



# Management großer Softwareprojekte

Prof. Dr. Holger Schlingloff

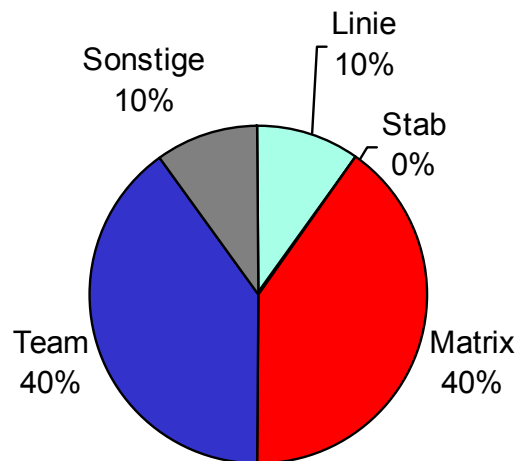
Humboldt-Universität zu Berlin,  
Institut für Informatik

Fraunhofer Institut für Rechnerarchitektur  
und Softwaretechnik FIRST

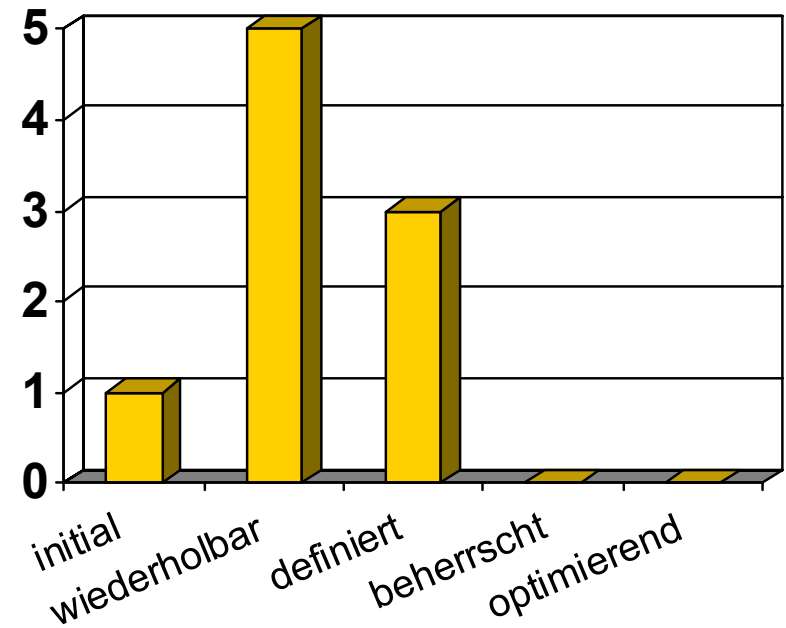
# Hausaufgabe von vorletzter Woche

- Beschreiben Sie Organisations- und Ablaufstruktur eines SW-Projektes, bei dem Sie mitgearbeitet haben!

## Aufbauorganisation



## Ablauforganisation



# Vorbereitendes Experiment

---

Für die folgende Aufgabe haben Sie 10 Sekunden Zeit:

- Schätzen Sie, wie viele Geldmünzen sich in diesem Raum befinden!

Notieren Sie bitte Ihr Ergebnis als (a)

Für die folgende Aufgabe haben Sie 60 Sekunden Zeit:

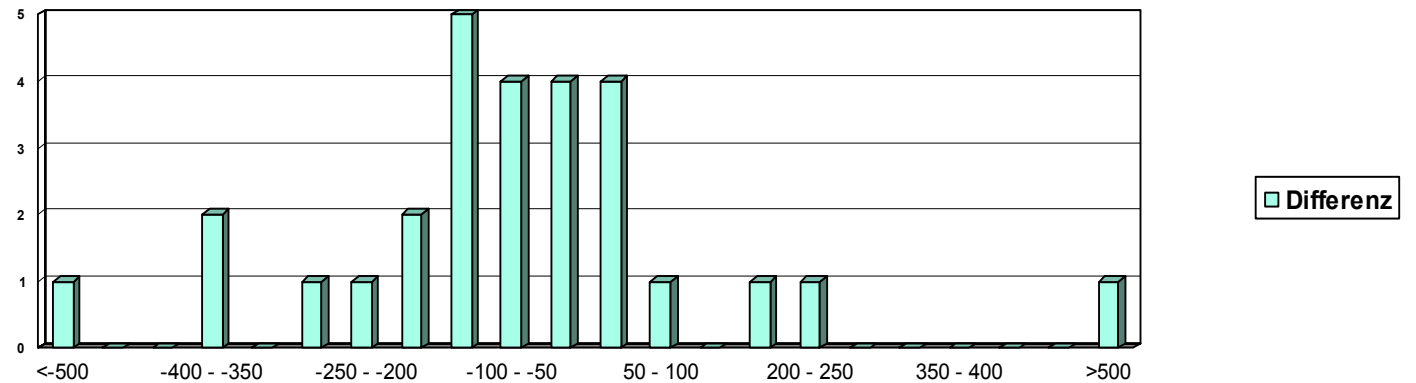
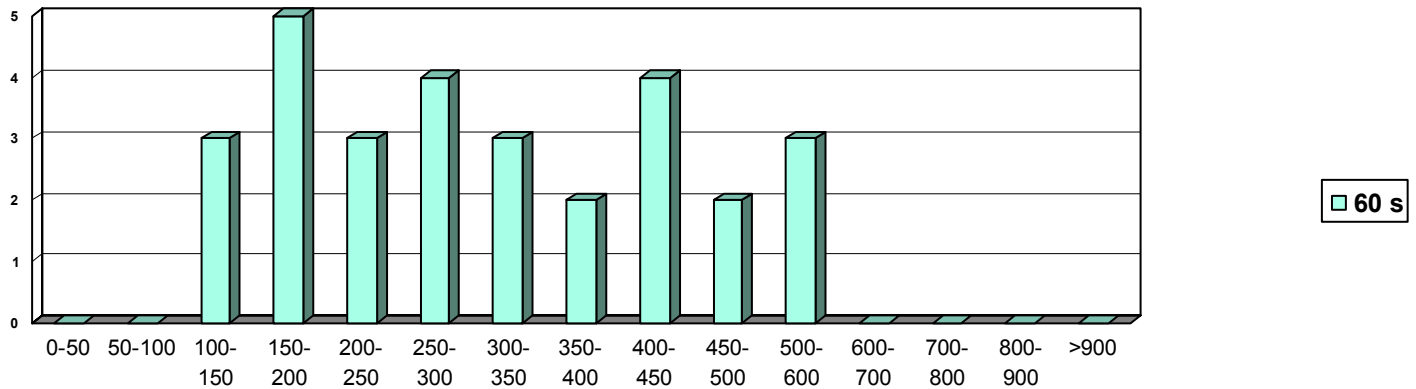
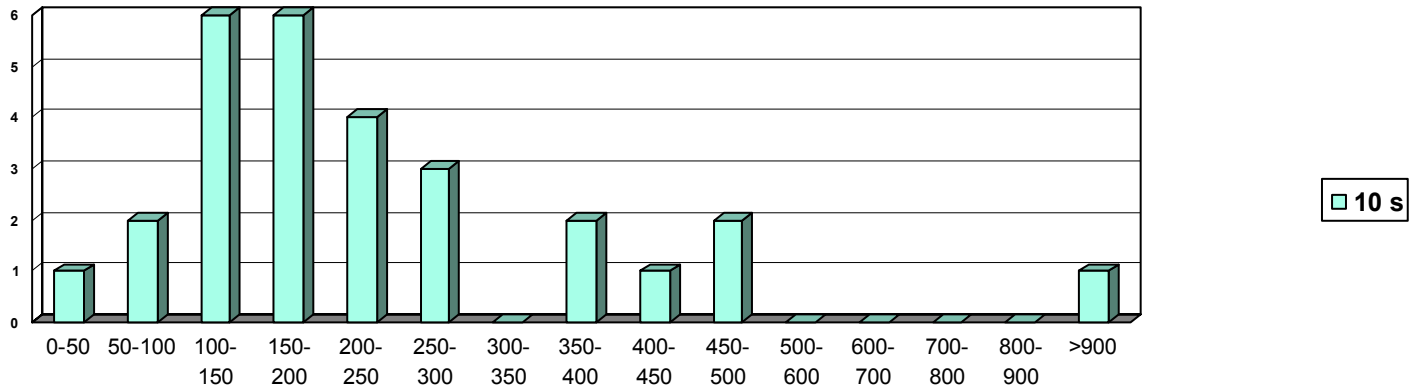
- Schätzen Sie, wie viele Geldmünzen sich in diesem Raum befinden!

Notieren Sie bitte Ihr Ergebnis als (b)

Bilden Sie die Differenz  $(c) = (a) - (b)$



# Ergebnisse



# Ergebnisse

---

- Bei höherem Schätzaufwand wird das Ergebnis genauer (genau: 427)
- Der Mittelwert mehrerer Schätzungen ist oft genauer als einzelne Schätzungen
- Hemmschwelle gegen nochmalige Schätzung des selben Gegenstands

# 4. Aufwandsschätzung

---

Warum ist es notwendig, zu schätzen?

- Grundlage für Angebot und Auftrag
- Grundlage für Zeit- und Kostenplanung
- Entscheidungsgrundlage für Vorgehen

Problem: Wie kommt man zu verlässlichen Werten?

- frühzeitig (vor Beginn!) und akkurat (Limits!)
- Schätzung zu hoch → kein Projekt
- Schätzung zu gering → finanzieller Engpass (Ruin)



# Problematik beim Schätzen

---

- **extrem viele Parameter**

Personal, Projektkomplexität, Dokumentation, Methoden- und Werkzeugeinsatz, Organisationsform, Motivation, ...

- **prinzipielle Ungenauigkeit**

Schätzung beeinflusst den Prozess (Planung und Durchführung)

- **Ergebnisse nicht verifizierbar**

keine separat messbaren Größen: schlechtes Management kann jede noch so gute Schätzung ad absurdum führen

- **nur bedingt übertragbar**

Einmaligkeit der Projektbedingungen



# Henne/Ei Problem des Schätzens

---

- Schätzung beruht auf einer festen Anforderungsdefinition
- Durch das Schätzergebnis werden die Anforderungen beeinflusst
- detaillierte Anforderungsanalyse kann bereits hohe Vorlaufkosten verursachen
- in der Praxis oft *Kosten* statt *Anforderungen* fix





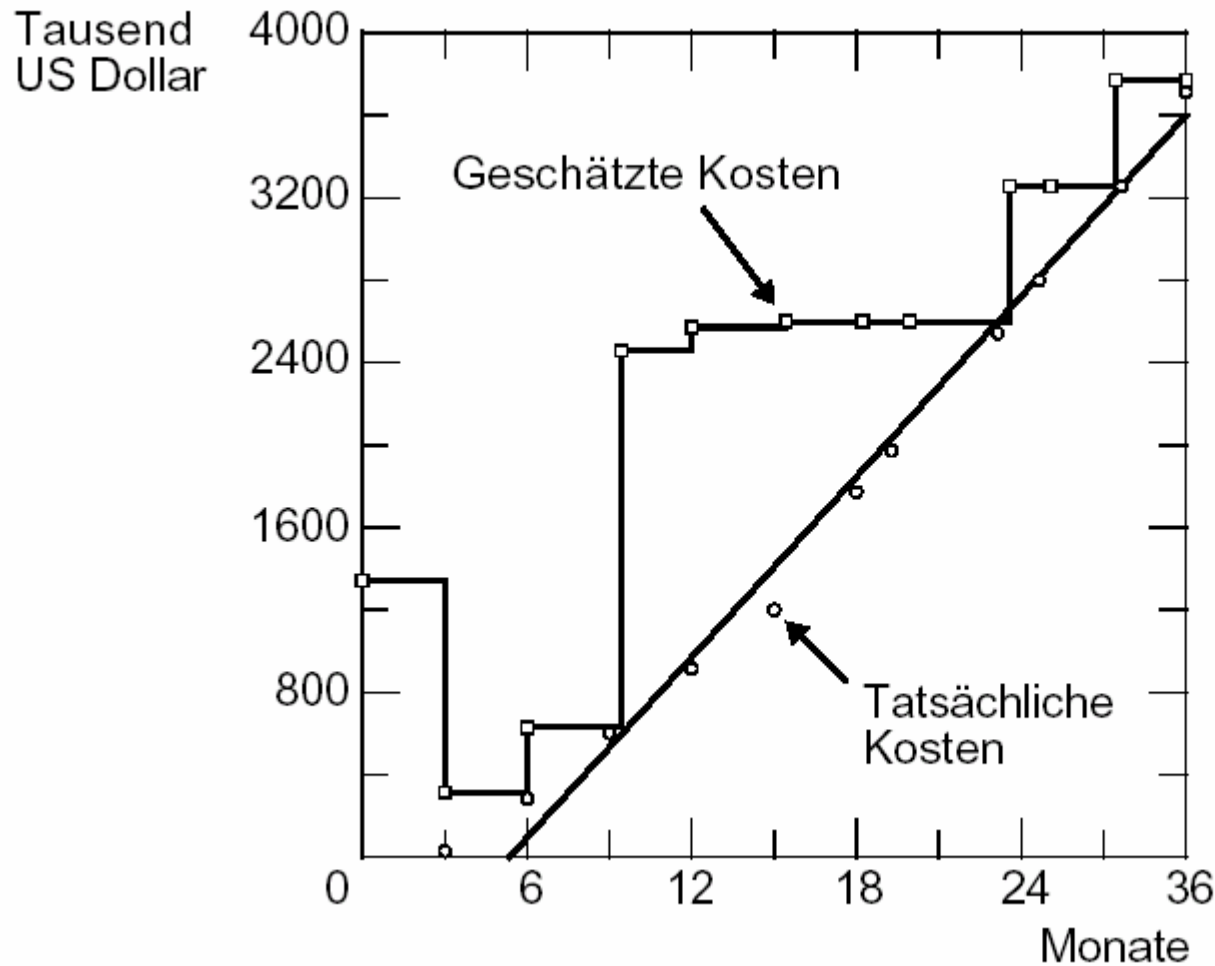
# Dilemma des PM beim Schätzen

---

- Projektbudget wird überschritten
  - Firmenleitung hat Projekt geschätzt  
→ „schlechtes Projektmanagement“
  - PM hat geschätzt → „schlechte Schätzung“
- Projektbudget wird eingehalten
  - Firmenleitung hat Projekt geschätzt  
→ „gute Schätzung, PM wie erwartet“
  - PM hat geschätzt → „zu großzügig geschätzt“



# Negativbeispiel: „Nicht-Schätzung“



nach Boehm (1981)

# Softwarebeispiel

---

- Schätzen Sie bitte die Zeit  $d$  (in Minuten), die ein durchschnittlicher Programmierer braucht, um folgendes Programm zu schreiben:  
**Aus einer Textdatei sollen alle Dubletten (d.h., doppelt vorkommende Wörter) gelöscht werden**
- Schätzen Sie bitte die Zeit  $e$  (in Minuten), die Sie selber für dieses Programm brauchen!
- Schreiben Sie bitte das Programm jetzt (Zeit läuft!)
- Korrigieren Sie ihre ursprüngliche Schätzung!



# Softwarebeispiel - Fortsetzung

---

- Wie lange brauchen Sie jetzt noch, um das einzutippen, zu debuggen und zu testen? Schätzen Sie einen Faktor  $f$
- Wie stabil ist Ihr Programm (z.B. Parallelzugriff)? Wo gibt es Effizienzverbesserungen?
- Korrigieren Sie Ihre vorherige Schätzung
  - unter der Annahme, dass das Programm nur ein paar Mal gebraucht wird, um einige Dateien manuell aufzubessern
  - unter der Annahme, dass das Programm den Kern eines großen Informationssystems bildet, und ständig mit sehr großen Dateien aufgerufen wird



# Ergebnisse

Durchschnittszeit  $d$ , eigene Zeit  $e$ , tatsächliche Zeit  $t$

- $d < e : 14$
- $d = e : 4$
- $d > e : 12$
- $t * f < e : 2$
- $t * f = e : 9$
- $t * f > e : 10$

Faktor  $f$ , Qualität  $f_{-}$  und  $f^{-}$ :

- $f : 4, 3.5, 3, 1.3$
- $f_{-} : 1, 2.5, 0.7, 1.5,$
- $f^{-} : 10, 12, 50, 20, 6,$



# Folgerungen

---

- Gefahr der Selbstüberschätzung (bzw. Überschätzung der eigenen Gruppe)
- Unterspezifizierte Aufgabenstellung ergibt ungenaue Schätzung; Qualitätsfaktor nicht vernachlässigbar
- ein Programm „zum Laufen zu bringen“ dauert etwa dreimal so lange, wie das bloße Aufschreiben



# Grundregeln beim Schätzen

---

- 1. Grundregel: Müll rein – Müll raus  
→ präzise Anforderungen!
- 2. Grundregel: Je ferner die Zukunft, desto schwieriger sind die Schätzungen  
→ kontinuierliche Neuschätzungen!
- 3. Grundregel: große Blöcke sind schwieriger zu schätzen als kleine, abstrakte schwieriger als konkrete  
→ Granularität, Konkretisierung
- 4. Grundregel: Schätzungen sind keine Weissagungen, d. h. keine *verbindlichen* Voraussagen (selbsterfüllende Prophezeiung)  
→ Toleranzspielraum!

