universitätbonn ROOTS

Kapitel 3. Projektmanagement

– Stand: 24.10.2011 –

Übersicht

Erste Doppelstunde

Kap. 3a Projektmanagement (1)

- Konzepte und Terminologie
- Projektmanagementpläne
- Projektverantwortlichkeiten
- Teamstrukturen
- Projektplanung
- Kommunikationsmanagement

Zweite Doppelstunde

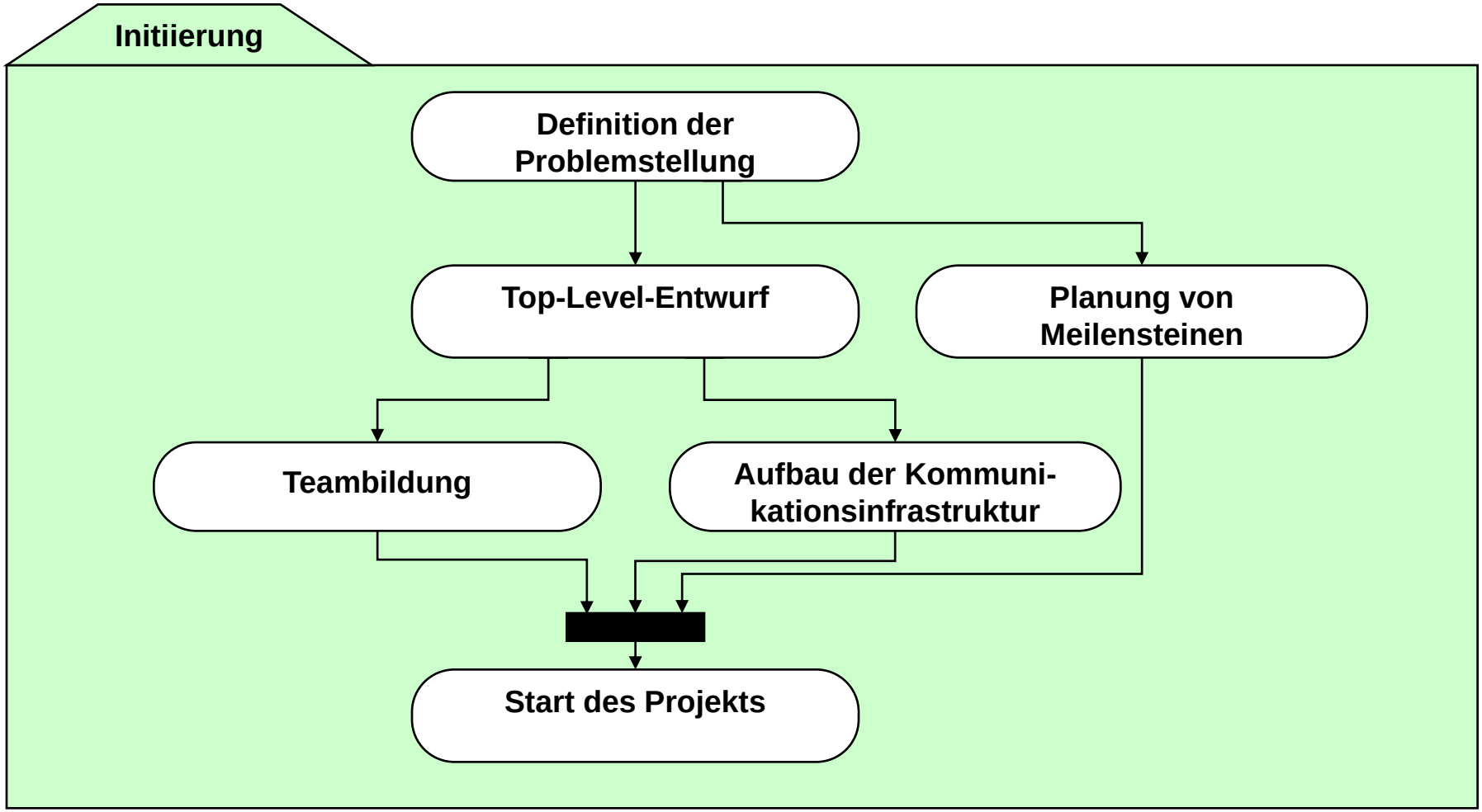
Kap. 3a Projektmanagement (2)

- Abhängigkeiten
- Zeitplan
- Zeitplanungswerkzeuge

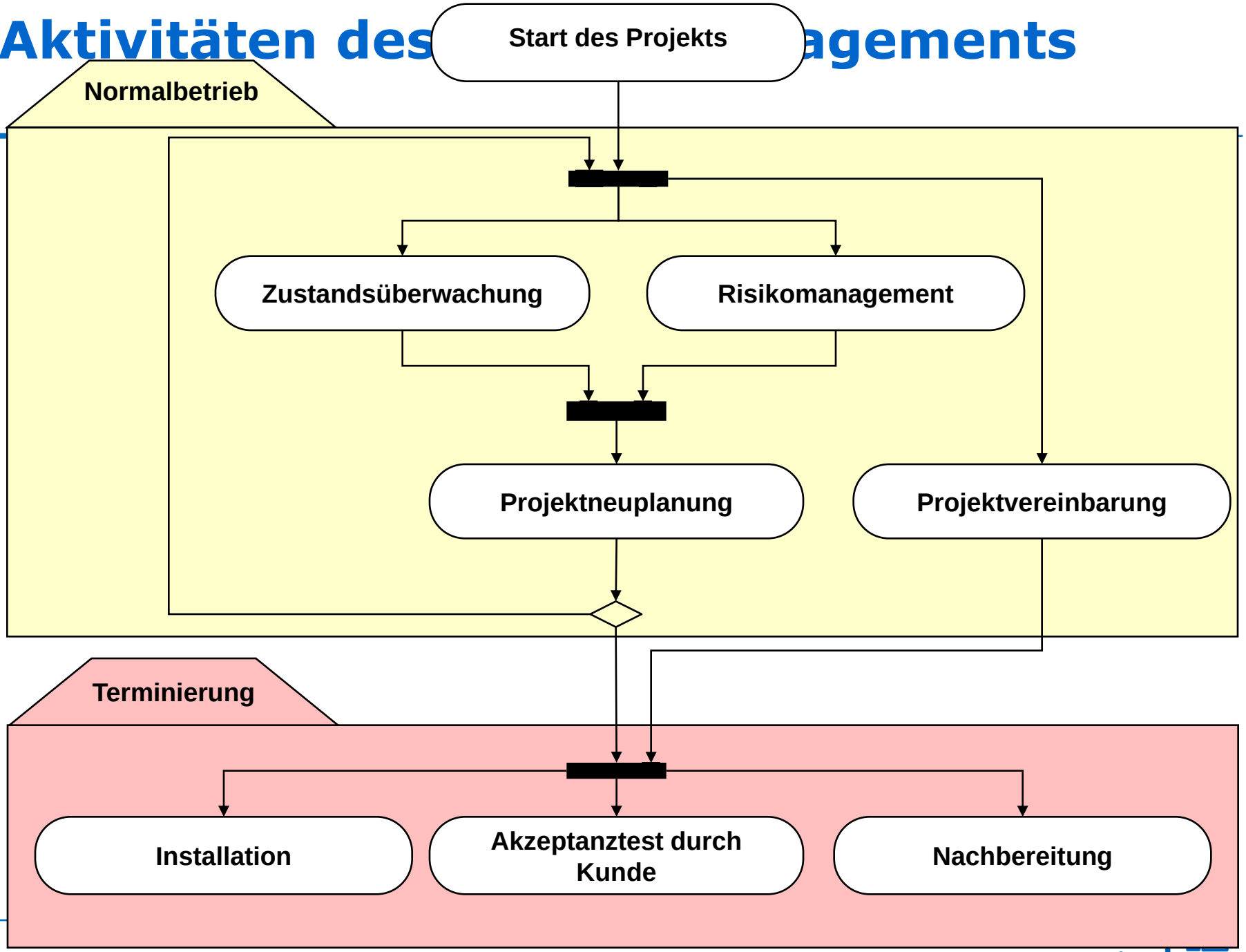
Kap. 3b Issue-based Change Management

- ✓ Configuration Management
- Issue Management → Jira
- Task-based GUIs → Mylin

Aktivitäten des Projektmanagements



Aktivitäten des agements

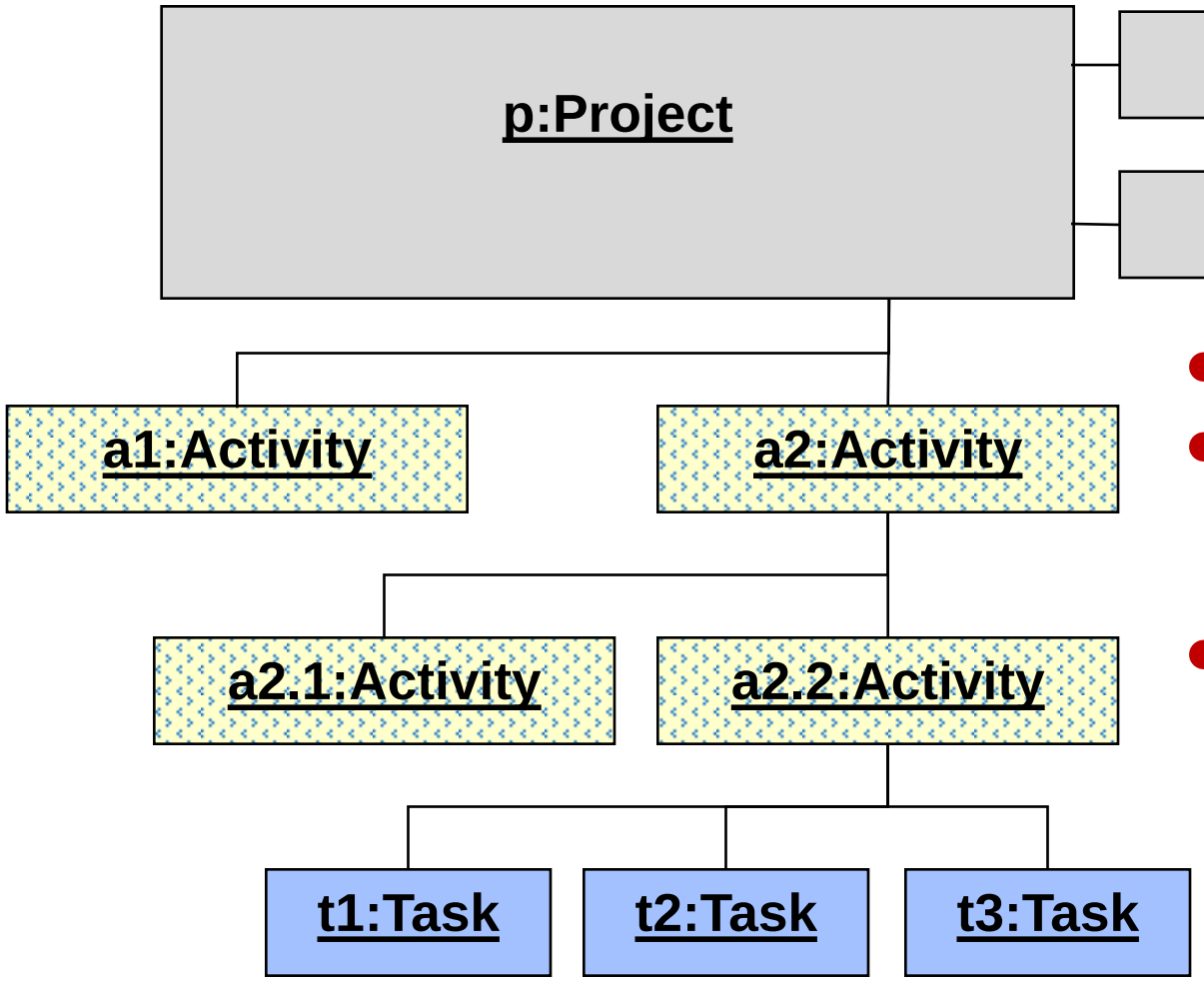


Projekt = Integrale Prozesse, Aktivitäten, Tasks, Action Items

Integrale Prozesse (IEEE 1074) / Funktionen (IEEE 1058)

- Tätigkeiten die die Dauer des gesamten Projekts umfassen und sich nicht zyklisch wiederholen
  - Projektmanagement
  - Konfigurationsmanagement (SCM)
  - Aufgabenmanagement (Issue Management)
  - Qualitätskontrolle (Verifikation und Validierung)
  - Dokumentation
  - Training
  - …

Aktivität



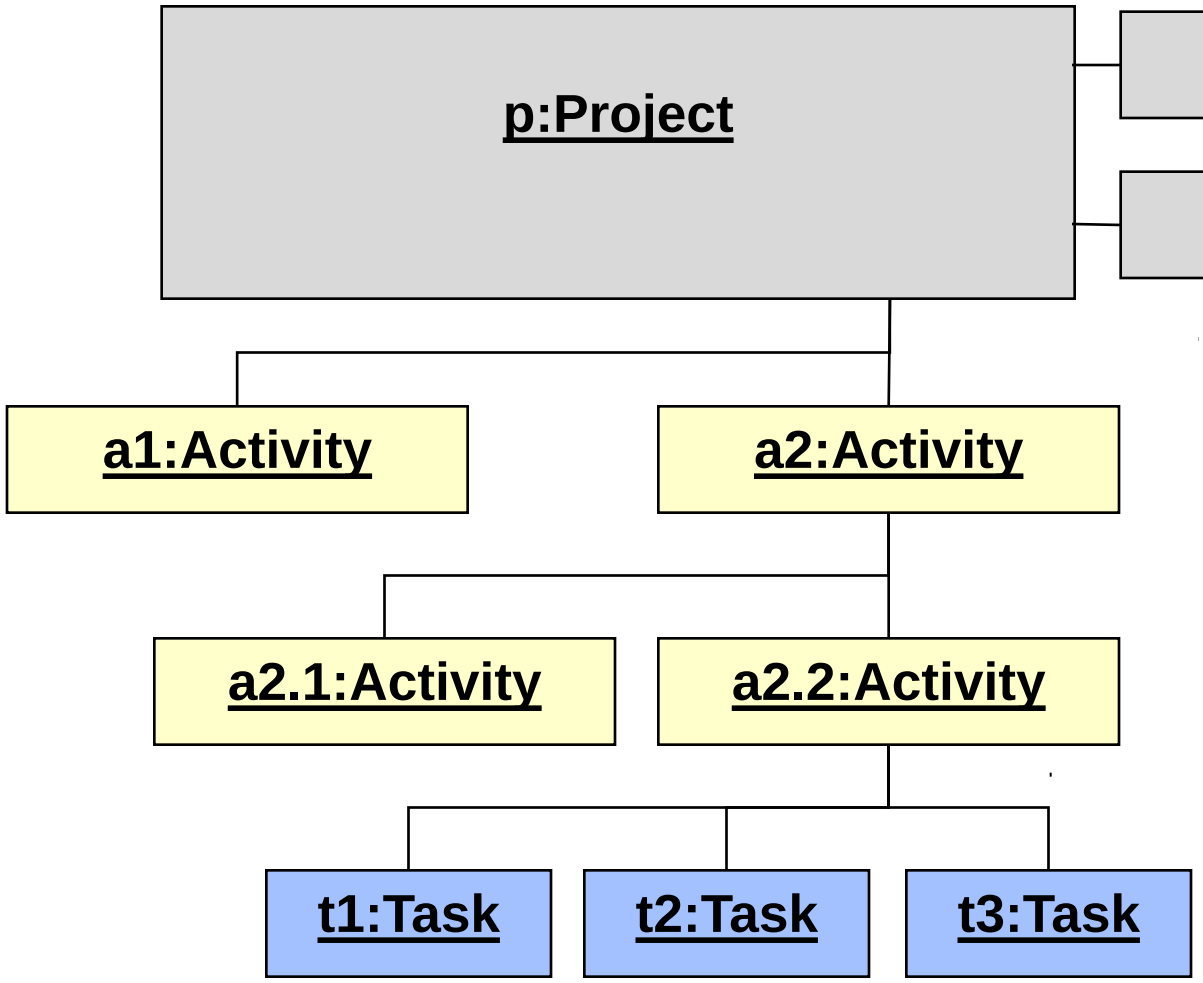
- Tritt in Phasen auf
- Besteht aus Teilaktivitäten oder Tasks
- Gipfelt in Projekt-Meilenstein

Beispiele für Aktivitäten („Workflows“)

- Planung
- Anforderungserhebung
- Anforderungsanalyse
- Systementwurf
- Objektentwurf
- Implementierung
- Testen
- Auslieferung

- Teil-Aktivitäten der Anforderungsanalyse
  - Verfeinern von Szenarios
  - Use-Case-Modell definieren
  - Objektmodell definieren
  - Dynamisches Modell definieren
  - Benutzerschnittstelle entwerfen

Aufgabe (“Task”)



- Kleinste Einheit von Arbeit, die noch Gegenstand des Managements ist
- Klein genug für adäquate Planung und Verfolgung
- Groß genug, um Mikromanagement zu vermeiden

Tasks (Aufgaben)

- Kleinste Einheit für Verantwortlichkeit des Managements
  - Atomare Einheit für Planung und Verfolgung
  - Endliche Dauer, benötigt Ressourcen, produziert verifizierbare Ergebnisse (Dokumente, Code)

- Spezifikation einer Task
  - Name, Beschreibung der zu leistenden Arbeit
  - Vorbedingungen, Dauer, benötigte Ressourcen
  - Erwartete Arbeitsergebnisse
  - Erfüllungs- / Akzeptanzkriterien für die Arbeitsergebnisse
  - Mit der Task verbundenes Risiko

Größe von Tasks

- Jede Entwicklungsaktivität identifiziert neue und modifiziert existierende Tasks.
- Zusammenhängende Tasks werden zu hierarchischen Mengen gruppiert.

- Tasks müssen in Größen aufgebrochen werden, die ein Monitoring zulassen.
  - Arbeitspakete entsprechen i.d.R. wohl-definierten Arbeitsanweisungen für einen Arbeiter und eine Woche (einen Monat).
  - Abhängig von der Art der Arbeit und davon, wie gut die Aufgabe verstanden wird.
- Die angemessene Größe von Tasks zu finden, ist problematisch.
  - Es ist evtl. anfänglich nicht bekannt, wie ein Problem in Tasks zu zerlegen ist.
  - Während der anfänglichen Planung sind Tasks notwendigerweise groß.
  - Aus TODO-Listen früherer Projekte lernen!

Beispiele für Tasks

- Teste das Subsystem “Bla”
- Unit test für Klasse „Foo“

- Schreibe das Benutzerhandbuch
- Schreibe ein Memo über „Linux vs. Windows“
- Schreibe ein Protokoll zur letzten Sitzung und verteile es.

- Entwickle den Projektplan
- Lege den Zeitplan für die „Code Review“ fest.

Action Item

- Definition: Ein Task, der einer Person zugeordnet wird, und der innerhalb einer Woche oder weniger erledigt sein muss.

- Action Items
  - Tauchen auf der Agenda im „Status“-Abschnitt auf
  - Klären: Wer? Was? Wann?

- Beispiel für Action Items:
  - Das VIP Team entwickelt einen Projektplan bis 18. Sep.
  - Florian erledigt Unit Tests für Klasse “Foo” bis nächste Woche.
  - Bob verschickt die nächste Agenda für das Simulationsteam bis 10. Sep. 12:00

universitätbonn ROOTS

Software Project Management Plan

Struktur eines Software Project Management Plans

Einstieg

1. Einführung

2. Projektorganisation

3. Organisatorischer Prozess

4. Technischer Prozess

5. Arbeitselemente, Zeitplan, Budget

Optionale Anlagen

SPMP Teil 0: Einstieg

- Titelblatt
- Revision sheet (update history)
- Vorwort: Umfang und Ziel
- Inhalts- sowie Tabellen- und Abbildungsverzeichnisse

SPMP Teil 1: Einführung

1.1 Projektübersicht

- Beschreibung des Projekts, Projektzusammenfassung.

1.2 Endergebnisse des Projekts

- Alle auszuliefernden Artikel, incl. Datum und Ort der Auslieferung

1.3 Evolution des SPMPs

- Pläne bzgl. erwarteter und unerwarteter Änderungen

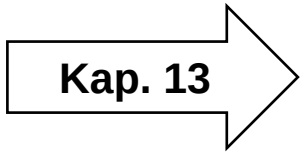
1.4 Referenziertes Material

- Vollständige Liste aller Materialien, die vom SPMP referenziert werden

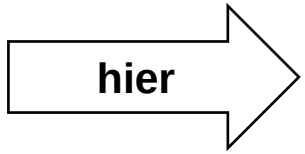
1.5 Definitionen und Akronyme

SPMP Teil 2: Projektorganisation

2.1 Prozessmodell
- Beziehungen zwischen Projektelementen

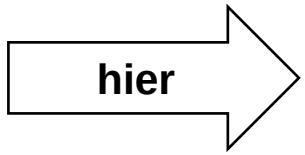


2.2 Organisatorische Struktur
- Interne Verwaltung, Organisationsdiagramm



2.3 Organisatorische Schnittstellen
- Beziehungen zu anderen Entitäten (außerhalb des Projekts)

2.4 Verantwortlichkeiten
- Rollen im Projekt und entsprechende Aktivitäten



2.1 Prozessmodell

- Zeigt Beziehungen zwischen
  - Funktionen, Aktivitäten, Tasks
  - Zeitplan, Meilensteinen
  - Releases
  - Reviews
  - Projektstrukturplan
  - Endergebnissen
- Visualisierung des Prozessmodells

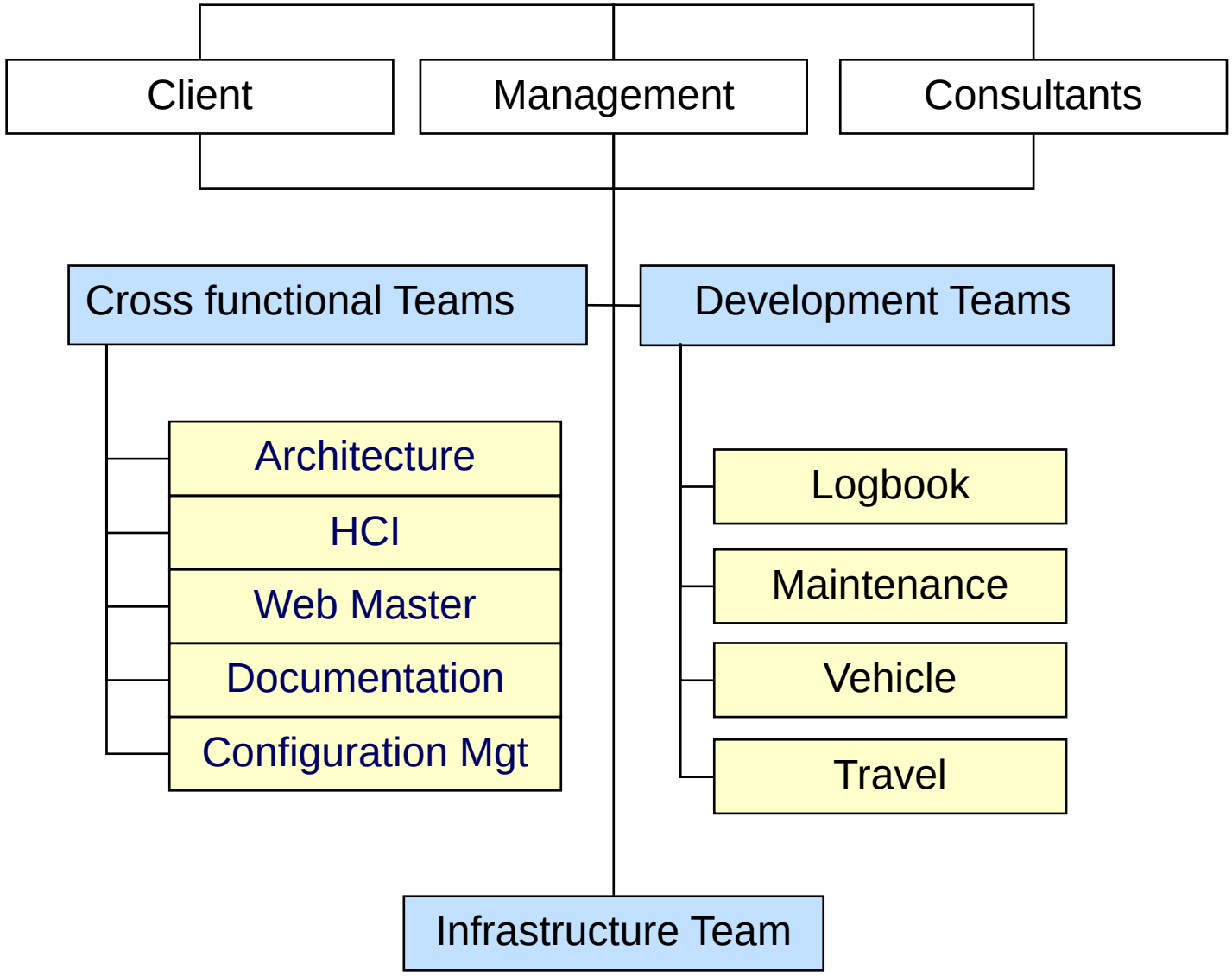
- Werkzeuge
  - MS Project (Microsoft)
  - Google „project management tool“ „project management software“
  - Gemeinsame Schwäche: Nicht für „agile“ Softwareprozesse geeignet

universitätbonn ROOTS

Software Project Managemet Plan:

2.2 Organisatiorische Struktur

Beispiel eines Organisationsdiagramms



Assoziationen in Organisationsstrukturen

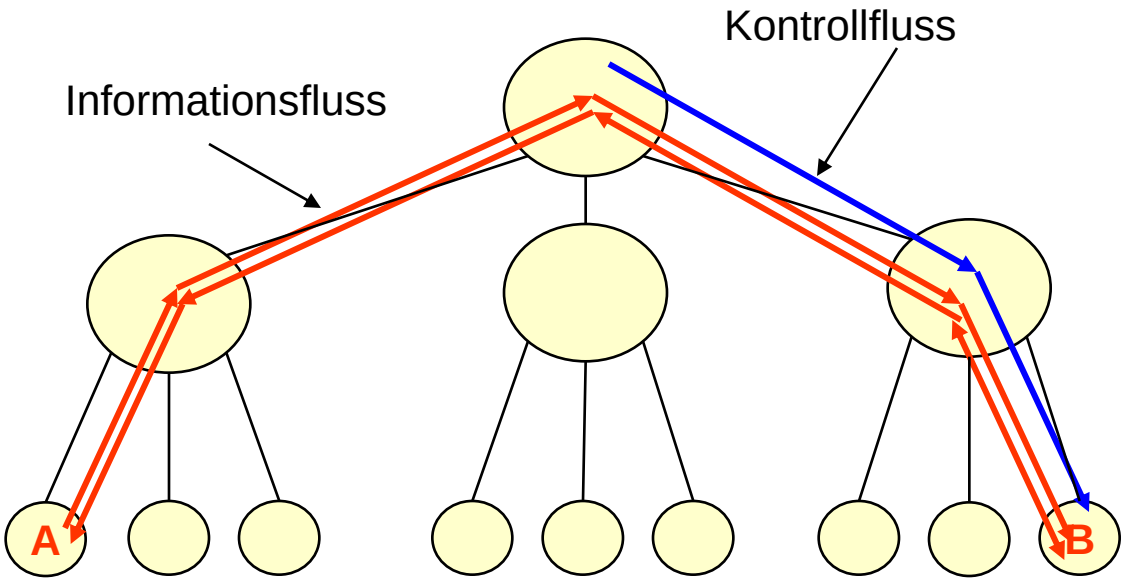
- Kommunikationsassoziationen
  - dienen dem Austausch von Informationen, die zur Entscheidungsfällung nötig sind (z.B. Anforderungen, Entwurfsmodelle, Issues...)
- Berichterstattungsassoziationen
  - dienen der Übermitteln von Statusinformationen
- Entscheidungsassoziationen
  - werden zum Propagieren von Entscheidungen verwendet

Betrachtungen zu Verwaltungsstrukturen

- Hierarchische Strukturen
  - *“erstattet Bericht”*, *“entscheidet”* und *“kommuniziert mit”* werden alle auf die selbe Assoziation abgebildet.
  - Funktionieren nicht gut zusammen mit iterativer und inkrementeller Softwareentwicklung.
  - Der Manager hat nicht notwendigerweise immer Recht

- Projektbasierte Strukturen
  - *“erstattet Bericht”*, *“entscheidet”* und *“kommuniziert mit”* sind unterschiedliche Assoziationen.
  - Abbau von Bürokratie reduziert Entwicklungszeit.
  - Es wird erwartet, dass auf jeder der Ebenen Entscheidungen getroffen werden.
  - Schwieriger zu verwalten aber oft effektiver

Hierarchische Kommunikationsstruktur



Leitender Angestellter

First Level Manager
(“Front-Line Manager”)

Projektteilnehmer

Komplizierter Informationsfluss: A möchte mit B reden.
Komplizierter Entscheidungsfluss: A möchte, dass B etwas bestimmtes tut.

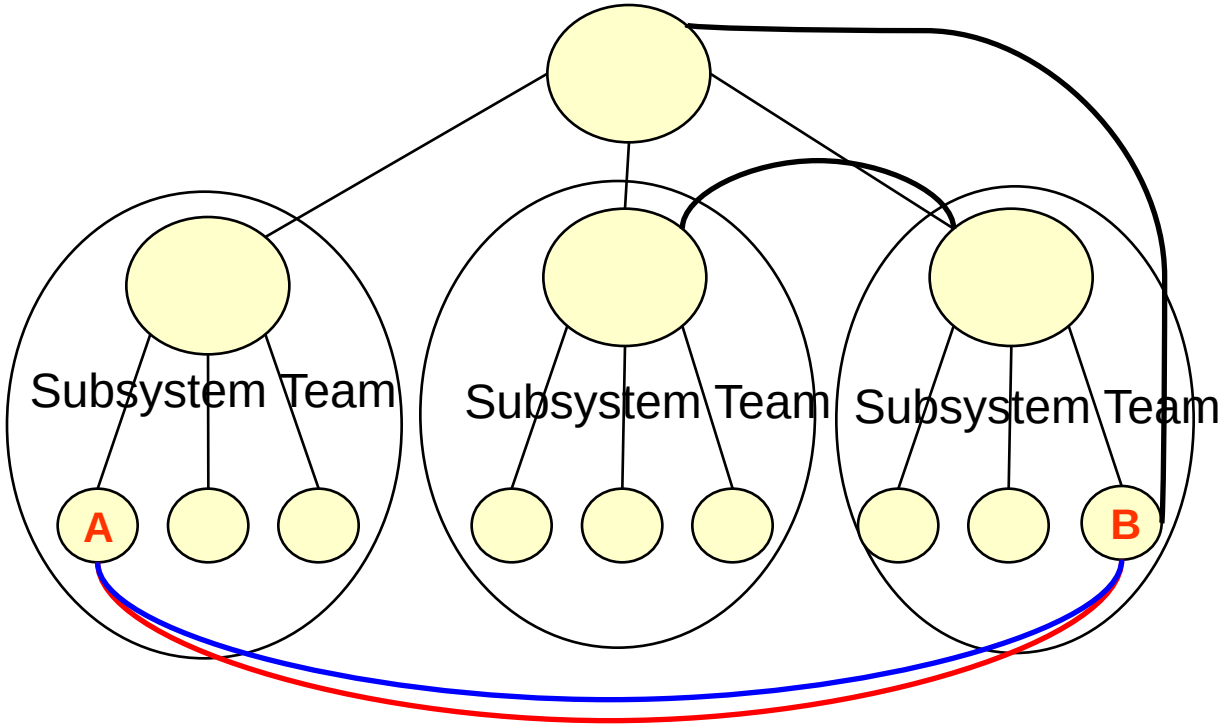
Organisationsgrundlage:
Komplizierter Informations- und Kontrollfluss über hierarchische Grenzen hinweg.

Hierarchische Struktur

- Projekte mit einem hohen Grad an Sicherheit, Stabilität, Einheitlichkeit und Wiederholung
  - Benötigen wenig Kommunikation
  - Rollen sind klar definiert.

- Wann?
  - Je mehr Leute im Projekt mitarbeiten, desto größer ist die Notwendigkeit einer formalen Struktur.
  - Evtl. besteht der Kunde darauf, dass das Testteam unabhängig vom Entwurfsteam ist.
  - Projektleiter besteht auf einer Struktur, die sich zuvor als erfolgreich erwiesen hat.

Projekt-basierte Kommunikationsstruktur



Direkter Informationsfluss: A will mit B reden.

Direkter Entscheidungsfluss: A möchte, dass B etwas bestimmtes tut.

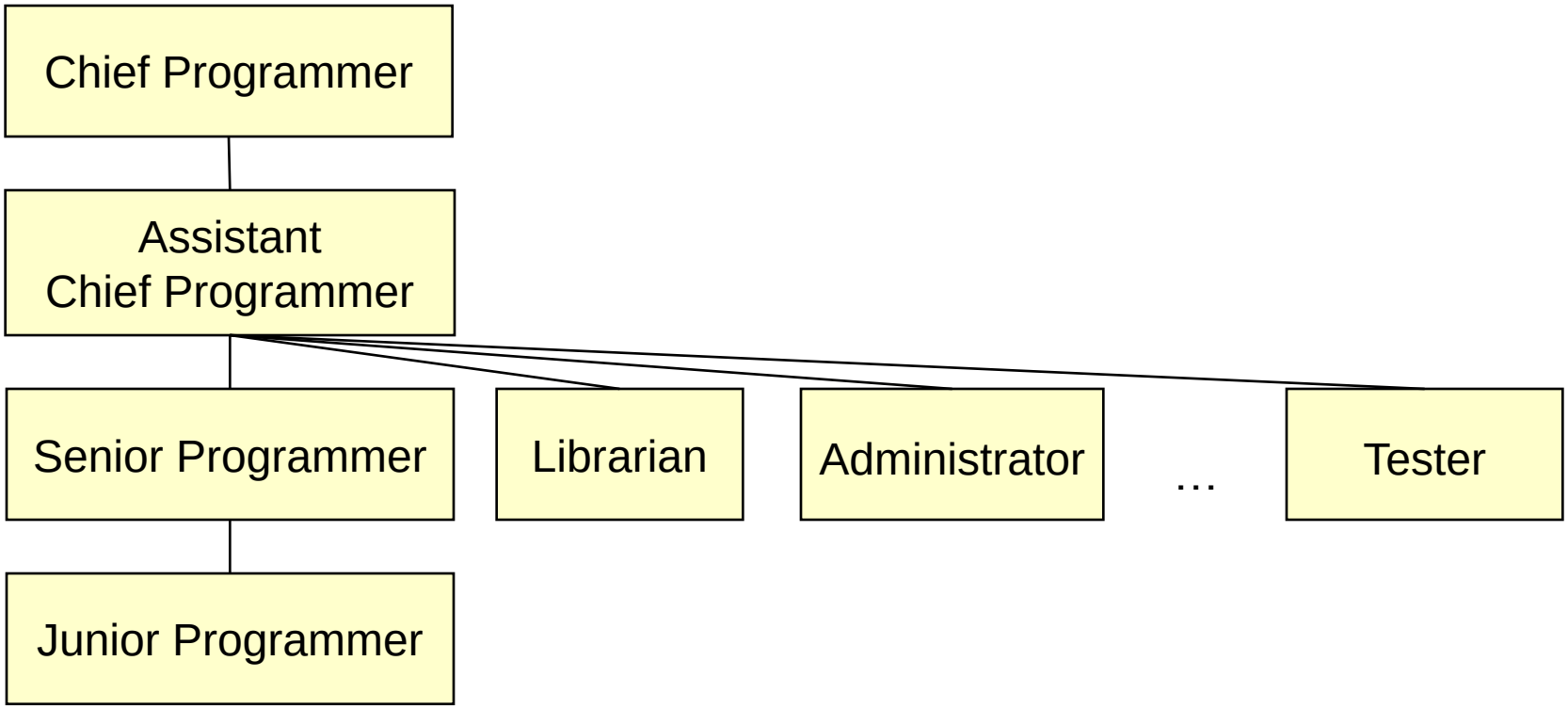
Organisationsgrundlage: Nicht-linearer Informationsfluss zwischen dynamisch formierten Einheiten.

Projekt-basierte Struktur

- Projekt mit einem Grad an Unsicherheit
  - Offene Kommunikation unter den Teilnehmern ist erforderlich.
  - Rollen werden auf Projektbasis definiert.

- Wann anwenden?
  - Anforderungen ändern sich während der Entwicklung.
  - Neue Technologie entwickelt sich während des Projekts.

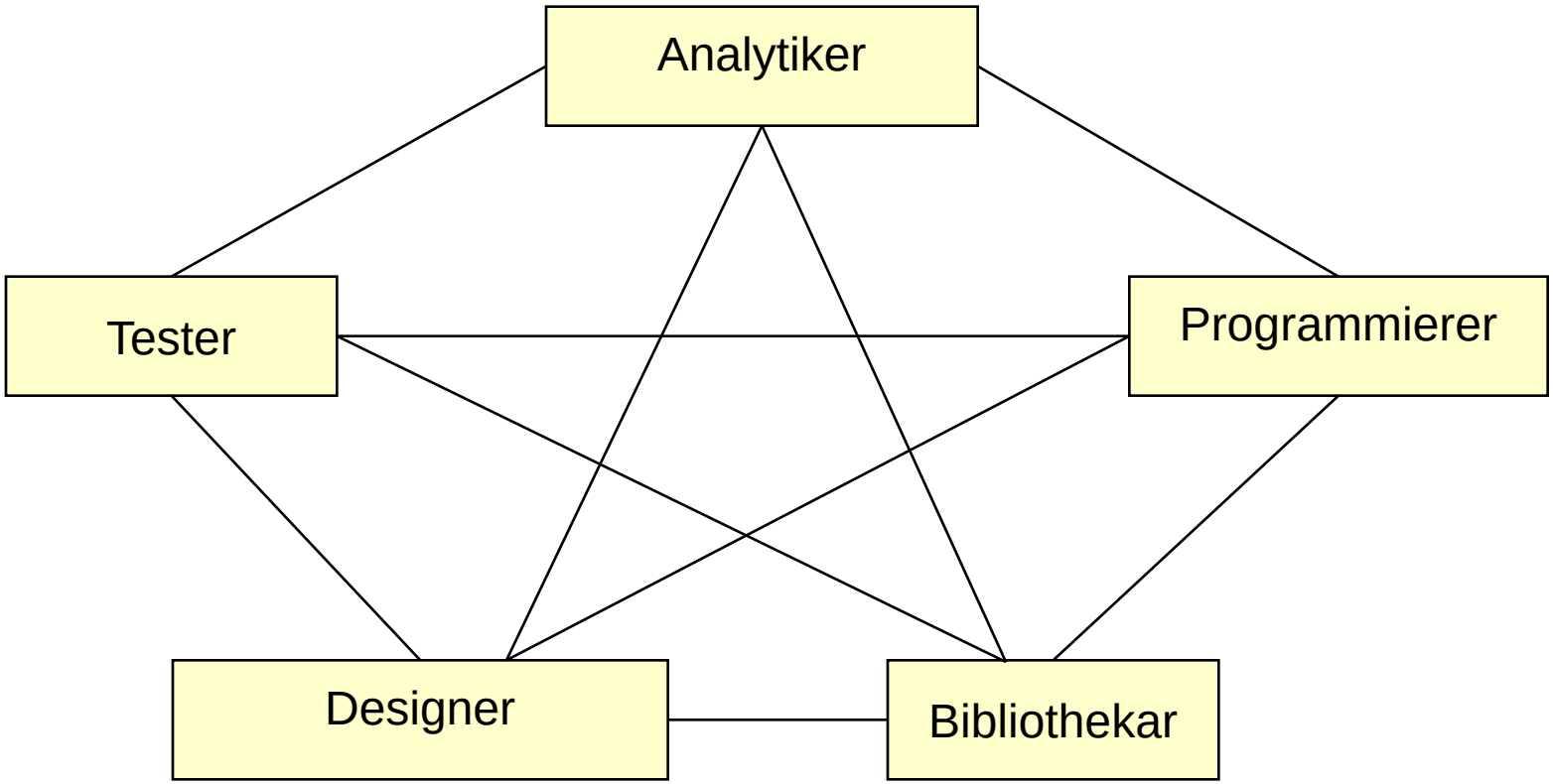
Beispiel einer hierarchischen Organisation: „Chief Programmer“ (Mills)



Beispiel einer hierarchischen Organisation: „Chief Programmer“ (Mills)

- Der Chief Programmer
  - ist der Hauptverantwortliche für den Entwurf und die Implementierung.
  - implementiert selbst kritische Funktionalität,
  - weist den anderen Entwicklern Aufgaben zu und kontrolliert den Arbeitsfortschritt.
- Vorteile
  - Schnelle Enticklung durch wenig Kommunikation, wenige Teammeetings, kurze Entscheidungswege
- Nachteile
  - Chief Programmer ist Engpass der Entwicklung, muss Manager und Hacker sein, demotiviert das Team
- Erfnder: Harlan D. Mills
  - Erstmals in den 70er Jahren bei IBM eingesetzt.
- Onlinematerial: http://everything2.com/index.pl?node_id=1292141

Eine andere Organisationsform: „Egoless Programming Team” (Weinberg)



Eine andere Organisationsform: „Egoless Programming Team” (Weinberg)

- Programmierer identifizieren sich oft zu stark mit "ihrem" Code
  - Akzeptieren keine Änderungen, testen nicht gründlich genug,…
- Egoless programming
  - “Collective code ownership”
  - Motiviere Teammitglieder Fehler in allen Modulen zu suchen
  - Fehler werden als normal betrachtet
  - Starke Gruppenidentität
  - Kleine Gruppen (<= 10)
- Vorteile von Egoless Teams
  - Sehr produktiv
  - Arbeiten am besten bei komplexen Problem
  - Haben sich im Forschungsumfeld als erfolgreich herausgestellt
- Problem
  - schwer planbar
- Vergleiche: eXtreme Programming (Kap. 14)

Teambildung

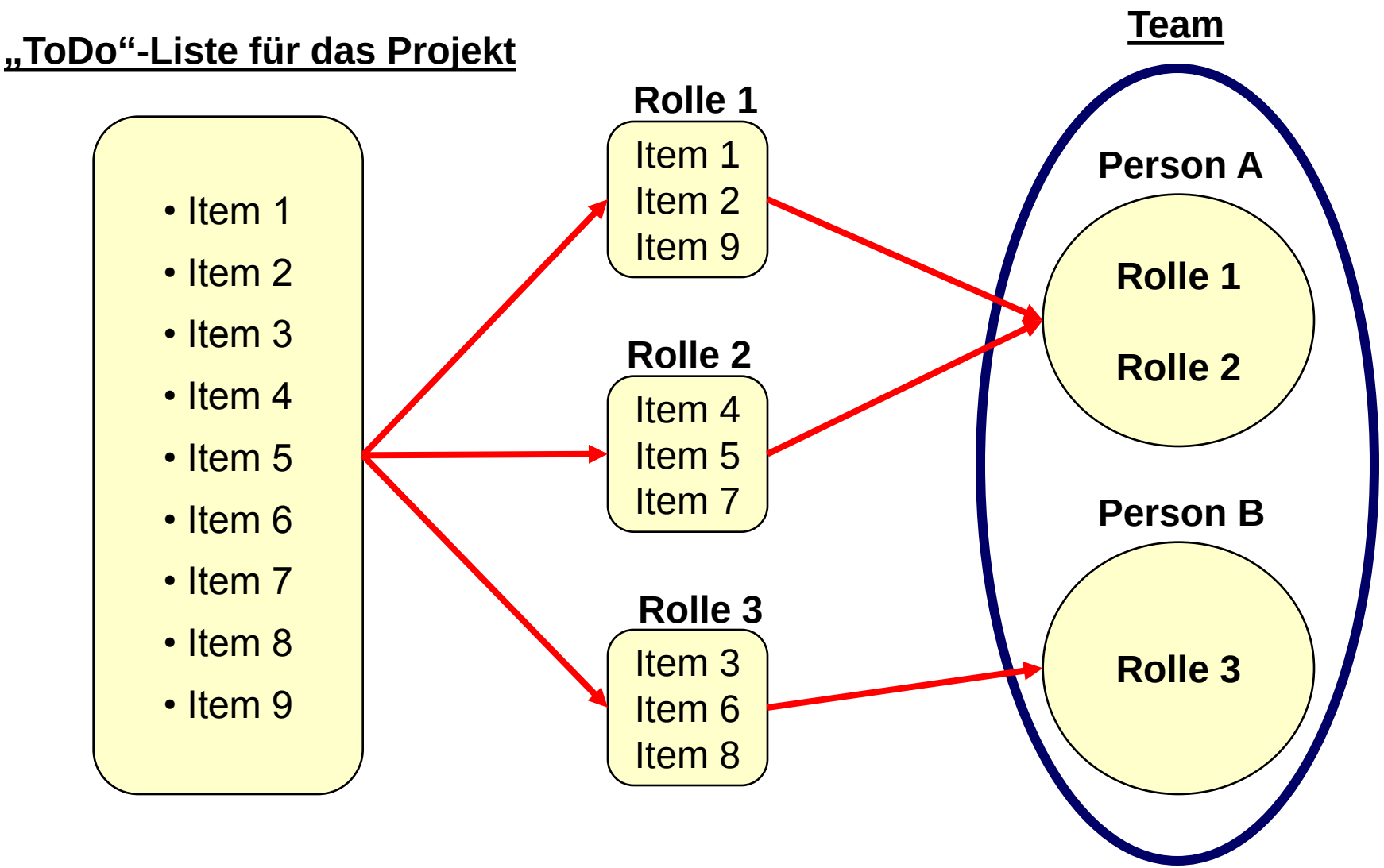
- Teambildung findet nach dem Top-Level-Entwurf statt.

- „Top-Level-Entwurf“
  - “Grobe” Subsystemzerlegung (vor der Anforderungsanalyse)
  - Geschieht vor der eigentlichen Entwicklung

- Heuristiken:
  - Ein Team für jedes (vermutete) Subsystem
  - Eine Funktionsübergreifende Aufgabe pro Team
  - 5-7 Mitglieder pro Team

- Teams müssen üblicherweise angepasst werden, sobald die tatsächliche Subsystemdekomposition fest steht (nach dem Systementwurf)

Zuweisung von Verantwortung



Mögliche Abbildungen von Rollen auf Personen

- Viele-zu-Wenige
  - Jeder Projektteilnehmer nimmt mehrere Rollen an („Hüte“).
  - Gefahr des „Über-Engagements“
  - „load balancing“ ist notwendig.

- Viele-zu-“Zu Viele“
  - Einige Teilnehmer spielen keine signifikante Rolle
  - „Zuschauer“
  - Verlieren den Anschluss an das Projekt

universitätbonn ROOTS

SNMP 2.4: Verantwortlichkeiten / Rollen

Rollen in einem Projekt

- Planer
- Analytiker
- Designer
- Programmierer
- Tester
- Wartungspersonal

- Gruppenleiter
- Liaison (Verbindungsmann)
- Projektleiter

- Dokument Editor
- Web Master
- Konfigurationsmanager

- Wissenspromotor
- Prozesspromotor
- Coach / Ausbilder

Rollen in einem Projekt

- Rollen im Management
  - Organisation und Durchführung des Projekts unter Berücksichtigung von Rahmenbedingungen. Beispiele: Projektleiter, Teamleiter.
- Rollen in der Entwicklung
  - Spezifikation, Entwurf und Konstruktion der Subsysteme. Beispiele: Analytiker, Systemarchitekt, Programmierer.
- Funktionsübergreifende Rollen
  - Koordination mehrerer Teams. Beispiele: API Ingenieur, Konfigurationsmanager, Tester
- Beratende Rollen
  - Unterstützung in Gebieten, in denen es den Projektteilnehmern an Erfahrung mangelt. Beispiele: Anwender, Kunde, ein Spezialist auf dem Anwendungsgebiet (Problemdomäne), Technischer Berater (Lösungsdomäne)

Wissenspromoter („Technologe“)

- Propagiert Änderungen, die in der Anwendungs- oder Lösungsdomäne auftreten

- **Aufgaben**
  - Schrittweise Informationen aneignen
  - Nutzen und grenzen neuer Technologie erkennen und mit den anderen Entwicklern über den Einsatz dieser Technologie diskutieren

- **Beispiel auf Projektebene:** Systemarchitekt
  - Berichtet dem Projektleiter
  - Hat keine direkten Untergebenen, der ihm Bericht erstattet
  - Hat das letzte Wort in jeder technischen Entscheidung

- **Beispiel auf Unternehmensebene:** Technischer Direktor

Prozesspromoter

- Der Prozesspromoter
  - ist mit den Prozessen und dem Prozedere des Projekts vertraut.
  - ist dafür verantwortlich einen Konsens über die langfristigen Ziele zu erreichen.

- **Aufgaben**
  - Verbindung zwischen Power- und Wissenspromoter, die oft nicht die selbe Sprache sprechen
- **Beispiel auf Projektebene**
  - Entwicklungsleitung. Verantwortlich für die administrativen Aspekte eines Projekts; beinhaltet Planung, Definition von Meilensteinen, Budgetplanung und Kommunikationsinfrastruktur
- **Beispiel auf Unternehmensebene**
  - Leiter der Technologieabteilung

Rollen im Projekt: Coach

- auf Probleme einzelner Teams eingehen
- Die wöchentliche Teamreports durchsehen
- Den wöchentlichen Projektmeetings beiwohnen
- Treffen mit dem Kunden arrangieren.
- Darauf bestehen, dass die Richtlinien befolgt werden
- Präsentationen zuweisen
- Konflikte lösen, wenn sie nicht teamintern gelöst werden können

Rollen im Projekt: Gruppenleiter

- Verantwortlich für die Kommunikation innerhalb des Teams (Meeting Management: Primärer Moderator)
  - Führt das wöchentliche Projektmeeting
  - Schickt die Agenda vor dem Treffen an alle Beteiligten
  - Definiert und verfolgt Action Items (wer, was, wann)
  - Misst Fortschritt (Setzt Meilensteine durch)
  - Abgeschlossene Arbeitspakete an das Projektmanagement ausliefern
  - Trägt Probleme und Status des Teams dem Projektleiter vor
- Die Rolle des Gruppenleiters kann (sollte!?!) unter den Teammitgliedern durchrotiert werden.

Gruppenleiter: Erstellen einer Tagesordnung

- Zweck des Treffens
- Angestrebtes Ergebnis
- Informationsaustausch
- Informationsverarbeitung
- Meetingkritik

Status-Bericht über Action Items des vorigen Treffens

Issues (Siehe voriges Treffen & Boards)

Rollen im Projekt: Liaison

- Verantwortlich für die Kommunikation zwischen den Teams
  - Macht öffentliche Schnittstellen von Subsystemen, die vom Team entwickelt werden für das Architekturteam verfügbar
  - Koordiniert teamübergreifende Aufgaben mit anderen Teams
- Verantwortlich für „Verhandlungen“ zwischen den Teams
- Beispiele: API Ingenieur, Konfigurationsmanager

Rollen im Projekt: Planer

- Plant und verfolgt Aktivitäten eines Teans und hat die folgenden Aufgaben:
- Definiert den Projektplan für das Team
  - PERT Diagramm, Ressourcetabelle und GANTT Diagramm, das die Arbeitspakete zeigt.
  - Gibt den Projektplan in das jeweilige Projektmanagementwerkzeug ein.
- Stellt den Projektplan dem Management zur Verfügung
- Erstattet dem Projektleiter bericht über den Teamstatus

Rollen im Projekt: Document Editor

- Sammeln, Korrekturlesen und verteilen von Dokumentation
- Legt Dokumentation des Teams dem Architekturteam vor
- Sammelt Tagesordnungen
- Führt Protokoll bei Teammeetings

Web Master

- Pflegt die Homepage des Teams inklusive…
  - … Liste stattgefundener Meetings
  - … Rationale zum Entwurf
- Veröffentlich Informationen über die Teammeetings auf der Homepage des Teams
  - Muss Tagesordnung, Protokoll, Action Items und Issues enthalten
  - Möglichkeiten:
    - ⇨ Ein HTML Dokument per Meeting, mit Verlinkung. (Pflegen → eine Rolle)
    - ⇨ Separate HTML Dokumente für Tagesordnung, Protokoll, etc. (Pflegen → mehrere Rollen)

Date	Agenda	Minutes	Action Items	Issues
9/9/96	Agenda	Minutes	Action Items	Issues
9/16/96	Agenda	Minutes	Action Items	Issues

universitätbonn ROOTS

SPMP Teil 3: Organisatorischer Prozess

SPMP Teil 3: Organisatorischer Prozess

3.1 Verwaltungsziele und -prioritäten

- Philosophie, Ziele, Prioritäten

3.2 Annahmen, Abhängigkeiten, Rahmenbedingungen

- Externe Faktoren

3.3 Risikomanagement

- Identifizierung, Abschätzung, Verfolgung von Risiken, *Contingency*

3.4 Überwachungs- und Kontrollmechanismen

- Berichterstattungsmechanismen und -formate, Informationsflüsse, Reviews

3.5 Personalplanung

- Benötigte Fähigkeiten (Was?, Wieviel?, Wann?)

Beispiele für Annahmen

- Die Entwicklungsrechner sind schnell genug.
- Sicherheit ist kein Thema.
- Es gibt keine Bugs in dem CASE Tool, das für das Projekt empfohlen wurde.
- Der Kunde führt eine Demonstration des Starnetwork-Systems vor.

Beispiel für Abhängigkeiten

- Das Datenbankteam hängt von der von DaimlerChrysler zur Verfügung gestellten EPC Datenbank ab.

- Die automatische Codeerzeugung des CASE-Tools hängt vom jeweiligen JDK ab. Die momentane Version unterstützt nur JDK 1.1.6

Beispiele für Rahmenbedingungen

- Die Dauer des Projekts ist 3 Monate. Beschränkte Zeit zum Aufbau des Systems.
- Das Projekt besteht aus Anfängern. Es wird Zeit brauchen, den Umgang mit den Tools zu lernen.
- Nicht jeder Projektteilnehmer ist immer auf den neusten Stand, was den Status des Projekts betrifft.
- UML und ein CASE-Took *müssen* verwendet werden.
- Neuer Code soll ausschließlich in Java geschrieben werden.
- Das System *muss* JDK-1.1.6 verwenden.

Risikomanagement

- **Risiko: Ein Subsystrem bietet nicht die Funktionalität, die von einem anderen benötigt wird.**
  - *Contingency: ?*
- **Risiko: Ibutton läuft nur ab JDK 1.2**
  - *Contingency:* ?
- **Risiko: Teilnehmer in Schlüsselpositionen fallen aus.**
  - Contingency: Rollen werden jemand anderem Zugewiesen. Funktionalität des Systems wird mit dem Kunden neu ausgehandelt.
- **Risiko: Das Projekt fällt hinter den Zeitplan zurück.**
  - Contingency: Extra Projektmeetings werden anberaumt.

universitätbonn ROOTS

SPMP Teil 4: Technischer Prozess

SPMP Teil 4: Technischer Prozess

4.1 Methoden, Werkzeuge und Techniken

- Computersystem, Entwicklungsmethode, Teamstruktur, etc.
- Standards, Richtlinien, Standards, Verfahren

4.2 Softwaredokumentation

- Dokumentationsplan, einschließlich Meilensteine, Reviews und Baselines

4.3 Hilfsfunktionen für das Projekt (‚Project Support Functions‘)

- Pläne für Funktionen (Qualitätssicherung, Konfigurationsmanagement)

Software Project Management Plan (SPMP)

- Softwareprojekt
  - Alle technischen und organisatorischen Aktivitäten, die notwendig sind, um die Endergebnisse an den Kunden auszuliefern
  - Ein Softwareprojekt hat eine spezifische Dauer, beansprucht Ressourcen und produziert Arbeitsergebnisse (engl. work products).
  - Managementkategorien zur Fertigstellung eines Projekts:
    - ⇨ Funktionen, Aktivitäten, Tasks

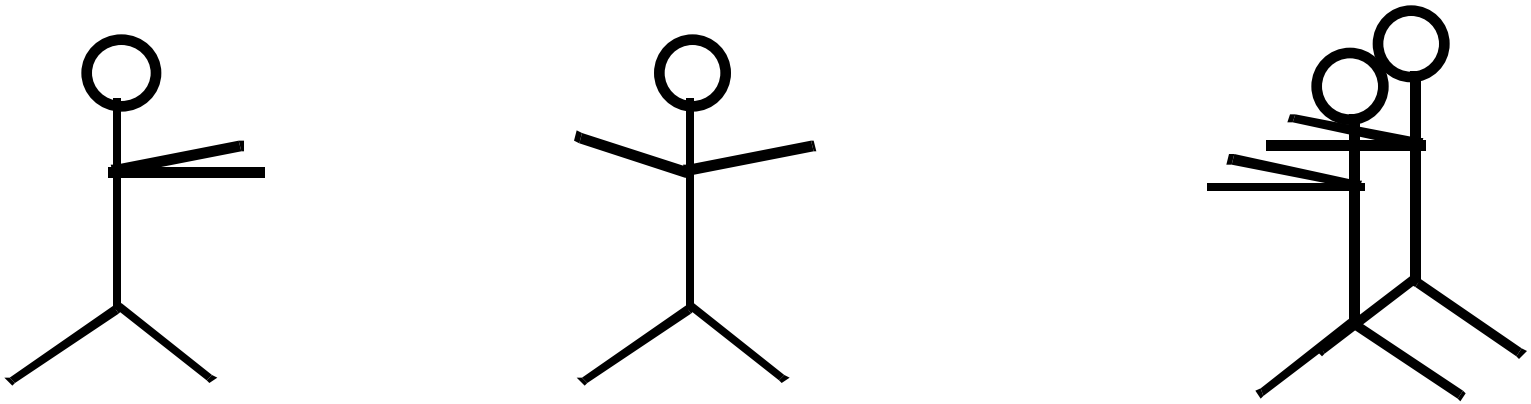
- Software Project Management Plan
  - Das Leitdokument eines Softwareprojekts
  - Spezifiziert die technischen und organisatorischen Ansätze für die Entwicklung des Softwareprodukts
  - Begleitdokument zum Anforderungsanalysedokument: Änderungen in einem der beiden erfordern evtl. Änderungen im jeweils andern.
  - SPMP kann Teil der Projektvereinbarung sein.

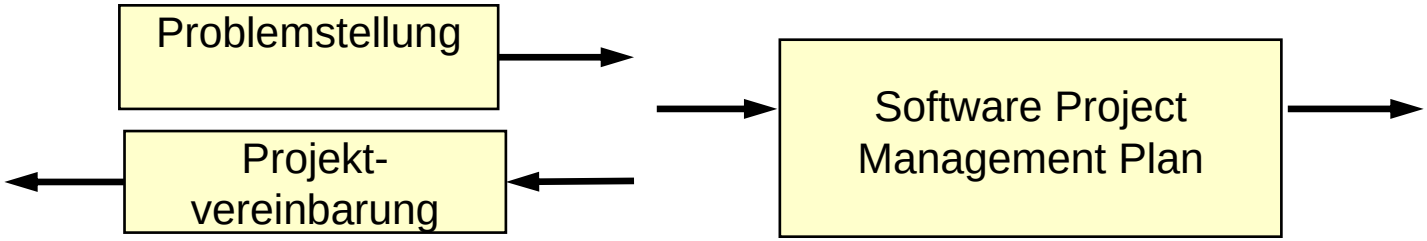
Projektvereinbarung

- Für den Kunden geschriebenes Dokument, das die folgenden Dinge festlegt:
  - Umfang, Dauer, Kosten und Endergebnisse des Projekts.
  - Exakte Angaben über Artikel, Mengen, Auslieferungstermine und Auslieferungsort
- Kann ein Vertrag, ein Statement of Work (SOW), ein Geschäftsplan oder eine Projektcharta sein.
- Kunde: Einzelne Person oder Organisation, der/die die Anforderungen spezifiziert und Endergebnisse erwartet.
- Endergebnisse (engl. Deliverables, Arbeitsergebnisse die an den Kunden ausgeliefert werden):
  - Dokumente
  - Demonstration der Funktionalität
  - Demonstration der nicht-funktionalen Anforderungen
  - Demonstrationen der Subsysteme

Projektvereinbarung vs. Problemstellung

Kunde (Sponsor) Manager Projektteam





universitätbonn ROOTS

SPMP Teil 5: Arbeitselemente, Zeitplan, Budget

Software Project Management Plan
▶ Struktur

Einstieg

1. Einführung

2. Projektorganisation

3. Organisatorischer Prozess

4. Technischer Prozess

5. Arbeitselemente, Zeitplan, Budget

Optionale Anlagen

SPMP Teil 5 ▶ Arbeitselemente, Zeitplan, Budget

5.1 Arbeitspakete (‚Work breakdown structure‘)

- Zerlegung des Projekts in Tasks; Definition der Tasks

5.2 Abhängigkeiten

- Präzedenzrelationen zwischen Funktionen, Aktivitäten und Tasks

5.3 Erforderliche Ressourcen

- Abschätzung für Ressourcen, wie z.B. Personal, Rechenzeit, spezielle Hardware, zusätzliche Software,...

5.4 Budget- und Ressourcenazuweisung

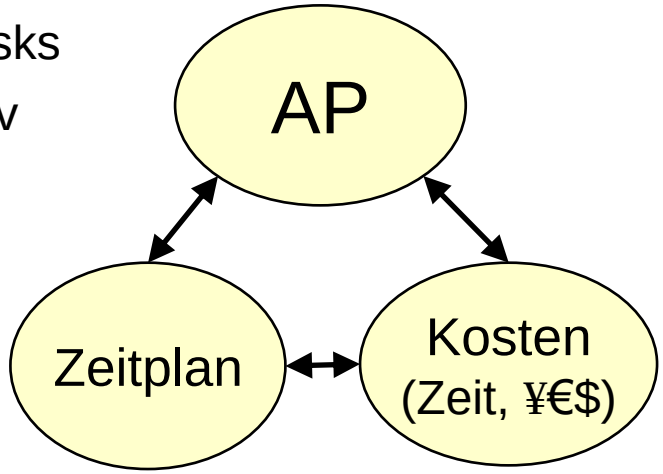
- Kosten mit Funktionen, Aktivitäten und Tasks in Verbindung setzen

5.5 Zeitplan

- Deadlines, Aufzeigen von Abhängigkeiten, notwendige Meilensteine

Erstellen von Arbeitspaketen

- Was tut man?
  - Aufbrechen des Projekts in Aktivitäten und Tasks
  - Noch ohne Abhängigkeiten zwischen den Tasks
  - Festlegung der AP ist inkrementell und iterativ
- Bedeutung
  - Aufschlüsselung der Arbeitspakete beeinflusst Zeitplan und Kosten



- Heuristik: Schwellwerte für die Kosten der Erstellung der AP in % der Gesamtkosten
  - Kleiners Projekt (7 Personen-Monate): mindestens 7% bzw. 0.5 PM
  - Mittleres Projekt(300 Personen-Monate): mindestens 1% bzw. 3 PMs
  - Großes Projekt (7000 Personen-Monate): mindestens 0.2 % bzw. 15 PMs

Quelle: [„Software Engineering Economics“, Barry W. Boehm, p. 47, Prentice Hall 1981]

Abhängigkeiten und Zeitplanung

- Abhängigkeitsgraph zeigt Abhängigkeiten unter den Tasks
  - Hierarchisch: „Ist Teil von“ / „Beinhaltet“
  - Zeitlich: „Setzt … voraus“ / „Muss vor ... passieren“
- Abschätzung der Dauer jedes Tasks
  - Abhängigkeitsgraph wird mit den Schätzungen beschriftet
- Ressourcenzuteilung
  - Wer / was ist verfügbar?
  - In welchem Umfang / wie belastbar?
  - Wie viel davon kann / will ich nutzen?

Abhängigkeiten + Taskdauer + Ressourcenzuteilung ➔ Zeitplan

Varianten von Abhängigkeitsgraphen

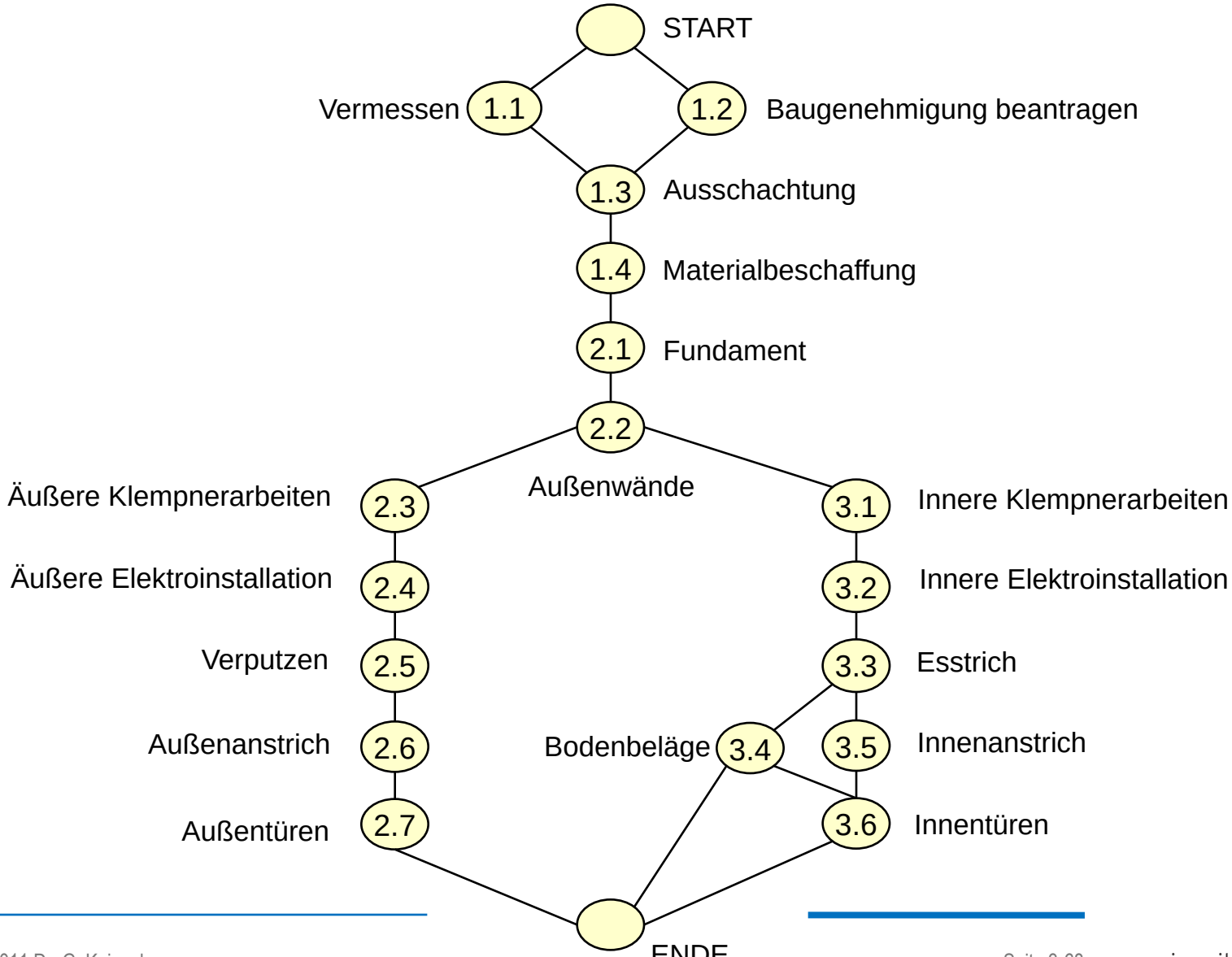
- Aktivitätengraph
  - Projektmeilensteine sind Knoten
  - Tasks sind Kanten
- Zeitplanungsdiagramm (Gant und PERT-Diagramme)
  - Tasks und Meilensteine sind Knoten
  - Kanten repräsentieren zeitliche Abhängigkeiten

→ Details siehe nächste Folien (anhand eines Beispiels)

Ein Haus bauen ▶ Aktivitäten

- Aktivität 1 : Baustelle vorbereiten
  - Task 1.1: Vermessen
  - Task 1.2: Baugenehmigung
  - Task 1.3: Ausschachten
  - Task 1.4: Materialbeschaffung
- Aktivität 2: Rohbau
  - Task 2.1: Fundament
  - Task 2.2: Außenwände
  - Task 2.3: Ext. Klempnerarbeiten
  - Task 2.4: Ext. Elektroinstallation
  - Task 2.5: Verputzen
  - Task 2.6: Außenanstrich
  - Task 2.7: Aussentüren
- Aktivität 3 : Innenausbau
  - Task 3.1: Int. Klempnerarbeiten
  - Task 3.2: Int. Elektroinstallation
  - Task 3.3: Esstrich
  - Task 3.4: Innenanstrich
  - Task 3.5: Bodenbeläge
  - Task 3.6: Innentüren

Ein Haus bauen ▶ Aktivitätsgraph



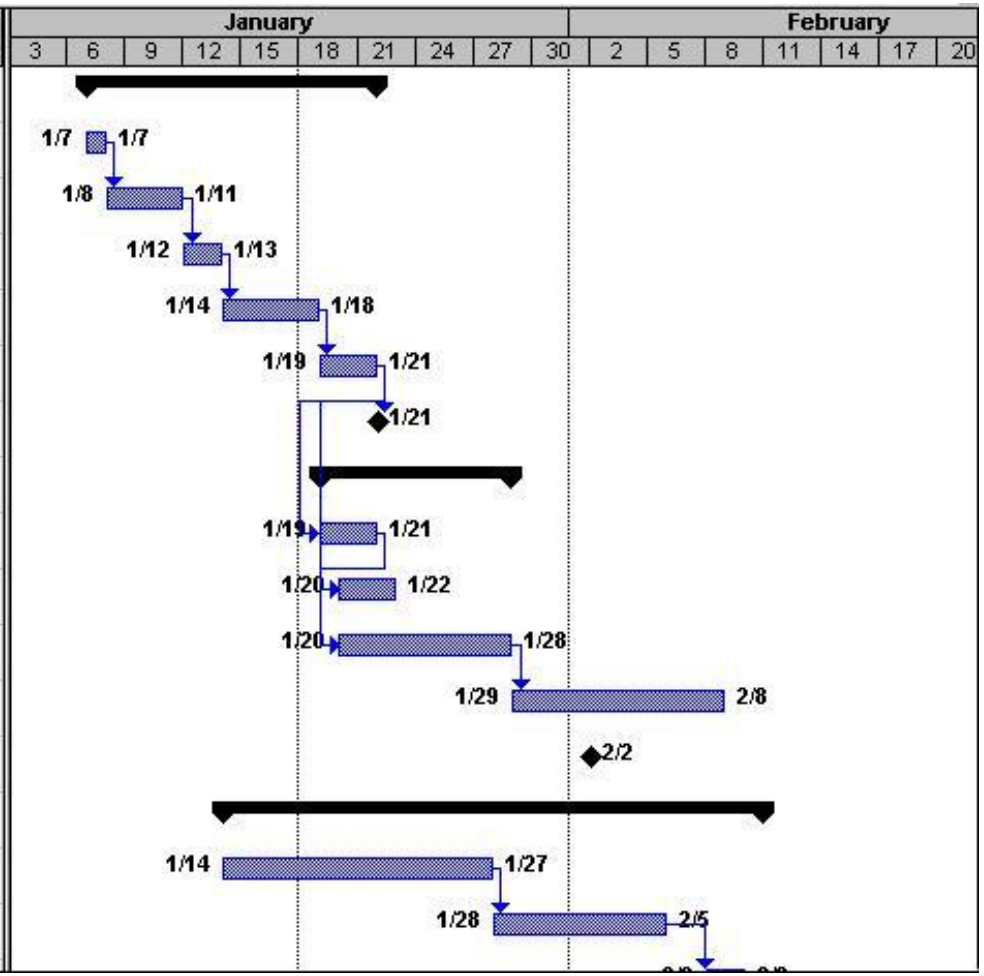
Zeitplanungsdiagramme ▶ Hilfe durch Projektverwaltungswerkzeuge

- Gantt Diagramm (Task Zeitleiste)
  - Zeigt Projektaktivitäte und -tasks parallel
  - Zeigt Dauer und Abhängigkeiten
  - Ermöglichen dem Projektleiter nachzuvollziehen, welche Tasks gleichzeitig bearbeitet werden können

- PERT Diagramm (Zeitplan)
  - Grafische Repräsentation von Abhängigkeiten zwischen Tasks und Milestones
  - PERT = Program Evaluation and Review Technique
    - ⇨ Ein PERT-Diagramm geht von einer Normalverteilung der Taskdauer aus.
    - ⇨ Nützlich für Analyse kritischer Pfade (‚Critical Path Analysis‘)
  - CPM = Critical Path Method
    - ⇨ Kritischer Pfad ist Folge von Aktivitäten deren Verzögerung den gesamten Ablauf verzögern würde

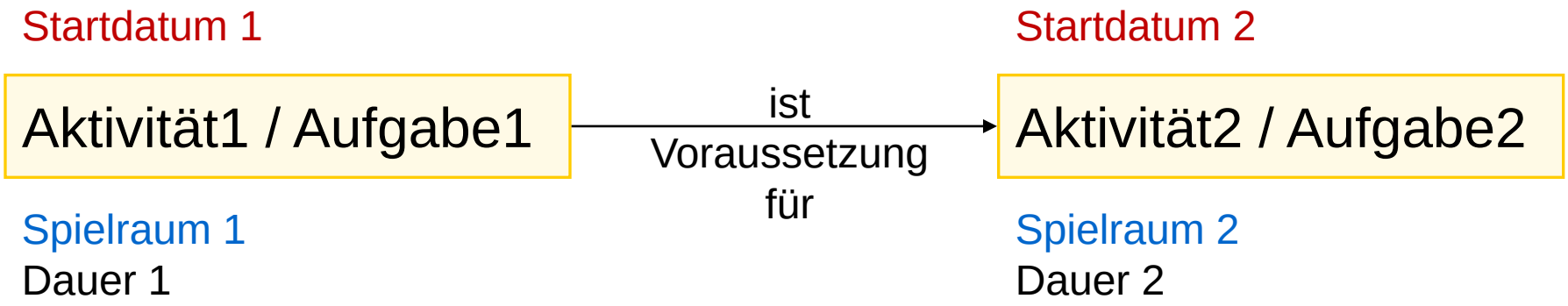
Gantt-Chart ▶ Beispiel

WBS	Task Name
1	**Define specifications**
1.1	Identify customers
1.2	Interview 10 customers
1.3	Interpret requirements
1.4	Benchmark products
1.5	Define target PDS
1.6	Target PDS Released
2	**Generate concepts**
2.1	Review comp products
2.2	Search patents
2.3	Brainstorm concepts
3	Select top 2 concepts
4	MQ Presented
5	**Profile motor power**
5.1	Design test stand
5.2	Build test stand



(Gantt-Charts sind benannt nach dem amerikanischen Ingenieur H. L. *Gantt* (1861-1919)

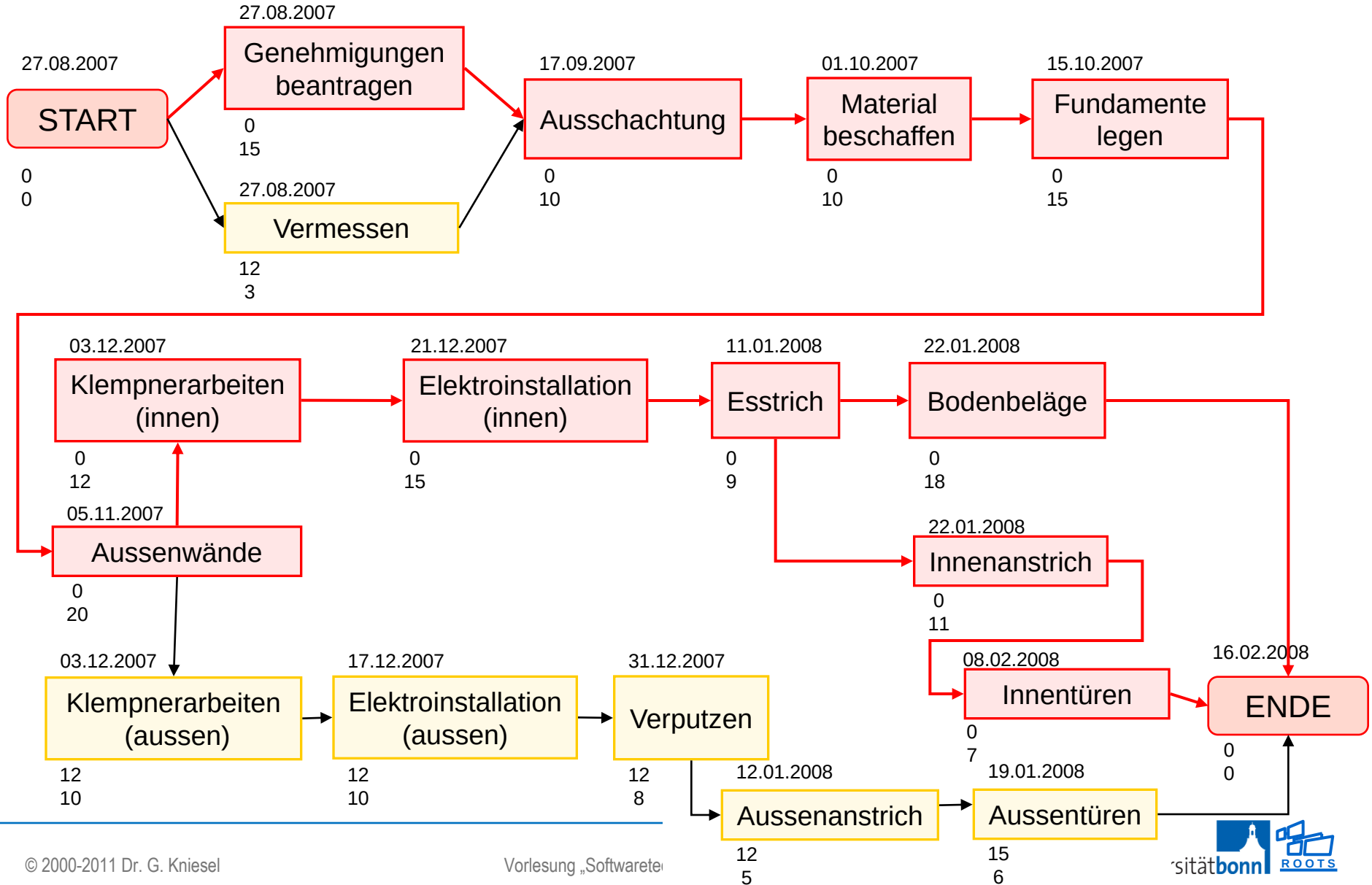
PERT-Charts ▶ „Spielraum“ und „Kritischer Pfad“



- Zeitlicher Spielraum (‚slack time‘)
  - geplante Startzeit minus früheste mögliche Startzeit
  - Spielraum 1 = Startdatum 2 – (Startdatum 1 + Dauer 1)
- Kritischer Pfad (‚critical path‘)
  - Der Pfad in einem Projektplan, für den der Spielraum an jedem Knoten (Task/Aktivität) null ist.
- Auf dem kritischen Pfad gibt es keinen Spielraum für Tasks/Aktivitäten.
  - Jede Verzögerung einer Aktivität / Task auf dem kritischen Pfad verzögert das gesamte Projekt!

Ein Haus bauen ▶ PERT Diagramm

(Aktivitäten auf kritischen Pfaden sind rot markiert)



Weitere Visualisierungshilfen

- Graphen (Zeitplan)
- Bäume (Arbeitspakete)
- Tabellen (Ressourcen)

Wie werde ich ein guter Projektplaner?

- Stell einen Projektplan auf!
  - Fange basierend auf Erfahrungen mit vergangenen Projekten an
- Behalte Aktivitäten und ihre Dauer im Auge
  - Bestimme die Differenz zwischen geplantem und tatsächlichen Durchsatz
  - Passe den Plan an
  - Überdenke die Ursachen der Verzögerungen und beuge vor
- Denk daran, eine Post-Mortem-Analyse zu machen
  - Bitte Entwickler um Feedback
  - „Was haben wir aus dem Projekt gelernt?“
  - Halte schriftlich fest, was verbessert werden könnte.
- „Post mortem“ ist zu spät! → Laufend Feedback einholen!

Heuristiken zum Projektmanagement

- Halte dir die Möglichkeit offen, einen Projektplan zu ändern oder auch komplett zu verwerfen.
  - Die Entwicklung komplexer Systeme ist eine nicht-lineare Angelegenheit.
- Wenn die Ziele unklar und komplex sind, benutze teambasiertes Projektmanagement. In diesem Fall...
  - Vermeide GANTT und PERT Diagramme für Projekte mit sich ändernden Anforderungen.
  - Blicke nicht zu weit in die Zukunft.
- Sei nicht überrascht, wenn aktuelle Projektverwaltungstools nicht funktionieren:
  - Sie wurden für Projekte mit klaren Zielen und festen Organisationsstrukturen entworfen.
- Vermeide Mikromanagement von Details.

Projektmanagement: Zusammenfassung

- Projekt
  - Integrale Prozesse, Aktivitäten, Tasks, Action Items
- Übereinkünfte zwischen Kunden, Managern und Teams
  - Problemstellung -- SPMP – Projektvereinbarung
- Projektplanung
  - Aufschlüsselung der Arbeitspakete (Work breakdown structure)
  - Abhängigkeiten und Strukturen identifizieren: Tasks, Aktivitäten, Funktionen
- Werkzeuge und Techniken
  - GANTT, Abhängigkeitsgraph, Zeitplan, Critical Path Analyse
  - Vorsicht mit Werkzeugen in Projekten mit viel Änderung
- Gibt es Alternativen zu PERT, Gant & Co?
  - → „Issue-based project management“ !?!