

# Ansätze des Qualitätsmanagements

Ein Überblick von Prinzipien und Maßnahmen

W3L AG  
info@W3L.de

2011



# Qualität

## ■ Qualitätsbegriff

- „Vermögen einer Gesamtheit inhärenter Merkmale eines Produkts, eines Systems oder eines Prozesses zur Erfüllung von Forderungen von Kunden und anderen interessierten Parteien.“ [DIN EN ISO 9000]
- Qualität ist ein relativer Begriff
  - Je nach Einsatzgebiet gibt es verschiedene Qualitätsvorgaben

## ■ Prozessqualität im Bereich Softwareentwicklung

- Qualität des Software-Entstehungsprozesses sowie der begleitenden Prozesse

## ■ Qualitätsansätze

- Produktbezogener Ansatz (Entwicklung)
  - Qualität wird an Produkteigenschaften festgemacht (Wartbarkeit, Performanz)
- Prozessbezogener Ansatz (Fertigung)
  - Der Fertigungsprozess ist auf die Qualität des Erzeugnisses ausgerichtet (=TDD)
- Kosten/Nutzen-bezogener Ansatz (Finanzen)
  - Qualitätsmaßnahmen verwenden, um später Folgekosten zu minimieren (~BDD)

# Qualitätsmanagement

- **Das Qualitätsmanagement umfasst Maßnahmen, welche der Verbesserung von Produkten, Prozessen oder Leistungen dienen**
- **Disziplinen des Qualitätsmanagements**
  - **Qualitätsplanung: Vorbereitende Maßnahmen**
    - Festlegung von Qualitätsanforderungen an den Prozess und an das Produkt
    - Wichtig: Die Prüfbarkeit muss sichergestellt werden.
  - **Qualitätslenkung: Begleitende Koordinierungsmaßnahmen im Lebenszyklus**
    - Überwachung und Steuerung des Prozesses
    - Beseitigung von Ursachen nicht zufriedenstellender Ergebnisse
  - **Qualitätssicherung: Begleitende analytische Maßnahmen im Lebenszyklus**
    - Erfassung und Überwachung der Qualitätsinformationen (z.B.: Performancemessung)
  - **Qualitätsprüfung: Abschließende Maßnahmen**
    - Auswerten qualitativer und quantitativer Qualitätsinformationen
    - Tests, Reviews, Audits
  - **Qualitätsverbesserung: Prozess-bezogene Maßnahmen**
    - Auswertung der Qualitätssicherungsergebnisse zur Prozessverbesserung

# Betrachtung von Ziele & Erwartungen

- **Das Qualitätsmanagement muss bestimmte Ziele und Erwartungen der verschiedenen Stakeholder beachten**
- **Gruppe 1: Sicht des *Managements***
  - Kosten der Software-Entwicklung (Kostenminimierung)
  - Produktivitätssteigerung
  - Risiken minimieren (Marktposition halten und verbessern, Time-to-Market)
  - Zertifizierung der Qualität (→ Marketing: TQM; ISO 9000)
- **Gruppe 2: Sicht des *Entwicklers***
  - Lesbarkeit & Wartbarkeit (Wiederverwendbarkeit, Erweiterbarkeit)
  - Effizienz und Effektivität
  - Vertrauen (Restfehler, MTBF – „Mean Time Between Failures“)
- **Gruppe 3: Sicht des *Kunden***
  - Abschätzungen (Budgettreue, Termintreue)
  - Qualität (Zuverlässigkeit, Korrektheit)
  - Return on Investment (Anlagenrendite)

# Prinzipien und Regeln #1

## ■ Prinzip der produkt- und prozessabhängigen Qualitätszielbestimmung

- Vorab produkt- und prozessabhängige Qualitätsziele festlegen
  - Maßnahmen für den Entwicklungsprozess und die Qualitätsprüfung werden deutlich

## ■ Prinzip der quantitativen Qualitätssicherung

- Durch die Quantisierung können Soll-Werte und Ist-Wert verglichen werden

## ■ Prinzip der maximalen konstruktiven Qualitätssicherung

- Fehler im Vorfeld durch konstruktive Maßnahmen wie Richtlinien und Checklisten vermeiden.
  - Direkte Verbesserung der Produktivität → Reduktion späterer QS-Maßnahmen

## ■ Prinzip der frühzeitigen Fehlerentdeckung und Fehlerbehebung

- Reduziert Kosten
- Mit höherer Wahrscheinlichkeit werden Fehler richtig korrigiert

# Prinzipien und Regeln #2

## ■ Prinzip der Standardisierung

- Bestehende Lösung wiederverwenden und die vorhandene Qualität erben
  - Vererbte Qualität
- Referenzarchitekturen; Entwurfsmuster; Richtlinien

## ■ Prinzip der Automatisierung → Vermeidung von menschlichen Fehlern

- Generierung von Quellcode; Automatische Tests; Auto-Deployment

## ■ Prinzip der Wissenskonservierung und -erweiterung

- Austausch und Schulungen
- Entwicklungsblogs & Feedback-Runden

## ■ Prinzip der Risikoorientierung

- → Fokussierung auf die Identifizierung von Fehlerursachen
- Balance zwischen Kosten und Qualität muss beachtet werden
  - Spätere Fehlerbehebungen sind stets kostenintensiver

# Maßnahmen #1

## ■ Konstruktive Maßnahmen

- Konstruktive Maßnahmen schaffen eine grundlegende Qualität durch Vorgaben und Richtlinien im Rahmen der Softwareentwicklung
- Methoden
  - Bewährte Konzepte: Entwurfsmuster
    - Ebenfalls relevant: Typische „Anti-Patterns“
  - Goldene Regeln
    - Astonishment: Principle of Least Surprise (Prinzip der geringsten Überraschung)
    - KISS: Keep It Simple Stupid (Möglichst eine einfache Lösung wählen)
    - YAGNI: You Ain't Gonna Need It (Funktionalität erst implementieren, wenn sie benötigt wird.)
    - DRY: Don't Repeat Yourself (Duplikationen vermeiden)
  - Richtlinien und Checklisten
    - Code Conventions / Coding Guidelines
    - Review Checklisten

# Maßnahmen #2

## ■ Konstruktive Maßnahmen

### ■ Methoden

- Mitarbeiterqualität: Wissensverbreitung
  - Schulungen, Workshops, Vorträge, Team-Rotation
  - Dokumentationen (Archivierung von Wissen)
  - Verbesserung des Umfelds und Kommunikation
- Mehraugenprinzip
  - Reviews & Dokumentenfreigabe
  - Pair-Programming
- Defensive Programmierung
  - Selbstprüfungen: Assertions in der Debug-Version
  - Logging und Fehlerfallbehandlungen



# Maßnahmen #3

## ■ Konstruktive Maßnahmen

### ■ Methoden

#### Werkzeuge

- Build-Server (Continuous-Integration)
  - Testautomatisierung: JUnit; MSTest; QUnit
  - Automatisierung von Code-Coverage-Tools (FindBugs; FxCop; JSLint; ... )
- Aufgabenverwaltung (JIRA; TFS)
- Release-Verwaltung (JIRA; TFS)
- Quellcodeverwaltung (SVN; GIT)

#### Produktionsprozesssteuerung

- SCRUM; KANBAN

#### Projektcontrolling

- Den Entwicklungsfortschritt und den QS-Status überwachen
- Überblick und Klarheit verschaffen: IST/Soll-Vergleiche

# Maßnahmen #4

## ■ Analytische Maßnahmen

- Analytische Maßnahmen sind diagnostische Maßnahmen
  - Hierdurch wird keine Qualität geschaffen
  - Es wird eher der Ist-Zustand gemessen
  - Sie können Fehler feststellen

## ■ Methoden

- Testfälle: Testfallspezifikation und Testdurchführungsprotokolle
- Statische Quellcodeanalyse
  - Automatisierte Durchführung einer Konformitätsanalyse des Quellcodes
    - Prüfung von Syntaxrichtlinien
  - Kontroll- und Datenflussanalysen:  $C_0$ ,  $C_1$ ,  $C_2$ ,  $C_3$
- Manuelle Prüfung: Durchführung von Code-Reviews
- Programmverifikation: Schleifeninvariante / Konditionierung & Zusicherungen

# Quantisierung

## ■ Die Quantisierung hilft Qualitätsprobleme zu identifizieren sowie diese messbar und vergleichbar zu machen

### ■ Kundenzufriedenheit

- Fragebögen (AttrakDiff2; UEQ)
- Das Kano-Modell

### ■ Performanz:

- Performancemessung; Stresstest

### ■ Speichereffizienz: Lasttest; Langzeittest

### ■ Wartbarkeit

- Komplexität: McCabe-Metrik (zyklomatische Komplexität)
- Volumen: Lines-of-Code; Halstead-Metrik

### ■ Softwareentwurfs

- Ziel: Kopplung minimieren / Bindung (Kohäsion) maximieren
- CBO: Coupling Between Object Classes
- Tiefe und Breite von Vererbungshierarchien

# Live-Demo von QS-Werkzeugen

## ■ Aufwandsreduktion durch Werkzeugunterstützung

- NDepend – Quellcodemetrik/Quellcodevisualisierung für .NET
- JSLint – Statische Quellcodeanalyse für Javascript
- JMeter – Lasttests für Client/Server-Anwendungen
- Jenkins – Open-Source Continuous-Integration-Server



Zur Live-Demo

# Literatur

- Balzert, Helmut (1997). Lehrbuch der Software-Technik, Bd.2, Software-Management, Software-Qualitätssicherung und Unternehmensmodellierung, DE: Spektrum Verlag
- Hoffmann, Dirk W. (2008). *Software-Qualität*. Heidelberg, DE: Springer.
- Wallmüller, Ernest (2001). *Software-Qualitätsmanagement in der Praxis*, DE: Hanser-Verlag
- DIN ISO 9126 → Qualitätsmerkmalen und Leitfaden zu deren Verwendung
- ISO 9000 + ISO 9001 → Modell zum Qualitätsmanagement und zur Qualitätssicherung

## Inhouse-Schulungen



Wir bieten Inhouse-Schulungen und Beratung durch unsere IT-Experten und -Berater.

### Schulungsthemen

- Softwarearchitektur (OOD)
- Requirements Engineering (OOA)
- Nebenläufige & verteilte Programmierung

Gerne konzipieren wir auch eine individuelle Schulung zu Ihren Fragestellungen.



Sprechen Sie uns an!  
Tel. 0231/61 804-0, [info@W3L.de](mailto:info@W3L.de)

## W3L-Akademie



*Flexibel online lernen und studieren!*

In Zusammenarbeit mit der Fachhochschule Dortmund bieten wir

### zwei Online-Studiengänge

- B.Sc. Web- und Medieninformatik
- B.Sc. Wirtschaftsinformatik

**und 7 Weiterbildungen im IT-Bereich an.**



Besuchen Sie unsere Akademie!  
<http://Akademie.W3L.de>