

Vorlesung "Softwaretechnik"

Projektmanagement 2

Lutz Prechelt

Freie Universität Berlin, Institut für Informatik

- Zeit- und Ressourcenplanung
 - MS-Project
 - Critical Path Method (CPM)
- Risikomanagement
 - Prinzipien
 - Risikolisten
 - DOs and DON'Ts

- Planung mit Aufgabenzerlegung, Abhängigkeiten und kritischen Pfaden verstehen
- Ziel, Wichtigkeit und Vorgehen von Risikomanagement verstehen

Wo sind wir?: Taxonomie "Die Welt der Softwaretechnik"

Welt der Problemstellungen:

- Produkt (Komplexitätsprob.)
 - Anforderungen (Problemraum)
 - Entwurf (Lösungsraum)
- Prozess (psycho-soziale P.)
 - Kognitive Beschränkungen
 - **Mängel der Urteilskraft**
 - **Kommunikation, Koordination**
 - Gruppendynamik
 - Verborgene Ziele
 - **Fehler**

Welt der Lösungsansätze:

- Technische Ansätze ("hart")
 - Abstraktion
 - Wiederverwendung
 - Automatisierung
- Methodische Ansätze ("weich")
 - Anforderungsermittlung
 - Entwurf
 - Qualitätssicherung
 - **Projektmanagement**

- Einsicht: Ein Projekt läuft nicht von allein gut ab
 - sondern benötigt Planung, Leitung und Risikomanagement
- Prinzipien:
 - **Zielsetzung**: Mache Ziele u. Prioritäten bekannt und akzeptiert
 - **Stabile Anforderungen**: Verhindere übermäßige Anf.änd.
 - **Iteration**: Erzeuge in kurzen Abständen wohldefinierte Ergebnisse (Meilensteine)
 - **Planung und Koordination**: Vergebe klare Aufgaben u. Zeitvorgaben, überwache, greife bei Abweichungen früh ein
 - **Kommunikation**: Mache allen die von ihnen benötigte Information rechtzeitig bekannt
 - **Konflikt**: Erkenne Konflikte rechtzeitig und löse sie sinnvoll auf
 - **Risikomanagement**: Entdecke regelmäßig vorbeugend wichtige Gefahren, vermeide Eintreten, bereite Reparaturmaßnahmen vor
 - **Normales Vorgehen**: Vermeide radikales Vorgehen

- Zeitplanung
 - Schätzung des Gesamtaufwands ("cost estimation")

• Aufstellen eines Zeit- und Arbeitsplans



- Risikomanagement
 - Risikoermittlung und -einschätzung
 - Riskovorbeugung und -behandlung
- Personalmanagement
 - Gruppen und Teams
 - Psychologische Effekte
- Integrierende Aufgaben
 - Projektplan
 - Projektleitung und -überwachung

Planung: Problem

- Es genügt für das Projektmanagement nicht, den Aufwand für das Gesamtprojekt zu kennen
- Man muss auch herausfinden, wie es zu organisieren ist: Planung
- Einzelfragen:
 - Welche Teilaufgaben sind zu lösen?
 - Wie lange dauern die?
 - Welche Abhängigkeiten bestehen dazwischen (A muss vor B)?
 - Wie viele Leute mit welchen Qualifikationen brauchen wir? Wann?
 - Welche anderen Ressourcen (z.B. SW, Dienste, Rechner, Hilfspersonal) brauchen wir? Wann?
 - Sobald das Personal feststeht: Wer ist qualifiziert für welche Aufgabe, die jeweils gerade dran ist?

Metaphorisches Beispiel: Wir bauen ein Haus

Ein Holzhaus ohne Keller.

Work breakdown Structure:

- Haus bauen besteht aus
 1. Fundament legen
 2. Aussenhaut bauen
 3. Innenausbau
- 1. Fundament legen besteht aus
 - Grundstück vermessen
 - Baugenehmig. einholen
 - Planieren/Ausschachten
 - Fundament gießen

Kann man das mit 20 Leuten
10 mal so schnell bauen
wie mit 2 Leuten? Warum?

2. Außenhaut bauen
besteht aus

- Tragwerk bauen
- Fußböden einziehen
- Außenwände bauen
- Dach decken
- Fenster einbauen
- Haustür einbauen



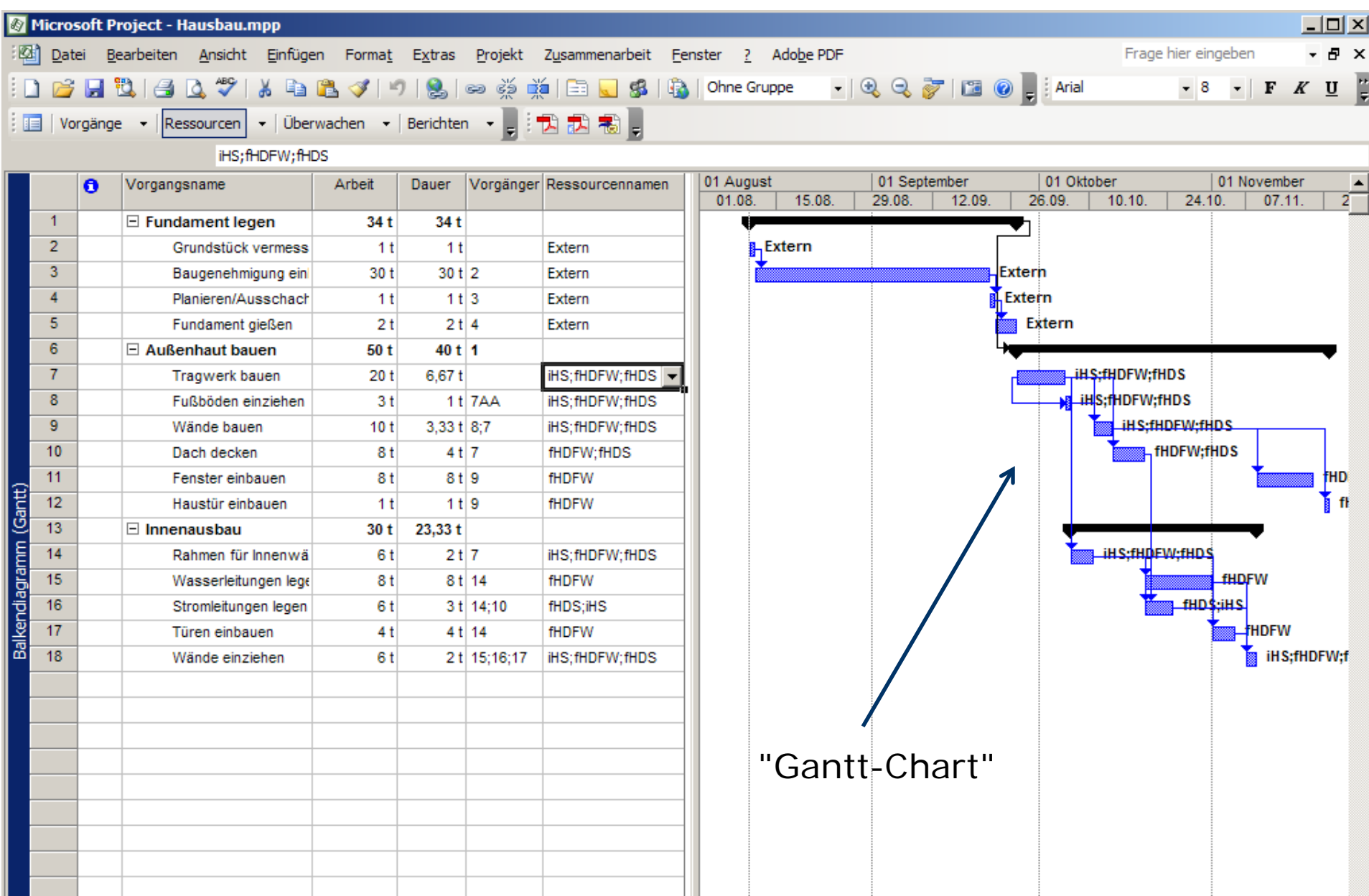
3. Innenausbau besteht aus

- Rahmen für Innenwände bauen
- Wasserleitungen legen
- Stromleitungen legen
- Türen einbauen
- Wände einziehen
- u.s.w. (+ feinere Zerlegung)

Naive Laiensicht!

- Im Prinzip wird die Klärung dieser Fragen stark durch Projektmanagement-SW unterstützt:
 - Liste alle Teilaufgaben auf und schätze deren Umfang
 - Personentage, also Zeitdauer bei 1 Person mit 100% Arbeitseinsatz
 - Liste alle benötigten/vorhandenen Ressourcen auf
 - Ermittle die Abhängigkeiten zwischen Aufgaben und trage sie ein
 - Weise die Ressourcen den Aufgaben zu
 - → Die Software ermittelt den Plan und Zeitverlauf
 - → Bei Änderungen an Aufgaben, Umfängen, Ressourcen wird der Plan in Sekundenschnelle angepasst
- Es gibt diverse Softwaresysteme für solche Unterstützung
 - Am populärsten ist Microsoft Project
 - Skaliert bis hin zu Dutzenden kooperierender Teilprojekt-Manager
 - Eine freie Alternative ist z.B. ProjectLibre (projectlibre.com)
 - Oft ausreichend

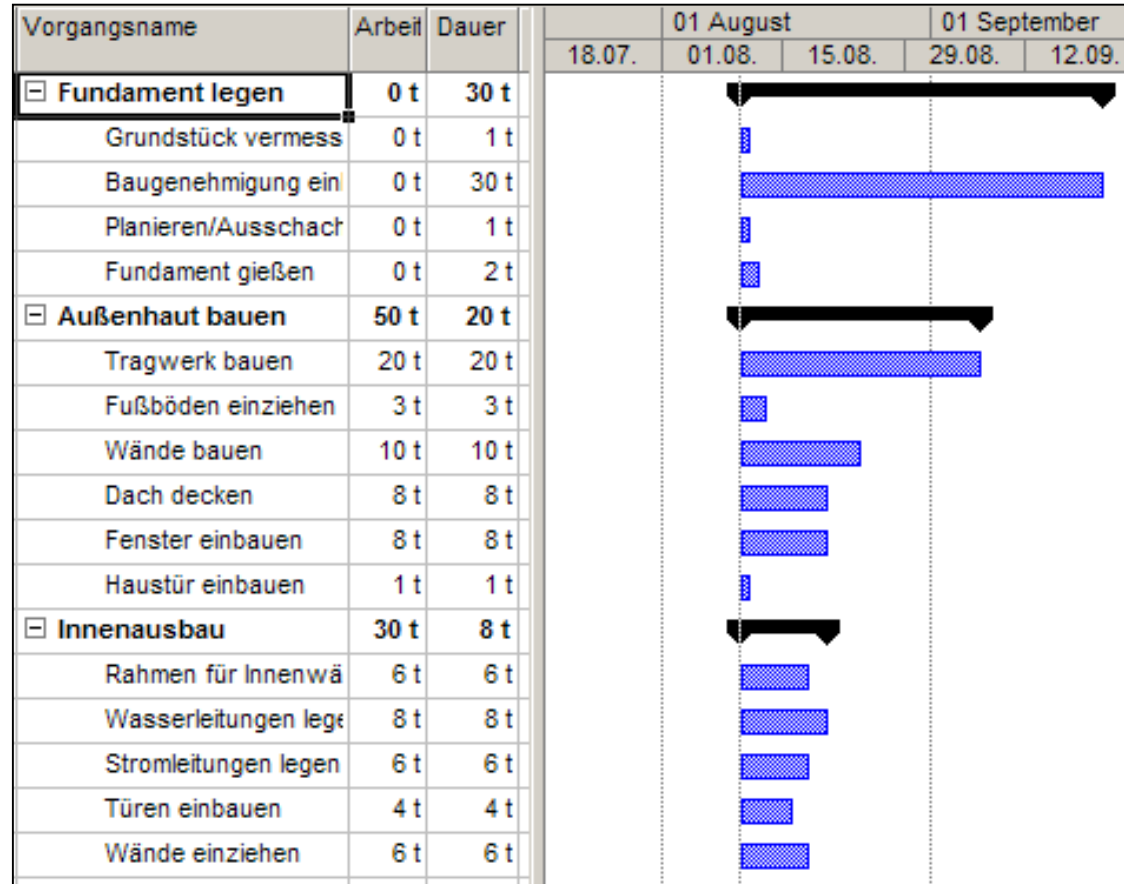
Microsoft Project: Fertiger Plan



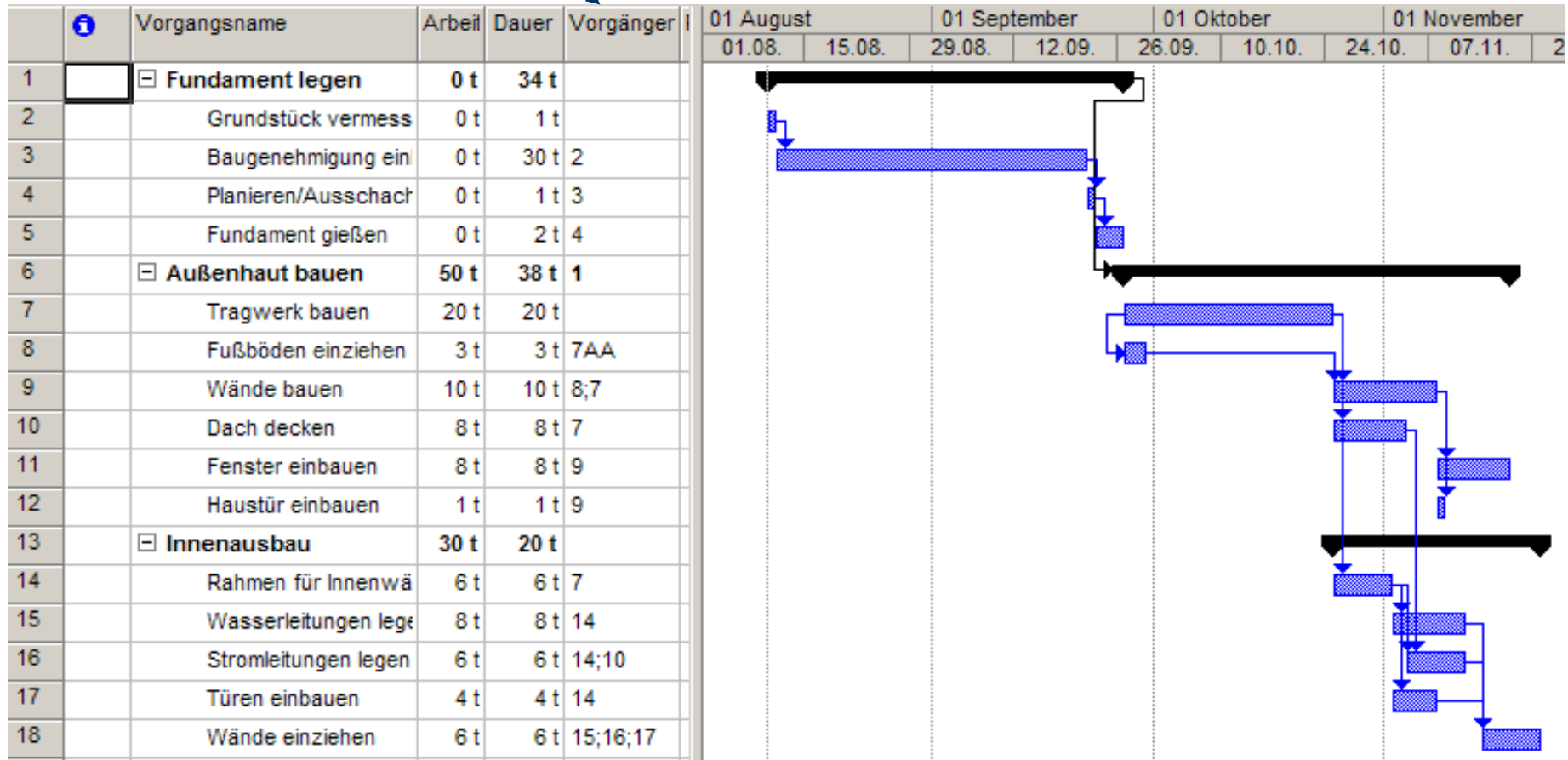
Microsoft Project: Aufgaben + Aufwand erfassen

- Teilaufgaben ("Vorgänge") mit Aufwand ("feste Arbeit" oder "feste Dauer") eingeben, gruppieren

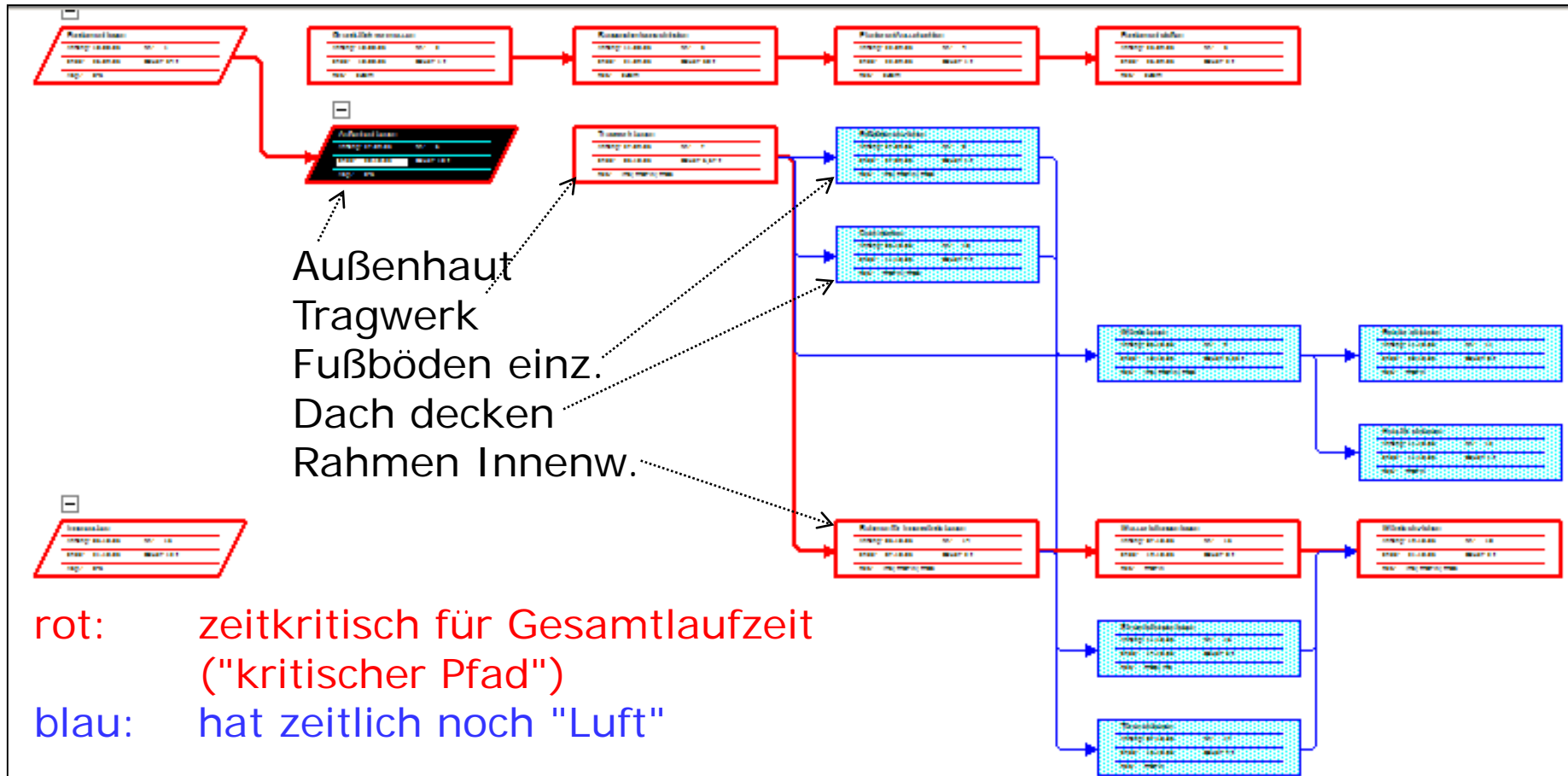
Vorgangsname	Arbeit	Dauer
Grundstück vermessen	0 t	1 t
Baugenehmigung einhole	0 t	30 t
Planieren/Ausschachten	0 t	1 t
Fundament gießen	0 t	2 t
Tragwerk bauen	20 t	20 t
Fußböden einziehen	3 t	3 t
Wände bauen	10 t	10 t
Dach decken	8 t	8 t
Fenster einbauen	8 t	8 t
Haustür einbauen	1 t	1 t
Rahmen für Innenwände	6 t	6 t
Wasserleitungen legen	8 t	8 t
Stromleitungen legen	6 t	6 t
Türen einbauen	4 t	4 t
Wände einziehen	6 t	6 t



- Welche Vorgänger müssen vor einer Aufgabe erledigt sein?
 - kürzester Zeitplan wird automatisch berechnet



- Die Darstellung der Abhängigkeiten ohne Zeitskala nennt man auch Netzplan (alternativ zum Gantt-Chart)



- Wir wollen unser Haus zusammen mit zwei Freunden bauen:
 - ich i, Freunde f
 - jeder voll verfügbar (100%)
- Jeder kann unterschiedliche Arbeiten erledigen (oder eben nicht):
 - H Holzarbeiten
 - S Strom, Elektrik
 - W Wasser
 - D Dach decken
 - F Fenster/Türen einbauen
- Diese Fähigkeiten sind hier in den Ressourcennamen angegeben (zwecks Klarheit)
- Die Vorbereitungsarbeiten (Vermessen, Fundament etc.) überlassen wir externen Dienstleistern
 - Ressource "Extern", hat beliebig hohe Kapazität


Ressourcename	Max. Einheiten
iHS	100%
fHDFW	100%
fHDS	100%
Extern	1.000%

100% = 1 Personentag pro Tag

MS Project: Zuordnung der Ressourcen

- Wir ordnen zunächst jedem alle Arbeiten zu, für die er/sie geeignet ist



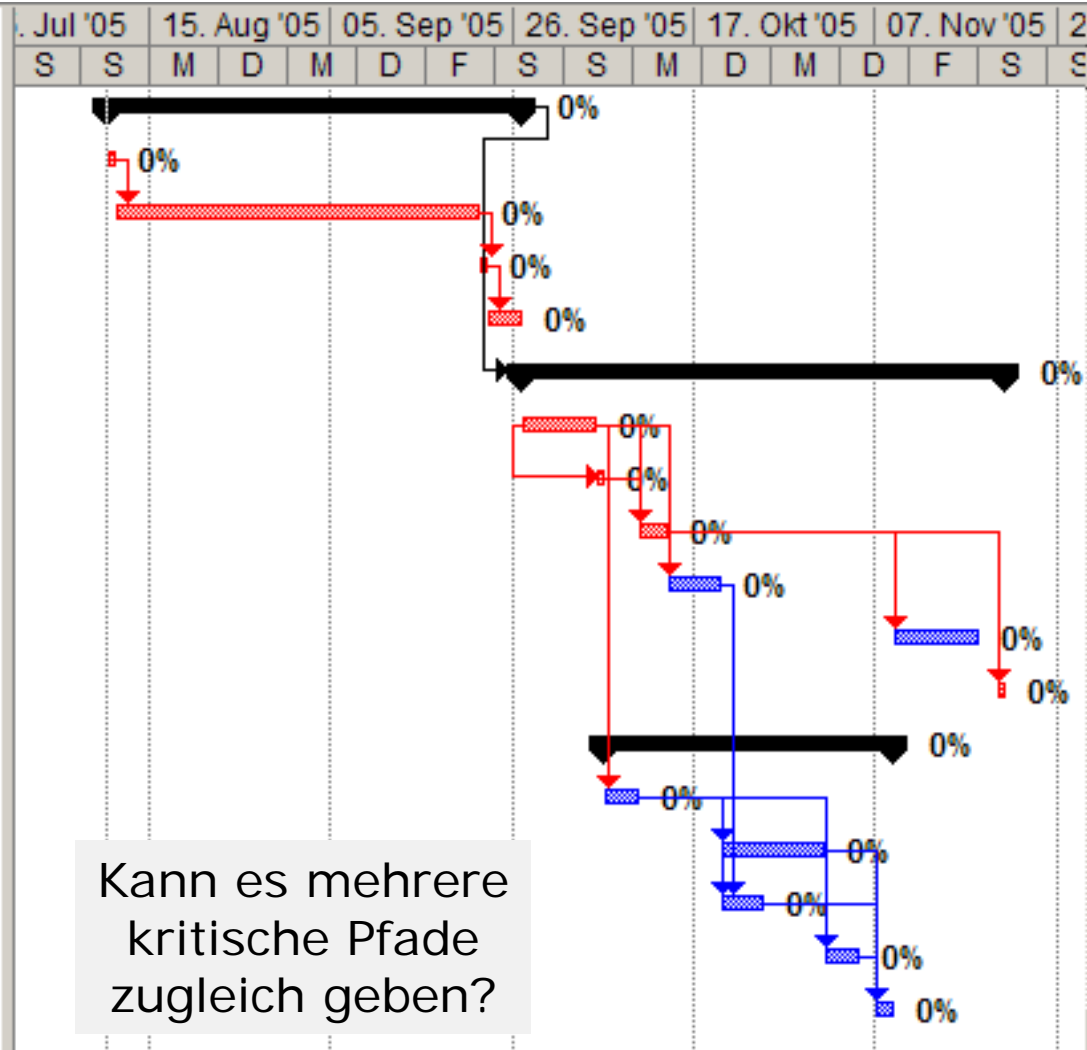
		Vorgangsname	Arbeit	Dauer	Vorgänger	Ressourcennamen
1	<input type="checkbox"/>	Fundament legen	34 t	34 t		
2		Grundstück vermessen	1 t	1 t		Extern
3		Baugenehmigung einholen	30 t	30 t	2	Extern
4		Planieren/Ausschachten	1 t	1 t	3	Extern
5		Fundament gießen	2 t	2 t	4	Extern
6	<input type="checkbox"/>	Außenhaut bauen	50 t	18 t	1	
7		Tragwerk bauen	20 t	6,67 t		iHS; fHDFW; fHDS
8		Fußböden einziehen	3 t	1 t	7AA	iHS; fHDFW; fHDS
9		Wände bauen	10 t	3,33 t	8;7	iHS; fHDFW; fHDS
10		Dach decken	8 t	4 t	7	fHDFW; fHDS
11		Fenster einbauen	8 t	8 t	9	fHDFW
12		Haustür einbauen	1 t	1 t	9	fHDFW
13	<input type="checkbox"/>	Innenausbau	30 t	12 t		
14		Rahmen für Innenwände	6 t	2 t	7	iHS; fHDFW; fHDS
15		Wasserleitungen legen	8 t	8 t	14	fHDFW
16		Stromleitungen legen	6 t	3 t	14;10	fHDS; iHS
17		Türen einbauen	4 t	4 t	14	fHDFW
18		Wände einziehen	6 t	2 t	15;16;17	iHS; fHDFW; fHDS

Jetzt tritt die eigentliche Automatisierung in Aktion:

- Ressourcenabgleich:
 - Die Beteiligung jeder Ressource an ihren Aufgaben wird so reduziert (in %), dass die Ressource nicht durch gleichzeitige Aufgaben überlastet ist
- Kritischer Pfad:
 - Es wird berechnet, bei welchen Aufgaben jede Verzögerung zu einer Verzögerung des Gesamtprojekts führt
 - Die Abfolge dieser Aufgaben bildet den so genannten "kritischen Pfad" durch das Projekt
 - Hier die Arbeiten besonders sorgfältig überwachen+unterstützen!
 - Alle Aufgaben außerhalb kritischer Pfade haben Spielraum (slack time), d.h. Zeitpuffer
 - Verzögerungen dieser Aufgaben, die geringer als der Spielraum sind, verzögern das Projekt nicht
 - Aber das Nutzen eines Spielraums verkleinert andere Spielräume

MS Project: Kritischer Pfad

Vorgangsname	Arbeit	Dauer	Vorgänger
Fundament legen	34 t	34 t	
Grundstück vermessen	1 t	1 t	
Baugenehmigung eintr.	30 t	30 t	2
Planieren/Ausschachte	1 t	1 t	3
Fundament gießen	2 t	2 t	4
Außenhaut bauen	50 t	40 t	1
Tragwerk bauen	20 t	6,67 t	
Fußböden einziehen	3 t	1 t	7AA
Wände bauen	10 t	3,33 t	8;7
Dach decken	8 t	4 t	7
Fenster einbauen	8 t	8 t	9
Haustür einbauen	1 t	1 t	9
Innenausbau	30 t	23,33 t	
Rahmen für Innenwände	6 t	2 t	7
Wasserleitungen legen	8 t	8 t	14
Stromleitungen legen	6 t	3 t	14;10
Türen einbauen	4 t	4 t	14
Wände einziehen	6 t	2 t	15;16;17



MS Project: kompliziertere Fälle

Im Prinzip kann MS Project (und ähnliche SW) auch kompliziertere Situationen verwalten, z.B.

- Ungleichmäßige Verfügbarkeit einer Ressource
 - z.B. Urlaub
 - z.B. dynamische Aufteilung der Kapazität auf mehrere Projekte
- Verwaltung von Teilprojekten durch andere Personen
 - Kommunikation über Projektserver
- Zeitliche Randbedingungen für einzelne Aufgaben
- Zuordnungsprioritäten
- etc.

- Aber die Benutzung wird dann auch entsprechend schwierig, aufwändig und fehleranfällig
 - und die Ergebnisse evtl. undurchsichtig

- Planung mit Wahrscheinlichkeitsverteilungen
 - minimaler, maximaler, erwarteter Aufwand
 - probabilistische Planberechnung (PERT-Methode)
- Kalenderlogik
 - manche R. arbeiten am Wochenende, andere nicht etc.
- Planverfolgung und -überwachung
- Kostenplanung und -verfolgung
 - mit getrennten Kostensätzen für Überstunden etc.
- Ziemlich leistungsfähiges (und kompliziertes) Programm!

In der Praxis läuft es allerdings meist erheblich anders:

- Eine grobe Planung kann man ohne Werkzeug überblicken
- Für eine feine Planung funktioniert das Werkzeug nicht gut genug:
 - Oft ist der Umfang mancher Aufgaben so ungewiss, dass die Planung kaum Information liefert
 - Der Umfang einer Aufgabe hängt davon ab, wer sie erledigt
 - Das kann man aber nicht mit unterschiedlicher Kapazität der Personen modellieren, denn die hängt wieder von der Aufgabe ab
 - Außerdem gibt es so oft Störungen im Ablauf, dass das ständige Umplanen viel zu aufwändig wird
- Bei kleinen und mittleren Projekten wird deshalb eine SW wie MS Project selten eingesetzt, weil sich der Aufwand dafür gar nicht lohnt
 - Bei Großprojekten ist sie aber sehr hilfreich
 - Typisch bei kleineren Projekten: Excel-Tabellen



Wie findet man die Aufgabenzerlegung (work breakdown structure, WBS)?

- Top-down-Ansatz:
 - Meistens sinnvoll
 - Identifiziere die größten nötigen Schritte und breche Sie schrittweise in immer feinere Schritte herunter
 - Die obere Ebene kann man nach verschiedenen Kriterien wählen:
 - Aktivitäten des Prozessmodells
 - Komponenten des SW-Systems
- Brainstorming- und Bottom-up-Ansatz:
 - u.U. sinnvoll
 - Trage mit einer Gruppe von Personen wild durcheinander *irgendwelche* Aufgaben zusammen und ordne sie anschließend in Gruppen
 - Das muss man vor einer Top-Down-Zerlegung machen
 - sonst ist man betriebsblind

- Bei lang laufenden oder iterativen Projekten sollte man nicht gleich alles bis ins Detail zerlegen, sondern:
- Zerlege Phase 1 (oder Iteration 1) genau
- Zerlege Phase 2 (oder Iteration 2) mäßig detailliert
- Zerlege Phase/Iteration 3, 4, ... nur sehr grob
- So verhindert man übertriebenen Neuplanungsaufwand
 - denn die späteren Phasen sehen doch meist anders aus als ursprünglich gedacht



Agile Projekte planen
nur in einer Iteration:
Viel einfacher!

So viel zu globaler Zeit- und Aufgabenplanung

- Zeitplanung
 - Schätzung des Gesamtaufwands ("cost estimation")
 - Aufstellen eines Zeit- und Arbeitsplans
- Risikomanagement
 - Risikoermittlg./-einschätzung
 - Riskovorbeugg./-behandlung
- Personalmanagement
 - Gruppen und Teams
 - Psychologische Effekte
- Integrierende Aufgaben
 - Projektplan
 - Projektleitung und -überwachung



Definition "Risiko"

- Risiko ist die Möglichkeit, dass ein unerwünschtes Ereignis eintritt
- Mögliche unerwünschte Ereignisse vorab zu erkennen, ist eine wichtige Managementaufgabe
 - nur relevante (plausible) Ereignisse für dieses Projekt
 - möglichst konkrete Beschreibung
- Risikohöhe = Eintrittswahrscheinlichkeit mal Schadenshöhe
 - engl.: risk exposure = probability of loss * size of loss
 - lässt sich meist nur sehr schwer quantitativ bestimmen
 - qualitative Bestimmung (d.h. sehr grob und nur relativ) reicht aber für Risikomanagement (RM)

Managementstile in Bezug auf Risiken:

- **Ignorierend/optimistisch:**

Kümmere Dich nur um das, was Du erreichen willst; nicht um das, was dabei in die Quere kommen könnte oder schon gekommen ist.



- führt zu sehr langen Terminüberschreitungen

- **Reparierend:**

Kümmere Dich um Probleme, die auftreten. Alles, was glatt geht, bearbeiten die Mitarbeiter selbständig.



- häufigster Stil. Kein Risikomanagement

- **Vorbeugend** (Risikomanagement):

Kümmere Dich um Probleme, bevor sie auftreten



- *"If you don't attack the risks, the risks will attack you."* (Tim Lister)

Was ist Risikomanagement (RM)?



- "Create a focus on the critical success factors"
- Schritte:
 1. Risiken identifizieren
 2. Risiken bewerten:
 - Eintrittswahrscheinlichkeit des Problems
 - Dann erwarteter Schaden (Geld, Zeit, Aufwand)
 3. Priorisieren: Welche Risiken werden behandelt?
 - Dann erwarteter Schaden (Geld, Zeit, Aufwand)
 4. Risiken managen:
 - Maßnahmen zum Senken der Chancen wo möglich
 - Jetzt einleiten, dann aufrecht erhalten
 - Maßnahmen zum Senken des Schadens sonst
 - Jetzt vorbereiten
 - bei Eintreten nachverfolgen bis Ziel erreicht ist
 5. Regelmäßig wiederholen/überprüfen

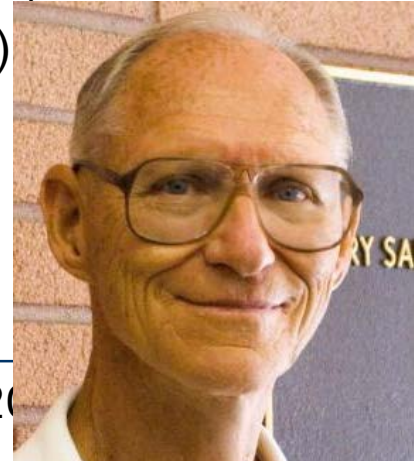


1. Projekt ganzheitlich betrachten
2. Vorausblickende Arbeitsweise
3. Offener Kommunikationsstil
4. Management in einer Hand
5. Dauerhafte Angelegenheit
6. Gemeinsame Sicht auf die Ziele
7. Teamwork

<http://www.sei.cmu.edu/risk/>

Bis auf (2.) ist das bei agilen Projekten alles eine Selbstverständlichkeit, deshalb ist dort RM oft eher implizit

- Barry Boehm: "*Software Risk Management: Principles and Practices*", IEEE Software 8(1):32-41, Januar 1991.
 - Personnel shortfalls (wenig oder schwaches Personal)
 - Unrealistic schedules and budgets (Todesmarsch-Projekt)
 - Developing the wrong functions and properties (falsche Anforderungen)
 - Developing the wrong user interface (schlechte Benutzbarkeit)
 - Gold-plating (technischer Übereifer)
 - Continuing stream of requirements changes (häufige Änderungswünsche)
 - Shortfalls in externally furnished components (Mängel in externen Teilen)
 - Shortfalls in externally performed tasks (Mängel in ext. Dienstleistungen)
 - Real-time performance shortfalls (zu lange Antwortzeiten)
 - Straining computer-science capabilities (radikales Vorgehen)
- (nicht für jedes Projekt gleichermaßen gültig!)
- 30 Jahre alt, aber nur zwei Punkte sind (etwas) überholt. Welche?

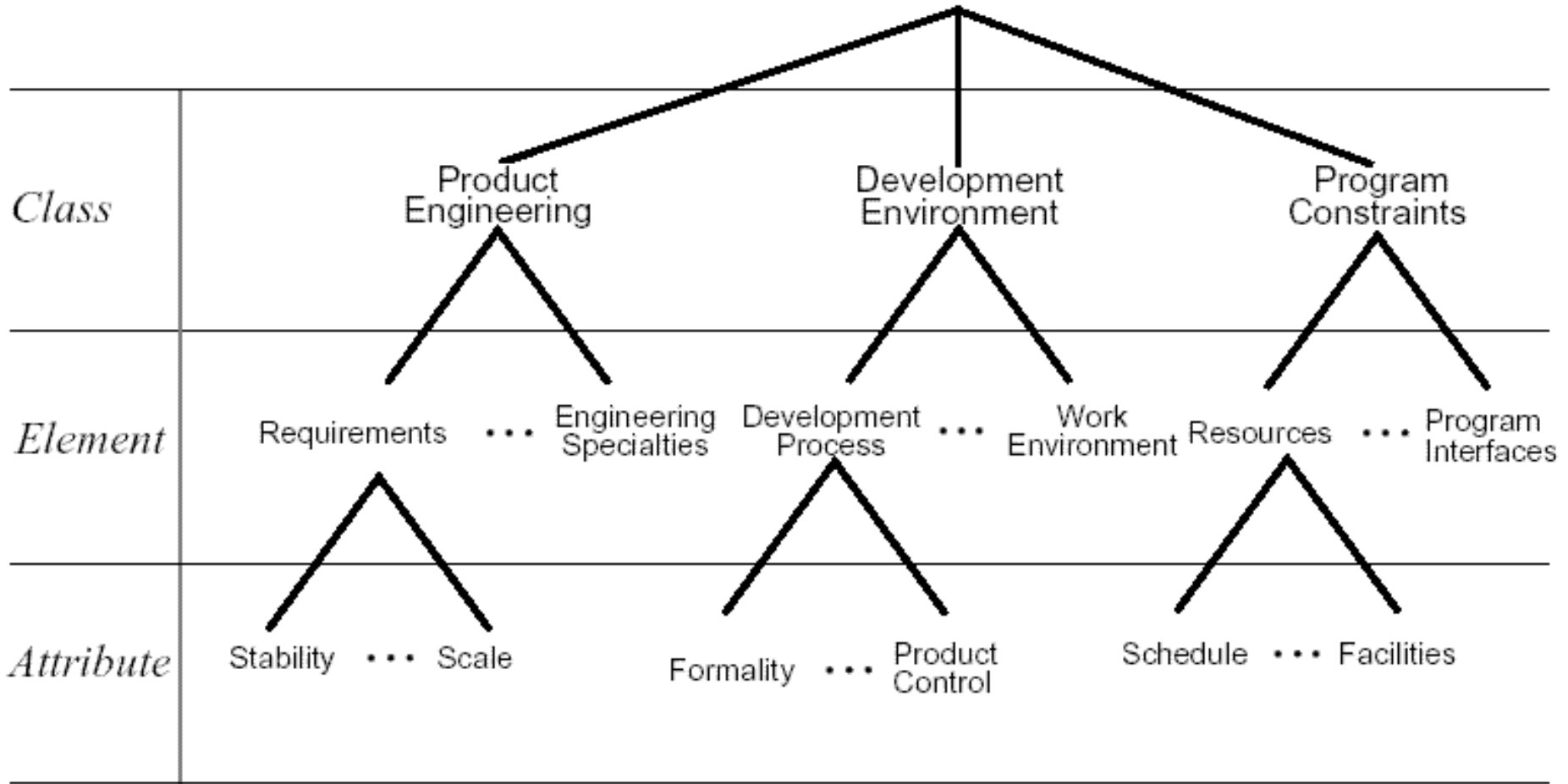


Hilfe zur Risikoidentifikation

- Marvin Carr, Suresh Konda, Ira Monarch, Carol Ulrich, Clay Walker:
"Taxonomy-Based Risk Identification"
 - <http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf>
- Enthält eine Risikotaxonomie
 - genauer: eine Taxonomie von Risikoarten oder –bereichen
- und einen Fragebogen, um Risiken zu identifizieren
- Wurde mittels Feldtests entwickelt
- Risiken sind bekannt, unbekannt oder unerkennbar ("unknowable")
- Die meisten Risiken sind im Prinzip bekannt, werden aber schlecht kommuniziert

sozusagen
"Beyond Boehm"

Software Development Risk



Generische Risikotaxonomie (2)

A. Product Engineering

1. Requirements
 - a. Stability
 - b. Completeness
 - c. Clarity
 - d. Validity
 - e. Feasibility
 - f. Precedent
 - g. Scale
2. Design
 - a. Functionality
 - b. Difficulty
 - c. Interfaces
 - d. Performance
 - e. Testability
 - f. Hardware Constraints
 - g. Non-Developmental Software
3. Code and Unit Test
 - a. Feasibility
 - b. Testing
 - c. Coding/Implementation
4. Integration and Test
 - a. Environment
 - b. Product
 - c. System
5. Engineering Specialties
 - a. Maintainability
 - b. Reliability
 - c. Safety
 - d. Security
 - e. Human Factors

B. Development Environment

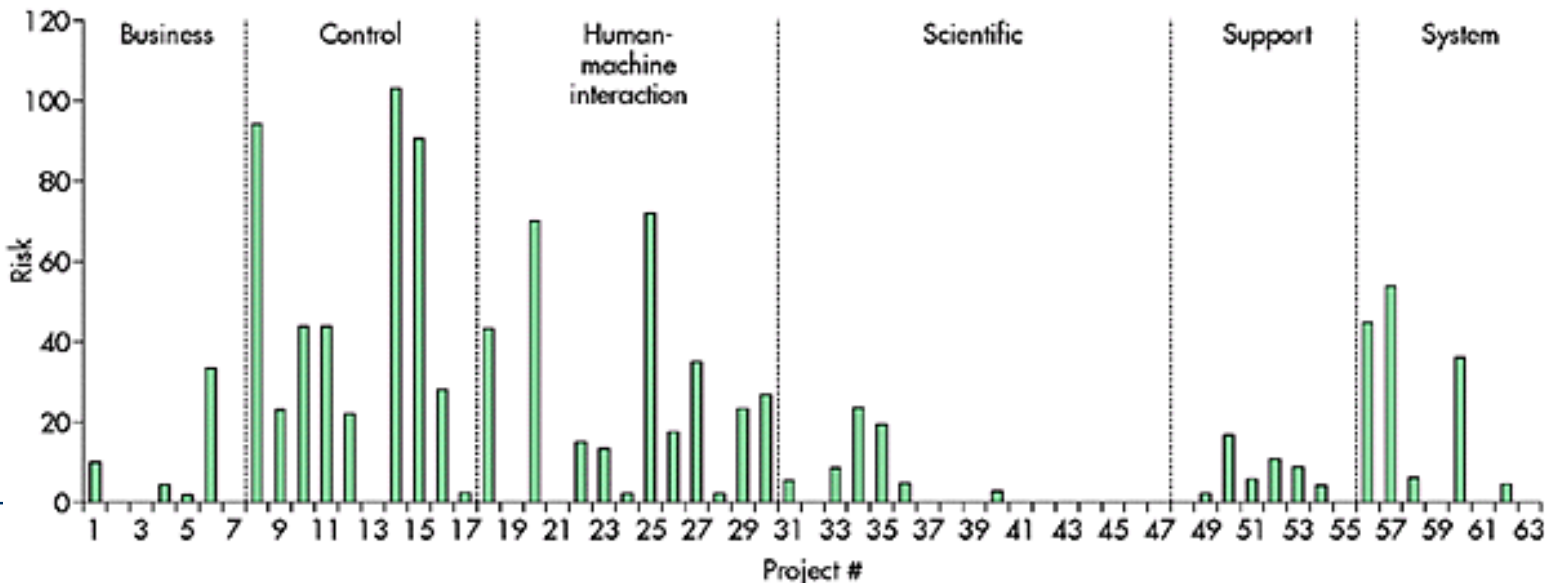
1. Development Process
 - a. Formality
 - b. Suitability
 - c. Process Control
 - d. Familiarity
 - e. Product Control
2. Development System
 - a. Capacity
 - b. Suitability
 - c. Usability
 - d. Familiarity
 - e. Reliability
 - f. System Support
 - g. Deliverability
3. Management Process
 - a. Planning
 - b. Project Organization
 - c. Management Experience
 - d. Program Interfaces
4. Management Methods
 - a. Monitoring
 - b. Personnel Management
 - c. Quality Assurance
 - d. Configuration Management
5. Work Environment
 - a. Quality Attitude
 - b. Cooperation
 - c. Communication
 - d. Morale

C. Program Constraints

1. Resources
 - a. Schedule
 - b. Staff
 - c. Budget
 - d. Facilities
2. Contract
 - a. Type of Contract
 - b. Restrictions
 - c. Dependencies
3. Program Interfaces
 - a. Customer
 - b. Associate Contractors
 - c. Subcontractors
 - d. Prime Contractor
 - e. Corporate Management
 - f. Vendors
 - g. Politics

"program" meint hier "Vorhaben", also Projekt

- Themenausgabe über Risikomanagement
 - <http://www.computer.org>
- Lohnende Lektüre!
- Nachfolgend einige Highlights
 - vor allem praktische Hinweise
 - ohne Anspruch auf Vollständigkeit



Tim Lister: Is Risk Management Risky? Point:
**Risk Management Is
Project Management for Adults**

- Stupid risks are those taken even though they could be mitigated or avoided
- No project *ever* runs exactly to plan!
- Software process describes what is common from project to project.
Risk management is about what is different.
- Beware of anyone who tries to turn RM into a wholly standard process



"Good luck on your project – just don't expect it."

Marvin J. Carr: Is Risk Management Risky? Counterpoint: **Risk Management May Not Be for Everyone**

Many organizations are unable to manage risks effectively for any of the following three reasons:

1. A risk-averse culture

- "the severity of the risks identified was inversely proportional to the observer's hierarchical position within the company"

2. An inadequate management infrastructure to support effective risk management

- tendency to manage problems, rather than risks
- ad hoc methods identify global, acceptable risks only

3. The lack of a systematic and repeatable method to identify, analyze, and plan risk mitigation

Welcher ist am schwierigsten?

Ray C. Williams, Julie A. Walker, Audrey J. Dorofee:
Putting Risk Management into Practice

- Tasks:
 - 1 identify, 2 analyze, 3 plan, 4 track, 5 control, 6 communicate
- Practical advice:
 - 1–6 Emphasize sponsorship
 - 1 Make risks visible: It's harder to ignore if written down
 - use a standard format, e.g., condition/consequence
 - 2 Insistence on quantification leads to "analysis paralysis"
 - 3 Classify/group risks that can be solved together
 - 3 You cannot fix everything. Some risks must be accepted.
 - 4 Will you know when mitigation worked and a risk is gone?
 - 5 Review and adjust mitigation plans
 - 6 Publish & archive information about risks



Was NICHT funktioniert

- Vage oder zu allgemeine Risikoformulierungen
 - z.B. "Wir könnten unser Budget überschreiten"
- Separate Bekämpfungsteams und -strategien für stark überlappende Risiken
 - denn überlappende Lösungsansätze erzeugen hohe Reibungsverluste
 - Beispiel: "Testwerkzeuge unzureichend leistungsfähig" und "Teststrategie nicht klar definiert"
- Risikosuche oder -priorisierung zu oft wiederholt. Problem?
 - macht betriebsblind
- Zu viele Risiken managen
 - kleine Risiken lohnen evtl. den Aufwand gar nicht
 - große bleiben dann wegen Ressourcenengpass evtl. liegen

Was NICHT funktioniert (2)

- Gegenmaßnahmen, deren Durchführung nicht verfolgt wird
 - bleiben gegenüber dem Tagesgeschäft auf der Strecke
- ...oder deren Erfolg nicht zu ermitteln ist
 - oder nicht ermittelt wird

(from "Putting Risk Management into Practice")

- When you have effective risk management in place
 1. You can evaluate activities in work plans for their effect on overall project risk
 2. You can routinely apply lessons learned to avoid crises in the future
 3. You can focus your planning on avoiding future problems rather than solving current ones
 4. You can structure important meeting agendas to discuss risks before discussing specifics of technical approach and current status

Danke!