

Ein Projekt abwickeln

Einführung ins Projektmanagement



Lernziele

- Die Lernenden können in eigenen Worten die Eigenschaften eines Projekts beschreiben.
- Die Lernenden kennen mindestens eine Methode/Vorgehensweise zur Abwicklung eines Projekts.

Wer hat schon bei einem Projekt mitgearbeitet?



Was ist ein «Projekt»?

Wann werden Projekte «gemacht»?



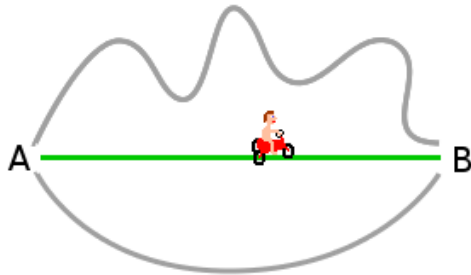
Projekt, Definition

Ein Projekt ist ein Vorhaben, das...

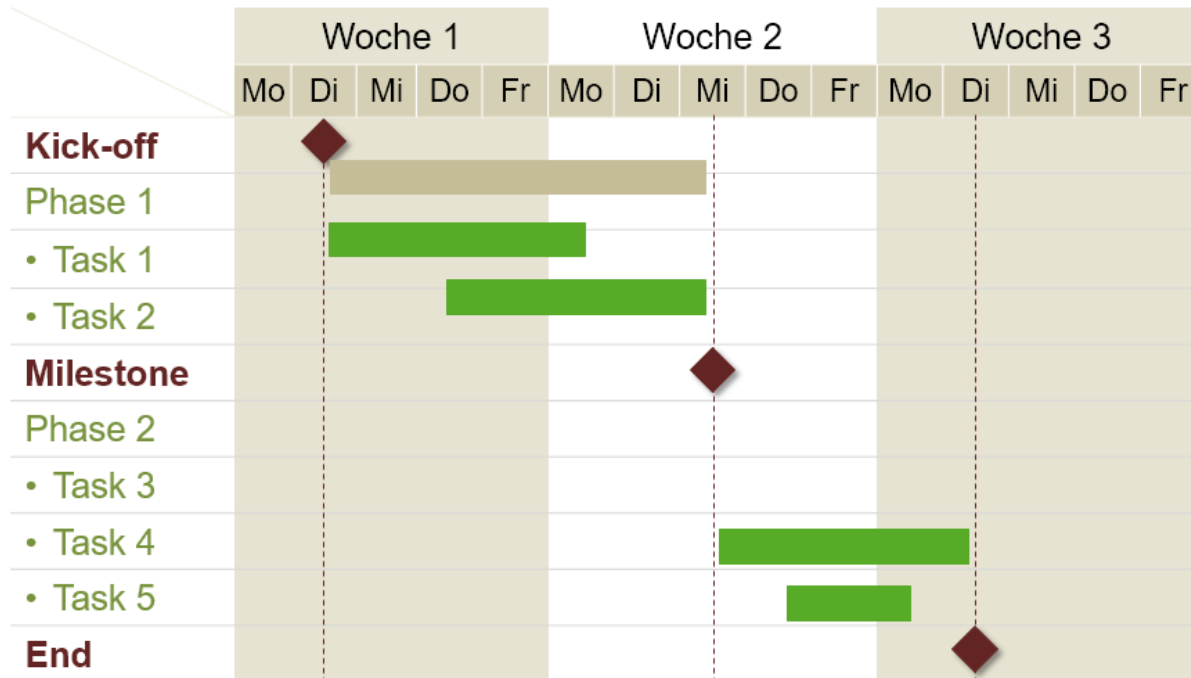
...einmalig ist.

...zeitlich begrenzt ist.

...zielgerichtet ist.



Eigenschaften eines Projekts



Projekt, wann?

Bei Aufgabenstellungen...

...die relativ **komplex** erscheinen.

...deren **Lösungsweg** zunächst **unbekannt** ist.

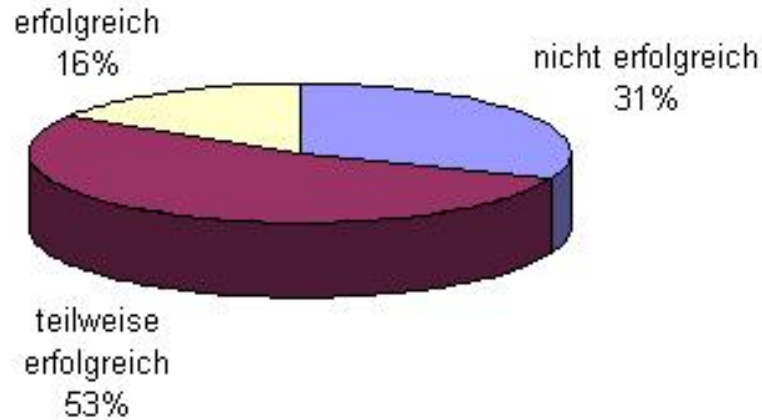
...mit einer **Zielrichtung** u. einem **Zeitraumen**.

...die eine **bereichs-/fachübergreifende**
Zusammenarbeit erfordern.

Können Projekte auch scheitern



1994: 84% aller IT Projekte in den USA scheitern



erfolgreich: Projekt wurde innerhalb der vorgegebenen Zeit und Budget abgeschlossen. Projektergebnis ist im Einsatz und erfüllt alle Anforderungen.

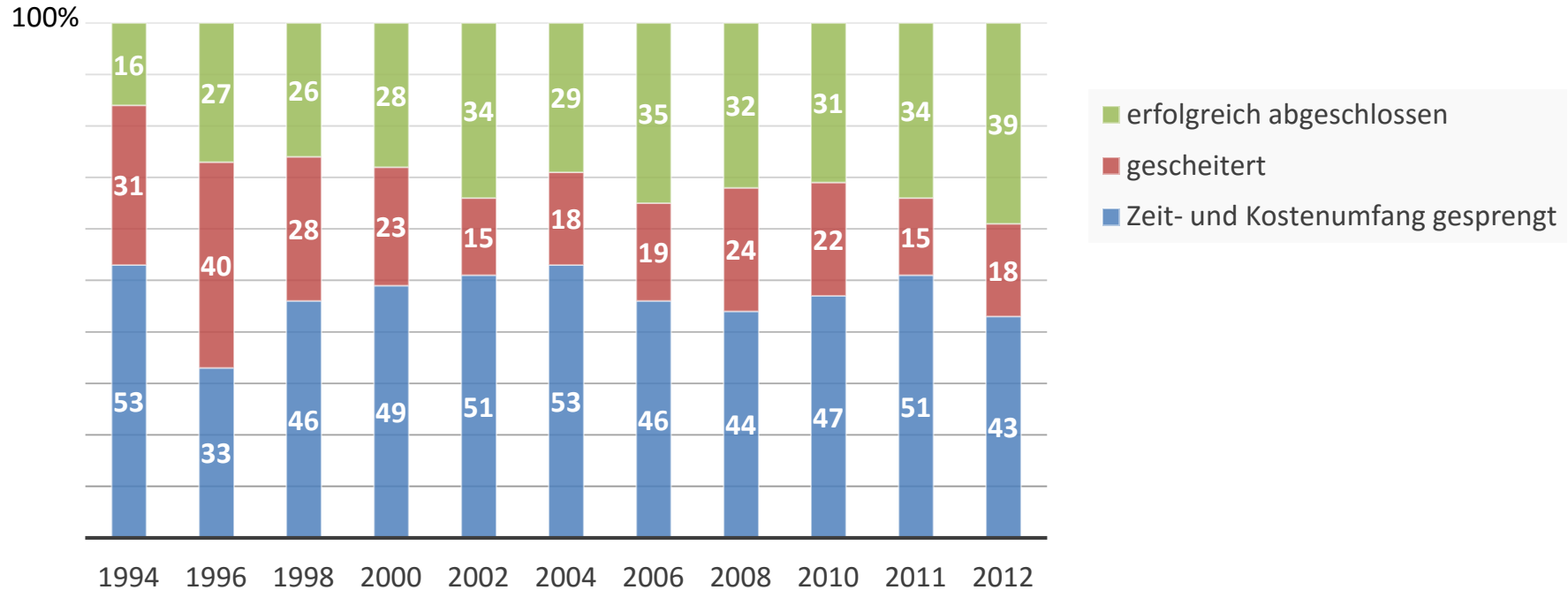
nicht erfolgreich: Das Projekt wurde vorzeitig abgebrochen oder das Projektergebnis wurde nie eingesetzt.

Verluste 1994: \$ 80 Mrd.

teilweise erfolgreich: Projekt ist abgeschlossen. Projektergebnis ist im Einsatz. Zeit, Budget oder Leistung sind aber nicht im vorgegebenen Umfang erreicht worden.

Verluste 1994: \$ Billionen (Schätzung, nicht messbar)

Chaos Report: Entwicklung (USA)



Quelle: Standish Group, Chaos Report

Aufgabe: Warum die Vasa sank - Analogien zur Softwareentwicklung



«The Royal Swedish Navy
launched the Vasa. After
sailing only 1'300 meters,
it sank. 53 lives were
lost.»

Warum gehen IT Projekte schief?

- Ziele falsch gesetzt
- unklare/fehlende Anforderungen
- häufige Anforderungsänderungen
- Endbenutzer nicht eingebunden
- zu grosse Komplexität
- (Ressourcen-) Planung ungenügend
- ungenügendes Testen
- unzureichende Kommunikation
- etc...





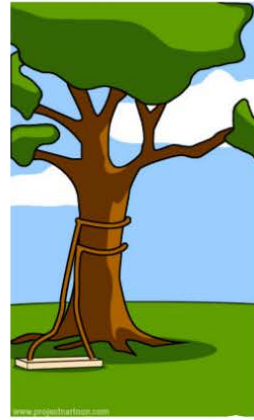
Was der Kunde erklärte



Was der Projektleiter
verstand



Wie es der Analyst
desigte



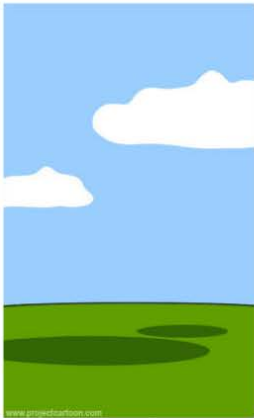
Wie es der Entwickler
entwickelte



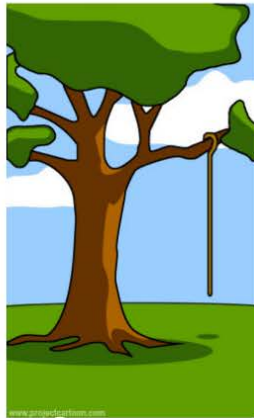
Was die Beta-Tester
erhielten



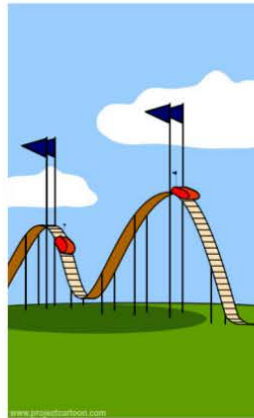
Wie es der Berater
beschrieb



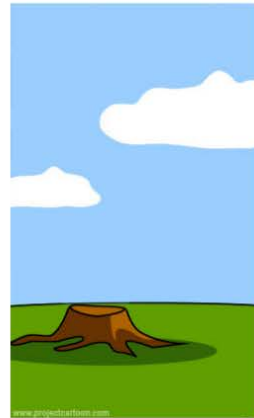
Wie es dokumentiert
wurde



Wie es ausgeliefert
wurde



Was dem Kunden in
Rechnung gestellt wurde



Wie es supported
wurde



Wie es Marketing
verkaufte



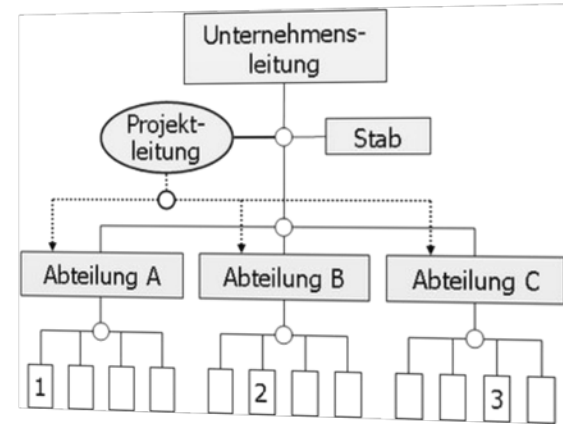
Was der Kunde
eigentlich wollte

Wie ein Projekt abwickeln?

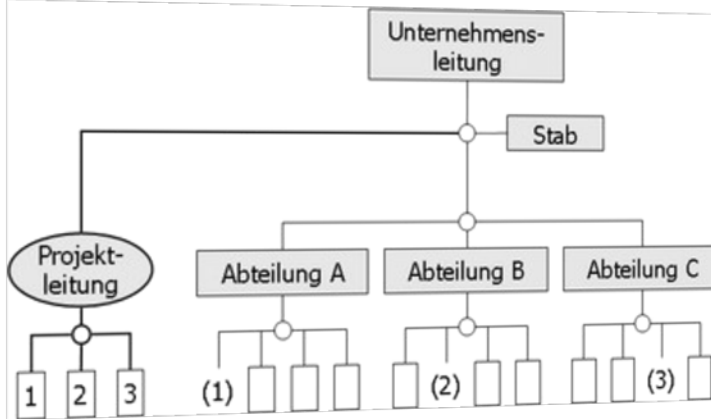


Projektorganisation

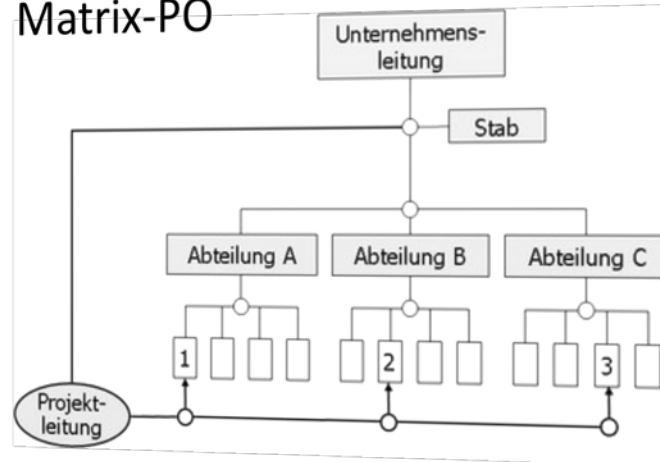
Stab-PO



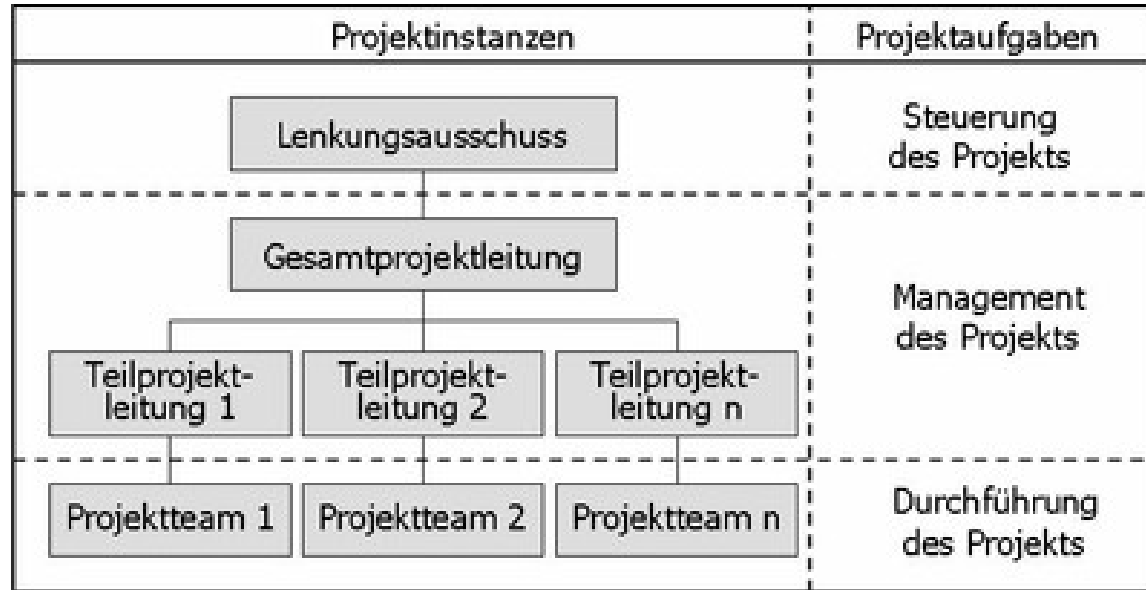
Reine PO



Matrix-PO



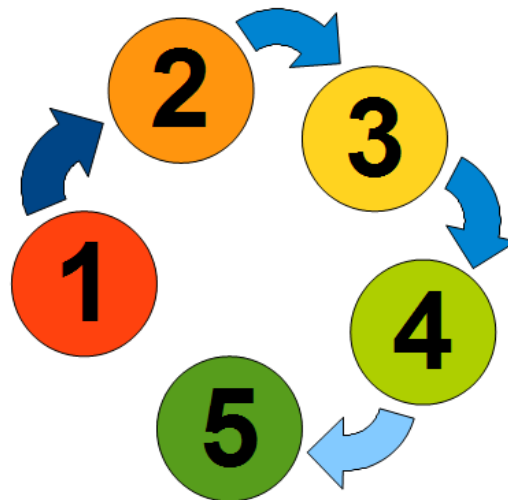
Projektinterne Struktur





Methodisch vorgehen!

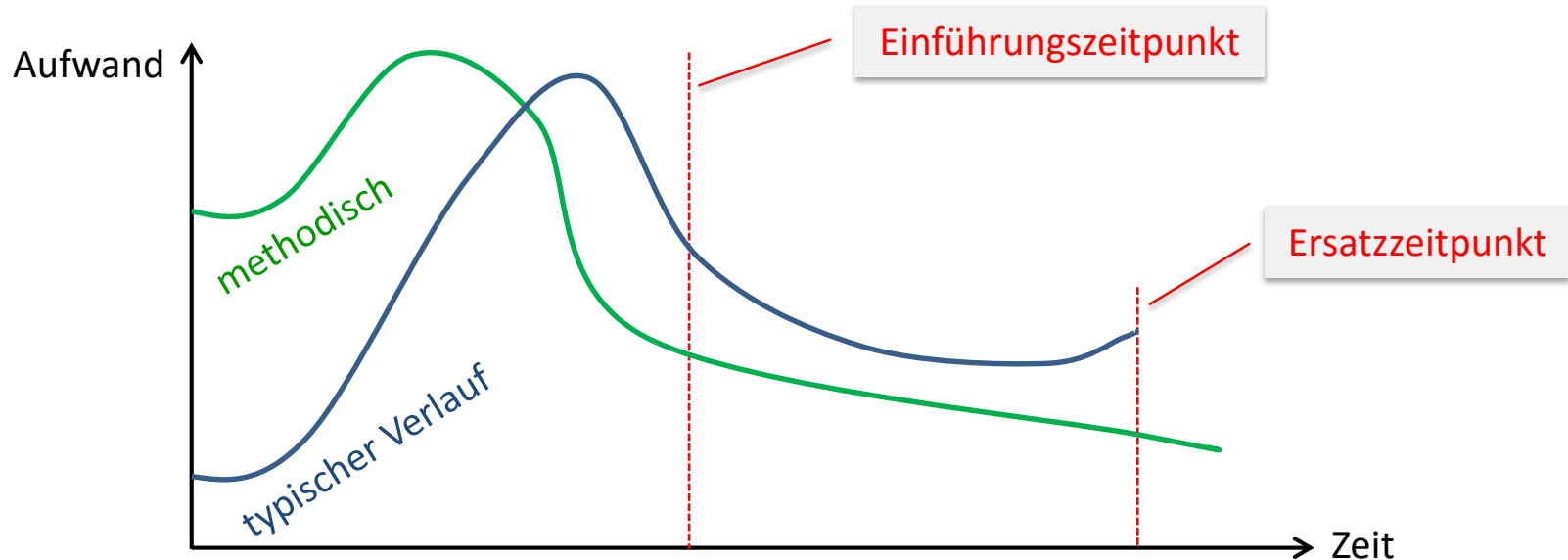
Als **Methode** bezeichnet man in der Informatik eine «*systematische und zielgerichtete Vorgehensweise, die für eine Vielzahl von Problemen zu einer sinnvollen Lösung führt*».



Methodische Entwicklung

Ziele:

- Reduktion der Kosten in der Wartungsphase
- Verlängerung der Lebensdauer



5 Einführung

- Setup and Deployment
- Installation
- Schulung
- Lernprogramme
- Benutzerhandbuch

4 Testing

- Testplan (Umfang, Vorgehensweise)
- Komponententests
- Integrationstest
- Systemtest
- Acceptance-Test (Beta-Test)

3 Implementierung

- Umsetzung des Designs
- Code schreiben/kommentieren
- Source Control
- Debugging/Bugfixing
- Unit Tests



1 Analyse (Was?)

- Aktuelle Situation (IST)
- Ziel definieren (SOLL)
- Anforderungen ermitteln
- Geschäftsabläufe des Kunden verstehen
- Dokumentation als Grundlage für das Design

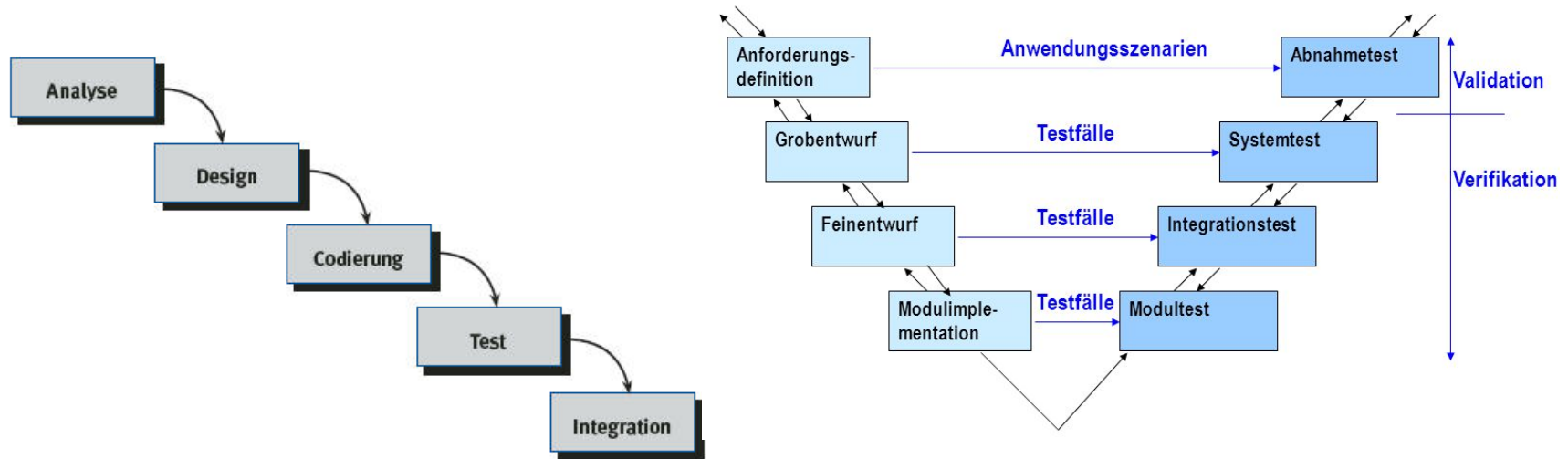
2 Design (Wie?)

- Lösung(en) erarbeiten
- Systemarchitektur bestimmen
- GUI entwerfen
- Persistenz u. Datensicherung (wo? wie?)
- Vor- u. Nachteile verschiedener Lösungswege abwägen
- Risiken beurteilen
- Dokumentation als Grundlage für die Implementierung

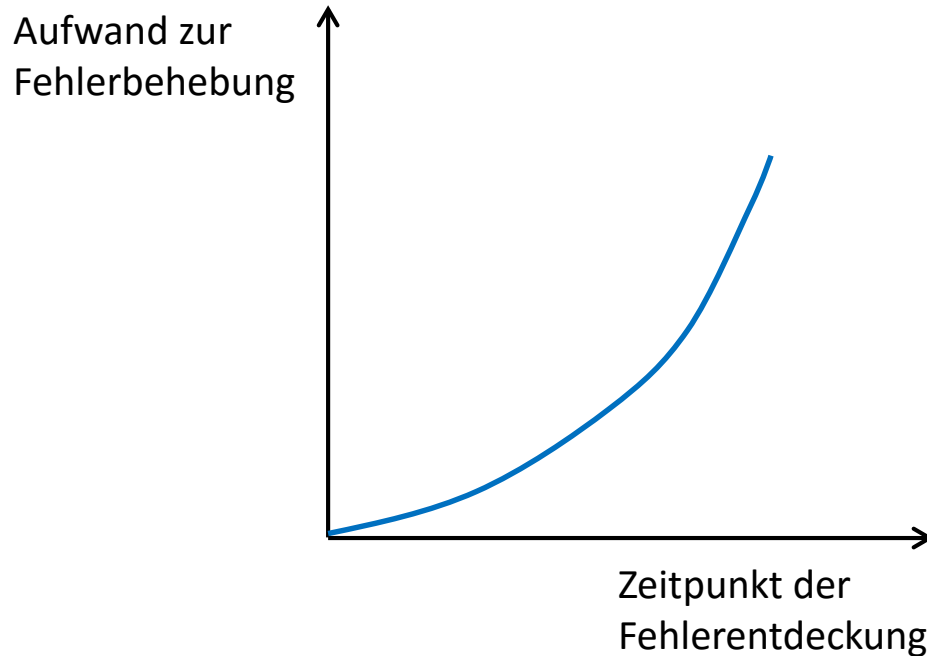
Phasen der Softwareentwicklung

Frühe Phasenmodelle

- Sequentielles Vorgehen



Probleme des sequentiellen Vorgehens



Fehler werden spät erkannt.

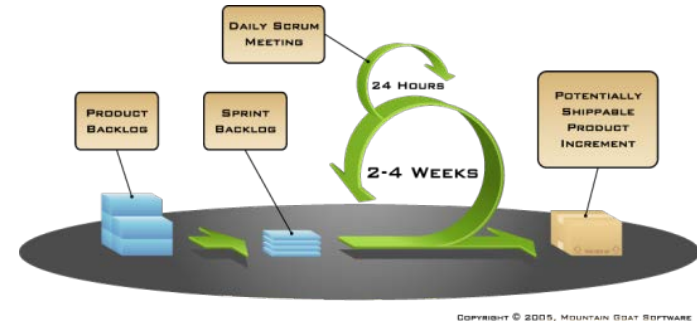
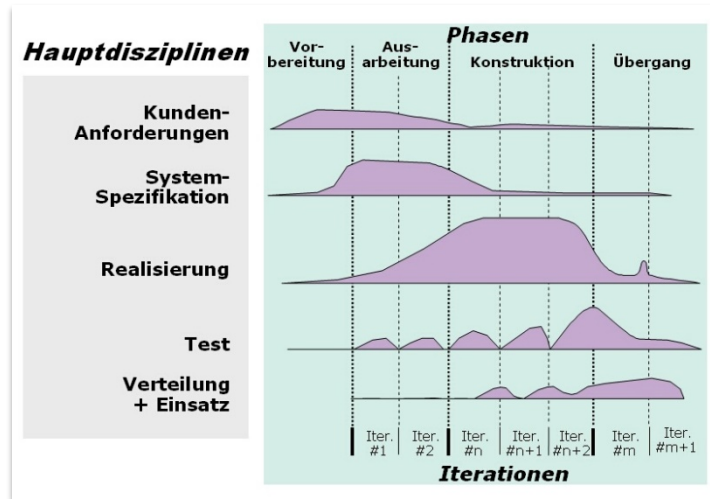
- Späte Entdeckung von Anforderungs-, Analyse- und Designfehlern (oft erst während Integrationstest)
- Risiken werden lange mitgeschleppt («weil jetzt noch nicht codiert/ausprobiert werden darf»)

Nachträgliche Anforderungen können nicht berücksichtigt werden.

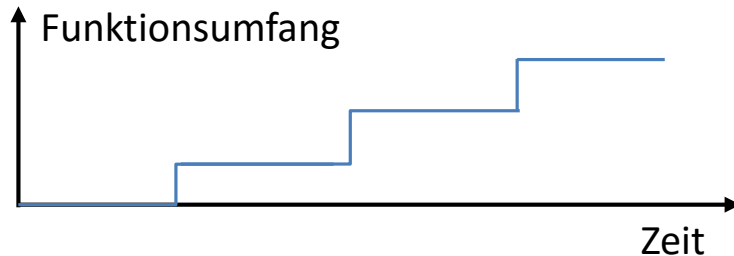
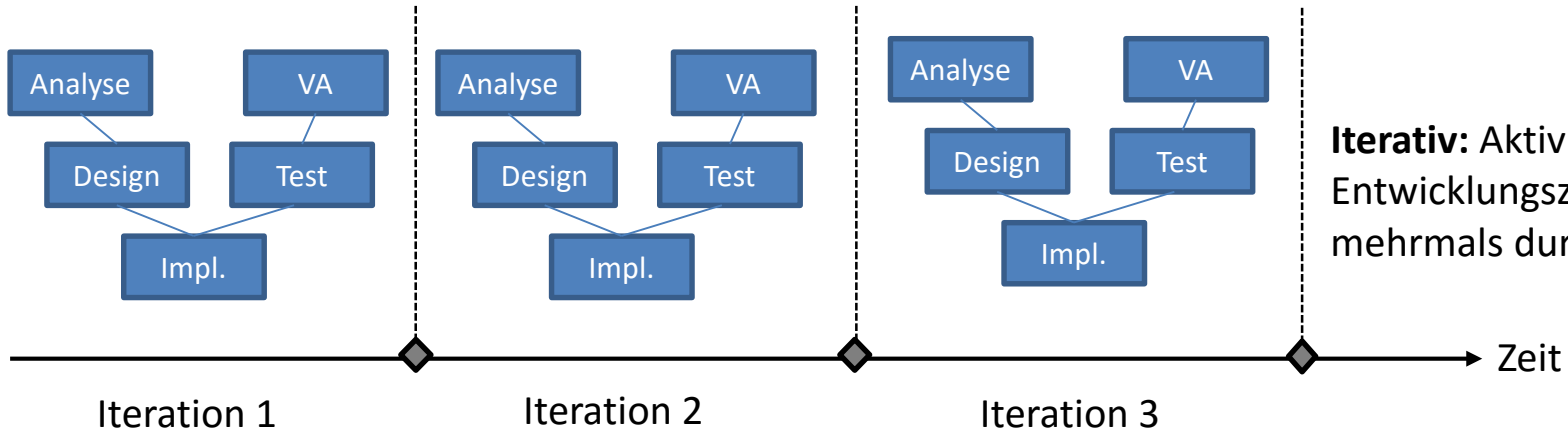
Projektfortschritt ist über lange Zeit nicht messbar.

Neue Phasenmodelle

- Iterativ inkrementelles Vorgehen



Iterativ inkrementell



Inkrementell: Releases weisen nach jeder Iteration einen grösseren Funktionsumfang auf.

Vorteile des iterativ inkrementellen Vorgehens

- Es entsteht schnell und laufend funktionierende Software
- Fortlaufend Kundenfeedback
- Kommunikation (mit Kunde, im Team)
- Anforderungsänderungen können berücksichtigt werden
- Probleme und Risiken werden früh erkannt
- Greatest Risk first: Wenn das Projekt scheitert, dann früh u. bevor viel Geld ausgegeben wurde
- Timeboxing (Zeitplan wird eingehalten)
- Erfahrungen u. Korrekturen können schnell eingebracht werden

Agile Manifesto

„Wir zeigen bessere Wege auf, Software zu entwickeln, indem wir es selber tun und anderen dabei helfen, es zu tun. Durch unsere Arbeit sind wir zu folgender Erkenntnis gekommen:

1. **Menschen und Interaktionen** sind wichtiger als *Prozesse und Werkzeuge*.
2. **Funktionierende Software** ist wichtiger als *umfassende Dokumentation*.
3. **Zusammenarbeit mit dem Kunden** ist wichtiger als *Vertragsverhandlungen*.
4. **Eingehen auf Veränderungen** ist wichtiger als *Festhalten an einem Plan*.

Das heisst: obwohl die Punkte auf der rechten Seite durchaus wichtig sind, halten wir die Punkte links für wichtiger.“

Methoden, Vorgehensweise



Gruppe 1 (App) : Was ist **Scrum** ?

Gruppe 2 (Sys) : Was ist **IPERKA** ?

- Präsentation
- Merkblatt



Anforderungen an die Dokumentation

- Vollständigkeit – keine Lücken
- Relevanz – nichts Überflüssiges
- Verlässlichkeit – Inhalt aktuell, fehlerfrei
- Ordnung – übersichtlich, gut strukturiert
- Verständlichkeit – Sprache, Illustrationen
- Quellenangaben – Verfasser, fremde Unterlagen

