


Prinzipien für Embedded-Softwarearchitekturen

Inhalt

	ARCHPRINZ
Inhalt	1
1 Inhalt	3
2 Themenorientierung: Rund um den Entwurf von Software-Architekturen	4
3 Entwurfsprinzipien	5
3.1 Definition	5
3.2 Übersicht und Zusammenhang	6
3.3 Geheimnisprinzip	7
3.4 Abstraktion	8
3.5 Offen-/Geschlossen-Prinzip	9
3.6 Kapselung	10
3.7 Modularisierung	11
3.8 Lose Bindung / Kopplung	12
3.9 Dependency Injection zur Externalisierung von Abhängigkeiten	13
3.10 Umkehrung von Abhängigkeiten	15
3.11 Hohe Kohäsion	17
3.12 Trennung von Verantwortlichkeiten	18
3.13 Konzeptionelle Integrität	19
3.14 Hohe Kohäsion	20
3.15 Sieben bis Zwölf Prinzip	21
3.16 Pareto-Verteilung	22
4 Resümee	23

Vortragsreihe Architektur – Embedded Software Engineering Kongress 2014




Embedded Software Engineering Kongress


Prinzipien für Embedded-Softwarearchitekturen

Die Prinzipien kennen und richtig einsetzen

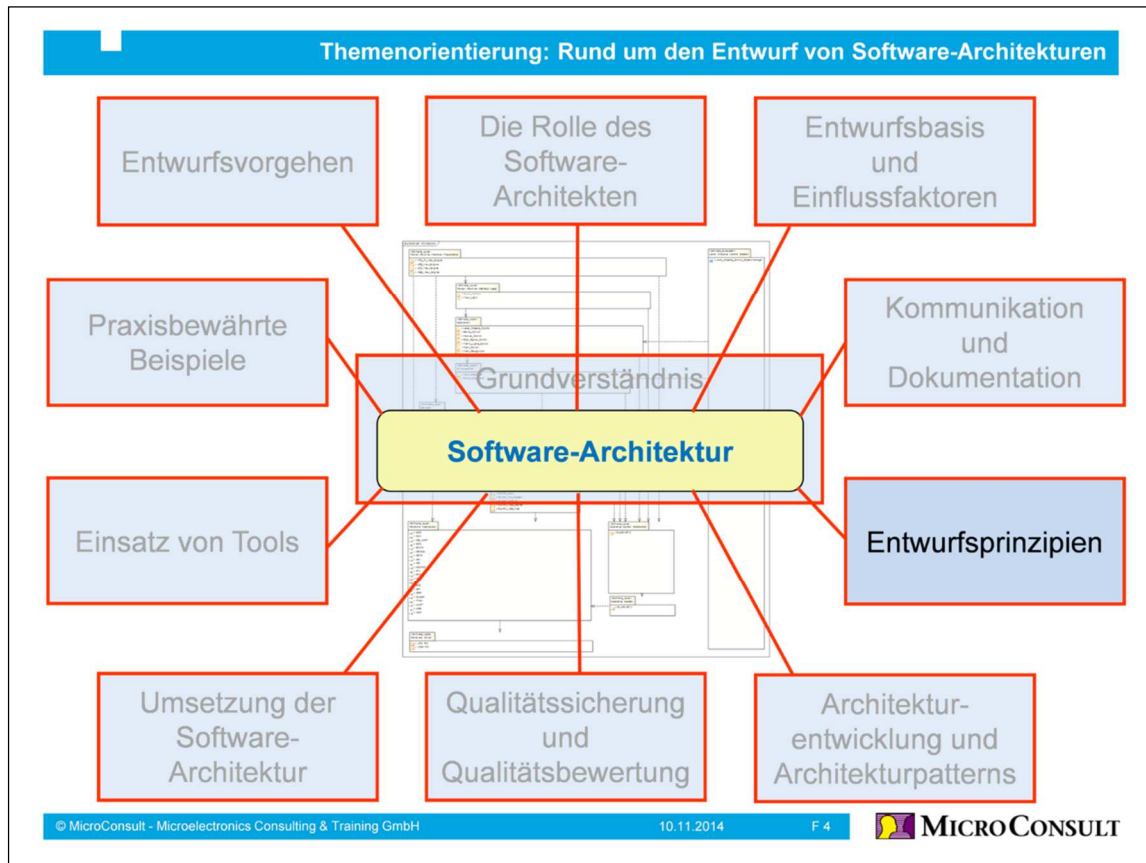
MICROCONSULT GmbH
Dipl.-Ing. (FH) **Thomas Batt**
Manager Management-Training und -Coaching
Trainer & Coach für Embedded- und Echtzeitsysteme
Charles-de-Gaulle-Str. 6 • 81737 München • Germany
Tel.: +49 (0)89 450617-35
FAX: +49 (0)89 450617-17
E-Mail: t.batt@microconsult.com
Internet: www.microconsult.de



1 Inhalt

Inhalt
▪ Themenorientierung: Rund um den Entwurf von Software-Architekturen
▪ Definition von Entwurfsprinzipien
▪ Übersicht und Zusammenhang einzelner Entwurfsprinzipien
▪ Vorstellung ausgewählter Entwurfsprinzipien
▪ Resümee
Download-Link für diese Präsentation: http://download.microconsult.net/ese2014/archprinz.zip
© MicroConsult - Microelectronics Consulting & Training GmbH 10.11.2014 F 3  MICROCONSULT

2 Themenorientierung: Rund um den Entwurf von Software-Architekturen



3 Entwurfsprinzipien

3.1 Definition


Entwurfsprinzipien – Definition

- Entwurfsprinzipien sind für alle Architekturelemente **gültige Grundlagen**, an denen der Software-Architekt sein Handeln zugrunde legt.
- Der Software-Architekt gewichtet und verwendet die Entwurfsprinzipien auf **Basis** der zu erfüllenden funktionalen und nicht-funktionalen **Anforderungen**.
- Die Entwurfsprinzipien setzt der Software-Architekt durch die Anwendung von Regeln und Mustern (**Pattern**) um.
- Im Rahmen der **Qualitätssicherung** muss der Software-Architekt die Einhaltung der Entwurfsprinzipien gewährleisten und nachweisen.
- Prinzip == **Heuristik**

© MicroConsult - Microelectronics Consulting & Training GmbH

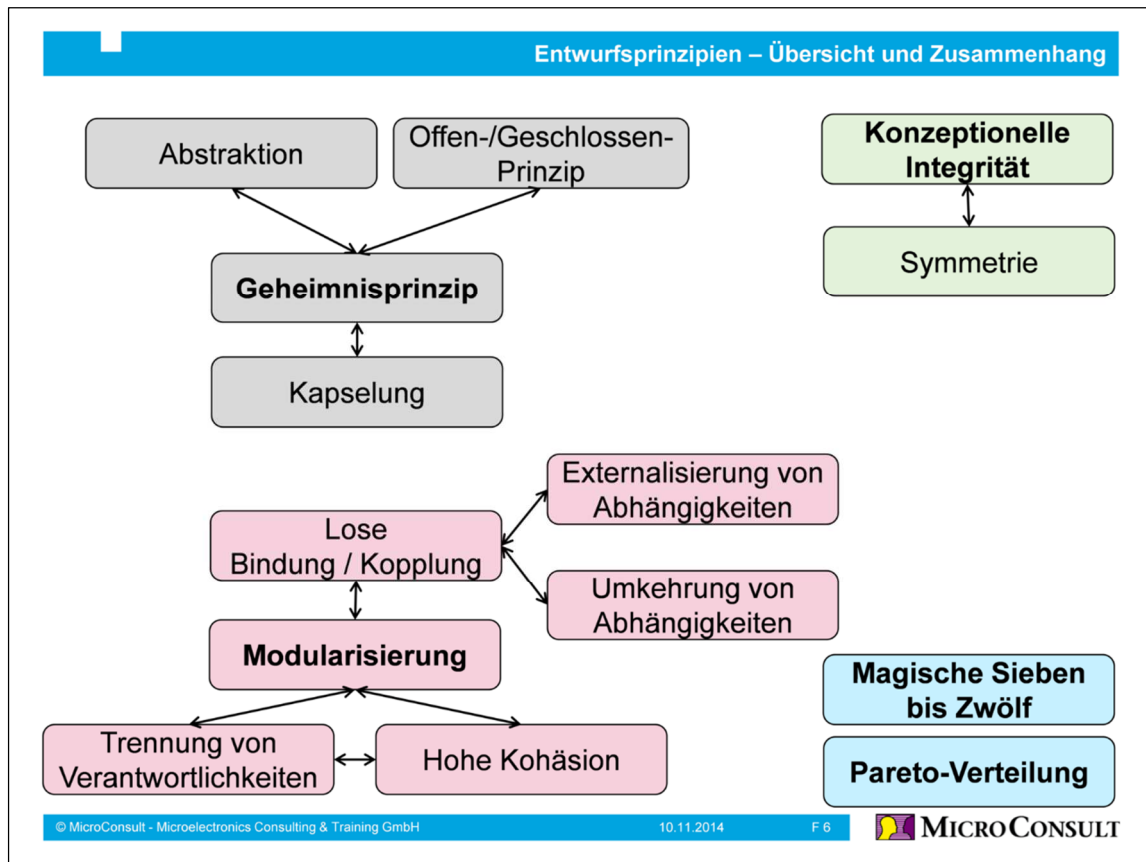
10.11.2014

F 5

 MICROCONSULT

Neben dem fachlichen Wissen beherrscht der Software-Architekt auch die folgenden Entwurfsprinzipien.

3.2 Übersicht und Zusammenhang

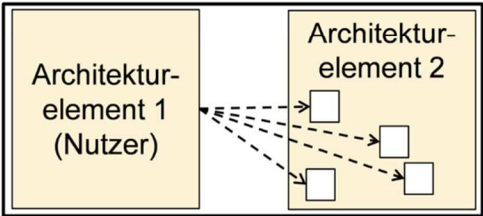


3.3 Geheimnisprinzip

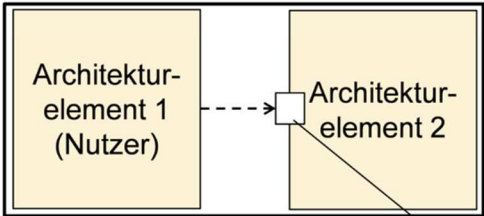
Entwurfsprinzipien – Geheimnisprinzip

Geheimnisprinzip

- Prinzip: Bewusstes Verstecken von Informationen (information hiding) für den Nutzer eines Architekturelements



X schlecht



✓ gut

- Kriterien:
 - Minimale Anzahl von Abhängigkeiten zwischen Architekturelementen
 - → Hoher Grad der Entkopplung
 - Trennung von Schnittstelle und Implementierung
- Ziel:
 - Verwendung von Diensten oder Schnittstellen, ohne die Kenntnisse über deren Implementierungen zu besitzen
 - Robustheit gegenüber Änderungen
- Mittel: Interfacebildung, Kapselung

Schnittstelle mit API

© MicroConsult - Microelectronics Consulting & Training GmbH
10.11.2014 F 7
 MICROCONSULT

Das Geheimnisprinzip wurde 1972 von David Parnas entwickelt.

API (Application Programers Interface)

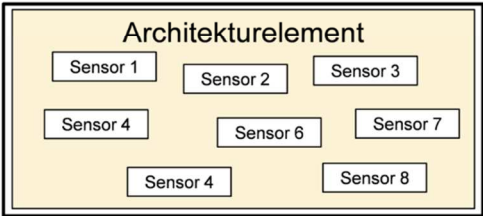
AE (Architekturelement)

3.4 Abstraktion

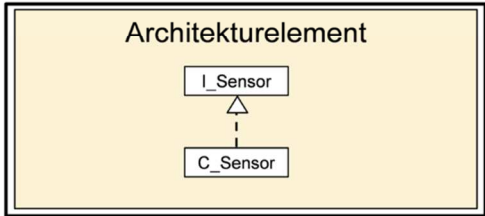
Entwurfsprinzipien – Abstraktion

Abstraktion

- Prinzip: Weg vom Konkreten, hin zum Abstrakten
 → Betrachtung eines idealisierten, vereinfachten Modells ohne Details




X schlecht



✓ gut

- Kriterien:
 - Erkennen von Gemeinsamkeiten auf allen Ebenen (AE, Daten, Funktionen, Verhalten)
 - Erkennen von Schnittstellen
 - Gruppierung und Abstraktion der Gemeinsamkeiten
- Ziel:
 - Einsatz eines Top-Down-Ansatzes in der Entwicklung → Beginn mit dem Abstraktesten
 - Erreichen von Vielgestaltigkeit und Erweiterbarkeit
- Mittel: Interfacebildung, Kapselung

© MicroConsult - Microelectronics Consulting & Training GmbH
10.11.2014
F 8

 **MICROCONSULT**

3.5 Offen-/Geschlossen-Prinzip

Entwurfsprinzipien – Offen-/Geschlossen-Prinzip

Offen-/Geschlossen-Prinzip

- Prinzip: Architekturelemente sind offen für Erweiterungen und geschlossen für Änderungen

Architekturelement

X schlecht

Architekturelement

✓ gut

- Kriterien:
 - Erkennen von Gemeinsamkeiten auf allen Ebenen (AE, Daten, Funktionen, Verhalten)
 - Erkennen von Schnittstellen
 - Gruppierung und Abstraktion der Gemeinsamkeiten
 - Erweiterbarkeit ist bereits beim Entwurf des Architekturelements zu berücksichtigen
- Ziel:
 - Erweitern, ohne alles erneut zu testen, ...
 - Punktuell erweitern, ohne anderes zu ändern
 - ➔ Erweitern ohne Seiteneffekte
- Mittel: Interfaces, Vererbung

© MicroConsult - Microelectronics Consulting & Training GmbH
10.11.2014
F 9

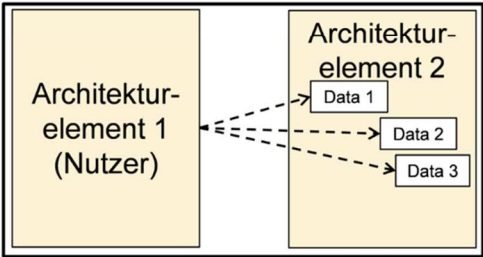
MICROCONSULT

3.6 Kapselung

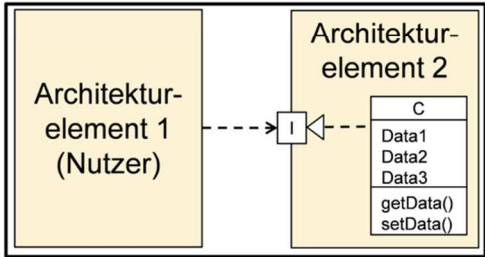
Entwurfsprinzipien – Kapselung

Kapselung

- Prinzip: Nutzung eines Architekturelementes ohne den Zugriff auf enthaltene Elemente




X schlecht



✓ gut

- Kriterien:
 - Kapselung von (globalen) Elementen (Daten, Funktionen → Verhalten)
 - → Hoher Grad der Entkopplung
- Ziel ist die Verbesserung von:
 - Wiederverwendbarkeit
 - Austauschbarkeit
- Mittel: Schichten- und Subsystembildung, Interfacebildung, Modulbildung, objektorientierter Ansatz: Klassen- und Objektbildung

© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014F 10 MICROCONSULT

3.7 Modularisierung

Entwurfsprinzipien – Modularisierung


Modularisierung

- Prinzip: **Zerlegung** des Gesamten in Teile (Architekturelemente)

Architektur- element 1	Architektur- element 2	Architektur- element 3	Architektur- element 4
Architektur- element 5	Architektur- element n

- Kriterien:
 - **Grenze**
 - **Granularität** (nicht zu groß und nicht zu klein)
 - **Durchgängige Abstraktionsebene**
- Ziel:
 - Arbeitsteilung ermöglichen (Verteilung der Umsetzung)
 - Handhabbarkeit erleichtern (Konfiguration, Test, Entwicklung, Dokumentation)
- Mittel: Sachverstand, Schichten- und Subsystembildung

© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014F 11

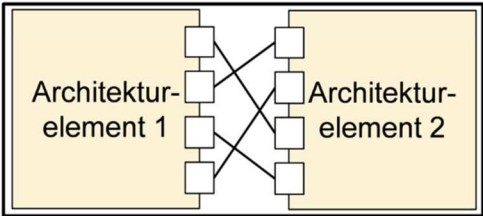
 **MICROCONSULT**

3.8 Lose Bindung / Kopplung

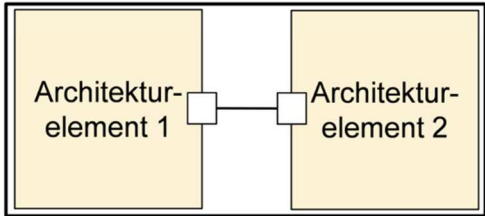
Entwurfsprinzipien – Lose Bindung / Kopplung

Lose Bindung / Kopplung

- Prinzip: **Reduktion der Abhängigkeiten** zwischen Architekturelementen




X schlecht



✓ gut

- Kriterien:
 - Minimale Anzahl von Abhängigkeiten zwischen zwei Architekturelementen
 - Minimale Anzahl von Abhängigkeiten zu anderen Architekturelementen
- Ziel:
 - Arbeitsteilung ermöglichen (Verteilung der Umsetzung)
 - Handhabbarkeit erleichtern (Konfiguration, Test, Entwicklung, Dokumentation)
- Mittel: Interfacebildung, Vererbung, Ressourcenteilung, u.a.

© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014F 12 **MICROCONSULT**

3.9 Dependency Injection zur Externalisierung von Abhängigkeiten

Entwurfsprinzipien – Externalisierung von Abhängigkeiten (1..)

Externalisierung von Abhängigkeiten

- Prinzip: Klasse stellt Schnittstelle
 - zur Initialisierung von Zeigern/Referenzen auf konkrete Objekte bereit
 - zum Arbeiten mit konkreten Objekten bereit

Architektur-
element 1

C1

—

Architektur-
element 2

C2

X schlecht

Architektur-
element 1

C1

- - -

Architektur-
element 2

I2

<--

C2

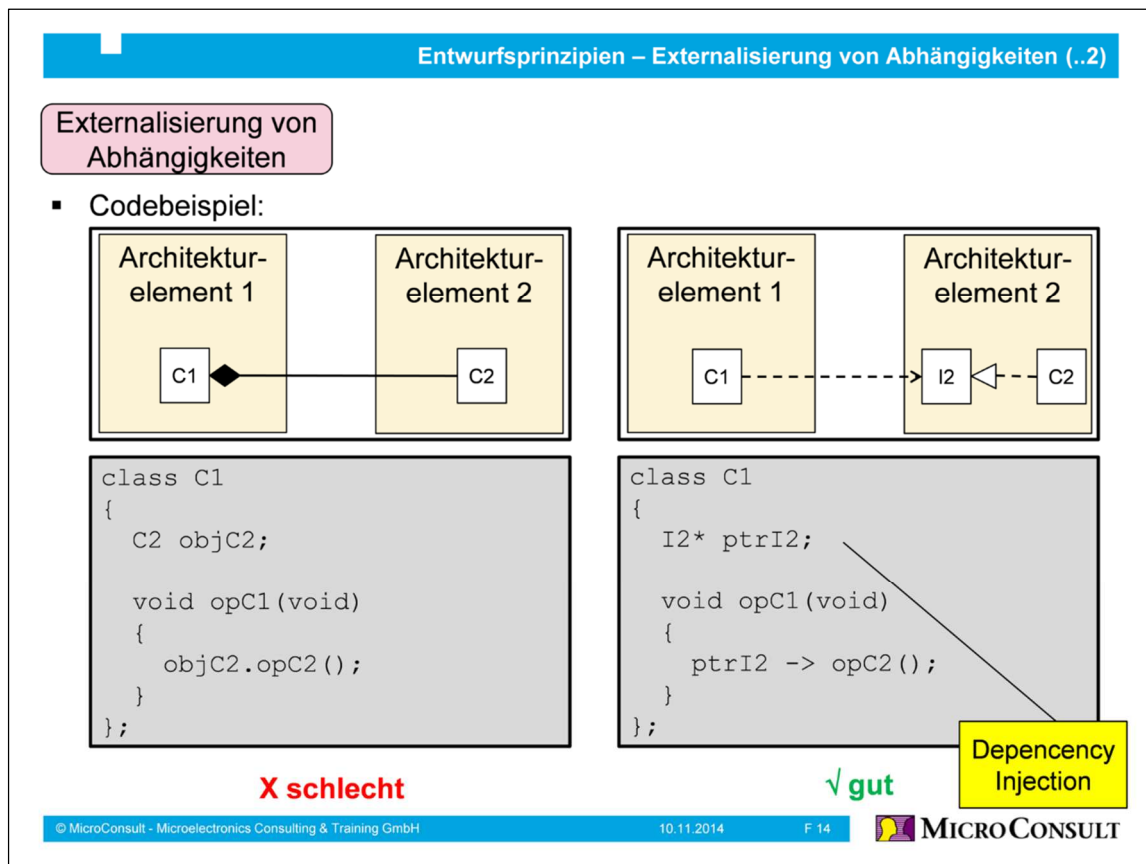
✓ gut

- Kriterien:
 - Fokus auf der Erzeugung und Initialisierung
 - Wissen über die zu verwendende Klasse liegt außerhalb
 - Abhängigkeiten zu konkreten Klassen vermeiden
- Ziel:
 - Minimierung der Kopplung zwischen Klassen
- Mittel: Interfacebildung, Erzeugungsmuster

Dependency
Injection

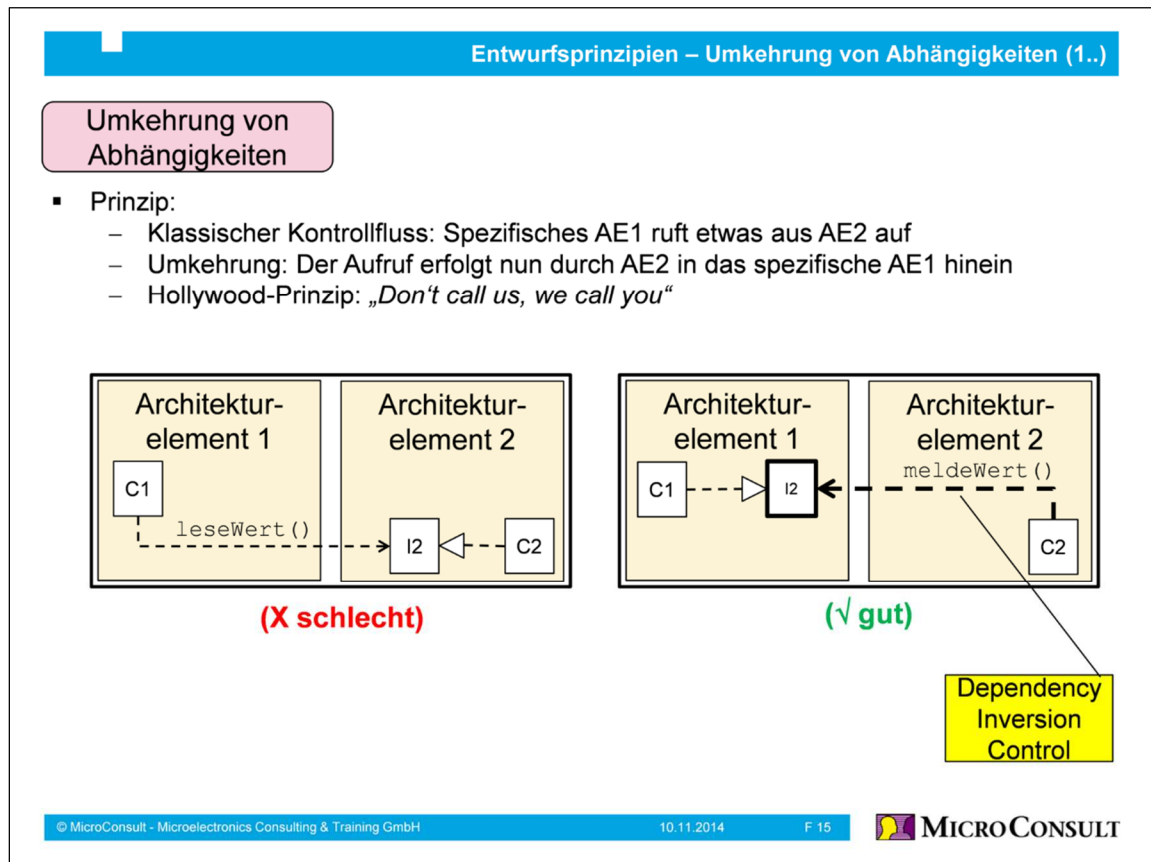
© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014F 13

MICROCONSULT



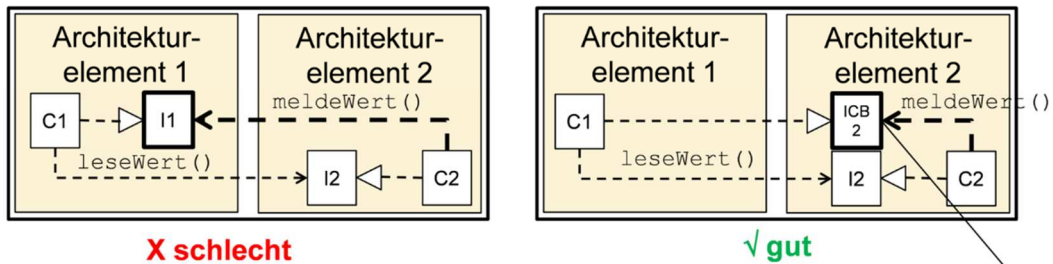
Der Pointer ptrI2 muss vor der Verwendung unbedingt ungleich Null initialisiert sein. Vor der Verwendung sollte hierzu eine entsprechende Sicherheitsabfrage erfolgen.

3.10 Umkehrung von Abhängigkeiten



Entwurfsprinzipien – Umkehrung von Abhängigkeiten (..2)

Umkehrung von Abhängigkeiten



X schlecht
(bidirektionales include)

✓ gut

Dependency
Inversion
Control

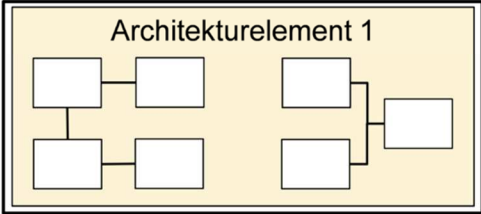
- Kriterien:
 - Umkehrung der Abhängigkeiten zwischen Architekturelementen
 - Umkehrung der Inkludierungsrichtung
 - Bidirektionale und zirkulare Abhängigkeiten vermeiden
- Ziel: Verbesserung der
 - Portabilität
 - Wiederverwendbarkeit
 - Austauschbarkeit
- Mittel: Callback-Strukturen, Dependency Injection

3.11 Hohe Kohäsion

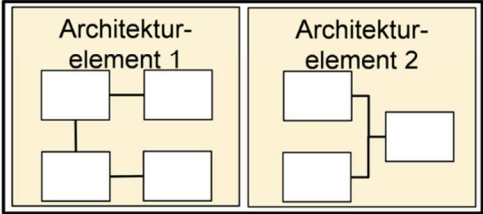
Entwurfsprinzipien – Hohe Kohäsion

Hohe Kohäsion

- **Prinzip:**
Kohäsion: Grad der Bindung zwischen Elementen innerhalb eines Architekturelements




X schlecht



✓ gut

- **Kriterien:**
 - Hohe Kohäsion → Nur logisch **Zusammengehöriges in ein Architekturelement**
 - Keine redundante Abbildung von Funktionalitäten
 - Geringe Kohäsion → Vermischte Verantwortlichkeiten in einem Architekturelement
- **Ziel:**
 - Erhaltung einer fachlich, logisch strukturierten Architektur zur Erhöhung der Wiederverwendbarkeit und Erweiterbarkeit
- **Mittel:** Gegenseitige logische / fachliche Trennung und Abgrenzung

© MicroConsult - Microelectronics Consulting & Training GmbH
10.11.2014
F 17

 **MICROCONSULT**

3.12 Trennung von Verantwortlichkeiten

Entwurfsprinzipien – Trennung von Verantwortlichkeiten

Trennung von Verantwortlichkeiten

- Prinzip: Überlappungsfreie Trennung von Verantwortlichkeiten

X schlecht

✓ gut

- Kriterien:
 - Betrachtung der Architekturelemente selbst
 - Betrachtung der Abhängigkeiten zwischen den Architekturelementen
- Ziel ist die Verbesserung der:
 - Wiederverwendbarkeit
 - Erweiterbarkeit
 - Wartbarkeit
- Mittel: Sachverstand, Modularisierung, hohe Kohäsion

© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014F 18

MICROCONSULT

3.13 Konzeptionelle Integrität

Entwurfsprinzipien – Konzeptionelle Integrität

Konzeptionelle Integrität

- Prinzip: Für das mehrfache Auftreten von gleichartigen Aufgaben und Problemen immer die gleiche Lösung verwenden.

X schlecht

✓ gut

- Kriterien:
 - Erkennung von inkonsistenten Ansätzen in der Architektur
- Ziel:
 - Erhaltung einer fachlich, logisch strukturierten Architektur zur Erhöhung der Wiederverwendbarkeit und Erweiterbarkeit
- Mittel: Sachverstand, Einsatz von Pattern und Architekturregelwerken

© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014 F 19 MICROCONSULT

3.14 Hohe Kohäsion

Entwurfsprinzipien – Symmetrie

Symmetrie

- Prinzip: Die Wahl der Elemente ist zueinander symmetrisch.
Diese strukturelle Symmetrie überträgt sich auch auf die Abläufe.

X schlecht

✓ gut

- Kriterien:
 - Parallele / ähnliche Pfade sind symmetrisch identisch strukturiert
 - Pfade auf gleicher Abstraktionsebene sind symmetrisch identisch strukturiert
- Ziel ist die Erhaltung einer:
 - verständlicheren Architektur
 - in sich konsistenten Architektur
- Mittel: Sachverstand

© MicroConsult - Microelectronics Consulting & Training GmbH10.11.2014F 20 MICROCONSULT

3.15 Sieben bis Zwölf Prinzip

Entwurfsprinzipien – Magische Sieben bis Zwölf

Magische Sieben bis Zwölf

- Prinzip: Das Menschliche Gehirn kann sich „auf den ersten Blick“ maximal sieben bis zwölf Elemente merken.

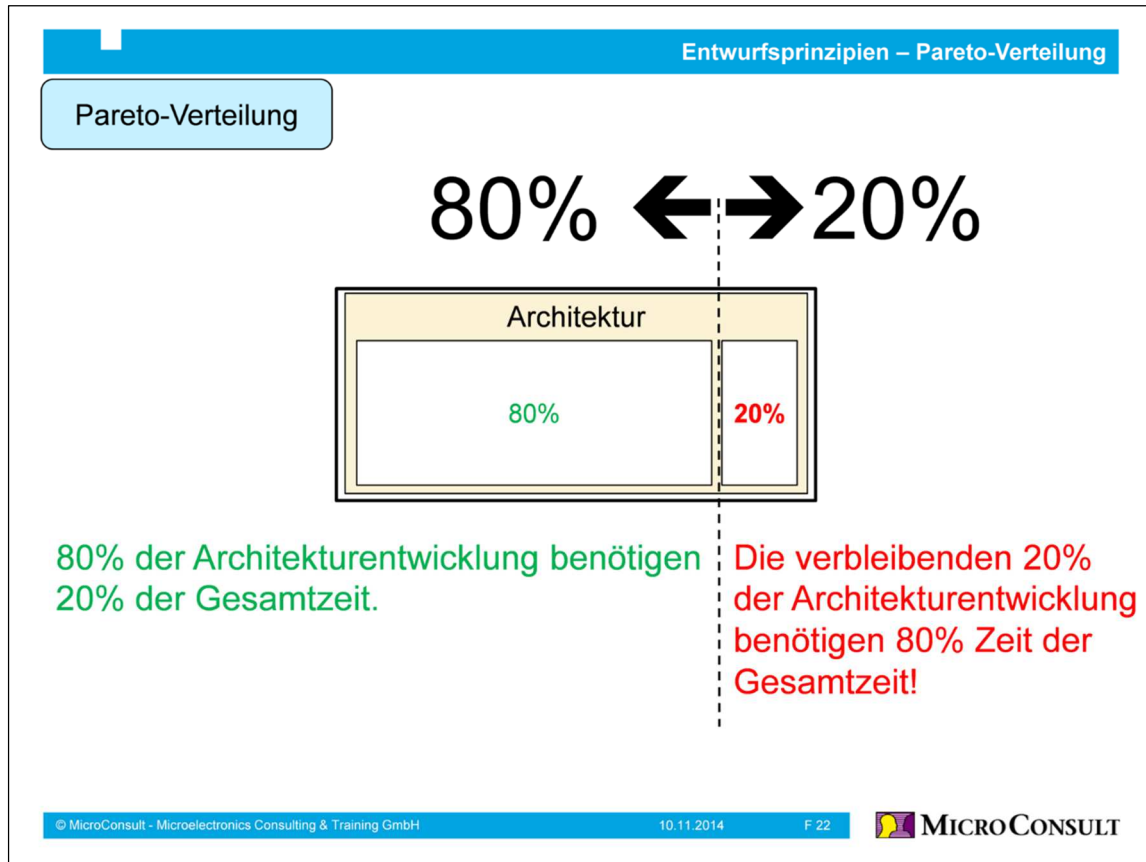
X schlecht

✓ gut

- Kriterien:
 - Eine Architektur besteht auf der obersten Hierarchie aus maximal 12 Elementen
 - Jedes Element enthält wiederum maximal 12 Elemente
 - Es gibt maximal 12 Hierarchieebenen
 - Jedes Element hat maximal 12 Abhängigkeiten
 - u.s.w.
- Ziel:
 - Erhöhte Übersichtlichkeit und erhöhtes Verständnis der Architektur
- Mittel: Weitestgehende Einhaltung der magischen Zahlen Sieben bis Zwölf „für alles“

© MicroConsult - Microelectronics Consulting & Training GmbH
10.11.2014
F 21
 MICROCONSULT

3.16 Pareto-Verteilung



Die Pareto-Verteilung bzw. das 80/20 Prinzip nach Vilfredo Pareto lässt sich auf die Entwicklung einer auf Software-Architektur übertragen.

4 Resümee

Resümee


Software-Architektur-Entwurfsprinzipien ...

- helfen dem Software-Architekten, **qualitativ hochwertige Software-Architekturen** zu entwerfen
- sind durch individuell neuentwickelte Prinzipien **erweiterbar**
- sind Teil der **Software-Architekturrichtlinie**

© MicroConsult - Microelectronics Consulting & Training GmbH

10.11.2014

F 23

 **MICROCONSULT**