

Projekt- und Qualitätsmanagement

Definition

Projekte sind **einmalige, komplexe und zeitlich begrenzte** Vorhaben, zu dessen Realisierung unterschiedliche Ressourcen herangezogen werden müssen.

⇒ Ressourcen sind z.B. Personen, Arbeitsgruppen, etc.

Projektgrösse

Die Projektgrösse wird im Normalfall anhand der **Kosten, der Zeitspanne und den Arbeitsstunden** gemessen.

⇒ Akademisch auch in Form von Function Points.

Projekterfolg

Der Projekterfolg hängt dabei direkt mit der Grösse des Projekts zusammen.

⇒ Grosse Projekte sind i.d.R. nicht Erfolgreich!

Make or Buy

Es ist nicht immer sinnvoll, ein Projekt selbst durchzuführen. Je nach Kosten und Aufwand kann sich auch eine «schlechtere» Standardlösung lohnen.

⇒ Wir nennen das den «Make or Buy» Entscheid.
⇒ z.B. CMR oder eigene Webseite?

OTOBOS

Wir können den Stand eines Projekts mittels OTOBOS beurteilen. Wir stellen uns also die Frage: Ist das Projekt...

- on Time (OT)
- on Budget (OB)
- on Specification / Scope (OS)

Konflikte

Die 3 Aspekte von OTOBOS stehen immer miteinander im Konflikt. Ändern wir einen Aspekt, so beeinflussen wir auch die anderen.

⇒ Ein neues Feature (Scope) braucht mehr Zeit (Time).
⇒ Ein besserer Mitarbeiter (Budget) arbeitet schneller (Time).

Projektantrag

Bevor wir ein Projekt starten können, müssen wir meistens zuerst einen Projektantrag schreiben. Dieser beinhaltet:

1. **Ausgangslage:** Was ist die aktuelle Situation?
2. **Ziele:** Was wollen wir erreichen?
3. **Motivation:** Warum wollen wir es erreichen?
4. **Ressourcen:** Welche Ressourcen stehen uns zur Verfügung?
5. **Termine:** Welche Zeitvorgaben, Meilensteine, etc. haben wir?
6. **Risiken:** Welche Risiken hat das Projekt?

⇒ Denke dabei an KISS: «Keep it short and simple.»

Innovation

Wir bezeichnen ein Projekt als innovativ, wenn es Fortschritte in der **Technologie** und der **Organisation** erzielt.

Projektkontrolle

Definition

Unter «Controlling» in einem Projekt verstehen wir mehrere Tätigkeiten:

1. Planung
2. Kontrolle & Abweichungsanalyse
3. Informierung & Berichtswesen
4. Steuerung & Koordination

Grundsätzlich geht es darum, den Projektstand zu **ermitteln**, diesen zu **kommunizieren** und allfällige **Änderungen** am Projekt **vornehmen**.

⇒ «Controlling» ist also mehr als nur «kontrollieren».

Wer kontrolliert die Projekte?

Schlussendlich dient das «Controlling» besonders den Entscheidungsträgern in einem Projekt. Diese sind:

- Der **Lenkungsausschuss**, also die Auftraggeber und Kunden (Soll).
- Die **internen Mitarbeiter** wie Projekt-Controller, Audit und Portfolio- und Programm-Manager (Kann).

Einschub: Portfolio & Programm

In den meisten Unternehmen gehören Projekte immer einem Programm und darüber einem Portfolio an.

⇒ Portfolio: Alle Projekte, die ein Unternehmen ausmachen.
⇒ Programm: Zusammenhängende Projekte, die eine Teilmenge des Portfolios bilden.

Kontrolle & Abweichungsanalyse

Ausgangslage

In einem ersten Schritt müssen wir den aktuellen Projektstand ermitteln. Das bedeutet, wir müssen den **Projektfortschritt** irgendwie **messen**.

Methoden

Leider ist es faktisch kaum möglich, den exakten Projektfortschritt zu ermitteln. Wir können aber:

- Das **Produkt** betrachten und dessen Fertigungsgrad bestimmen.
- Die **Entwickler** fragen, wie viel Zeit sie noch benötigen.

⇒ Unschärfe ist dabei vorprogrammiert.

Messwerte

Um nun den Projektfortschritt bestimmen zu können, messen wir in bestimmten Abständen verschiedene Werte.

⇒ Wir können so den Projektstand als Trend abbilden.

1. Zeit, Kosten, Leistung

Gemäss OTOBOS messen wir mindestens die verbrauchte Zeit, die aktuellen Kosten sowie die erbrachte Leistung.

2. Earned-Value-Analyse (EVA)

Die EVA ist die bekannteste Messgrösse für den Projektfortschritt. Sie bestimmt den **Fertigstellungswert** eines Projekts, woraus dann die **Kosteneffizienz** abgeleitet werden kann. Die EVA beinhaltet:

- **Planned Cost (PC)**
- **Actual Cost (AC)**
- **Earned Value (EV)**
- **Cost Variance (CV)**
- **Cost Performance Index (CPI)**

$$CV = EV - AC \qquad CPI = \frac{EV}{AC}$$

⇒ Wobei EV = Fertigstellungswert, CPI = Kosteneffizienz
⇒ Kosteneffizienz: Verhältnis der Kosten zur erbrachten Leistung
⇒ Wir streben immer eine Kosteneffizienz > 1 an.

Es gibt 3 Berechnungsmethoden:

1. **Strikt:** Alle **vollständig abgeschlossenen** Komponenten werden beachtet.
 $EV = K_1 + K_2 + K_3 + \dots$
2. **Zwischenresultate:** Alle **brauchbaren** Komponenten werden beachtet.
 $EV = K_{1.1} + K_{1.3} + K_{2.2} + \dots$
3. **Restaufwand:** Die Berechnung erfolgt über die Schätzung des Restaufwands.
 $EV = \frac{PC}{AC + Rest} \cdot AC$

⇒ «Zwischenresultate» sind z.B. Module einer Software.
⇒ Bei «Strikt» muss die gesamte Software fertig sein.

3. Meilenstein-Trend-Analyse (MTA)

Bei der MTA werden die Deadlines der Projektmeilensteine rückwirkend analysiert. Somit zeigt diese Analyse die **Verschiebungen der Meilensteine** über das Projekt hinweg auf.

⇒ Optimal sind keine Verschiebungen (horizontale Linien).
⇒ Diese Analyse zeigt eindrucksvoll den Projektverlauf.

Gutes Projekt

Schlechtes Projekt

Weiteres

Im Zusammenhang mit dem Projektstand beachtet man auch oft:

- Risiken und Chancen
- Aktuelle Issues
- Restaufwandschätzung
- Kommentare

⇒ Meistens bestimmt das Unternehmen den Inhalt.

Informierung & Berichtswesen

Ausgangslage

Die meisten Projekte scheitern aufgrund **ungenügender Kommunikation**. Um das zu verhindern, benötigen wir im «Controlling» ein robustes Berichtswesen.

Kommunikationsvarianten

Die beste Methode für das Erreichen einer guten Kommunikation ist das Umwandeln der ermittelten Werte über den Projektstand in «einfache» Metriken.

⇒ Dies vereinfacht insbesondere die Kundenkommunikation.

1. Definition of Done

Die einfachste Variante ist die Einteilung des Arbeitsfortschritts in einfache Kategorien. Wann etwas «fertig» ist, bestimmen wir dabei selbst.

⇒ z.B. 0% nicht begonnen, 30% in Arbeit, 80% fertig.

2. Ampel-Prinzip

Beim Ampel-Prinzip drücken wir den Projektstand in Form einer Ampel aus. Dies hilft, die aktuelle Situation transparent und klar zu kommunizieren.

⇒ Jedes Unternehmen hat dabei eine eigene Farbdefinition.
⇒ Der Projektleiter muss somit klare Stellung nehmen.

3. Aggregiertes Ampel-Prinzip

In Bezug auf OTOBOS können wir auch mehrere Ampeln anhand des maximum Prinzips aggregieren.

⇒ Maximum Prinzip: Ist etwas Orange, ist alles Orange.

4. Cockpit

Ein Projekt Cockpit ist eine Sammlung von verschiedenen Messwerten und Analysen. Es erlaubt uns, den Projektstand schnell zu ermitteln und zu beeinflussen.

Steuerung & Koordination

Change Management

Kein Projekt wird so durchgeführt, wie es ursprünglich geplant wurde. Um mit Änderungen umzugehen, brauchen wir ein klares «Change Management».

⇒ Projektplanung bedeutet nicht, die Zukunft vorherzusagen.
⇒ Bei agilen Projekten ist dieses Thema nicht relevant.

Vorgehen bei Abweichungen

Bei klassischen Projektmethoden müssen wir bei Abweichungen vom Plan irgendwie handeln. Wir können z.B.:

- Die Vorgehensweise ändern
- Überzeiten anordnen
- Coaching & Unterstützung anfordern

Wenn diese Massnahmen keine Verbesserungen bringen, müssen wir einen «Change Request» beantragen.

⇒ Die Vorgehensweise ändern heisst z.B. serielle Tätigkeiten in parallele umzuwandeln.

Change Requests

Ein «Change Request» ist eine Anfrage beim Kunden, gewisse Aspekte des Projekts abzuändern. Change Requests müssen immer begründet sein.

⇒ Meistens ändern wir Aspekte in Bezug auf OTOBOS.
⇒ z.B.: Weniger Inhalt, damit das Geld reicht (Scope)
⇒ Oftmals ist mehr Budget besser als weniger Scope.

Kategorien

Streng genommen gibt es genau 3 Kategorien von Change Requests: **Scope, Budget und Time**. In der Praxis sind aber auch weitere Kategorien anzutreffen:

- **Legal Terms:** z.B. Vertragsort
- **Ressourcen:** z.B. Hersteller

Voraussetzungen

Um einen Change Request zu stellen, müssen zuerst einige Voraussetzungen erfüllt sein:

1. Ein Change-Prozess ist definiert (Wie und an wen muss ich den Request stellen).
2. Die Änderung ist fassbar und allen bewusst.

3. Die technischen und organisatorischen Voraussetzungen sind erfüllt, wie z.B.:

- Der Change ist machbar
- Die Ressourcen sind vorhanden
- Die Termine sind realistisch
- etc.

Beschreibung eines Changes

In einem Change Request muss sinngemäss die gewünschte Änderung beschrieben sein. Die Beschreibung soll dabei:

- Kurz und prägnant sein.
- Sich an die Zielgruppe orientieren.
- Keinen Entscheid erpressen.
- Wahlfreiheit suggerieren.
- Die eingetragenen Risiken erwähnen.
- Wörter verwenden wie: «Ermöglichen, ausserordentlich, neue Rahmenbedingungen»
- Wörter vermeiden wie: «muss, darf nicht sein, keine Ahnung, aus heiterem Himmel»

Auswirkungen und Risiken

Die Auswirkungen und Risiken eines Changes sollten in einem separaten Kapitel beschrieben werden. Dabei sollten u.a. auch diese Punkte erwähnt werden:

- Auswirkungen bei einer Ablehnung
- Neue Risiken bei einer Änderung

Quantifizierung

Mit einem Change Request will man oftmals mehr Budget oder Zeit für ein Projekt anfordern. Diese Werte müssen im Request sinnvoll quantifiziert sein.

⇒ z.B. «Wir benötigen 3 Monate mehr, damit wir...».

⇒ Quantifiziere so hoch wie möglich und so tief wie nötig.

⇒ Bei neuen Anforderungen immer Budget und Zeit erhöhen.

Vorgehen

Änderungswunsch

Änderungswunsch formulieren

Grobe Aufwandschätzung durchführen

Change Request nötig?

Nein

Änderungswunsch dokumentieren

Änderungs-wunsch

Ja

Aktualisierung der Projektplanung und Aktivitäten zuordnen

Abstimmung mit Steering Committee (mind. Auftraggeber und Fachcoach)

Genehmigung?

Nein

Ablehnung kommunizieren

Ja

Change Request dokumentieren

Änderungs-wunsch

Change Request realisieren

Änderungswunsch ausgeführt

Projektmanagement (Klassisch)

Vorgehensmodelle

Vorgehensmodelle beschreiben die Art und Weise, wie ein Projekt realisiert wird. Wir unterscheiden zwischen agilen und klassischen Modellen.

⇒ Das Modell wird zu Beginn des Projekts festgelegt.

⇒ Änderungen daran sind mit grossem Aufwand verbunden.

Agile Modelle

Agile Modelle basieren auf iterativen Arbeitsschritten und schlanken Prozessen. Sie benötigen insbesondere ein gutes Team Management.

⇒ Grundsätzlich geringer administrativen Aufwand.

Klassische Modelle

Klassische Modelle sind formeller und strikter, wodurch sie klare Aussagen zum Fortschritt und Inhalt eines Projekts zulassen. Sie benötigen ein gutes Change Management.

⇒ Grundsätzlich hoher administrativen Aufwand.

Wasserfallmodell

Beim Wasserfallmodell werden sequenziell mehrere sogenannten Phasen durchlaufen. Das Vorgehensmodell hat dabei folgende Eigenschaften:

- Phasen sind aufbauend, d.h. der Output einer Phase ist Input der nächsten.
- Phasen müssen immer vollständig abgeschlossen werden.
- Das Zurückkehren in eine abgeschlossene Phase ist nicht erlaubt.
- Die Parallelisierung der Phasen soll vermieden werden.

Das Wasserfallmodell erlaubt eine klare und transparente Aussage über den Projektfortschritt.

⇒ Das Modell ist aber enorm statisch.

⇒ Daher gilt: Je später eine Änderung, desto teurer.

Definition von «Phase»

Eine Phase beschreibt eine Sammlung von mehreren Arbeitspaketen, welche zeitlich und inhaltlich zusammenhängen. Phasen werden mit Meilensteinen terminiert und sollen ein Projekt in logische Abschnitte unterteilen.

⇒ Die Arbeitspakete sind dabei alle Aufgaben / Tasks.

Konzeption

Entwicklung

Einführung

⇒ Phasenübergänge werden z.T. auch «Gates» genannt.

⇒ Eine Parallelisierung von Phasen ist nicht vorgesehen.

Rational Unified Process (RUP)

Beim RUP-Modell werden sequenziell 4 Phasen durchlaufen. Anders als beim Wasserfallmodell können fachliche Arbeiten (sogenannte «Disciplines») auch über mehrere Phasen verteilt sein.

Disciplines

Business Modeling

Requirements

Analysis & Design

Implementation

Test

Deployment

Configuration & Change Mgmt

Project Management

Environment

Phases

Inception

Elaboration

Construction

Transition

Iterations

Initial

Elab #1

Elab #2

Const #1

Const #2

Const #N

Tran #1

Tran #2

Phasen vom RUP-Modell

Die 4 Phasen vom RUP-Modell sind:

1. Inception: Konzeption & Planung
2. Elaboration: Design & Prototyp
3. Construction: Entwicklung & Tests
4. Transition: Übergabe & Auslieferung

⇒ RUP endet immer mit dem «Product Release Milestone».

HERMES

HERMES ist ein strikt definiertes Vorgehensmodell des Bundes. Das Modell besteht aus exakt 4 Phasen:

1. Initialisierung
2. Konzept
3. Realisierung
4. Einführung

Alle Phasenübergänge müssen dabei zwingend vom Lenkungsausschuss freigegeben werden.

⇒ Unabhängiges Arbeiten also nur innerhalb einer Phase.

⇒ HERMES: «Handbuch der Elektronischen Rechenzentren des Bundes, eine Methode zur Entwicklung von Systemen».

Projekt-Initialisierungs-auftrag

Projekt-freigabe

Phasen-freigabe

Phasen-freigabe

Projekt-abschluss

Initialisierung

Konzept

Realisierung

Einführung

Szenarien & Module

Bei HERMES gibt es 8 Standardszenarien für Projekte. Ein Szenario besteht dabei aus mehreren Modulen, welche wiederum aus Aufgaben, Rollen und Ergebnissen bestehen.

Szenarien

Module

Ergebnisse

Aufgaben

Rollen

Module

Szenarien

Partner & Hierarchien

Weiter werden bei HERMES alle «Rollen» in eine Hierarchie und eine oder mehrere Partner unterteilt. Es gibt genau 3 Hierarchien und 3 Partner.

Partner

Anwender

Ersteller

Betreiber

Hierarchie

Steuerung

Führung

Ausführung

⇒ z.B. Entwickler: Hierarchie «Ausführung», Partner «Ersteller»

⇒ z.B. Kunde: Hierarchie «Steuerung», Partner «Anwender»

⇒ Alle Rollen sind dabei strikt definiert.

Vor- und Nachteile

Vorteile:

- Hohe Standardisierung
- Viele Tools und Vorlagen
- Zertifizierung möglich
- Einbettung von Scrum möglich
- Passt für öffentliche Ausschreiben

Nachteile:

- Sehr starke Vorgaben
- Vier Phasen sind etwas knapp
- In der Privatwirtschaft kaum relevant
- Im Ausland gar nicht relevant
- Kann Projekte verkomplizieren

CYNEFIN Framework

Das CYNEFIN Framework soll bei der Wahl eines Vorgehensmodells (agil vs. klassisch) helfen. Dazu unterteilt es Projekte in 4 Kategorien:

1. Simple: Die Lösung ist offensichtlich.
2. Complicated: Die Lösung benötigt eine Fachanalyse (stabile Herausforderungen).
3. Complex: Die Lösung muss neu erarbeitet werden (dynamische Herausforderungen).
4. Chaotic: Das Problem hat keine Lösung.

Simple und Complicated sollen dabei klassisch und Complex agil gelöst werden. Chaotic lässt sich nicht lösen.

⇒ Stabil: Ich kenne die Herausforderungen zu Beginn.

⇒ Dynamisch: Die Herausforderungen sind noch unbekannt.

CYNEFIN FRAMEWORK

COMPLEX

COMPLICATED

DISORDER

CHAOTIC

SIMPLE

Einordnung

Umfang

HERMES

Prinze2

RUP

V-Modell

PM/PM/MA

Kanban

Scrum

Vorgehen (Klassisch vs. Agil)

⇒ Grün: Standards, Rot: Vorgehensmodelle

Finanzen

Gewinne & Kosten

Der Gewinn G eines Projektes berechnet sich aus Projektertrag und Kosten.

$$G_{\text{Projekt}} = \text{Ertrag} - \text{Kosten}$$

Weiter lässt sich der Gesamtgewinn eines Unternehmens mittels der Summe der einzelnen Projektgewinne bestimmen:

$$G_{\text{Gesamt}} = G_1 + G_2 + G_3 + \dots$$

⇒ Das Ziel ist meistens ein Gewinn von 10% des Ertrags.

⇒ Dazu kommen 10% Projektreserve, also insgesamt 20%.

Kostenarten

Wir unterscheiden 2 Kostenarten:

1. Variable Kosten V: Entstehen während eines Projektes (z.B. Spesen, Material, ...)
2. Fixe Kosten F: Entstehen fortlaufend, unabhängig davon, ob Projekte laufen oder nicht (z.B. Lohnkosten, Miete, ...)

Deckungsbeitrag

Die Kosten eines Projekts stammen aus vielen verschiedenen Quellen:

1. Projektspesen V: Reisen, Übernachtungen
2. Lohnkosten V F: Gehälter, AHV
3. Indirekte Kosten F: Telefonie, Lizenzen
4. Verwaltungskosten F: Miete, Reinigung
5. Auslastung und Gewinn F: Abschreibungen, Zinsen, Steuern

Ofmals wird anhand dieser Punkte eine Deckungsbeitragsrechnung erstellt. Diese soll aufzeigen, wie sich die Kosten auf den Gewinn auswirken.

⇒ Die Kosten können sich je nach Land stark unterscheiden.

⇒ z.B. Sind die Lohnkosten in der Schweiz sehr hoch.

Ertrag

Projektspesen

Deckungsbeitrag 1

Lohnkosten

Deckungsbeitrag 2

Indirekte Kosten

Deckungsbeitrag 3

Verwaltungskosten

Deckungsbeitrag 4

Auslastung

Deckungsbeitrag 5 / Gewinn

\$100,000.00

~~\$10,000.00~~

\$90,000.00

~~\$30,000.00~~

\$60,000.00

~~\$5,000.00~~

\$55,000.00

~~\$15,000.00~~

\$40,000.00

~~\$20,000.00~~

\$20,000.00

20%

90%

60%

55%

40%

20%

⇒ Da Gewinn $\geq 20\%$ war das Projekt erfolgreich.

⇒ Als Faustregel gilt: Ertrag = 2.5 · Lohnkosten.

Erfolgsfaktoren

Wichtig für den finanziellen Erfolg sind:

- Effektivität: Das Richtige tun
- Effizienz: Es richtig tun

Nutzen

Nutzbetrachtung

Jedes Projekt braucht einen Nutzen, welcher zu Beginn des Projektes auch festgelegt und akzeptiert wurde. Wir unterscheiden zwischen:

- Quantitativer Nutzen Quant (materiell)
- Qualitativer Nutzen Qual (immateriell)

Nutzelemente

Es gibt 4 Nutzelemente:

- 1. **Finanzen** **Quant**: Mehr Einnahmen, weniger Ausgaben
- 2. **Compliance** **Qual**: Erfüllung von Gesetzen, Einhaltung von Normen
- 3. **Agilität** **Qual**: Schnellere Prozesse, flexiblere Strukturen
- 4. **Qualität** **Qual**: Image, Zuverlässigkeit, Daten-hygiene

Return on Investment (ROI)

Beim ROI geht es um die Frage, wann eine Investition wieder zurückgeholt werden kann. Man unterscheidet dabei bei den Projektarten zwischen:

- **Neuanschaffung**: Einmalige Kosten mit jährlichen Betriebskosten (zukünftiger Nutzen relevant).
- **Ersatzinvestition**: Einmalige Kosten mit neuen/alten Betriebskosten (Unterschied der neuen/alten Nutzen relevant).

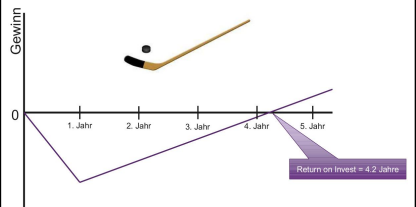
Weiter unterscheidet man die Finanzierung von einem Projekt:

- **Fremdfinanzierung**: Betrachtung der Zins- und Rückzahlungskosten
- **Eigenfinanzierung**: Betrachtung der Diskontierung (Zins & Risiko)

=> Der ROI ist vor allem für den Auftraggeber relevant.

Berechnung

Beim ROI werden die Kosten und Erträge eines Projekts über die Zeit hinweg ausgerechnet. Das **Resultat** ist die Zeit, die es dauert, bis wir einen Gewinn erzielen.



Beispiel: Neuanschaffung

	1. Jahr	2. Jahr
Entwicklungskosten	\$80'000.00	
Betriebskosten	\$0.00	\$10'000.00
Zinskosten für Kapital	6.0%	\$4'800.00
Total Kosten	\$80'000.00	\$14'800.00
Kumuliert	\$80'000.00	\$94'800.00
Productertrag	\$0.00	\$40'000.00
Total Ertrag	\$0.00	\$40'000.00
Kumuliert	\$0.00	\$40'000.00
Kontostand CH Bank		\$25'200.00
Zins	0.50%	\$126.00
Gewinn/ Verlust	-\$80'000.00	-\$54'674.00

Beispiel: Ersatzinvestition

	1. Jahr	2. Jahr
Entwicklungskosten	CHF 29'000.00	
Betriebskosten	CHF 1'250.00	CHF 1'000.00
Zinskosten für Kapital	4.3%	CHF 1'247.00
Total Kosten	CHF 31'497.00	CHF 2'247.00
Kumuliert	CHF 31'497.00	CHF 33'744.00
Productertrag	CHF 12'000.00	CHF 12'000.00
Total Ertrag	CHF 12'000.00	CHF 12'000.00
Kumuliert	CHF 12'000.00	CHF 24'000.00
Gewinn/ Verlust	CHF -19'497.00	CHF -9'744.00

Diskontierung

Der Diskontierungssatz beschreibt einen internen Zinssatz in Kombination mit einem Risikozuschlag. Das soll miteinbeziehen, dass Gewinne in der Zukunft weniger Wert haben wie Gewinne heute.

$$G_{\text{Mit Disk.}} = G_i \cdot (1 - \text{Diskontierung})^i$$

=> Wobei i das aktuelle Betrachtungsjahr darstellt.
=> Wird meistens bei Eigenfinanzierungen verwendet.

Beispiel: Diskontierung

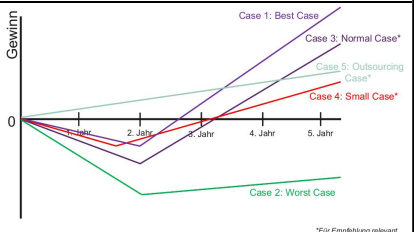
	1. Jahr	2. Jahr
Entwicklungskosten	CHF 945'000.00	
Betriebskosten	CHF 350'000.00	CHF 200'000.00
Total Kosten	CHF 1'295'000.00	CHF 200'000.00
Gewinnsteigerung	CHF 300'000.00	CHF 600'000.00
Total Ertrag	CHF 300'000.00	CHF 600'000.00
Jahresrechnung (FCF)	CHF -995'000.00	CHF 400'000.00
Diskontierungsfaktor	4.50%	1
Diskont (DCF)	CHF -995'000.00	CHF 382'000.00
Gewinn/Verlust (CCF)	CHF -995'000.00	CHF -613'000.00

Business Case

In einem Business Case werden mehrere Geschäftsfälle und Varianten (Best Case, Worst Case, etc.) miteinander verglichen. Ein Business Case beinhaltet:

- Management Summary
- Abgrenzungen
- Kosten pro Case
- Nutzen (Quant. und Qual.) pro Case
- ROI-Berechnung pro Case
- Risiken und Chancen
- Empfehlungen

=> Anschließend wird eine Variante umgesetzt.
=> Der Business Case dient als Entscheidungsgrundlage.



Finanzierung & Liquidität

Finanzierung

Jedes Projekt muss finanziert werden, d.h. die finanziellen Mittel müssen zur richtigen Zeit am richtigen Ort vorhanden sein. Wir unterscheiden bei der Finanzierung zwischen:

- **Interne Mittel** des Unternehmens
- **Externe Mittel** in Form des Marktertrags (z.B. Teilzahlung durch Kunden)

Projekte, die mittels externen Mitteln finanziert werden, sind **Hochrisiko-Projekte** (Konkursrisiko).

=> D.h.: Diese Projekte unterliegen einer schärferen Kontrolle.
=> Das Risiko hängt von der Liquidität des Unternehmens ab.

Risikomanagement

Ausgangslage

Risiken und Chancen sind Unsicherheitsfaktoren in einem Projekt, welche sich positiv oder negativ auf das Ergebnis auswirken können. Die Risikoanalyse ist ein wichtiger Bestandteil des Projektmanagements.

=> Die Chancenganalyse ist seltener, aber hilfreich.
=> Ein eingetretenes Risiko nennt man auch «Issue».

Bestandteile von Risiken

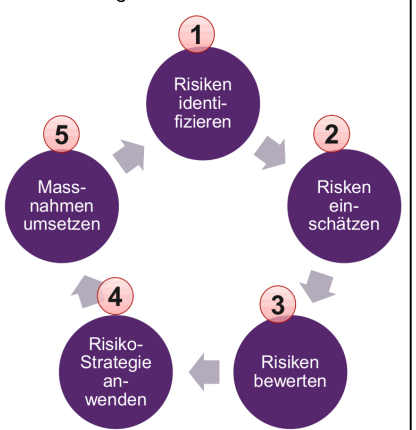
Ein Risiko besteht aus 2 Faktoren:

- Eintrittswahrscheinlichkeit
- Schadensausmass

=> Diese Faktoren dienen u.a. der Priorisierung von Risiken.

Risikomanagement

Das Risikomanagement selbst kann in 5 Schritte aufgeteilt werden.



1. Identifizieren

Für die Risikoidentifikation werden in der Praxis meistens diese 4 Analysen angewendet:

- Impact Analyse
- Bedrohungsanalyse
- Schwachstellenanalyse
- Beliebige Kombinationen

Die Kombination von allen 3 Methoden entspricht einer **vollständigen** Analyse.

Impact Analyse

Bei der Impact Analyse werden die kritischen Geschäftsprozesse und Infrastrukturen eines Projekts analysiert und deren Störungen als Risiko formuliert.

Bedrohungsanalyse

Bei der Bedrohungsanalyse werden möglichen Bedrohungen aus einem **Bedrohungskatalog** analysiert und die davon relevanten als Risiko formuliert.

Schwachstellenanalyse

Bei der Schwachstellenanalyse werden anhand ähnlicher Projekte die grössten Schwachstellen identifiziert und dann als Risiko formuliert.

2. Einschätzen

Jedes Risiko muss entweder **qualitativ** (Hoch, Mittel, Tief) oder **quantitativ** (Zeit, Kosten) eingeschätzt werden. Es gibt 3 Arten, um dies zu tun:

- Schätzung der maximalen Werte
- Schätzung der mittleren Werte
- Schätzung via statistischer Verteilfunktion

Im Normalfall wird zuerst die Eintrittswahrscheinlichkeit und dann das Schadensausmass geschätzt.

=> Dies kann «Bottom-Up» oder «Top-Down» gemacht werden.
=> Bottom-Up: Zuerst Detailanalyse, dann Übersicht.
=> Top-Down: Zuerst Übersicht, dann Detailanalyse.

Beispiel «Serverausfall»

Eintrittswahrscheinlichkeit	Ø in Monate
R1 Ausfall Harddisk	36
R2 Ausfall Stromversorgung	36
R3 Unbeabsichtigtes Herunterfahren	36
R1-R3: Serverausfall alle 12 Monate.	
Berechnung: In 36 Monaten können 3 Ausfälle auftreten. D.h. Ø 12 = 36 / 3	
Schadensausmass	CHF
R1 Ausfall Harddisk	5000
R2 Ausfall Stromversorgung	2000
R3 Unbeabsichtigtes Herunterfahren	500
R1-R3: Ein Ausfall kostet 2'500 CHF.	
Berechnung: Durchschnittsrechnung der Werte.	
=> Wir verwenden meist den MTBF (Mean Time between Failure).	

3. Bewertung

Bei der Risikobewertung werden Risiken in einen Kontext gebracht und wenn möglich **terminlich fixiert oder priorisiert**. Man verwendet dazu 2 Methoden.

1. Risikoliste

Die Risikoliste ist eine Sammlung von allen identifizierten Risiken. Sie beinhaltet meistens:

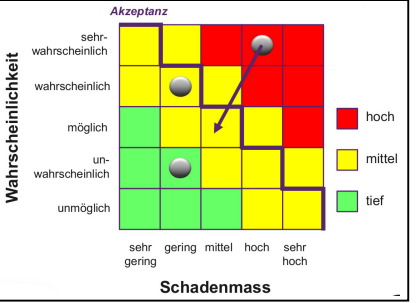
- Risiko (mit ID)
- Massnahmen
- Kosten der Massnahmen
- Schadensausmass (Kosten)
- Eintrittswahrscheinlichkeit
- Gewichteter Schaden
- Priorität / Termin

=> Dies ist wie eine «Lebensversicherung».

2. Risikograph / Risikomatrix

Anhand der Risikoliste können die Risiken nun in einen **Risikograph** eingetragen werden. Die meisten Unternehmen definieren dabei eine sogenannte **Akzeptanzlinie**, unter welcher alle Risiken liegen müssen.

=> Risiken darüber müssen entsprechend reduziert werden.



Beispiele von Risiken

Bekannte Risiken sind:

- Personelle Defizite (Quantität & Qualität)
- Unrealistische Termine und Kosten
- Entwicklung von falschen Funktionen
- Entwicklung der falschen Schnittstellen
- Vergolden vom Projekt
- Ständige Anforderungsänderungen
- Defizite bei externen Komponenten
- Defizite bei externen Aufgaben
- Defizite in der Echtzeitleistung
- Überfordern der Softwaretechnik

=> In den Vorlesungsunterlagen finden sich noch mehr.
=> Die «20 Fehler nach Lindecker» sind sehr ähnlich.

4. Bewältigung

Um ein Risiko zu bewältigen, können wir eine oder mehrere der folgenden 4 Methoden anwenden:

1. **Vermeiden**: z.B. eine risikoreiche Methodik durch eine andere ersetzen.
2. **Vermindern**: z.B. Schulungen in einem bestimmten Bereich durchführen.
3. **Überwälzen**: z.B. eine entsprechende Versicherung abschliessen.
4. **Selbst tragen**: z.B. den Schaden beim Eintritt akzeptieren und abzahlen.

=> Überwälzen bedeutet, das Risiko auf andere zu übertragen.
=> Merke: Es bleibt immer ein Restrisiko vorhanden!

5. Massnahmen

Grundsätzlich gilt: Eine Massnahme soll nicht teurer sein, als das Risiko selbst. **Wenige aber griffige** Massnahmen sind dabei vom Vorteil. Einige Taktiken sind:

- Risiken mit hoher Eintrittswahrscheinlichkeit vermeiden.
- Risiken mit hohem Schadensausmass vermindern.
- Seltene Risiken überwälzen.
- Für alle anderen Risiken eine zeitliche / finanzielle Projektreserve bereitstellen.

=> Risiken ohne gute Massnahmen sollen akzeptiert werden.
=> Es lassen sich auch Eintrittsmassnahmen definieren.

Beispiele von Massnahmen

Einige Massnahmen sind:

- Gegen **Personal**ausfall: Jeder Projektmitarbeitende hat einen Stellvertreter und informiert diesen wöchentlich über den aktuellen Stand.
- Gegen **schlechte Datenqualität**: Zusätzliche Tests überprüfen systematisch die Inhalte der Datenbank.
- Gegen **unklare Anforderungen**: Zusätzliche Meetings mit dem Kunden werden ange-setzt.

Chancen

Chancen sind das **Gegenteil** von Risiken. Es lohnt sich, mögliche Chancen zu identifizieren und deren Eintrittswahrscheinlichkeit und Nutzen zu bestimmen. Typische Chancen sind:

- Neue Hardware-Generationen machen Tuning überflüssig.
- Bestimmte Klassen werden auch von anderen Kunden benötigt.
- Benötigte Hardware trifft früher ein.

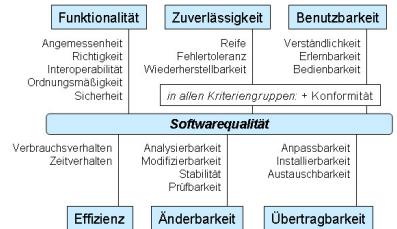
=> Man kann dies auch in eine SWOT-Analyse verpacken.

Ausgangslage

⇒ Der Begriff stammt aus Softwarequalität

1. Funktionalität
2. Zuverlässigkeit
3. Benutzbarkeit
4. Effizienz
5. Änderbarkeit
6. Übertragbarkeit

Qualitätsmerkmale von Softwaresystemen (ISO 9126)



Wir können uns bei der Softwarequalität z.B. diese Frage stellen:

- **Funktionalität:** Ist der Code sinnvoll strukturiert und getestet?
- **Änderbarkeit:** Ist der Code dokumentiert und modularisiert?
- **Zuverlässigkeit:** Werden Fehler im Code abgefangen und geloggt?
- **Übertragbarkeit:** Kann der Code auf anderen Systemen ausgeführt werden?

Ungemein wichtig für Softwaresysteme sind auch die 4 Themen im Bereich der Informationssicherheit:

1. **Vertraulichkeit:** Informationen sind nur für erlaubte Personen sichtbar.
2. **Integrität:** Informationen sind vor unerlaubten Änderungen geschützt.
3. **Verfügbarkeit:** Systeme sind verfügbar, wenn sie gebraucht werden.
4. **Vertrauenswürdigkeit:** Transaktionen passieren nur zwischen vertrauten Parteien.

Daten sind die **wertvollsten** Bestandteile von Softwaresystemen. Bei Daten gilt aber «Garbage in - Garbage out»: Nur mit hoher Datenqualität lassen sich gute Ergebnisse erzielen.

Die Qualität von Daten lassen sich an 5 Faktoren messen:

1. Konsistenz
2. Gültigkeit
3. Vollständigkeit
4. Korrektheit
5. Aktualität

- ⇒ Diese Aspekte müssen regelmässig geprüft werden.
- ⇒ z.B. Beim Speichern oder Auswerten der Daten

Bei diesem Verfahren werden die Lief­er­er­geb­nis­se von einer weiteren Person auf Spezifikation und Codequalität überprüft.

- ⇒ Dies verbessert neben der Qualität auch die Wartbarkeit.
- ⇒ Kann bis ins Extreme praktiziert werden (Pair-Programming).

2. Continuous Delivery

Bei diesem Verfahren wird die Software fortlaufend kompiliert, getestet, verpackt und in die Produktion gestellt.

- ⇒ Kleine Schritte reduzieren das Schadenmass von Fehlern.
- ⇒ Dies benötigt ein sinnvolles Versionssystem wie Git.

Das Diagramm zeigt die Continuous Delivery Pipeline als einen Prozess, der von der Entwicklung bis zum Kunden führt. Es besteht aus drei Hauptkomponenten, die in Kreisen dargestellt sind:

- Änderung/Erweiterung im Code**: Der Ausgangspunkt des Prozesses.
- Continuous Delivery Pipeline**: Der zentrale Prozess, der die Automatisierung der Bereitstellung ermöglicht. Er umfasst die folgenden Schritte:
 - Versionsüberprüfung
 - Build
 - Automatisierte Tests
 - Manuelle Tests
- Kunde**: Der Endnutzer, der die bereitgestellten Funktionen erhält.

Die Komponenten sind durch Pfeile verbunden, die den Fluss des Prozesses von links nach rechts darstellen.

- ### 3. Ad absurdum und Negativtests

Bei diesem Verfahren werden bewusst alle Anforderungen und Ergebnisse angezweifelt und hinterfragt.

- ⇒ Ist diese Anforderungen wichtig? Oder ist sie unnötig?
- ⇒ Erstellte Testfälle sollen bewusst Fehler auslösen.

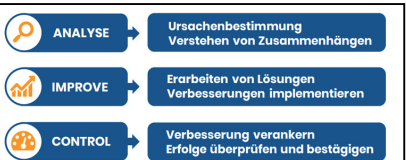
QM-Handbuch
Ein gutes QM-Handbuch («How-To») kann dabei helfen, die Qualitätssicherung in einem Projekt sauber durchzuführen. Oft stossen solche Handbücher aber auf diese Probleme:

- Die Ziele sind nichts aussagend
- Die Ziele sind unspezifisch
- Die Ziele sind unvollständig
- Die Ziele sind unkonkret

⇒ Es gilt: Lieber kein Handbuch als ein schlechtes.

Qualitätssicherung im Unternehmen
SixSigma
SixSigma ist ein mathematisches Modell zur Messung und Optimierung von Geschäftsprozessen. Es basiert auf dem DMAIC-Prinzip

- ⇒ SixSigma ist unabhängig von Prozess und Branche.
- ⇒ Die Anwendung der Methodiken ist dabei frei.



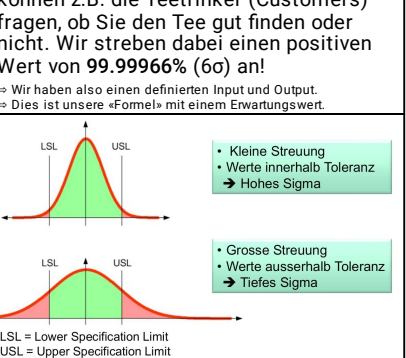
Bei Define wollen wir das Betrachtungsfeld eingrenzen. Wir können z.B. ein Prozess mittels SIPOC definieren.

- ⇒ SIPOC: Suppliers, Inputs, Process, Outputs, Customers
- ⇒ s. Beispiel «Teezubereitung»

Suppliers	Inputs	Process	Outputs	Customers
Teelieferant	Teebeutel	Wasser kochen	Tasse Tee	Teerinker
	Zucker	Teebeutel und Wasser in die Tasse einfüllen		
Energieversorger	Wasser		Verbraucher Teebeutel	Mülltonne
Wasserversorger	Tasse	5 Minuten ziehen lassen		
	Wasserkocher			
	Strom			
	Teeöffel	Teebeutel entnehmen		

Measure

Nun werden die Werte gemessen.



Klassisch vs. SixSigma

- Klassisch sind 99% (3.8σ) gut:
- 20'000 verlorene Briefe pro Stunde.
 - 5'000 falsche chirurgische Eingriffe pro Woche in Europa.
 - 2 Landungen ausserhalb der Rollbahn auf den grössten Flughäfen täglich.

- Bei SixSigma sind 99.99966% (6σ) gut:
 7 verlorene Briefe pro Stunde.
 1.7 falsche chirurgische Eingriffe pro Woche
 in Europa.
 0.0007 Landungen ausserhalb der Rollbahn
 auf den grössten Flughäfen.

⇒ Bei SixSigma passieren nur 3.4 Fehler auf eine Million.

improve