

Achtung:  
Deutsch/Englisch-Gemisch

# Vorlesung "Softwaretechnik"

## Buchkapitel 14

### **Projektmanagement 3**

Lutz Prechelt, Bernd Brügge, Allen Dutoit

Freie Universität Berlin, Institut für Informatik

<http://www.inf.fu-berlin.de/inst/ag-se/>

- Teams
  - Sportteam oder Chor?
  - Organisationsstrukturen
  - Rollen
  - Kommunikationsstrukturen
- Psychologische Faktoren
  - Schätzen von Wahrscheinlichkeiten
  - Motivation
  - Attribution
  - Haltungen
    - Selbsterfüllende Prophezeiungen
  - Soziale Einflüsse

# 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

- Qualität der Mannschaft: Wer entscheidet das Spiel?
- Michael Jordan, vermutlich bester Basketballer aller Zeiten

Der Beste (der Star) entscheidet!



# Teams: Gewinnfaktoren (2)

- Preisfrage: Wer ist der beste Chorsänger aller Zeiten?
- Wer entscheidet über die Qualität des Konzerts?
- Hannes Schmilthup (gestern Nacht beim Schachspielen versumpft)



- SW-Entwicklung hat beide Aspekte:
  - Ein Star kann über die Genialität der gelieferten Funktionalität oder der Architektur entscheiden
  - Ein Versager kann über die Qualität der Realisierung entscheiden
- Hat das Team beide, Star und Versager, kommt es auf die Art des Produkts an
  - Ein Routineprodukt kann von einem Star nur mäßig profitieren
    - es kommt vor allem darauf an, Schwächen zu vermeiden: An- oder Abwesenheit von Versagern ist oft entscheidend.
  - Ein sehr innovatives Produkt kann eventuell nur mit einem Star überhaupt gelingen
    - es kommt vor allem darauf an, Stärken zu optimieren
    - denn es kann selbst mit Qualitätsmängeln sehr erfolgreich sein

- Bei den meisten Projekten zieht ein Star lediglich die Qualität ein Stück nach oben
  - Meist ist dies der/die Architekt/in:  
Teile passen besser zusammen,  
es sind weniger Korrekturen nötig,  
dadurch bleibt mehr Zeit zum Programmieren und Testen
- Bei den meisten Projekten zieht ein Versager (oder auch mehrere) lediglich die Qualität ein Stück nach unten
  - Das Produkt wird größer, braucht mehr Speicher und CPU-Zeit,  
funktioniert unzuverlässiger, ist schlechter bedienbar und schlechter dokumentiert
- Dennoch ist das Personal der wichtigste entscheidbare Faktor für ein Softwareprojekt



- Als Projektmanager schreibt man Ihnen oft diverse Faktoren für die Durchführung vor, z.B.:
  - Betriebssystem/DBMS, Programmiersprache, Werkzeuge, Prozessmodell, Teammitglieder
- Unter typischen Umständen ist das Team davon der weitaus wichtigste (vor allem bei kleinen Projekten):
  - Der Leistungsunterschied der schwächeren zur stärkeren Hälfte aller SW-Ingenieure liegt ca. bei 1:2
  - Der Unterschied zwischen einer/m unterdurchschnittlichen (aber nicht ganz inkompetenten) und einer/m hervorragenden SW-Ingenieur/in liegt in komplexen Situationen oft bei 1:10
  - Bei kleinen Teams (z.B. 3 Leute) dominiert das alles Übrige
  - Bei größeren Teams ergibt sich meistens ein mittlerer Fall
- Bei größeren Projekten kommt es deshalb letztlich vor allem darauf an, ob das Team gut zusammenarbeitet
  - Genau wie bei Sportmannschaften und bei Chören

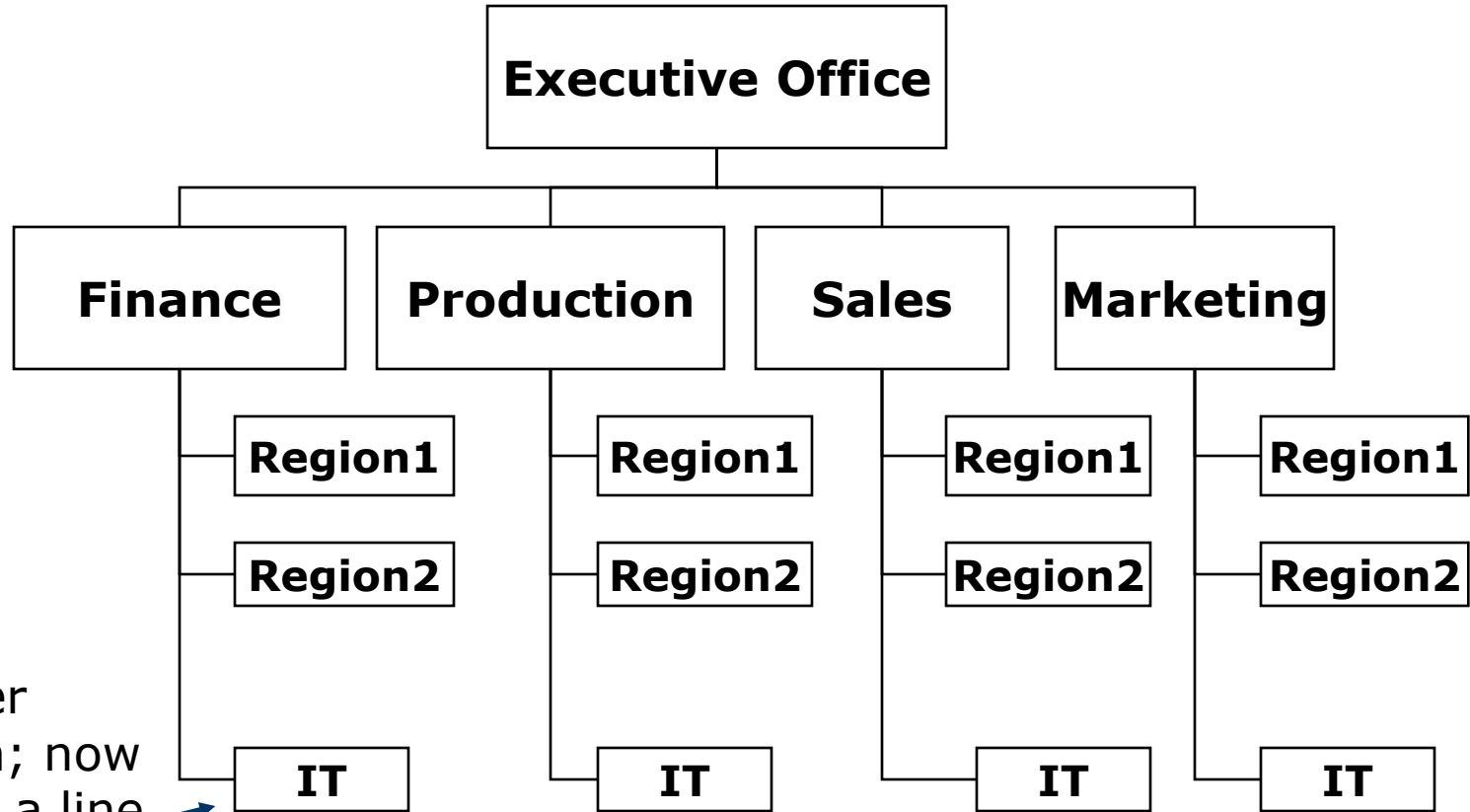
# Faktoren für Zusammenarbeit eines SW-Teams

- Motivation der Beteiligten
  - siehe gleich unter "psychologische Faktoren"
- Kommunikationsfluss
  - siehe später unter "Projektleitung"
- Auftreten und Behandlung von Konflikten
  - siehe (teilweise) unter "Persönlichkeitstypen"
- Geeignete Regelung der Zuständigkeiten und direkte Einbindung aller Betroffenen
  - folgt jetzt gleich

- **Definition Organization:** A set of *organizational* units and their different *relationships* with each other
- Organizational units can be organized according to many different categories (e.g. by function or by project type). Typical examples of organizational units:
  - in a functional organization: Research, Development, Marketing, Sales, ...
  - in a project organization: Project 1, Project 2, ...
- A organization usually has 3 different types of relationships between organizational units.
  - Reporting structure: To collect status information
  - Decision structure: To propagate decisions
  - Communication structure: To exchange information

- **Definition:** In a **functional organization** ("Linienorganis.") participants are grouped into so-called **departments**, each of which addresses a function
- Examples of departments:
  - Traditional businesses: Research, development, production, sales, finance
  - In software organizations the departments correspond to the activities in the software process: Analysis, design, integration, testing departments
- Key properties:
  - Projects are usually pipelined through the departments of a functional organization. The project starts in research, then it moves to development, then it moves to production, ...
    - Only a few participants are involved begin to end
  - Separate departments often address the same cross-functional needs (Examples: configuration management, IT infrastructure)

# Example of a Functional Organization



no longer common; now typically a line in itself →

Line organization of a "traditional business"



- Advantages:
  - Members of a department have a good understanding of the functional area they support
  - Departments don't compete with one another to get the support of their support teams
- Disadvantages:
  - Because each department has its own support team, work procedures and reporting systems vary
  - It is difficult to make major investments in equipment and facilities
    - Example: Two departments (remaining budget 5,000 € each) need a printer that costs 9,000 €. Both need only 40% of its capacity.
    - Neither department can buy it alone
  - High chance for overlap or duplication of work among departments
  - Tendency towards "we vs. they" thinking

- In a **project organization** participants are grouped into **projects**
  - each of which has a problem to be solved within time and budget
- **Key properties:**
  - Teams are assembled for a project as it is created. Each project has a project leader.
  - In principle, all participants are involved in the project as a whole
    - Rather than in just one aspect of the project
  - Teams are disassembled when the project terminates

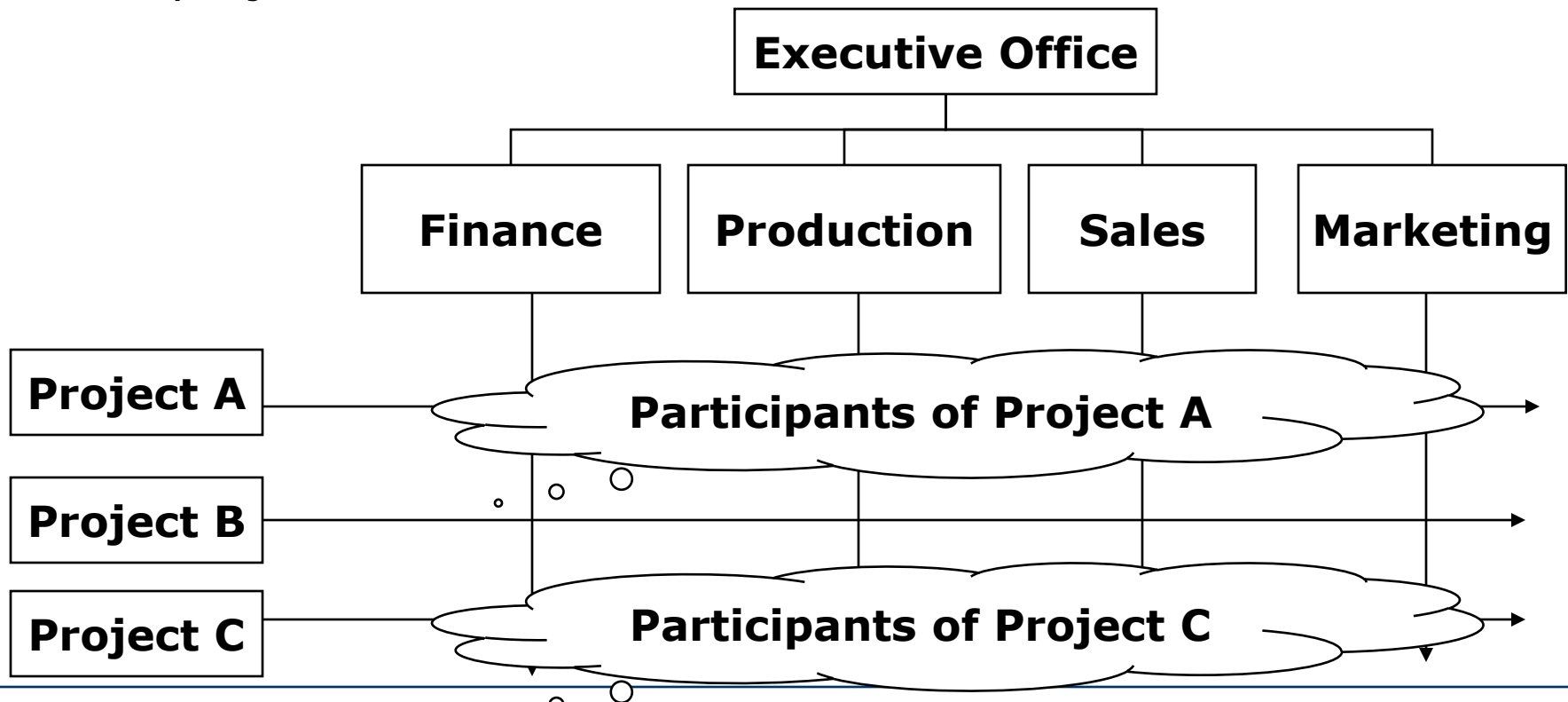
- Advantages
  - Responsive to new project requests
    - because the project is newly established and can be tailored around the problem
  - New people can be hired/selected who are very familiar with the problem or who have special capabilities
  - There is less overlap and waste of staff workload
- Disadvantages:
  - Teams cannot be assembled rapidly.  
Often it is difficult to manage the staffing/hiring process.
    - Difficult to have each employee in exactly one project at any time
  - Because there are "no predefined lines", roles and responsibilities need to be defined at the beginning of the project

Pure project organizations are rare



# Matrix Organization

- In a matrix organization, participants from different departments of the functional organization are assigned to work on projects as they are created.
- The project manager and team members may be assigned to the project for less than 100 % of their time



- Advantages:
  - Teams for projects can be assembled rapidly
  - Scarce expertise can be applied to different projects as needed
  - Consistent work and reporting procedures can be used for projects of the same type
- Disadvantages:
  - Team members are often not familiar with each other
  - Team members may have different departmental working styles
  - People have at least two bosses (project managers, department head) and hence multiple loyalties
    - This leads to conflicts
- Matrix organizations are more common than purely functional or pure project organizations

- Team members must respond to two different bosses with different focus:
  - Focus of the functional manager: **Assignments to different projects, performance appraisal**
  - Focus of the project manager: **Work assignments, project team support**
- Team members working on multiple projects have competing demands for their time
  - Team members working on more than one project have even more bosses to report to
  - Some people who have claim on the team member's time may be at similar levels in the organization's hierarchy

# Key Roles in Organizations

- **Project Manager:** The person ultimately responsible for the successful completion of the project
  - in a project or matrix organization, often also in functional org.
- **Project Team Member:** Participants who are responsible for performing individual activities and tasks
  - in a project or matrix organization
- **Functional Manager, Line Manager** ("fachlicher Vorgesetzter", "Linienvorgesetzter"): The team member's supervisor in the department
  - in a functional or matrix organization
- **Upper management:** People in charge of the departments and/or projects
- In the following we focus on roles in project and matrix organizations

# Other Team Member Roles

- Project Management
  - Coach
  - Team leader
  - Planner
- Meeting Management
  - rare
  - Minute Taker, Scribe
  - Primary facilitator
- Development
  - Analyst
  - Designer (Software Architect)
  - Programmer
  - Tester
  - Maintainer
  - Trainer
  - Document Editor
  - Web Master
  - Configuration Manager

Team members have technical responsibilities  
plus communication responsibilities

# Note: List of project stakeholders

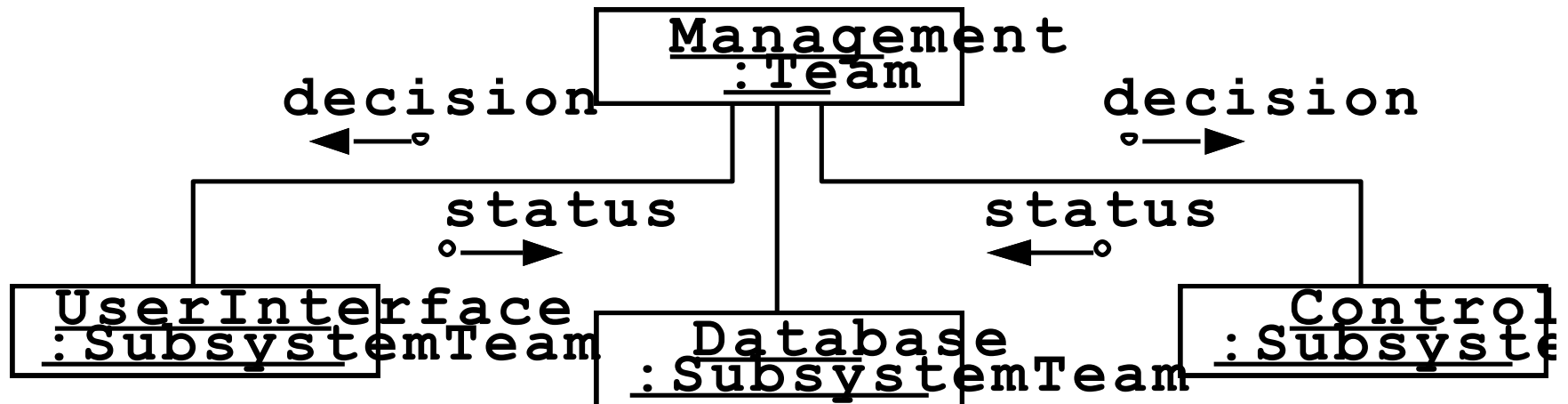
## There are many more stakeholders beyond the development team!

- Internal
  - Upper Management
  - Requester
  - Project Manager
  - Team members
  - People with special knowledge
- External:
  - Clients or customers
  - Collaborators
  - Vendors, suppliers and contractors
  - Regulators
  - The general public
- Support Groups
  - Human Resources
  - Legal services
  - Contracting
  - Finances
  - Security
  - Computing Facilities
- End users of your project's deliverables
- People who will maintain or support your deliverables

- There is hardly ever any project where every member has only one role
- Mapping many roles to few people
  - Each project member assumes several roles
    - They are "wearing different hats"
  - Danger of over-commitment → helpful to make rules less formal
  - Need for load balancing
- Mapping many roles to too many people (overlarge project)
  - rare
    - Most members have only one role
    - Some people don't have significant roles (essentially bystanders)
    - People may loose touch with the project

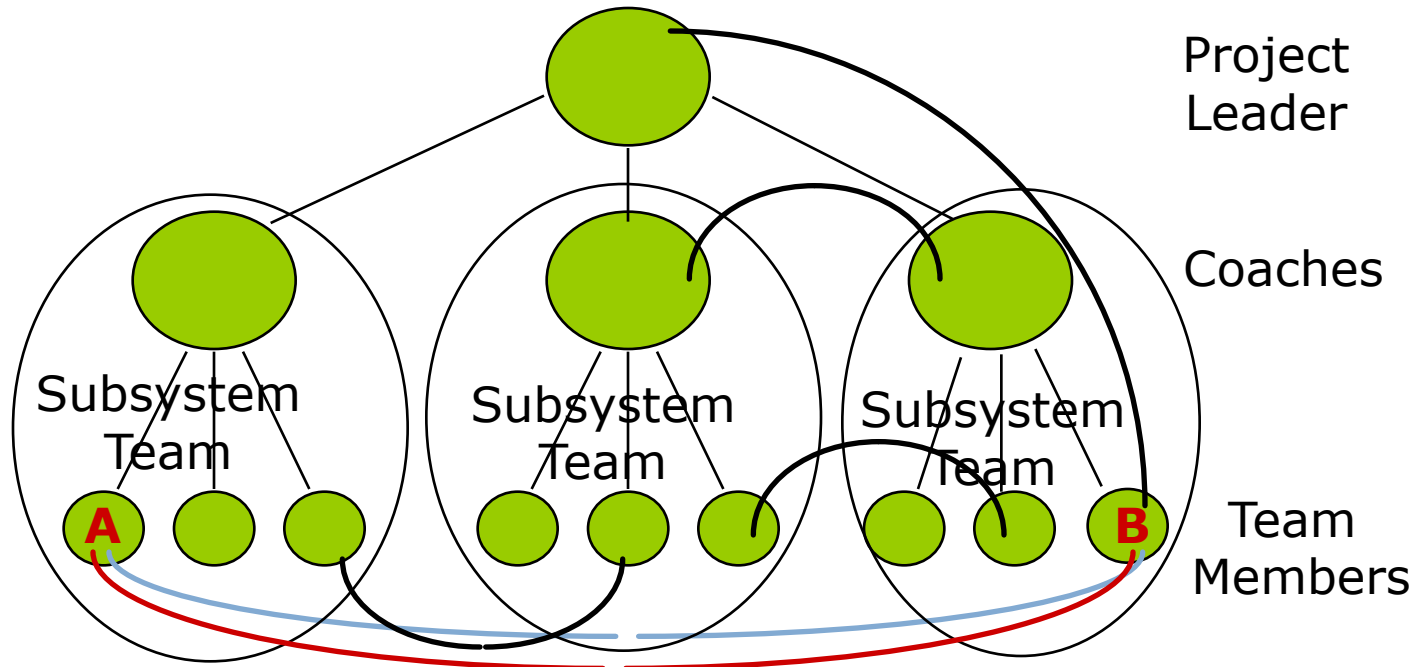
# Example: A project's reporting and decision structure

- The **developers** make local decisions and reports them via a status report to the leader (team leader, project manager)
  - They also communicate with each other
- The **team leader**, who has a local overview of the subsystem, can override these decisions
  - She reports them to the project manager
- The **project manager**, who has a global view of the project, can virtually override any decision





# Communication must not be constrained to the hierarchy



**A wants to talk to B: Communication Flow**

**B wants to make sure A does a certain change: Decision Flow**

# Observations on Organizational Structures

- Strictly hierarchical structures
  - "Reports", "Decides" and "Communicates-With" all mapped on the same association (vertically along the hierarchy)
  - Does not work well with iterative and incremental software development process
  - Manager is not necessarily always right
- Modern project structures
  - "Reports", "Decides" and "Communicates-With" are different associations
    - Decisions are expected to be made at each level
  - Cuts down on bureaucracy
  - Reduces development time
  - Harder to manage
    - agreement on authoritativeness of decisions is needed

- A linear responsibility chart is a matrix that depicts the role that each project participant will play in different activities identified in the work breakdown structure
- Rows: Project activities
- Columns: Roles/Project participants
- Entries: Type of responsibility
  - **P** (Primary responsibility):  
You are committed to ensure that the desired result is achieved
  - **S** (Secondary responsibility):  
You are committed to some portion of the result
  - **A** (Approval):  
You are not doing the work, but will approve the results
  - **R** (Review):  
You will review and comment on the work product of an activity
  - **O** (Output): You will receive the work product of the activity
  - **I** (Input): You will provide input for the activity

# Example of a Responsibility Chart

	Project Manager	Team Leader	Team Member A	Team Member B
<b>Develop SPMP</b>	<b>P</b>	<b>S</b>		
<b>Run weekly meeting</b>		<b>A</b>	<b>P</b>	<b>S</b>
<b>Write SDD</b>	<b>P</b>	<b>S</b>	<b>S</b>	<b>S</b>
	<i>Legend:</i> <b>P</b> = Primary responsibility <b>S</b> = Secondary responsibility <b>A</b> = Approval			

- 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

- Softwareprojekte werden stark von psychologischen Phänomenen bestimmt
- Diese betreffen zwei Bereiche:
  - Zuverlässigkeit der Denkvorgänge Einzelner (Kognitionspsychologie)
  - Gegenseitige Beeinflussung der Projektteilnehmer (Sozialpsychologie)

## Wo sind diese Effekte relevant?

- Kognitionspsychologie:
  - Technische Arbeiten (Programmieren, Entwurf, Test, ...) ←
  - Entscheidungen im SW-Prozess (Projektmanagement) ←
- Sozialpsychologie:
  - Entscheidungen im SW-Prozess (Projektmanagement) ←
  - Motivation, Haltungen, Verhalten des Teams ←

- Kognitionspsychologie
  - Induktives Schließen (Basisratenregel, Repräsentativitätsheuristik, Verfügbarkeitsheuristik)
  - Bestätigungsneigung
- Sozialpsychologie
  - Motivation
  - Attribution
  - Kognitive Dissonanz
  - Selbsterfüllende Prophezeiungen
  - Konformität, Reziprozität
  - soziale Förderung, soziales Faulenzen
  - Bindung an Vor-Engagement

- Menschen sind sehr leistungsfähig darin, vom Speziellen auf Allgemeines zu schließen (Induktion)
  - Dies ist die Grundlage fast allen Lernens
- Wenn die Schlüsse unsicher sind (nur Wahrscheinlichkeiten), überschätzen wir aber oft ihre Qualität
- Beispiel:
  - Grundsätzlich sind Ihre Mitarbeiter zufrieden. Wenig Fluktuation
  - Letzten Monat kündigte ein wichtiger Mitarbeiter, nachdem er in zwei aufeinander folgenden Wochen je einen Tag Urlaub gehabt hatte
  - Nun will ein anderer wichtiger Mitarbeiter, der letzte Woche einen Tag Urlaub hatte, nochmal einen Tag Urlaub
  - Sie befürchten, dass er höchstwahrscheinlich bald kündigen wird



- Eigentlich sollten wir in diesem Beispiel die Wahrscheinlichkeit als gering einschätzen:
  - 2 x 1 Tag Urlaub kommt öfters mal vor
  - Danach Kündigen kommt nur sehr selten vor (bislang nur 1x)
  - Das wäre die Anwendung der **Basisratenregel**:  
"Seltene Ereignisse sind unwahrscheinlich"
- Jedoch benutzen wir oft die **Verfügbarkeitsheuristik**:
  - Ein Ereignis, das ich kürzlich erlebt habe, schätze ich als wahrscheinlicher ein, als angebracht wäre
- Hinzu kommt oft die **Repräsentativitätsheuristik**:
  - Wenn ein Ereignis  $x$  zu einer bekannten Klasse  $X$  gehört, erwarte ich jetzt auch Ereignisse  $a, b, \dots$  wenn diese oft mit Ereignissen der Art  $X$  einher gehen
  - Auch das führt oft zur Überschätzung von Wahrscheinlichkeiten

Diese Überschätzungen können zu negativen Konsequenzen im Projektmgmt. führen

- Angewandt auf positive Ereignisse:

- Vernachlässigung des Risikomanagements oder der Qualitätssicherung
- Nachlässigkeit in der Planung

*"Alles läuft prima!"*

- Angewandt auf negative Ereignisse:

- Unnötiger Mehraufwand für die Risikovorsorge
- oder Selbsterfüllende Prophezeiung (siehe dort)

*"Alles geht den Bach runter!"*

- Allgemein:

- Persönliche Eindrücke (z.B. aus Gesprächen) werden überbewertet gegenüber anderen Informationsquellen (z.B. Dokumente)
  - Das ist ein Nebeneffekt der Verfügbarkeitsheuristik:  
Persönliche Begegnungen sind eindrucksvoller, also besser verfügbar

- Wollen wir etwas auf Brauchbarkeit überprüfen und wir erwarten, dass es brauchbar ist, dann neigen wir dazu, nur solche Beobachtungen und Prüfungen zu machen, die die Brauchbarkeit bestätigen
  - aber nicht solche, die die Brauchbarkeit widerlegen könnten
  - Psychologische Bezeichnung: **Bestätigungsneigung**
- Beispiel:
  - Bei der Durchsicht eines Dokuments, das von einer bekanntermaßen kompetenten Kollegin stammt, neigt man dazu, sich nur zu vergewissern dass alle benötigten Inhalte da sind,
  - anstatt zu überprüfen ob alle da sind
  - oder gar gezielt zu suchen was nicht da ist.

Motivation für ein Vorhaben X wird von drei Faktoren bestimmt:

- 1. Grundmotivation für X-artiges (→ Persönlichkeitsstruktur)
  - Ehrgeiz: Wunsch, Hindernisse zu überwinden
  - Machtstreben: Wunsch, Umgebung zu kontrollieren
  - Soziale Bedürfnisse: Wunsch nach Beziehungen u. Anerkennung
  - Neugier: Wunsch, Umgebung zu erforschen und zu verstehen
- 2. Subjektiver Wert eines eventuellen Erfolgs
  - am stärksten variabler Faktor
- 3. Subjektive Einschätzung der Erfolgsaussichten
  - (nicht zu verwechseln mit der tatsächlichen Wahrscheinlichkeit!)

# Motivation: Einfluss von Projektmgmt. (1)

- Grundmotivation kann bestmöglich genutzt werden  
Aufgaben an Personen mit passender Grundmotivation vergeben:
  - Exploratives an Neugierige
  - Kommunikatives an sozial Interessierte
  - Leitungsaufgaben an Machtbewusste
- Grundmotivation kann beschädigt werden
  - durch Belohnungen, die nicht genug von der Leistung abhängen
    - z.B. Bezahlung nach Menge ohne Beachtung der Qualität
  - Normalerweise hat Softwarebauern für SW-Ingenieure eine hohe "intrinsische Motivation"

# Motivation: Einfluss von Projektmgmt. (2)

- Subjektiver Wert kann gesteigert werden
  - durch Aufzeigen von Sinnhaftigkeit
    - z.B. für Feedback von Benutzern sorgen
  - durch Anlehnung an eine vorhandene Grundmotivation
    - siehe oben
  - durch Belohnung
    - Lob, Freizeit, Geld, Party etc.
- Subjektiver Wert kann gesenkt werden
  - durch mehrfaches Ausbleiben einer Belohnung
    - z.B. Dokumente schreiben, die niemand liest
- Subjektive Erfolgsaussicht kann verbessert werden
  - Ermutigung, gute Unterstützung
- Subjektive Erfolgsaussicht kann beschädigt werden
  - "Erlernte Hilflosigkeit"

- Jeder Mensch bildet ständig Theorien, mit denen er sich das Verhalten anderer Menschen erklärt
  - Dieser Prozess wird Attribution genannt
  - Die Theorien sind implizit und oft kaum verbalisierbar
- Wichtigste Arten von Erklärungselementen:
  - Dispositive Faktoren: Eigenschaften der erklärten Person
  - Situative Faktoren: Eigenschaften der aktuellen Situation
- Welchen Mix der Elemente unsere Theorie benutzt, hängt von drei Eigenschaften des beobachteten Verhaltens ab:
  - Konsens: Verhalten, das allgemein viele Personen zeigen  
→ wir vermuten überwiegend situative Faktoren
  - Konsistenz: Verhalten, das eine Person immer wieder zeigt  
→ wir vermuten überwiegend dispositive Faktoren
  - Eigentümlichkeit: Verhalten, dass in einer Situation immer wieder auftritt → wir vermuten überwiegend situative Faktoren
- Diese Regeln stimmen tendenziell, aber wir machen Fehler:

- Fundamentaler Attributionsfehler:
  - Wir neigen dazu, einen viel zu hohen Anteil dispositiver Faktoren zu unterstellen
  - denn wir verstehen die situativen Gründe nur für unser eigenes Verhalten gut
    - Bsp: "Person A hat Modul X nicht rechtzeitig fertig → ist inkompetent"
    - Kann zu unnötig kritischer Beurteilung von Persönlichkeiten führen
- Falscher Konsens:
  - Wir neigen dazu, unsere eigene Ansicht für typisch zu halten
    - Führt zu Überraschungen, wenn im Projekt nicht genügend viel klar und explizit kommuniziert wird





- Eigenerfolg und Fremdversagen:
  - Wir neigen dazu, unsere eigenen Erfolge dispositiven Faktoren zuzuschreiben, unsere Misserfolge jedoch situativen
    - Das ist nötig zur Wahrung eines positiven Selbstbildes
    - Führt in Verbindung mit dem fundamentalen Attributionsfehler dazu, dass wir andere im Vergleich zu uns selbst zu kritisch beurteilen
- Verursachungsillusion:
  - Wir neigen dazu, die Wirksamkeit unserer eigenen Handlungen stark zu überschätzen
    - Bsp: "not invented here"-Syndrom
    - Bsp: "A hat Problem X nur so gut lösen können, weil ich ihm Tipp Y gegeben habe."
    - Führt dazu, dass bei der Verteilung von Belohnungen sich fast jeder ungerecht behandelt fühlt
- Repräsentativitätsheuristik:
  - Wir neigen zum Denken in Stereotypen

- Ein wichtiger Einfluss auf unser Handeln sind veränderliche dispositive Faktoren namens *Haltungen*
- Haltungen zu einem Thema T haben drei Aspekte:
  - Affektive Komponente: Die Gefühle, die wir gegenüber T haben
    - Bildet sich durch eigene Erlebnisse oder Übernahme der (evtl. unterschwellig wahrgenommenen) Gefühle anderer
  - Kognitive Komponente: Was wir über T zu wissen glauben
    - Wird meist von anderen übernommen, nur selten selbst gebildet
  - Verhaltenskomponente: Die Handlungen, zu denen wir im Zusammenhang mit T neigen
    - Die Verhaltenskomponente wird nicht nur durch die anderen beiden bestimmt

Warum steht die Verhaltenskomponente oft im Widerspruch zur affektiven und kognitiven?

- Allgemeinheitsgrad:
  - Haltungen sind allgemein, Handlungen müssen spezifisch sein
- Relevanz für Ziele:
  - Eine Haltung zu haben, kostet nicht viel Aufwand, aber eine Handlung danach lohnt sich in vielen Fällen subjektiv nicht
- Selbstattribution:
  - Eine von anderen übernommene Haltung fließt oft nicht in Handeln aus
- Verhaltensbeschränkungen:
  - Äußere Umstände machen entsprechendes Handeln oft schwierig
- Wie gehen wir mit solchen Widersprüchen um?:

- Klaffen Haltungen und Handlungen auseinander, empfinden wir *kognitive Dissonanz*
  - Das ist unangenehm, denn wir wollen, dass unsere Haltungen konsistent sind
    - wenigstens einigermaßen
- Um die Dissonanz zu verringern, gibt es drei Wege:
  1. Beseitigen des dissonanten Elements
    - durch Ändern der Haltung oder des Verhaltens
  2. Zufügen konsonanter Elemente
    - durch Ändern der Haltung oder des Verhaltens
  3. Senkung der Wichtigkeit
    - durch Ändern der Haltung



# Kognitive Dissonanz: Beispiele

- Haltung: "Wir sind ein sehr erfolgreiches Projekt"
- Dissonantes Element: "Wir haben 378 offene Defekte"
- Auflösung:
  - Haltung ändern:  
"Wir sind ein *im Wesentlichen* erfolgreiches Projekt"
  - Konsonante Elemente zufügen:  
"Davon gehören 327 zu einer minderschweren Kategorie"
  - Konsonante Elemente zufügen:  
"Wir haben schließlich unter *enormem Zeitdruck* gearbeitet"
  - Dissonante Elemente entfernen (beste Lösung!):  
"Wir starten jetzt eine intensive Defektbeseitigungsphase"
- Gefahr: Weitere dissonante Elemente ignorieren
  - "und außerdem sind wir 11 Wochen hinter dem Zeitplan zurück"
  - "und die Benutzer sind auch extrem unzufrieden"

- Wenn eine Haltung so intensiv ist, dass die Klärung immer im Beseitigen der dissonanten Elemente besteht, nennt man sie eine Selbsterfüllende Prophezeiung
- Das kann positiv sein, wie eben beschrieben:
  - "Wir sind erfolgreich" → Defektbeseitigungskampagne → Erfolg
- Aber: Negative Haltungen führen dann zur Katastrophe
  - "Projekte, die ihren Zeitplan einhalten, gibt es gar nicht"
  - "Defektfreie Software gibt es gar nicht"
  - "UML-Werkzeuge funktionieren sowieso nicht richtig"
  - "Anwender sind nie zufrieden"
  - u.s.w.
- Sehr wirkungsmächtiges Phänomen!
- **Sehr gefährlich!**

# Beispiel: Selbsterfüllende Prophezeiung

Versuchsperson(en)



Telefonat  
mit immer  
gleicher  
Partnerin



Aufzeichnen (nur  
die Frauenstimme)



Schiedsrichtern vorspielen:  
Sie finden die Frauenstimme  
sympathischer, wenn der Mann  
zuvor das attraktivere Bild  
gesehen hatte!

zeige ein  
Bild



# Selbsterfüllende Prophezeiungen: Vorsicht!

- Da selbsterfüllende Prophezeiungen so mächtig sind, möchte man negative Prophezeiungen unbedingt vermeiden
- Die erfolgreichsten Organisationen verwenden positive selbsterfüllende Prophezeiungen!





- Konformität

- Wir möchten tendenziell so sein, "wie die Anderen"
- Sehr gefährlich, wenn eigentlich eine abweichende Meinung benötigt wird (z.B. bei allgemeiner Fehleinschätzung)



- Soziale Förderung (social facilitation)

- Wenn jemand an meinen Taten teilnimmt (z.B. beobachtet, mitmacht), erhöht das meine Motivation
  - zumindest bei einfachen Arbeiten erhöht das die Leistung

- Soziales Faulenzen (social loafing)

- Wenn mein eigener Beitrag zu einer Gruppenleistung nicht als solcher erkennbar ist, beschränkt das meine Motivation

- Reziprozität
  - Was mir angetan wurde, möchte ich auch anderen antun.
  - Führt im negativen Fall zur Eskalation von Unfreundlichkeiten und Kooperationsverweigerung
  - Führt im positiven Fall zu hoher Hilfsbereitschaft und Motivation
- Vor-Engagement
  - Wenn man einmal einer Sache zugestimmt hat, fällt es schwer, sich später davon wieder zurückzuziehen
  - Gefahr: Salami-Taktik bei schlechten Nachrichten, Kosten etc.
  - Gefahr: Weiterverfolgen eingeschlagener falscher Wege

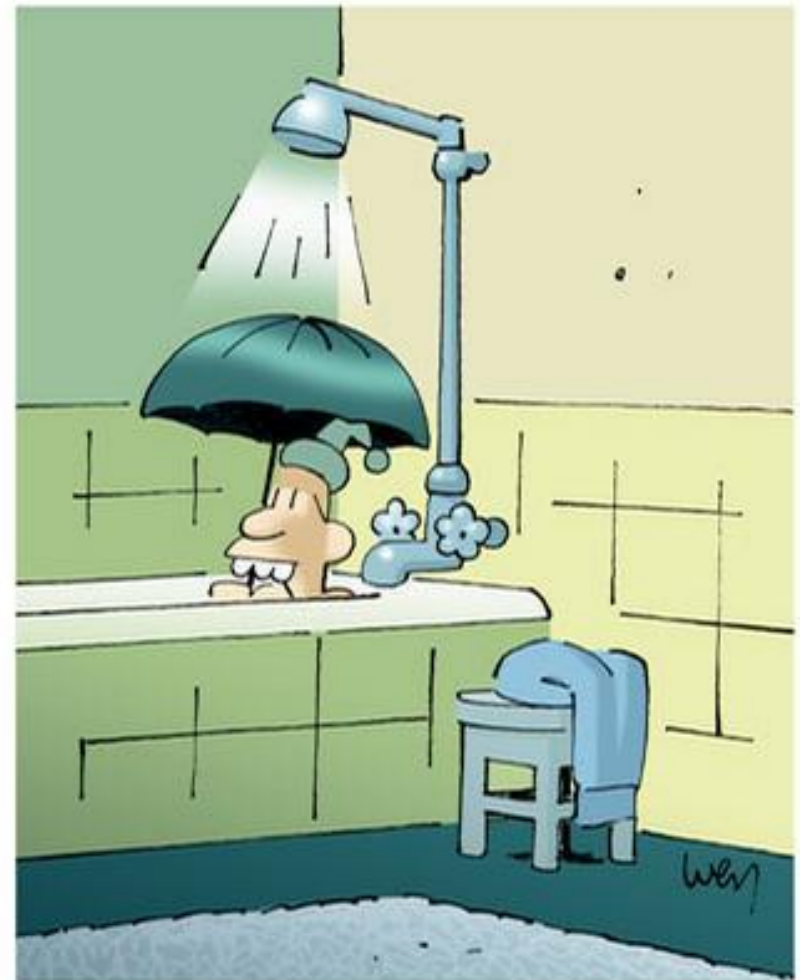
# Vor-Engagement: Fehler eingestehen!

- Rechtzeitiger Richtungswechsel nach Fehlentscheidung ist ein wichtiger Erfolgsfaktor!



# Soziale Einflüsse (3)

- Traditionen
  - Wenn etwas längere Zeit üblich war, ist es schwierig, sich davon zu lösen
  - Gefahr: Veränderungen in den Randbedingungen können ehemals sinnvolles Verhalten inzwischen unsinnig gemacht haben
  - Extremfälle: In manchen Organisationen oder Teams gibt es quasi Rituale, die fast keinerlei Sinn ergeben



WIE ALLE HARTEN KERLE DUSCHTE AUCH HORST JEDEN MORGEN  
EISKALT!

# Danke!

Bitte halten Sie selber nach Beispielen  
für diese psychologischen Phänomene Ausschau.  
Nur so können Sie dieses wertvolle Wissen praxistauglich machen!

Es gibt noch viel mehr davon. Siehe z.B.  
[http://en.wikipedia.org/wiki/List\\_of\\_cognitive\\_biases](http://en.wikipedia.org/wiki/List_of_cognitive_biases)