

Software-Qualität

Kapitel 5

Reviews und Inspektionen

Probleme finden, nicht nur im Code

Idee und Grundlagen

Prinzip
Rollen
Hilfsmittel

Reviews im Kontext

Nutzen und Kosten
Varianten und Tipps
Forschungsfragen
Differenzierung und Optimierung
Experiment: Perspective-Based Reading

Prof. Dr. Kurt Schneider



Statische Prüfung: Reviews

Grundlagen

Prinzip

Dokument wird von mehreren Personen gemeinsam geprüft

- Dokument: Code oder anderes Dokument
- Definiertes Vorgehen, oft Hilfsmittel wie Formulare
- Mehrere kritische Gutachter
- Gemeinsame Sitzung: Befunde sammeln, bewerten, dokumentieren
- Gemeinsame Entscheidung

Ideen dabei

- Abstand vom Bildschirm, Prüfung auf ausgewählte Aspekte
- Vier-Augen-Prinzip: Vier (bis 10) Augen sehen mehr als zwei

Ziel

- Auch nicht-ausführbare Resultate (z.B. Entwurf, Testplan) prüfen
- Fehler werden nicht behoben, nur fixiert

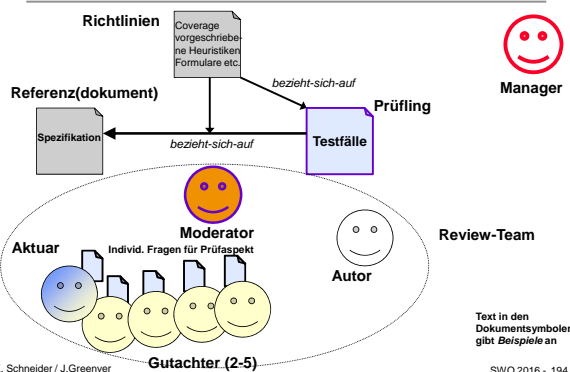
Wichtig

- Aufwand und Nutzen(!) sind erstaunlich hoch

K. Schneider / J. Greenyer
Viele Anregungen aus Frühauf/Ludewig/Sandmayer: Software-Prüfung, Eine Fibel. Vdf, Teubner, 1991
SWQ 2016 - 193

Technische Inspektion (Formales Review)

Rollen

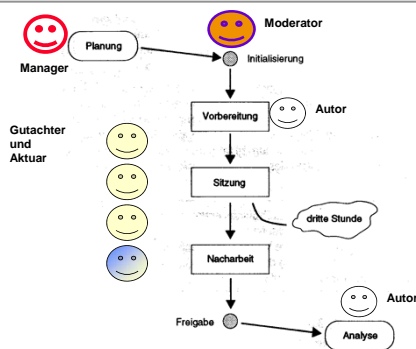


K. Schneider / J. Greenyer

SWQ 2016 - 194

Grober Ablauf einer technischen Inspektion

ähnlich IEEE 1028



Vgl.: Frühauf, K.; Ludewig, J.; Sandmayer, H. (1991):
Software-Prüfung, Eine Fibel. Vdf/Teubner

SWQ 2016 - 195

Prüfung auf ausgewählte Aspekte

Zuweisung der Aspekte zu Gutachtern

Prüfaspekte	Lesbarkeit	Komplexität	Kommentar- verständlichkeit	Ablauf- effizienz
Gutachter				
😊	x		x	x
😊		x		x
😊	x		x	

Regeln: (1) mindestens zwei Gutachter pro Aspekt
(2) meist mehrere Aspekte pro Gutachter
(3) solche Aspekte, die die Gutachter gut beurteilen können

K. Schneider / J. Greenyer

SWQ 2016 - 196



Wichtigste Regeln

Die Review-Sitzung dauert höchstens zwei Stunden

- Danach „dritte Stunde“: offene Diskussion

Moderator kann Sitzung absagen oder abbrechen

- Falls Reviewer unvorbereitet o.ä.

Der Prüfling wird diskutiert, nicht der Autor!

- Sehr wichtig: keine Angriffe, sachlich bleiben

Lösungen, Stilfragen (außer Richtlinien) werden nicht diskutiert

Review-Team formuliert Empfehlung über Prüfling

- akzeptieren ohne Änderung
- akzeptieren mit Änderung, kein weiteres Review
- nicht akzeptieren, weiteres Review erforderlich

K. Schneider / J. Greenyer

SWQ 2016 - 197

SWQ

Tipps für die Beteiligten

Rolle Gutachter (Reviewer, Inspektor)

- **Ziele**
 - Verantwortlich, möglichst viele Befunde (potenzielle Mängel) zu finden.
 - Intensives Studium der Dokumente (Prüfung, Referenz)
 - Bewertung der Kritikalität von Befunden
- **Aufgaben**
 - Genug Zeit einplanen
 - Teilnahme an Vorbereitung u. Sitzung
 - Gewissenhafte Prüfung
 - Dokumentation der Befunde und rechtzeitige Weitergabe
 - Aktive Teilnahme in Sitzung
 - Dauern und Aufwände erfassen
- **Fallen**
 - Fachkenntnisse fehlen: sinnlos
 - Lösungen besprechen
 - Geschmacksfragen diskutieren
 - Autor angreifen, aggressiv formulieren
 - Abschwefeln, verzögern, mit anderen Gutachtern „abdriften“


Rolle des Gutachters

Verlangt vor allem Fachwissen und ein wenig Disziplin.

Jeder sollte sie ausfüllen können.

SWQ 2016 - 199

[illegible]



Checklisten mit Prüfassenkten

Formular

Prüfen Sie bitte das Dokument als Vorbereitung zur Sitzung gemäß den Ihnen Punkt D der Einladung zugewiesenen Aspekten aus der folgenden Liste.

Aspekt 'Form': Ist die Darstellung im Dokument einwandfrei?

- a1 Sind Anforderungen als solche erkennbar, d.h. von Erklärungen unterscheidbar?
- a2 Sind alle Anforderungen eindeutig referenzierbar?
- a3 Ist die Spezifikation jeder Anforderung eindeutig?
- a4 Sind alle Anforderungen überprüfbar formuliert?

Aspekt 'Schnittstellen': Sind alle Schnittstellen eindeutig spezifiziert?

- b1 Sind alle Objekte der Umgebung (Benutzer, andere Systeme, Basis-Software etc.) sowie alle Informationsflüsse von und nach diesen Objekten spezifiziert?
- b2 Sind alle Benutzerklassen (Datenbenutzer, gelegentliche Benutzer, System-Administratoren etc.) des Systems identifiziert?
- b3 Ist die Bedienschnittstelle für jede der Benutzerklassen festgelegt?
- b4 Ist die Bedienphilosophie einheitlich?
- b5 Ist das beschriebene Bedienungskonzept den Vorkenntnissen der Benutzer angemessen?
- b6 Ist die Schnittstelle zur Datenfassung eindeutig festgelegt?

Aspekt 'Zustände und Ereignisse': Sind alle Zustände und Ereignisse eindeutig spezifiziert?

- c1 Sind alle möglichen Betriebsarten (off-line, online, etc.) definiert?
- c2 Sind alle möglichen Betriebsmodi (Betriebs für die verschiedenen Benutzerklassen, Test, Wartung, etc.) definiert?
- c3 Sind alle möglichen Betriebszustände (Normalbetrieb, reduzierter Betrieb, Notbetrieb, etc.) definiert?
- c4 Sind alle Ereignisse spezifiziert, die zu Änderung des Betriebsmodus oder -zustands führen?

Aspekt 'Vertraulichkeit': Sind die wesentlichen Aspekte des Datenschutzes berücksichtigt?

- d1 Ist spezifiziert, welche Information vertraulich zu behandeln ist?
- d2 Sind die Zugriffskriterie für Benutzerklassen definiert?
- d3 Ist definiert, gegen welche Art von unberechtigtem Zugriff die Information geschützt werden soll?

- **Nebenstehend:**
 - Aspekte für Anforderungsdokument
- **Regeln für Checklisten**
 - Mit Ja/Nein beantwortbar
 - „Ja“ ist „Gutfall“
 - „Nein“ ist Problem
 - Präzise fragen
 - Nicht zu viele Fragen
- **Aus Erfahrung lernen**
 - Welche Fragen fanden Fehler?
 - Speziell-Checklisten

Frühauk, P., Ludwig, J., Sandmayr, H. (1991): Software-Prüfung, Eine Brief. VdV/Teubner

K. Schneider / J. Greenyner

[illegible]

Verschiedene Checklisten (Ausschnitte als Beispiel)

Hilfsmittel Formular

Prüfaspekte für Entwurfsdokument

In Klassendiagrammen wurden alle Klassen und ihre Beziehungen erfasst	A1
Für alle in der Spezifikation (Referenz) geforderten Interaktionen sind Sequenzdiagramme vorhanden	A2
Alle Diagramme entsprechen dem UML-Standard, Version 2.0 (Referenz)	A3
Wichtige Konzepte, Algorithmen und Patterns sind verständlich erklärt	A4
Wichtige Schnittstellen sind beschrieben	A5
.....	

Prüfaspekte für Testplan

Für alle Parameter im Entwurf (Referenz) sind Äquivalenzklassen angegeben	T7
Die Wahl der Äquivalenzklassen ist nachvollziehbar begründet	T8
Jede Äquivalenzklasse wird durch mindestens einen Testfall angesprochen	T9
Die Testfälle enthalten konkrete Werte für Testeingaben und Sollresultate	T10
Die Sollwerte stimmen mit den Angaben aus Spezifikation (Referenz) überein oder sind daraus korrekt errechnet	T11

K. Schneider / J.Greenyber

SWQ 2016 - 207

Exkurs: Fehlerkosten(t)

Frühau, K.; Ludewig, J.; Sandmayer, H. (1991): Software-Prüfung. Eine Fibel. Vdf/Teubner

Relative Fehlerbehebungskosten

Fehlerherkunft

- Codierfehler
- Entwurfsfehler
- Anforderungsfehler

Kosten steigen exponentiell

Testphase	Codierfehler	Entwurfsfehler	Anforderungsfehler
Review der Anforderungen	~0.5	~0.5	~1.0
Review des Entwurfs	~0.5	~1.0	~1.5
Code Review	~1.0	~1.5	~4.0
Strukturtest	~1.5	~4.0	~14.0
Funktionstest	~4.0	~9.0	~24.0

Aufwand und Nutzen

Aufwand	Dokumente	Code	Abschätzung des Aufwands
Erstellungsaufwand relativ	2 Seiten/Tag	60 Zeilen/Tag	5 Gutachter * 5h=25Ph (Mod, Autor+5Gut.)=7P Sitzung: 7P*2h=14Ph
Maximaler Umfang für Review	50 Seiten	20 Seiten	
Erstellungsaufwand absolut	25 Personentage	20 Personentage	Analog mit 3 Gutachtern: 3P*4h (bei Code reichen die)
Review-Vorbereitung relativ	10 Seiten/Std.	5 Seiten/Std.	
Aufwand Review-Vorbereitung	25 Stunden	12 Stunden	
Aufwand Review-Sitzung absolut	14 Stunden	10 Stunden	
Summe Review-Aufwand	5 Personentage	3 Personentage	
Review- zu Erstellungsaufwand	20%	15%	

Frühaut, K., Ludewig, J., Sandmayer, H. (1991): Software-Prüfung: Eine Fibel. Vdt/Teubner

Nutzen

- 60-70% der Fehler gefunden
- Gefundene verweilen viel kürzer
- Kosten steigen exponentiell(t)
- ROI deutlich größer als 1

Nutzen-Nachweis ist wichtig.
Um dieses Prinzip **konkret** anzuwenden, braucht man Fehlerzahlen und -klassen

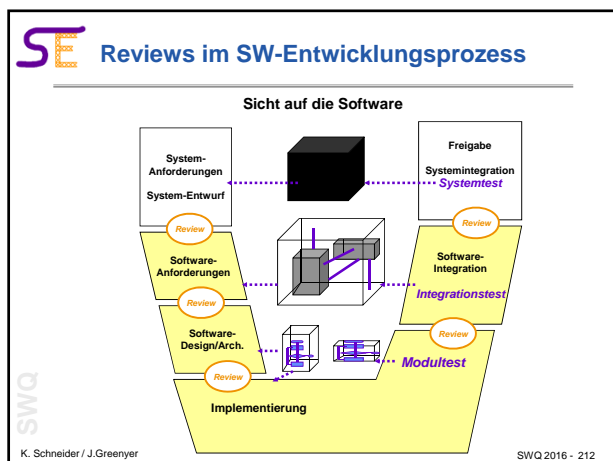
K. Schneider / J.Greenyer SWQ 2016 - 210

Was wird eigentlich gereviewt?

Näher hingesehen

- Was **kann** gereviewt werden?
 - Schriftlich fixierte Dokumente
 - Dokumente, zu denen es eine Referenz gibt
 - Dokumente, die Menschen lesen können (kaum: Object-Code)
- Vorgaben, was gereviewt werden **soll**
 - Allgemeiner SW-Entwicklungsprozess
 - Vorgaben im Wasserfallmodell nach Royce
 - Vorgaben im V-Modell (und firmenspezifischen)
- Was sind die Vor- und Nachteile von Reviews?
 - Aufwands-/Nutzen-Betrachtungen
- Was sollte **mit Vorteil** gereviewt werden?
 - Eine pragmatische Entscheidung

K. Schneider / J.Greenyer SWQ 2016 - 211



Reviews im V-Modell 97

viele firmeneigene Modelle sind ähnlich

Object to be Assessed	Method
User Requirements	REV + T** + STAT* + SIMU****
Technical Requirements	REV + T** + STAT* + SIMU****
System Architecture	REV + T** + STAT*
SW Architecture	REV + T** + STAT*
SW Design	REV + T** + STAT*
Implementation Documents	REV + STAT*
SW Module, Database, SW Component, SW Unit, Segment, System	T + SIMU****
Integration Plan	REV
Operational Information (Information for the User Manual, Information for the Diagnosis Manual, Information for the Operator Manual, Other Application Information)	REV + T
Data Dictionary	REV + STAT*
Interface Overview	REV + STAT*
Interface Description	REV + STAT*
Assessment Plan	REV
Assessment Specification	REV
Assessment Procedure	REV + T****
CM Plan	REV
Configuration Identification (Document System, SW, HW)	REV
Change Order	REV
Change Memo	REV

Prüfmethode

- REV: Review
- T: test
- STAT: statische Analyse

K. Schneider / J.Greenyer SWQ 2016 - 213

Häufige Fehler

- Gutachter**
 - Kommt nicht zum Kickoff/zur Sitzung
 - Unvorbereitet, findet nur Schreibfehler
- Management**
 - kürzt Zeit, Ressourcen
 - Kaum Vorbereitungszeit
 - Prüfung zu lang
- Moderator**
 - Keine Prüfaspkte verteilt
 - Dokumente als Word.doc verschickt
 - „Review“-Regeln nicht erklärt
 - Keine Zeit für Nacharbeit eingeplant
- Psychologisches**
 - Vorgesetzter im Review dabei
 - Kosten ohne Nutzen werden kolportiert

K. Schneider / J.Greenyer SWQ 2016 - 214

Spielarten von „Reviews“

Varianten

- Schreibtischtest**
 - Der Entwickler allein „zieht sich zurück“
 - versucht, Dokument fern vom Bildschirm „neu zu sehen“.
 - Dies ist Voraussetzung, nicht Ersatz für ein formaleres Review
- Stellungnahme**
 - Weiter verbreitet: Autor bittet andere („Experten“) um ihr Urteil
 - Einzelurteile gehen ein, Autor verwendet sie
 - Vorteil:**
 - einfach, scheinbar billig, „passt immer noch rein“
 - Nachteile:**
 - Oft Inkonsistenzen durch verschiedene Versionen: Kommentare wertlos
 - unklar, was mit den Stellungnahmen geschieht (kein Plan)
 - unklar, wie mit widersprüchlichen Kommentaren umzugehen ist

K. Schneider / J.Greenyer SWQ 2016 - 215




Weitere Varianten von Reviews

- **Walkthrough** Wichtig: Unbedingt mitprotokollieren!
 - Autor stellt Dokument vor, Experten hören zu und geben Feedback
 - **Vorteil:** Vorbereitung nur für Autor, Experten „kommen einfach dazu“
 - **Nachteil:** ad hoc Feedback bleibt oberflächlich; weniger tiefe Befunde
- **Beispiel für Bewährte Abstufung (RIT = review and inspection technique)**
- **Ziel:** so gut wie möglich prüfen – in der Realität möglich
 - RIT-4: vollständige technische Inspektion mit allen Regeln
 - RIT-3: mehrere Gutachter mit Vorbereitung. Abstimmung informell
 - RIT-2: min. zwei Gutachter, kaum Abstimmung (z.B. Walkthrough)
 - RIT-1: ein Gutachter, keine Abstimmung nötig
 - RIT-0: keine Prüfung
- **Bereichs- oder projektweite Anpassung des Prüfplans**
 - Vorgabe im Qualitätsplan: viele Reviews
 - Alle „vorgeschriebenen“ Reviews werden hinterfragt: RIT-4 bis -0
 - Kunst des Möglichen („wie wenig ist gerade noch sinnvoll?“)



K. Schneider / J. Greenyer


SWQ 2016 - 216



Testen vs. Reviews

Vor- und Nachteile

Vorteile von Test	Nachteile von Test
<ul style="list-style-type: none">• Reproduzierbar• Mehrfach nutzbar<ul style="list-style-type: none">– Rechenzeit ist billig• Umgebung wird mitgeprüft<ul style="list-style-type: none">– Bibliotheken– Virtuelle Maschine• Systemverhalten veranschaulicht• Falls automatisiert: schnell und billig	<ul style="list-style-type: none">• Bedeutung „fehlerloser“ Tests wird überschätzt („schlecht gesucht“)• Nicht alle Eigenschaften von Software sind testbar („Lesbarkeit“ ?)• Nicht alle Situationen reproduzierbar• Test zeigt Fehlerursache nicht• Falls <i>nicht</i> automatisiert: Aufwändig, insbesondere Wiederholung



K. Schneider / J. Greenyer

SWQ 2016 - 217