

Martin Glinz Harald Gall

Software Engineering

Wintersemester 2005/06

Kapitel 17

Mehrfachverwendung



Universität Zürich
Institut für Informatik

Motivation

Die Masse macht es.

Ziel: **Kosten senken** für Software

⇒ Entwicklungs**produktivität steigern** – Möglichkeiten **begrenzt**

⇒ **Stückzahlen erhöhen**

- Das Geheimnis der billigen Hardware
- Bei Software durch Mehrfachverwendung

○ Mehrfachverwendung möglich durch

- **Große Produktserien** mit identischer Software
- **Wiederverwendung** der gleichen Software in mehreren Produkten
- **Beschaffung**

Wiederverwendung

- Wiederverwendbare Software muss erst einmal **geschaffen** werden
 - ⇒ **Komponenten** bilden, entwickeln und dokumentieren
- Entwicklung wiederverwendbare Software ist **teurer** als die von Einzweck-Software
 - ⇒ **Explizite Anreize** für Entwicklung wiederverwendbarer Software
- Wiederverwendbares wird **nicht gefunden**
 - ⇒ **Kataloge**
- Wer **pfligt** wiederverwendete Software
 - ⇒ **Pflegeverträge**
- **Förderung** der Wiederverwendung
 - **Handel** mit Komponenten
 - **Informationsstelle** für Wiederverwendung

Beschaffung

- **Wo immer möglich**, denn:
fast immer **billiger** als Eigenentwicklung (Stückzahlen!)
- Aber
 - **Anpassungs- und Parametrierungskosten** berücksichtigen
 - **Wirtschaftlichkeit** im Einzelfall **prüfen**

Beschaffungsprozess ist **eingebettet** in **Entwicklungsprozess**

- Spezifikation, Test und Installation auch bei Beschaffung notwendig
- Hauptaktivitäten für Beschaffung in der Phase der Konzipierung der Lösung

Mehrfachverwendung im Entwicklungsprojekt

Die **Beteiligten** müssen **mitmachen**:

- **Nicht** alles **selbst** machen wollen
- Nicht das **Perfekte** (und Teure), sondern das **Gebrauchstaugliche** schaffen
- Mitarbeiterinnen und Mitarbeiter **nicht** nach der produzierten Software-**Menge** beurteilen
- **Suche** nach **Beschaffbarem/Wiederverwendbarem** zum **selbstverständlichen Bestandteil** jedes Entwicklungsprojekts machen

Literatur

Siehe Literaturverweise im Kapitel 3 des Skripts.

Im Skript [M. Glinz (2005). *Software Engineering*. Vorlesungsskript, Universität Zürich] lesen Sie Kapitel 3.4.

Im Begleittext zur Vorlesung [S.L. Pfleeger, J. Atlee (2006). *Software Engineering: Theory and Practice*, 3rd edition. Upper Saddle River, N.J.: Pearson Education International] lesen Sie in Kapitel 12.4 die Seiten 565-574, in Kapitel 13.2 die Seiten 612-613.