

Qualitätsmodelle

- Top-down (z.B. ISO 9126)

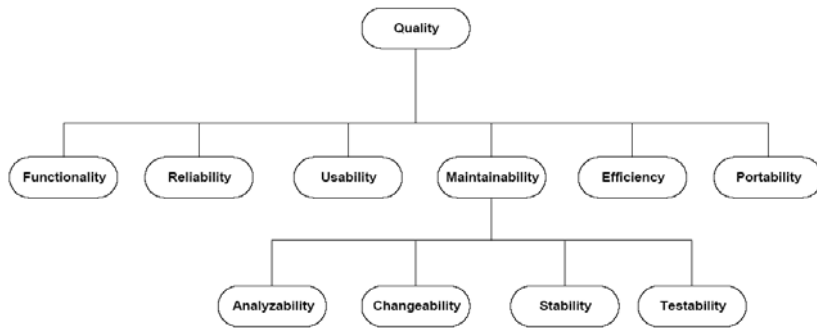
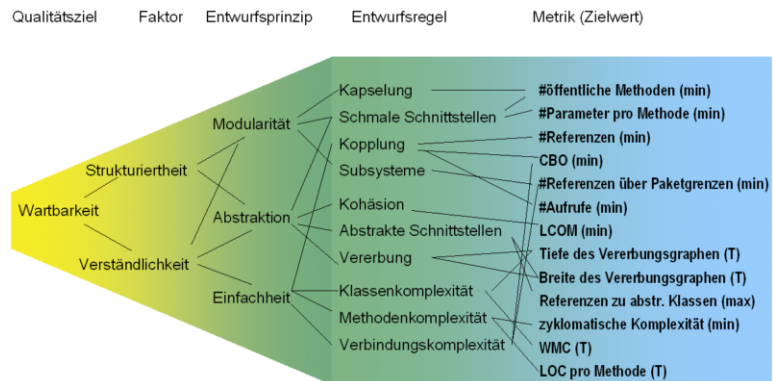


Figure B 2: ISO/IEC quality goals of maintainability

Entwurfszielgetriebenes Qualitätsmodell



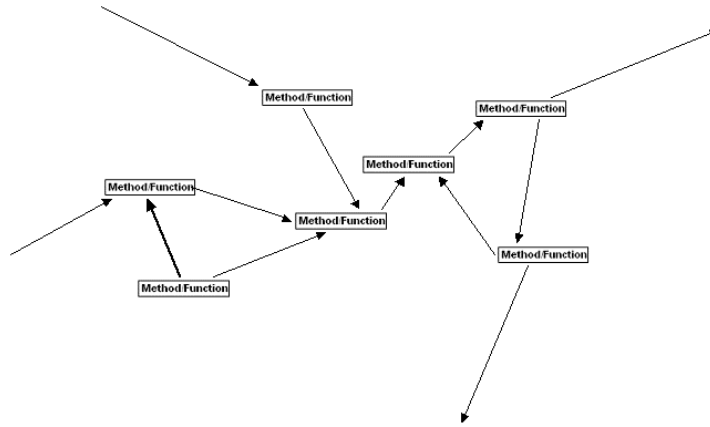
Qualitätsmodelle

- Bottom-up (oder was kann ich messen)
 - Harte Fakten
 - Architekturabweichungen und Schnittstellenverletzungen
 - Projektspezifische Architekturregeln
 - Zyklen
 - Grosse duplizierte Code Blöcke
 - Regelverletzungen
 - Hinweise auf potentielle Probleme
 - Typischerweise durch Metriken
 - Grösse
 - Kopplung
 - Komplexität
 - Metrikwerte müssen interpretiert werden

Architekturanalyse

- Grundvoraussetzungen
 - Abstraktion
 - Analyse der Veränderung über die Zeit

Analyse auf Symbolebene



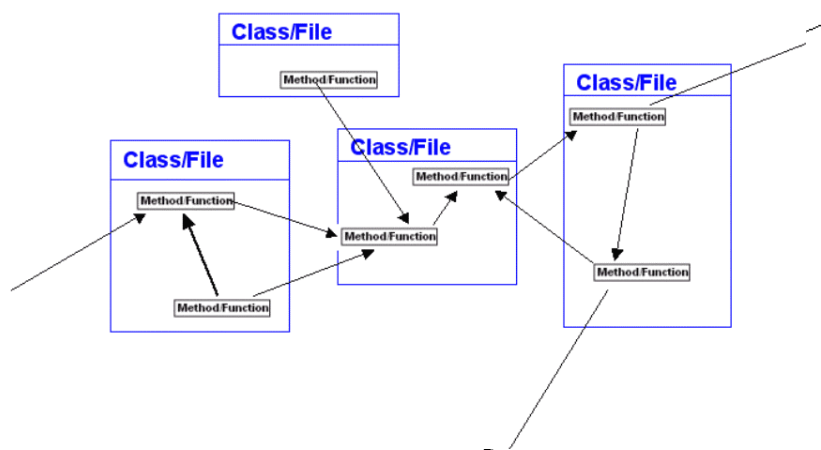
500.000 LOC
50.000 Symbole

Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

7

Analyse auf Klassen/Dateiebene



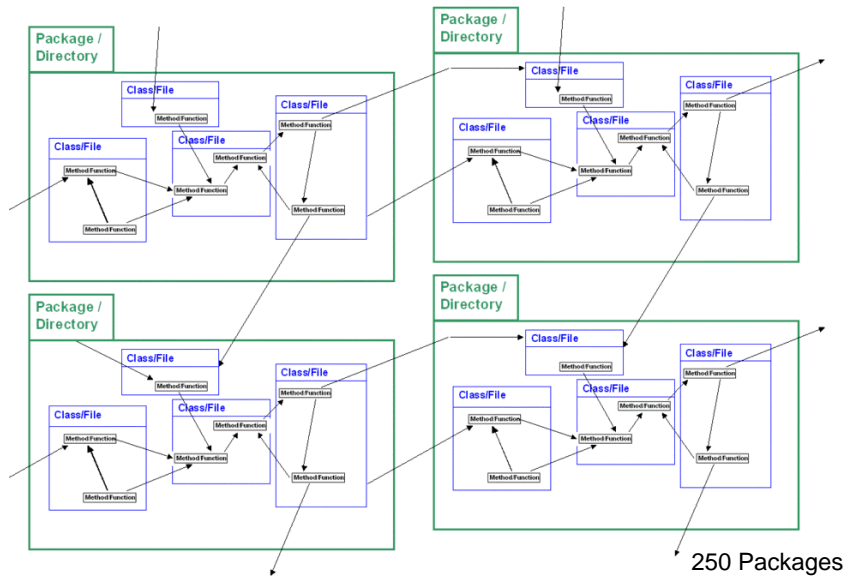
4.000 Klassen

Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

8

Analyse auf Paket/Verzeichnisebene

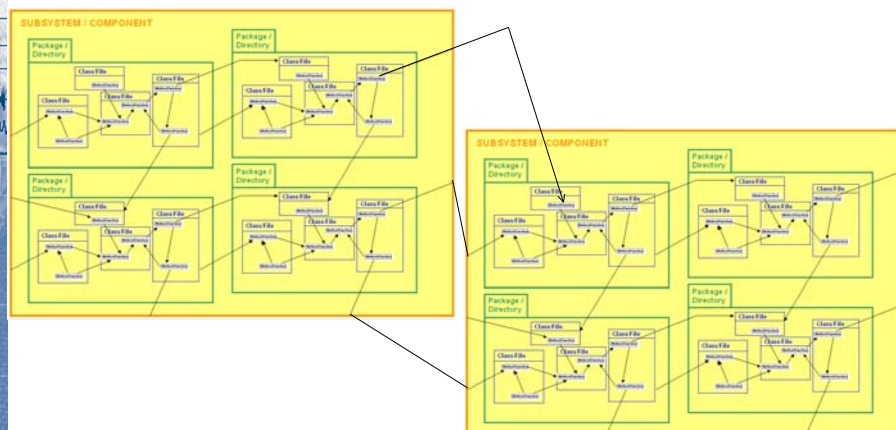


Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

9

Analyse auf Subsystemebene



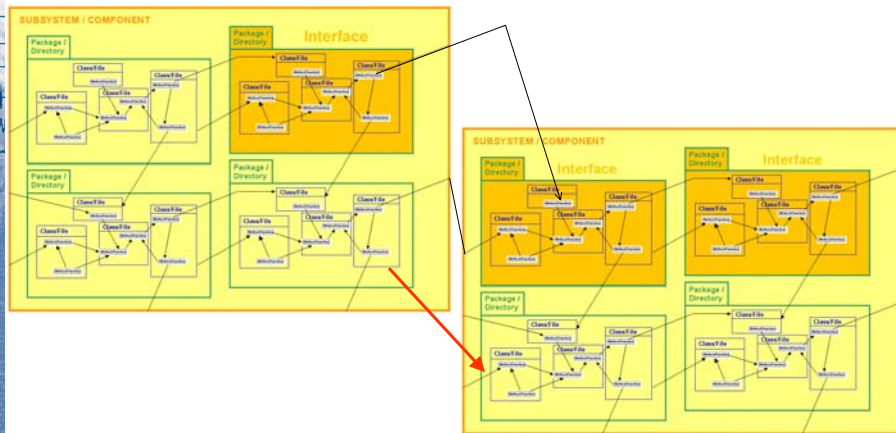
20 Subsysteme

Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

10

Subsystemschnittstellen

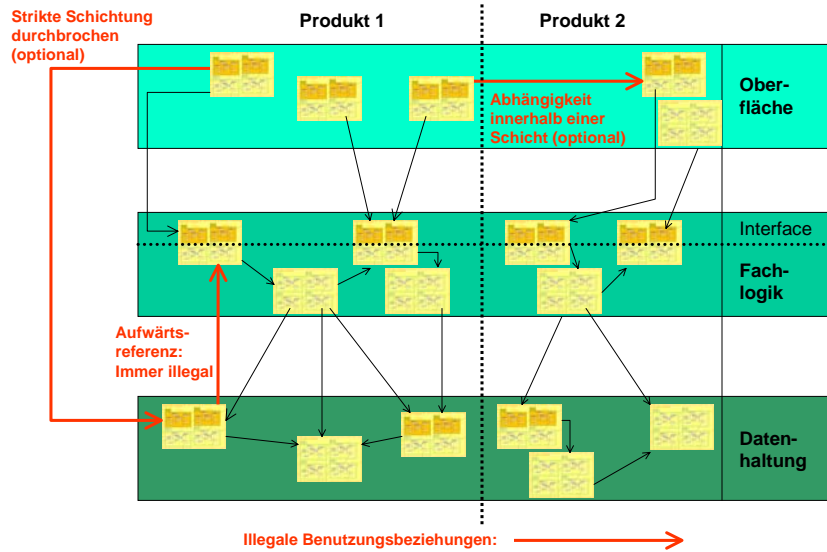


Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

11

Analyse auf Architekturebene



Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

12

Zyklenbasierte Architekturanalyse

- Wieso sind zyklische Beziehungen problematisch?
 - Artefakte die zyklisch gekoppelt sind können nicht einzeln **getestet** werden.
 - Artefakte die in verschiedenen Zyklen gebraucht werden spielen dabei häufig mehrere Rollen, was sie schlecht **verständlich** macht.
 - Artefakte die in verschiedenen Zyklen gebraucht werden können nicht mehr einfach **ausgetauscht** werden.
- Erfahrung in grossen Projekten zeigt dass
 - Zyklen die Wartbarkeit deutlich verschlechtern
 - das Auflösen von Zyklen nicht mehr Wartbare Systeme wieder erweiterbar machen kann

Identifikation von dupliziertem Code

- Erfahrung aus der Praxis zeigt dass
 - Man in mittelgrossen Projekten häufig Kopien von Klassen oder grossen Teilen von Klassen findet
 - sehr grossen Softwaresystemen häufig 70% des Codes aus Kopien besteht
- Auswirkungen
 - Kopien bedeuten, dass bei Änderungen mehrere Stellen berücksichtigt werden müssen. Häufig ist man sich dessen aber nicht bewusst.
 - Kopien, speziell wenn sie noch leicht angepasst sind komplizieren das Testen.

Metrik- und regelbasierte Analyse

- Arten
 - Architekturmetriken
 - Z.B. Stabilitätsmetriken von Robert C. Martin
 - Größen- und Kopplungsmetriken
 - Z.B. Anzahl Klassen pro Package, Anzahl verwendeter Packages
 - Komplexitätsmetriken
 - Z.B. Cyclomatic Complexity
 - Regeln
 - Z.B. eine Klasse darf die von ihr abgeleiteten Klassen nicht kennen
 - Bad Smells
 - Z.B. Flaschenhalse, unbenutzte Artefakte

Erfahrungen aus der Praxis

- Kurzanalysen
 - Ein Tag genügt um den Finger in die offenen Wunden zu legen.
 - Die Entwickler sind sich der Ursachen ihrer Probleme häufig nicht bewusst.
- Software Qualitätsanalyse durch Qualitätsberater
 - Sehr tiefgehende Qualitätsanalysen grosser, komplexer Softwaresysteme zwei Personenwochen Aufwand
 - Expertenwissen notwendig
- Kontinuierliche Qualitätsüberwachung
 - Mit kleinem wöchentlichen Aufwand möglich
 - Prozessveränderungen und Auseinandersetzung mit der Architektur kosten Zeit
 - Sollte früh im Projekt einsetzen
 - Basisanalyse und Einarbeitung: 2-3 Tage

Buchempfehlung

- Beschreibt
 - Architecture-Smells. D.h. problematische Muster auf architekturebene
 - Wie man Architecture-Smells identifiziert
 - Wie man Architecture-Smells ausmerzt



Dr. Walter Bischofberger

Software-Tomography GmbH © 2004

17

Software-Tomography GmbH
Cottbus, München, Zürich

www.software-tomography.com

software
tomography