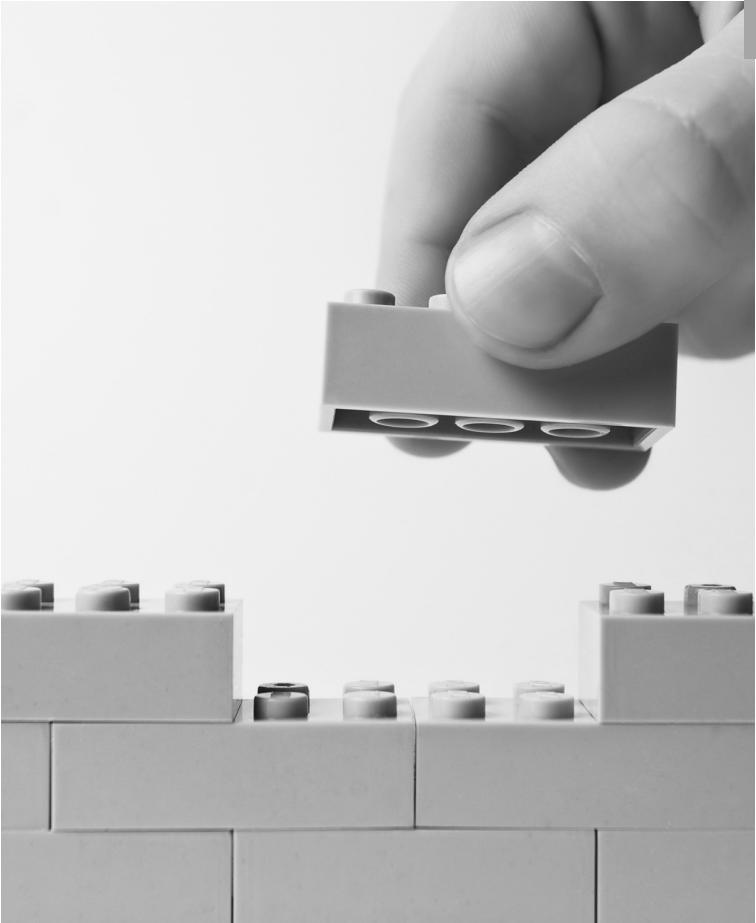


4 – Statischer Test

1. Statischer Test – Grundlagen
2. Feedback- und Reviewprozess

Statischer Test – Grundlagen



4.1 Statischer Test – Grundlagen

- FL-3.1.1 (K1) Sie können Arten von Arbeitsergebnissen, die durch statischen Test geprüft werden können, erkennen
- FL-3.1.2 (K2) Sie können den Wert statischer Tests erklären
- FL-3.1.3 (K2) Sie können statischen Test und dynamischen Test vergleichen und gegenüberstellen

Statischer Test – Grundlagen

Statische Tests prüfen und bewerten Arbeitsergebnisse (= Dokumente oder Code).

- **Reviews** – manuelle Prüfung von Arbeitsergebnissen
- **Statische Analyse** – automatisierte Prüfung von Code, Prozessspezifikation, Systemarchitektspezifikation oder anderer Arbeitsergebnisse (d.h. mit Hilfe eines Werkzeugs)

Das Testobjekt/Programm wird dabei nicht ausgeführt.

Testziele beim statischen Test

- Verbesserung der Qualität
- Fehlerzustände aufdecken
- Merkmale bewerten, z. B. **Lesbarkeit, Vollständigkeit, Korrektheit, Testbarkeit und Konsistenz**

Statische Tests können zu Verifizierung und Validierung eingesetzt werden.

Statischer Test – Zusammenarbeit

Tester, Fachbereichsvertreter (Product Owner, Businessanalysten etc.) und Entwickler arbeiten zusammen, beim

- Example-Mapping
- schreiben von User-Storys
- verfeinern des Backlogs (Refinement)

um sicherzustellen, dass die User-Storys und die zugehörigen Arbeitsergebnisse definierten Kriterien entsprechen, z. B. der Definition-of-Ready.

Reviewverfahren können angewendet werden, um sicherzustellen, dass User-Storys ...

- vollständig und verständlich sind
- testbare Akzeptanzkriterien enthalten.

Indem Tester die richtigen Fragen stellen können sie die vorgeschlagenen User-Storys analysieren, hinterfragen und verbessern.

Statische Analyse

Fehlerliste

0 Fehler | 5 Warnungen | 0 Nachrichten

| Beschreibung | |
|--------------|--|
| ! | 1 Die Variable 'wichtig_nicht_vergessen' wurde definiert, aber nicht verwendet |
| ! | 2 Der Variable 'ergebnis' wurde ein Wert zugewiesen, dieser aber nicht genutzt |
| ! | 3 Dieser Programmteil kann nicht erreicht werden |
| ! | 4 Gefahr einer Endlosschleife |
| ! | 5 Implizite Konvertierung ändert Vorzeichenmöglichkeit |

 Fehlerliste |  Ausgabe

Statische Analyse

- kann Probleme vor dem dynamischen Testen aufdecken
- weniger Aufwand, da keine Testfälle erforderlich sind
- verwendet in der Regel Werkzeuge
- ist häufig in CI-Frameworks integriert

Einsatz von statischer Analyse

- Erkennung spezifischer Fehlerzustände im Code
- Bewertung der Wartbarkeit und IT-Sicherheit
- Prüfung von Rechtschreibung und Lesbarkeit

Arbeitsergebnisse, die durch statische Tests untersucht werden können

Arbeitsergebnisse, die durch statische Tests untersucht werden können

Fast jedes Arbeitsergebnis ...

- kann mit statischen Tests untersucht werden.
- das gelesen und verstanden werden kann, kann Gegenstand eines Reviews sein.

Für die statische Analyse benötigen Arbeitsergebnisse jedoch eine Struktur, anhand derer sie überprüft werden können.

Beispiele für Arbeitsergebnisse, die durch statische Tests untersucht werden können...

- Spezifikation der Anforderungen
- Testkonzepte, Testfälle, Test-Charts
- Produkt-Backlog-Einträge
- Projektdokumentation
- Verträge
- Modelle
- Code
- Text mit einer formalen Syntax



Arbeitsergebnisse, die sich **nicht** für statische Tests eignen

Zu den Arbeitsergebnissen, die sich nicht für statische Tests eignen, gehören solche, ...

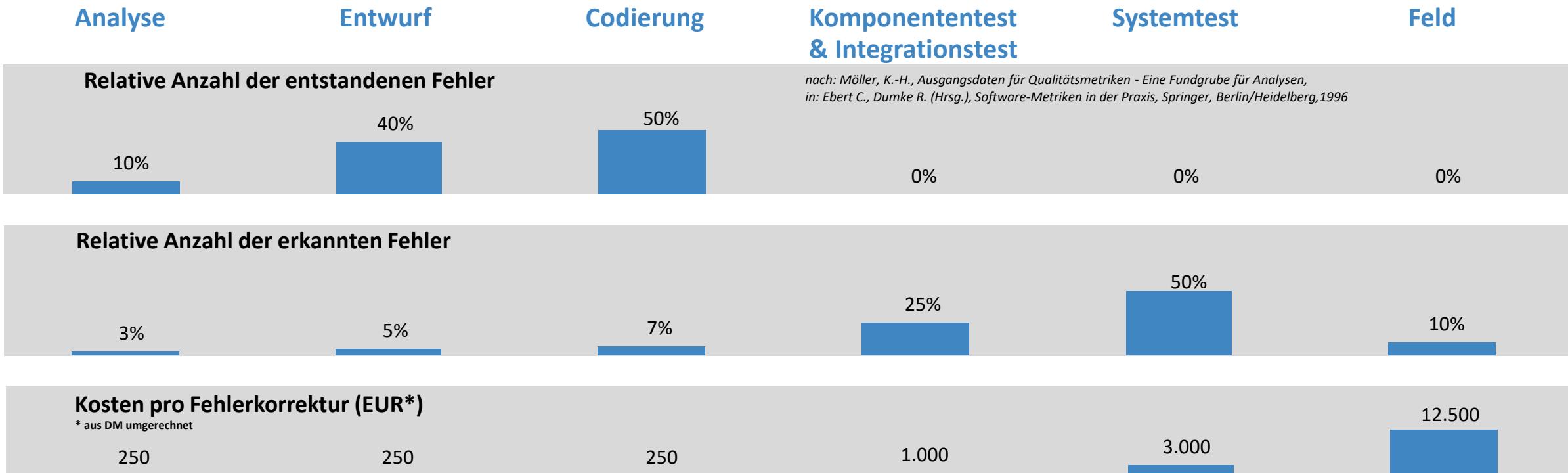
- die für den Menschen schwer zu interpretieren sind
- die nicht mit Hilfe von Werkzeugen analysiert werden sollten
 - z. B. ausführbarer Code von Drittanbietern, welcher aus rechtlichen Gründen nicht untersucht werden darf.

Wert des statischen Tests



Statischer Test – Grundlagen

Fehlerquellen und Fehlerkorrekturkosten



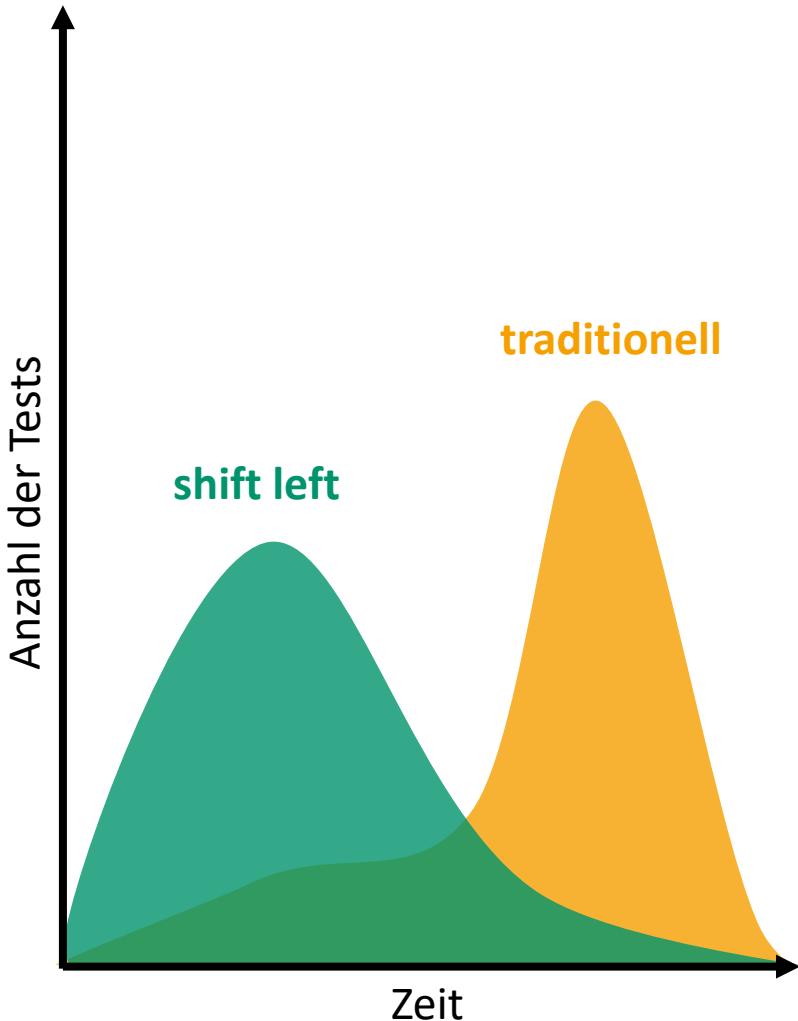
Um Fehler früher zu finden als beim dynamischen Test, sowie um Fehlerkosten zu reduzieren, können und sollten im SDLC frühzeitig statische Tests durchgeführt werden.

Umfrage



- Gibt es in Ihrem Umfeld statische Tests?
- Wie sind Ihre Erfahrungen mit statischen Tests?
- Welche Werkzeuge verwenden Sie?

Wert des statischen Tests I



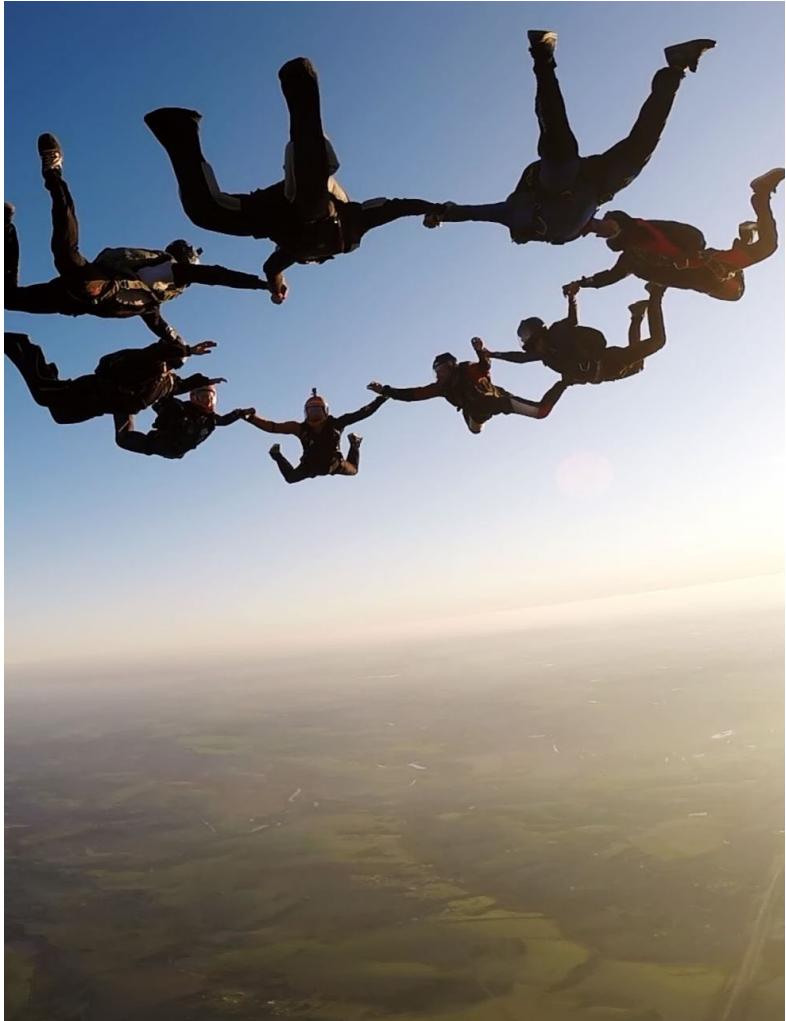
Fehler früh finden

Statischer Test kann Fehlerzustände früh im SWE-Lebenszyklus aufdecken und erfüllt damit den Grundsatz des frühen Testens.

Statischer Test findet Fehlerzustände, die mit dynamischen Tests nicht erkannt werden können. Z. B.

- nicht erreichbarer Code
- nicht wie gewünscht implementierte Entwurfsmuster
- Fehlerzustände in nicht ausführbaren Arbeitsergebnissen

Wert des statischen Tests II



Qualität bewerten und Vertrauen aufbauen

Stakeholder überprüfen die Anforderungen

- Dadurch stellen sie sicher, dass diese ihre tatsächlichen Bedürfnisse beschreiben.

Statische Tests können früh im Projekt durchgeführt werden.

- Bildung von einem gemeinsamen Verständnis zwischen den beteiligten Stakeholdern.
- Verbesserung der Kommunikation zwischen den Stakeholdern.

Es ist empfehlenswert, viele verschiedene Stakeholder in statische Tests einzubeziehen.

Wert des statischen Tests III



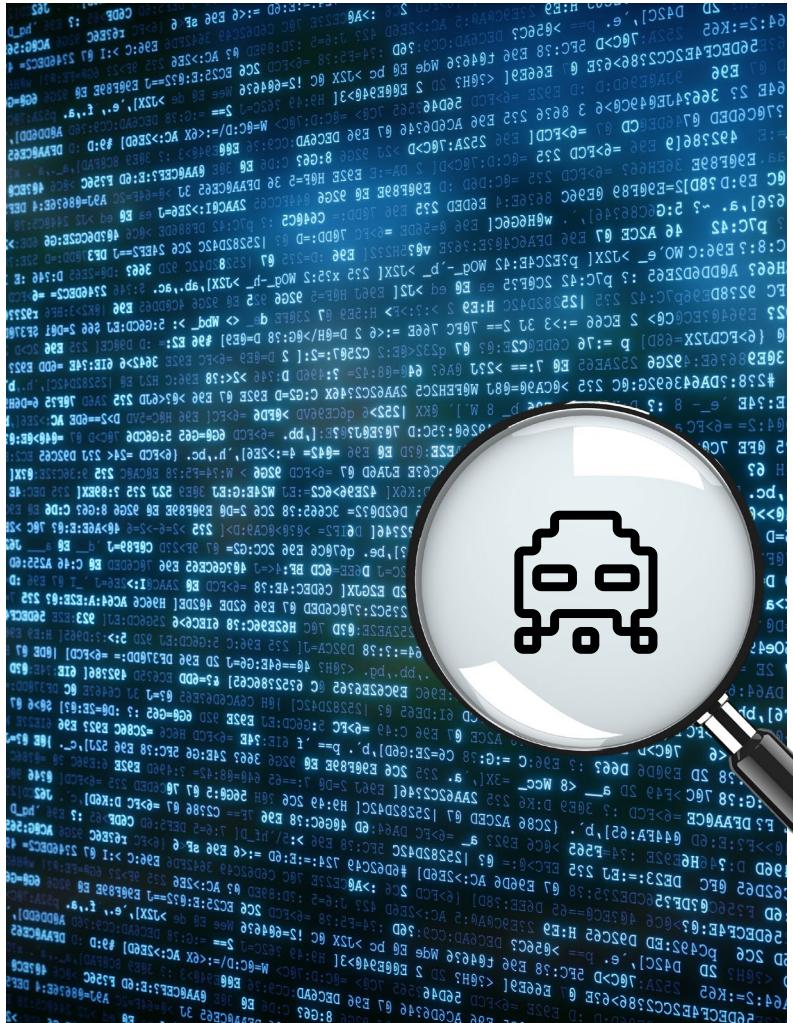
Kosten sparen

Reviews durchführen verursacht Kosten.

Aber die Gesamtkosten von Projekten, in denen Reviews durchgeführt werden, sind in der Regel wesentlich geringer.

- Weniger Zeit und Aufwand für die Behebung von Fehlerzuständen im späteren Verlauf des Projekts

Wert des statischen Tests IV



Fehlerzustände finden

Mit statischer Analyse können bestimmte Fehlerzustände im Code effizienter aufgedeckt werden als durch dynamische Tests.

- weniger Fehler im Code
- geringerer Gesamtentwicklungsaufwand

Unterschiede zwischen statischem Test und dynamischem Test

Statischer Test oder dynamischer Test?

Beide ergänzen sich, können die gleichen Ziele verfolgen,
(Beurteilung der Qualität, Fehlerzustände so früh wie möglich identifizieren)
finden aber unterschiedliche Fehlerarten.

| Statischer Test | Dynamischer Test |
|--|---|
| Findet Fehlerzustände in Arbeitsergebnissen direkt. | Findet Fehlerwirkungen , wenn die Software läuft. (Debugging ermittelt die zugehörigen Fehlerzustände) |
| Bei selten genutzten oder schwer erreichbaren Pfaden → unter Umständen geringerer Aufwand. | Bei selten genutzten oder schwer erreichbaren Pfaden → ggf. hoher Aufwand, um Tests zu konstruieren oder durchzuführen. |
| Bewertung von Qualitätsmerkmalen, die durch das Ausführen von Code nicht gefunden werden können (z. B. Wartbarkeit). | Bewertung von Qualitätsmerkmalen, die von der Ausführung des Codes abhängen (z. B. Performanz). |



Statischer Test – typische Fehlerzustände

Fehlerzustände in den Anforderungen

- Inkonsistenzen, Mehrdeutigkeiten, Widersprüche
- Auslassungen, Ungenauigkeiten, Duplikationen

Fehlerzustände im Entwurf

- ineffiziente Datenbankstrukturen
- schlechte Modularität

Bestimmte Arten von Fehlerzuständen im Code

- Variablen mit undefinierten Werten
- nicht deklarierte Variablen
- unerreichbarer oder duplizierter Code
- übermäßige Komplexität des Codes

Abweichungen von Standards

- mangelnde Einhaltung von Namenskonventionen in Programmierstandards

Falsche Spezifikation von Schnittstellen

- nicht übereinstimmende Anzahl, Art oder Reihenfolge von Parametern

Spezifische Arten von Schwachstellen in der IT-Sicherheit

- Pufferüberläufe

Lücken oder Ungenauigkeiten in der Überdeckung der Testbasis

- fehlende Tests für ein Akzeptanzkriterium



Schlüsselbegriffe – Statischer Test – Grundlagen

statischer Test

Testen, das die Ausführung eines Testelements nicht beinhaltet.

dynamischer Test

Testen, das die Ausführung des Testelements beinhaltet.

statische Analyse

Der Prozess der Bewertung eines Testobjekts (Komponente oder System) basierend auf seiner Form, seiner Struktur, seines Inhalts oder seiner Dokumentation, ohne es auszuführen.

Review

Eine Art statischer Test, bei dem ein Arbeitsergebnis oder -prozess von einer oder mehreren Personen bewertet wird, um Fehlerzustände zu erkennen oder Verbesserungen zu erzielen.



Statischer Test

Prüfbare Arbeitsergebnisse, z. B.

- Anforderungsspezifikationen
- Testkonzepte, Testfälle, Test-Chartas
- Produkt-Backlog-Elemente
- Projektdokumentation
- Verträge
- Modelle
- Code
- Text mit einer formalen Syntax

Wert des statischen Tests

- Fehler früh finden
- Qualität bewerten und Vertrauen aufbauen
- Kosten sparen
- Fehlerzustände finden

Jedes Arbeitsergebnis, das gelesen und verstanden werden kann, kann Gegenstand eines Reviews sein.

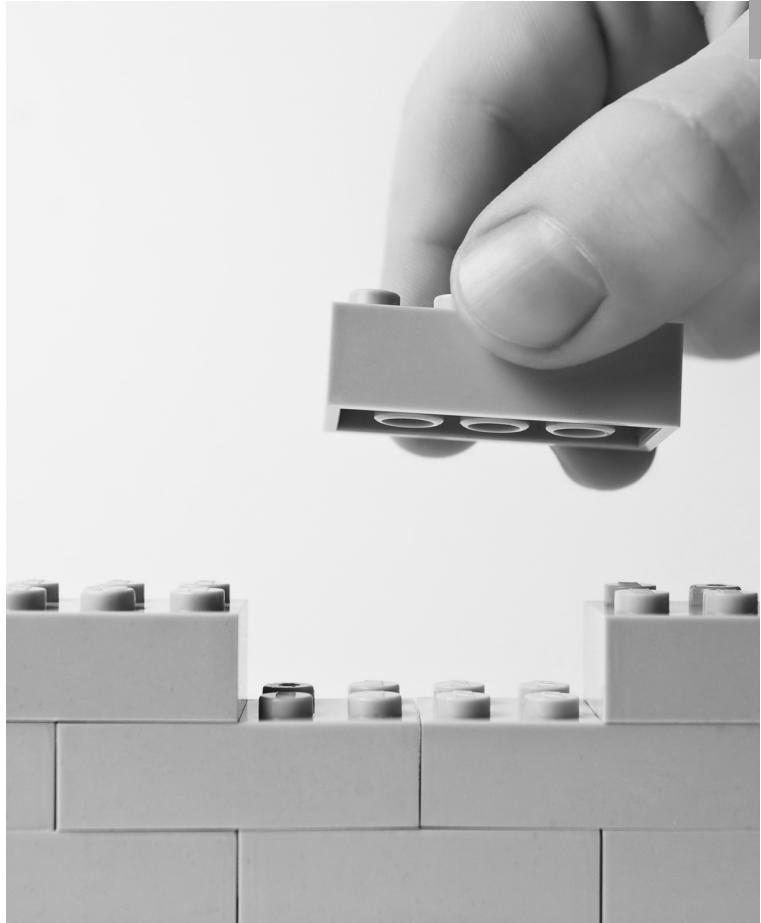


Statischer Test – Dynamischer Test

| | Statischer Test | Dynamischer Test |
|----------------------------------|--|--|
| Verfahren | <ul style="list-style-type: none">▪ Reviews▪ statische Analyse | <ul style="list-style-type: none">▪ Black-Box-Testverfahren▪ White-Box-Testverfahren▪ Erfahrungsbasierte Testverfahren |
| Ziele | Fehlerzustände aufdecken | Fehlerwirkungen finden |
| Typische gefundene Fehler | <ul style="list-style-type: none">▪ Anforderungsfehler▪ Entwurfsfehler▪ Programmierfehler▪ Abweichungen von Standards▪ fehlerhafte Schnittstellenspezifikationen▪ Lücken oder Ungenauigkeiten in der Verfolgbarkeit | <ul style="list-style-type: none">▪ Funktionale Fehler▪ Nicht-funktionale Fehler |
| Rolle im SWE-Lebenszyklus | Findet in den frühen Phasen des SDLC statt, meist bei der Entwicklung | (unabhängiges) Testen auf allen Teststufen |

4 – Statischer Test

1. Statischer Test – Grundlagen
2. Feedback- und Reviewprozess



4.2 Feedback- und Reviewprozess

- **FL-3.2.1 (K1)** Sie können Vorteile eines frühzeitigen und häufigen Stakeholder-Feedbacks erkennen
- **FL-3.2.2 (K2)** Sie können die Aktivitäten des Reviewprozesses zusammenfassen
- **FL-3.2.3 (K1)** Sie können die bei der Durchführung von Reviews den Hauptrollen zugewiesenen Verantwortlichkeiten wiedergeben
- **FL-3.2.4 (K2)** Sie können verschiedene Arten von Reviews vergleichen und gegenüberstellen
- **FL-3.2.5 (K1)** Sie können die Faktoren, die zu einem erfolgreichen Review beitragen, wiedergeben

Vorteile von frühzeitigem und häufigem Feedback

Frühzeitiges und häufiges Stakeholder Feedback

Ein frühzeitiges und häufiges Feedback von Stakeholdern ermöglicht die frühzeitige Kommunikation von potenziellen Qualitätsproblemen.



Häufiges Feedback der Stakeholder während des SDLC

Häufiges Feedback der Stakeholder ...

- kann Missverständnissen über Anforderungen vorbeugen
- kann sicherstellen, dass Änderungen an den Anforderungen verstanden und früher umgesetzt werden.
- hilft dem Entwicklungsteam dabei, besser zu verstehen, was es entwickelt.
- ermöglicht dem Team, sich auf die Features zu konzentrieren, die für die Stakeholder den größten Nutzen bringen und die für die Risikominderung relevant sind.





Beispiel: Häufiges Feedback in Scrum

Daily Scrum

- Tägliches Treffen des Entwicklungsteams zur Berichterstattung über den Fortschritt seit dem letzten Meeting, der Aktivitäten bis zum nächsten Meeting und der Behinderungen bei der Erledigung der Aufgaben.

Sprint Review

- Präsentation der Sprint-Ergebnisse / Vorstellung der erstellten Funktionalität vor dem Product-Owner und ggf. sonstigen Stakeholdern am Ende jeder Iteration (Sprint).

Sprint Retrospektive

- Nach jedem Sprint stattfindendes Meeting des Scrum-Teams zur Reflektion auf Basis des zurückliegenden Sprints: Was hat gut geklappt? Was hat nicht gut geklappt und muss verändert werden?



Beispiel: Häufiges Feedback in Scrum

Ständig informelle Reviews

- Das Scrum-Team (PO, Developer) ist bei der Erstellung der Anforderungen (User-Storys) beteiligt
- Informelle Reviews (z.B. Pairing PO / Developer und /oder Backlog Grooming) während der Erstellung und nicht erst, wenn die Anforderungen komplett in einem ersten Entwurf vorliegen.
- Tester spielen hier durch kritisches Hinterfragen eine sehr wichtige Rolle.

Continuous Integration

- Frühes Erkennen von Integrationsproblemen und widersprüchlichen Änderungen (tägliches Feedback über die Produktqualität).

Fehlendes häufiges Feedback der Stakeholder

Das zu entwickelnde Produkt entspricht möglicherweise nicht den ursprünglichen oder aktuellen Vorstellungen der Stakeholder.

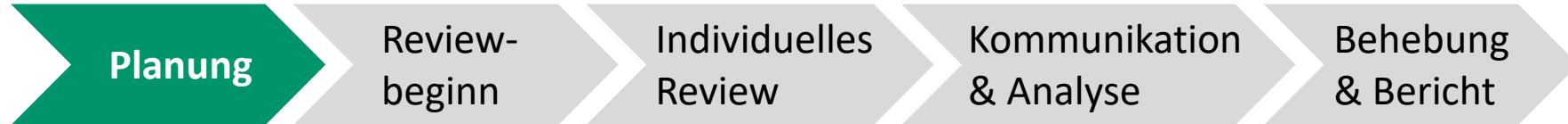
Wenn die Wünsche der Stakeholder nicht erfüllt werden, kann dies zu Problemen im Projekt führen.

- kostspielige Nacharbeiten
- verpasste Termine
- Schuldzuweisungen
- komplettes Scheitern



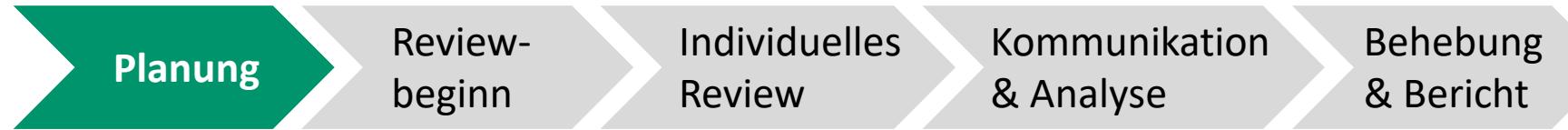
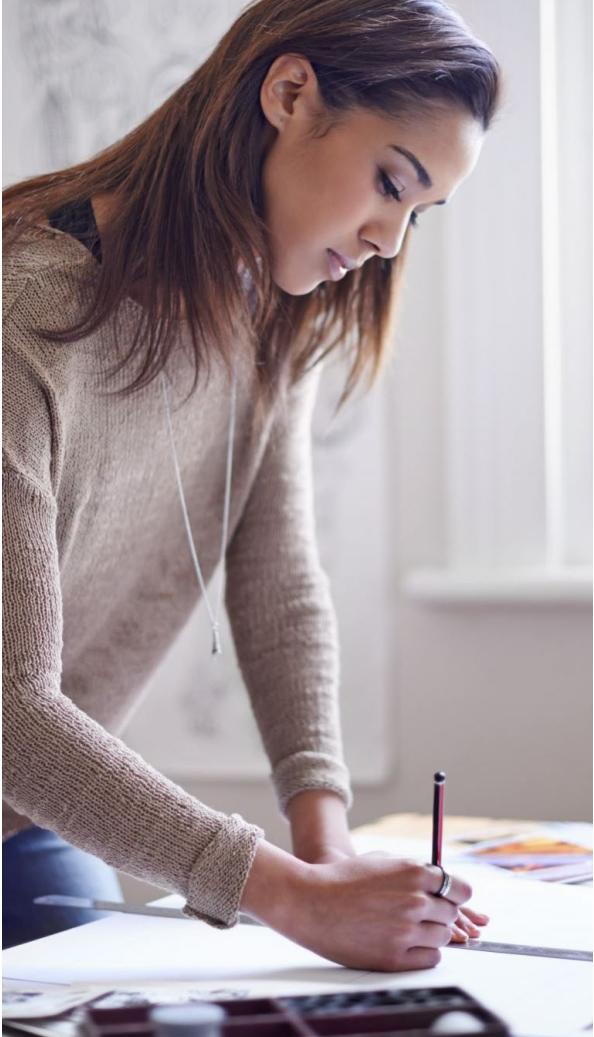
Reviewprozess – Aktivitäten

Reviewprozess – Aktivitäten



- Die ISO/IEC 20246 definiert einen generischen Reviewprozess, der einen strukturierten, aber flexiblen Rahmen bietet, auf dessen Grundlage ein spezifischer Reviewprozess auf eine bestimmte Situation zugeschnitten werden kann.
- Formale Reviews erfordern für die verschiedenen Aktivitäten detaillierte Aufgabenbeschreibungen.
- Bei umfangreichen Arbeitsergebnissen wird der Reviewprozess mehrfach durchgeführt, um das Review für das gesamte Arbeitsergebnis zu vervollständigen.

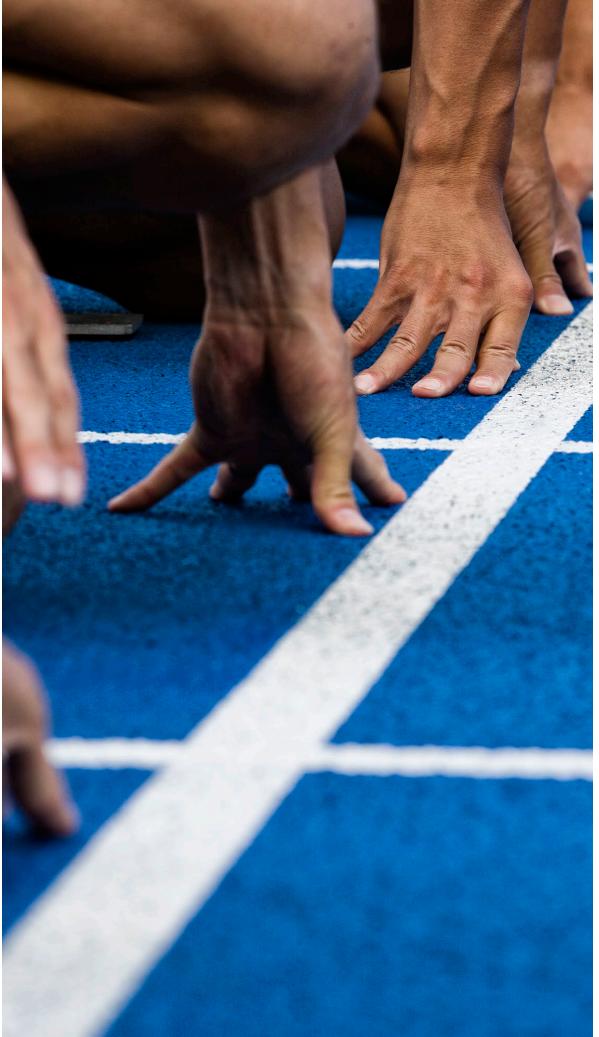
Planung



In der **Planungsphase** wird festgelegt:

- Zweck und Umfang des Reviews
- Aufwand und Zeitrahmen
- das zu überprüfende Arbeitsergebnis
- die zu berücksichtigenden Bereiche
- Endekriterien
- die zu bewertenden Qualitätsmerkmale
- unterstützende Informationen, z. B. Normen

Reviewbeginn

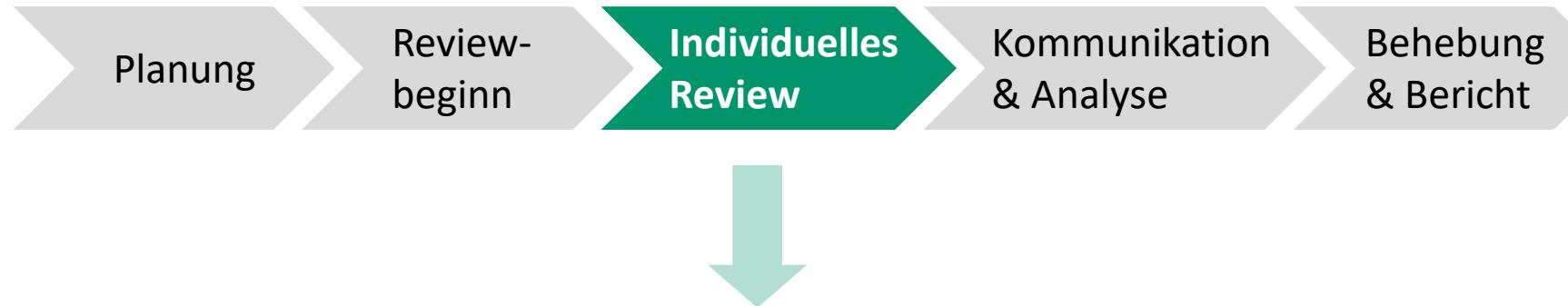
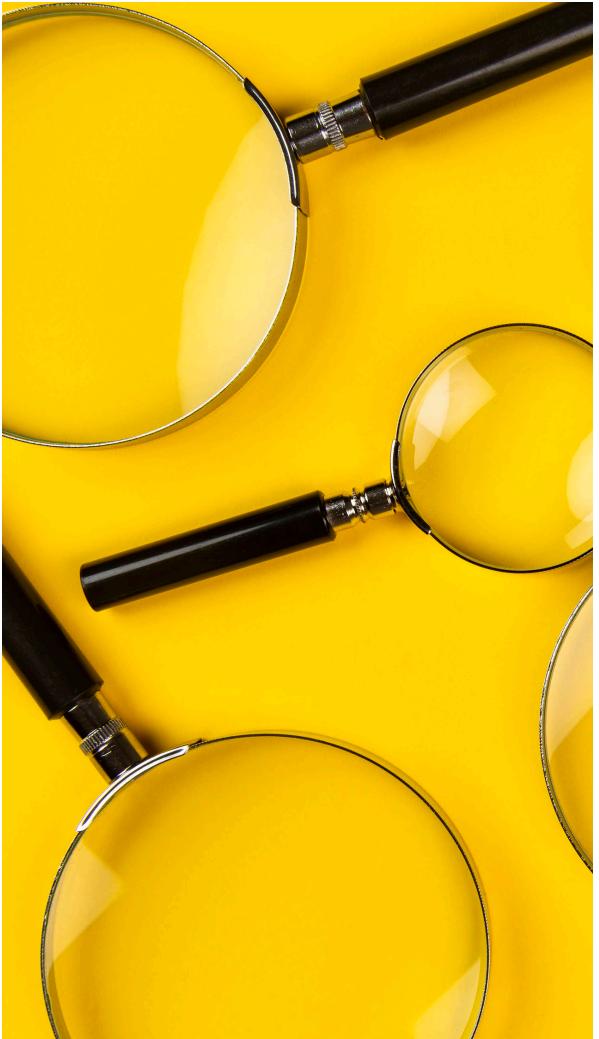


Der **Reviewbeginn** stellt sicher, dass alles für den Start des Reviews vorbereitet ist.

Dazu gehört, dass die Teilnehmer ...

- Zugang zum zu prüfenden Arbeitsergebnis haben
- die Rollen und Verantwortlichkeiten kennen und verstehen
- alle benötigten Informationen erhalten

Individuelles Review



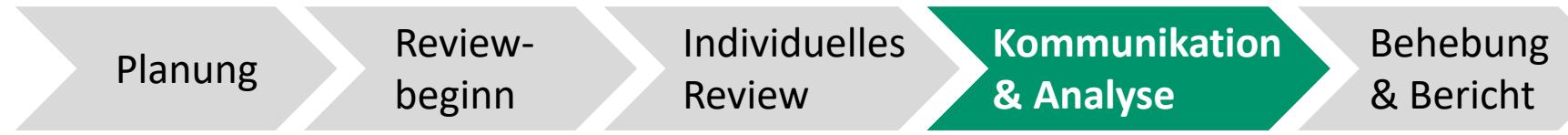
Gutachter führen **individuelle Reviews** durch, um

- die Qualität des zu prüfenden Arbeitsergebnisses zu bewerten.
- Anomalien, Empfehlungen und Fragen zu identifizieren und zu protokollieren.

Reviewverfahren helfen dabei (z. B. checklistenbasiertes Review und szenariobasiertes Review).

Die ISO/IEC 20246 geht näher auf die verschiedenen Reviewverfahren ein.

Kommunikation und Analyse



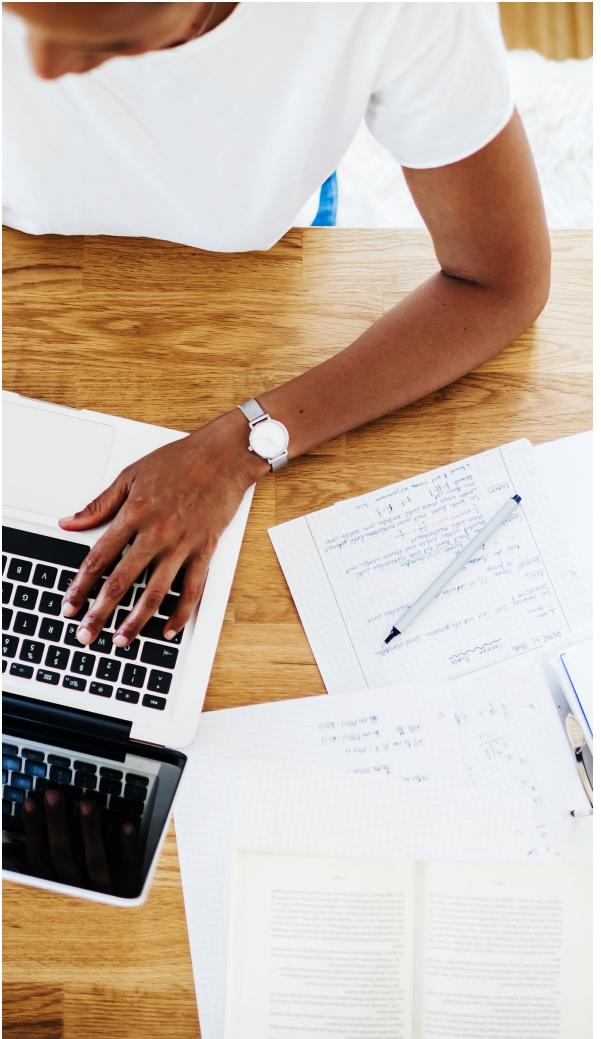
Nicht bei allen Anomalien, die während eines Reviews festgestellt werden, handelt es sich um Fehlerzustände. Daher müssen Anomalien analysiert und diskutiert werden.

Für jede Anomalie sollte eine Entscheidung über ihren Status, ihre Verantwortlichkeit und die erforderlichen Maßnahmen getroffen werden. Dies geschieht in der Regel in einer **Reviewsitzung**, um

- die Qualität des geprüften Arbeitsergebnisses zu ermitteln.
- über erforderliche Folgemaßnahmen zu entscheiden.

Nach Abschluss der Maßnahmen kann ein Folgereview erforderlich sein.

Behebung und Berichterstattung



Für jeden Fehlerzustand sollte ein **Fehlerbericht** erstellt werden, damit die Korrekturmaßnahmen nachverfolgt werden können.

Wenn die Endekriterien erreicht sind, kann das Arbeitsergebnis abgenommen werden.

Über die Ergebnisse des Reviews wird berichtet.

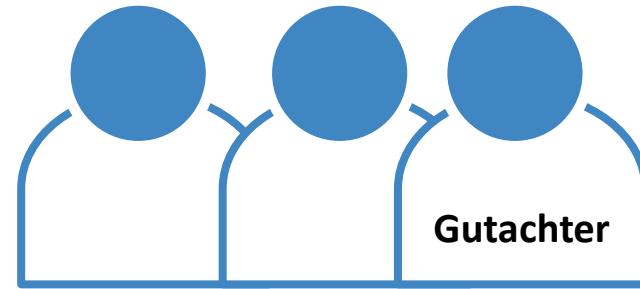
Rollen und Verantwortlichkeiten bei Reviews

Rollen und Verantwortlichkeiten bei Reviews I

Entscheidet, was geprüft werden soll, und stellt Ressourcen wie Personal und Zeit für das Review zur Verfügung.



Manager



Gutachter

Sorgt für einen effektiven Ablauf der Sitzungen, einschließlich Mediation, Zeitmanagement und einer geschützten Umgebung, in der jeder frei sprechen kann.



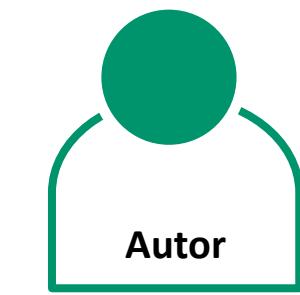
**Moderator
(Facilitator)**



Übernimmt die Gesamtverantwortung für das Review, z. B. die Entscheidung, wer daran teilnimmt, und die Organisation, wann und wo das Review stattfindet.



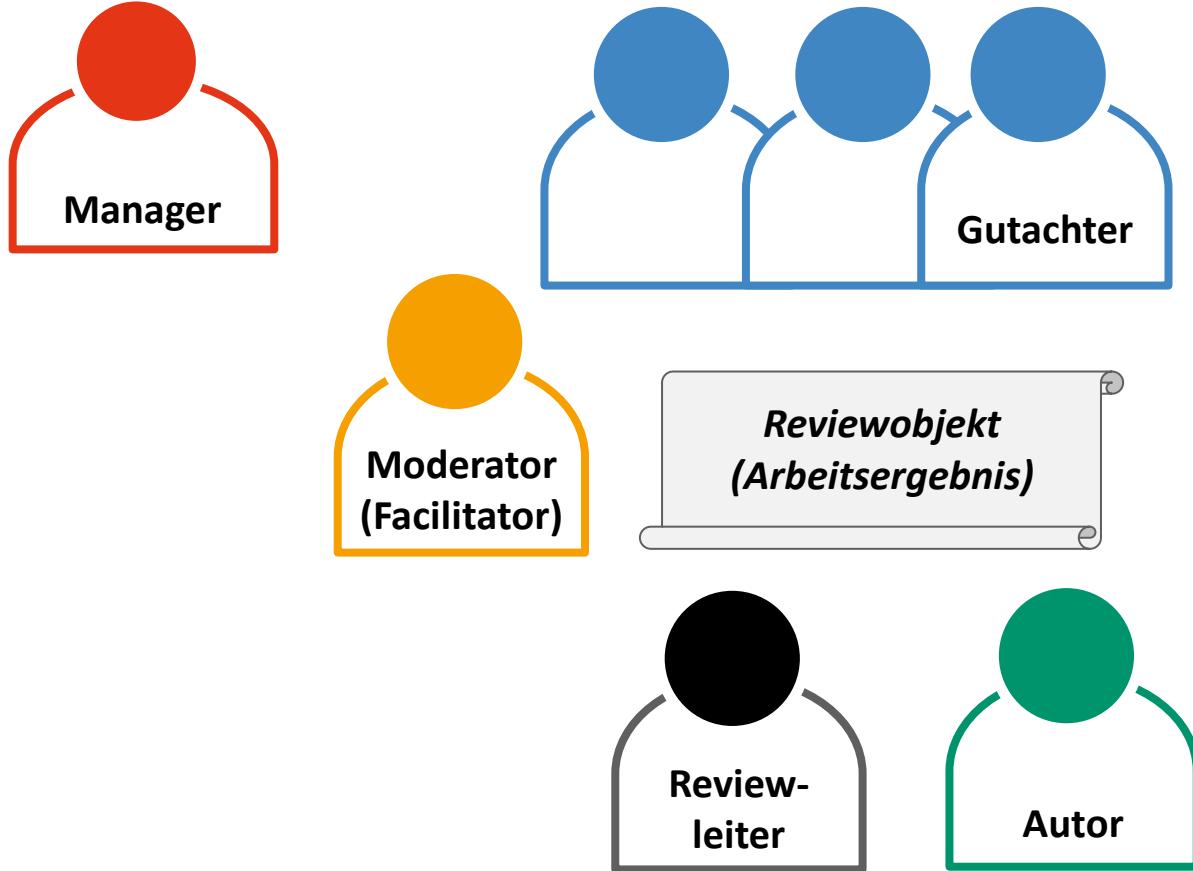
**Review-
leiter**



Autor

Andere, detailliertere Rollen sind möglich, siehe ISO/IEC 20246

Rollen und Verantwortlichkeiten bei Reviews II



Führen Reviews durch.
Ein Gutachter (auch Reviewer genannt)
kann ein Projektmitarbeiter, ein Fachexperte oder
ein anderer Stakeholder sein.

Sammelt Anomalien von Gutachtern
und zeichnet Informationen auf,
z. B. Entscheidungen und neue
Anomalien, die während der Sitzung
gefunden werden.

Erstellt und korrigiert das zu prüfende
Arbeitsergebnis



Arten von Reviews

Arten von Reviews I

Es gibt viele Arten von Reviews, die von informellen Reviews bis zu formalen Reviews reichen.



Informelles Review



Walkthrough



Technisches Review



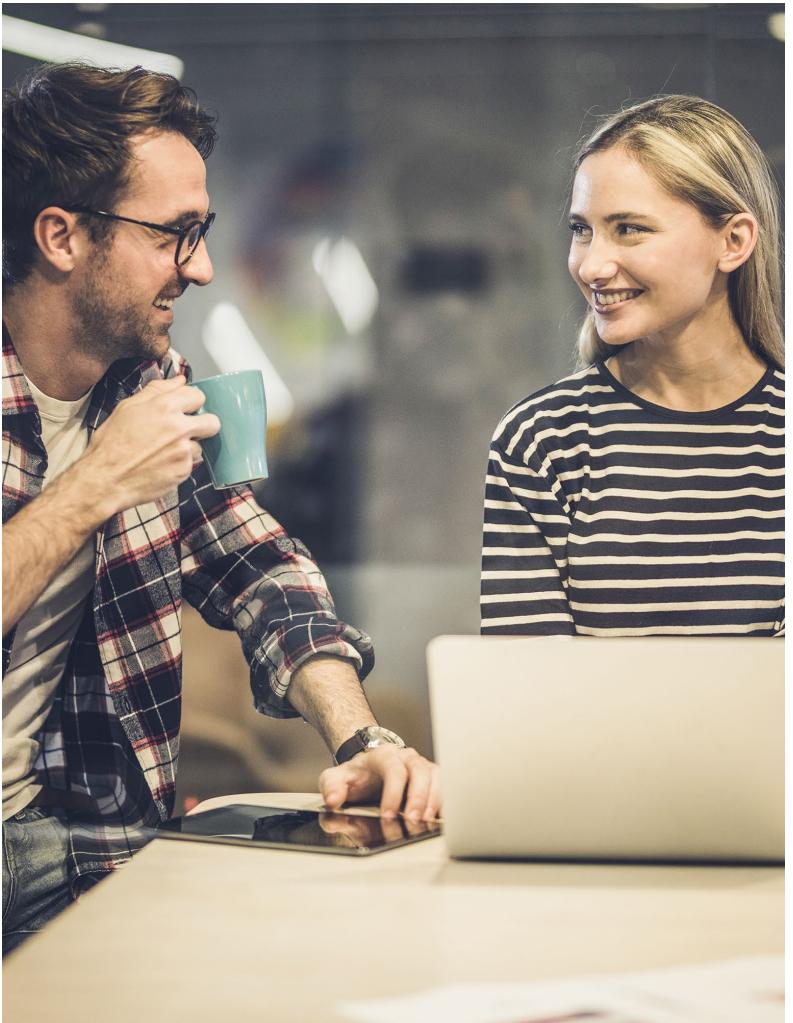
Inspektion

Der erforderliche Grad an Formalität hängt ab von Faktoren wie

- dem angewandten SDLC
- der Reife des Entwicklungsprozesses
- der Kritikalität und Komplexität des zu prüfenden Arbeitsergebnisses
- gesetzlichen oder regulatorischen Anforderungen
- dem Bedarf an einem Prüfnachweis

Ein und dasselbe Arbeitsergebnis kann mit verschiedenen Reviewarten geprüft werden, z. B. zunächst mit einem informellen und später mit einem formalen Review.

Informelles Review



Informelle Reviews

- folgen keinem definierten Prozess
- erfordern keine formalen, dokumentierten Ergebnisse

Hauptziel

- Anomalien aufdecken

Beispiel – informelles Review I



Vorbereitung

Ziel des Reviews

Schnell und kostengünstig Fehlerzustände finden.

Der Autor, Ferdinand Müller, übergibt das Dokument „Anleitung_Login_in_VSR-II“ an seine Kollegin, Erika Musterfrau, zum Review.

Endekriterien

Es gibt keine formalen Eingangs- und Endekriterien.

Individuelles Review

Erika liest das Dokument und notiert alle gefundenen potenziellen Fehlerzustände und alle Fragen im Nachverfolgungsmodus des Textverarbeitungsprogramms.



Beispiel – informelles Review II



Planung

Review-
beginn

Individuelles
Review

Kommunikation
& Analyse

Behebung
& Bericht

Planung

Review-
beginn

Individuelles
Review

Kommunikation
& Analyse

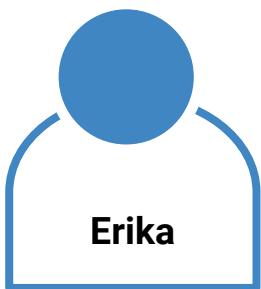
Behebung
& Bericht

Kommunikation & Analyse

Ferdinand und Erika besprechen gemeinsam alle gefundenen Punkte. Ferdinand beantwortet Erikas Fragen.



Ferdinand



Erika

Behebung & Berichterstattung

Ferdinand überarbeitet das Dokument „Anleitung_Login_in_VSR-II“ im Korrekturmodus des Textverarbeitungsprogramms.



Walkthrough



Ein Walkthrough

- wird vom Autor geleitet
- Gutachter können ein individuelles Review vor dem Walkthrough durchführen, es ist jedoch nicht verpflichtend

Ziele

- Bewertung der Qualität
- Aufbau von Vertrauen in das Arbeitsergebnis
- Schulung von Gutachtern
- Erzielung eines Konsenses
- Generierung neuer Ideen
- Autoren zur Verbesserung motivieren
- Aufdeckung von Anomalien

Technisches Review



Technisches Review

- Leitung durch Moderator
- Durchführung mit technisch qualifizierten Gutachtern

Ziele

- Konsens erzielen
- Entscheidungsfindung zu einem technischen Problem
- Anomalien aufdecken
- Qualität bewerten
- Vertrauen in das Arbeitsergebnis aufbauen
- neue Ideen entwickeln
- Motivation und Befähigung der Autoren zur Verbesserung

Inspektion



Inspektionen

- formalste Art der Reviews
- folgen dem vollständigen Reviewprozess
- Autor kann nicht Reviewleiter oder Protokollant sein
- Metriken werden gesammelt
- Kontinuierliche Verbesserung von Softwareentwicklungs- und Inspektionsprozess, durch Nutzung von Metriken

Hauptziel

- die maximale Anzahl von Anomalien zu finden

Weitere Ziele

- Qualität bewerten
- Vertrauen in das Arbeitsergebnis aufbauen
- Motivation und Befähigung der Autoren zur Verbesserung



Beispiel – Inspektion I



Planung

Hauptziel: Freigabe des Dokuments „Anleitung_Login_in_VSR-II“.

Weitere Ziele: Falls möglich sollen Maßnahmen zur Prozessverbesserung identifiziert werden.

Zusammen mit dem Management wird festgelegt, welche Teilnehmer (Rollen) am Review teilnehmen werden. Das Management bestimmt einen Reviewleiter, der sich um die Ausführung kümmert.

Eingangskriterium: Das Dokument entspricht den formalen Vorgaben.

Endekriterium: Alle Befunde und Fragen wurden besprochen und der Fehlerstatus dokumentiert.

In der Planungsphase wird geprüft, ob das Eingangskriterium erreicht wurde. Sobald das Eingangskriterium erfüllt ist, kann das Review beginnen.

Beispiel – Inspektion II



Reviewbeginn

Im Kick-Off-Meeting verteilt der Reviewleiter folgende Informationen an die Teilnehmer:

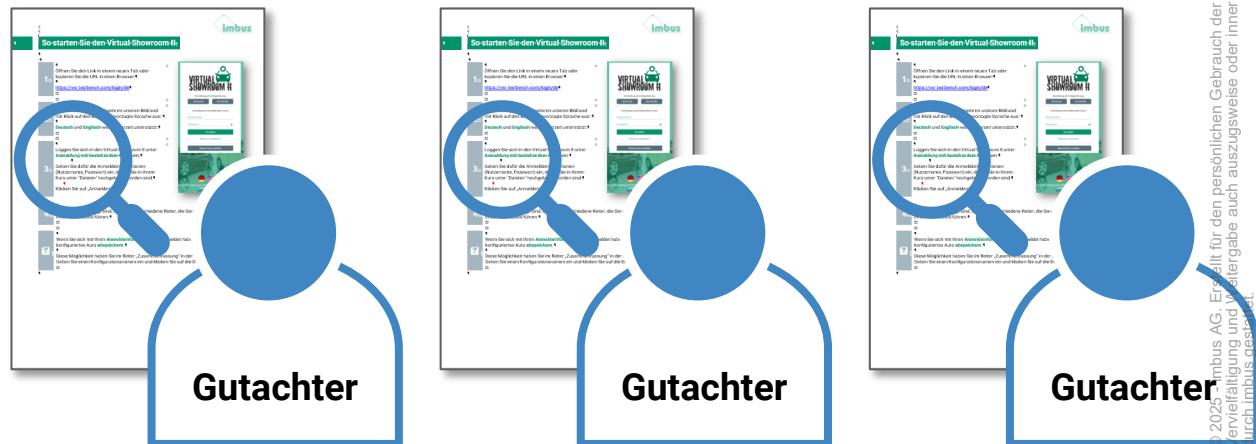
- Dokument „Anleitung_Login_in_VSR-II“
- Template zum Eintragen der Befunde
- Zeitplan und Agenda
- Checkliste
- Zuteilung bestimmter Prüfaspakte an bestimmte Gutachter

Die Gutachter bekommen eine Woche Zeit, um das individuelle Review abzuschließen.



Individuelles Review

Jeder Gutachter prüft das Dokument „Anleitung_Login_in_VSR-II“ und notiert gefundene Fehler, Fragen und Kommentare im Befund-Template.



Beispiel – Inspektion III



Kommunikation & Analyse

Das Eingangskriterium – alle Gutachter haben Reviewanmerkungen eingereicht – ist erfüllt.

Die Befundkommunikation erfolgt in einer Reviewsitzung.

Die Sitzung sollte nicht länger als 2 Stunden dauern.

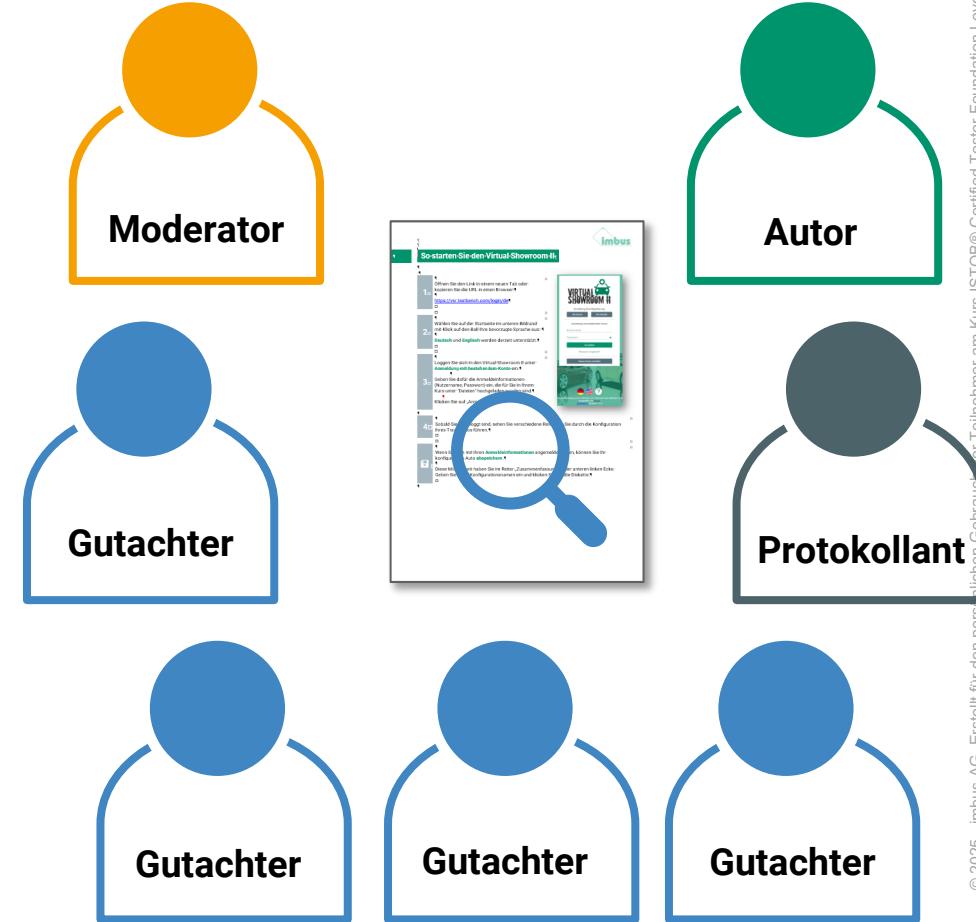
Die Sitzung wird von einem geschulten Moderator geleitet.
Dieser führt durch die Sitzung und moderiert Diskussionen.

Ideen für Prozessverbesserungen werden im Reviewbericht dokumentiert.

Die Punkte der Gutachter werden bewertet.

Der Protokollant dokumentiert die Ergebnisse der Sitzung.

Die Sitzung ist beendet, wenn das Endekriterium – alle Punkte wurden besprochen – erfüllt ist.



Beispiel – Inspektion IV



Behebung & Berichterstattung

Ferdinand Müller überarbeitet das Dokument „Anleitung_Login_in_VSR-II“ gemäß Protokoll und aktualisiert im Review-Protokoll den Status der einzelnen Punkte.

Die im Review identifizierten Prozessverbesserungsmaßnahmen werden im Folgeprojekt vom Management umgesetzt.

Für jeden Fehlerzustand wird ein Fehlerbericht erstellt, damit die Korrekturmaßnahmen nachverfolgt werden können.



Arten von Reviews II

Die Auswahl der richtigen Reviewart ist der Schlüssel zum Erreichen der geforderten Reviewziele.

Die Auswahl richtet sich nicht nur nach den Zielen, sondern auch nach Faktoren wie



Projektbedarf



verfügbare
Ressourcen



Art des
Arbeitsergebnisses
& seinen Risiken



Unternehmens-
bereich



Unternehmens-
kultur



Schlüsselbegriffe – Arten von Reviews

Anomalie

Ein Zustand, der von der Erwartung abweicht.

formales Review

Ein Review, das einem definierten Prozess folgt und ein formell dokumentiertes Ergebnis liefert.

informelles Review

Ein Review, das keinem definierten Prozess folgt und kein formell dokumentiertes Ergebnis liefert.

Walkthrough

Eine Reviewart, bei der ein Autor die Reviewteilnehmer durch ein Arbeitsergebnis leitet und die Teilnehmer Fragen stellen und Auffälligkeiten kommentieren.

Technisches Review

Ein formales Review durch technische Experten, die die Qualität eines Arbeitsergebnisses untersuchen und Abweichungen von Spezifikationen und Standards feststellen.

Inspektion

Eine formale Reviewart, die festgelegte Teamrollen und Messungen verwendet, um Fehlerzustände in einem Arbeitsergebnis zu identifizieren und den Reviewprozess sowie den Softwareentwicklungsprozess zu verbessern.

Erfolgsfaktoren für Reviews

Erfolgsfaktoren für Reviews I

- Reviews zu einem Teil der Unternehmenskultur machen, um Lernen und Prozessverbesserung zu fördern.
- Unterstützung des Reviewprozesses durch das Management.
- Geeignete Schulungen für alle Teilnehmer anbieten, damit sie wissen, wie sie ihre Rolle erfüllen können.
- Die Festlegung klarer Ziele und messbarer Endekriterien. Die Bewertung der Teilnehmer sollte niemals ein Ziel sein.
- Auswahl der geeigneten Reviewart, um die vorgegebenen Ziele zu erreichen und um der Art des Arbeitsergebnisses, den Reviewteilnehmern, den Projektanforderungen und dem Kontext gerecht zu werden.



Während der Planung

Erfolgsfaktoren für Reviews II

- Durchführung von Reviews in kleinen Teilen, damit die Gutachter während eines individuellen Reviews und/oder der Reviewsitzung (sofern diese stattfindet) nicht die Konzentration verlieren.
- Bereitstellung von ausreichend Zeit für die Teilnehmer zur Vorbereitung auf das Review.
- Moderation der Sitzungen
- Lieferung von Feedback an Stakeholder und Autoren, damit diese das Produkt und ihre Aktivitäten verbessern können.



Während der Durchführung



Frühzeitiges Feedback – Vorteile

Vorteile eines frühzeitigen und häufigen Feedbacks von den Stakeholdern

- Ermöglicht die frühzeitige Kommunikation von potenziellen Qualitätsproblemen
- Beugt Missverständnissen über Anforderungen vor
- Stellt sicher, dass Änderungen an den Anforderungen verstanden und früher umgesetzt werden
- Hilft, dass das zu entwickelnde Produkt den Vorstellungen der Stakeholder entspricht
- Vermindert kostspielige Nacharbeiten, verpasste Termine, Schuldzuweisungen und kompletten Ausfall des Projekts
- Hilft dem Entwicklungsteam dabei, besser zu verstehen, was sie entwickeln
- Ermöglicht dem Entwicklungsteam, sich auf die Funktionen zu konzentrieren, die für die Stakeholder den größten Nutzen bringen und die größten Auswirkungen auf die identifizierten Risiken haben



Reviewprozess – Aktivitäten & Verantwortlichkeiten

Aktivitäten

1. Planung
2. Reviewbeginn
3. Individuelles Review
4. Kommunikation und Analyse
5. Behebung und Berichterstattung

| Rolle | Verantwortlichkeit |
|--------------|---|
| Management | Verantwortet und steuert den gesamten Prozess. |
| Leiter | Trägt die Verantwortung für das Review und bestimmt die Teilnehmer. |
| Autor | Erstellt das Reviewobjekt und behebt ggf. Fehlerzustände. |
| Moderator | Stellt einen erfolgreichen Ablauf sicher und verhandelt ggf. mit Teilnehmern. |
| Gutachter | Findet Fehlerzustände im Reviewobjekt. |
| Protokollant | Dokumentiert gefundene Fehlerzustände. |



Reviewarten

| Reviewart | Besondere typische Merkmale |
|--------------------|---|
| Informelles Review | Gegenlesen unter Kollegen → kein formaler Prozess und ggf. keine Sitzung und kein Protokoll. |
| Walkthrough | Autor leitet durch das Arbeitsergebnis. Individuelle Vorbereitung ist optional, kann informell oder sehr formal sein. |
| Technisches Review | Review mit Fachexperten, individuelle Vorbereitung und Protokoll sind obligatorisch. |
| Inspektion | Folgt einem definierten formalen Prozess, klar definierte Rollen, Eingangs- und Endekriterien, Einsatz von Metriken. Autor ist nicht Reviewleiter oder Protokollant. |



Erfolgsfaktoren für Reviews

Während der Planung des Reviews, z. B.

- Reviewkultur etablieren
- Managementunterstützung
- Wissen über Reviews etablieren
- Geeignete Reviewart wählen
- Klare Ziele und Endekriterien festlegen
- Reviewprozess kontinuierlich verbessern

Während der Durchführung, z. B.

- Reviews in kleinen Teilen durchführen
- Ausreichend Zeit für Vorbereitung und Durchführung zur Verfügung stellen
- Fehler objektiv behandeln
- Moderation der Sitzungen
- Feedback liefern