

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**

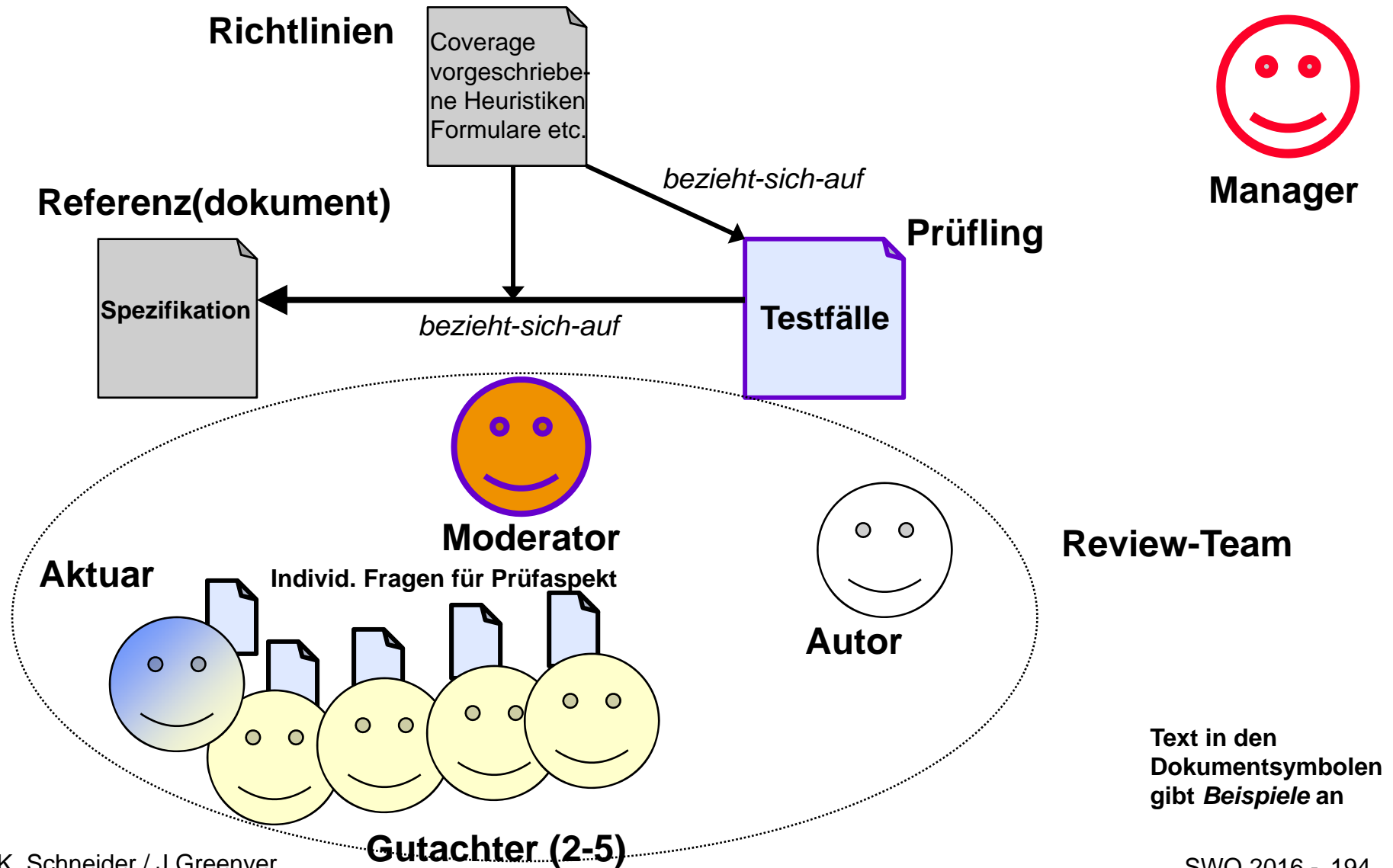
- Auch nicht-ausführbare Resultate (z.B. Entwurf, Testplan) **prüfen**
- Fehler werden nicht behoben, nur fixiert

- **Wichtig**

- **Aufwand und Nutzen(!) sind erstaunlich hoch**

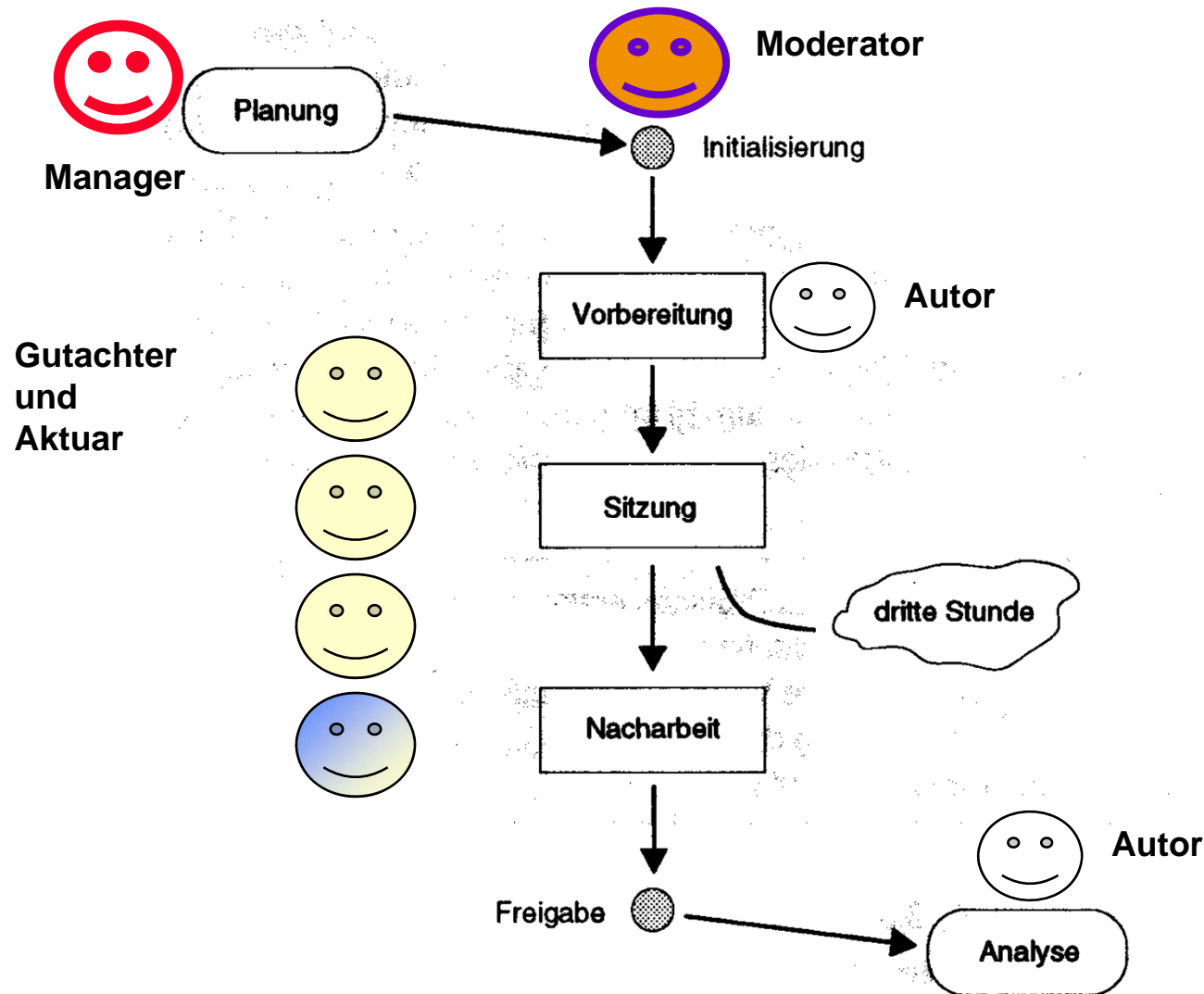
Technische Inspektion (Formales Review)

Rollen





Grober Ablauf einer technischen Inspektion

ähnlich IEEE 1028



Prüfung auf ausgewählte Aspekte

Zuweisung der Aspekte zu Gutachtern

| Prüfaspekte Gutachter | Lesbarkeit | Komplexität | Kommentar- verständlich- keit | Ablauf- effizienz |
|---|-------------------|--------------------|--|------------------------------|
|  | 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



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



• Ziele

- Verantwortlich für Qualität des Reviews, nicht des Produkts
- Beschafft Dokumente und Referenzen, motiviert und leitet ein
- Guter Ablauf durch Moderation und Regel-Einhaltung
- Daten über Review erheben

• Aufgaben

- Planung und Org. des Ablaufs
- Kickoff/Einführung moderieren
- Material (Prüfling, Referenzen) beschaffen und verteilen
- Durchführung der Sitzung
 - Einzelprüfresultate einsammeln
 - Aktiv moderieren; Regeln!
- Nacharbeit überwachen
- Abschluss dokumentieren

• Fallen

- Alles unkritisch ohne Fokus reviewen
- Nutzen und Aufwand nicht explizit erfassen und kommunizieren
- Ressourcen nicht reservieren
- Einarbeitungs- und Lernzeit unterschätzen
- Dokumente nicht/unklar verteilt

Moderator

ist eine schwierige Rolle und verlangt viele verschiedene Fähigkeiten:

- technisches Wissen
- Moderationsfähigkeiten
- soziale u. kommunikative Fähigk.
- Fähigkeit zu motivieren
- Neutralität
- Akzeptanz bei den Teilnehmern



• Ziele

- Verantwortlich, möglichst viele Befunde (potenzielle Mängel) zu finden.
- Intensives Studium der Dokumente (Prüfling, 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
- Abschweifen, verzögern, mit anderen Gutachtern „abdriften“

Rolle des Gutachters

Verlangt vor allem Fachwissen
und ein wenig Disziplin.

Jeder sollte sie ausfüllen können.



• Ziele

- Verantwortlich für (gemeinsames) Reviewprotokoll
- Zügig Beiträge vollständig erfassen
- Diskussion nicht stören, Endergebnis aufnehmen

• Aufgaben

- Formular Reviewprot. besorgen
- Projektor oder Beamer besorgen
- Prüfling lesen, um Beiträge besser zu verstehen
- Unaufdringlich „in Echtzeit“ mitschreiben, Korrekturen aufnehmen
- Änderungswünsche sofort aufgreifen
- Evtl. nachher schön abschreiben

• Fallen und Fehler

- Unvorbereitet: kann nicht schnell genug folgen
- Lückenhafte, zu stark abgekürzte oder unleserliche Mitschrift
- Mitschrift ist nicht sofort für alle sichtbar
- Übernimmt Moderation

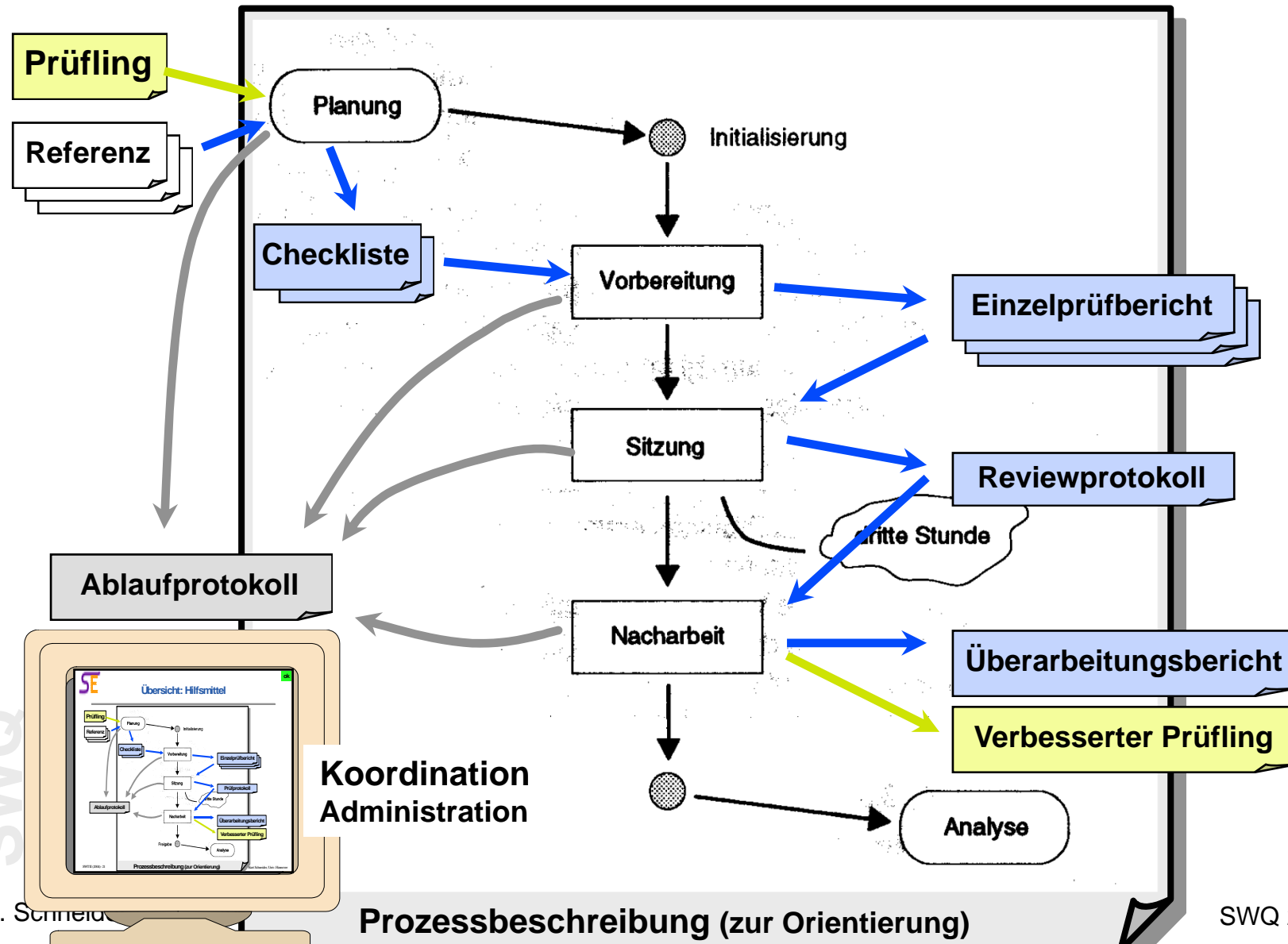
Rolle des Aktuars

Verlangt mehr als ein normales Protokoll.

Feinfühligkeit bei der schnellen Aufnahme und Korrektur ist wichtig

Macht oft ein Gutachter nebenher mit

Übersicht: Hilfsmittel





Formular „Einzelprüfbericht“

Hilfsmittel: Formular

| <Projekt- Name/Logo> | | Einzelprüfbericht | | |
|--|----------------------|---|---------|--------------------------------|
| | | Gutachter: _____ | | Prüfling: _____ |
| Anzahl kritischer Befunde <input type="text"/> | | Ankreuzen, was zutrifft: | | Datum <input type="text"/> |
| Anzahl nicht-kritischer Befunde <input type="text"/> | | <input type="checkbox"/> Prüfling vollständig inspiziert <input type="checkbox"/> Alle Befunde sind hier aufgelistet | | Seite von <input type="text"/> |
| Nr. | Befund: Beschreibung | Position | Checkl. | Kritikalität |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |



Checklisten mit Prüfaspekten

Formular

- **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
 - Spezial-Checklisten

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

K. Schneider / J.Greenyer

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

Aspekt "Form": Ist die Darstellung im Dokument sinnvoll?

- 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 (Dauerbenutzer, gelegentliche Benutzer, System-Administrator, etc.) des Systems identifiziert?
- b3 Ist die Bedienschnittstelle für jede der Benutzerklassen festgelegt?
- b4 Ist die Bedienphilosophie einheitlich?
- b5 Ist das beschriebene Bedienkonzept den Vorkenntnissen der Benutzer angemessen?
- b6 Ist die Schnittstelle zur Datenerfassung eindeutig festgelegt?
- b7 Sind Vorgaben gemacht bezüglich Verwendung von Betriebssystem-Funktionen, Bibliotheken und Hilfsprogrammen?

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 (Betrieb 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 zur 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 Zugriffsrechte aller Benutzerklassen definiert?
- d3 Ist definiert, gegen welche Art von unberechtigtem Zugriff die Information geschützt werden muß?

Formular „Einzelprüfbericht“

Hilfsmittel: Formular, ausgefüllt am Beispiel GameFrame

Einzelprüfbericht

Gutachter: K.Schneider
Prüfling: GameFrame-Spez-GruppeFiktiv-v07

Anzahl kritischer Befunde

6

Anzahl nicht-kritischer Befunde

34

Ankreuzen, was zutrifft:

☒
Prüfling vollständig inspiziert

☒
Alle Befunde sind hier aufgelistet

Datum

16.4.05

Seite Von

1(3)

| Nr. | Befund: Beschreibung | Position | Checkl. | Kritikalität |
|--------------------------|--|----------|---------|--------------|
| 1 | Anforderung oder Wunsch? „sollte auch der Spieler...“ | 4, Z13 | a1 | kritisch |
| 2 | Drei Anforderungen in zu langem Satz vermischt | 5, Z2-7 | a2 | unkritisch |
| 3 | Schreibfehler: GaimeFrame | 5, Z30 | s | unkritisch |
| 4 | Nur schwer prüfbar: „Entwickler werden rasch ‚süchtig‘ nach ...“ | 7, Z23f | a4 | unkritisch |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| <div>und so weiter</div> | | | | |



Formular „Reviewprotokoll“

Hilfsmittel: Formular (vor Sitzung vorbereiten)

| <Projekt-Name/Logo> | | Protokoll der Review-Sitzung | | | |
|---------------------|-----------|------------------------------|---|---------|--------------|
| | | Ort, Zeit (von-bis): _____ | | | |
| Prüfling: _____ | | Protokoll: _____ | | | |
| Moderator | Gutachter | | Entscheidung | Datum | |
| | | | <input type="checkbox"/> Akzeptiert | Seite | |
| | | | <input type="checkbox"/> Akzeptiert mit Änderungen | von | |
| | | | <input type="checkbox"/> Überarbeiten, neues Review | | |
| Nr. | von | Befund: Beschreibung | Position | Checkl. | Kritikalität |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |



Formular „Reviewprotokoll“

Hilfsmittel: Formular mit *Beispielen aus GameFrame*

| GameFrame | | | | | |
|---|-----|---|----------|--|-------------------------------|
| Protokoll der Review-Sitzung | | | | | |
| Ort, Zeit (von-bis): MMHörs., 18.4.2005 (13:00-15:30) | | | | | |
| Prüfling: <i>GameFrame-Spez-GruppeFiktiv-v07</i> Protokoll: <i>D.Scribe</i> | | | | | |
| Moderator <i>J.Körner</i> | | Gutachter <i>TF: T.Flohr</i> <i>KS: K.Schneider</i> | | XY: <i>X. Ysanti</i> | |
| | | | | Entscheidung <input type="checkbox"/> Akzeptiert <input checked="" type="checkbox"/> Akzeptiert mit Änderungen Überarbeiten, neues Review | |
| | | | | Datum Seite von | <i>19.4.05</i> <i>1(4)</i> |
| Nr. | von | Befund: Beschreibung | Position | Checkl. | Kritikalität |
| 1 | KS | Anforderung oder Wunsch? „sollte auch der Spieler...“ | 4, Z13 | a1 | kritisch |
| 2 | TF | Unklar: „sollte auch der Spieler entscheiden können“ | 4, Z13f | a3 | kritisch |
| 3 | KS | Drei Anforderungen in zu langem Satz vermischt | 5, Z2-7 | a2 | kritisch |
| 4 | KS | Schreibfehler: GaimeFrame | 5, Z30 | s | unkritisch |
| 5 | XY | Unklar, Prüfung?: „Oft werden Entwickler dieses Feature...“ | 6, Z21f | a3,a4 | kritisch |
| 6 | TF | Anf. im Bild versteckt: leicht zu übersehen | 6,Abb.2 | a2 | unkritisch |
| 7 | KS | Schwer prüfbar: „Entwickler werden rasch ‚süchtig‘ nach “ | 7. Z23f | a4 | unkritisch |
| | | | | | |
| | | <i>und so weiter</i> | | | |

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 |

Ablaufprotokoll

Hilfsmittel: Formular oder Inhaltsverzeichnis

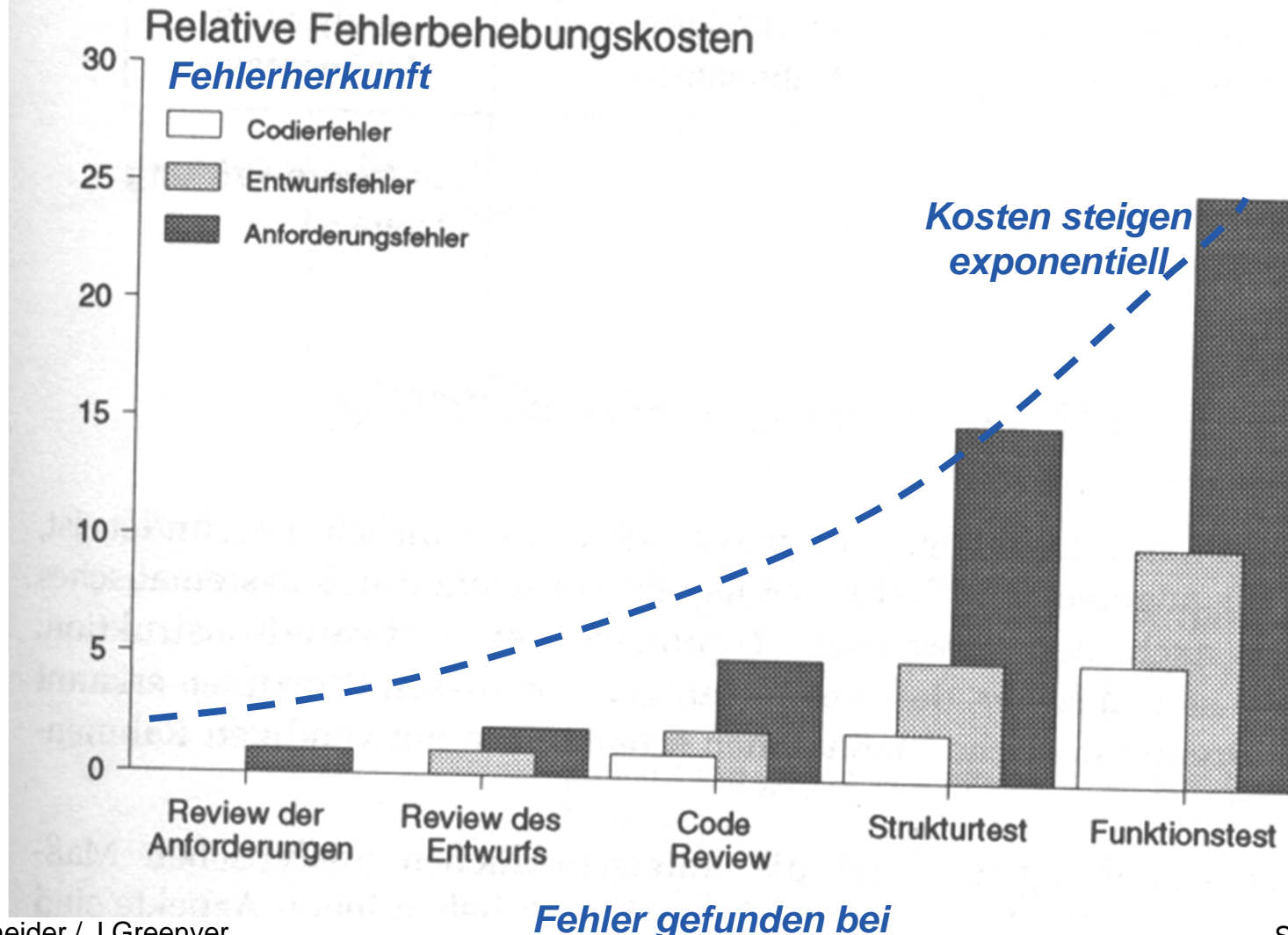
**Ziel: Alle Schritte
des Review-Prozesses
planen, anleiten und verfolgen**

Formular mit Checklisten-Teilen
Oft mehrseitiges Dokument,
das fortgeschrieben wird

- **Planung**
 - Geplante Teilnehmer
 - Prüfling, Referenzen, Hilfsmittel
 - Wann sind Sitzungen geplant?
 - Checkfragen
 - Eingangskriterien erfüllt?
 - Referenz vorhanden?
- **Einführung/Kick-off**
 - Alle anwesend?
 - Aufwand (Dauer*Personenzahl)
 - Checkfragen: U.a. Aufgaben klar?
- **Sitzung**
 - Aufwand, Fehlerzahl pro Anwesendem
 - Kommentare zum Ablauf
 - Aufwand (Dauer*#Personen)
 - Nutzen
 - Fehler Anzahl und verm. Kosten?
 - Checkfragen
 - Gutachter vorbereitet?
 - Einzelprüfberichte eingesammelt?
 - Alle Befunde angesprochen?
 - Alles in Prüfprotokoll erfasst?
- **Ähnlich: Überarbeitung**
 - Alle Befunde berücksichtigt?
 - Dauer und Aufwand?
- **Auswertung: Aufwand/Nutzen**

Exkurs: Fehlerkosten(t)

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



• Aufwand

| | Dokumente | Code |
|--------------------------------|-----------------|-----------------|
| Erstellungsaufwand relativ | 2 Seiten/Tag | 60 Zeilen/Tag |
| Maximaler Umfang für Review | 50 Seiten | 20 Seiten |
| Erstellungsaufwand absolut | 25 Personentage | 20 Personentage |
| 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 % |

Abschätzung
des Aufwands

5 Gutachter * 5h=25Ph
(Mod, Autor+5Gut.)=7P
Sitzung: 7P*2h=14Ph

Analog mit
3 Gutachtern: 3P*4h
(bei Code reichen die)

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



Was wird eigentlich gereviewt?

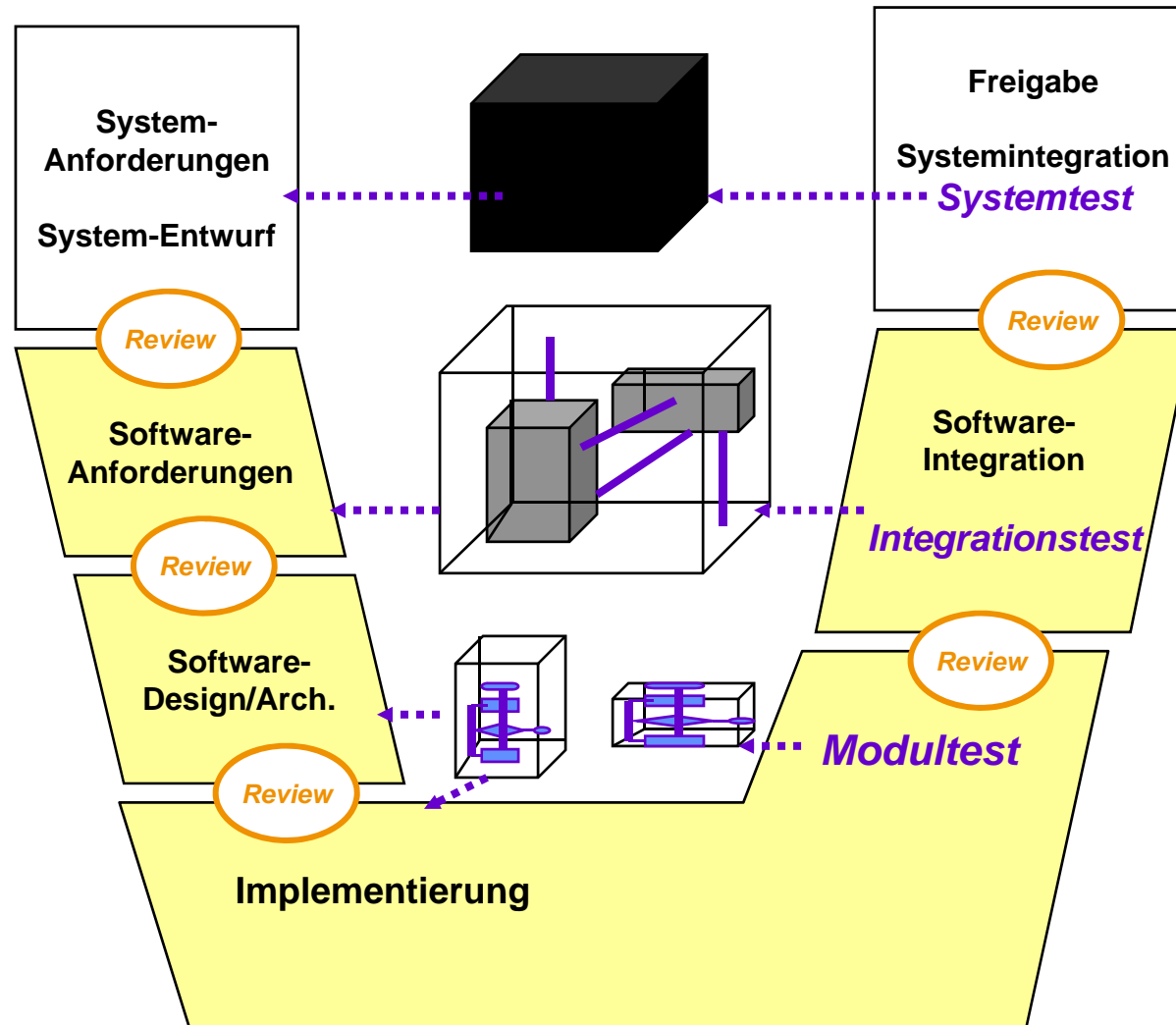
Näher hingesehen

- 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



Reviews im SW-Entwicklungsprozess

Sicht auf die Software





Reviews im V-Modell 97

viele firmeneigene Modelle sind ähnlich

**Dokumente
in der
SW-Entwicklung**

| 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**

SWQ



Häufige Fehler

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



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



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?“)



Testen vs. Reviews

Vor- und Nachteile

Vorteile von Test

- Reproduzierbar
- Mehrfach nutzbar
 - Rechenzeit ist billig
- Umgebung wird mitgeprüft
 - Bibliotheken
 - Virtuelle Maschine
- Systemverhalten veranschaulicht
- Falls automatisiert: schnell und billig

Nachteile von Test

- Bedeutung „fehlerloser“ Tests wird überschätzt („schlecht gesucht“)
- Nicht alle Eigenschaften von Software sind testbar („Lesbarkeit“ ?)
- Nicht alle Situationen reproduzierbar
- Test zeigt Fehler*ursache* nicht
- Falls *nicht* automatisiert: Aufwändig, insbesondere Wiederholung