



Universität
Zürich^{UZH}

Institut für Informatik

Martin Glinz Harald Gall

Software Engineering

Kapitel 18

Produktivitätsfaktoren

18.1 Überblick

18.2 Mehrfachverwendung



Was beeinflusst die Software-Produktivität?

- **Menschen**
 - Ein dominierender Produktivitätsfaktor
 - Ausbildung
 - Motivation
 - Förderung
 - Folienkapitel 19
- **Technische Hilfsmittel**: Werkzeuge (→ Folienkapitel 22)
- **Projektmanagement** (→ Folienkapitel 14 und 19)
 - Gut geplanter Personaleinsatz
 - Vermeidung von Kontextwechsel-Verlusten
 - Vermeidung von Koordinations- und Kommunikationsverlusten
- **Mehrfachverwendung**

18.1 Überblick

18.2 Mehrfachverwendung



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
 - Nutzung von **Software-Diensten** (services)
 - **Beschaffung** von Software (Systeme oder Komponenten)
 - Nutzung von **quelloffener (open source)** Software

Wiederverwendung

- Wiederverwendbare Software muss erst einmal **geschaffen** werden
 - ⇒ **Komponenten/Dienste** 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 **pfl egt** wiederverwendete Software
 - ⇒ **Pflegeverträge**
- **Förderung** der Wiederverwendung
 - **Bereitstellung / Nutzung** von **Komponenten** und **Diensten**
 - **Handel** mit **Komponenten** und **Diensten**
 - **Informationsstelle** für Wiederverwendung

Nutzung von Diensten (services)

- **Finden** geeigneter Dienste
 - mit der gewünschten Funktionalität
 - mit den notwendigen Leistungs- und Qualitätsmerkmalen
- **Einbettung** in eigene Software
 - Semantik der Schnittstellen?
 - Erprobung/Test erforderlich
- **Verfügbarkeit** sichern
 - Dienstverfügbarkeit
 - Netzverfügbarkeit
- **Kosten**: Nutzungskosten abwägen gegen Beschaffungskosten für Komponenten bzw. Eigenentwicklung
- **Nutzungsrechte**: Verträge erforderlich

Beschaffung / Nutzung quelloffener Software

- Fast immer **billiger** als Eigenentwicklung (Stückzahlen!)
- Aber
 - **Anpassungs- und Parametrierungskosten** berücksichtigen
 - Vorab zu klären
 - **Qualität**
 - Wer übernimmt die **Pflege**
 - **Lizenzfragen**
 - **Wirtschaftlichkeit** im Einzelfall **prüfen**

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

- Spezifikation, Test und Installation auch bei Beschaffung notwendig
- Hauptaktivitäten für Beschaffung während 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 **verwendbarer Software von Dritten** 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 (2010). *Software Engineering: Theory and Practice*, 4th edition. Upper Saddle River, N.J.: Pearson Education International] lesen Sie in Kapitel 12.4 den Abschnitt über Software Reusability (Seiten 627-636), in Kapitel 13.2 die Seiten 674-675.