

# **Software-Engineering – Teil 2, SS**

## **3. Projektmanagement**

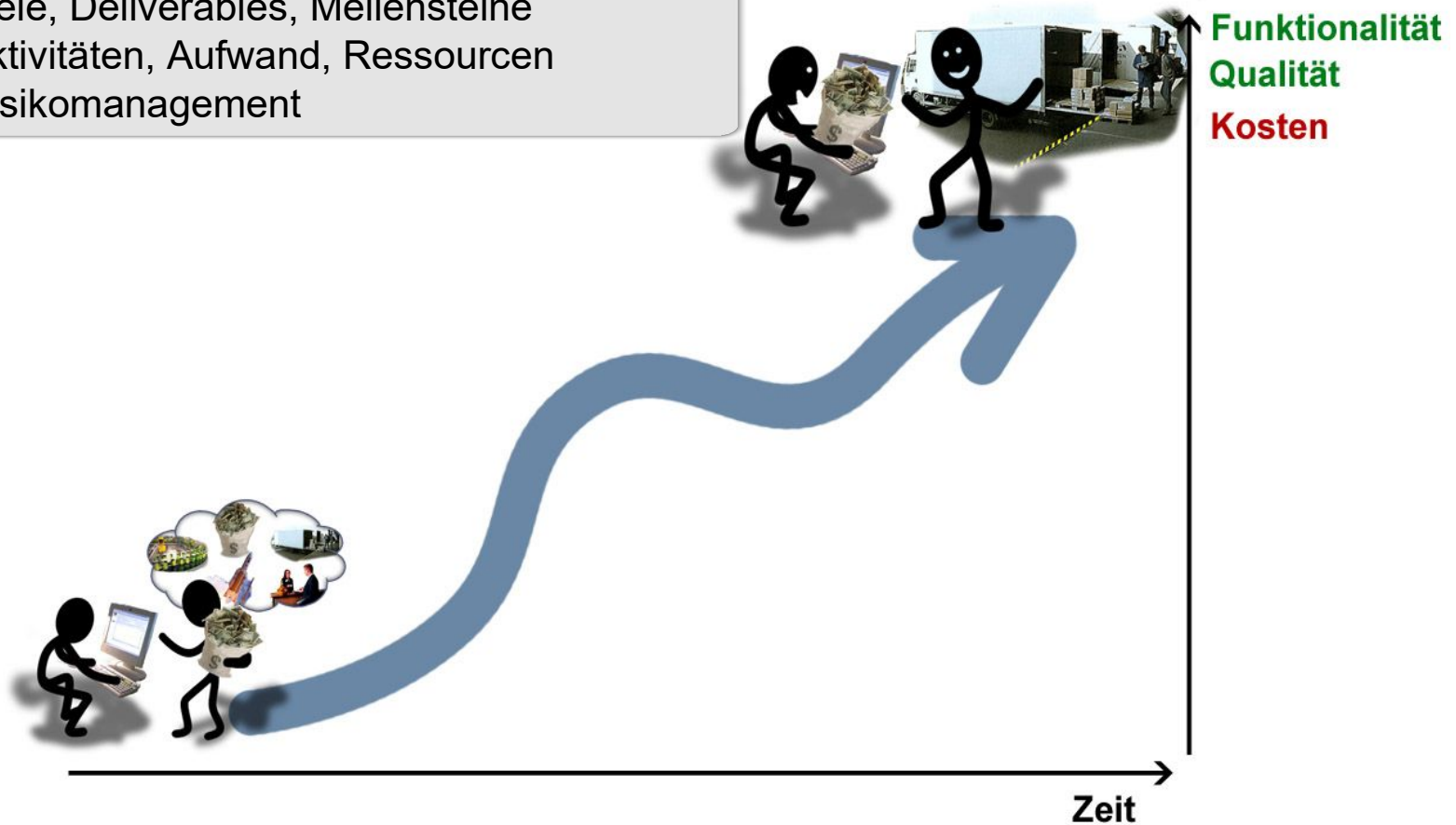
Prof. Dr. Eckhard Kruse

DHBW Mannheim

# Projektmanagement

## Projektplanung, -überwachung, -steuerung

- Ziele, Deliverables, Meilensteine
- Aktivitäten, Aufwand, Ressourcen
- Risikomanagement



# Projektmanagement - Lessons learned?

## **Projektmanagement Rückblick WS-Projekt – Ausblick SS-Projekt**

- Was lief gut im Wintersemester?
- Was möchten Sie wieder auf ähnliche Weise machen?
- Was lief nicht so gut?
- Was möchten Sie diesmal anders/besser machen?



# Projektmanagement und SW-Engineering

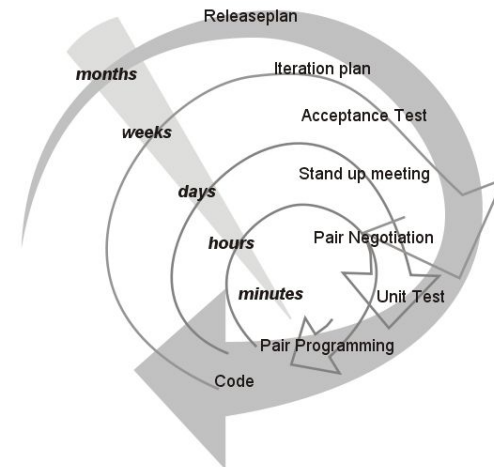
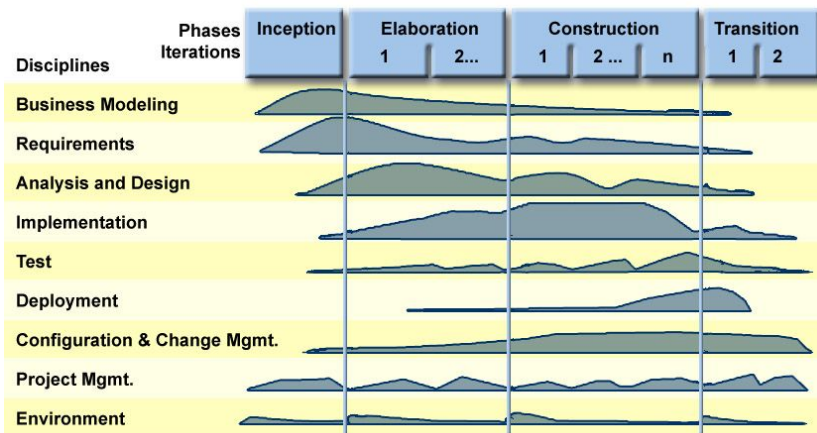
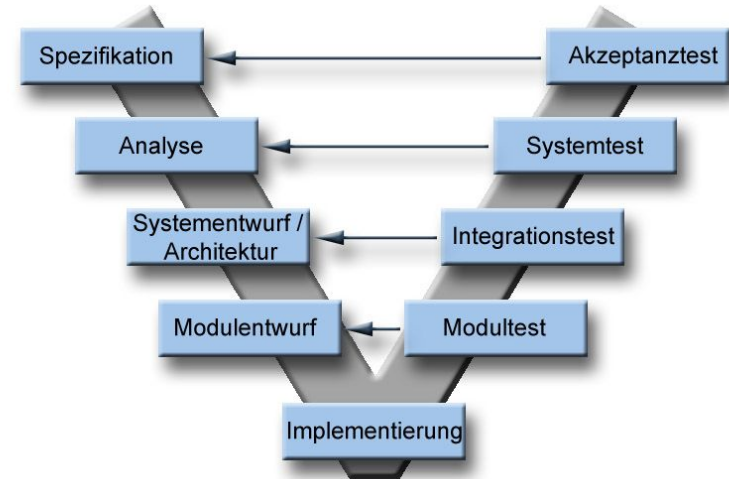
SW.-Eng.	Proj.Mgmt.	Funktions- umfang	Qualität	Zeit Kosten	Management Controlling Risiken
Definierte Ziele					
Anforderungsanalyse					
Prozess-/Vorgehensmodelle					
Team, Klima, Soziales					
Dokumentation					
Planung					
Aufwandsschätzung					
Modelle und Methoden					
Programmiersprachen					
Code-Reviews					
Tools und Frameworks					
Re-Use					
Configuration Management					
Testen					
Qualitätsmanagement					
...					

→ Auswirkungen auf?

Können Sie diese Tabelle füllen?

Wie sieht es konkret in Ihrem Projekt aus?

# Vorgehensmodelle



# Wahl eines Vorgehensmodells

## Entscheidungskriterien:

- Welches Modell wurde (im Unternehmen, vom Team) bisher verwendet? Wie erfahren ist das Team damit? Hat sich das Modell bewährt?
- Wie groß ist das Projekt? Wie hoch sind die Projektrisiken? Wie hoch sind die Qualitätsansprüche?
  - sicherheitskritische Lösungen erfordern z.B. zwingend einen starren, sehr genauen dokumentierten Prozess (→ Zertifizierung).
- Gibt es äußere Randbedingungen, unbedingt zu verwendende Standards?
  - Unternehmensrichtlinien, z.B. Unternehmen strebt CMMI-Zertifizierung an.
- Wie genau sind die Anforderungen am Anfang bekannt? Kann es noch größere Änderungen im Laufe des Projektes geben?
  - Extrembeispiel: Forschungsprojekte → Projektergebnis ist am Anfang vage
- Grundsatzentscheidung: Agil oder Traditionell?
- Was ist der richtige Umfang/Aufwand für den Prozess (welche Dokumente usw. sind erforderlich?)
- Sind Abweichungen vom Modell erlaubt? Vereinfachungen (z.B. bestimmte Phasen weglassen)?

Egal, welches Modell letztlich gewählt wird, wichtig ist:  
Es gibt ein definiertes, geplantes Vorgehen und das Team versteht es und befolgt es!



# Bekannte (Prozess-)Probleme?

Aus **OpenUP –The Best of Two Worlds**, Bjorn Gustafsson

<http://www.methodsandtools.com/archive/archive.php?id=69>

**Some common problems can be observed in troubled projects:**

- **The project team doesn't share a clear vision of how the system will appear to its users.** Without a clear vision of the final system, there is no guiding framework for the work in the project. The team's analysts have no means to calibrate their requirements to the scope and effort of the project, which results in ill-fitting requirements statements; and the development team can not properly prioritize their work.
- **Requirements do not drive development work.** Some development cultures regard requirements as "incidental" input to the project only, and drive the development work based on other, internal and technical, conditions. This is commonly found in "silo" organizations where there are separate teams for requirements capture and development.
- **The system's architecture has not been articulated.** Although projects that only evolve and maintain existing systems may not need to pay much attention to architecture, there are many projects that do. As Dean Leffingwell points out: "... how much architecture a team needs depends in large part on what the team is building".
- **Plans are not connected to the engineering reality.** Plans are often created and maintained separately from the actual project work. We have all seen nebulous Gantt charts that project managers spent days or even weeks creating, with hundreds of line items in nifty breakdown structures, purportedly believed to bring the project to "completion" at some well-defined point in time. Of course, these plans become outdated even before they are pinned on the wall.
- **Risks are ignored.** All projects run the risk of building the wrong product or not being able to build the product as envisioned, yet very few projects acknowledge this uncertainty and actively try to reduce it.

## Software-Engineering-Projekt

### 3.1 Vorgehensmodell

Sie haben den Auftrag erhalten und wollen nun entscheiden, welches Vorgehensmodell die Basis Ihrer Projektplanung werden soll.

- a) Welches Vorgehensmodell erscheint Ihnen grundsätzlich geeignet? Warum? (Vorteile/Nachteile?)
- b) Welche Teile des Vorgehensmodells wollen Sie verwenden, wo nehmen Sie Vereinfachungen oder Änderungen vor? Welche Bausteine des Modells sind für Sie wichtig, welche können Sie weglassen?
- c) Der Qualitätsmanager Ihres Unternehmens, der auch für die Definition der internen Prozesse verantwortlich ist, möchte die Vorgehensweise in ihrem Projekt begutachten. Dabei ist er auch bereit, für dieses Projekt ggf. von den Standardprozessen abzuweichen und etwas Neues auszuprobieren. Bereiten Sie sich darauf vor, Ihren Qualitätsmanager von Ihrer Vorgehensweise zu überzeugen.



## Teamarbeit

### 3.2 Projektmanagement

Im Rahmen des Projektmanagements sind folgende Dokumente zu erstellen:

a) Projektplan

- Projektziele (s. Pflichtenheft)
- Grober Strukturplan
- Projektorganisation

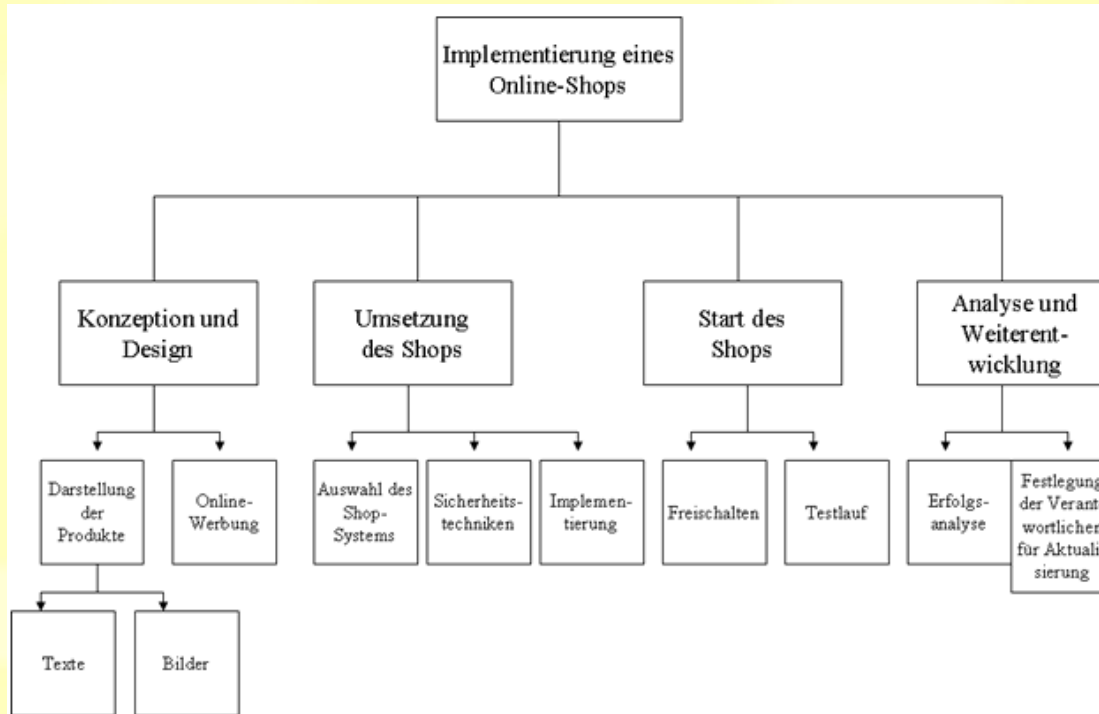
b) als Anhang zu a): Detaillierte Aktivitätsliste (s. Template)  
Oder alternativ: Microsoft Project Plan (oder mit anderem Tool)

c) Risikoliste (s. Template)

Alle Dokumente sind stets aktuell zu halten, d.h. mindestens nach Abschluss einer Projektphase und ereignisbasiert bei Änderungen.

# Planung Top-Down

## Work breakdown structure / Projektstrukturplan



## Planung:

- Aktivitäten
- Abhängigkeiten
- Zeitplanung + Ressourcenzuordnung

# Wie planen?

## Grundkonzepte für gute Planung:

- Top-Down: Zwischenziele und Aktivitäten vom Groben ins Feine
  - keine Details, solange an anderen Stellen noch Grundlegendes offen ist.
  - Black-Box-Aufgabenblöcke und Schnittstellen (wer liefert wem was)
- Priorisierung:
  - 1. Wichtiges, 2. Schwieriges, Unsicheres und Planungslücken 3. Rest
- Vollständig: Gesamtüberblick, nichts 'vergessen'/ausblenden.
  - Alle Aspekte: Aufgaben, Ressourcen, Kosten, ....
- Iterativ: Planung auf tieferer Ebene kann Rückwirkungen auf Gesamtplanung haben.
- Wenig Abhängigkeiten zwischen Arbeitspaketen
  - erlaubt Parallelisierung, gut für 'kritischen Pfad'
- Fokussierte Aufgabenzuordnung:
  - schlecht: jeder macht überall ein bisschen mit.
  - gut: jeder hat (zu einem Zeitpunkt) nur ein Hauptthema.
- Klare Arbeitspaketdefinition
- Überschaubar: Angemessene Arbeitspaketgröße

# Planung Aktivitäten

Activity ID	Name	depends on (ID)	required by (ID)	Estimated effort (h)	Effort completed (h)	Remaining	Resources	Planned start	Planned ready	Completed	Comments
<b>1</b>	<b>Anforderungsanalyse</b>										
1.1	Anwenderinterviews			7	7	0	Peter, Petra	11.05.2009	20.05.2009	done	
1.2				21	14	7					
						0					
<b>2</b>	<b>Vorstudien</b>					0					
2.1	Technologieevaluation ....					0					
	Vorstudien					0					
						0					
<b>3</b>	<b>Entwurf</b>					0					
3.1	Komponente 1	1.1, ...	4.1, ...			0					
3.2	GUI					0					
3.3	Business Logic					0					
3.4	Architekturdoku finalisieren					0					
						0					
<b>4</b>	<b>Implementierung</b>					0					
4.1	Komponente 1					0					
	....					0					
						0					
<b>5</b>	<b>Testen</b>					0					
5.1	System Integration test Alpha 1					0					
	....					0					
						0					
<b>6</b>	<b>Integration/Inbetriebnahme</b>					0					
6.1	Installation beim Kunden					0					
	...					0					
						0					
<b>9</b>	<b>Projektmanagement</b>					0					
9.1	Projektplan 1st draft					0					
9.2	...					0					
<b>Sum</b>				<b>28</b>	<b>21</b>	<b>7</b>					
<b>Project completion status:</b>						<b>75,00%</b>					

→ Template: Activity list

## Teamarbeit

### 3.3 Planung

Das Pflichtenheft ist fertig und der verbindliche Vertrag abgeschlossen.

Planen Sie nun das weitere Vorgehen:

- a) Planen Sie Aktivitäten und Verantwortliche → Template Activity list
- b) Wie können die abgeschätzten Aufwände über die Zeit verteilt werden?
- c) Welche wesentlichen Phasen (Entwurf, Implementierung, Test) unterscheiden Sie?
- d) Was sind wichtige Meilensteine? Für wann werden sie geplant?
- e) Sind Änderungen am ursprünglich vorgeschlagenen Grobplan erforderlich? An welchen Stellen?

# Risikomanagement

Risikomanagement ist der Teil des Projektmanagements, der sich mit der Identifizierung, Analyse und Beherrschung von Risiken für die geplante Projektabwicklung beschäftigt.

Vorgehen:

- Projektrisiken identifizieren und auflisten
- Mögliche Auswirkungen (auf den Projekterfolg) analysieren
- Ggf. bereits vorbeugende Maßnahmen initiieren
- Gegenmaßnahmen überlegen, falls der Risikofall eintritt
- Risikomanagement als regelmäßigen Prozessschritt durchführen, da sich Projektrisiken verändern können.

→ **Template**



## Teamarbeit

### 3.4 Risikomanagement

Initiieren Sie das Risikomanagement.

- a) Erstellen Sie die Risikoliste.
- b) Wann/wie planen Sie die regelmäßige Aktualisierung der Risikoliste?



# Infrastruktur und Werkzeuge

In Softwareentwicklungsprojekten wird in der Regel eine Reihe von unterstützenden Softwaretools und Hardware benötigt. Die Bereitstellung und Pflege einer solchen Infrastruktur sollte frühzeitig geplant werden.

Beispiele für Infrastruktur:

- Entwicklungsumgebung (IDE)
- Aufgabenspezifische Anwendungen (z.B. Grafikprogramm, 3D ...)
- Build-Umgebung (ggf. separater Build-Server)
- Repository (für Source Code, Konfigurationen etc.)
- Dokumentenmanagement (Dokumente, Plänen etc.)
- Änderungsmanagement und Bug/Issue Tracker.
- Kommunikation (im Team + nach außen): E-Mail, Instant messaging ...
- Testumgebung, Testtools
- ...

## Teamarbeit

### 3.5 Infrastruktur

Welche technische Infrastruktur möchten Sie im Projekt verwenden?

- a) Für welche Einsatzbereiche werden SW-Werkzeuge benötigt? Wo können Sie auf spezielle Werkzeuge verzichten? Warum?
- b) Für welche Produkte entscheiden Sie sich? Warum (Vor-/Nachteile)?
- c) Wer ist verantwortlich für Beschaffung, Installation und Wartung?
- d) Ergänzen Sie Ihr Projektmanagement-Dokument mit einer kurzen Beschreibung der eingesetzten Infrastruktur.

# Prozesse und Abläufe

In Rahmen des Managements von (Software-)Projekten ist es wichtig, grundlegende Abläufe und Prozesse verbindlich zu vereinbaren.

Beispiele für Fragen zu Prozessen und Abläufen:

- Wer ist für welche Entscheidungen verantwortlich?
- Wer muss über welche Informationen verfügen? Wie wird informiert?
- Wann gibt es regelmäßige oder außerplanmäßige Treffen (z.B. SteCo)?
- Wann laufen Systembuilds? Wann/wie Code einchecken?
- Wie sieht der Review-Prozess von Projektergebnissen aus?
- Welche Maßnahmen zur Qualitätssicherung müssen durchgeführt werden? (z.B. Testen)
- Wann/wie werden Ergebnisse/Komponenten als fertig/verbindlich "eingefroren"?
- Wie wird das Änderungsmanagement gehandhabt?
- Wie erfolgt die Abnahme der Ergebnisse?
- ...

## Teamarbeit

### 3.6 Prozesse

Definieren Sie für Ihr Projekt wesentliche Abläufe und Vorgehensweisen:

- a) Adressieren Sie alle relevanten Bereiche: Softwareentwicklung, Testen, Dokumentreview, Kommunikation, Qualitätsmanagement etc.?
- b) Vereinbaren Sie die wichtigsten Abläufe und dokumentieren Sie sie in Ihrem Projektmanagement-Dokument.
- c) Gibt es offene Punkte, die später ergänzt werden müssen?