

Vorlesung "Softwaretechnik"

Projektmanagement (PM)

Lutz Prechelt

Freie Universität Berlin, Institut für Informatik

- Was und wofür?
- Aufgabenfelder
 - Integrierende Aufgaben
 - Umfangsmanagement
 - Zeitmanagement
 - Kostenmanagement
 - Qualitätsmanagement
 - Personalmanagement
- Kommunikationsmngmt
- Risikomanagement
- Beschaffungsmngmt
- Schätzen
 - Schätzverfahren
 - Funktionspunktschätzung
 - Todesmarschprojekte

- Unterschiede in Wesen und Anwendbarkeitsbereich zwischen Agilem PM und Planendem PM verstehen
 - und warum wir hier öfter von Planendem PM sprechen werden
- Aufgabenbereiche bei planendem PM verstehen
- Methoden zum Schätzen des Gesamtaufwands 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.ändrgn.
 - **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 möglichst

2 Stile von Projektmanagement

Planend-leitend: **P**

- a) Explizite, dokumentierte, evtl. detaillierte Planung
- b) Projektleiter_in mit Weisungsbefugnis
- c) Projektsteuerung zur Planeinhaltung
- d) Geeignet für: kleine+große Teams, klare Anford., Vorhaben mit Nicht-SW-Anteilen

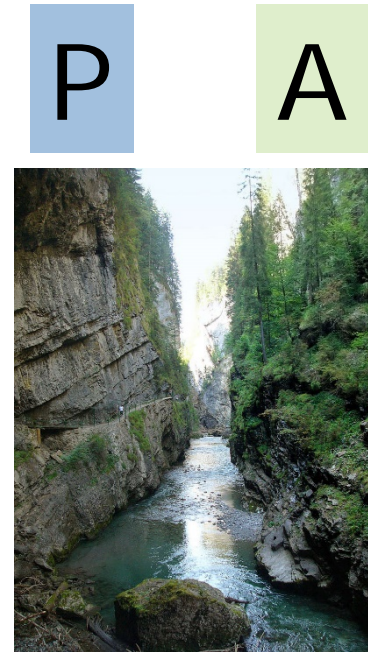
Agil-selbstorganisierend: **A**

- a) Geringere, implizitere Planung; kurzer Horizont
- b) Selbstorganisation: Gemeinsames Entscheiden
- c) Während Iteration: Koordination; Umsteuern nur dazwischen
- d) Geeignet für kleine Teams, auch unklare/veränderl. Anford.

Stark radikale Vorhaben sind mit beiden Stilen schwierig

➔ Risikomodell, mittlerer Planungsgrad

Lösung?



- Anna und Bodo ziehen um (entspricht Softwareprojekt) **P** **A**
- ...aus 2 getrennten WG-Zimmern in 3-Zimmer-Wohnung als billiger Umzug alleine
 - agiler Stil: Kisten irgendwie vollpacken, irgendwie rüberschaffen, dort beim Auspacken die neue Ordnung herstellen
 - kein Planungsaufwand, aber viel Rungelaufe am Ziel
 - planender Stil: Kisten gezielt für je 1 Zielzimmer vollpacken
 - Planungsaufwand, aber weniger Rungelaufe am Ziel
- ...aus 2 getrennten WG-Zimmern in 3-Zimmer-Wohnung mit 8 Umzugshelfern, die Ein- und Auspacken
 - agiler Stil: Zimmer am Zielort wiederherstellen lassen, dann umräumen → ineffizient
 - planender Stil: Zielorte aller Sachen gruppenweise markieren → Planungsaufwand, dann effizient; b. Planungsfehlern ineffizient

Situationen für Projektmanagement

Metaphorisches Beispiel (2)

- Anna und Bodo ziehen um (entspricht Softwareprojekt) P A
- ...aus 28-Zimmer-Villa in Beverly Hills
in heruntergekommenes Schloss an der Loire:
 - agiler Stil: führt ins Chaos
 - planender Stil ist nötig
 - Dies entspricht einem Softwareprojekt mit
 - großem Team,
 - unflexiblen Anteilen mit Hardware-Entwicklung und
 - ein paar radikalen Anteilen
- Für sehr große Projekte ist gutes (planendes) Projektmgmt
der wichtigste Erfolgsfaktor
 - siehe Flughafen BER

Ähnlich für versch. Aufgaben (auch z.B. Brückenbau, ...)

P

- Project Management Institute (pmi.org):
 - "A *project* is a **temporary** endeavour, undertaken to create a **unique** product, service, or result."
 - bitte nicht Projekt mit Produkt verwechseln
 - "Project Management is the application of knowledge, skills, tools and techniques to project activities to meet project requirements."
 - "Project management includes
 - (1) identifying requirements,
 - (2) establishing clear objectives,
 - (3) balancing the competing demands for time, quality, and cost,
 - (4) adapting the specifications, plans, and approach to the different concerns and expectations of the various stakeholders."
- DIN 69901:
 - "*Gesamtheit von **Führung**aufgaben, -organisation, -techniken und -mitteln für die Abwicklung eines Projektes*"

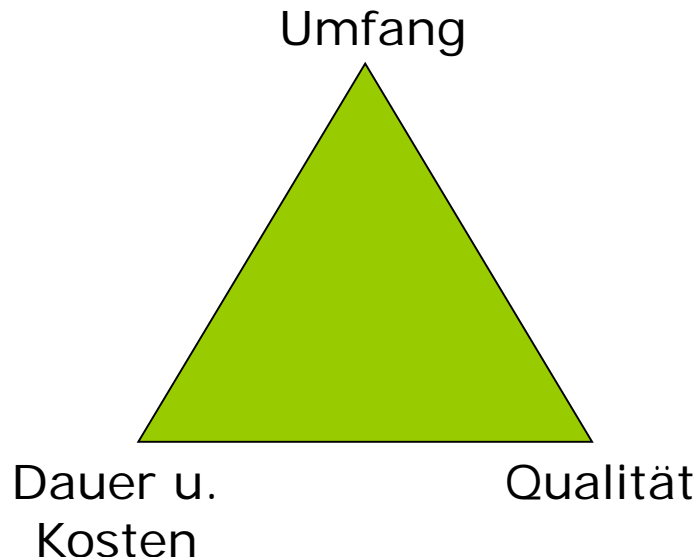


- Planendes Projektmanagement ist schwierig!
 - Deshalb gibt es viele Fehlschläge und viele zynische Aussprüche über das Projektmanagement, siehe z.B. http://en.wikiquote.org/wiki/Project_management
- Hier ein paar solche Sprüche (mit kleinem wahren Kern):
 - Persönlichkeit: *"If you're 6 months late on a milestone due next week but really believe you can make it, you're a project manager."*
 - Zeitschätzung: *"The hardest 10% of a project take 90% of the time and the remaining 90% take the other 90% of the time."*
 - Straffung von Zeitplänen: *"Even the nine most competent women cannot have one baby in only one month."*
 - Rolle von Planung: *"The nice thing about not planning is that failure comes as a complete surprise rather than being preceded by a period of worry and depression."*

P

Das magische Dreieck

- Normales Projektmanagement versucht, widerstreitende Kräfte zu balancieren:
Qualität, Kosten, Projektdauer
 - höhere Qualität verlangt meist längere Zeit und/oder erhöht Kosten etc.



- Im SW-Projektmgmt. spaltet man Qualität oft auf in
 - Umfang (Funktionalität) und
 - ist bei SW leichter unterwegs reduzierbar als z.B. bei einer Brücke
 - Qualität (z.B. Zuverlässigkeit, Bedienbarkeit)
- Dagegen hängen die Kosten eng mit der der Projektdauer zusammen (Personalaufwand)
 - und treten weniger als separate Entscheidungsgröße auf
- Deshalb ist bei SW-Projekten das Dreieck meistens **Umfang, Qualität, Projektdauer**
 - manchmal sieht man auch ein Viereck

P

Laut Project Management Institute (PMI) besteht PM aus
9 Aufgabenfeldern:

P

1. Integrierende Aufgaben (integration mgmt)
 2. Umfangsmanagement (scope mgmt)
 3. Zeitmanagement (time mgmt)
 4. Kostenmanagement (cost mgmt)
 5. Qualitätsmanagement (quality mgmt)
 6. Personalmanagement (human resource mgmt)
 7. Kommunikationsmgmt. (communication mgmt)
 8. Risikomanagement (risk mgmt)
 9. Beschaffungsmgmt. (procurement mgmt)
- Es folgt eine Kurzübersicht über deren Themen
 - anschließend betrachten wir ein paar Fragestellungen genauer

Vorbemerkung: Wann wie viel PM?

- Diese PMI-Aufgabenfelder sind tief im planenden Projektmanagement verankert
 - Wir diskutieren agile Ansätze als Alternativen
- Viele der nachfolgend beschriebenen Aufgaben sind nur für Großprojekte im vollen Umfang sinnvoll
- Bei mittelgroßen Projekten sollten viele davon vereinfacht werden
- Bei kleinen Projekten können/sollten manche ganz entfallen
 - Bei kompetentem Personal ist dann agiles PM meist einfacher



- Diese Tätigkeiten verbinden die Tätigkeiten der restlichen Felder miteinander
- Entwickeln des umfassenden Projektplans
 - enthält Teilpläne aus den anderen Aufgabenfeldern
 - Anmerkung: Ein solcher Plan existiert immer (evtl. nicht detailliert, evtl. nur in Köpfen; evtl. sogar vage)
- Projektleitung
 - konkrete Anweisungen geben, Entscheidungen fällen etc.
- Projektüberwachung
 - kontinuierlich den Ablauf gegenüber den Plänen verfolgen,
 - bei Abweichungen geeignete Maßnahmen ergreifen
- Planänderungs-Überwachung und -Steuerung
 - Änderungen am Plan auf Wirkungen abklopfen und entweder zurückweisen oder komplett ins Projekt einfädeln



- Umfangsdefinition
 - Was ist Aufgabe des Projekts? Was nicht?
 - Bei SW: → Anforderungsbestimmung ✓
- Erarbeiten einer Arbeitszerlegung (WBS: Work Breakdown Structure)
 - In welche handhabbaren Teile sollte man das Gesamtvorgehen (d.h. den konkreten Prozess) zerlegen?
 - Wie fügen sich diese Teile zum ganzen Vorgehen zusammen?
 - Bei SW hauptsächlich: → Entwurf + Prozessmodell ✓
- Umfangsverwaltung (bei SW: → Anforderungsverwaltung ✓)
 - Änderungen am Umfang (durch externe Einflüsse, z.B. Kundenwunsch) werden nicht einfach zugelassen, sondern die Auswirkungen überprüft.
 - Akzeptierte Änderungen werden sorgfältig in die Umfangsbeschreibung eingearbeitet
 - und Beteiligte geeignet benachrichtigt ✓

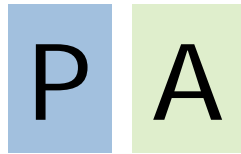
- Entwickeln einer Aktivitätenliste
 - Zu jedem Teilprodukt laut WBS gehören eine oder mehrere Aufgaben (→ Aktivitäten im Prozessmodell ✓)
- (Zeit)Aufwandsschätzung für die Aktivitäten
 - Wie hoch ist der Aufwand (z.B. Personentage)?
 - Wie lange dauert die Erledigung (Kalendertage)?
- Aufstellung eines Zeit- und Arbeitsplans (schedule)
 - Wie hängen die Aufgaben von einander ab?
 - Wer erledigt von wann bis wann welche Aktivität?
- Zeitplanüberwachung (schedule control)
 - Wird der Plan eingehalten?
 - Kontinuierliche Fortschreibung des Zeitplans



- Kostenschätzung
 - Budgetaufstellung
 - Kostenüberwachung
-
- Bei SW-Projekten sind außer den Personalkosten meistens nur wenige Posten relevant (SW-Lizenzen, HW, Räume etc.)
 - Dadurch hängen die Kosten so eng an den Zeitaufwänden, dass diese Aufgabe meistens kaum separat gelöst werden muss



- Qualitätsplanung (quality planning)
 - Aufstellen von Qualitätsanforderungen
(bei SW: → Anforderungsbest. der nichtfunktionalen Anf. ✓)
 - Planen, wie man sie erfüllt
(bei SW: → Architektur ✓ und → konstruktive QS ✓)
- Qualitätssicherung (quality assurance)
 - bei SW: → analytische Qualitätssicherung ✓
- Qualitätssteuerung (quality control)
 - wenn die Qualität nicht stimmt, geeignete Maßnahmen ergreifen
 - z.B. korrigieren, Teil wegwerfen und neu bauen, Aufgabe an andere Person übergeben, Entwicklungsprozess ändern, etc.



- Personalplanung
 - Definition von Rollen, Verantwortlichkeiten, Weisungsbefugnissen
 - Aufstellung eines Personalzeitplans
 - z.B. Analyst_innen überwiegend am Anfang des Projekts, Programmierer_innen jedoch nicht von Anfang an u.s.w.
- Team einwerben, Teamformung
 - Teilnahme an einem Projekt sollte freiwillig sein
 - Planendes PM: Team oft spezifisch für Projekt
 - Agiles PM: Team oft dauerhaft
- Teamführung
 - Leistung der Mitglieder verfolgen, Feedback geben, Konflikte auflösen, Arbeit bei Änderungen neu koordinieren




- Kommunikationsplanung
 - Welche Beteiligten brauchen wann welche Information?
- Informationsverteilung
 - Den Betroffenen relevante Information stets zügig zukommen lassen
- Fortschrittsberichte
 - Statusberichte und Planfortschreibung regelmäßig erarbeiten und allen Betroffenen zuleiten
- Interaktion mit Beteiligten
 - Besprechungen und Schriftverkehr mit allen Beteiligten (z.B. Projektteam, Auftraggeber, Anwender, Interessierte), um deren Bedürfnisse zu erfüllen und Angelegenheiten aller Art zu klären



- Risiken identifizieren
 - Welche Ereignisse könnten die plangerechte Durchführung bedrohen?
- Risikoeinschätzung
 - Risikogrößen abschätzen, Risikobehandlung priorisieren
- Vorbeugung und Gegenmaßnahmen planen
 - Was wird getan, um welchem Risiko vorzubeugen?
 - Was wird getan, wenn welches Risiko trotzdem eintritt?
- Risikomanagement-Überwachung
 - Kontinuierlich nach neuen oder veränderten Risiken Ausschau halten
 - Fortführung und Wirkung eingeleiteter Vorbeugungen und Gegenmaßnahmen überwachen. Ggf. korrigierend eingreifen



- Planung und Durchführung der Beschaffung aller Dinge, das Projekt braucht, aber nicht selbst herstellt 
- Betrifft bei SW-Projekten z.B. Möbel, Hardware, SW-Lizenzen
 - Ist meist nicht sehr kompliziert



- Voilà, das waren die 9 Aufgabenfelder.

Wir betrachten im Folgenden einige Fragestellungen von:

P A

- **Integrierende Aufgaben (integration mgmt)**
- Umfangsmanagement (scope mgmt) ✓
- **Zeitmanagement (time mgmt)**
- Kostenmanagement (cost mgmt)
- Qualitätsmanagement (quality mgmt) ✓
- **Personalmanagement (human resource mgmt)**
- **Kommunikationsmgmt. (communication mgmt)**
- **Risikomanagement (risk mgmt)**
- Beschaffungsmgmt. (procurement mgmt)

Wann?: Heute plus nächste 3 Doppelstunden

- Zeitplanung
 - Schätzung des Gesamtaufwands ("cost estimation") heute
 - Aufstellen eines Zeit- und Arbeitsplans nächste Std.
- Risikomanagement nächste Std.
 - Risikoermittlung und -einschätzung
 - Riskovorbeugung und -behandlung
- Personal- u. Kommunik.mgmt. übernächste Std.
 - Gruppen und Teams
 - Psychologische Effekte
- Integrierende Aufgaben überübernächste Std.
 - Projektplan
 - Projektleitung und -überwachung

- Zeitplanung
 - **Schätzung des Gesamtaufwands**
 - Aufstellen eines Zeit- und Arbeitsplans
- Risikomanagement
 - Risikoermittlung und -einschätzung
 - Riskovorbeugung und -behandlung
- Personalmanagement
 - Gruppen und Teams
 - Psychologische Effekte
- Integrierende Aufgaben
 - Projektplan
 - Projektleitung und -überwachung

- Zuverlässige (und halbwegs genaue) Aufwandsschätzungen sind eine wichtige Grundlage für wirtschaftlich erfolgreiche SW-Projekte
 - am stärksten für Auftragnehmer bei Festpreisprojekten
 - ➔ machen bei Unterschätzung Verluste, bekommen bei Überschätzung kaum Aufträge
- Neben Aufwands- sind auch Zeitschätzungen wichtig
 - weil der Nutzen oft stark abnimmt, wenn man zu spät an den Markt oder in den Einsatz kommt
- Zu schlechte (dann meist zu optimistische) Schätzungen führen zu
 - Stress und Frustration bei den Projektmitgliedern
 - langfristig Ausbrennen
 - Wirtschaftlichen Verlusten, auch bei den Auftraggebern

- Leider ist das Schätzen von Softwareprojekten aufgrund von deren hoher Komplexität sehr schwierig
 - außerdem gibt es Störfaktoren, insbesondere unbekannte oder schwankende Projektanforderungen
- Wichtige Fragen beim Schätzen
 - Sind die Anforderungen schon bekannt?
 - Alle? Genau genug? Auch die nichtfunktionalen?
 - Ist die Architektur schon verstanden? (Standardarchitektur?)
 - Ist die Technologie vorgegeben? Ein passendes Team vorhanden?
 - Gibt es sonstige schwierige/ungewöhnliche Randbedingungen?
 - z.B. verteilte Entwicklung, kein Benutzerkontakt, junges Team etc.
- Faustregeln:
 - Nur normales Vorgehen lässt sich solide schätzen, radikales nicht
 - Hochinnovative Projekte sind also schwierig
 - Projekte mit fließenden Anforderungen kann man nicht verlässlich schätzen



- Schätzungen werden fast immer mit den folgenden grundlegenden Methoden hergeleitet
 - meist durch eine Kombination mehrerer dieser Methoden
- Ferner gibt es ausgefeilte benannte Schätzmethoden,
 - firmenspezifisch oder aus der allgemeinen Literatur,
 - die immer auch Kombinationen dieser Basismethoden sind,
 - ergänzt z.B. um konkrete Zahlenwerte für Schätzformeln

- Basismethoden:
 - Schätzen durch Vergleich
 - Schätzen durch Zerlegung
 - der Anforderungen
 - des Entwurfs
 - Expertenschätzung
 - Kombinationsschätzung
 - Schätzen mit Korrekturfaktoren
 - Schätzen mit Stellvertretergröße
- Wir stellen die Basismethoden in Form von Mustern vor:
 - Problem/Kontext
 - Vorgehensidee
 - Vorteile, Nachteile
 - evtl. Varianten, Anmerkungen

- Problem/Kontext:
 - Wenn keine bessere Schätzmethode verfügbar ist
 - aber Erfahrungen mit Entwicklung ähnlicher Dinge
 - und deren Aufwände noch bekannt sind
- Vorgehensidee:
 - Wähle aus dem Erfahrungsschatz das ähnlichste Ding aus und benutze dessen Aufwand als Schätzung
- Vorteile: Einfach, schnell
- Nachteile: evtl. zu wenig Erfahrung; Ähnlichkeitsmaß unklar
- Variante:
 - Wähle mehrere Ähnlichste und mittele die Schätzung
- Anmerkung:
 - Alle anderen Schätzverfahren basieren letzten Endes auf Vergleich (explizit oder intuitiv)



Schätzen durch Zerlegung (Anforderungen)

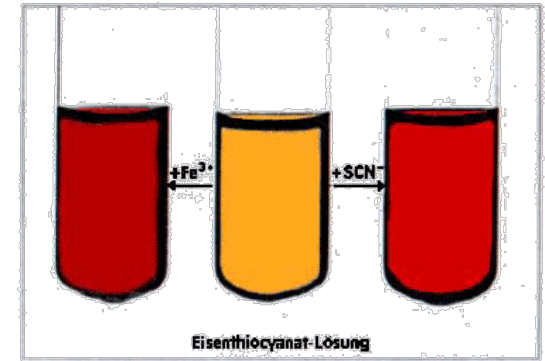
- Problem/Kontext:
 - Wenn Anforderungen wenig voneinander abhängen
 - d.h. das System nur einen kleinen Kern aufweist
- Vorgehensidee:
 - Schätze Aufwand pro Anforderung einzeln; summiere
- Vorteile:
 - Starke Reduktion der Komplexität
 - Konzentration aufs Wesentliche (Benutzeranforderungen)
- Nachteile:
 - Nur in wenigen Domänen sinnvoll
 - Oft ist der nicht in Anforderungen sichtbare SW-Kern aufwändig
 - Aufwandsmessungen liegen selten in dieser Form vor
- Anmerkung:
 - Schätzverfahren "function points" (Funktionspunkte) funktioniert so und ist für einfache Informationssysteme sehr bewährt



- Problem/Kontext:
 - Wenn die Architektur des Systems bereits erkennbar ist
- Vorgehensidee:
 - Schätze den Aufwand für die Subsysteme und addiere
- Vorteile:
 - Im Prinzip ist eine sehr genaue Zerlegung und Schätzung möglich
- Nachteile:
 - Irrtümer über Architektur möglich (insbes. Teile übersehen)
 - Vernachlässigt Aufwand für Integration und nichtfunktionale Anforderungen
- Anmerkung:
 - Zumindest grobe Zerlegung wird in der Praxis fast immer eingesetzt
 - Oft läuft ein großer Teil der Zerlegung auf bloße Zählung hinaus
 - z.B. Anzahl "Dialoge" (Bildschirmmasken)

- Problem/Kontext:
 - Wenn es Erfahrungen und Aufwandsdaten nicht schriftlich, wohl aber im Kopf von Expert_innen gibt (meist nur implizit)
- Vorgehensidee:
 - Bitte jede Expert_in separat um eine Schätzung
 - Lasse die Expert_innen Ihre Schätzungen und Begründungen diskutieren und modifizieren
 - Resultat: Konsens über einen geschätzten Aufwandsbereich
- Vorteile:
 - Sehr breitbandig, bezieht enorm viele Faktoren ein
 - Unsicherheit der Schätzung wird ggf. klar sichtbar
- Nachteile:
 - Kann bei "Groupthink" zu dramatisch falschen Ergebnissen führen, die scheinbar vertrauenswürdig sind
- Anmerkung:
 - Agile Methode: "Planning Poker" für einzelne Anforderungen

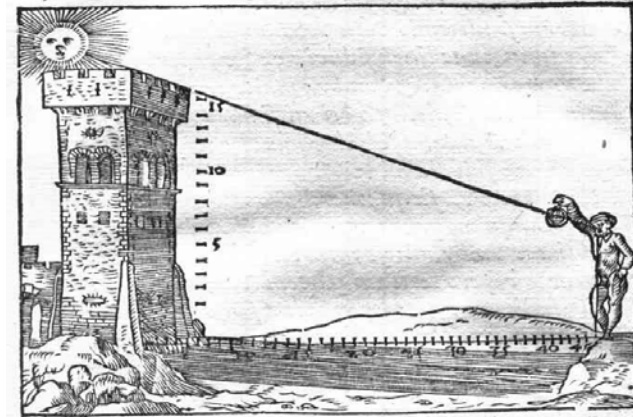
- Problem/Kontext:
 - Wenn mehrere Schätzungen verfügbar sind, deren Qualität aber unklar ist
- Vorgehensidee:
 - Kombiniere die Schätzungen zu einer (durch Bereichsbildung, Mittelung, Zurückweisung von Ausreißern etc.)
- Vorteile:
 - Erhöht die Robustheit der Schätzung
- Nachteile:
 - Gibt es systematische Fehler in vielen der Schätzungen, dann führt das Kombinieren in die Irre
- Anmerkung:
 - Delphi-Methode wendet dies oft mit an



- Problem/Kontext:
 - Wenn ein ähnliches Ding zum Vergleich verfügbar ist
 - und es zum zu schätzenden Ding erkennbare Unterschiede gibt
- Vorgehensidee:
 - Benutze den Aufwand des ähnlichen Dings und korrigiere ihn für jeden Unterschied um einen prozentualen Faktor herauf/herunter
 - Die einzelnen Faktoren werden wiederum geschätzt oder aus Erfahrungen entnommen
- Vorteile:
 - Recht flexibel
- Nachteile:
 - Bei zu vielen Korrekturen wird das Ergebnis dubios
 - Bei zu wenig Erfahrung über einzelnen Korrekturfaktor ebenfalls
- Anmerkung:
 - Boehm's Verfahren CoCoMo basiert stark auf Korrekturfaktoren



- Problem/Kontext:
 - Wir besitzen Aufwandsdaten nur über andere Erfahrungsgrößen als die, die wir schätzen wollen
 - aber diese Erfahrungsgröße (oder etwas Verwandtes) lässt sich schätzen
- Vorgehensidee:
 - Schätze schätzbare Ersatzgröße
 - und rechne dann um
- Vorteile:
 - Evtl. ist Stellvertretergröße anschaulicher und besser zu schätzen
- Nachteile:
 - Evtl. wird Kontextabhängigkeit übersehen



Häh?

Beispiel: Funktionspunktverfahren (FP)

- Von IFPUG (International Function Point User Group) genormt
- Geeignet für einfache Informationssysteme
 - d.h. Systeme, die Daten nur interaktiv verwalten, aber nicht kompliziert verarbeiten
- **Grundidee:**
 - Zähle und klassifiziere (als einfach, mittel, komplex) wenige Arten von Anforderungen
 - vergebe dafür jeweils *Funktionspunkte* (FP), summiere diese und
 - bestimme Aufwand daraus per empirischem Umrechnungsfaktor.
 - FP ist also ein **Stellvertreterverfahren** (→FPs) kombiniert mit **Vergleichsverfahren** (→Umrechnungsfaktor)
- Außerdem kann man noch Projektumstände berücksichtigen, um die Schätzung anzupassen
 - Hinzu kommt also ein **Korrekturfaktorverfahren**



FP
geht
so:

FP: Arten von Anforderungen

- Eingabe (input)
 - Eine Eingabemaske, über die Datensätze eingegeben werden können (z.B. Kundenstammdaten: Name, Adresse, etc.)
 - einfach: 3 FP, mittel: 4 FP, komplex: 6 FP
- Ausgabe (output)
 - Eine Datenpräsentationsdarstellung ("Report")
 - einfach: 4 FP, mittel: 5 FP, komplex: 7 FP
- Abfrage (query)
 - Eingabe eines Suchkriteriums plus Anzeige des Gefundenen
 - einfach: 3 FP, mittel: 4 FP, komplex: 6 FP
- Datenbestand (file)
 - Intern verwalteter Datenbestand (z.B. DB-Tabelle)
 - einfach: 7 FP, mittel: 10 FP, komplex: 15 FP
- Schnittstelle (interface)
 - Datenformat zur Anbindung an anderes System
 - einfach: 5 FP, mittel: 7 FP, komplex: 10 FP

Dafür gibt's
konkrete
Kriterien

FP: Korrekturfaktoren

- Nun werden Korrekturwerte gemäß dem Schwierigkeitsgrad diverser globaler Anforderungen gezählt:
 - Verflechtung mit anderen Systemen (0–5 Korrekturpunkte, 0 für "keine", 5 für "sehr stark");
 - verteiltes System (0–5);
 - hohe Transaktionsrate (0–5);
 - komplexe Rechenoperationen (0–10), Steuerung (0–5), Ausnahmeregelungen (0–10), Verarbeitungslogik (0–5)
 - hohe Wiederverwendbarkeit verlangt (0–5)
 - viele Datenkonvertierungen nötig (0–5)
 - hohe Anpassbarkeit verlangt (0–5)
- Das ergibt zusammen 0 bis 60 Korrekturpunkte p
 - werden als $K = 0,7 + p/100$ in einen Korrekturfaktor im Bereich 0,7...1,3 verwandelt

FP: Gesamtschätzung

- Der Umrechnungsfaktor basiert auf historischen Daten (aus früheren Projekten)
 - z.B. "In unserer Firma haben solche Teams für solche Projekte einen Aufwand **T** von 2,3 Personentagen pro Funktionspunkt gehabt."
- Beispiel: Gesamtschätzung
 - Angenommen, die gewichteten Anforderungen ("rohe" Funktionspunkte) ergeben $\mathbf{FP}_{roh} = 130$
 - und angenommen, 40 Korrekturpunkte ergeben Korrekturfaktor $\mathbf{K} = 0,7 + 40/100 = 1,1$
 - Das ergibt justierte (adjusted) Funktionspunkte $\mathbf{FP} = \mathbf{FP}_{roh} * \mathbf{K} = 130 * 1,1 = 143$
 - Das ergibt als Gesamtaufwand $\mathbf{A} = \mathbf{FP} * \mathbf{T} = 143 * 2,3 = 329$ Personentage (PT)
 - Ein Personenmonat (PM) hat ca. 18 Arbeitstage (~220 Tg/Jahr)
 - Projektumfang: $329/18 \approx 18$ PM. Also 6 Monate für 3er-Team

FP: Eignung, Historische Daten

Wer noch keine historischen Daten hat, kann

- Vergleichsdaten aus der Literatur entnehmen oder
- wo er den Gesamtaufwand alter Projekte kennt:
nachträglich FP-Zählungen für dieser Projekte machen

- Die FP-Methode ist in jedem Fall schlecht geeignet für folgende Fälle:
 - Projekte, deren Schwierigkeit woanders steckt als in FP-Eingaben, FP-Ausgaben, FP-Abfragen, FP-Datenbeständen und FP-Schnittstellen
 - z.B. alle eingebetteten Systeme, Systemsoftware, Realzeit-SW etc.
 - Projekte, für die die funktionalen Anforderungen erst grob ermittelt wurden
 - (da tun sich andere Schätzverfahren allerdings auch schwer...)

- Projektschätzungen verwenden fast immer eine Kombination von mehreren Verfahren
 - Wie die Kombination genau aussieht, ist von Firma zu Firma verschieden
 - Hängt auch stark von der Domäne ab
- Wenn Sie Lehrbücher lesen:
 - Am häufigsten beschrieben: Funktionspunkte und CoCoMo
 - Ich habe noch nie gehört, dass jemand CoCoMo wirklich einsetzt
- In der Praxis:
 - Funktionspunkte werden in der Praxis in entsprechenden Bereichen durchaus verwendet
 - oft in modifizierter Form und meist ergänzt durch andere Verfahren
 - Zerlegung ist der wichtigste "Trick"
 - Rat: Machen Sie Schätzungen auf so viele Arten Sie nur können
 - und dann eine Kombinationsschätzung

P A

Agile Projekte schätzen
nur in einer Iteration:
Viel einfacher!

P

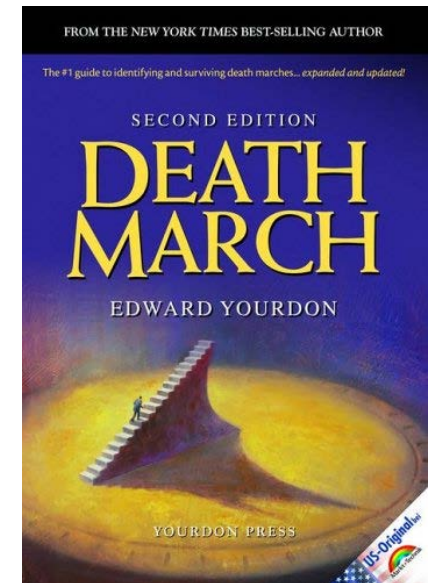
1. Oft sind Schätzungen gefragt, noch bevor man auch nur halbwegs den Projektinhalt versteht
 - und dennoch werden die Schätzergebnisse anschließend verbindlich
2. Viele Firmen sammeln Aufwandsdaten nicht explizit
 - Sondern nur ungenau in den Köpfen der Leute
3. Oft wird das Ergebnis der Schätzung nicht akzeptiert
 - sondern es gibt politischen Druck, die Schätzung zu verringern
 - aber die künstlich verminderte Schätzung wird trotzdem zur verbindlichen Vorgabe an das Projektteam.
 - Dramatischste Ausprägung: "Todesmarsch-Projekte"
 - Schätzung liegt 50% oder mehr unterhalb "vernünftiger" Werte
4. Pseudo-Agile Projekte, die zu schätzen versuchen

Danke!

Es folgen Zusatzfolien zum Thema
Todesmarsch-Projekte

- Definition: Ein Projekt, dessen zentrale Ressourcen-Parameter um mindestens Faktor 2 unter dem Normalen liegen
 - Halbe Entwicklungszeit oder halbes Personal/Budget oder doppelt so scharfe Leistungsanforderungen wie vernünftigerweise anzunehmen wäre.
- Sind heute seltener als früher
 - vermutlich, weil Softwareleute so knapp sind.
 - Bei agilen Methoden sowieso unwahrscheinlich.

Quelle: Edward Yourdon: Death March
2nd ed., Yourdon Press 2004



Warum kommen sie überhaupt zustande?

- Politik
 - Machtkämpfe verschiedener Art (ich bin besser; allgemeiner Prestigegewinn; letzte Chance; u.a.)
- Naive oder rücksichtslose Versprechungen durch Vertrieb oder oberes Management
 - "Wir haben den Auftrag! Leider mussten wir unseren Preis halbieren."
- Jugendlicher Optimismus
 - oft technik-getrieben: "Das schaffen wir [mit XYZ] an einem Wochenende"
- Startup-Mentalität
 - Adrenalinreicher Zweckoptimismus bei gleichzeitigem Personalmangel, instabiler Organisation, unerfahrenem Management und kaum verstandenem Markt

- Elitetruppen-Mentalität
 - "Echte Programmierer brauchen keinen Schlaf"
- Intensiver Wettbewerb aufgrund der Globalisierung
 - "Eine indische Firma hat für die Hälfte unseres Preises angeboten"
- Intensiver Wettbewerb aufgrund neuer Technologie
 - Nur der erste Anbieter, der damit auf den Markt kommt, kann noch hohe Gewinne erzielen
- Hoher Zeitdruck durch geänderte Gesetzeslage
 - Nur 3 Monate Zeit, bis 400 Seiten neuer Steuergesetze in Kraft treten
- Unerwartete oder ungeplante Krisen
 - "Oh, Mist."

Durch agile Prozesse viel seltener geworden, aber im kurzen Zeitmaßstab immer noch gängig.

Gute Fragen für die Verhandlungen über Termine/Ressourcen:

- "Jeder möchte gern alles gut und schnell und billig. Und jeder weiß, dass man nur 2 dieser 3 Eigenschaften bekommen kann. Welche sind hier am wichtigsten?"
- "Welches sind die kritischen 20% der Funktionalität, die 80% des Nutzens stiften?"
 - Und dann weiter: "Wenn wir mit denen erst eine Woche später fertig werden, gehen wir dann pleite?"
- Ziel ist hier
 - akzeptable Abwägungen zu finden
 - und harte Todesmarsch-Verursacher unvernünftig aussehen zu lassen