Teil III: Software-Architekturentwurf

16 Architekturentwurf – Einführung und Überblick

16.1 Software entwerfen – Warum?

- ☆ Beim Arbeiten im Kleinen nicht oder nur ansatzweise (Detailentwurf)
- ☆ Größere Software braucht einen systematischen Aufbau
  - ✧ Verstehen: Software >> 1 Kopf
  - ✧ Arbeit verteilen: Software-Entwicklung ist Teamarbeit
  - ✧ Einbetten in vorhandene Software: Schnittstellen und Wechselwirkungen
  - ✧ Verteilen auf Hardware-Komponenten, geographische Verteilung
  - ✧ Lokalisieren: Lokal ändern können, Auswirkungen von Erweiterungen und Ergänzungen abschätzen/begrenzen
- ➪ Aufbau im Großen: Software-Architektur, Konzept
- ➪ Aufbau im Detail: Software-Entwurf (und Programme)

16.2 Definitionen

Software – Die Programme, Verfahren, zugehörige Dokumentation und Daten, die mit dem Betrieb eines Rechnersystems zu tun haben. (IEEE 610.12)

Software-Entwurf (software design) – 1. Der Prozess des Definierens von Architektur, Komponenten, Schnittstellen und anderen Charakteristika eines Software-Systems oder einer Software-Komponente. 2. Das Ergebnis des Prozesses gemäß 1.

Software-Architektur (software architecture) – Die Organisationsstruktur eines Software-Systems oder einer Software-Komponente.

Architekturentwurf (Grobentwurf, architectural design) – 1. Der Prozess des Definierens einer Software-Architektur. 2. Das Ergebnis des Prozesses gemäß 1.

Detailentwurf (Feinentwurf, detailed design) – 1. Der Prozess der Festlegung der algorithmischen und datenstrukturmäßigen Details einer Software-Architektur (oder eines Teils davon. 2. Das Ergebnis des Prozesses gemäß 1.

Implementierung (implementation) – 1. Der Prozess des Realisierens einer Software-Lösung. Umfasst in der Regel Detailentwurf, Codierung, Integration und Prüfung. 2. Das Ergebnis des Prozesses gemäß 1.

16.3 Entwurfsprinzipien

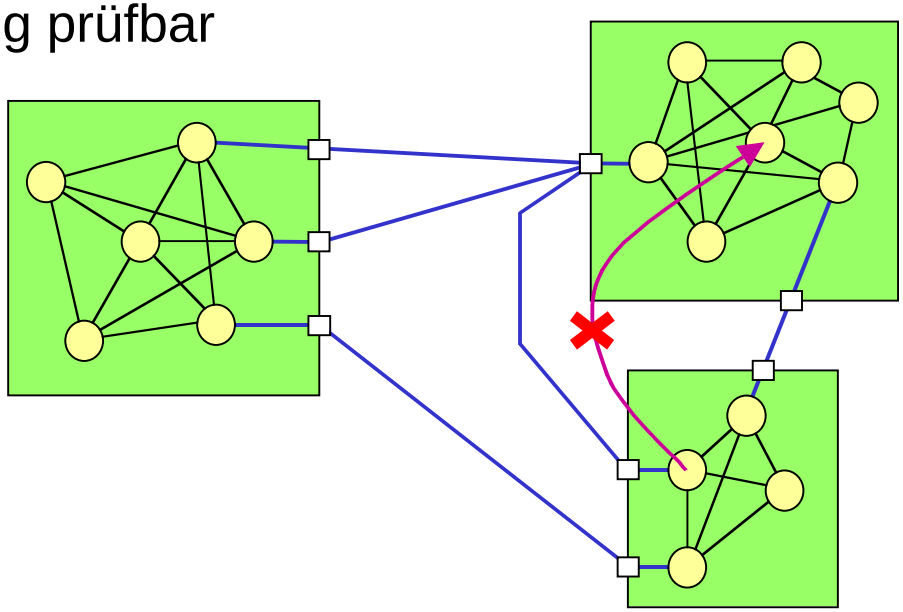
Systematischer Aufbau: Die Grundlage guten Entwurfs

- **Strukturen** (Module, Prozesse, Kommunikation...):
  - Gliederung in Komponenten und Interaktion zwischen Komponenten

- **Abstraktionen** (Komposition, Benutzung, Generalisierung, Aspekte)
  - Systematische Vergröberung und Verfeinerung nach verschiedenen Kriterien
  - Verständnis großer Zusammenhänge unter Weglassung der Details
  - Verständnis eines Details unter Weglassung/starker Vergröberung des Rests
  - Systematischer Zusammenhang zwischen Übersichten und Detailsichten

Modularität: Divide et impera oder *Das* Entwurfsprinzip schlechthin

- ☆ Gliederung der Gesamtlösung in kleinere, überschaubare Teillösungen
- ☆ Nicht irgendwie gliedern, sondern:
  - Modul = in sich geschlossene Einheit
  - Verwendung erfordert keine Kenntnisse über inneren Aufbau
  - Kommunikation mit Umgebung ausschließlich über Schnittstelle
  - Rückwirkungsfreie Änderbarkeit im Modulinnern
  - Korrektheit ohne Kenntnis der Einbettung prüfbar
- ☆ zwingend bei Entwurf großer Systeme



Das Geheimnisprinzip (Information Hiding):
Das Fundamentalprinzip zur Gliederung komplexer Systeme

- Modularisierungsprinzip: Kapselung von Entwurfsentscheidungen
- Modulverwender wissen, welche Entwurfsentscheidungen ein Modul implementiert, aber nicht wie.
- Typische Arten von Entwurfsentscheidungen:
  - wie eine Funktion realisiert ist
  - wie eine Datenstruktur aufgebaut ist und wie sie bearbeitet werden kann
  - wie Leistungen Dritter genutzt werden

Zusicherungen und Verträge: Die starken Geschwister des Geheimnisprinzips

- Zusicherungen definieren das Leistungsangebot eines Moduls
  - Voraussetzungen – was vorher erfüllt sein muss
  - Ergebniszusicherung – was nachher erfüllt ist
  - Invarianten – was nicht verändert werden darf
  - Verpflichtungen – mit der Verwendung übernommene Pflichten

- Verträge regeln die Zusammenarbeit (z. B. Delegation von Teilaufgaben) zwischen Modulen ("Design by Contract")

Qualität: Du sollst nicht schludern

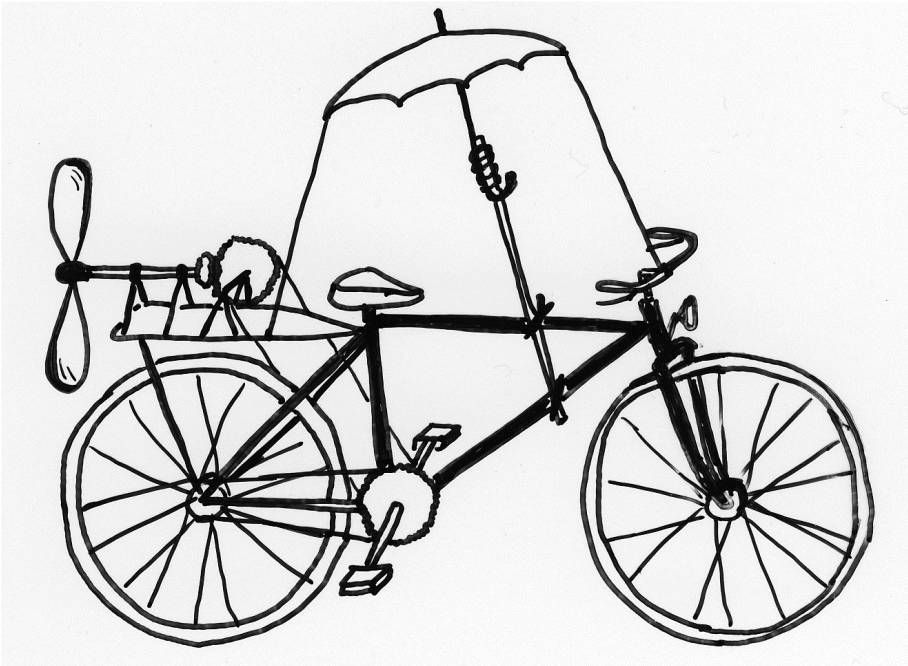
Ein guter Entwurf ist:

- ☆ Effektiv: erfüllt die Vorgaben, löst das Problem des Auftraggebers
- ☆ Robust
- ☆ Wirtschaftlich: gebrauchstauglich, kostengünstig, mehrfachverwendbar/mehrfachverwendet

- ➪ Erfordert kontinuierliche Prüfung

Ästhetik und Eleganz: Was gut ist, ist auch schön

- ☆ Gestaltet statt geworden
- ☆ Einfach und klar
- ☆ Problemadäquate Architektur



16.4 Produkte

- Resultat des Architekturentwurfs: Software-Architektur
  - Niedergelegt in einem Dokument (genannt Lösungskonzept, Software-Architektur, Architectural design [document], o.ä.)
  - Findet ihren Niederschlag in Coderahmen und/oder Code
  - Die Struktur des Codes ist eine bruchfreie oder zumindest formal rückverfolgbare Umsetzung der Architektur
  - Dokumentiert auch Einbettung in vorhandene Software sowie Beschaffungs- und Wiederverwendungsentscheide

- Resultat des Detailentwurfs: Dokumentierter Coderahmen für jede Komponente
  - Die Struktur des Detailentwurfs ist eine möglichst bruchfreie, mindestens aber formal rückverfolgbare Umsetzung der Architektur
  - Die Struktur des Codes ist eine 1:1-Umsetzung der Struktur des Detailentwurfs