Zusammenfassung

**Lernziele**

* Sie verstehen es Projekte erfolgreich zu starten, durchzuführen und abzuschliessen
* Sie kennen unterschiedliche Vorgehen von agil bis zum klassischen Wasserfall
* Sie beherrschen die wichtigsten Führungsinstrumente eines Projektleiters
* Sie besitzen grundlegende Kenntnisse für Management und Sicherung von Qualität in Projekten und Unternehmen
* Sie erlangen die weltweit anerkannte Zertifizierung «IPMA Level D» (freiwillig)

**Lerninhalte**Diese Vorlesung ist ein Muss für alle Ingenieurinnen und Ingenieure, welche sich das Rüstzeug für Projektmanagement holen möchten. Im Vordergrund steht die Methodik zur Abwicklung von zeitlich begrenzen Vorhaben. In den praxisorientierten Übungsstunden lernen Sie Beispiele und Techniken aus der Industrie kenne. Folgender Inhalt im Detail:

* Was versteht man unter einem Projekt
* Projekttypen und deren interne Organisation
* Projektkontrolle
* ITIL als Qualitätsinstrument
* Klassische Projektvorgehensmodelle
* Agile Projektvorgehensmodelle
* Finanzierung von Projekten
* Umgang mit Projektrisiken
* Qualitätsmanagement/-sicherung
* Kommunikation mit wichtigen Anspruchsgruppen
* Konfigurationsmanagement
* Methoden zur Lösungsfindung

Übersicht des Modules

[Projekt und Qualität 4](#_Toc502129929)

[Ziele des Kapitels 4](#_Toc502129930)

[Innovation ist Teamwork 4](#_Toc502129931)

[Was ist ein Projekt 4](#_Toc502129932)

[Typische Merkmale eines Projektes 4](#_Toc502129933)

[Das magische Dreieck 4](#_Toc502129934)

[Typische Projektarten in der IT 5](#_Toc502129935)

[Projektgrösse 5](#_Toc502129936)

[Make or Buy 6](#_Toc502129937)

[Was bedeutet Qualität 6](#_Toc502129938)

[Begriff «Qualität» 6](#_Toc502129939)

[Begriff «Qualitätsmanagement» 6](#_Toc502129940)

[Qualitätskreislauf nach Deming 6](#_Toc502129941)

[Beispiel eines Qualitätsplans in einem Projekt 7](#_Toc502129942)

[Projekte initialisieren, durchführen und abschliessen 8](#_Toc502129943)

[Ziele des Kapitels 8](#_Toc502129944)

[Unternehmensziele 8](#_Toc502129945)

[Projekttypen 8](#_Toc502129946)

[Wie werden Projekte gestartet, wie beendet 9](#_Toc502129947)

[Der Projektstart 9](#_Toc502129948)

[Vorgehen bei der Planung 10](#_Toc502129949)

[Projekteende/Projektabschluss 12](#_Toc502129950)

[Projektteam / Organisation 12](#_Toc502129951)

[Mögliche Organisationen 12](#_Toc502129952)

[AKV definieren 12](#_Toc502129953)

[Gruppendynamik positiv umsetzen 12](#_Toc502129954)

[Requirements Engineering 13](#_Toc502129955)

[Agil oder Wasserfall – Auswirkungen auf die Anforderungen 13](#_Toc502129956)

[Projektkontrolle 14](#_Toc502129957)

[Ziele des Kapitels 14](#_Toc502129958)

[Wer kontrolliert die Projekte 14](#_Toc502129959)

[Berichtswesen 14](#_Toc502129960)

[Welches Projektkontrollsystem nehme ich? 14](#_Toc502129961)

[Messwert und Vermessen 15](#_Toc502129962)

[Effektiver Verlauf in den Projekten 15](#_Toc502129963)

[Wie messe ich den genauen Fortschritt? 15](#_Toc502129964)

[ISO 9126 – Sicherstellung der Softwarequalität 15](#_Toc502129965)

[Das Ampel-Prinzip 16](#_Toc502129966)

[Projektcockpit 16](#_Toc502129967)

[Standards für die Projektkontrolle 16](#_Toc502129968)

[Earned-Value-Analyse 17](#_Toc502129969)

[Wie berechne ich den Earned Value (Fertigstellungsgrad)? 17](#_Toc502129970)

[Beispiel einer EVA an einem Studentenprojekt 17](#_Toc502129971)

[Meilenstein-Trendanalyse 18](#_Toc502129972)

[Klassisches Projektmanagement 18](#_Toc502129973)

[Ziele des Kapitels 18](#_Toc502129974)

[Vorgehen 18](#_Toc502129975)

[Vorgehensmodelle 18](#_Toc502129976)

[Das Vorgehen 18](#_Toc502129977)

[Klassische Vorgehensmodelle 19](#_Toc502129978)

[Phasen am Beispiel RUP 19](#_Toc502129979)

[CYNEFIN – Wann ein Phasenmodell 19](#_Toc502129980)

[HERMES, Vorgehensmodell des Bundes 20](#_Toc502129981)

[Phasen 21](#_Toc502129982)

[Rollenverteilung bei HERMES 21](#_Toc502129983)

[Aufgaben der Projektsteuerung 21](#_Toc502129984)

[Aufgaben der Projektführung 21](#_Toc502129985)

[Alle Projektteilnehmer nach HERMES (standardisiert) 21](#_Toc502129986)

[Pro Projektteilnehmer ein definiertes AK(F) 22](#_Toc502129987)

[Jede Aufgabe ist bis ins Detail beschrieben 22](#_Toc502129988)

[Vor- und Nachteile von HERMES 22](#_Toc502129989)

[Prototypenmodell 22](#_Toc502129990)

[Merkmale 22](#_Toc502129991)

[Spiralmodell 23](#_Toc502129992)

[Diskussion 23](#_Toc502129993)

[Finanzierung, Kosten und Beschaffung 24](#_Toc502129994)

[Ziele des Kapitels 24](#_Toc502129995)

[Kosten 24](#_Toc502129996)

[Nutzen 25](#_Toc502129997)

[Nutzenelemente von Projekten 25](#_Toc502129998)

[ROI 25](#_Toc502129999)

[ROI Beispiel – Einwicklung einer neuen App 26](#_Toc502130000)

[ROI Beispiel – WLAN Access Points ersetzen 26](#_Toc502130001)

[Diskontierung – Was ist das? 27](#_Toc502130002)

[ROI Beispiel – Online Shop einer Handelskette 27](#_Toc502130003)

[Business Cases 27](#_Toc502130004)

[Finanzierung und Liquidität 28](#_Toc502130005)

[Projektliquidität 28](#_Toc502130006)

[Beschaffung und Verträge 29](#_Toc502130007)

[Was ist ein Vertrag? 29](#_Toc502130008)

[Unterscheidung der Verträge für Dienstleistungen 29](#_Toc502130009)

[IT Service Management nach ITIL v3 30](#_Toc502130010)

[Risikomanagement 31](#_Toc502130011)

[Agiles Projektmanagement I 32](#_Toc502130012)

[Agiles Projektmanagement II 33](#_Toc502130013)

[Qualitätsmanagement 34](#_Toc502130014)

[Kommunikation und Dokumentation 35](#_Toc502130015)

[Lösungsfindung und Teamarbeit 36](#_Toc502130016)

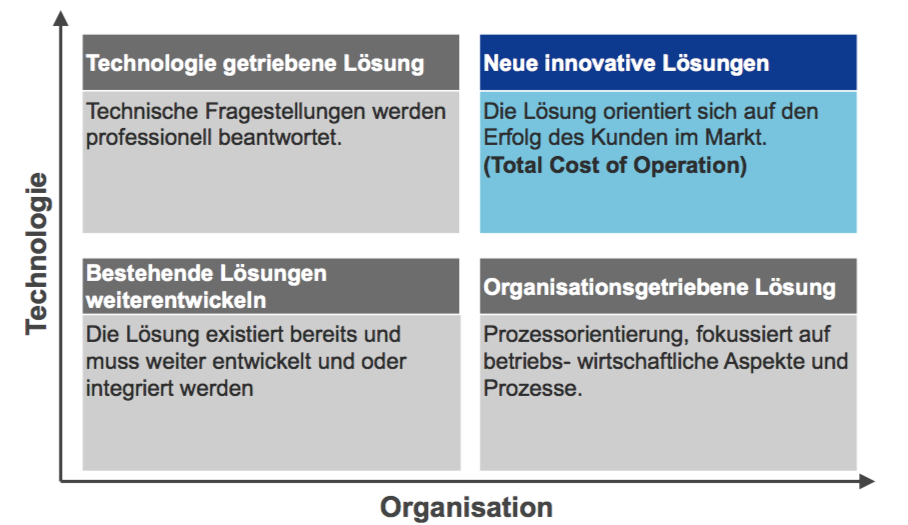
[Projekte in der Praxis 37](#_Toc502130017)

# Projekt und Qualität

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Was ist ein Projekt * Was bedeutet Qualität | * erarbeiten sich selbständig Beispiele von Innovation * erkennen ein Projekt aufgrund von Merkmalen * verstehen was Qualitätsmanagement ist |

## Innovation ist Teamwork

Die Schweiz ist gemäss dem WEF das Wettbewerb fähigste und gemäss GII das innovativste Land der Welt. Viel Innovation entsteht aus Projekten. Aber was genau ist Innovation? Lösungen, welche neue durch Organisation und neue Technologien getrieben sind.

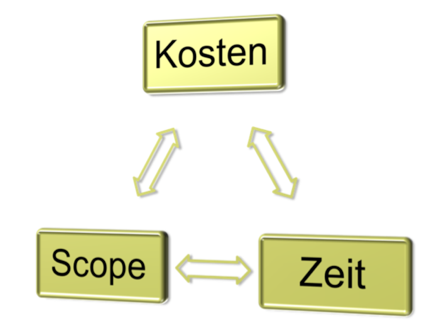
**Beispiele** dafür sind: Facetune App, Zipline Dronen, ...

## Was ist ein Projekt

### Typische Merkmale eines Projektes

* Zeitlich begrenzt
* Beschränkte Ressourcen (Zeit, Geld, etc.)
* Neue Problemstellung (Neue Technologie, Neue organisatorische Aspekte, Beide in Kombination = Innovation)
* Ungewissheit aufgrund komplexer Herausforderungen
* Benötigt Personen aus verschiedenen Organisationseinheiten

### Das magische Dreieck

Das magische Dreieck zeigt die Abhängigkeiten in einem Projekt. Kosten beeinflussen Zeit und Scope, Scope wiederrum Kosten und Zeit und die Zeit Kosten und Scope. Folgend werden ein paar Beispiele von Konflikten genannt.

* **Zeit ⬄ Kosten**
  + Ein fester Endtermin für ein Entwicklungsprojekt kann unter Umständen nicht mehr gehalten werden (Möglichkeit: Mehr Entwickler einsetzen um den Termin zu halten, was aber mehr Kosten verursacht)
* **Kosten ⬄ Scope**
  + Das Budget ist fast aufgebraucht und eigentlich besteht noch der Wunsch das Webfrontend mit Logo und CO des Kunden auszustatten. (Möglichkeit: Webfrontend im Standard stehenlassen um Geld zu sparen)
* **Zeit ⬄ Scope**
  + Die Applikation ist fast abgeschlossen und der Projektleiter würde aber gerne noch automatische Performance Tests über zwei Woche sowie einige Tuning Arbeiten ausführen (Möglichkeit: Je nach Interesse Performance Tests durchführen und dafür Verspätung in Kauf nehmen oder darauf verzichten)

### Typische Projektarten in der IT

Projekt Neuentwicklung, Projekt Weiterentwicklung, Integration einer gekauften Software, Aufbau einer Infrastruktur, Forschungsprojekt, Organisationsprojekt.

### Projektgrösse

Die Grösse eines Projekts lässt sich über verschiedene Faktoren messen. Kosten, Zeit oder Arbeitsstunden. Auch möglich ist der akademische Ansatz von Capers Jones. Die Berechnung (<http://groups.engin.umd.umich.edu/CIS/course.des/cis375/projects/fp99/main.html>) beruht dort auf Functions Points.

|  |  |
| --- | --- |
| Diese Abbildung zeigt einige Beispiele von Projektgrössen marktüblicher Software (angegeben in Functions Points). | Je grösser das Projekt, je geringer die Wahrscheinlichkeit eines guten Resultats. |

### Make or Buy

In Projekten muss man irgendwann immer einmal entscheiden, ob man es selbst macht oder kauft. Bei einer Stanardlösung müssen meist die Prozesse angepasst werden, während bei der individuellen Lösung der Aufwand an IT höher liegt. Die optimale Lösung wäre eine ausgeglichene Variate in der Mitte.

#### Resultat

Wie die Entscheidung fällt ist je nach Person abhängig für einen Hersteller von Velo-Computer würde ich sagen, dass diese Firma alle folgenden Sachen kaufen/outsourcen soll: Webseite, Zeiterfassungssoftware, ERP, CRM, Schliesssystem Gebäude.

## Was bedeutet Qualität

### Begriff «Qualität»

Die Herkunft statt aus dem Lateinischen und zwar Qualitas. Der Begriff Qualität beinhaltet nicht zwingend eine Wertung. Im Sprachgebrauch ergänzt man daher oft auch Wörter wie gute oder schlechte Qualität.

### Begriff «Qualitätsmanagement»

Qualitätsmanagement ist ein stetiger Prozess, um nachhaltig den Output zu steigern. QM ist je nach Brache gar nicht oder auch sehr stark vorgeheben. Im Textil-Bereich ist dies sehr gering, während Qm in der Luftfahrt sehr wichtig ist. QM selbst möchte verbessern, um den dafür investierten Aufwand mehr als zurückzugewinnen.

Viele Aktoren haben Qualitätsrevelente Interessen. Die Umwelt, Gesetze, der Kunde (Funktion, Ergonomie, Betrieb, Lebendauer, Kontinuität, Service, Support und Preis) sowie der Produzent (Mitarbeiter, Produktionsmittel, Material, Organisation, Entwicklung, Zulieferanten, Energie, Kosten und Termine).

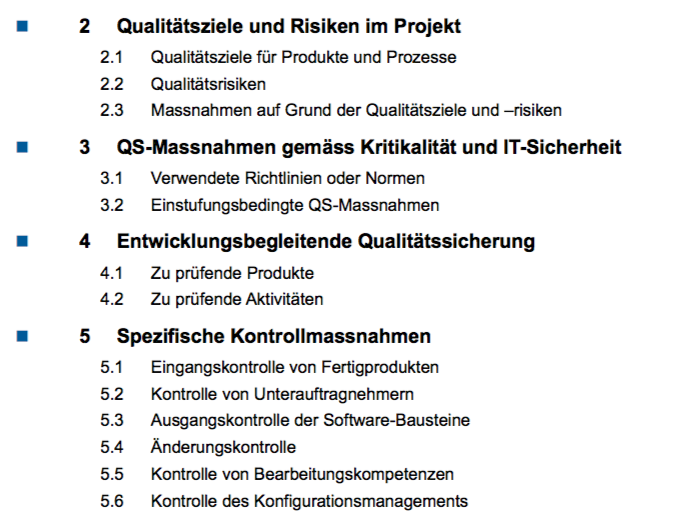
### Qualitätskreislauf nach Deming

Ein Qualitätskreislauf definiert nach Deming lautet wie folgt.

**Plan** Output bestimmen und planen  
**Do** Notwenige Arbeiten ausführen  
**Check** Output messen  
**Act** Massnahmen zur Korrektur bestimmen

|  |  |
| --- | --- |
| **Plan – Think Smart**   * Output mit SMART Ansatz definieren   + **S**pezifisch   + **M**essbar   + **A**kzeptiert   + **R**ealistisch   + **T**erminiert * Strategie der Umsetzung   + Maximal   + Minimal   + Optimiert * Risiken beachten und Reserve einbauen | Do – Do it right   * Mach es OTOBOS   + On time   + On budget   + On scope * Wenn der Output nicht klar ist, nach einen Prototyp   + Eine Zusatzrunde mit Check – Act – (Re)Plan |
| **Check – Please mind the gap**   * Überlege wie du checkst * Finde Abweichungen früh | **Act – Let’s go**   * Fixe die Abweichung * Mach dein Produkt reif und bring es an die Front * Definiere Massnahmen für den nächsten Zyklus |

### Beispiel eines Qualitätsplans in einem Projekt



# Projekte initialisieren, durchführen und abschliessen

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Unternehmensziele * Projekttypen * Wie werden Projekte gestartet, wie beendet * Projektteam / Organisation * Requirements Engineering | * kennen die Stolpersteinen wenn Projekte nicht zu Unternehmen passen * vermögen nach der Vorlesung ein Projekt sauber zu starten und zu beenden * sehen die Möglichkeiten, wie man ein Projekt organisiert * erhalten einen Fresh-Up zu Requirements Engineering * diskutieren wann Anforderungen noch ändern dürfen |

## Unternehmensziele

Projektziele müssen möglichst mit Unternehmenszielen vereinbar sein, ansonsten riskieren Sie ein Scheitern wegen zu viel Wiederstand. Unternehmensziele sind langfristig (grösser 5 Jahre), mittelfristig (3-5 Jahre) und kurzfristig (bis 3 Jahre) ausgelegt.

In dieser Tabelle ein paar Beispiele, wenn es nicht passt.

|  |  |
| --- | --- |
| **Unternehmensziel** | **Projektziel** |
| Der Verwaltungsrat hat beschlossen Produkte ausschliesslich über indirekte Kanäle zu verkaufen. | Das Produktmanagement möchte mit Ihnen ein Projekt lancieren, um Velo-Computer direkt übers Internet zu verkaufen. |
| In der Zeitung lesen Sie: «Handelsgeschäft steht zum Verkauf, mehrere ausländische Unternehmen interessiert» | Sie bewerben sich als Projektleiter für den Aufbau einer neuen Handelsplattform. |
| Innert 5 Jahren sollen alle Standorte ausserhalb der Deutschschweiz geschlossen werden. | Ihr Chef möchte Sie als Projektleiter damit beauftragen eine mehrsprachige Lernplattform zu entwickeln. |
| Für «bahnfremde» Kommunikationsdienste gilt eine Outsourcing-Strategie an etablierte Telefongesellschaften. | Sie versuchen als Jungunternehmer einen Bahnbetreiber ein Projekt für WLAN im Zug zu verkaufen. |

## Projekttypen

Es gibt keine einheitliche Typendefinition. In der Literatur von Thomas Grossner findet man man folgende vier Quadrate, welche eine gute Definiton sind.

|  |  |
| --- | --- |
| **Linienprojekt** | **Stablinienorganisation** |
| **Matrixorganisation** | **Reine Projektorganisation** |

|  |  |
| --- | --- |
| **Linienprojekt**  Alle wichtigen Akteure sind in der gleichen Einheit, ihrer Stammorganisation. Der Chef der Einheit ist gleichzeitig Projektleiter. Vereinzelt werden Personen aus anderen Einheiten beigezogen. Für kleine Projekte ohne grossen Anspruch an interdisziplinärem Mitwirken oder für Start-Up Unternehmen bestens geeignet. | **Stablinienorganisation**    Die Akteure sind im Unternehmen in verschieden Stammesorganisationen verteilt. Ein PL agiert aus einer Stabsstelle heraus. Der Projektleiter kann einmal eingesetzt und anschliessend wieder verschwinde (z.B. externe). Flexible Lösung für mittelgrosse Projekte, da der PL nur die wirklich notwendige Ressource benötigt. |
| **Matrixorganisation**    Die Akteure sind auf verschiedene Einheiten verteilt. Der PL ist einem Pool von Projektleitern zugeordnet und führt vielleicht mehrere Projekte parallel durch. Mit den Stammesorganisationen sind die Einsatzpläne abzustimmen. Der PL ist «normaler Mitarbeiter», die Unterstellung muss also geregelt werden. Für grosse Unternehmen geeignet, welche Bedarf für einen eigenen Pool an PL haben. | **Reine Projektorganisation**    Alle wichtigen Akteure werden zu 100 % für ein Projekt in der gleichen Einheit zusammengezogen oder sogar angestellt. Der Chef der Einheit ist gleichzeitig Projektleiter. Vereinzelt werden Personen aus anderen Einheiten beigezogen. Für grosse Projekte mit langfristigem Fokus. Mitarbeiter müssen nach dem Projekt entlassen oder reorganisiert werden. |

## Wie werden Projekte gestartet, wie beendet

### Der Projektstart



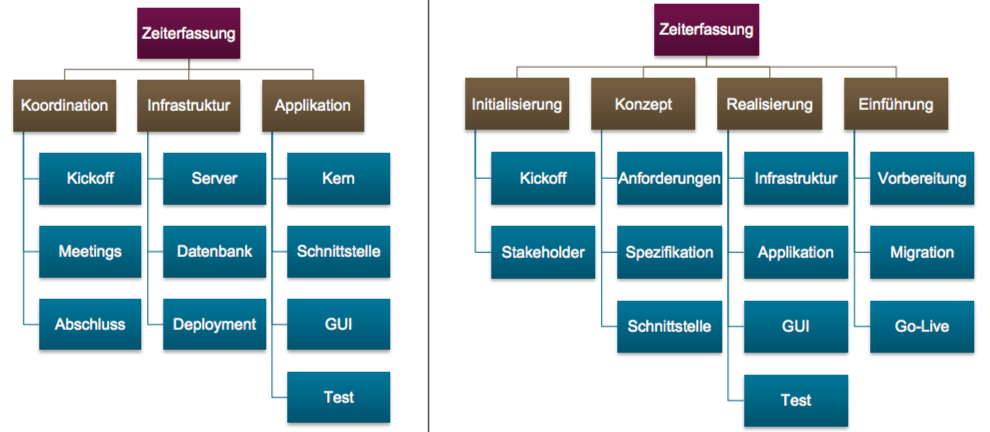
#### Häufige Fehler beim Projektantrag

* Die Argumentation des Zweckes ist nicht schlüssig, es fehlt ein roter Faden und die Gründe lassen sich nicht nachvollziehen
* Der Projektantrag widerspiegelt die Wünsche des Auftragsgebers nicht oder nur teilweise
* Der Antrag basiert auf einer teilweisen falschen Ausgangslage
* Die definierten Ziele sind nicht messbar
* Der Nutzen des Projekts wird viel zu optimistisch definiert
* Bedeutende Risiken werden nicht erkannt

### Vorgehen bei der Planung

#### Projektstrukturplan

* Top-Down oder Bottom-Up Zerlegung des Projekts
* Mehrstufige Strukturierung von Arbeitspaketen



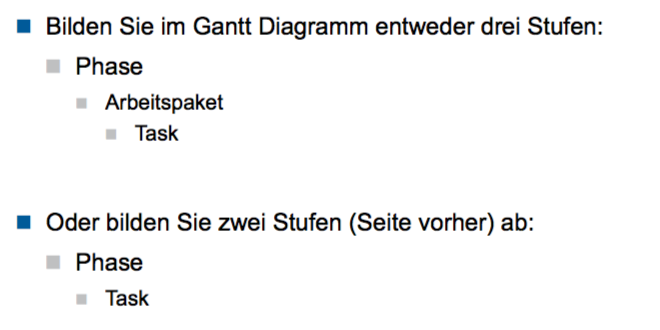
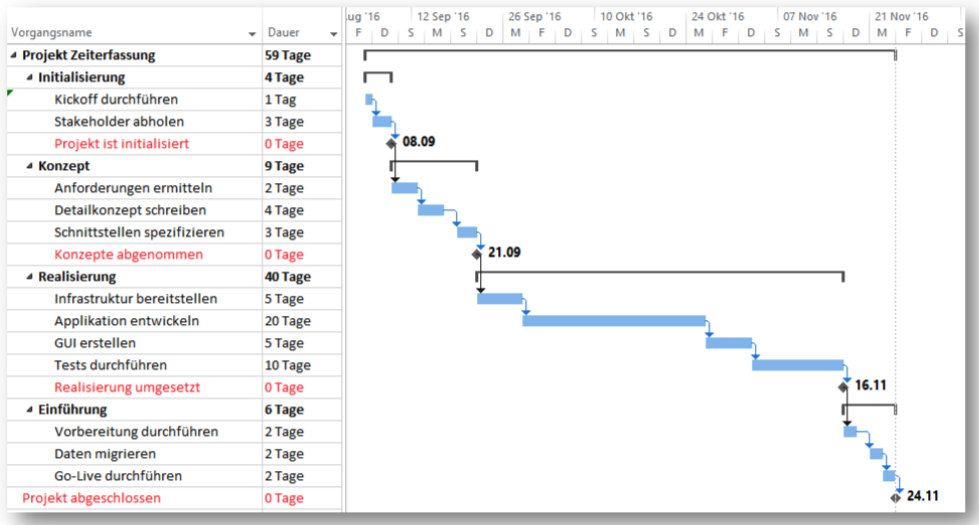
Strukturpläne könne Sie nach Thema, Projektablauf, Funktion, etc. strukturieren.

**Vorgehen beim Strukturplan**

* Erstellen Sie den Strukturplan nie ganz alleine, holen Sie Input und andere Meinungen ein
* Identifizieren Sie die Arbeitspakete systematisch Top-Down oder Bottom-Up
  + **Top-Down**: Projekt zuerst grob strukturieren und dann immer detaillierter werden
  + **Bottom-Up:** Sammeln aller realen Arbeitspaketen und nachträglich die Gliederung definieren
* Der Strukturplan ist das Fundament, wenn Sie zu viel oder zu wenig detailliert sind, können Sie das Projekt nicht mehr vernünftig steuern.
* Trennen Sie nur Dinge voneinander ab, welche wenig gegenseitige Abhängigkeiten haben

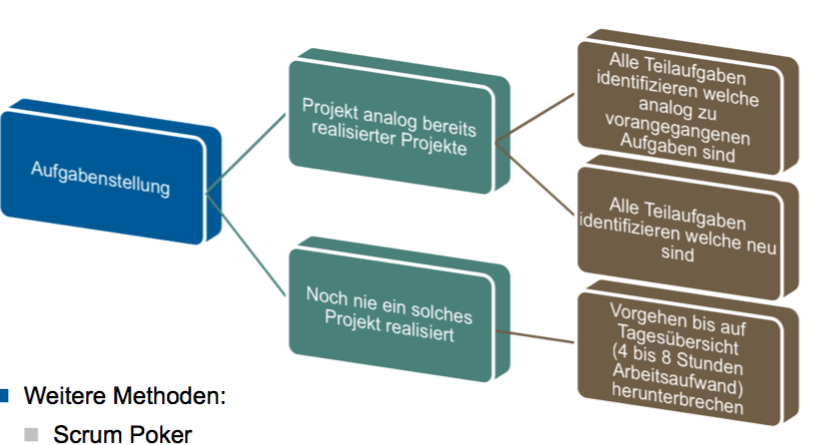
#### Gantt Diagramm

* Aufwand und Zeitschätzung für jedes Arbeitspaket
* Abhängigkeiten ermitteln und in Phasen aufteilen



**Zu beachten beim Gantt-Diagramm**

* Beim Gantt-Diagramm scheiden sich Anfänger und Profis, investieren Sie etwas Zeit um diesen korrekt und übersichtlich zu gestalten.
* Projekte die mehrheitlich agil sind, benötigen diesen Plan nicht zwingend
* Strukturierung in Phasen, Arbeitspakete und Tasks unbedingt beachten
* Phasen und Arbeitspakete benennen Sie mit einem Nomen
* Verwenden Sie für Tasks eine aktive Formulierung «Nomen + Verb» mit welcher Sie genau sagen, was zu tun ist (erstellen, spezifizieren, begleiten, etc.) und für Meilensteine «Nomen + Partizip II»
* Pro Phase definieren Sie minestens einen Meilenstein, bevorzugt immer am Ende einer Phase um den Übergang (Gate) sauber zu steuern.

**Analogien nutzen für die Zeitschaufschätzung  
**

**Weitere Methoden:** Scrum Poker, Gauss-Annäherung (1x Worst Case) + (4x Normal Case) + (1x Best Case) durch 6, White Elephant

#### Ressourcenplanung

* Zuweisung der personellen und materiellen Ressourcen
* Kostenbetrachtung für die eingesetzten Ressourcen

Tragen Sie die notwendigen Ressourcen für die Erfüllung wie Puzzle-Teile zusammen. Durch die Zuweisung dieser Ressourcen an die einzelenen Arbeitspakete erhält man einen geeigneten Überblick. Hinter den Ressourcen sind Kosten zu hinterlegen. Reserven von etwa 10% hinterlegen Sie entweder auf Ressourcen oder im Gantt Plan, nicht aber an beiden Orten.

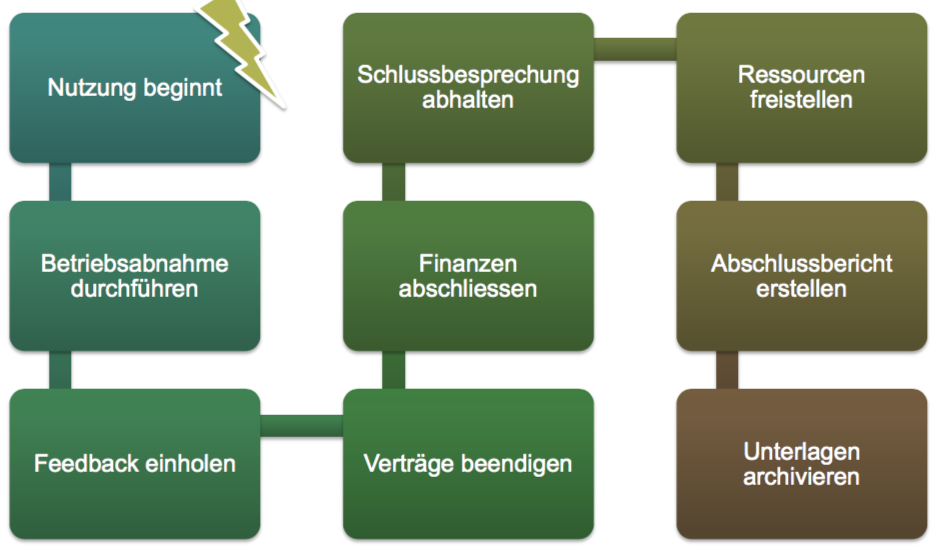
#### Arbeitspakete

* Input und Output pro Arbeitspaket festlegen
* Arbeitspakete detailliert beschreiben

Arbeitspakete werden mit folgenden Parametern definiert. Name und eindeutige Nummer/Code, Verantwortliche Person, Ziel, Inout, Beschreibung, Ressourcen, Zeitplan und Output (Lieferergebnisse).

Die Beschreibung eines Arbeitspakets weicht nicht wesentlich von einem Projektauftrag ab, ein Arbeitspaket ist also ein kleines Projekt im Projekt.

### Projekteende/Projektabschluss



## Projektteam / Organisation

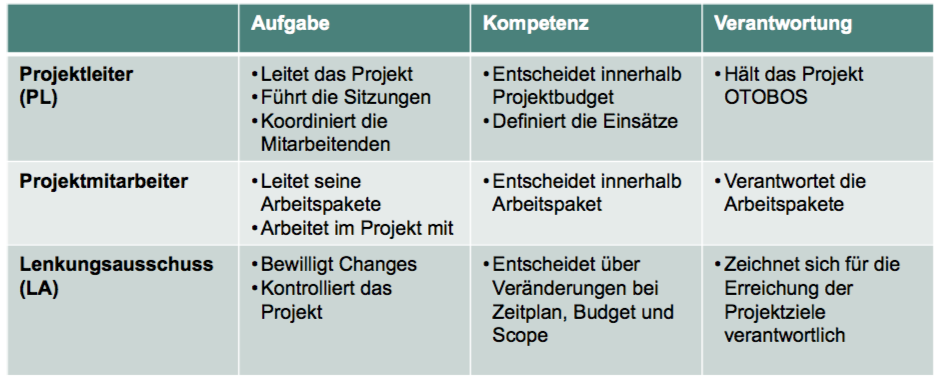
Ab zwei Personen ist mit Koordinationsaufwand zu rechnen. Nehmen Sie sich bei der Zusammenstellung der Teams Zeit und legen Sie die Organisation fest, zeichnen Sie das Organigramm. Handeln Sie mit den Stammorganisationen aus, wer wie viel im Projekt beiträgt.

### Mögliche Organisationen

|  |  |  |
| --- | --- | --- |
|  |  |  |

### AKV definieren

Definieren Sie ein AKV pro Rolle im Projekt.



### Gruppendynamik positiv umsetzen

Sprechregeln vereinbaren, zuhören nicht abschweifen; Ich-Botschaften anstelle von Man-Formeln; Ansichten sachlich diskutieren, Kritikkultur fördern; Zusammengehörigkeit fördern; Konflikte früh angehen und im Keim ersticken; Pünktlich und zuverlässig; Soziale Events planen; Motivieren; Jeder ist ein Vorbild.

## Requirements Engineering

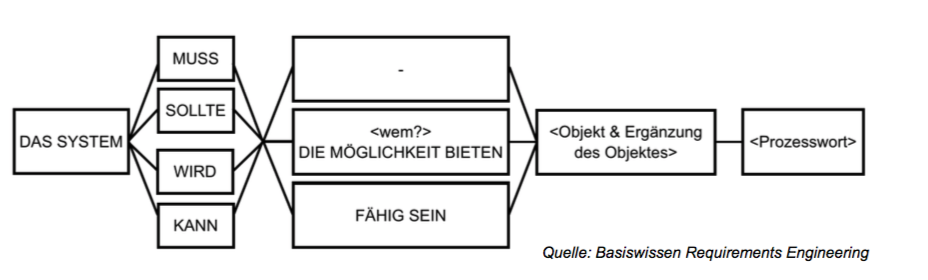
Das Erheben von Anforderungen wurde gemäss dne vorliegenden Unterlagen bereits im Modul SE1 grundlegend behandelt.

**Ein kleines Fresh-Up**

* Anforderungen sind der Grundstein für ein IT-Projekt
* Im agilen Vorgehen «User Story genannt
* Vom Kunden erwarten wir funktionale Anforderungen, die beschreiben «**Was** wird gewünscht»
* Meistens kann der Kunde nicht strukturieren, wir müssen dies tun:
  + Wie genau lautet die Anforderung?
  + Abhängigkeiten zu anderen Anforderungen?
  + Wie wichtig ist die Anforderung?
* Bilder zeigen mehr als tausend Worte, modellieren Sie die Anforderungen und verwenden Sie dazu eine normierte Sprache wie BPMN oder UML.

|  |  |
| --- | --- |
| **Ermittlungstechniken für Anforderungen** | **Beispiel Anforderungsliste für eine Fahrschul-Theorie App** |

**Achtung vor Modalverben**  
In der Theorie werden oft Modalverben verwendet, z.B. «Das System muss die Fehler des Benutzers protokollieren.»



Dies bringt den Nachteil, dass bei einer Änderung der Gewichtung diese Verben wieder ausgetauscht werden müssen, einfacher ist m. E. daher: «Das System protokolliert die Fehler des Benutzers».

### Agil oder Wasserfall – Auswirkungen auf die Anforderungen

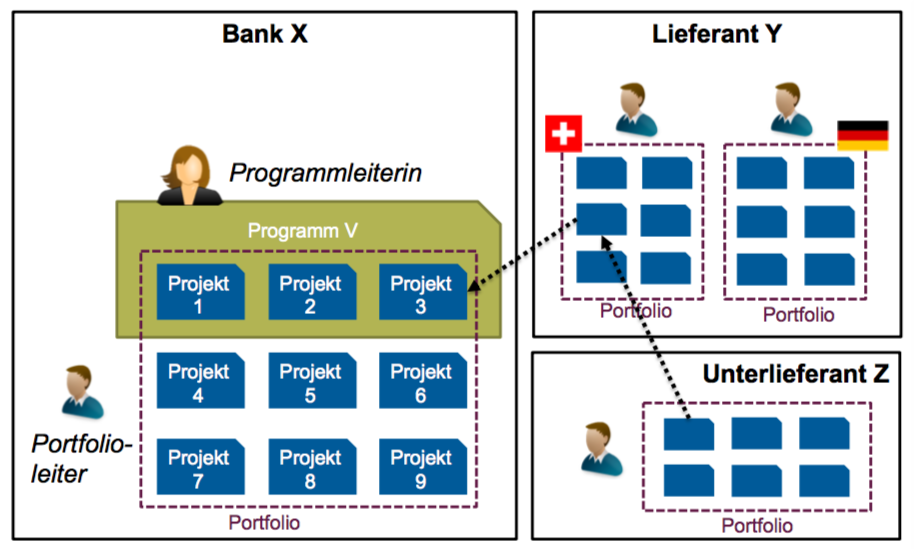


# Projektkontrolle

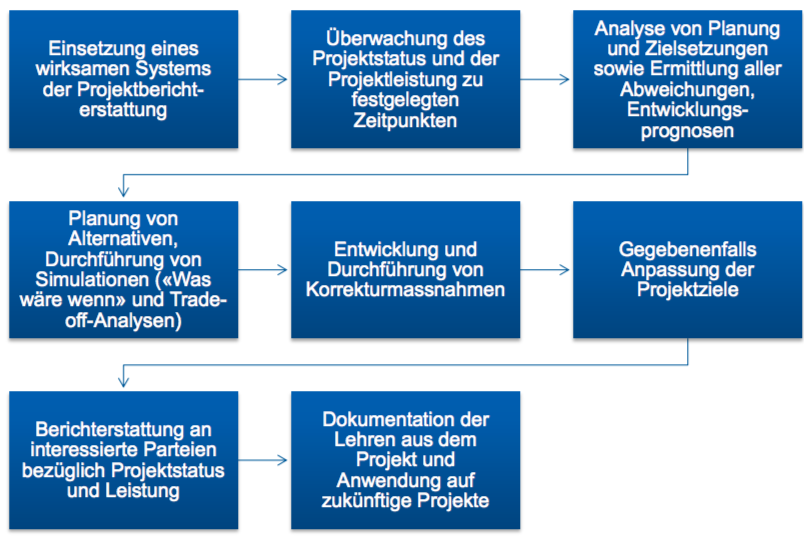
## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Wer kontrolliert die Projekte * Berichtswesen * Projekt Cockpit * Standards für die Projektkontrolle * Earned-Value-Analyse * Meilenstein-Trendanalyse | * kennen die Stolpersteinen wenn Projekte nicht zu Unternehmen passen * verstehen, dass Berichte nur ein Teil einer Kontrolle sind * kennen Standard zur Projektkontrolle * wenden zwei Analysen an |

## Wer kontrolliert die Projekte

Sie selber als Projektleiter steuern grundsätzlich selber innerhalb Budget, Zeitplan und Scope. Der Auftraggeber oder der Kunde kontrolliert Sie. Weiter verfügen Projekte über einen Lenkungssausschuss, der wichtige Entscheide fällt. Je nach Projektvorgenen gibt es weitere regelmässige Kontrollen durch das Finanzwesen und Audits durch den Qualitätsbeauftragten. Arbeiten mehrere Projekte einem Ziel zu, ordnet man diese einem Programm (Multi-Projekt) unter. Der Leiter des Programms überprüft dann die ihm unterstellten Projekte zusätzlich. Unternehmen mit mehreren Projekten verfügen über einen Portfolio Manager. Diese kontrollieren die Projekte in ihrem Portfolio. Das Portfolio Management ist eine stetige Aufgabe und zeitlich unbefristet.

## Berichtswesen

Projekte «Insieme» zeigen das ein Berichtwesen nötig. Ist. Ein Aufbau einer Kontrolle nach NCB sieht wie folgt aus.

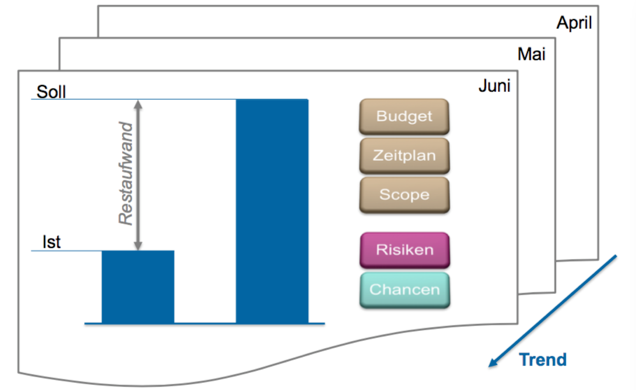
### Welches Projektkontrollsystem nehme ich?

Nehmen Sie grundsätzlich immer jenes, welches bereits im Unternehmen vorhanden ist. Falls keines vorhanden ist, nehmen Sie ein anderes oder bauen Sie selber ein Berichtswesen unter Beachtung folgender Punkte auf :

* Status von Budget, Zeitplan und Scope
* Qualitative Beschreibung des Fortschritts oder Restaufwandschätzung
* Risiken und Chancen im Projekt
* Issues
* Weitere Analyse-Charts
* Möglichkeit für Kommentare

Achtung! Ein Berichtswesen ist lediglich eine formale Meldung und noch eine vollständige Kontrolle. Kontrolle heisst auch Steuern.

### Messwert und Vermessen

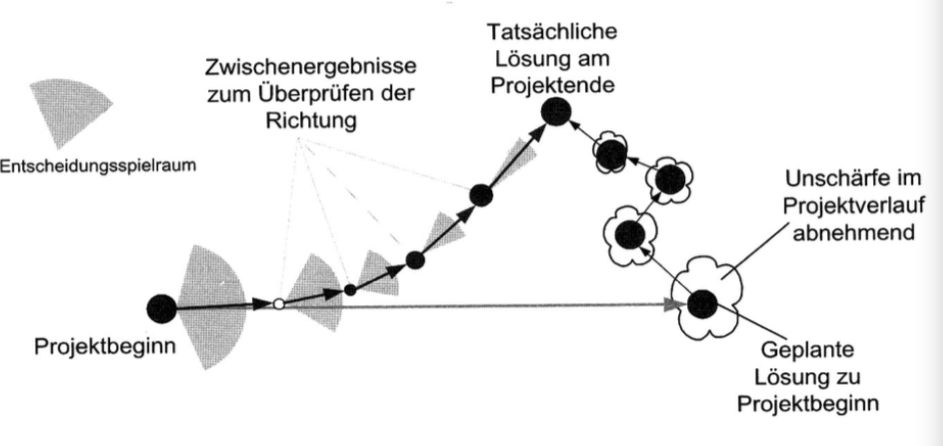
Aufgabe des Projektleiters ist es geeignete Mess- und Überprüfungsverfahren zu Beginn des Projekts einzuführen. Aufgabe des Controllers ist es zu überprüfen, ob das gemacht wurde und auch sinnvoll ist. Je nach Standardisierung von Projekten ist das einfacher oder schwieriger. Der Controller sollte rechtzeitig mit einbezogen werden.

Bei Abweichungen zum Plan muss gehandelt werden. Folgende Massnahmen zum Beispiel

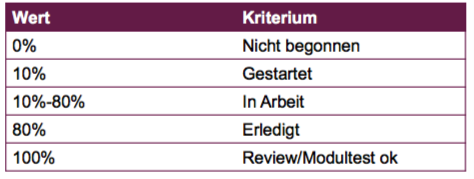
* Änderung in der Planung, Serielle in parallele Tätigkeiten ändern
* Überzeit anordnen
* Coaching oder zusätzliche Unterstützung anfordern

Bringen die Massnahmen keine Verbesserung, muss ein Change-Request gestellt werden.

### Effektiver Verlauf in den Projekten



### Wie messe ich den genauen Fortschritt?

Faktisch kaum möglich, Unschärfe vorprogrammiert. Am besten Produkt betrachten und Fertigkeitsgrad ermitteln. Fragen Sie einen Entwickler lieber, wie viel Zeit er noch benötigt und nicht wie weit er prozentual liegt. Legen sie die Metriken für die Kommunikation mit dem Auftraggebener fest, z.B. 80 Prozent bedeutet fertig aber noch nicht getestet (Definition of Done). Nun ein Beispiel einer Definition of Done in einem Projekt.

### ISO 9126 – Sicherstellung der Softwarequalität

|  |  |
| --- | --- |
|  |  |

### Das Ampel-Prinzip

Das Ampel-Prinzip hilft, Fakten sehr transparent und klar zu kommunizieren. Jedes Unternehmen hat eigene Definitonen von den Farben, bei IPG gilt.

* **ROT** – Abweichung grösser als 5 % 🡺 Eskalation
* **GELB** – Abweichung 0 – 5 % 🡺 Beobachtung
* **GRÜN** – Alles läuft nach Plan

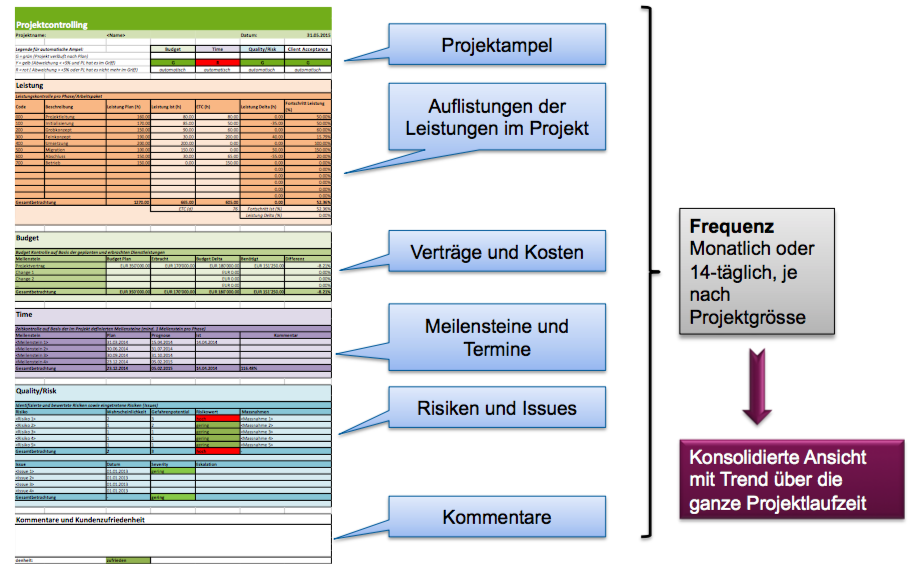
Die drei Stufen zwingen den Projektleiter klar Stellung zu nehmen. Portfoliomanager können sich schnell einen Überblick verschaffen.

#### Ampeln aggregieren

Innerhalb eines Projektes gilt generell die Aggregation nach dem Maximum Prinzip, Ausnahmen bestätigen die Regel. Wenn also Budget GELD, Zeitplan GRÜN und Scope GRÜN dann ist das Gesamtprojekt GELD.

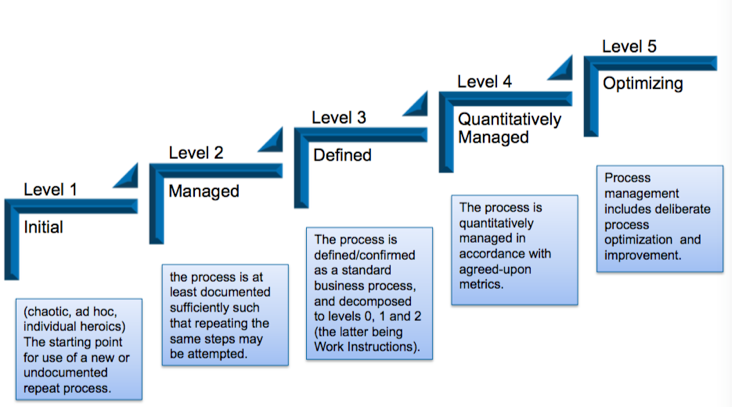
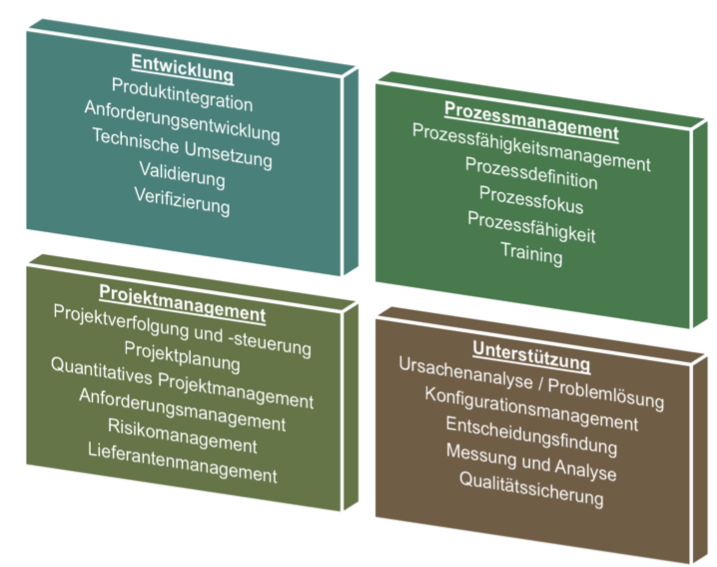
## Projektcockpit

Bedeutet eigentlich Hahnengrube, ist aber mit einem Flugzeugcockpit zu vergleichen. Ein Auszug eines Projektcockpits (Controlling) könnte wie folgt aussehen.



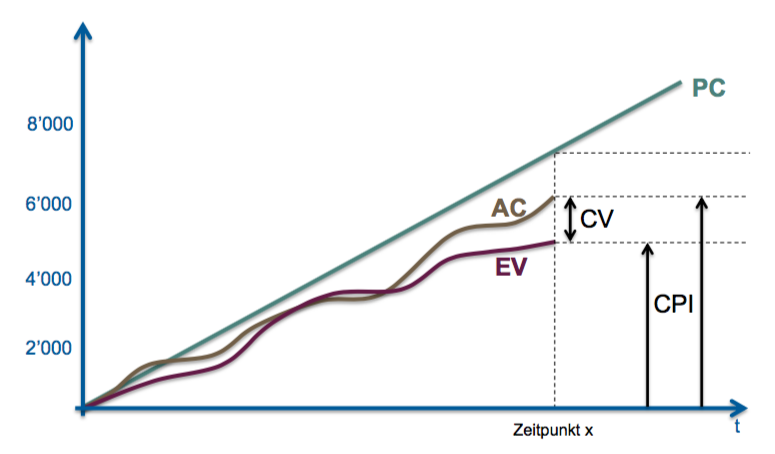
## Standards für die Projektkontrolle

Qualitäts- und Reifegradstandards helfen Ihnen um das gleiche Verständnis von Qualität oder Reife zu schaffen. Als Beispiel CMMI zur allgemeinverständlichen Definiton von Reife. Es wurde 1987 entwickelt vom SEI (Software Engineering Institute). Es beschreibt fünf Reifegrade von Softwareentwicklungsorganisationen). Enthält keine Anleitung wie, sondern legt Ziele fest, welche eine Software, Organisation oder Abteilung erreichen muss. Prüfung erfolgt durch Assessments.

**Für CMMI relevante Prozesse**  


## Earned-Value-Analyse

Die Earned-Value-Analyse (EVA) ist die meist genannte und bekannteste Analyse um den Projektfortschritt zu analysieren und vor allem darzustellen. In Büchern und Artikeln omnipräsent, in der Praxis meist in abgewandelter Form anzutreffen. Die Analyse steht und fällt mit der Bestimmung des Fertigstellungswerts (Earned Value). Ziel der Analyse ist eine Ermittlung der Kosteneffizienz, also ib die verbrauchten Ressourcen im Verhältnis zum Fortschritt stehen. Angestrebt wird eine Kosteneffizienz von >1.

**Planned Cost (PC)**  
Die zu Beginn geplanten Kosten/Aufwände für das Projekt

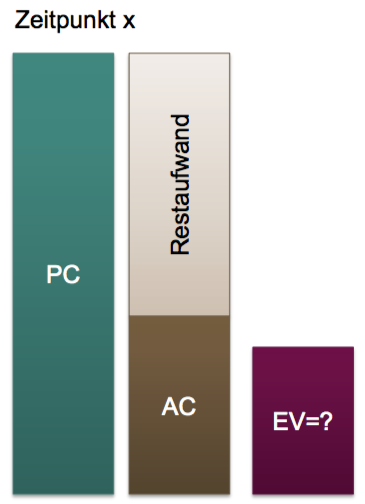
**Actual Cost (AC)**  
Die bis jetzt ausgegebenen Kosten /Aufwände für das Projekt

**Earned Value (EV)**  
Der Wert der bis jetzt erbrachten Arbeit

**Cost Varaince (CV)** Wird berechnet mittels EV – AC

**Cost Performance Index (CPI)** Wird berechnet mittels EV/AC

### Wie berechne ich den Earned Value (Fertigstellungsgrad)?

**Variante «Strikt»**  
Nur realisierte Werte haben einen Nutzen, daher halbfertige Software = 0 CHF.

**Variante «Zwischenresulate»**  
Die Summe aller brauchbaren Komponenten definiert den EV

**Variante «Restaufwand»**  


### Beispiel einer EVA an einem Studentenprojekt

Sie arbeiten in einem Projekt und rechen mit Kosten von CHF 100.- pro Stunde. Im Projekt sind 300h vorgesehen. In der Mitte des Projekts (nach 150h aufgewendeter Arbeit) machen sie eine Restzeitschätzung und realisieren, dass Sie noch 225h für die Fertigstellung benötigen. Wie hoch sind zu diesem Zeitpunkt folgende Kennzahlen?

**Earned Value**PC = 300 \* 100, AC = 150 \* 100, REST = 225 \* 100 🡺 EV = 12000

**Cost Variance**  
EV = 12000, AC = 15000 🡺 CV = -3000

**Cost Performance Index**  
EV = 12000, AC = 15000 🡺 CPI = 0.8

## Meilenstein-Trendanalyse

Die Meilenstein-Trendanalyse (MTA) ist eine rückwartsbetrachtete und sehr eindrucksvolle Analyse, da sie den Verlauf des Projekts schonungslos aufdeckt. Die Analyse basiert darauf, dass Sie mit klug gewählten Meilensteinen arbeiten. Die Meilensteine müssen möglichst wichtige Ereignisse (z.B. Phasenende, Erreichung gewisser Funktionalität, etc.) terminlich fixieren. Die MTA zeichnet den Verlauf der Meilensteine auf. Angestrebt werden möglichst horizontale Linien.

|  |  |
| --- | --- |
|  |  |

# Klassisches Projektmanagement

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Vorgehen * HERMES, Vorgehensmodell des Bundes * Prototypenmodell * Spiralmodell | * kennen mehrere Vorgehensmodelle und können diese abwägen * verstehen klassisches Projektmanagement auf Phasen basierend * verstehen was HERMES beinhaltet * wissen wann klassische Projektmethoden eingesetzt werden |

## Vorgehen

### Vorgehensmodelle

In den Modulen SE1 und SE2 hatten Sie bereits einige Vorgehensmodelle in den Vorlesungen behandelt. Unteranderem RUP, V-Modell und Scrum. Vorgehensmodelle werden in der Praxis frei kombiniert, dies ist ein legitimer Ansatz zur Bewältigung von Herausforderungen. Folgende weitere Modelle werden in diesem Kapitel ebenfalls noch erläutert: **HERMES**, **Prototypenmodell** und **Spiralmodell**.

### Das Vorgehen

Das Vorgehen wird zu Beginn des Projekts definiert, eine Änderung ist nur schwer möglich (aber nicht unmöglich). Es wird zwischen agil und klassisch unterschieden

* Agile Vorgehensmethoden basieren dank interaktivem Vorgehen auf schlanken Prozessen im Projekt, damit weniger Zeit für administratives aufgewendet wird.
* Klassische Vorgehensmethoden sind formeller und strikter, was klare Aussagen zu Fortschritt und Inhalt zulässt, wiederrum aber mehr administrativer Aufwand bedeutet.

Ein «projekttechnisch» guter Projektleiter kennt mehrere Methoden und hat diese auch erfolgreich (oder nicht erfolgreich) angewendet. Klassische Modelle benötigen ein gutes Change Management.

### Klassische Vorgehensmodelle

|  |  |
| --- | --- |
| Wasserfall unter der Lupe | Beispiel |

#### Definition einer Phase

Im klassischen Projektmanagement werden die Arbeitspakete oder Tasks in logische Abschnitt, sprich Phasen unterteilt. Alles was von der Tätigkeit her zeitlich und inhaltlich zusammenhängend ist, wird in eine Phase gepackt, zum Beispiel

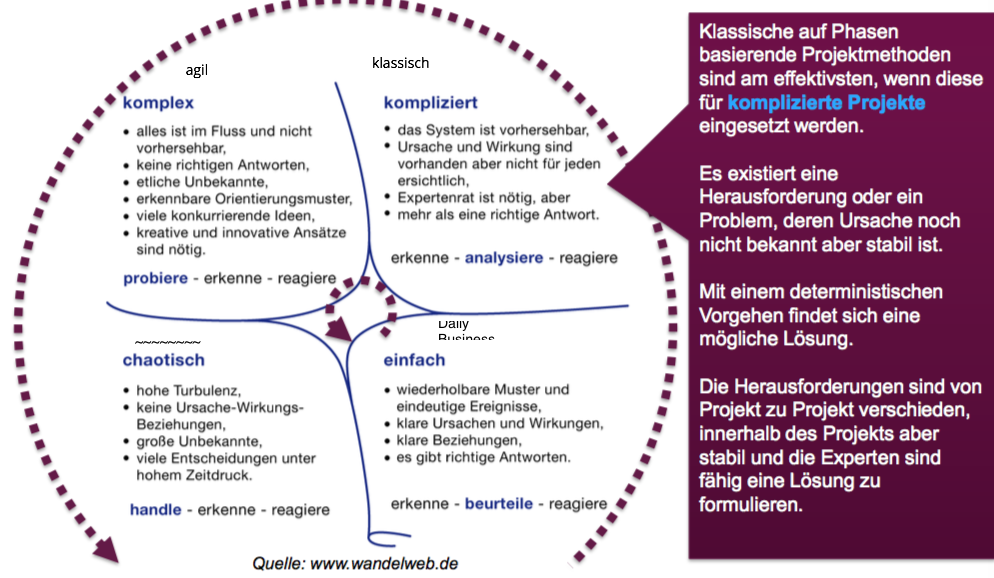


In der Regel terminiert ein Meilenstein eine Phase (schwarzes Viereck). Phasenübergänge werden auch «Gates» genannt. Eine Parallelisierung von Phasen ist nicht vorgesehen, kommt jedoch in der Praxis vor.

### Phasen am Beispiel RUP

RUP den die vier Phasen Inception, Elaboration, Construction und Transition. RUP arbeitet sequentiell. Die fachlichen Zusammenhänge sind jedoch nicht strikt gegeben, da RUP davon ausgeht, dass zum Beispiel noch bis in die Transition-Phase Tasks in der Disziplin «Analysis & Design» erledigt werden müssen.

### CYNEFIN – Wann ein Phasenmodell



## HERMES, Vorgehensmodell des Bundes

HERMES steht für Handbuch der Elektronischen Rechenzentren des Bundes, eine Methode zur Entwicklung von Systemen (hermes.admin.ch). Aktuell gibt es die Version 5.1. Es gibt genau vier Phasen, welche klar vorgegeben sind.

|  |  |
| --- | --- |
| Phasen | Rollenverteilung bei HERMES |
| Aufgaben der Projektsteuerung | Aufgaben der Projektführung |
| Alle Projektteilnehmer nach HERMES (standardisiert) |  |

|  |  |
| --- | --- |
| Pro Projektteilnehmer ein definiertes AK(F) | Jede Aufgabe ist bis ins Detail beschrieben |

### Vor- und Nachteile von HERMES

|  |  |
| --- | --- |
| **Vorteile** | **Nachteile** |
| * Hohe Standardisierung mit klar definierten Phasen, Rollen, Aufgaben, Ergebnissen, etc. * Eine Zertifizierung erlaubt die Mitarbeit in Bundesprojekten * Einbettung von Scrum ist klar definiert * Passt zu Institutionen, welche ihre Aufträge öffentlich ausschreiben oder Dienstleister welche an den Ausschreibungen teilnehmen. | * Sehr starke Vorgaben mit wenig Spielraum * Vier Phasen sind etwas knapp bemessen * In der Privatwirtschaft und sowieso ausserhalb der Schweiz kaum relevant |

## Prototypenmodell

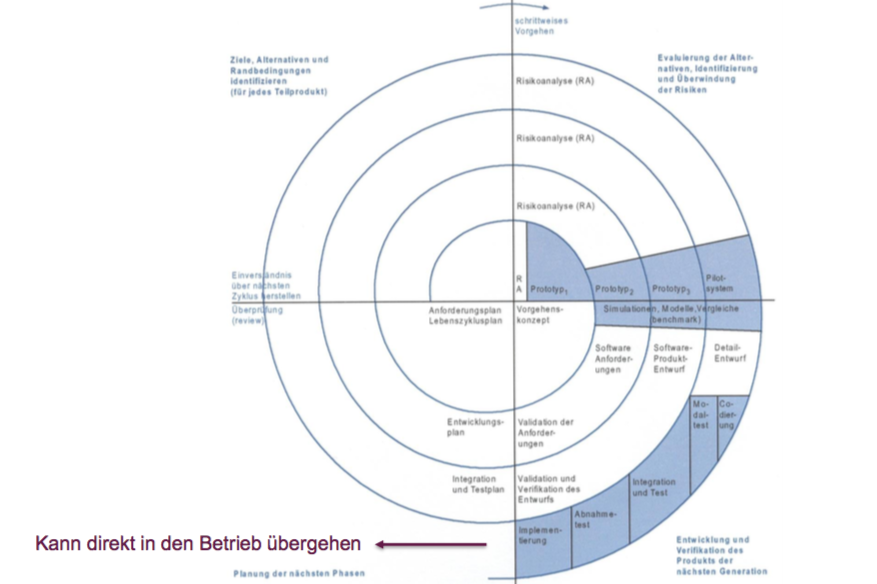
|  |  |
| --- | --- |
|  |  |

Jeder Prototyp selbst muss definiert, entworfen und implementiert werden. Beispiele aus der Praxis sind Aeromobil oder Rollable Display.

### Merkmale

* Auftraggeber und Endbenutzer sind oft nicht in der Lage, die Anforderungen zu beschreiben
* Während der Entwicklung ist oft eine Absprache mit dem Anwender notwendig
* Kann mit anderen Modellen wie Wasserfall, V-, RUP-Modellen kombiniert werden
* Realisierbarkeit von Anforderungen lässt sich manchmal nicht garantieren
* Unterstützt die Akquisitionsphase, indem der Anwender mit Prototypen von speziellen Funktionen überzeugt werden kann
* Es gibt verschiedene Prototypen (Demo, Muster, Pilot) aber diese sind in der Regel einzigartig

## Spiralmodell

Risikogetriebenes Modell, bei dem oberstes Ziel die Minimierung des Risikos ist. Jede Spirale stellt einen iterativen Zyklus durch dieselben Schritte dar. Die Ziele für jeden Zyklus werden aus den Ergebnissen des letzten Zyklus abgeleitet. Keine Trennung der Entwicklung und Wartung. Das Ziel ist es beim Kleinen zu beginnen und damit die Spirale so eng wie möglich zu halten und erreiche somit die Entwicklungsziele mit minimalen Kosten. Bei der Zielbestimmung werden auch Qualitätsziele aufgeführt. Für jede Aktivität und jeden Ressourcenverbrauch wird gefragt «Wieviel ist genug?». Dadurch wird ein «Overengineering» vermieden.

## Diskussion

|  |  |
| --- | --- |
| **Projektstandards/Zertifizierungen und ihre Modelle** | **Einordnung der Standards und Vorgehen** |

# Finanzierung, Kosten und Beschaffung

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Kosten * Nutzen * ROI * Business Case * Finanzierung und Liquidität * Beschaffung und Verträge | * ermitteln Kosten/Nutzen/ROI eines Projekts * wissen was ein Business Case beinhaltet * erkennen die Schlüsselpunkte bei der Finanzierung * kennen die Stolpersteine bei der Beschaffung und die verschiedenen Vertragsarten |

## Kosten

Die Kosten in einem Projekt können sich aus folgenden Aufwänden zusammensetzen.

* Personalaufwand (intern/extern)
* Material (intern/extern)
* Software (Kauf/Miete)
* Spezielle Ausbildungen (intern/extern)
* Spesen (Hotel, Transport, Essen, etc.)
* Finanzierung (Zins, Versicherungen)
* Diverses (Bücher, Normen)
* Rückstellungen (Reverse)
* Risiken (Risiko Positionen)

|  |  |
| --- | --- |
| **Beispiel Projekt-Kosten-Kalkulation** | **Personalkosten in einem Unternehmen**  Der Gehalt ist nur die Hälfte der wirklichen Kosten, für das Unternehmen kommt nochmals der Bruttolohn dazu. |

Die Auslastung des Mitarbeiters hat einen Einfluss auf den Stundensatz gegenüber dem Kunden. Die Unternehmen haben ein Zielband von 60 bis 80 Prozent, dass bei 80 % in 160 Franken pro Stunde resultiert.

## Nutzen

### Nutzenelemente von Projekten

|  |  |
| --- | --- |
| **Finanzen** Return of Invest Weniger Ausgaben Mehr Einnahmen | **Compliance** Erfüllung der Gesetze Einhaltung Normen Einhaltung Soft-Law |
| **Agilität** Schnelle Prozesse Flexible Strukturen Hohe Geschwindigkeit | **Qualität** Image erhalten Zuverlässigkeit Datenhygiene |

Der Nutzen eines Projekts muss wie die Kosten auch immer vor dem Projektstart bekannt und vom Management akzeptiert sein. Quantitativer Nutzen (materiell) zeigt vor allem einen Finanz-Nutzen. Qualitativer Nutzen (immateriell) zeigt vor allem Elemente von Compliance, Agilität und Qualität, z.B.

* Erfüllung neues Datenschutzgesetz
* Schneller am Markt als Konkurrenz
* Stärkung der eigenen Marke dank hoher Verfügbarkeit

## ROI

Return of Invest 🡺 Hole ich meine investierten Kosten über die Zeit wieder zurück und wenn ja wann?. Bei der Berechnung der gibt es eine Fallunterscheidung. Folgende Fälle gilt es zu unterscheiden.

|  |
| --- |
| **Betrachtung ob Neuanschaffung oder Ersatzinvestition**   * Neuanschaffung   + Einmalige Kosten und jährliche Betriebskosten   + Zukünftiger Nutzen * Ersatzinvestition   + Einmalige Kosten   + Unterschied Betriebskosten neu/alt   + Unterschied Nutzen neu/alt |
| **Betrachtung der Finanzierung**   * Fremdfinanzierung   + Betrachtung der Zins- und Rückzahlungskosten * Eigenfinanzierung   + Diskontierung (Zins + Risiko) einbeziehen |

### ROI Beispiel – Einwicklung einer neuen App

Ein amerikanischer Partner gibt Ihnen 80'000 Dollar als Darlehen, welches Sie für die Entwicklung der Fitness-App verwenden. Sie zahlen dafür ab dem zweiten Jahr jährlich 6 Prozent Zins für dieses 5-Jahres-Darlehen.

Sie brauchen genau ein Jahr für die Entwicklung, dann können Sie das App auf einem Server betreiben. Sie benötigen keine Lizenzen oder separate Hardware. Der Serverbetrieb (as a Service) kostet 10'000 Dollar pro Jahr. Pro App verdienen Sie einen Dollar. Sie rechnen nach Entwicklungsende vier Jahr lang mit je 40'000 bezahlen Downloads pro Jahr. Wenn Sie den jeweiligen Ertrag auf eine Schweizer Bank legen, kriegen Sie 0.5 Prozent Zins pro Jahr.

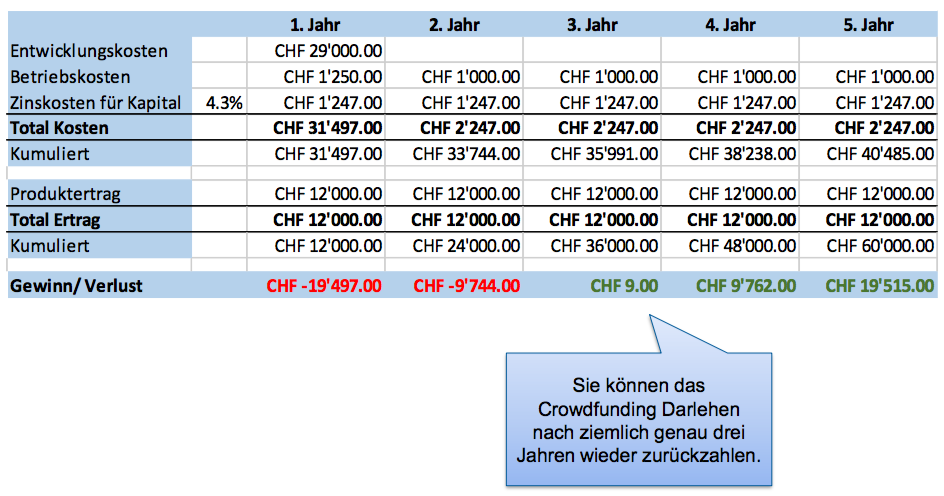
|  |  |
| --- | --- |
|  |  |

### ROI Beispiel – WLAN Access Points ersetzen

Sie betreiben als Student nebenbei für mehrere Hotels jeweils ein WLAN Access Point Netzwerk. Die Hotels schätzen diesen Dienst sehr, weil sie wenig damit zu tun haben und die WLAN Codes an die Gäste weiterverkaufen können. Sie nehmen dadurch total jedes Jahr CHF 12’000.- ein und haben selber sehr geringe Betriebskosten von jährlich CHF 2’000.-.

Nun möchten Sie für CHF 29’000.- alle Access Points innerhalb von drei Monaten und dabei Modelle einkaufen, die nur noch CHF 1’000.- Betriebskosten pro Jahr verursachen. Das Ersetzen ist innerhalb drei Monate möglich. Während dem Vierteljahr sind die Betriebskosten gemischt.

Leider fehlt Ihnen momentan das Geld für den Ersatz. Mittels Crowdlending haben sie die CHF 29’000.- als Darlehen gesammelt, 4.3% Zins pro Jahr. Wann erreichen Sie den ROI?



### Diskontierung – Was ist das?

Die Diskontierung ist für einen Ingenieur ohne betriebswirtschaftliche Vorbildung ein schwer zu verstehendes Konstrukt. In diesem Modul behandeln wir das Thema so oberflächlich wie möglich und zwar mit folgenden Grundsätzen:

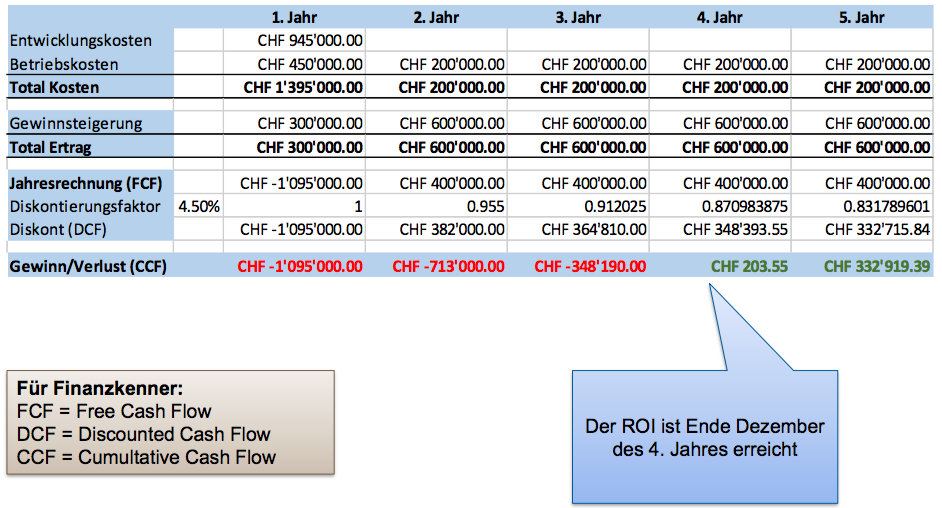
* 100 Franken heute sind nicht 100 Franken morgen
* Gewinne in der Zukunft sind somit weniger wert als Gewinne heute
* Wenn ich Geld ausleihe, möchte ich Zins und eine Risikoprämie
* Jedes Unternehmen hat seinen eigenen internen Zinssatz, wenn es für «sich selber Geld ausleiht» und in ein Projekt investiert.
* **Diskontierungssatz = Interner Zinssatz + Risikozuschlag**

### ROI Beispiel – Online Shop einer Handelskette

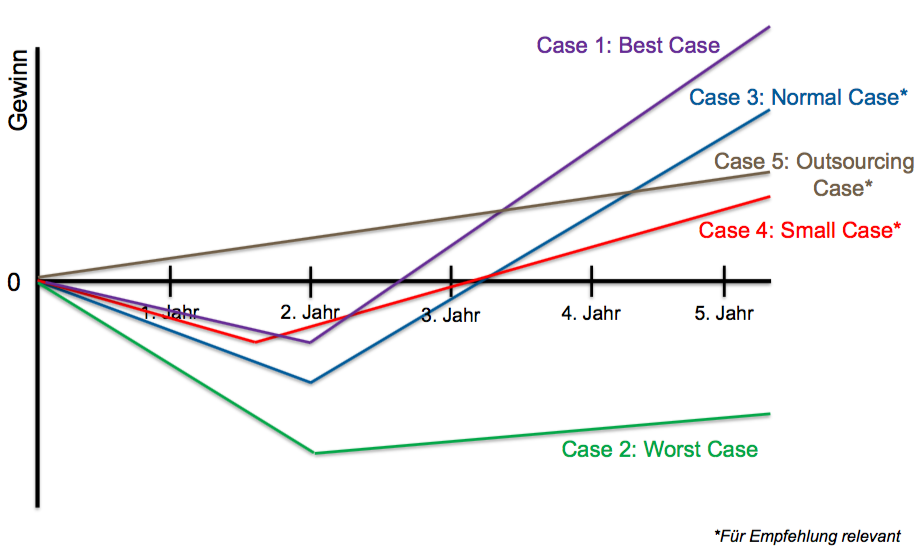
Eine bestehende Online Shop Software hat Betriebskosten in der Höhe von CHF 300’000.- jährlich. Die Entwicklung einer verbesserten Version kostet CHF 945’000.- und kann vom Unternehmen selber bezahlt werden, Diskontierungssatz beträgt 4.5%.

Die verbesserte Software hat Betriebskosten in der Höhe von CHF 200’000.-. Mit der verbesserten Software wird ihr Online Shop CHF 50’000.- pro Monat mehr Gewinn erzielen.

Sie benötigen für die Überarbeitung 6 Monate, in dieser Zeit fallen die Betriebskosten der neuen und alten Software an. Wann ist der ROI erreicht?



## Business Cases

Ein Business Case vergleicht mehrere Geschäftsfälle oder auch mehrere Varianten miteinander. Im Prince2 Standard muss vor jedem Projekt ein Business Case gemacht werden. Ein Business-Case beinhaltet

* Management Summary
* Abgrenzung
* Kosten der verschiedenen Cases
* Monetäre Nutzen der Cases
* ROI Berechung pro Case
* Quantitative Nutzen pro Case
* Risiken und Changen sofern noch nicht einberechnet, - Empfehlung Entscheidung

## Finanzierung und Liquidität

Ein Projekt muss finanziert werden, das heisst die notwendigen finanziellen Mittel müssen am richtigen Ort und im richtigen Moment bereits sein. Eine Finanzierung kann auf zwei Arten passieren:

* Interne Mittel des Unternehmens
* Externe Mittel durch Ertrag am Markt (z.B. Teilzahlung durch Kunde)

Projekte die extern finanziert werden, wie die internen Mittel nicht ausreichen sind per se Hochrisiko-Projekte (Konkursriskio). Extern finanzierte Projekte sind tendenziell rentabler und unterliegen einer schärferen Kontrolle.

### Projektliquidität

Die Liquidität beschreibt die vorhandenen flüssigen (finanziellen) Mittel. Bei intern finanzierten Projekten muss das Unternehmen selber die notwendigen Liquidität bis zum Schluss bereitstellen, meistens werden die Mittel phasenweise freigegeben. Bei extern finanzierten Projekten müssen die stetigen Einnahmen jeweils vor den Ausgaben erzielt werden. Wenn Sie für einen Kunden arbeiten, dann gibt es zwei mögliche Zahlungsmodalitäten:

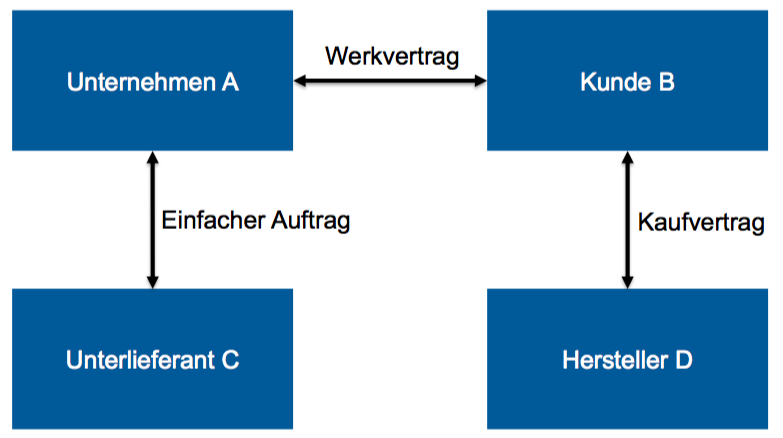
* Gestaffelte Zahlung, abhängig von Zeit und Fortschritt (z.B. Meilensteine)
* Regelmässige Zahlung nach Aufwand (z.B. monatlich)

|  |  |
| --- | --- |
| **Gestaffelte Zahlung abhängig von der Zeit**    In diesem Beispiel reicht die externe Finanzierung jeweils Ende Quartal nicht mehr aus, weswegen auch eine interne Finanzierung von ca. CHF 100’000.- notwendig ist. | **Gestaffelte Zahlung abhängig von den Milestones**    Hier ist zu Beginn ebenfalls eine interne Finanzierung notwendig, wobei es abhängig von dem Projektfortschritt ist, wann das Projekt nur noch extern finanziert wird. |
| **Auftragsprojekt – Zahlung nach Aufwand**    Hier zeigt sich, dass nur zu Beginn eine interne Finanzierung von ca. CHF 100’000 notwendig ist, nach einem halben Jahr ist das Projekt komplett extern finanziert. |  |

#### Wichtige Punkte

* Zahlungsziel wird meistens NICHT eingehalten (Zahlungsmoral)
* Rechtzeitige Verrechnung der Projektleistungen
* Vorfinanzierung
* Wie wir gesehen haben, benötigen die meisten Projekte eine interne Vorfinanzierung. Wenn diese intern nicht möglich ist, gibt es nebst Kunden folgende weitere externen Möglichkeiten:
  + Investitionskredit bei einer Bank (4%-8%)
  + Betriebskredit/Kontokorrent ausschöpfen (4%-8%)
  + Kapitalerhöhung
  + Crowdlending
* Bauen Sie wenn immer möglich eine Liquiditätsreserve ein
* Liquiditätsplanung soll mit dem Risikomanagement verknüpft werden
* Liquidität ist wie die Luft zum Atmen für den Menschen!

## Beschaffung und Verträge

Sie können selten alle Mittel für ein Projekt selber zur Verfügung stellen sondern müssen daher auch Mittel projektspezifisch beschaffen, dies können sein:

* Hardware
  + Kaufvertrag nach Art. 184ff OR
  + Miete (Innominatvertrag)
* Softwarelizenzen (Lizenzvertrag)
* Externe Dienstleistungen
  + Einfacher Auftrag nach Art. 394ff OR
  + Werkvertrag nach Art. 363ff OR

### Was ist ein Vertrag?

Eine gegenseitige Verpflichtung (Obligation) von zwei oder mehreren Parteien. Es drei mögliche Vertragsformen: Schriftlich, mündlich und konkludent. In der Schweiz gibt es nur für wenige Verträge eine Formvorschrift, daher Sie können als zum Beispiel einen Arbeitsvertrag auf einem Bierdeckel oder mündlich vereinbaren. Konkludente Verträge, also ohne Schrift und Wort, sind eher für kleinere Verträge vorgesehen, z.B. Besteigen eines Postautos (Billett beim Fahrer lösen).

### Unterscheidung der Verträge für Dienstleistungen

|  |  |
| --- | --- |
| **Werkvertrag** | **Einfacher Vertrag** |
| * Lieferergebnis geschuldet * Preis ist festgelegt * Fixer Termin mit Pönale * Lieferant ist Projektleiter * Erbringung ortsunabhängig * Abnahme der Lieferergebnisse * Changes gegenseitig * Keine gesetzliche Haftung * Risiko beim Lieferanten | * Arbeitsleistung geschuldet * Preis nach Aufwand * Fixer Termin ohne Pönale * Kunde ist Projektleiter * Erbringung idR beim Kunden * Keine Abnahme der Arbeit * Changes einseitig * Beratungshaftung * Risiko beim Kunden |

# IT Service Management nach ITIL v3

## Einleitung

### Motivation

Unternehmen sind zunehmend von der IT abhängig (wachsender Bedarf an IT Services). Zudem sind die Informationen meist das wertvollste Gut einer Unternehmung. Es ist ein Trend weg von der Entwicklung der IT Anwendungen hin zum Management von IT Services (Saas). Da unterstützt das ITIL (Information Technology Infrastructure Libaray), ein defacto Standard welcher Best Practices behinhaltet.

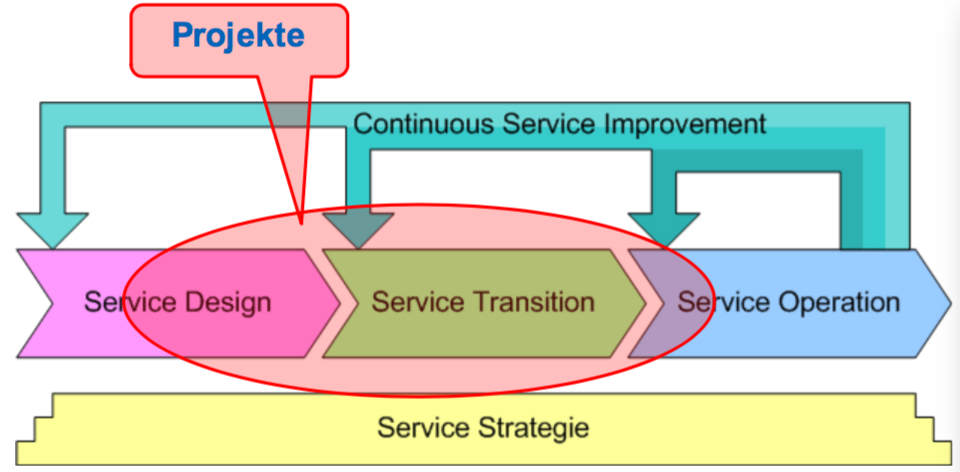
### Historie

|  |  |
| --- | --- |
| **1989** | ITIL V1  80486 und i860 Prozessoren  Portabler Mac  Nintendo Game Boy  HTML wird erfunden und www wird entwickelt |
| **2000** | ITIL V2  Tablets und Mobiles  Erstes Mobile mit Kamera  PlayStation 2  USB Flash Drive  Windows 2000 |
| **2007** | ITIL V3  iPhone  Amazon Kindle  Dropbox  Windows Vista |

### Ziele von IT Service Management

* Zuverlässige und konsistente IT Services
  + Mit hoher Qualität
  + Zu tragbaren Kosten
* Abstimmung der IT Services auf die Geschäftsanforderungen
  + Kostenoptimierung
  + Qualität Sicherung (steuerbares Qualitätsniveau)
  + Betriebsoptimierung
* Trennung von Entwicklung und Betrieb 🡺 neuer Trend: Dev-Ops
* Definierte Schnittstelle zum Business (Kunde / Benutzer) 🡺 neuer Trend: Produkte-Teams

### ITIIL Übersicht v3



## Service Design

## Service Transition

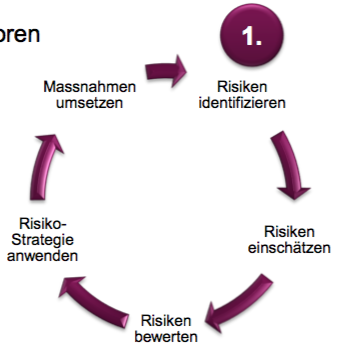
## Service Operation

# Risikomanagement

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Analyse * Bewertung * Bewältigung * Risiken aus der Praxis * Chancen | * identifizieren Risiken und schätzen diese ein * lernen wie Risiken beurteilt werden * üben die Bewältigung von Risiken * wissen wie das Risikomanagement in der Praxis funktioniert |

## Analyse

Risiken und Chancen sind Unsicherheitsfaktoren, welche das gewünschte Ergebnis negativ oder positiv beeinflussen können. Das Management der Risiken muss Bestandteil im Projektmanagement sein, das Management der Chancen sieht man eher stelten, ist jedoch gerne gesehen.

Ein Risiko besteht im Wesentlichen aus zwei Faktoren. Der Eintretens Wahrscheinlichkeit (oder Häufigkeit) und dem Schadensausmass (oder Auswirkung). Risiken müssen zuerst identifiziert werden.

### Beispiele aus der Schweiz

|  |  |
| --- | --- |
|  | **Hochwasser Sihl** |

### Identifizieren

Die in der Fachwelt vier häufigsten Analyse für das Identifizieren sind:

**Impact Analyse**  
Dies ist eine geläufige Alternative zu einer vollständigen Risikoanalyse, mit der ich die wichtigsten Auswirkungen eines Projekts untersuche und deren Störung als Risiko strukturiere

**Bedrohungsanalyse**  
Damit untersuche ich das Projekt gegen alle möglichen Bedrohungen aus einem Bedrohungskatalog und strukturiere die relevanten als Risiko

**Schwachstellenanalyse**  
Aus der Erfahrung ähnlicher Projekte identifiziere ich die grössten Schwachstellen eines Projekts und strukturiere diese als Risiko

**Beliebige Kombinationen**  
Jegliche Kombinationen sind möglich, wobei eine Analyse von Impact, Bedrohung und Schwachstellen einer vollständigen Analyse entspricht

### Einschätzen

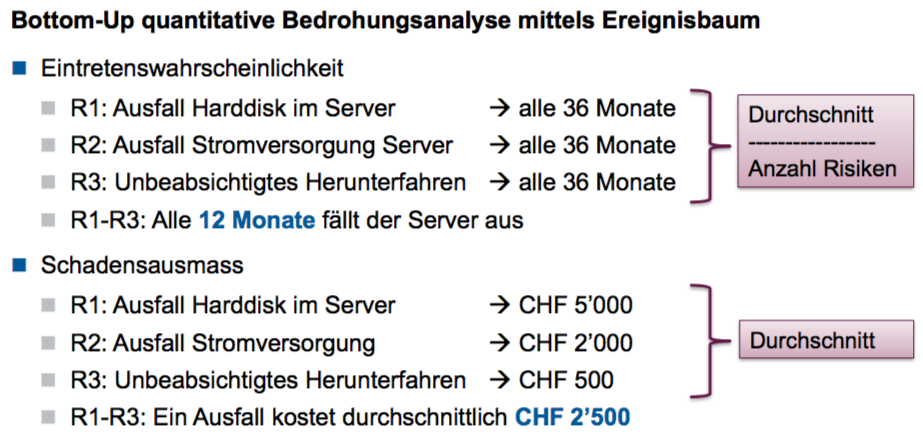
Sie müssen jedes Risiko qualitativ (hoch, mittel, tief) oder quantitativ (Zeit und Kosten) einschätzen, dies können Sie auf verschiedene Arten tun:

* Schätzung der maximalen Werte
* Schätzung der mittleren Werte
* Schätzung der Werte mittels statistischen Verteilungsfunktionen

Schätzen Sie zuerst die Eintretens Wahrscheinlichkeit und dann das Schadensausmass, entweder Bottom-Up oder Top-Down

* Bottom-Up bedeutet eine Analyse mit Detail die sie dann zu einer grossen Analyse aggregieren
* Top-Down bedeutet eine übergreifende grobe Analyse, die sie dann Schritt für Schritt detaillieren

### Einschätzungsbeispiel «Serverausfall»

Der Ergebnisbaum fasst also von den einzelnen Ereignissen kommend ähnliche Risiken zusammen und gruppiert diese so, wobei an Schärfe verloren geht.

## Bewertung

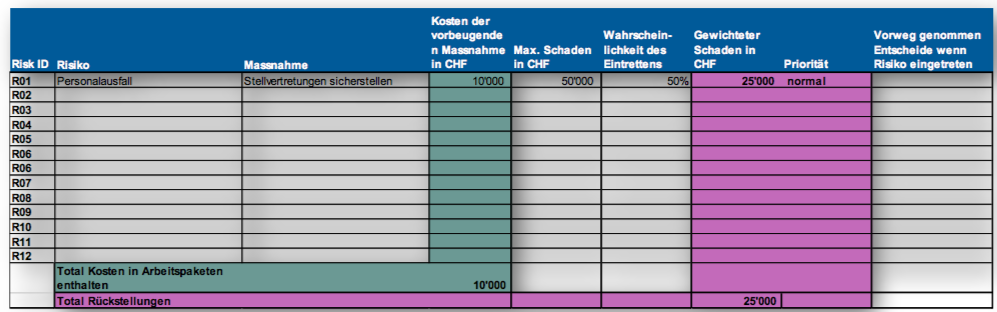
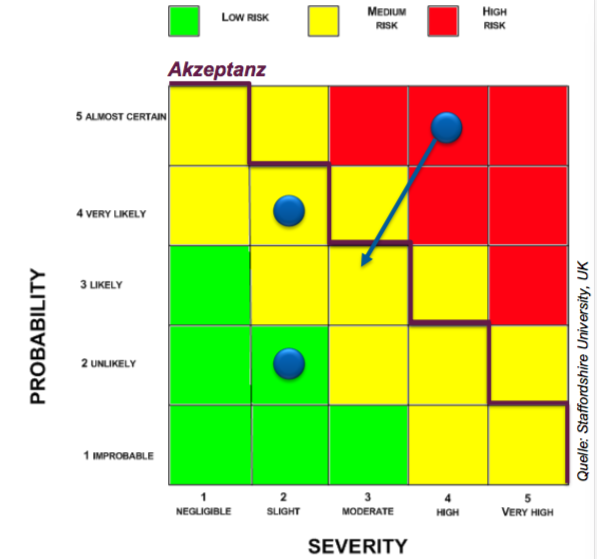
Durch die Bewertung versuchen Sie die erhaltenen Risikowerte in einen Kontext zu bringen und diese auch wenn möglich terminlich zu fixieren oder priorisieren. Beispiele

**Beispiel 1** In einem grossen Entwicklungsprojekt mit 30 Mitarbeitenden wird der Personalausfall aufgrund der Fehltage im Unternehmen auf qualitativ «hoch» angesetzt. Eine Betrachtung im Kontext zeigt aber auf, dass nur ein Teilprojekt mit 5 Personen auf dem kritischen Pfad einen leichten Verzug hat. Mit gesundem Menschenverstand drängt sich daher eine Reduktion auf «mittel» auf.

**Beispiel 2** In einem Infrastrukturprojekt werden Mitarbeitende eines Modehauses mit einem speziell gehärteten Tablet versorgt, auf welchem das zu entwickelnde App Beratungsvorschläge erteilt. Die Tablets müssen spätestens 3 Monate vor Projektende geliefert werden. Das Risiko einer Verspätung wurde qualitativ auf «mittel» angesetzt. Hier drängt sich eine terminliche Fixierung des Risikos auf.

### Risikoliste und Risikograph

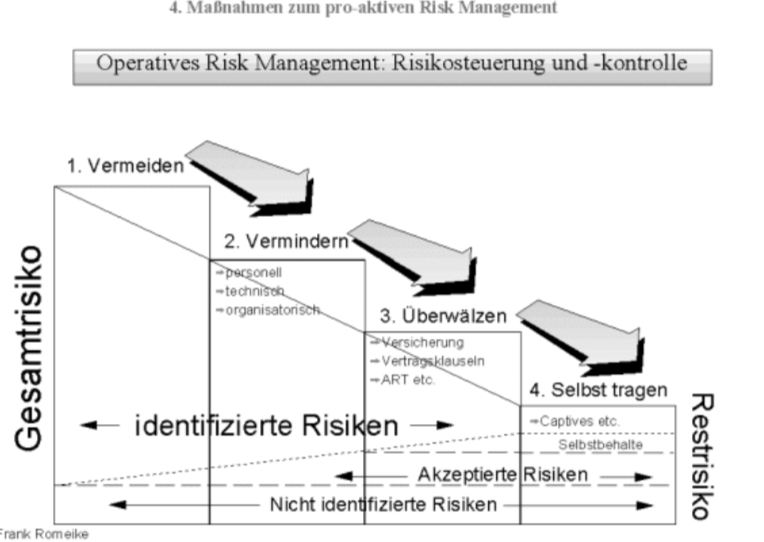
Führen Sie in einem Projekt immer eine Risikoliste, diese kann ihre «Lebensversicherung» sein, wenn etwas schief gehen sollte. Zum Schluss der Bewertung tragen Sie alle Risiken auf einem Graphen ein. Die Risikostrategie ihres Unternehmens definiert die Risiko Akzeptanzlinie. Alle Risiken über der Linie reduzieren sie prioritär und auf jeden Fall.



### 20 Fehler nach Lindecker (Software Projects, 1989)

|  |  |
| --- | --- |
|  |  |

## Bewältigung

Vorbeugende Massnahmen zur Risikoreduktion dürfen im Wesentlichen nicht teurer sein, als das Risiko selber. Nehmen Sie sich wenige aber griffige Massnahmen aufs Mal vor und kommunizieren Sie dies im Projekt. Akzeptieren Sie die Restrisiken, wenn keine Massnahmen mehr von Nutzen sind. Wenn das Risiko trotzdem eintreffen sollte, können Sie bereits Entscheidungen vorweg vorbereiten.

### Beispiel von Massnahmen

**Gegen Personalausfall** Jeder Projektmitarbeiter hat einen Stellvertreter und informiert diesen wöchentlich über den Stand der Dinge.

**Gegen schlechte Datenqualität** Zusätzliche Tests überprüfen systematisch die Anzahl Records und stichprobenweise die Inhalte der Records.

**Gegen unklare Anforderungen** Zusätzliche QS Stelle im Projekt

**Gegen Währungsrisiken** Das Projektvolumen im Euro wird mit einem Hedge abgesichert.

## Risiken aus der Praxis

### Top-Ten Risiken einer Software-Entwicklung

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Risikobasiertes Vorgehen für ein nicht konfiguriertes Produkt | Risikobasiertes Vorgehen für ein konfiguriertes Produkt |
| Risikobewertung für eine kundenspezifische Applikation |  |

## Chancen

Chancen sind das Gegenteil von Risiken, überlegen Sie sich auch hier ob es welche gibt und welche **Eintretens Wahrscheinlichkeit** und **Nutzen** es gibt. Typische Chance sind:

* Neue Hardware-Generation machen Tuning überflüssig
* Entsprechende Klassen werden auch von einem anderen Kunden verlangt (Kostenteilung

Mit einer SWOT Analyse macht man in der Betriebswirtschaft meist einen Gegenüberstellung.

# Agiles Projektmanagement I

## Ziele des Kapitels

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

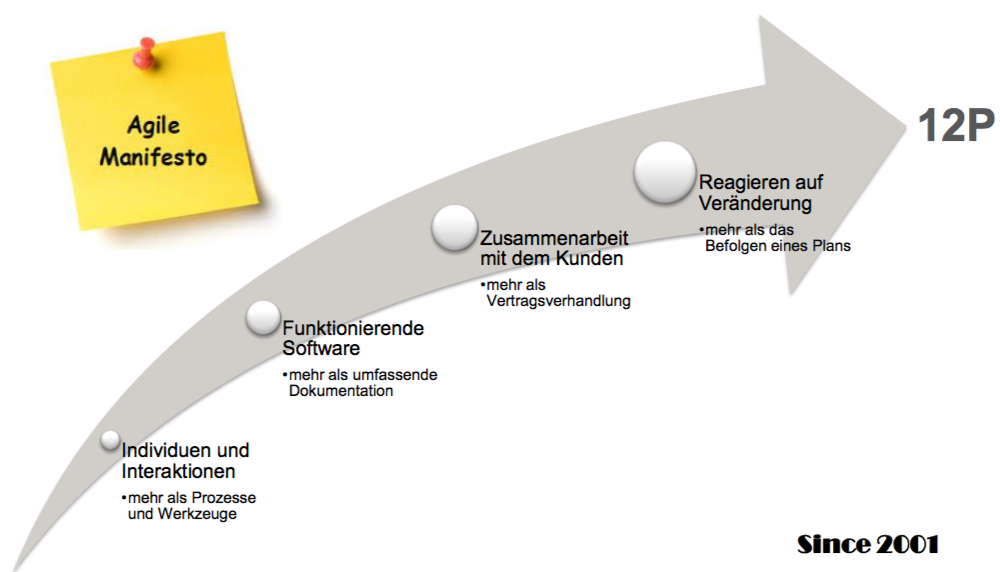
## Motivation für Agilität

Meist ist der Plan eine gerade Strasse, in der Realität trifft man dann aber eine Hügellandschaft mit weiteren Herausforderungen an. Die Agilität kann hier helfen. Die beiden folgenden Statistiken zeigen den Unterschied zwischen Wasserfall und Agil in der Schweiz.

|  |  |
| --- | --- |
|  |  |

Je grösser das Projekt, desto höher ist die Wahrscheinlichkeit zu scheitern. Dies aufgrund von vielen Abhängigkeiten, Ursache – Wirkung nicht erkannt, Fachwissen zu tief oder Anforderungen unklar. Daher Iterationen anstatt unpräzise Analyse zu beginn. Die Wahrscheinlichkeit lässt sich so von 3 Prozent auf 18 Prozent steigern bei grösseren Projekten.

## Agiles Manifest



### Zwölf Prinzipien

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

Damit Prinzip 10, Einfachheit erreicht werden kann muss man immer alle Tests laufen lassen, es darf kein redundanter Code vorhanden sein, und die Methoden und Klassen müssen immer minimalistisch sein.

## Bestandteile eines agilen Vorgehens

|  |  |
| --- | --- |
| **Inspect & Adapt** | **Transparenz** |
| **Selbstorganisation** | **Timeboxing** |

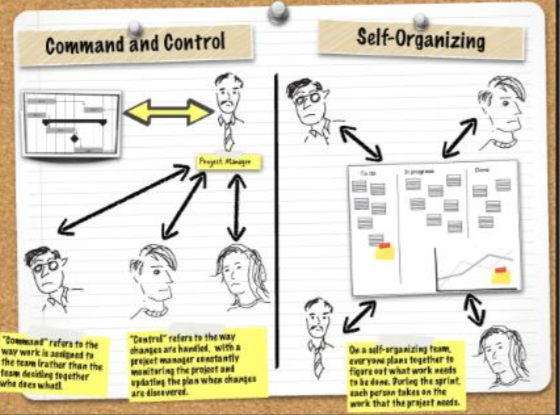
### Inspect & Adapt

Sämtliche Bestandteile im Projekt werden regelmässig überprüft und wenn notwendig angepasst, daher auch wichtige Anforderungen dürfen jederzeit einfliessen.

### Transparenz

Informationen stehen jedem zur Verfügung. Es gibt keine versteckte Agenda oder alternative Fortschrittskontrollen. Dies verlangt nach einer Visualisierung der Daten, zum Beispiel mit Post-Ist auf einer Wand.

### Selbstorganisation

Ein Unternehmen beauftrage Ken Schwaber (Erfinder von Scrum) agiles Projektmanagement einzuführen. Für ein Grossprojekt waren 140 Personen vorgesehen. Nun sollte man von Ken Schwaber wissen, wie die Projektorganisation aussehen sollte und wie viele Teams in welcher Grösse zu bilden sind. Seine Philosophie lautete : «*Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. Fewer than three Development Team members decrease interaction and results in smaller productivity gains. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Having more than nine members requires too much coordination.”*

### Timeboxing

|  |  |
| --- | --- |
| **Zeit vor Inhalt** | **Inhalt vor Zeit** |
|  |  |

#### User Stories

Eine User Story wird im agilen Projektmanagement anstelle von Anforderungen erhoben und in einem Backlog abgelegt. Schema: «Wer» möchte «was» (und «warum»).

**Beispiel für Zeiterfassungssoftware**  
Ich als Projektleiter möchte die Buchungen meiner Mitarbeitenden freigeben können, damit diese Gültigkeit erlangen.

## Heute verwendete agile Methoden

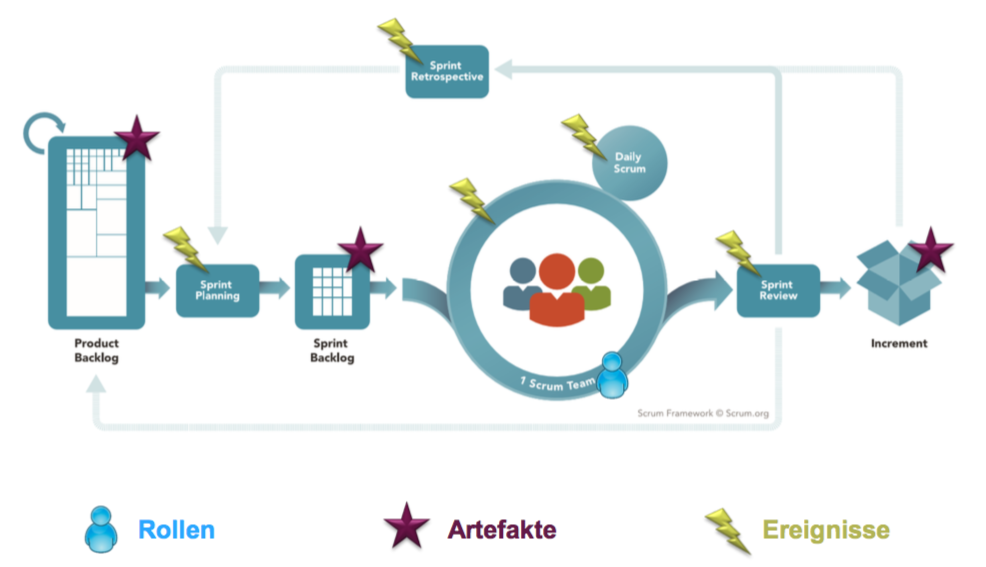
|  |  |
| --- | --- |
| **Welche agilen Methoden werden in der Schweiz verwendet** |  |
|  |  |
|  |  |

# Agiles Projektmanagement II

## Ziele des Kapitels

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

## Scrum Framework in der Tiefe



### Die Elemente

|  |  |  |  |
| --- | --- | --- | --- |
| **Rollen** | **Artefakte** | **Ereignisse** | Dauer |
| Product Owner | Product Backlog | Sprint | 2- 4 Wochen |
| Development Team | Spring Backlog | Sprint Planning | <4 – 8 Stunden\* |
| Scrum Master | Increment | Sprint Review | 2 – 4 Stunden\* |
|  |  | Sprint Retrospektive | 1.5 – 3 Stunden\* |
|  |  | Daily Scrum | <15 Minuten\* |

\* Streng ausgelegt kein Timeboxing

### Fundament von Scrum

|  |  |  |
| --- | --- | --- |
| **Transparenz** | **Überprüfung** | **Anpassung** |
| * Wesentliche Aspekte sind sichtbar * Gemeinsamer Standard | * Fortschritt ständig überprüfen * Zu gegebenen Zeiten überprüfen | * Anpassung so schnell wie möglich * Anpassung ist stetig möglich |

### Scrum ist eine empirische Methode

|  |  |  |
| --- | --- | --- |
| **Ereignis** | **Überprüfung** | **Anpassung** |
| Sprint Planning | Backlog, Increment | Sprint Ziel |
| Daily Scrum | Fortschritt | Sprint Backlog, Tagesplanung |
| Sprint Review | Increment, Sprint, Product Backlog | Product Backlog |
| Sprint Retrospektive | Verhalten, DoD | Regeln |

### Unterscheidungen

**Sprint Review und Sprint Retrospektive**

|  |  |
| --- | --- |
| Das **Increment** wird überprüft Das Product Backlog darf **angepasst** werden Stakeholder erhalten **Einblick** in das Produkt | **Sprint Review** |
| Das Team überprüft DoD Das Team ändert DoD  Interaktion im Team besprechen | **Sprint Retrospektive** |
| **Lenkungsausschuss** **Statusreport** zu Handen Auftraggeber (nicht in Scrum vorgesehen)  Der Scrum Master zeigt die **Performance** (Burn Down)  Eine Earned Value Analyse wird gemacht | **Keines von beiden** |

**Product und Sprint Backlog**

|  |  |
| --- | --- |
| Das Product Owner priorisiert  Kann so viele Stories wie möglich beinhalten  Einträge daraus werden alle 2-4 Wochen besprochen | **Product Backlog** |
| Das Development Team priorisiert/nimmt entgegen  Enthält nur eine bestimme Anzahl Stories  Das Development Team passt es während Sprint an  Einträge daraus werden täglich besprochen  Maximal 10 % der Zeit für die Vereinfachung verwenden | **Sprint Backlog** |
| Besteht ausschliesslich aus Testfällen  Der Scrum Master priorisiert | **Keines von beiden** |

**Zwischen den Rollen**

|  |  |
| --- | --- |
| Formuliert die Product Backlog Einträge  Nimmt nie am Daily Scrum teil  Optimiert das Increment hinsichtlich Geschäftsnutzen | **Product Owner** |
| Modifiziert das Sprint Backlog während des Sprints  Bestimmt was im Sprint Backlog Platz hat  Ist Eigentümer der Elemente im Sprint Backlog  Legt Definition of Done (DoD) fest | **Development Team** |
| Nimmt am Daily Scrum teil oder stellt Durchführung sicher  Beseitigt Hindernisse die Sprints aufhalten  Schafft Verständnis für die Produktplanung | **Scrum Master** |

### «Verunreinigung» von Scrum

* Scrum kennt keinen Zero-Sprint
* Sprints werden nie verlängert oder verkürzt (Timebox)
* Nur der Product Owner darf Sprint abbrechen
* Burn Down Charts verschiedener Teams nie vergleichen
* Teams werden nicht bestimmt
* Daily Standup ist unbekannt, es heisst Daily Scrum
* Der Product Owner gehört zum Team und ist nicht (nur) Stakeholder
* Arbeit wird nicht verteilt (Push) sondern genommen (Pull)

## Semi-Agil mit Kanban

### 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.»*

### Nutzen

* Arbeit visualisieren
* Ist-Zustand abbilden
* Workflow beobachten
* Beschränkungen einbauen (Fokus auf aktive durch WiP-Limit
* Gesamtarbeit wird in gut überschaubare Aufgaben unterteilt ( Zu grosse Aufgaben fliessen schlecht durch das System)
* Prognosen ableiten (und reagieren)

### Woher kommt Kanban?

Es kommt aus dem japanischen und stellt dort ein System zur Herstellung dar. Bis 1947 hat man nach dem Push Mechanismus produziert. Als einfach mal darauf los. Dies brauchte immer mehr Lagerhallen. 1947 sagte einer bei Toyota, dies sei nicht effizient und ruft den Pull Mechanismus ins Leben. Es wurde nur noch auf Anfrage produziert. Das Lager meldet einen Aufbau bei der Fabrik für Nachfüllungen und das Schiff wiederum beim Lager, wenn es etwas brauchte.

Die genaue Idee lautete wie folgt : «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
* Verschwendung durch Lagerhaltung wird weitgehend vermieden
* Kürzere Reaktionszeiten auf sich ändernde Rahmenbedingungen
  + Marktsegmente, Gesetze
* Senkung der Kosten -> Steigerung des Gewinnes

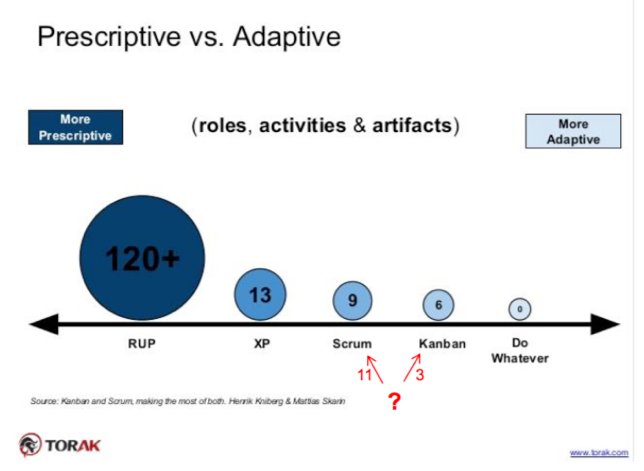
Später (um 2010) wurde dieser Mechanismus auf die Software-Entwicklung angewendet. Folgende Bilder zeigen eine solche Implementierung.

|  |  |
| --- | --- |
|  | **/Users/urs/OneDrive/HSR/Projekt- und Qualitätsmanagement/01_Vorlesung/10_Agiles_PM_II/Kanban.png** |
|  |  |

### Kommentar

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.

### Weitere agile Methoden

Wenn wir Kaban gegen Agilität prüfen, dann wird Inspect & Adapt sowie Timeboxing nicht erfüllt. Die anderen beiden Aspekte schon.

Die Lösung dafür ist **Scrumban**, zum Beispiel mit Sprints (pro Sprint ein Board) und Sprints Reviews.

Somit ergänzen wir also Kanban um agile Elemente aus Scrum.

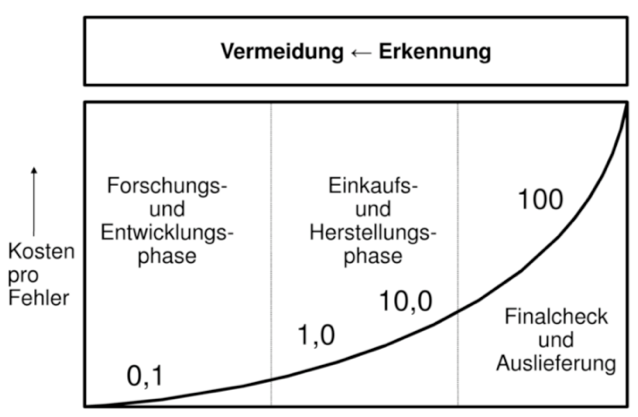
# Qualitätsmanagement

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * QS im Projekt * Change-Management * QS im Unternehmen mit TQM und SixSigma | * kennen Verfahren, um Fehler möglichst gut und früh zu erkennen * verstehen warum Änderungen im Projekt sauber abgehandelt werden * erweitern ihr Wissen um zwei QS Standards im Unternehmen |

## QS im Projekt

Unklare Anforderungen lassen keine exakte Messung zu, womit auch kein genaues Ergebnis erzielt werden kann.

Qualitätssicherung heisst strukturierte Qualitätsprüfung. In den Projekten können frühzeitige Qualitätsinstrumente sicherstellen, dass die Lieferobjekte den Anforderungen entsprechen. Je früher eine Fehlerbehebung ist, desto günstiger – späte Korrekturen sind um einiges teurer.

Negative Beispiel wäre Toyota, welche über 6.5 Millionen Autos in die Werkstatt zurückrufen musste.

### Das V-Modell im Bestreben einer Früherkennung

|  |  |
| --- | --- |
|  |  |

### Verfahren zur Qualitätssicherung in Softwareprojekten

**Review** Die Lieferergebnisse werden von einer weiteren Person einem Review unterzogen. Dies betrifft Spezifikation sowie auch Code. Nebst der Qualität wird die Wartbarkeit sehr stark verbessert. Dies kann bis zur Extreme praktiziert werden (z.B. Paarprogrammierung).

**Continuous Delivery** Regelmässige Auslieferung von Software in die Acceptance oder Produktion. Es wird im Zuge von Exterme Programming verwendet. Entwickler arbeiten nur in einer Versionsverwaltung, eingecheckter Code wird automatisch kompiliert, getestet und paketiert sowie integriet.

**Ad absrudum / Negativtests** Die involvierten Personen müssen die Anforderungen und das Ergebnis aus Prinzio bezweifeln und damit ernsthaft hinterfragen. Gegenteilige Anforderungen werden diskutiert bis diese im Widersprich enden. Es werden Testfälle kreiert, die bewusst einen Fehler auslösen.

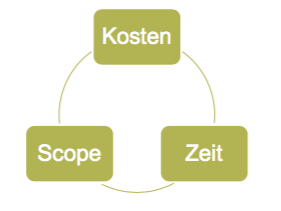
### Beispiel Ad Absurdum

**Der Kunde sagt:** Im Papeterie Webshop für Geschäftskunden sollen die Produkte einzeln ausgewählt und in den Warenkorb gelegt werden. Nachdem der Kunde auf «Einkauf» drückt, darf nichts mehr am Warenkorb geändert werden. Die Währung wird vor der Bezahlung ausgewählt. Danach wird die Mehrwertsteuer mit einer Multiplikation von 1.08 erhoben. Die Zahlung wird vom Vorgesetzten und dem Kostenstellenverantwortlichen des Kunden nacheinander freigegeben.

**Ich behaupte das Gegenteil:** Im Papeterie Webshop sollen die Produkte einzeln/**mehrfach** ausgewählt und in den Warenkorb gelegt werden. Nachdem/**bevor** der Kunden auf «Einkauf» drückt, darf nichts/**noch** mehr am Warenkorb geändert werden. Die Währung wird vor/**nach** der Bezahlung ausgewählt. Zum Schluss wird die Mehrwertsteuer mit einer Multiplikation/**Division** von 1.08 erhoben. Die Zahlung wird dem Vorgesetzten und den Kostenstellenverantwortlichen des Kunden nacheinander/**gleichzeitig** freigegeben/**abgelehnt**.

Der Kunde muss Ihnen die gegenteilige Anforderung soweit widerlegen, dass diese wirklich keinen Sinn macht und daher angenommen werden kann, dass die ursprüngliche Anforderung korrekt und gewollt ist. Beim Code-Review muss der Entwickler Ihnen beweisen, dass die gegenteilige Anwendung nicht möglich ist. Beim Schreiben der Testfälle wählen Sie die gegenteiligen Anforderungen, z.B. das Wechseln der Währung von Euro in tschechische Kronen nach Festlegung der Betragshöhe, bewusst aus und definieren eine Fehlermeldung als erwartetes Resultat.

## Change-Management im Projekt

Ein Change ist eine Änderung im Projekt, ein Request also dessen Antrag. Beispiel: Die Vorabklärung hat zu viel Geld benötigt, weshalb für Konzeption und Entwicklung nicht mehr der benötigte Betrag zur Verfügung steht. Entweder Scope reduzieren (Weniger Inhalt, damit das Geld reicht) oder Budget beantragen (Zusätzliche Mittel um Projektziel zu erreichen. Folgende Kapitel zeigen die verschiedenen Teile eines Change Request.

### Kategorien von Changes Request (I)

Streng genommen sind es genau drei Kategorien (Mehrfachauswahl möglich). **Scope**, **Budget** und **Time**. Weitere Kategorien sind der Praxis anzutreffen:

**Legal Terms**  
Änderung des Vertrages zwischen Lieferant und Kunde, zum Beispiel aufgrund der Verlegung des Projektortes

**Ressourcen**  
Änderung von namentlich genannten Personen, zum Beispiel der Projektleiter als Hauptansprechperson wird ausgewechselt

### Voraussetzungen eines Changes (II)

Der Change Prozess muss definiert sein, das heisst Sie müssen wissen wie und an welches Board (z.B. Lenkungsausschuss) Sie den Change richten müssen. Die Änderung muss fassbar und die beteiligten Personen müssen sich dessen bewusst sein. Die technischen und organisatorischen Voraussetzungen müssen erfüllt oder als Voraussetzung definiert sein.

* Change ist machbar
* Ressourcen sind vorhanden
* Termine sind realistisch
* Alle sonstigen «Hausaufgaben», damit der Change durchführbar ist.

### Beschreibung eines Changes (III)

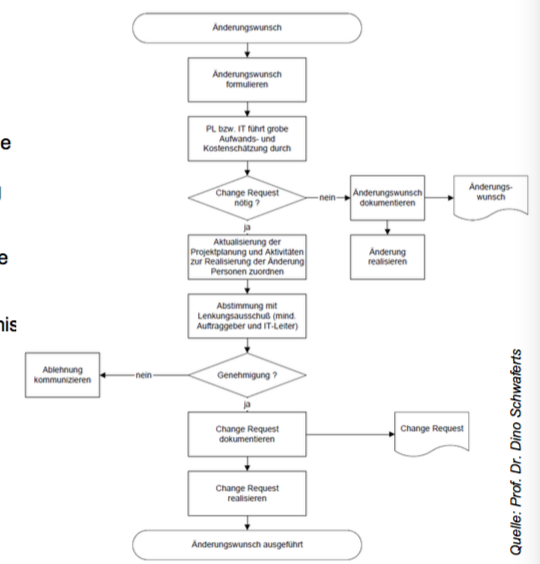
Begründen Sie den Change kurz und prägnant zu Handen einer Zielgruppe, die sich im Projekt nur oberflächlich auskennt. Erpressen Sie keinen Entscheiden sondern suggerieren Sie Wahlfreiheit. Erwähnen Sie unbedingt, wenn der Change auf Basis eines früher erkannten Risikos notwendig wird (macht guten Eindruck, weil PL es früh erkannt hat). **Verwenden** sie Wörter wie: *Ermöglichen, ausserordentlich, neue Rahmenbedingungen, mehr Gewissheit, mehr Sicherheit, Nutzen, Vorteile, etc.*  und **vermeiden** Sie Wörter wie *muss, darf nicht sein, keine Ahnung, aus heiterem Himmel, keine Chance, verlangen, nur diese Lösung, etc..*

### Auswirkungen und Risiken (IV)

Beschreiben Sie die Auswirkung des Change mit all seinen Konsequenzen separat und nicht in der Change Beschreibung. Beschreiben Sie auch die Auswirkungen, wenn der Change abgelehnt wird z.B. *Eine Ablehnung des Changes würde bedeuten, dass der Funktionsumfang begrenzt bleibt und damit bessere Akzeptanz ermöglicht wird.*

Erwähnen Sie, welche neuen Projektrisiken ein Change mit sich bringt z.B. Durch das neue Modukl des Lieferanten XY aus den USA, erhöhen sich die Währungsrisiken, da wird in USD einkaufen.

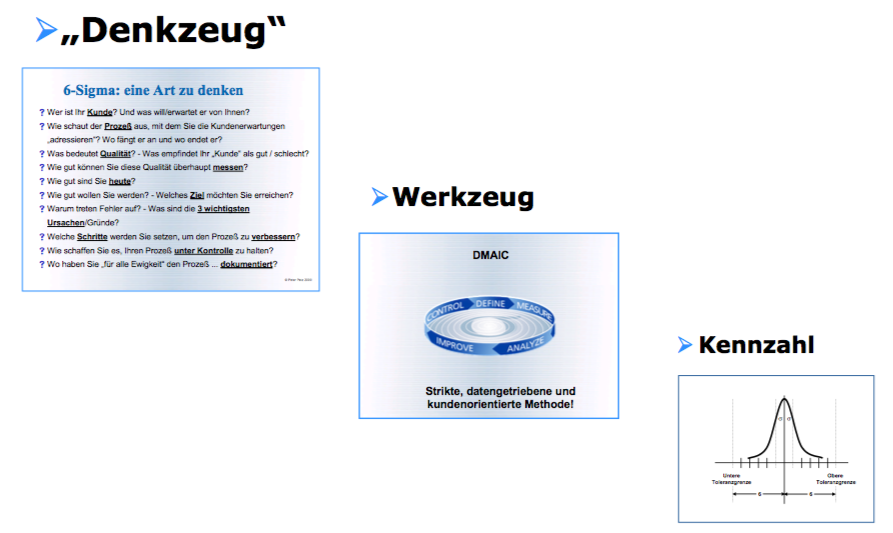
### Change quantifizieren (V)

In einem Change Request müssen Sie immer Budget und Zeit quantifizieren. Oftmals steht die Ampel des Projekts vor dem Change auf rot, da man z.B. erkannt hat, dass das Budget aufgrund von Zusatzanforderungen nicht mehr ausreicht. Mit dem Change haben Sie die Möglichkeit, mehr Budget und mehr Zeit zu «legalisieren» und damit ihr Projekt wieder auf Status grün zu setzen. Quantifizieren Sie die Changes so hoch wie möglich und so tief wie nötig. Wenn Sie neue Anforderungen aufs Auge gedrückt bekommen, ändern Sie nicht nur das Budget sondern verlangen Sie auch mehr Zeit für die Ausführung

### Vorgehen

Gehen Sie nur mit dokumentieren und quantifizierten Anträgen zum Entscheidungsgremium. Nummerieren und dokumentieren Sie die Changes in gleichem Masse, wie einen Projektauftrag oder einen Projektvertrag. Dokumentieren Sie entgegen der nebenstehenden Grafik auch abgelehnte Changes (cover my ass). Im Kunden/Lieferanten Fixpreis-Verhältnis sind Changes eine Änderung des Werkvertrages.  Beim Kunden und Lieferanten müssen gemäss Unterschriftenregelung kompetente Personen unterschreiben

## QS im Unternehmen mit SixSigma

Das Ziel ist Six Sigma. Sigma ist ein statistischer Ausdruck, der die Abweichung von der Norm misst; ein Prozess der auf einem Six-Sigma-Niveau arbeitet, wird in 99,99966 % der Fälle fehlerfrei ablaufen»

### Vergleich klassisches QS mit SixSigma

|  |  |
| --- | --- |
| **Klassisch «99% gut», also (3,8)** | **SixSigma «99,99966% gut» (6)** |
| * **20000** verlorene Briefe pro Stunde in Europa * **5000** falsche chirurgische Eingriffe pro Woche in Europa * **2** Landungen ausserhalb der Rollbahn auf den grössten europäischen Flughäfen täglich | * **7** verlorene Briefe pro Stunde in Europa * **1,7** falsche chirurgische Eingriffe pro Woche in Europa * **0,0007** Landungen ausserhalb der Rollbahn auf den grössten europäischen Flughäfen täglich |

|  |  |
| --- | --- |
| Was bedeutet Sigma genau | SixSigma – 5 stetige Verbesserungsschritte |

## Konfigurationsmanagement

### 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

**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

|  |  |
| --- | --- |
| **Lebenszyklus eines Produkts** | **Beispiel aus der Praxis – Versäumnisse beim Konfigurieren**    Der Fehler trat in der nächsten Version wieder auf, da er nur in der Produktion geändert wurde. |

### Releases und ihre Versionsbezeichung

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

|  |  |
| --- | --- |
|  | 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. |
|  | **Merge Strategien**  *Speeren «Check-Out»* Die Artefakte werden solange gesperrt, wie sie in Bearbeitung sind. Paralleles Arbeit 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

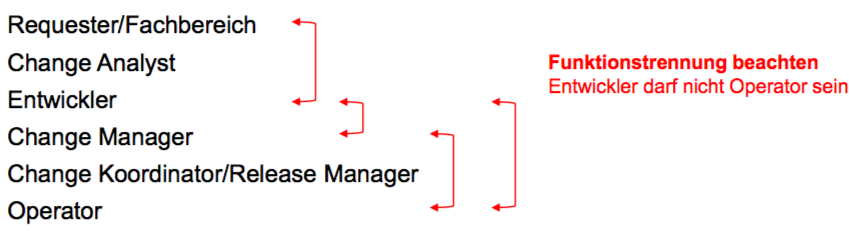
* Verwaltet alle Komponenten an einem Ort, z.B. mit einem virtuellen Dateisystem über mehrere Standorte hinweg
* Änderungen können identifiziert und beurteilt werden
  + Annehmen
  + Verwerfen
* Macht Unterschiede zwischen Versionen sichtbar, mittels Compare-Funktion oder  automatischem Merging
* Wiederherstellung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

### ITIL

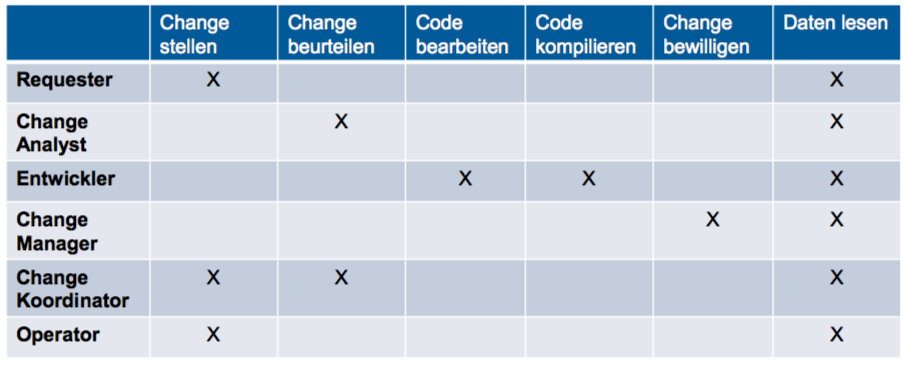
ITIL gibt als Best Practise Framework einen Prozess vor, welcher von Konfigurationsmanagement flankiert wird. Das Service Level Management gibt die Anforderungen, diese werden umgesetzt und durch das Change Management freigegeben, so dass diese vom Release Management eingeplant werden. Das Change Management unter ITIL hat mit dem Change Management aus der Projektwelt nur am Rande zu tun. ITIL beschreibt vor allem Änderungen (auch noch so kleine) im Betrieb und im Projektmanagement sind Changes grössere Abweichungen von Budget, Scope, etc.

### Berechtigungen in SCM

Wenn Sie Konfigurationen und Prozesse in einem Tool verwalten, ist es notwendig die Berechtigungen korrekt zu setzen. Aus Sicht Funktionsbreite modellieren Sie pro Prozessteilnehmer am besten eine Rolle, zum Bespiel:



Aus Sicht Datenbreite unterteilen Sie wenn notwendig (Need-to-Know) noch die verschiedenen Produkte oder Projekte in unterschiedliche Datenräume und hinterlegen Sie jeweils einen Owner. So könnte eine Berechtigungsmatrix aussehen:



Wenn Sie ein Multi-Role System haben, also wenn Personen mehrere Rollen zugewiesen werden dürfen, müssen Sie die Funktionstrennung zusätzlich beachten und dies möglichst ausschliessen.

# Kommunikation und Dokumentation

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Stakeholder * Führung / Ressourcenmanagement * Kommunikation * Projektmarketing * Anschlussdokumentation und laufende Dokumentation | * 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 interessierten Parteien am Projekt. Die Stakeholder bilden verschiedene Anspruchsgruppe 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. Diese werden in diesem Unterkapitel noch genau erläutert.

### Sechs Fragen um Stakeholder zu finden

* Wer arbeitet am Projekt mit oder stellt Material zur Verfügung?
* Wer kann Einfluss nehmen auf das Projekt?
* Wer ist vom Projekt besonders betroffen?
* Wer kann den Projekterfolg beeinflussen?
* Wer ist Nutzniesser des Projekts?
* Wer muss zwingend involviert werden?

### 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 passen auf die Mitglieder des Projektteams
* **CONTRA**: Stakeholder sind Anspruchsgruppen die es zu identifizieren und analysieren gibt, für das Projektteam ist dies nicht notwendig.

|  |  |
| --- | --- |
| **Wer ist alles Stakeholder eines Projekts?** | **Nach der Identifikation kommt die Analyse** |
| **Abwägen von Einfluss und Interessen** | **Klassische Einordnung nach IPMA** |

## Führung / Ressourcenmanagement

«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-Vorhanden» benötigt Arbeitstechnik und Disziplin. Projektführung ist die Anleitung der Arbeitstechnik für eine Gruppe. Heutige IT-Projekte sind keine «Ein-Mann-Vorhaben» mehr und benötigten daher zwingend Führung. Wer selber eine gute Arbeitstechnik hat, beweist Methodenkompetenz und besitzt eine wichtige Eigenschaft eines guten Projektleiters.

|  |  |
| --- | --- |
| Führungsstile nach Tannenbaum | Führungs-(Nahrungs-)kette – Gleichgewicht der Interessen |

### Führungsrhythmus

Der Führungsrhythmus 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 Projektsitzungen, 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 Sie sich nicht mit dem «B-Kader» abspeisen. Planen Sie so grob wie möglich, aber so detailliert wie nötig. Führen Sie Änderungen stetig nach.

## Kommunikation

Kommunikation haben einen sehr grossen Zusammenhang mit dem Stakeholder Management und der Führung. Daher haben gute Projekte ein Kommunikationskonzept. In 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 Projekts 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 Kommunikationswegen zwischen den einzelnen Einheiten dar?
  + Wer sind die Entscheider für ein Projekt?
  + Welches sind die Ansprechpersonen innerhalb der Projektorganisation?

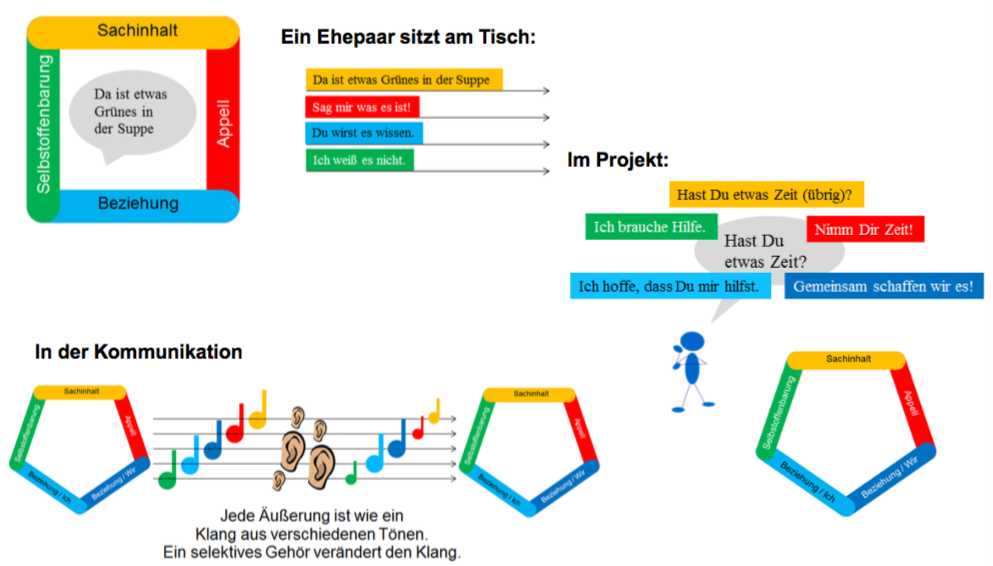
Zudem sollen die wichtigen Termine und Meilensteine aufgelistet werden, die für die Kommunikation im Projekt relevant sind.

|  |  |
| --- | --- |
| **Beispiel einer Komm-Matrix aus der Praxis** | **Sitzungslandkarte** |

### Qualität vor Quantität

|  |  |
| --- | --- |
| **Gute Kommunikation**   * Jeder weiss fundiert was er benötigt und ein wenig darüber hinaus * Sitzungen sind eine Nebensache aber notwendig * Die Stakeholder fühlen sich «abgeholt» | **Schlechte Kommunikation**   * Alle wissen nichts oder jeder weiss alles * Viele Sitzungen behindern das Arbeiten * Relevante Stakeholder sind nicht informiert |

### Kommunikation nach Schulz von Thun



## Projektmarketing

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

|  |  |
| --- | --- |
|  | Beispiele von Marketing-Aktionen |

### 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 sich «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, Gelbgebern sowie anderen Stakeholdern. Bei wichtigen Ereignissen wird erwartet, dass der Projektleiter persönlich teilnimmt.

## Dokumentation

Projekte wie Insieme haben gezeigt, dass Dokumentation etwa arg wichtiges ist.

|  |  |
| --- | --- |
| **Projektdokumentation** | **Beispiele von Dokumenten** |
| Am besten erstellt man zu Beginn des Projekts einen Dokumentenplan, welcher zeigt, welche Dokumente sie wem schuldig sind. | |

# Lösungsfindung und Teamarbeit

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Verschiedene Methoden zur Lösungsfindung * Reviewtechniken * Sitzungstechniken * Verhandlungstechnik | * kennen mehrere Methoden zur Lösungsfindung * verstehen den Ablauf eines Reviews * wenden die Sitzungsplanung an * erweitern ihr Wissen hinsichtlich Verhandlungstechnik |

## Verschiedene Methoden zur Lösungsfindung

Das methodische Vorgehen zur Findung von Lösungen mag auf den ersten Blick als übertrieben scheinen (wie mit Kanonen auf Spatzen schiessen). Es gibt aber durchaus Situationen, wo eine geordnete Findung der einzige Ausweg ist oder die Bewertung nachvollziehbar sein muss. Hier ein Auszug einer Methoden, welche nun noch genauer betrachtet werden.

|  |  |
| --- | --- |
| **Finden** | **Bewerten** |
| * Brainstorming * Methode 635 * Morphologischer Kasten | * Nutzwertanalyse * Entscheidungsbaum |

### Finden | Brainstorming

Die **Zusammensetzung** der Gruppe sollte eine möglichst breite Streuung des Fachwissens haben. Innerhalb der Gruppe dürfen keine zu grossen hierarchische Unterschiede vorhanden sein (sonst Angst etwas zu sagen). **Teilnahme** daran **freiwillig** oder wenigstens nicht **widerwillig**. Das Problem sollte bereits **vorher** **bekanntgegeben** werden. Besonders bei komplexen Problemen etwa zwei Tage vor der Sitzung. Der Sitzungsraum **soll** **vorbereitet** werden, dass eine **Isolierung** von **äusseren** **Einflüssen** vorhanden ist. Hilfsmittel wie Handtafel und Flipchart gelten vorbereitet. Eventuell Verpflegung zur Schaffung einer zwanglosen Atmosphäre. Protokollführer für schriftliche Fixierung der geäusserten Ideen.

#### Regeln

**Für den Diskussionsleiter**

* Er formuliert das Problem nochmals möglichst exakt (paraphrasieren)
* Er überwacht den Ablauf der Sitzung
* Er greift ein, wenn die Regeln nicht beachtet werden oder der Ideenfluss zu versiegen droht

**Für den Protokollführer**

* Er schreibt alle Ideen auf
* Er strukturiert die Ideen und fertigt ein Protokoll für die Entscheidungsinstanz an

**Für alle anderen Teilnehmer**

* Kritik ist verboten
* Der Phantasie soll freiten 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
* Gelegenheit geben innert 24h nochmals darüber nachzudenken

### Finden | Methode 653

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 pro Runde haben sich als geeignet erwiesen:

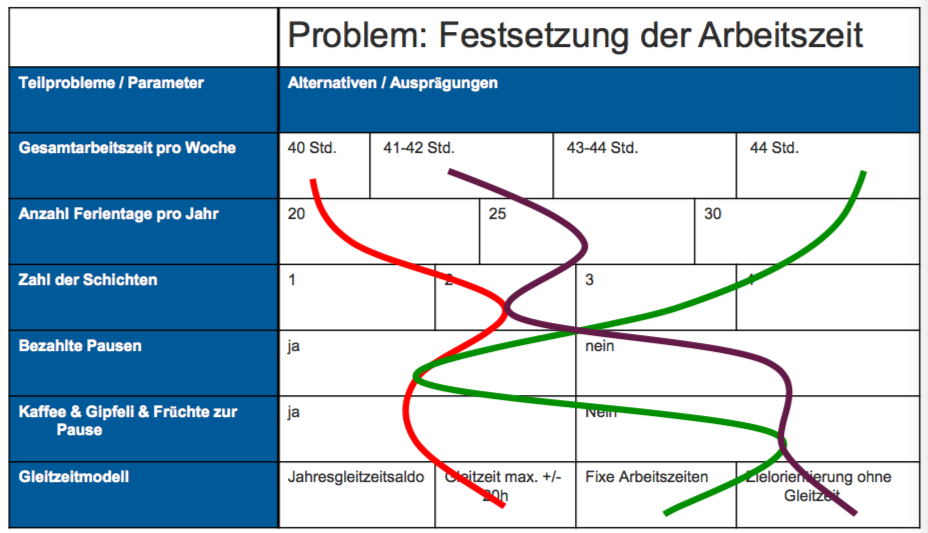
* 1. Runde | 5 min
* 2. Runde | 6 min
* 3. Runde | 7 min
* 4. Runde | 8 min
* 5. Runde | 9 min
* 6. Runde | 10 min

### Finden | Morphologischer Kasten

Die Morphologie wurde vom Schweizer Forscher F. Zwicky entwickelt (er spricht dabei von «diskursiver Problemlösung»). Mit dieser Methode soll durch das systematische Zusammenstellen aller logischen denkbaren Möglichkeiten über einen bestimmen 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 Handlungsfelder
3. Lösungsvarianten pro Teilproblem oder Handlungsfeld ermitteln
4. Kombination herausfinden, zum Beispiel jede Person kriegt einen andersfarbigen Stift.

**Beispiel  
**

### Bewerten | 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 werden benötigt:

1. Kritieren (oder Anforderungen) auflisten
2. Kriterien gewichten, wobei Gesamtgewicht = 1 (oder 100 %) sein muss
3. Jedes Kriterium für jede Variante bewerten
4. Punktzahl zusammenzählen

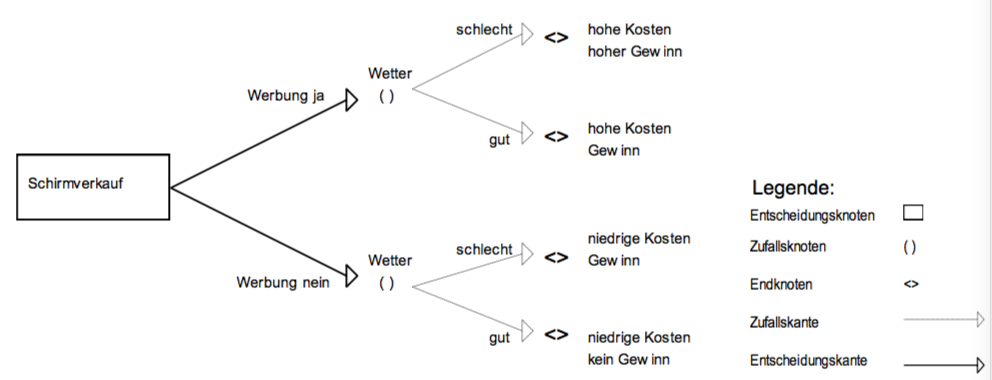
Bei knappen Unterscheidungen empfiehlt es sich die Sensivität der Gewichtung mittels klienen Verschiebungen zu überprüfen.

|  |  |
| --- | --- |
|  |  |

### Bewerten | Entscheidungsbaum

Im Entscheidungsbaum wird ein Problem systematisch dargestellt. Die möglichen Entscheidungen werden mit dem zufälligen Ereignis verknüpft und die möglichen Endzustände mit den entsprechenden Konsequenzen dargestellt. Nimmt man weiter an, dass die zufälligen Ereignisse mit bestimmen Wahrscheinlichkeiten eintreten und ordnet man einzelnen Aktionen und Knoten gewisse Vergleichsgrössen (Nutzen, Aufwand) zu, kann man Entscheidungsbäume für technische Problemlösungen nutzen.

**Beispiel eines Entscheidungsbaums**  
Ein Schrimgeschäft 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).

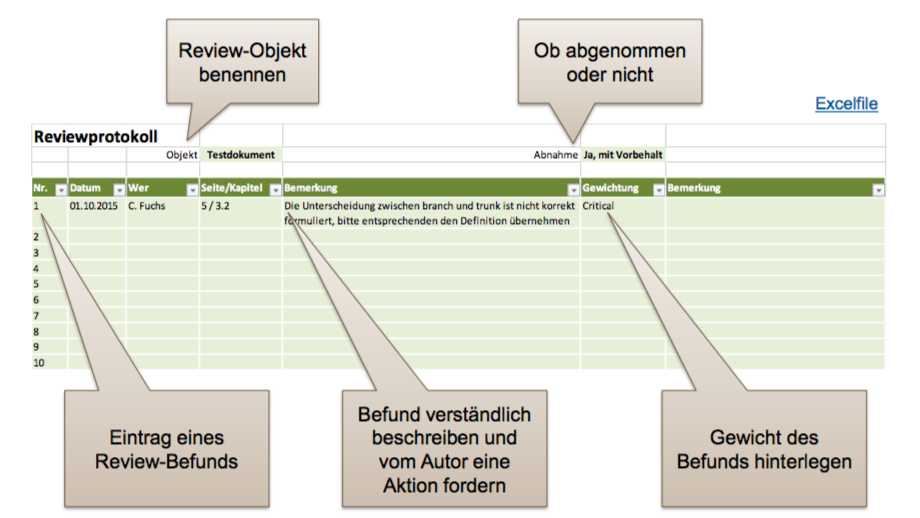


## Reviewtechnik

Ein Review ist ein systematischer Check durch einen Stakeholder oder durch unabhängige Personen. Bespiele von Review-Objekten

* Dokumente
* Code
* Konzepte
* 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 Object wird nach dem Review abgenommen, unter Vorbehalt abgenommen oder zurückgewiesen.

**Beispiel eines Reviewprotokolls**  


## Sitzungstechnik

Der Sitzungsleiter bereitet die Sitzung vor und eröffnet diese, 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 Teilnhemer auch Zeit für die Sitzung?
* Was muss am Ende für jeden heraus kommen?
* Wie sind die Abhängigkeiten zum Umfeld?
* Welche Traktanden und in welcher Reihenfolge werden besprochen?

Frühzeitge Festlegung des Termins und eine persönliche Einladung verstehen mit Traktandenliste und Zielen empfiehlt ist sich sehr.

### 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 Abschweifung 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 einer Sitzung

|  |  |
| --- | --- |
|  |  |
|  |  |

### Führungsrhythmus

Projektteams werden in der Regeln mit Sitzugen geführt. Zum Beispiel ist es in Scrum das Daily Meeting. Sie als Projektleiter definieren die «**Sitzungslandkarte**» (siehe vorherige Kapitel) im Projekt und legen damit einen passenden Rhythmus fest. Ändern Sie diesen bei Bedarf wie ein Dirigent in einem Orchester.

## Verhandlungstechnik

Wir erinnern uns an Schulz von Thun.

### Verhandlungskommunikation

* Sorgen Sie schon im Vorfeld um eine gute Beziehungsebene
* Verhandeln Sie möglichst auf der Sachebene
* Vermeiden Sie Selbstoffenbarungen
  + Definieren Sie die gewünschten Ziele vorgängig und legen Sie die Mindestziele fest, welche Sie aber verdeckt halten.
* Verzichten Sie auf Appelle

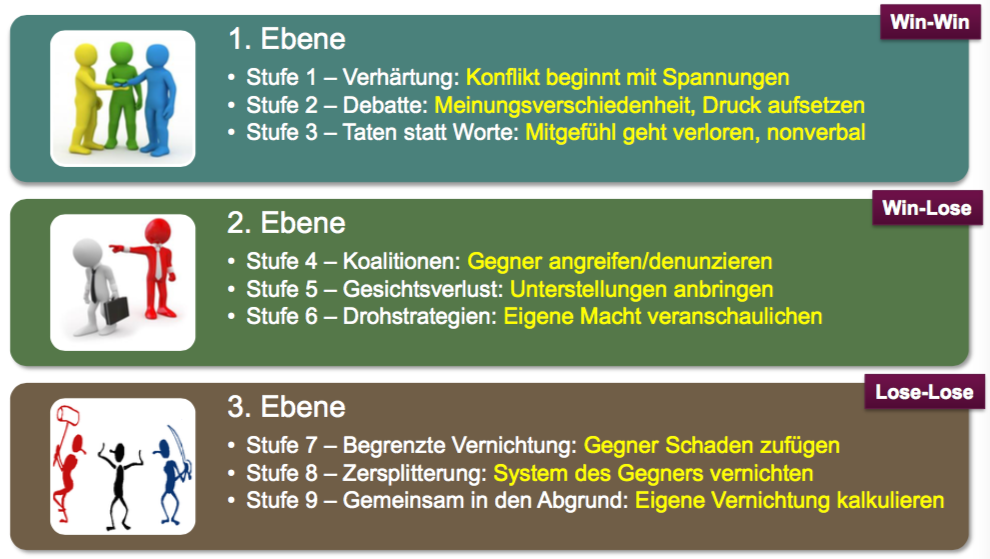
### Die Harvard-Methode | 5 Grundsätze

|  |  |
| --- | --- |
|  | Diskutieren Sie nicht nur, was die Gegenseite möchte, sondern auch, warum Sie es möchte. |
| Versuchen Sie, die Beschränkungen Ihres Gegenübers zu verstehen und Abhilfe zu schaffen. |
| Gehen Sie konstruktiv mit den Forderungen der Gegenseite um. |
| Schaffen Sie Gemeinsamkeiten mit ihren Kontrahenten. |
| Haken Sie nach, auch wenn die Verhandlungen zu scheitern drohen. |

### Kompromiss vs. Konsens

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

### Konflikt-Eskalation nach Friedrich Glasl



# Projekte in der Praxis

## Ziele des Kapitels

|  |  |
| --- | --- |
| **Lerninhalte** | **Ziele: Die Studierenden...** |
| * Projektziele/-abgrenzung * Projektantrag/-auftrag * Projektstruktur (Objekt-,Aktivität, Phasenorientierung) * Gantt-/Netzplan * Phasen / Sprints * Meilensteine * Ressourcenplanung / AKV * Kickoff Veranstaltung * Typische Lieferergebnisse * Einführung, Big Bang | * vertiefen mit Beispielen aus der Praxis ihre Kenntnisse |

## Projektziele/-abgrenzung

|  |  |
| --- | --- |
| **Beispiel Privatwirtschaft** | **Beispiel Bundesauschreibung** |

## Projektantrag/-auftrag

|  |  |
| --- | --- |
| **Projektantrag einer Unternehmung mit Prince2** | **Projektantrag mit Stakeholder besprechen** |

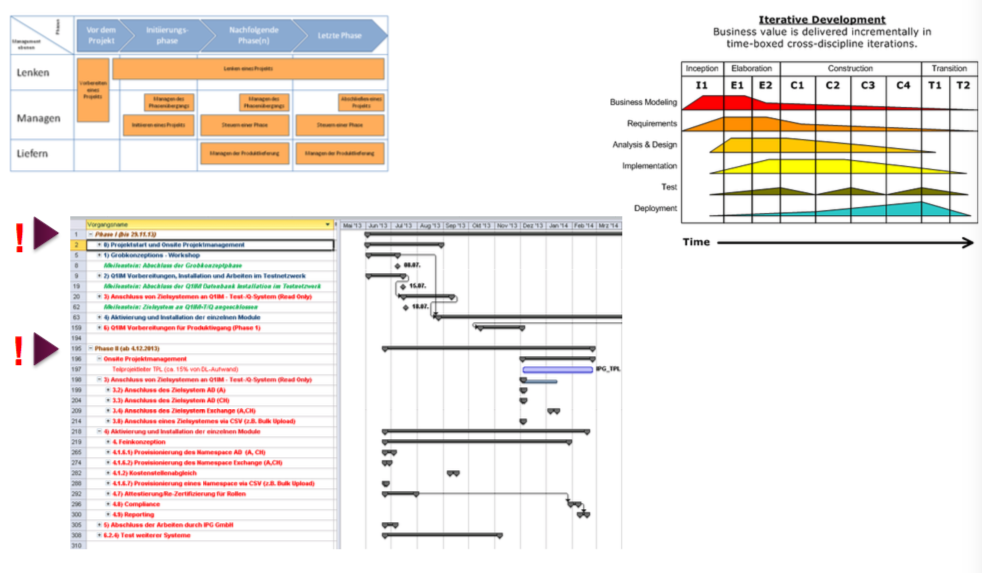
## Projektstruktur (Objekt-,Aktivität, Phasenorientierung)

|  |  |
| --- | --- |
| **Beispiel 1 | Service Management** | **Beispiel 2 | Einführung IAM System** |
| **Beispiel 3 | Erweiterung IAM System** |

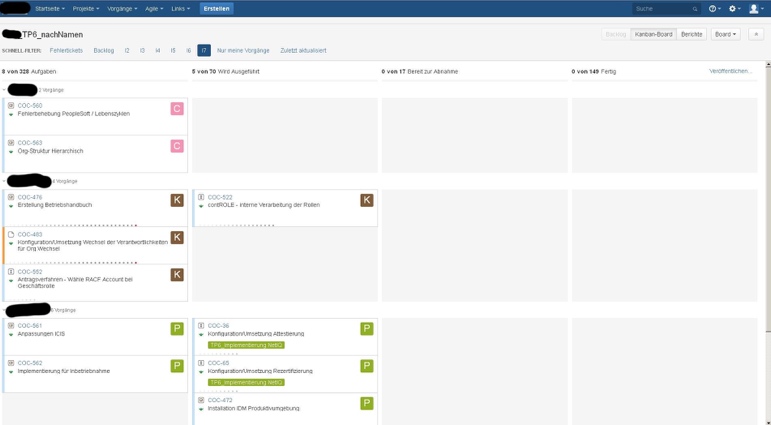
## Gantt-/Netzplan

|  |  |
| --- | --- |
| **Beispiel | Betrachtung «von Weitem»** | **Agile Umsetzung in einem Ganttplan** |

## Phasen / Sprints

Phasen – ein oft missbrauchter Begriff  


**Wasserfall | Häufige Fehler bei der Planung der Phasen**  
Zu vielen parallele Phasen unterwandern den Grundgedanken einer seriellen Abarbeitung und machen die Vorteile hinsichtlich Transparenz zu Nichte (z.B. RUP nicht sehr vorteilhaft, HERMES viel strikter). Die Phasen werden oftmals mit Ausbaustufen verwechselt. Phasen weisen keine Meilensteine auf. Arbeitspakete und Tätigkeiten werden phasenübergreifend definiert, wodurch eine Phase nicht wirklich abgeschlossen werden kann.

Die Bilder zeigen eine nicht ideale Situation. In einer Phase, welche vorüber ist, dürfen keine offenen Arbeitspakete oder Tasks mehr vorhanden sein.

## Meilensteine

## Ressourcenplanung / AKV

## Kickoff Veranstaltung

## Typische Lieferergebnisse

## Einführung, Big Bang

|  |  |
| --- | --- |
| **Beispiel | Betrachtung «von Weitem»** | **Agile Umsetzung in einem Ganttplan** |