

Software Engineering

SW Qualitätssicherung

Prof. Dr. Peter Jüttner

Hochschule Deggendorf

Software Qualitätssicherung

5. Methoden

5.1 Requirments Engineering

5.2 Design

5.3 Codierung

5.4 Test

5.5 SW Qualitätssicherung

5.5.1 Messen von Qualität

5.5.2 Review

5.5.2.1 Objekt-Review

5.5.2.2 Meilenstein-Review

Software Qualitätssicherung

Literatur

- Peter Liggesmeyer, Software Qualität, Spektrum Akademischer Verlag
- N. Fenton, S. Peeger. Software Metrics: A Rigorous & Practical Approach, (2nd ed.) International Thomson Computer Press, London, 1996.

Software Qualitätssicherung

Zur Erinnerung

Qualität

**Übereinstimmung mit den
vereinbarten Anforderungen**



Software Qualitätssicherung

Aspekte der Qualitätssicherung

1. **Sicherstellen, dass das richtige Produkt entwickelt wird!**
2. **Sicherstellen, dass das Produkt richtig entwickelt wird!**



Software Qualitätssicherung



Methoden der Software Qualitätssicherung

- „Direkte Methoden“
 - Testen (hier als Bestandteil des Engineering, Kap. 5.4)
 - **Objekt-Reviews** (Inspektionen, Durchsichten)
 - Formale Verifikation (Korrektheitsbeweise, hier nicht weiter behandelt)
- „Indirekte Methoden“
 - Einsatz bestimmter Entwicklungsprozesse
 - Einsatz bestimmter Programmiersprachen
 - Einsatz geeigneter Tools
 - Prozessverbesserung (s. Kap. 4)
 - Projektmanagement, Risikomanagement, Training
 - **Meilenstein-Reviews**

Software Qualitätssicherung



Methoden der Software Qualitätsmessung

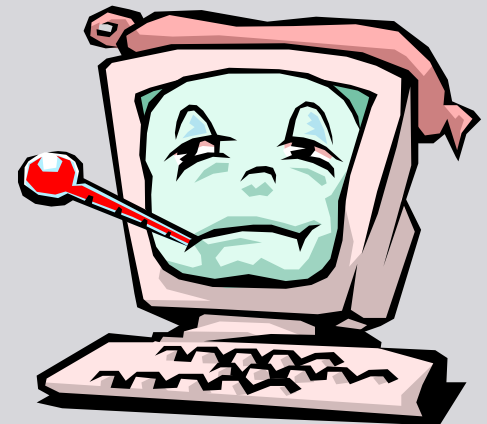
- „Direkte Methoden“
 - Metriken
 - **intern gefundene Fehler**
 - **Fehler vom Kunden gefunden**
 - **Reviewergebnisse**
 - Feedback (subjektiv)
 - Messen bestimmter Entwicklungsergebnisse (z.B. Codemetriken, Testmetriken)
 - Messen anderer Eigenschaften der Software (z.B. Laufzeit, Speicherplatzverbrauch)
- „Indirekte Methoden“
 - Prozessmessung (z.B. CMMI, SPICE)

Software Qualitätssicherung

5.5.1 Messen von SW Qualität

Achtung:

- Das Messen der SW Qualität an sich hat keinen Wert. Die gemessenen Werte müssen geeignet interpretiert werden und es müssen die richtigen Schlüsse gezogen werden!
 - Das Anlegen von Zahlenfriedhöfen macht keinen Sinn!
- ➔ GQM-Ansatz (Goal-Question-Metric)
- ➔ Ziel definieren
 - ➔ Fragen definieren, die Zielerreichung prüfen
 - ➔ Metriken definieren, die Fragen beantworten



Software Qualitätssicherung

5.5.1 Messen von SW Qualität

Fehlermetriken:

- Anzahl gefundener Fehler im Projekt
 - bei Reviews ggf. bezogen auf das gereviewte Dokument
 - beim Testen ggf. bezogen auf die Testphase
 - vom Kunden / Anwender gemeldete Fehler (pro Version)
- falls möglich, Gewichtung der Fehler (critical, high, low)



Software Qualitätssicherung

5.5.1 Messen von SW Qualität

Fehlermetriken:

→ Auswertungen, z.B.

- Anzahl gefundener Fehler zu Anzahl behobener Fehler
→ Qualitätsfortschritt
→ Restfehlerrate
- Anzahl der vom Kunden gefundenen Fehler zu Anzahl intern gefundener Fehler
→ Wirksamkeit des internen Tests
- beim Testen ggf. bezogen auf die Testphase
→ Wirksamkeit der Testphasen



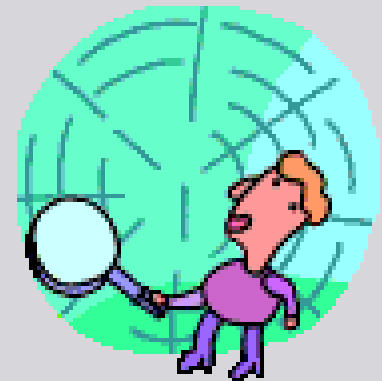
Software Qualitätssicherung

5.5.2.1 Objekt-Review

ANSI-Definition

.... a formal evaluation technique in which software requirements, design, or code are examined in detail by a person or group other than the author to detect faults, violations of development standards, and other problems
(ANSI/IEEE Standard 729-1983)

Begriffe: Review, Inspection, Walkthrough, Document Control



Software Qualitätssicherung

5.5.2.1 Objekt-Review

mögliche Objekte sind alle (!) Ergebnisse der Entwicklung

- Requirements
- Architektur
- Feindesign
- Code
- Testfälle
- Testergebnisse
- Planungsdokumente
- Baselines
- ...

➔ Nicht alle können im Lauf eines Projekts gereviewt werden!

➔ Sinnvolle Auswahl treffen!

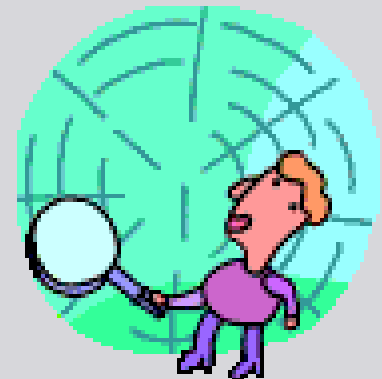


Software Qualitätssicherung

5.5.2.1 Objekt-Review

Ziele

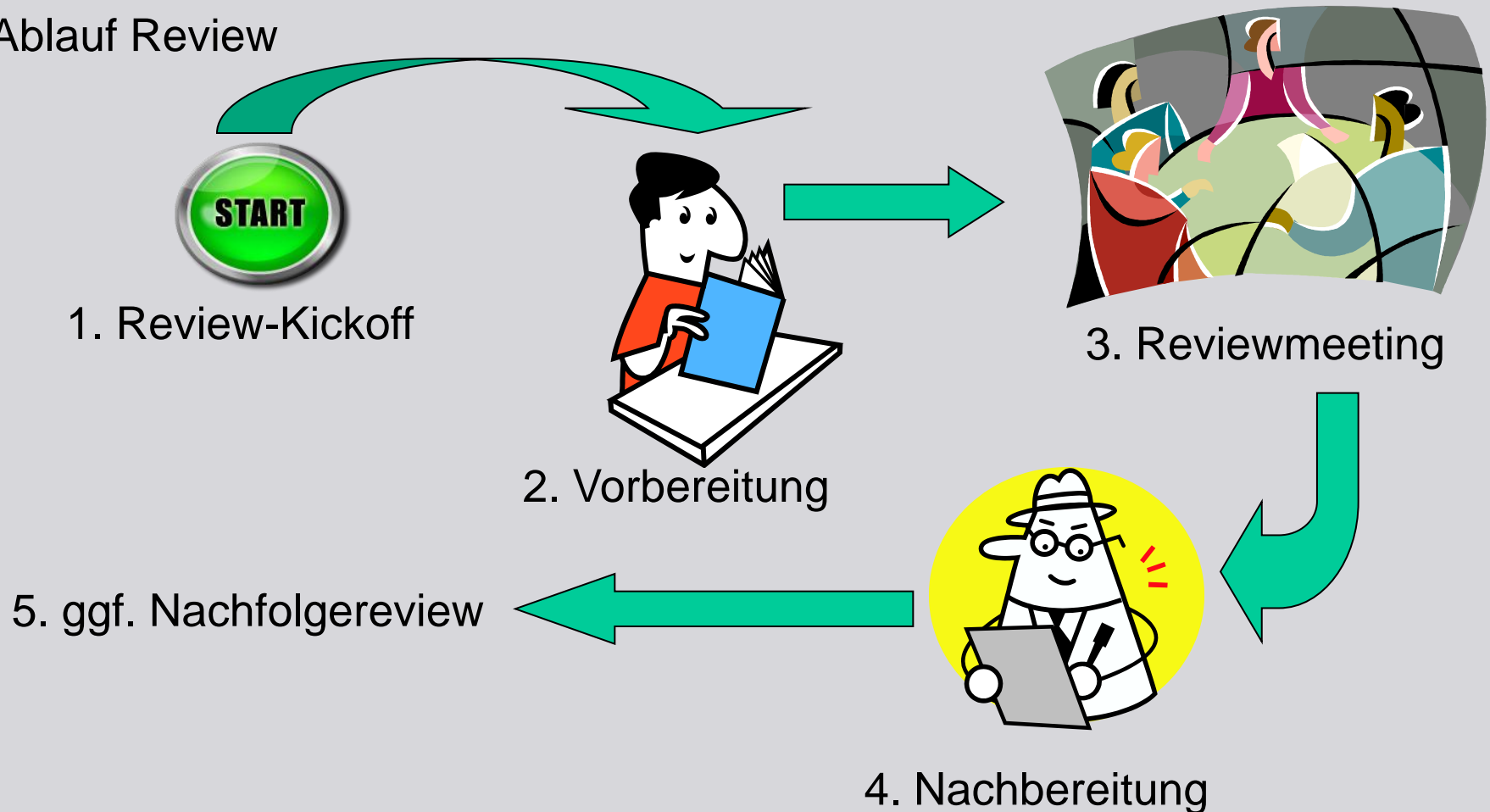
- Fehler (möglichst frühzeitig) finden (auch bevor lauffähiger Code existiert)
- Unklarheiten beseitigen
- Offene Fragen klären
- Wissensverbreitung
- Abstimmung mit dem Kunden
- Erfahrungen sammeln
- ggf. Erfüllen von gesetzlichen Anforderungen



Software Qualitätssicherung

5.5.2.1 Objekt-Review

Ablauf Review



Software Qualitätssicherung



5.5.2.1 Objekt-Review

Review-Kickoff

- Autor definiert Teilnehmer (Reviewer) und Reviewziele
- Autor verteilt Reviewobjekt und verteilt ggf. weitere Unterlagen (weitere Dokumente, Liste für Reviewanmerkungen)
- Autor setzt Termin für Durchsicht des Reviewobjekts (je nach Umfang 1-2 Wochen)
- Autor organisiert Raum für Reviewmeeting
- ggf. wird das Kickoff auch im Rahmen einer Besprechung mit der Reviewern durchgeführt

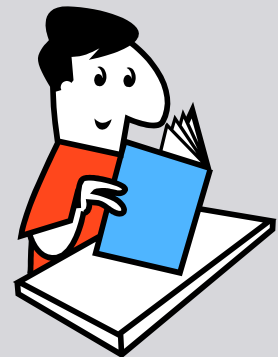
Software Qualitätssicherung



5.5.2.1 Objekt-Review

Review-Vorbereitung

- Reviewer prüfen Reviewobjekt
- Anmerkungen (gekennzeichnet als Fehler, Unklarheiten, Kommentare) werden in einer Liste notiert
- Ggf. unterstützen Checklisten die Vorbereitung
- Ggf. werden Vorbereitungszeit notiert



Software Qualitätssicherung



5.5.2.1 Objekt-Review

Reviewliste

No.	Reference	Comments / Actions	Classification	Responsible person / Planned date for completion	Completion (Name / Date) Optional: Change reference
			page	previous page	total
Sum of remarks					
Sum of errors / risk					

Software Qualitätssicherung

5.5.2.1 Objekt-Review

Review Checklist

No.	Category	Description	OK / NOK / N.R.	Comments / Actions	Respon sible / Planned date	Completi on (Name / Date)
M3.4	Documentati on	Are all uses of <i>#pragma</i> directives explained by comments? Yes => OK				
M3.5	Documentati on	Is there any bit fields access of a larger data type which relies on the way that the bit fields are stored? No => OK <i>For example splitting a word in high and low bytes based on struct unions. Only the usage of standard T_FLAG8, T_FLAG16 is allowed for this purposes.</i>				
M8.12	Declarations and definitions	Do all external array variables have explicit size, by definitions or initialization? Yes => OK <i>External array definitions like T_UBYTE a[]; are not allowed.</i>				
M12.5	Expressions	Are all operands of && or operators expressions primary expressions? Yes => OK (Primary expressions are identifiers, constants, string literals, or parenthesized expressions)				

Software Qualitätssicherung

5.5.2.1 Objekt-Review

Review-Meeting

- Autor und Reviewer treffen sich
- Anmerkungen der Reviewer werden präsentiert und in einer gemeinsamen Liste gesammelt
- Alternative: Jeder Reviewer sendet die Liste der Anmerkungen, kein Reviewmeeting, Autor sammelt Anmerkungen in eine gemeinsame Liste
- Ein Termin, bis zu dem alle Anmerkungen bearbeitet werden, wird festgelegt
- Reviewteam entscheidet, ob ein Nachfolgereview notwendig ist
- Ggf. wird der für das Review geleistete Gesamtaufwand und die Anzahl der Anmerkungen notiert



Software Qualitätssicherung

5.5.2.1 Objekt-Review

Review-Nachbereitung

- Autor entscheidet, welche Anmerkungen wie umgesetzt werden
- Anmerkungen werden im Reviewobjekt umgesetzt
- Die Liste der Anmerkungen wird entsprechend aktualisiert (Protokoll der Umsetzung)
- Das Review Objekt wird nach der Bearbeitung zusammen mit dem Protokoll der Umsetzung an die Reviewer verteilt
- ggf. organisiert der Autor ein Nachfolgereview
- ggf. werden Reviewmetriken (Aufwand, gefundene Probleme) erhoben



Übung

Übungsaufgabe Review

Beschreibung:

Führen Sie ein Review der ausgegebenen Spezifikation, des Designs und des zugehörigen C-Codes schrittweise durch.

1. Schritt: Vorbereitung des Reviews (Kick-Off)
2. Vorbereitung des Reviews (30 min)
3. Reviewmeeting (30 min)
4. Präsentation des Ergebnisses

Nutzen Sie zum Sammeln Ihrer Kommentare die vorbereitete Liste



Software Qualitätssicherung

5.5.2.1 Objekt-Review

Exkurs: Fagan Reviews

- Stark formalisierte Reviews mit genau definierten Rollen und Regeln
- Moderator (sorgt für Einhaltung der Regeln)
- Reviewer (Tester)
- Autor
- Ablauf
 - Kickoff mit Vorstellung des Reviewobjekts
 - Vorbereitungsphase
 - Reviewsitzung wobei ein Reviewer(!) das Reviewobjekt erklärt
 - Alle Reviewer können Anmerkungen machen
- Hauptsächlich für Code Reviews
- Sehr zeitaufwendig, aber auch sehr effektiv

Software Qualitätssicherung



5.5.2.1 Objekt-Review

Exkurs: Fagan Reviews

- Stark formalisierte Reviews mit genau definierten Rollen und Regeln
- Moderator (sorgt für Einhaltung der Regeln)
- Reviewer (Tester)
- Autor
- Ablauf
 - Kickoff mit Vorstellung des Reviewobjekts
 - Vorbereitungsphase
 - Reviewsitzung wobei ein Reviewer(!) das Reviewobjekt erklärt
 - Alle Reviewer können Anmerkungen machen
- Hauptsächlich für Code Reviews
- Sehr zeitaufwendig, aber auch sehr effektiv

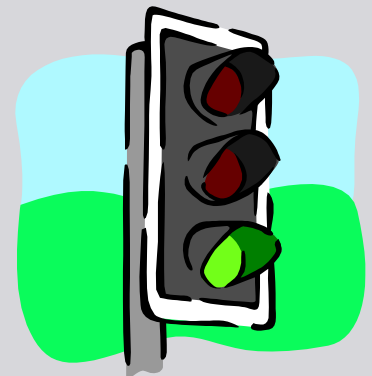


5.5.2.2 Meilenstein-Review

- Im Gegensatz zu Objekt-Reviews wird hier das Projekt zu einem bestimmten Zeitpunkt (Meilenstein) gereviewt
- Ziele: Feststellen, ob die notwendigen Bedingungen zur Erreichung des Meilensteins erfüllt sind bzw. ob die nächste Projektphase gestartet werden kann
- Durchführung im Rahmen eines Meetings
- Reviewteilnehmer: Projektleiter, Projektkernteam, Qualitätssicherung, ggf. Management

5.5.2.2 Meilenstein-Review

- Reviewergebnis Ampel
 - **rot**: Meilenstein nicht erreicht, Projekt benötigt externe Unterstützung (Management)
 - **gelb**: Meilenstein mit Abweichungen erreicht, Projekt kann Abweichungen selbst kompensieren
 - **grün**: Meilenstein ohne Abweichungen erreicht



Software Qualitätssicherung

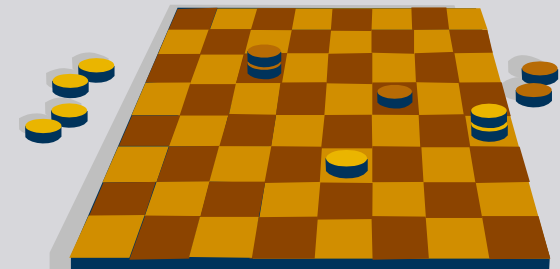
5.5.2.2 Meilenstein-Review

Reviewliste

No.	Questions	OK / NOK / N.R.	Comments / Actions	Responsible / Planned date	Completion (Name / Date)
1	Is the delivery content clearly defined and agreed between the Project Manager and the customer? <i>Delivery content:</i>				
2	Were the requirements reviewed with the customer and frozen for this release?.				
3	Is the Software planning up to date ?				
4	Are all needed resources available?				
5	Was a risk analysis done?				
6	Were functional safety aspects taken into consideration?				

5.5.2 Reviews

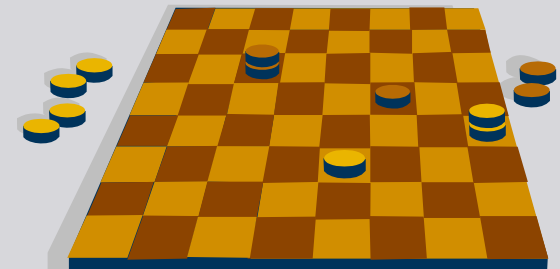
Spielregeln Objektreviews



5.5.2 Reviews

Spielregeln Objektreviews

- Kein Objekt-Review Meeting über 2 Stunden (falls 2 Stunden nicht ausreichen, Pausen machen und/oder auf mehrere Termine verteilen)
- Probleme im Reviewmeeting ansprechen, nicht lösen
- Verständnisdiskussionen im Reviewmeeting sind erlaubt
- Positives und Negatives ansprechen
- Checklisten verwenden
- ggf. Checklisten nach dem Review erweitern



Software Qualitätssicherung

Zum Schluß dieses Abschnitts ...

Noch Fragen ??