Entwicklungsmethoden

Prof. Dr. Josef M. Joller
jjoller@hsr.ch

WIDERVERWENDBARKEIT, PORTABILITÄT UND INTEROPERABILITÄT

Übersicht

- Konzept der Wiederverwendung
- Was begünstigt die Wiederverwendung?
- Fallstudien
- Objekte und Wiederverwendung
- Wiederverwendung in unterschiedlichen Phasen
- Wiederverwendung und Wartung
- Portabilität
- Techniken, um Portabilität zu erreichen
- Interoperabilität

Wiederverwendbarkeit

Zwei Typen der Wiederverwendung

- zufällige Wiederverwendung
  - zuerst wird ein Produkt erstellt
  - dann werden spezielle Module den Kollegen angeboten
- geplante Wiederverwendung
  - zuerst werden wiederverwendbare Module entwickelt
  - dann baut man daraus auslieferbare Module

Warum wiederverwenden?

Ein Grund dafür

- die Software Entwicklung und Dokumentation ist teuer
- Wiederverwendung geschieht auf unterschiedlichen Stufen
  - Designs
  - Implementation
  - Testen
  - Dokumentation

Hauptgrund

- Wartung

→ **Die Wartung kostet 2/3 der Gesamtkosten**

Was spricht gegen Wiederverwendung?

Not invented here (NIH) Syndrome

Der Entwickler glaubt er könne es besser

Wo finde ich die Module, die ich brauche?

Was kostet die Wiederverwendung?

Fallstudie

- theoretisch kann im Maximum 85% wiederverwendet werden
- wie sieht die Praxis aus?

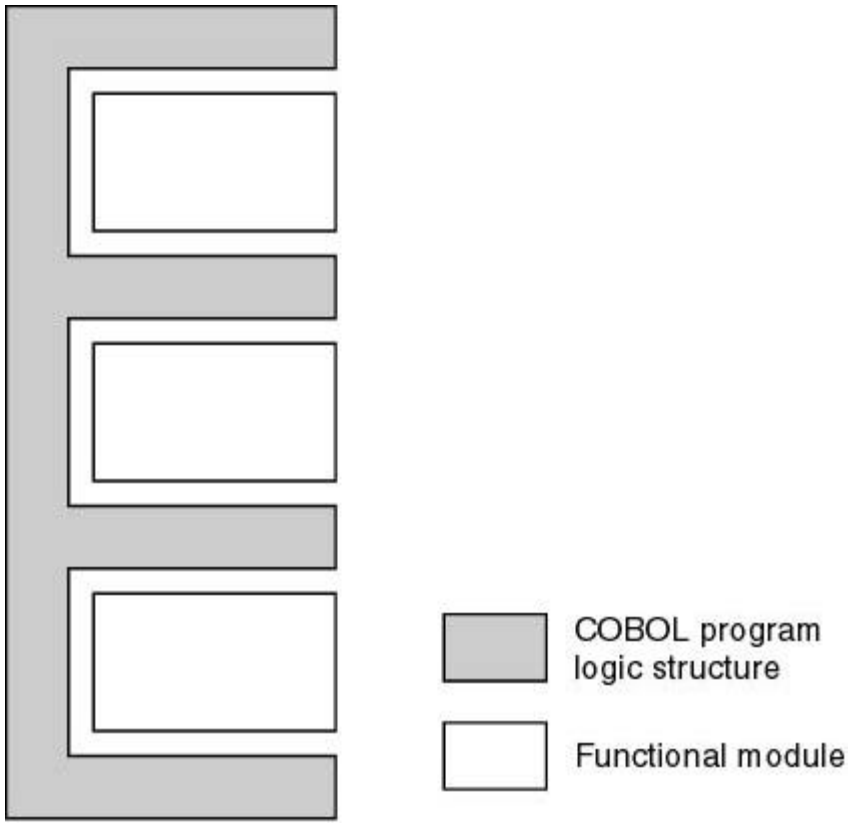
Raytheon Missile Systems Division

Kommerzielle SW

Geplante Wiederverwendung

- Designs
  - 6 Code Templates
- COBOL Code
  - 3200 wiederverwendbare Module

Rate: 60% (1976–1982) wiederverwendet



GTE Data Services

Klassische Datenverarbeitung

Wiederverwendung wurde durch das Management unterstützt

- alle, deren Module wiederverwendet werden, erhalten einen Bonus
- alle, die Module wiederverwenden, erhalten einen Bonus

Zufällige Wiederverwendung auf der Stufe “Module”

Ergebnisse

- (1988) 15% Wiederverwendung, $1.5 Millionen gespart
- (est. 1989) 20% Wiederverwendung
- (est. 1993) 50% Wiederverwendung

European Space Agency

Ariane 5 Rakete explodierte 37 Skunden nach dem "lift-off"

Kost: $500 Millionen

Grund: Versuch eine 64-bit Integer in einen 16-bit unsigned integer abzuspeichern, ohne Ada Exception **Handler**

On-board Computers stürzten ab, die Rakete wurde gesprengt

Eigentlich war die Konversion nicht nötig!

European Space Agency

Ursache

- zehn Jahre vor dem Absturz wurde mathematisch bewiesen, dass ein Overflow nicht möglich sein kann — bei Ariane 4
- die Software wurde ungeprüft übernommen und bei Ariane 5 eingesetzt

Lehre

- Falls Software n einer anderen Umgebung eingesetzt wird, muss sie getestet werden, in dieser neuen Umgebung!

Wiederverwendung im Design

Design Wiederverwendung

- Bibliotheken und Toolkits
  - gekauft oder selber entwickelt
- Framework
  - spezifisch für ein bestimmtes Arbeitsgebiet
  - aber sehr teuer in der Entwicklung
- Design Patterns / Entwurfsmuster
  - die gängige Technik
  - Standardwerk : Erich Gamma et al Design Patterns / Entwurfsmuster
- Software Architekture
  - gängige Technik
  - Standardwerk : Stahl, Sommerlat et a: Software Architektur (2 Bde)

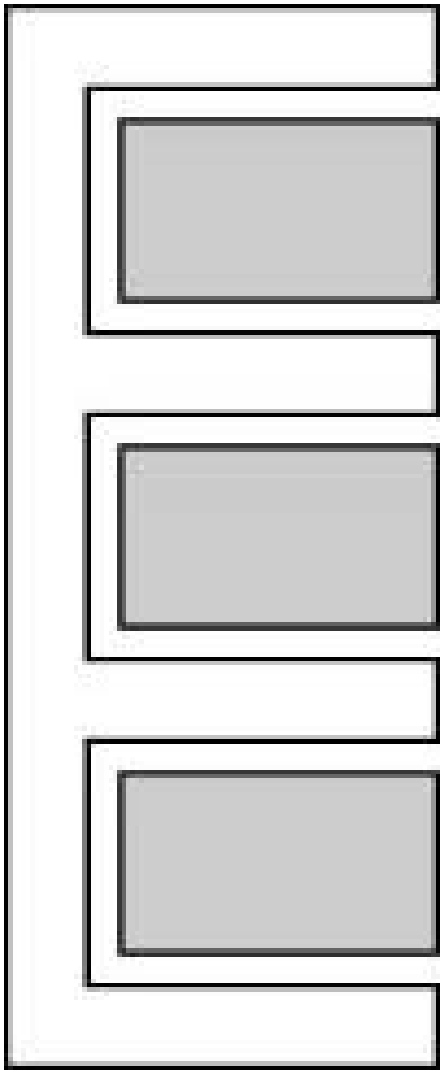
Bibliothek oder Toolkit

Set wiederverwendbarer Module

Beispiele:

- Std Bibliotheken (Matrizen)
- GUI Klassen und Toolkits

Der Benutzer spezifiziert die Programmlogik (weiss im nebenstehenden Bild)



Applikations Frameworks

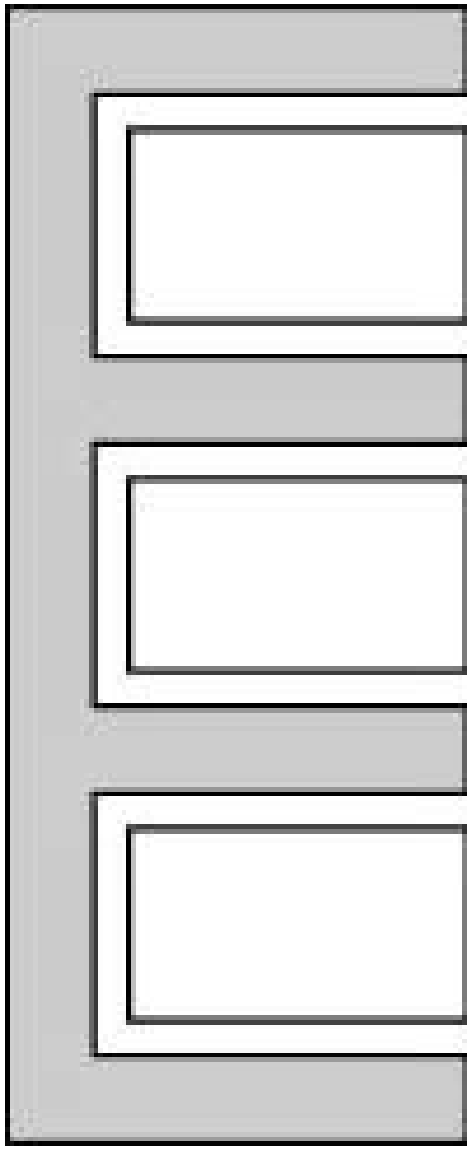
Kontrolllogik wird vorgegeben

“Hot spots” (weisse Teile in Bild) werden ergänzt

Kurze Entwicklunszeiten

Designs werden wiederverwendet

Aufwendig in der Erstellphase

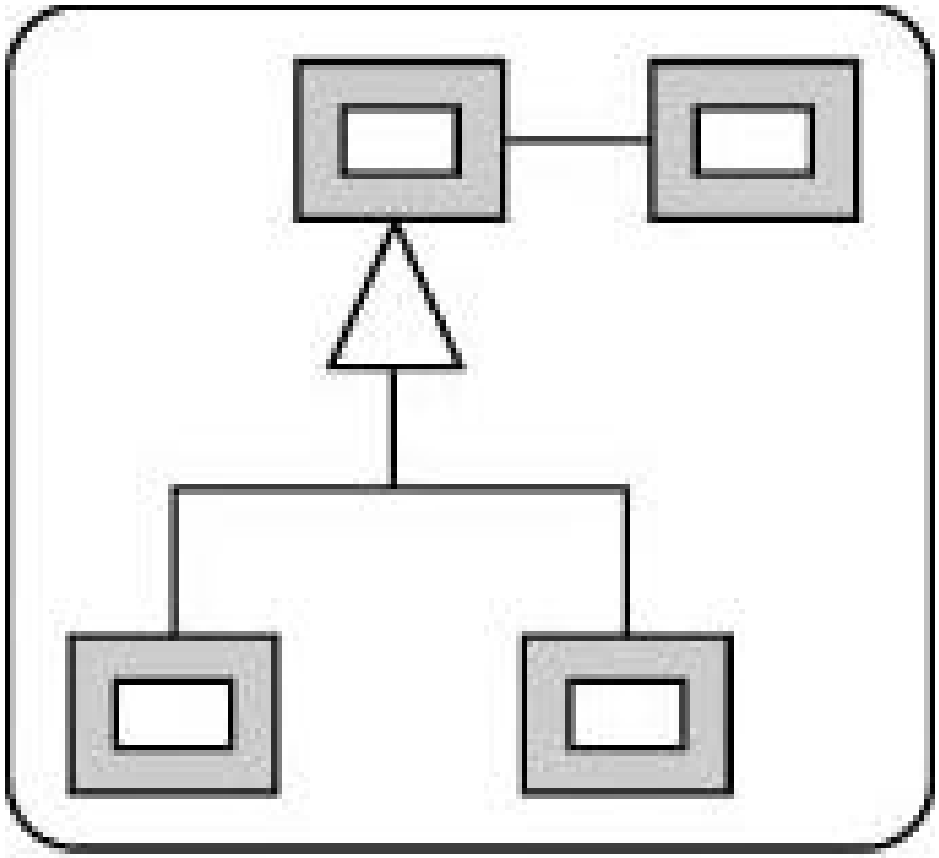


Design Pattern

Patterns = allgemeine Lsg

Patterns bestehen aus mehreren Klassen

Die aufgeführten Klassen müssen angepasst werden, das das Pattern allgemein ist.



Wiederverwendung von Software Architekturen

Architekturen wieder zu verwenden führt in der Regel zu grossen Einsparungen

Ein Beispiel

- Software Produktlinien

Fallstudie:

- Hewlett-Packard Printers (1995 bis 1998)
  - Aufwand zur Entwicklung der Printer Firmware sank um Faktor 4
  - Zeit zur Entwicklung anderer Firmware sank m Faktor 3
  - Wiederverwendung stieg um über 70%

Objekte und Produktivität

Wiederverwendung

- der Klassen (Objekte gehören alle zu einer Klasse)
- Objekte werden nicht wiederverwendet (Objekt = Instanz einer Klasse in einem konkreten Programm)

Allgemein

- Software Kosten können gesenkt werden
- die Qualität kann verbessert werden
- Grossprojekte werden möglich, durch Bündelung vieler kleinerer Module

Schwierigkeiten und Probleme

Lernkurve

- speziell schwierig im GUI Bereich

Probleme mit Vererbung

- neue Unterklassen erhalten die alten Klassen
- ABER
  - Änderungen der Oberklasse wirken sich global aus

Warum Portabilität?

Portabilität ist schwierig

- bei Kaufsoftware (Microsoft auf Mac)
- Hardware ist inkompatibel (Windows auf AS/400)
- Software und Hardware haben unterschiedliche Lebenszyklen

Portabilität spart Geld!

Portable System Software

- isoliert Betriebsystem abhängige Teile
  - UNIX kernel, device-drivers
- die Software wird abstrakter
  - Graphik Interface (Mac auf Palm)

Portabilität Strategien

Portable Applikations-Software bedingt

- populäre Programmiersprachen
- gängige Betriebsysteme
- Sprachstabdards
- nummerische Standards (IEEE Arithmetik)
- saubere Dokumentation

Trends

JAVA und CORBA sind populär

Java EnterpriseBeans (EJB's)

- Session / Transaction,
- Entity / DB,
- Message-Driven / Messaging

Web-basierte Applikationen

- XML (Extensible Markup Language) wird dominant
- XML Protocol (SOAP, Web Services) erlauben Services (Kosten!)