

Bachelor of Science (BSc) in Informatik
Modul Software-Projekt 3 (PM3)

Projektmanagement

Ausgabe: FS24

Um was geht es?

- Was versteht man unter Projektmanagement?
- Welche Aufgaben hat ein Projektleiter?
- Wie plant und steuert man ein Projekt in einem iterativ-inkrementellen Softwareentwicklungsprozess?
- Wie sieht der Softwareentwicklungsprozess für PM3 aus?
- Welche Projektmanagement-Artefakte werden für PM3 verlangt?



Lernziele

- Sie sind in der Lage,
 - die Grundsätze des Projektmanagements zu erläutern.
 - die Artefakte zur Planung und Steuerung eines iterativ-inkrementell entwickelnden Projektes zu erklären.
 - ein Projekt gemäss dem vorgegebenen Softwareentwicklungsprozess durchzuführen.
 - die geforderten Artefakte für Ihr Projekt in PM3 zu erstellen.

Agenda

1. Einführung in Projektmanagement
2. Softwareentwicklungsprozess für SWEN1/PM3
3. Grobplanung
4. Iterationsplanung
5. Risikomanagement
6. Wrap-up

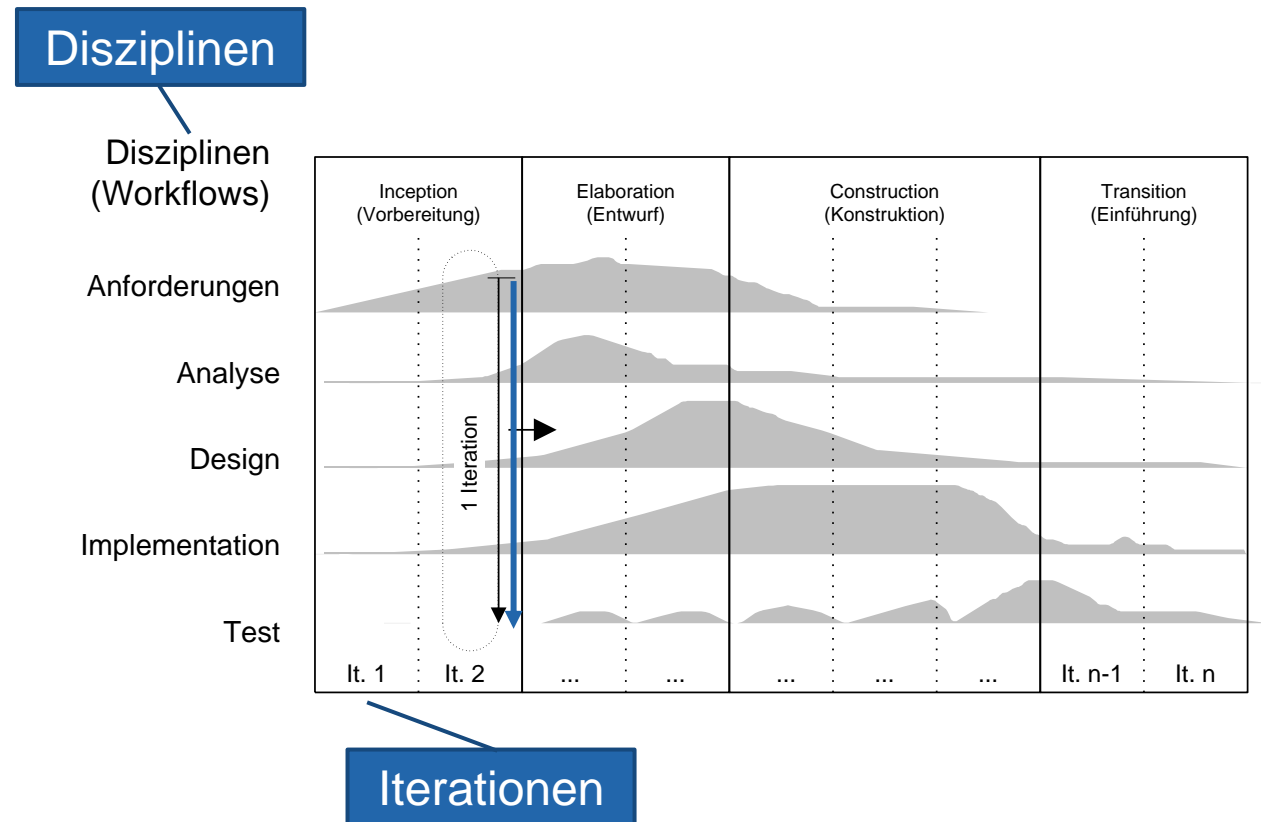
Einführung Projektmanagement (PM)

- Hauptziel des PM (Projektleiters)
 - Projektziel in der vereinbarten Zeit mit den vorgegebenen Mitteln erreichen
- Aufgaben des Projektleiters
 - Initiieren des Projekts
 - **Planen**
 - **Steuern**
 - **Kontrollieren**
 - Abschliessen des Projekts
- 3 zentrale Aspekte müssen geplant und überwacht werden
 1. Ziel (Produkt, Qualität)
 2. Mittel (Budget, Personalaufwand)
 3. Zeit
 - Diese beeinflussen sich gegenseitig!
- Zudem müssen die **Projekt-Risiken**
 - Identifiziert
 - Frühzeitig minimiert werden

Agenda

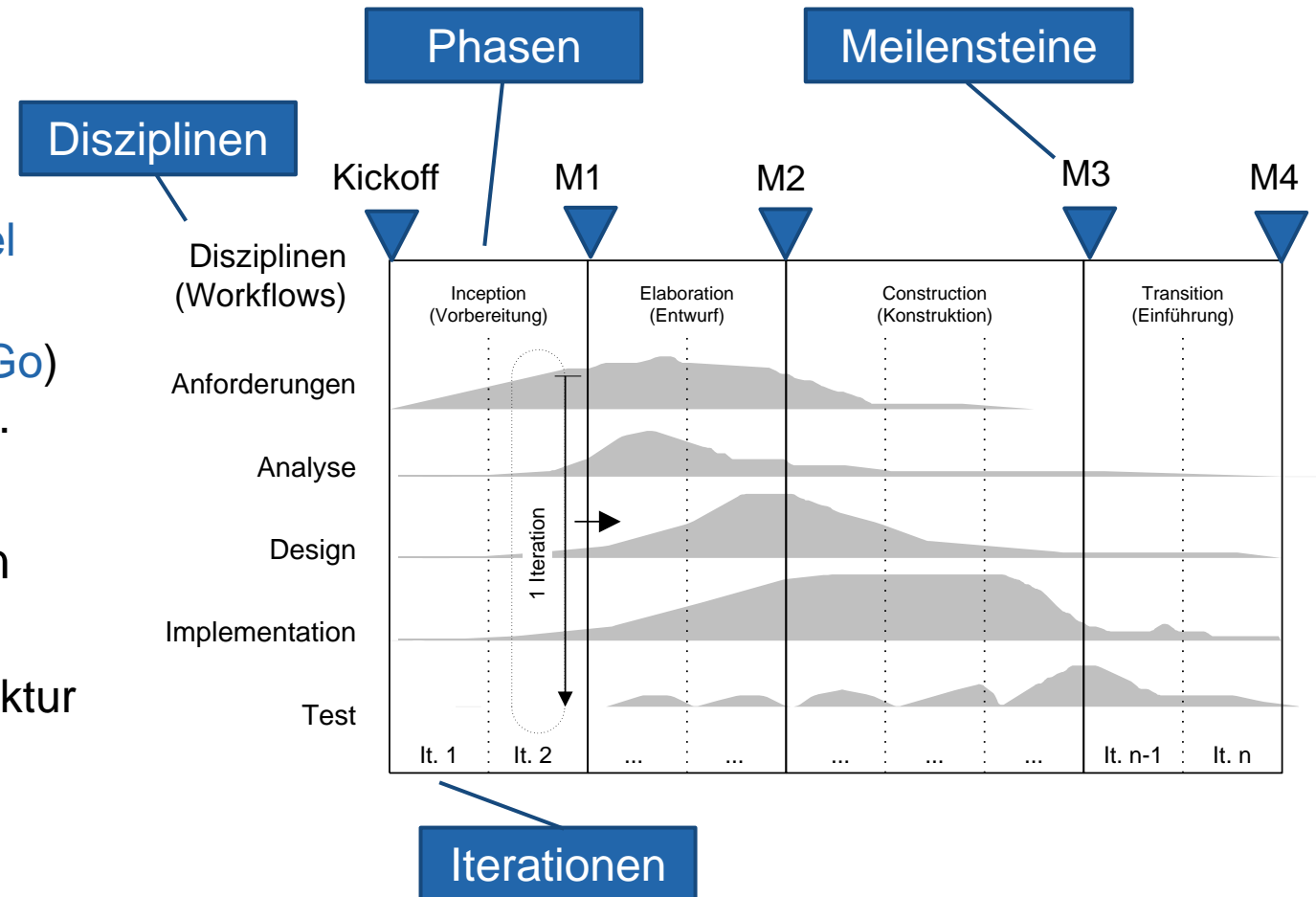
1. Einführung in Projektmanagement
- 2. Softwareentwicklungsprozess für SWEN1/PM3**
3. Grobplanung
4. Iterationsplanung
5. Risikomanagement
6. Wrap-up

- Wesentliche Merkmale
 - Iterativ-inkrementell
 - Projektdauer wird in **Iterationen** aufgeteilt
 - Kurze, fixe Iterationsdauer (2 Wochen)
 - Jede Iteration führt zu einem **messbaren Fortschritt (Software-Inkrement)** im Projekt
 - In jeder Iteration werden Arbeiten in **mehreren Disziplinen** ausgeführt
 - Anwendungsfall-orientiert
 - Projekt wird anhand von Anwendungsfällen (engl. Use Cases) geplant und durchgeführt
 - Risiko-getrieben
 - Risiken werden von Beginn weg identifiziert und frühestmöglich minimiert



Softwareentwicklungsprozess in SWEN1/PM3

- Meilensteine
 - Vereinbarte Zeitpunkte
 - Wo ein **vereinbartes Projektzwischenziel** erreicht sein muss
 - Wo wichtige Projektentscheide (**Go/NoGo**) von den Stakeholdern getroffen werden.
 - Normalerweise am Ende einer Phase
 - In PM3 sind **3 Meilensteine** vorgesehen
 - M1: Projektskizze
 - M2: Anforderungen und Lösungsarchitektur
 - M3: Prototyp



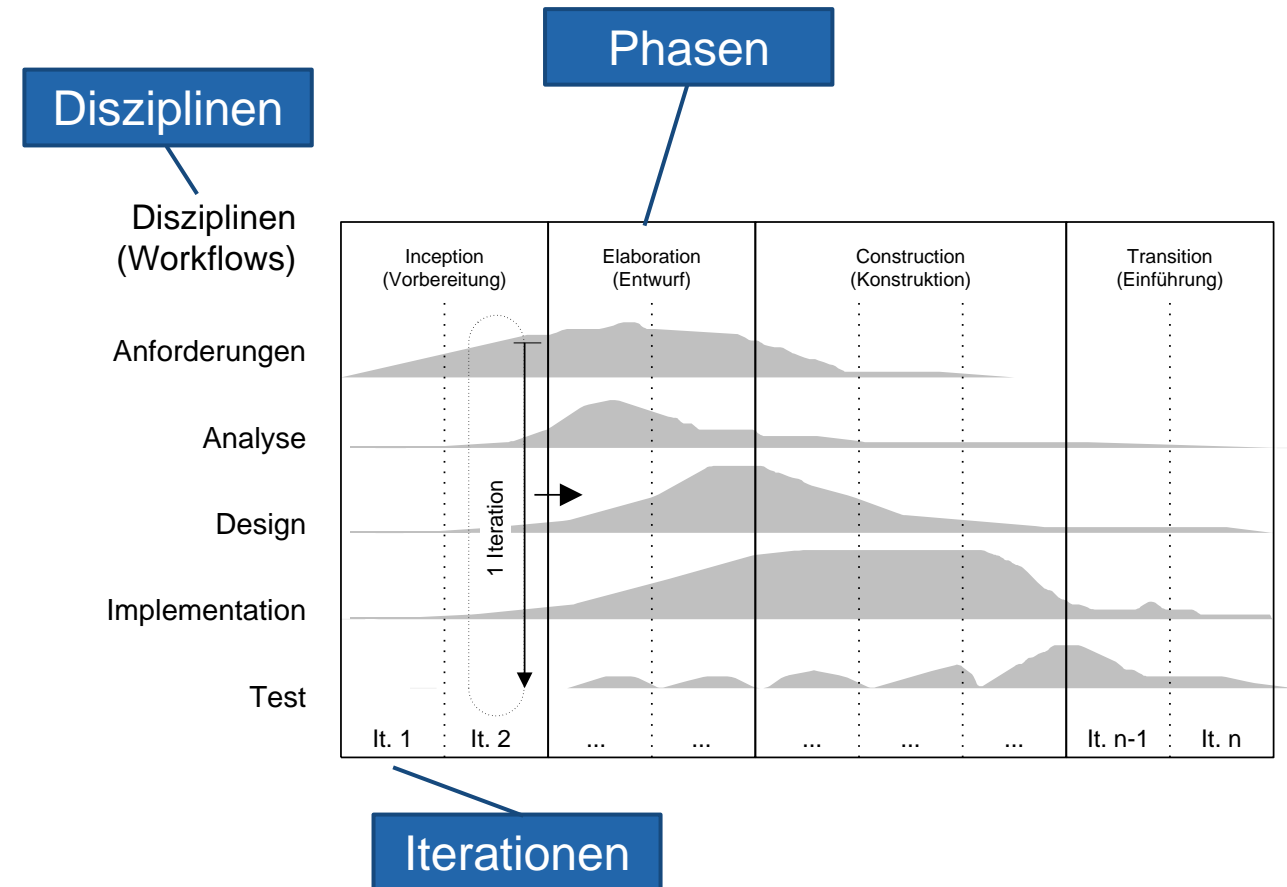
Agenda

1. Einführung in Projektmanagement
2. Softwareentwicklungsprozess für SWEN1/PM3
3. **Grobplanung**
4. Iterationsplanung
5. Risikomanagement
6. Wrap-up

Softwareentwicklungsprozess in SWEN1/PM3

Grobplanung

- Legt fest
 - Projektdauer
 - Meilensteine
 - Zeitpunkt
 - Ziele
 - Anzahl Iterationen pro Phase
- PM3
 - Projektdauer: 13 Wochen
 - 3 Meilensteine (M1-3)
 - Zeitpunkt vorgegeben
 - Ziele vorgegeben
 - Iterationsdauer: 2 Wochen (6 Iterationen)



- Für jeden Meilenstein:
 - Start und Ende (PM3: vorgegeben gemäss Wochenplan)
 - Grobe Beschreibung, was Sie bis dann erreichen möchten (vorgegeben in PM3)
 - Anzahl Iterationen mit ihrer Länge in Wochen (PM3: vorgegeben 2 Wochen)
 - Aufwandschätzung in Ph (Personenstunden)
(PM3: ableitbar aus den Anzahl ECTS * 30 Ph * Anz. Mitarbeitende im Team)
- Für jede Iteration: kurze Beschreibung der Ziele
 - Liste der Anwendungsfälle (Use Cases) / Risiken, die bearbeitet werden
 - Liste der Artefakte, die bearbeitet werden

Anwendungsfälle (Use Cases) (1/3)

- Textuelle Beschreibung einer konkreten Interaktion eines bestimmten Benutzers mit dem zukünftigen System
 - Aus Sicht des Akteurs
 - Enthalten implizite und explizite Anforderungen
 - Beschreiben das Ziel des Benutzers (= Grund für die Anforderungen)
 - Beschreiben den Kontext
- Beispiele: Geld abheben, Spiel spielen, Bestellung erfassen

Anwendungsfälle (Use Cases) (2/3)

- **Use Cases (UCs)** bilden in iterativen SWE-Prozessen eine **zentrale Rolle**
 - **Funktionale Anforderungen** werden hauptsächlich mit UCs dokumentiert
 - Mittels UCs können Anforderungen einfach mit dem Kunden diskutiert werden
 - UCs sind ein wichtiger Teil der **iterativen Projektplanung**
 - Projekt wird entlang von UCs geplant
 - UC-Realisierungen **bestimmen die Lösungsarchitektur** und treiben das Lösungsdesign
 - UCs werden für **funktionale Systemtests** eingesetzt
 - UCs bilden die **Basis für Benutzerhandbücher**

Anwendungsfälle (Use Cases) (3/3)

- Für die Grobplanung werden nur die **wichtigsten Use Cases mit einem Namen** identifiziert.
- In der LE03 werden dann Use Cases detailliert eingeführt zur Anforderungsanalyse.
- Schlechte UC-Namen
 - Initialisierung
 - Einloggen
 - Preis eintippen
 - Einkäufe machen
 - Kasse bedienen
- Gute UC-Namen
 - System initialisieren
 - System aufstarten
 - Artikel erfassen
 - (Einen) Einkauf erfassen

Beispiel: Grobplanung

Wird erst am Ende der
Iteration ausgefüllt

| Meilen-stein | I# | Start | Ende | Gep-lant [h] | Ist [h] | Ziele |
|----------------------------|-----------|-----------|----------------|--------------|---------|---|
| | #1 | 1.7.2015 | 30.7.2015 | 400 | 410 | Vision, GUI-Prototype |
| Projekt-skizze | M1 | | Ende #1 | | | Stakeholder-Agreement über Vision, Projektziele und Umfang |
| | #2 | 1.8.2015 | 31.8.2015 | 350 | 400 | Use Case Zum Zielort navigieren (Standardablauf) |
| | #3 | 1.9.2015 | 30.9.2015 | 400 | 410 | Use Case Gerät mit Smartphone koppeln (Standardablauf) |
| Lösungs-architektur | M2 | | Ende #3 | | | Vision, Ziele und Anforderungen stabil (80%), verifizierte Softwarearchitektur |
| | #4 | 1.10.2015 | 30.10.2015 | 450 | 450 | Use Case Power-on und Initialisierung, UC Zum Zielort navigieren |
| | #5 | 1.11.2015 | 30.11.2015 | 400 | 410 | Use Case Smartphone koppeln |
| | #6 | 1.12.2015 | 23.12.2015 | 200 | 190 | Use Case Karten aktualisieren |
| Beta-Release | M3 | | Ende #6 | | | Feature complete, Beta-Release stabil |

Grobplanung in PM3

- Vorgehen für Grobplanung in PM3
 - Projektdauer und Termine der Meilensteine im Grobplan eintragen gemäss Vorgabe
 - Iterationsdauer bzw. Anzahl Iterationen pro Phase festlegen
 - Anwendungsfälle (UCs) identifizieren (nur mit Titel bzw. Namen festhalten)
 - Liste der UCs priorisieren
 - Meilensteinvorgaben für M1-3 studieren gemäss Auftrag Projektergebnisse
 - Arbeiten auf die verschiedenen Iterationen verteilen
 - Arbeiten an UCs (Analyse, Design, Implementation, Test) anhand der Priorität der UCs
 - Arbeiten im Zusammenhang mit Risiken
 - Arbeiten an den geforderten Artefakten für die Meilensteine

Agenda

1. Einführung in Projektmanagement
2. Softwareentwicklungsprozess für SWEN1/PM3
3. Grobplanung
4. **Iterationsplanung**
5. Risikomanagement
6. Wrap-up

Iterationsplanung

- Legt im Detail fest, wer was macht in der nächsten Iteration
 - Arbeitspakete definieren gemäss verfügbaren Zeitbudget
 - Aufwand schätzen
 - Arbeitspaket einem oder mehreren Projektmitgliedern zuweisen
- Basierend auf
 - Grobplanung
 - Resultaten der letzten Iteration
 - Verfügbarkeit/Fähigkeiten der Teammitglieder
- Jedes Teammitglied ist verantwortlich für die ihm zugeteilten Arbeitspakete

Beispiel: Iterationsplan

| # | Task | Geplant [h] | Ist [h] | Verantwortlich |
|---|--|-------------|------------|----------------|
| 1 | Vision erstellen | 100 | 110 | PL |
| 2 | Analyse und Use Case Zum Zielort navigieren detaillieren | 100 | 80 | Developer 1 |
| 3 | Analyse und Use Case Gerät mit Smartphone koppeln detaillieren | 100 | 110 | Developer 2 |
| 5 | Entwicklungsumgebung und Environment aufsetzen | 50 | 50 | Developer 3 |
| 6 | UI Prototype erstellen | 50 | 60 | Developer 3 |
| | Total | 400 | 410 | |

Wird erst am Ende der
Iteration ausgefüllt

Iterations-Review

- Am Schluss einer Iteration wird im Team ein (kurzer) Review durchgeführt (s. Auftrag Iterations-Review)
 - Was wurde erreicht und was nicht?
 - Was sind die Learnings aus dieser Iteration?
 - Was soll verbessert werden in der/n nächsten Iteration/en?
 - Was sind die aktuellen Risiken?
 - Was ist in der nächsten Iteration geplant?

Iterations-Reporting

- Für Stakeholders
- Die (Nicht-) Erfüllung von Arbeitspaketen wird vermerkt
- Die effektiven Aufwände pro Arbeitspaket werden notiert (im Iterationsplan)
- Der Grobplan ist nachgeführt
- Die wichtigsten Punkte des Iterations-Reviews werden schriftlich festgehalten

Agenda

1. Einführung in Projektmanagement
2. Softwareentwicklungsprozess für SWEN1/PM3
3. Grobplanung
4. Iterationsplanung
- 5. Risikomanagement**
6. Wrap-up

Risikomanagement

- Risiko
 - Ein Ereignis mit negativen Konsequenzen (Schaden), das mit einer bestimmten Wahrscheinlichkeit eintritt.
- Projektrisiko
 - Risiko, das das Projekt an sich betrifft (Machbarkeit, Zeitplan, Kosten,...)
 - Andere Risiken: Marktrisiken, Betriebsrisiken, Haftungsrisiken, ...
- Projektrisiken möglichst früh identifizieren, angehen und überwachen
- Risikoliste führen und aktualisieren
 - Risiko benennen und allenfalls beschreiben
 - Eintretenswahrscheinlichkeit und Schadenspotential abschätzen
 - Massnahmen zur Minderung definieren

Risk Management (ISO 31000)

- Risk Assessment
 - Risk Identification
 - Was sind die Risiken und ihre Ursachen
 - Risk Analysis
 - Ursachen
 - Abhängigkeiten
 - Konsequenzen
 - Wahrscheinlichkeit
 - Risikomatrix
- Risk Evaluation
 - Priorität der Behandlung eines Risikos
- Risk Treatment
 - Risiko bewusst eingehen
 - Risiko einkalkulieren
 - Risiko vermeiden, abbrechen
 - Ursache des Risikos eliminieren
 - Risiko-Wahrscheinlichkeit vermindern
 - Impact vermindert
 - Risiko teilen oder auf andere übertragen

| Eintritts- wahrscheinlich- keit | Konsequenzen, Schaden | | |
|---------------------------------------|--------------------------|--------|------|
| | Tief | Mittel | Hoch |
| Hoch | M | H | H |
| Mittel | T | M | H |
| Tief | T | T | M |

- Risikostufen
 - T
 - Kleines Risiko
 - Routinemässige Behandlung
 - M
 - Mittleres Risiko
 - Möglicher Handlungsbedarf
 - Verantwortlichen bestimmen
 - H
 - Hohes Risiko
 - Handlungsbedarf
 - Managementaufgabe

Beispiel: Risikoliste

| # | Name | Beschreibung | Wahr- scheinlich- keit | Schadens- potential | Priorität | Massnahmen |
|---|--------------------------------------|---|------------------------------|------------------------|-----------|--|
| 1 | Firewall | Chat funktioniert nicht via Firewalls | Hoch | Hoch | 1 | Geeignetes Transportprotokoll wählen |
| 2 | Effizienz | Geforderte Antwortzeiten können nicht erreicht werden. | Mittel | Hoch | 2 | Prototyp entwickeln |
| 3 | Schnittstellen des externen Dienstes | Die Schnittstellen des externen Dienstes können sich in inkompatibler Art ändern. | Tief | Mittel | 3 | Design Massnahmen treffen, um die Änderungen so lokal wie möglich zu halten. |

Agenda

1. Einführung in Projektmanagement
2. Softwareentwicklungsprozess für SWEN1/PM3
3. Grobplanung
4. Iterationsplanung
5. Risikomanagement
6. **Wrap-up**

Wrap-up (1/2)

- Als Projektmanagement (PM) wird das **Initiieren, Planen, Steuern, Kontrollieren** und **Abschliessen von Projekten** bezeichnet.
- Wichtigste Aspekte dabei
 - **Ziel (Produkt, Qualität)**
 - **Mittel (Budget, Personal)**
 - **Zeit**
 - **Risiken**
- Wichtige Eigenschaften moderner Entwicklungsprozesse
 - **Iterativ-inkrementell**
 - **risiko-basiert**
 - **anwendungsfall-getrieben**

Wrap-up (2/2)

- Wichtige Artefakte für das Projektmanagement eines iterativ-inkrementellen Software-Projektes sind:
 - Grobplan
 - Iterationsplan, Iterations-Review
 - Risikoliste
- Ein iterativ-inkrementell Software-Projekt muss als **Lernprozess** verstanden werden.

Achtung: Ohne gemeinsame Iterationsplanungen und Iterations-Reviews im Team kann aber nicht gelernt werden!