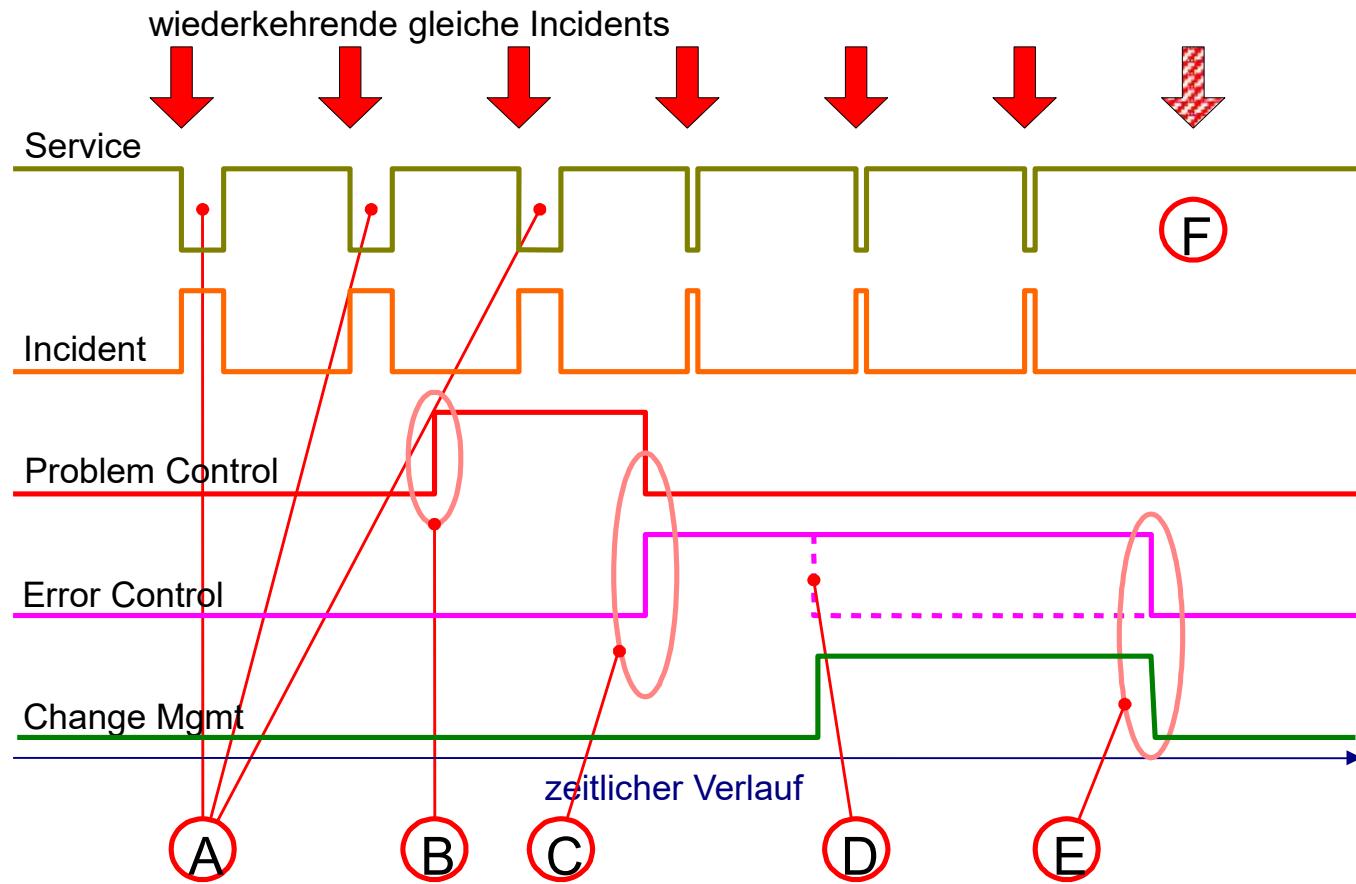
IT Asset und Service Configuration Management



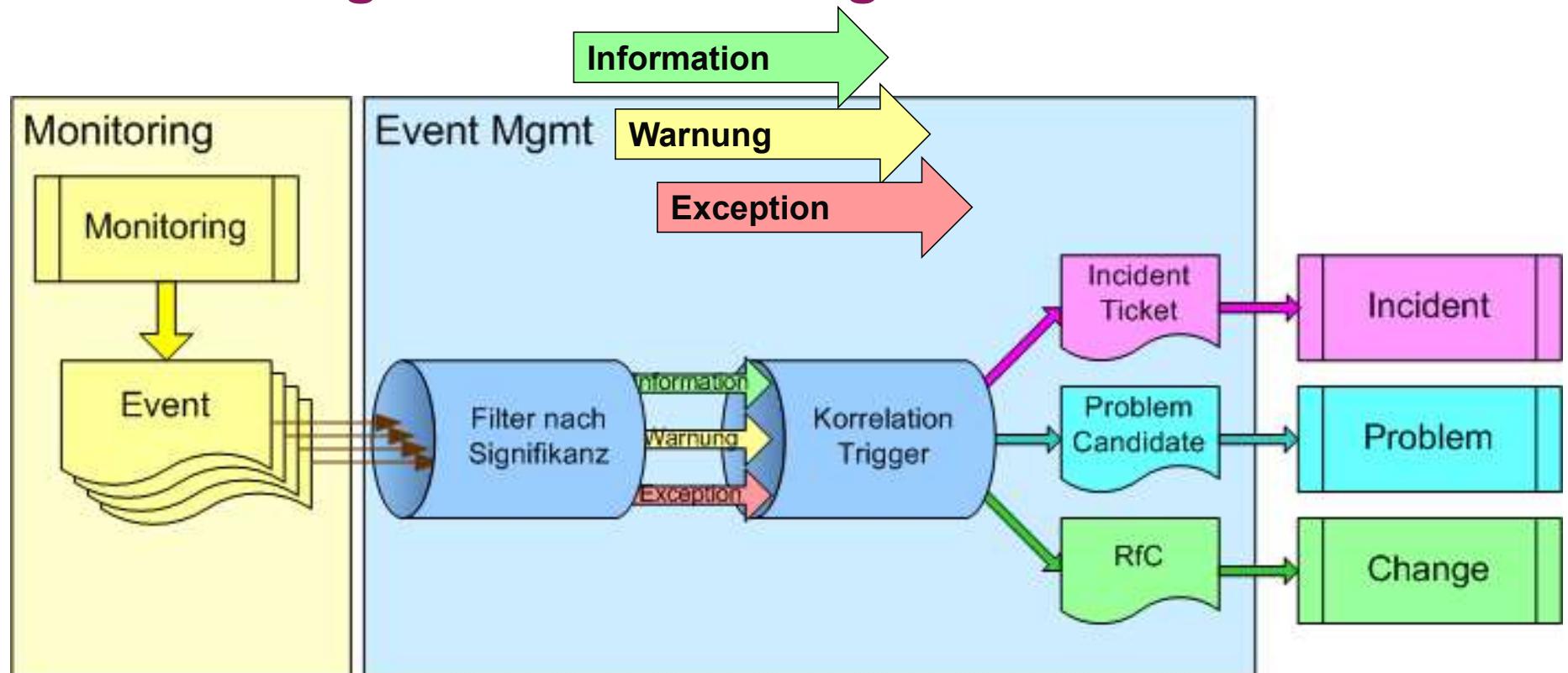
Service Desk und Incident Management



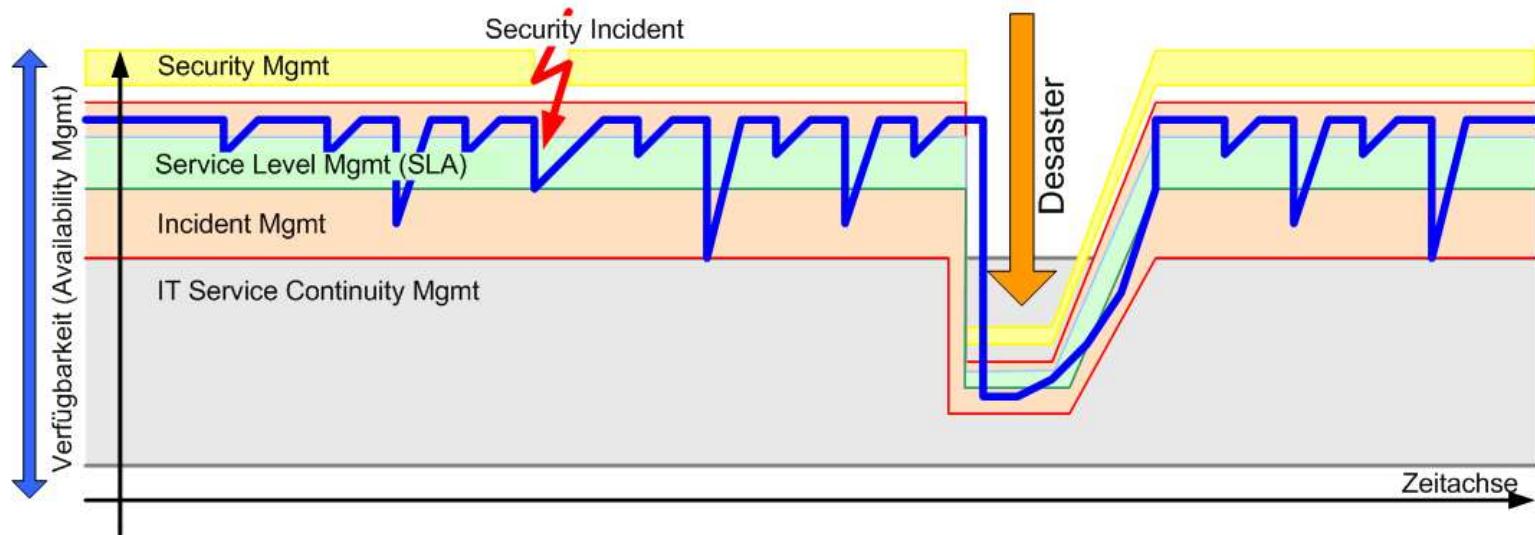
Incident – Problem - Change



Monitoring and Event Management



Service Continuity Management

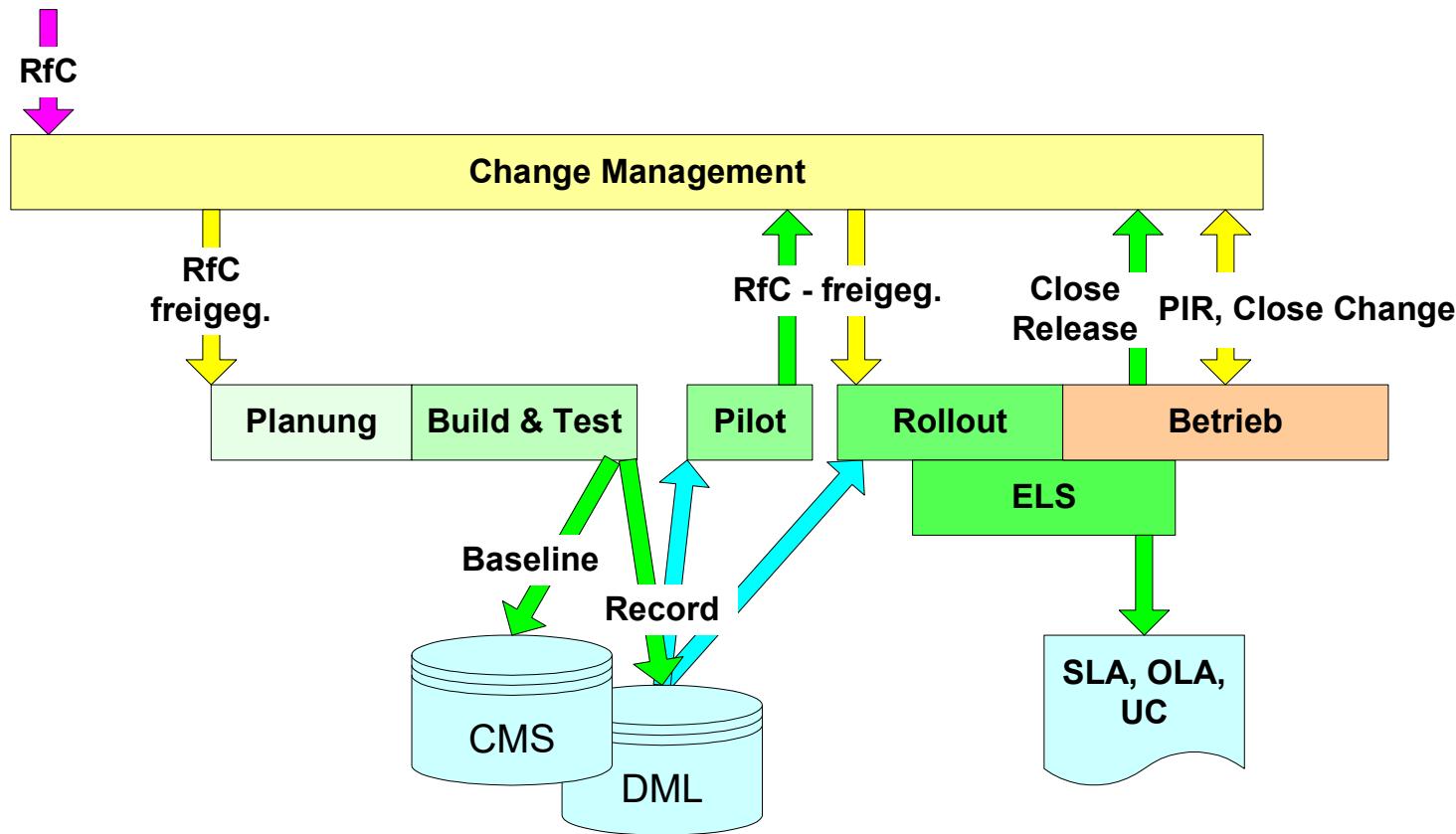


Wie wird der Service (Daten und Programme) nach einem totalen Ausfall eines RZ wieder hergestellt?

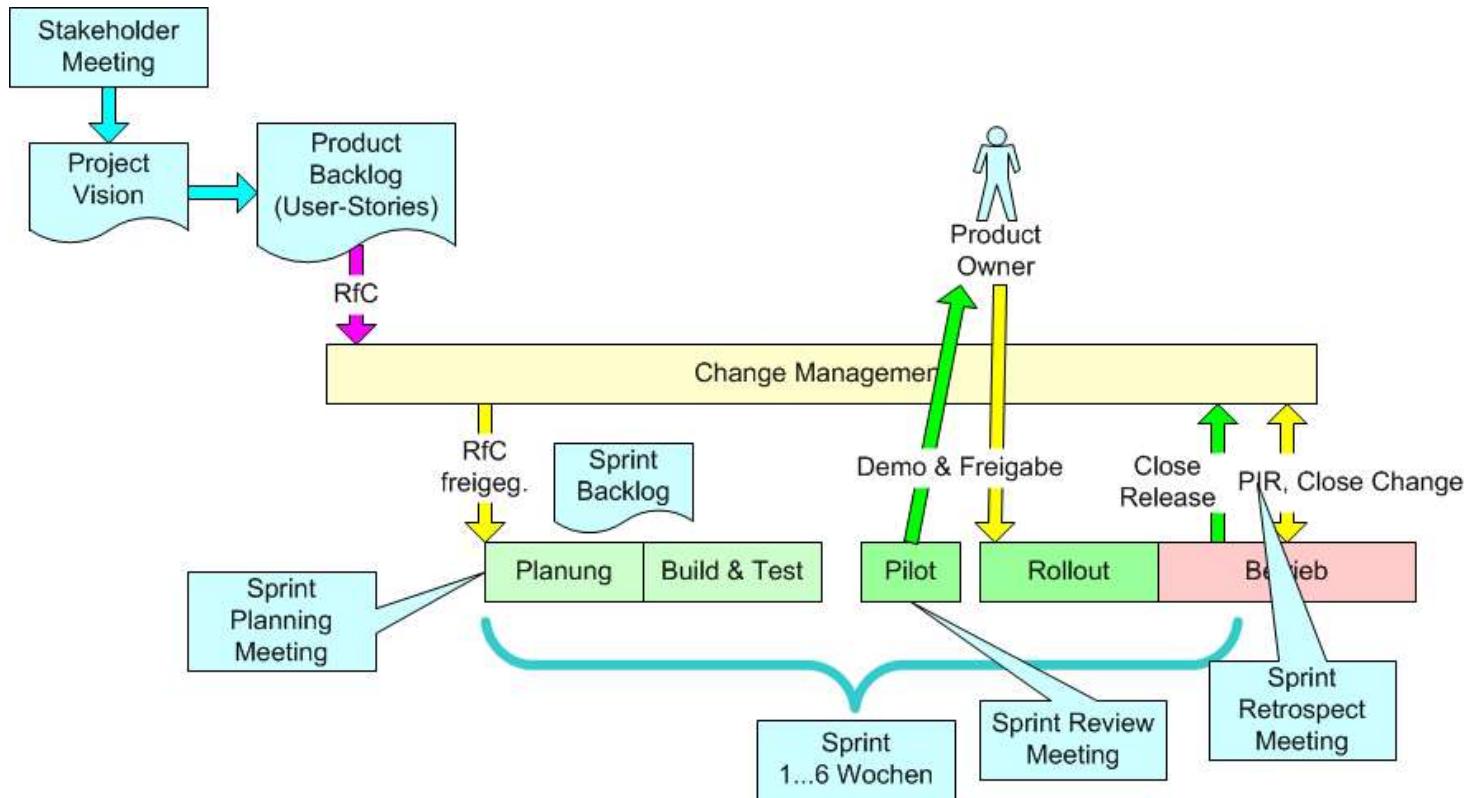
Technical Management Practices

Deployment Mgmt	schiebt neue oder geänderte Hardware, Software, Dokumentationen, Prozesse oder alle anderen Komponenten in die Produktionsumgebung
Infrastructure and Platform Mgmt	überwacht die Infrastruktur und Plattformen einer Organisation
Software Development and Management	stellt sicher, dass Applikationen die Anforderungen der internen und externen Stakeholder bezüglich Funktionalität, Zuverlässigkeit, Wartbarkeit, Konformität und Auditfähigkeit erfüllen

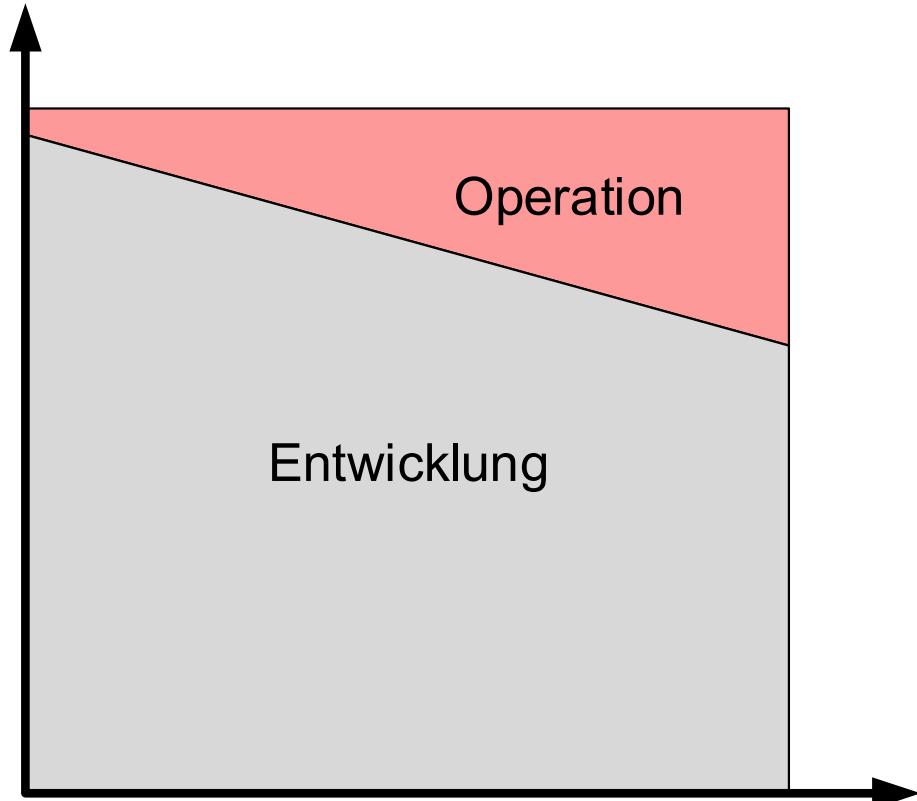
Change und Release Management (konventionell)



Change und Release Management (agil)



Infrastructure and Platform Management



DevOps Teams brauchen
bis zu 30% ihrer
Ressourcen für den Betrieb