



GTB

German Testing Board

Software. Testing. Excellence.



Basiswissen Softwaretest Certified Tester Testmanagement

HS@GTB
2019
Version 3.1

Nach dieser Vorlesung sollten Sie ...

- Wissen über die Organisation von Testteams haben und die wichtigsten Aufgaben von Testmanager und Tester kennen,
- Ziel und Inhalt eines Testkonzepts zusammenfassen können,
- die Testplanung durchführen und dabei zwei Verfahren zur Schätzung der Testkosten kennen,
- Metriken zur Testfortschrittsüberwachung und -steuerung kennen,
- den Inhalt eines Fehler- bzw. Abweichungsberichts kennen und einen Fehler- bzw. Abweichungsbericht erstellen können,
- Zweck und Inhalt eines Testabschlussberichts zusammenfassen können,
- den Zusammenhang zwischen Risiko und Testen kennen sowie das Risikomanagement für die Testplanung einsetzen können,
- die Anforderungen an ein Konfigurationsmanagement benennen können.

Lernziele für den Abschnitt Testmanagement

(nach Certified Tester Foundation Level Syllabus,
deutschsprachige Ausgabe, Version 2018)



5.1 Testorganisation

- FL-5.1.1 (K2) Vor- und Nachteile unabhängigen Testens erklären können.
- FL-5.1.2 (K1) Die Aufgaben eines Testmanagers und eines Testers benennen können

5.2 Testplanung und -schätzung

- FL-5.2.1 (K2) Den Zweck und Inhalt eines Testkonzepts zusammenfassen können.
- FL-5.2.2 (K2) Zwischen verschiedenen Teststrategien unterscheiden können.
- FL-5.2.3 (K2) Beispiele für mögliche Eingangs- und Endekriterien geben können.
- FL-5.2.4 (K3) Wissen über Priorisierung sowie technische und logische Abhängigkeiten anwenden können, um die Durchführung für ein gegebenes Testfallset zu planen
- FL-5.2.5 (K1) Faktoren benennen können, die den Testaufwand beeinflussen.
- FL-5.2.6 (K2) Den Unterschied zwischen zwei Schätzverfahren erklären können: das metrikbasierte Verfahren und das expertenbasierte Verfahren.

Lernziele für den Abschnitt Testmanagement

(nach Certified Tester Foundation Level Syllabus,
deutschsprachige Ausgabe, Version 2018)



5.3 Testüberwachung und -steuerung

- FL-5.3.1 (K1) Testmetriken wiedergeben können.
- FL-5.3.2 (K2) Zweck, Inhalte und Zielgruppen für Testberichte zusammenfassen können.

5.4 Konfigurationsmanagement

- FL-5.4.1 (K2) Zusammenfassen können, wie Konfigurationsmanagement das Testen unterstützt.

5.5 Risiken und Testen

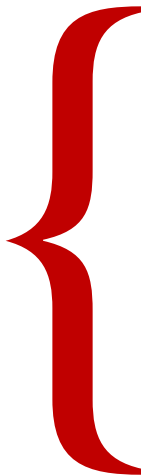
- FL-5.5.1 (K1) Risikostufe anhand der Wahrscheinlichkeit (des Eintritts) und Auswirkung (im Schadensfall) definieren können.
- FL-5.5.2 (K2) Zwischen Projekt- und Produktrisiken unterscheiden können.
- FL-5.5.3 (K2) Anhand von Beispielen beschreiben können, wie die Produktrisikoaanalyse Intensität und Umfang des Testens beeinflussen kann.

5.6 Fehlermanagement

- FL-5.6.1 (K3) Einen Fehlerbericht schreiben können, der während des Testens gefundene Fehler enthalten*. (* einen pro Bericht)

Kapitel 5

Test- management



Testorganisation

Testplanung und -schätzung

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

Konfigurationsmanagement

Unabhängiges Testen

- Testaufgaben können übernommen werden von Personen
 - in einer spezifischen Testrolle oder
 - in einer anderen Rolle, z.B.
Projektmanager, Qualitätsmanager, Entwickler, Fach- und
Bereichsexperte, Mitarbeiter in Infrastruktur oder IT-Betrieb
- etwas Unabhängigkeit verbessert die Effektivität der Fehlerfindung
 - 4-Augen-Prinzip
 - Autoren und Tester haben unterschiedliche kognitive Ausrichtungen

Unabhängigkeit und Testorganisation

Organisationsmodelle

- Jeder Entwickler testet selber
- Entwicklungsteam testet gegenseitig
- Unabhängige Testteams
- Unabhängige Testspezialisten für besondere Themen, z.B. Performanz, Security
- Externe Organisation



**Grad an
Unabhängigkeit**



Was sind Vor- und Nachteile unabhängigen Testens?



Vor- und Nachteile unabhängigen Testens

- Vorteile
 - Unabhängige Tester sind unvoreingenommen
 - Unabhängige Tester hinterfragen Annahmen, die von anderen Stakeholdern gemacht wurden
- Nachteile
 - Hoher Kommunikationsaufwand zwischen Entwicklung und Test
 - Isolation kann den Teamzusammenhalt und die Zusammenarbeit beeinträchtigen.
 - Ein unabhängiges Testteam kann als letzte Prüfinstanz einen Engpass darstellen
 - Es besteht die Gefahr, dass die Entwickler die Verantwortung für Qualität nicht mehr ausreichend wahrnehmen und an die unabhängigen Tester delegieren

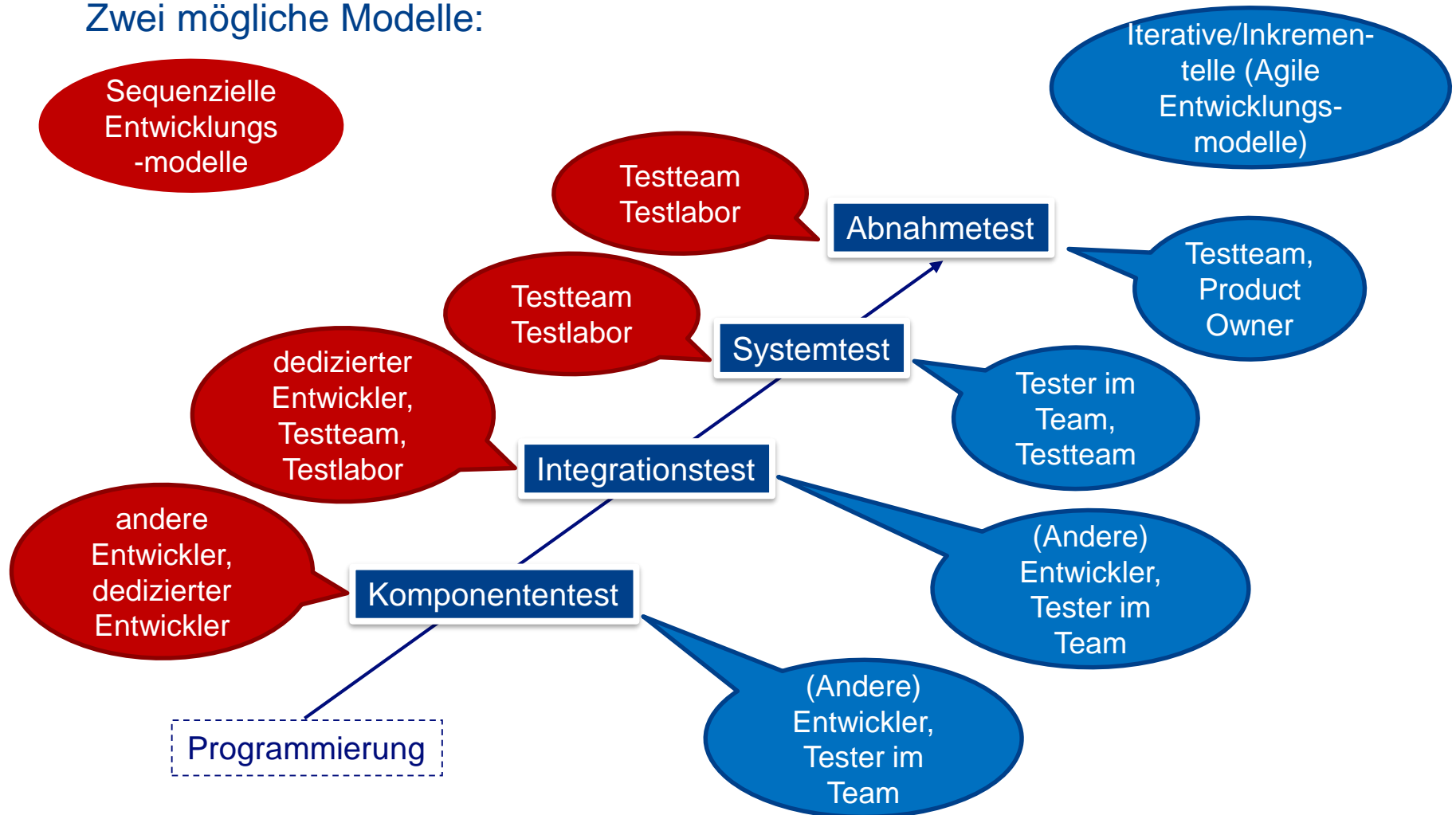


Aufgabenteilung zw. Entwicklung und Test: Tipps

- Je größer, komplexer oder sicherheitskritischer das System ist, desto wichtiger ist der Einsatz von unabhängigen Testern
- Die Unabhängigkeit zwischen Entwicklung und Test kann von Teststufe zu Teststufe variieren.
- Niedrige Teststufen können unter Beteiligung der Entwickler durchgeführt werden, wobei der Mangel an Objektivität dem eigenen Produkt gegenüber oft die Effektivität einschränkt
- Das gewählte Softwareentwicklungsmodell kann auch den Grad an Unabhängigkeit beeinflussen.
 - Beispiel: Bei der agilen Entwicklung führt der Product Owner Akzeptanztests durch, um die User Stories am Ende jeder Iteration zu validieren.

Unabhängigkeit und Testorganisation

Zwei mögliche Modelle:



Mitarbeiterqualifikation

- Technische Kompetenz, z.B. Informatik, Elektrotechnik
- Testspezifisches Knowhow
- Tester benötigen auch soziale Kompetenz, um erfolgreich zu sein:
 - Teamfähigkeit, politisches und diplomatisches Geschick
 - Bereitschaft, scheinbare Tatsachen zu hinterfragen
 - Durchsetzungskraft
 - sicheres Auftreten
 - Exaktheit und Kreativität
 - Fähigkeit, sich schnell in komplexe Anwendungsgebiete und Applikationen einzuarbeiten

Aufgaben und Mitarbeiterqualifikation (1 von 2)

- Know-how der im Test arbeitenden Personen sollte alle Aufgabenbereiche im Testprozess abdecken
- Folgende Rollen sind min. auszufüllen
 - Testmanager
 - Tester
- Möglicherweise Spezialisten für
 - Testanalyse
 - Testentwurf
 - spezifische Testarten
 - Testautomatisierung
 - ...

Aufgaben und Mitarbeiterqualifikation (2 von 2)

Einsatz der Tester auch abhängig von Teststufe oder Risiken:

- **Komponenten- und Integrationsstufe:**
Entwickler (mit Test Know-how)
- **System- und Systemintegrationsteststufe:**
unabhängiges Testteam
- **Abnahmeteststufe:**
Fachexperten, Businessanalysten und Benutzer
- **Abnahmetest auf operativer Ebene:**
Betreiber und/oder Mitarbeitern der Systemadministration



Welche Eigenschaften sollten typische Tester haben?



Aufgaben des Testmanagers (1 von 4)

- **Testrichtlinie** und **Teststrategie** für das Unternehmen entwickeln/prüfen
- **Testkonzepte** schreiben, mit den Projektmanagern, Product Ownern und weiteren Stakeholdern **abstimmen** und **aktualisieren**
- Die **Testaktivitäten** unter Berücksichtigung des Kontexts und des Verständnisses der Testziele und Risiken **planen**, z.B.
 - die Auswahl von Testvorgehensweisen
 - die Schätzung der Testdauer, des Aufwands und der Kosten
 - die Festlegung von Teststufen und Testzyklen und
 - die Planung des Fehlermanagements.
- Einbringen der Testperspektive in andere Projektaktivitäten, beispielsweise in die Integrationsplanung

Aufgaben des Testmanagers (2 von 4)

- Die Analyse, den Entwurf, die Realisierung und die Durchführung von Tests anstoßen
- Geeignete **Metriken** für die Messung des Testfortschritts und die Bewertung der Qualität des Testens und des Produkts einführen
- Den Testfortschritt und die Testergebnisse überwachen und den Stand der Endkriterien (oder der Definition-of-Done) prüfen
- **Testfortschrittsberichte und Testabschlussberichte** auf der Grundlage der gesammelten Informationen erstellen und verteilen
- Die Planung auf der Grundlage von Testergebnissen und Fortschritt anpassen
- notwendige Maßnahmen zur **Teststeuerung** in die Wege leiten

Aufgaben des Testmanagers (3 von 4)

- Das Aufsetzen des Fehlermanagementsystems und eines angemessenen Konfigurationsmanagements für Testmittel unterstützen
- Unterstützung bei der Auswahl und dem Einsatz von Werkzeugen für den Testprozess, einschließlich
 - der Empfehlung des Budgets für die Werkzeugauswahl (und möglicherweise Kauf und/oder Support),
 - der Zuordnung von Zeit und Aufwand für Pilotprojekte und
 - der Bereitstellung von kontinuierlicher Unterstützung bei der Nutzung der Werkzeuge
- Über die Realisierung von Testumgebungen entscheiden

Aufgaben des Testmanagers (4 von 4)

- Die Tester, das Testteam und das Berufsbild Tester innerhalb des Unternehmens entwickeln und fördern
- Die Fähigkeiten und Aufstiegsmöglichkeiten von Testern **weiterentwickeln**
- Die Umsetzung der Rolle des Testmanagers variiert in Abhängigkeit vom Softwareentwicklungslebenszyklus.
- In der **agilen Entwicklung** werden z.B. die täglichen Tests innerhalb des Teams vom agilen Team / **von einem im Team integrierten Tester** übernommen.
- **Test Coaches** (Testmanager außerhalb des Entwicklungsteams)
 - können Aufgaben übernehmen, die mehr als ein Team betreffen / mit Personalmanagement zu tun haben

Aufgaben des Testers (1 von 2)

- Testkonzepte prüfen und zu diesen beitragen
- Testbasis auf Testbarkeit analysieren, prüfen und beurteilen:
 - Anforderungen, User-Stories, Abnahmekriterien, Spezifikationen und Modelle
- Testbedingungen identifizieren und die Rückverfolgbarkeit zwischen Testfällen, Testbedingungen und der Testbasis erfassen
- Testumgebung(en) entwerfen, einrichten und verifizieren
 - oft in Abstimmung mit der Systemadministration und dem Netzwerkmanagement
- Testfälle und Testabläufe entwerfen und realisieren
- Testdaten vorbereiten und beschaffen

Aufgaben des Testers (2 von 2)

- Den detaillierten Testausführungsplan erstellen
- Tests durchführen, die Ergebnisse bewerten und Abweichungen von den erwarteten Ergebnissen dokumentieren
- Geeignete Werkzeuge zur Unterstützung des Testprozesses verwenden
- Bei Bedarf Tests automatisieren, evtl. unterstützt durch Entwickler oder Testautomatisierungsexperten
- Funktionale, wie nicht-funktionale Eigenschaften bewerten
- Tests prüfen, die von anderen entwickelt wurden

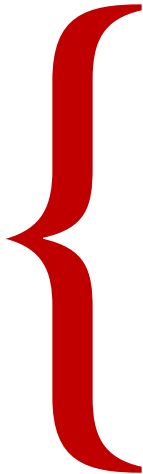


Folgende Fragen sollten Sie jetzt beantworten können

- Welche grundsätzlichen Modelle einer Organisation des Testens lassen sich hinsichtlich Aufgabenteilung zwischen Entwicklung und Test unterscheiden?
- Welches sind die typischen Aufgaben der Rolle Testmanager?
- Welches sind die typischen Aufgaben der Rolle Tester?

Kapitel 5

Test- management



Testorganisation

Testplanung und -schätzung

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

Konfigurationsmanagement



Wann soll mit Testen begonnen werden?



Wann soll mit Testen begonnen werden?

So früh im Projekt wie möglich und alle Phasen des Projekts kontinuierlich begleitend

- Das allgemeine V-Modell sieht deshalb am Ende jeder Entwicklungsphase eine Verifikation des jeweiligen Phasenergebnisses vor
- Viele Analyse- und Designfehler können so früh gefunden und behoben werden
- Die Teststufen im rechten Ast des Modells sind als Phasen der Testdurchführung zu verstehen
- Die zugehörige Testvorbereitung startet früher und wird parallel zu den Entwicklungsschritten im linken Ast durchgeführt
- In einem agilen Modell werden Testaktivitäten während jedes Sprints geplant und durchgeführt

Qualitätssicherungsplan, Testkonzept und Testplan

- Testen soll nicht als einzige Maßnahme zur Qualitätssicherung (QS) eingesetzt werden, sondern immer im Verbund mit anderen QS-Maßnahmen
- Die übergreifende Planung der qualitätssichernden Maßnahmen in einem Projekt wird im Qualitätssicherungsplan dokumentiert
- Das **Testkonzept** legt die Testaktivitäten für Entwicklungs- und Wartungsprojekte fest
- Im **Testplan** wird die zeitliche Planung der Testdurchführung detaillierter beschrieben
- Bei größeren Projekten kann die Planung in einem **Mastertestkonzept** und in separaten Testkonzepten für Teststufen oder für einzelne Testarten erfolgen

Aktivitäten der Testplanung (1 von 3)

- Bestimmung des Umfangs, der Ziele und der Risiken der Tests
- Festlegen der allgemeine Testvorgehensweise:
projektspezifische Teststrategie, Teststufen, Eingangs- und Endekriterien
- Integrieren und Koordinieren der Testaktivitäten in die Softwarelebenszyklusaktivitäten
- Planung der Aktivitäten zur Testanalyse, zum Entwurf, zur Realisierung, zur Durchführung und zur Bewertung
 - in fest definierten Zeiträumen oder im Kontext jeder Iteration
- Treffen von Entscheidungen darüber,
 - was wie intensiv zu testen ist
 - welche Personen und andere Ressourcen notwendig sind, um Testaktivitäten durchzuführen
 - wie die Testaktivitäten durchzuführen sind
 - wie die Testergebnisse zu bewerten sind

Aktivitäten der Testplanung (2 von 3)

- Auswahl der Metriken zur Testüberwachung und -steuerung
- Festlegen des Budgets für die Testaktivitäten
- Festlegung des Umfangs, des Detaillierungsgrads und der Struktur für die Testdokumentation
- Testaktivitäten/Testplanung können im Testkonzept dokumentiert sein
- Der Inhalt von Testkonzepten variiert und kann über die oben genannten Punkte hinausgehen, z.B. durch Vorgaben zu
 - Testautomatisierung
 - Testwerkzeugen
- ISO 29119-3 enthält ein Muster für ein Testkonzept

Aktivitäten der Testplanung (3 von 3)

- Mit dem Fortschritt des Projekts sind immer mehr Informationen und Details verfügbar
- Testplanung ist eine kontinuierliche Aufgabe, die während des gesamten Produktlebenszyklus (auch über das Projekt hinaus – z.B. während der Wartungsphase) durchgeführt wird
- Rückmeldungen aus dem Test können genutzt werden, ...
 - um zu erkennen, wenn sich Risiken ändern,
 - um die Testplanung entsprechend anzupassen.



ISO/IEC/IEEE 29119-3 2013 Part 3: Test Documentation

- Organizational Test Process Documentation
 - Test Policy
 - Organizational Test Strategy
- Test Management Processes Documentation
 - Test Plan
 - Test Status Report
 - Test Completion Report
- Dynamic Test Processes Documentation
 - Test Design Specification
 - Test Case Specification
 - Test Procedure Specification
 - Test Execution Log
 - Test Incident Reporting
- Appendices – Overview and outlines of documents



Welche Rahmenbedingungen beeinflussen das Testen?



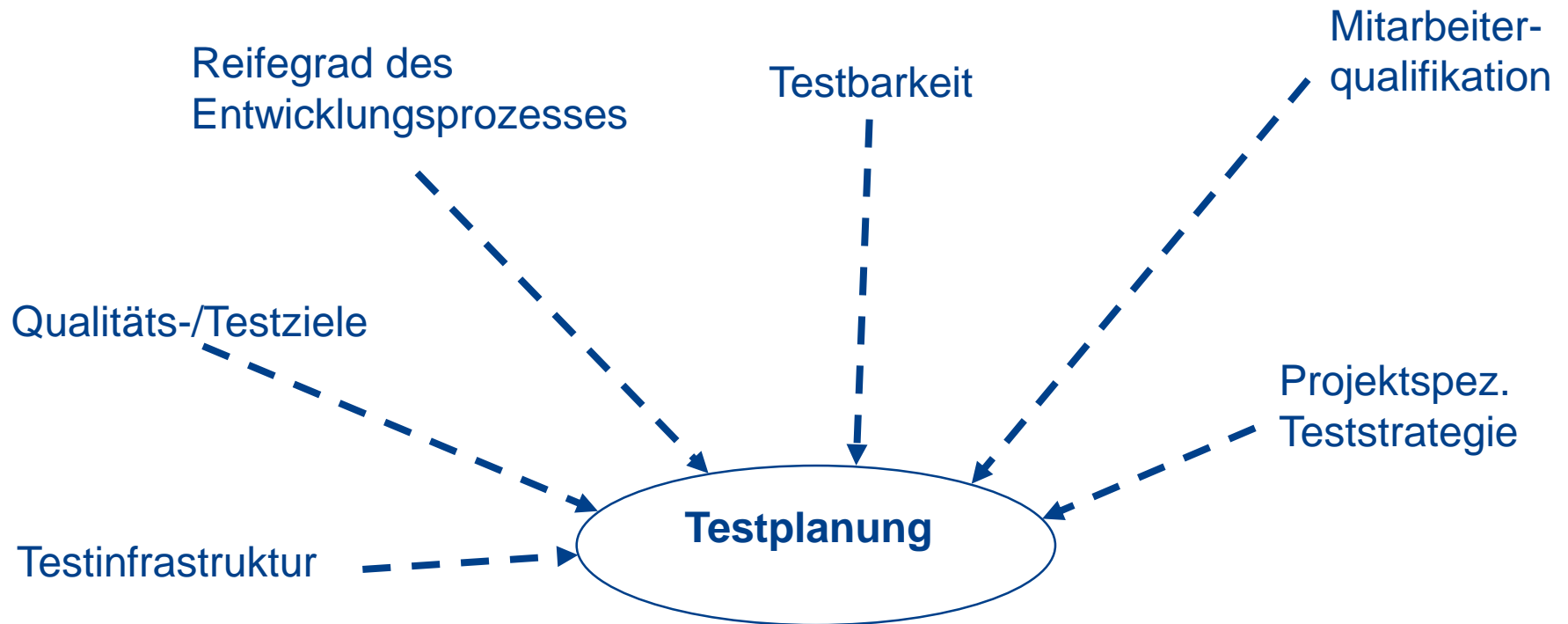
Einflussmöglichkeiten des Testmanagers (1 von 2)

- **Reifegrad des Softwareentwicklungsprozesses**
 - kurzfristig nicht zu beeinflussende, gegebene Größe;
nur langfristig durch Prozessverbesserung beeinflussbar
- **Testbarkeit der Software**
 - vom Reifegrad des Softwareentwicklungsprozesses abhängig
 - Vom Ergebnis des Systementwurfs / der Architektur abhängig
 - gut strukturierter Prozess mit Reviews führt zu besser strukturierter Software, die einfacher zu testen ist
- **Testinfrastruktur**
 - meist vorgegeben, kann aber im Projektverlauf gezielt ausgebaut werden, um punktuell Zeit und Kosten einzusparen

Einflussmöglichkeiten des Testmanagers (2 von 2)

- **Mitarbeiterqualifikation**
 - Kurzfristig: Auswahl des Testpersonals
 - Mittel-/langfristig: Aus- und Weiterbildung
- **Qualitätsziele**
 - Werden vom Kunden und anderen Interessengruppen vorgegeben und sind bedingt beeinflussbar (Priorisierung)
- **Testkonzept / inkl. projektspezifische Teststrategie**
 - Liegt im Verantwortungsbereich des Testmanagers
 - die einzige Stellgröße, die der Testmanager auch kurzfristig beeinflussen und kontrollieren kann

Zu betrachtende Einflussfaktoren

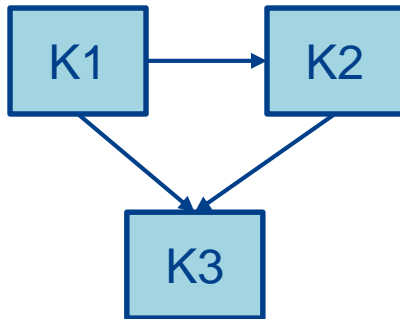


Testausführungsplan

- Definiert die Reihenfolge in der Testsuiten ausgeführt werden
- Einflussfaktoren auf die Ausführungsreihenfolge können sein:
 - Priorisierung (idealerweise werden Testfälle mit der höchsten Priorität zuerst ausgeführt)
 - Abhängigkeiten
 - Fehlernachtests
 - Regressionstests
 - Effizienz bei der Ausführung

Testausführungsplan

Abhängigkeitsgraph



Perfekte Ausführung

Bottom-Up Integrationsstrategie

Nr.	Testsuite	Treiber / Stellvertreter
1	KompTest K3	
2	KompTest K2	K3
3	IntegratTest K2-K3	
4	KompTest K1	
5	IntegratTest K1-K3	K2
6	IntegratTest K1-K2	K3
7	SysTest K1-K2-K3	

Testvorgehensweisen (1 von 4)

Testvorgehensweisen lassen sich klassifizieren basierend auf dem Zeitpunkt, wann mit den Testentwurf begonnen wird:

- **Präventive Vorgehensweise**
 - Tests so früh wie möglich entwerfen
- **Reaktive Vorgehensweise**
 - Es wird nur reagiert auf die Komponente / das System und die während der Testdurchführung auftretenden Ereignisse
 - Tests werden entworfen und implementiert und können umgehend als Reaktion auf vorherige Testergebnisse durchgeführt werden.
 - Exploratives Testen ist eine gängige Vorgehensweise

Testvorgehensweisen (2 von 4)

Typische Ansätze oder Strategien enthalten

- **Analytische Vorgehensweise**

- Beispiel: risikobasiertes Testen mit der Schwerpunktsetzung auf die Bereiche mit den größten Risiken (Analyse des Testobjektes)

- **Modellbasierte Vorgehensweise**

- Tests auf Grundlage eines Modells eines geforderten Produktaspekts entworfen, z.B. Funktion, Geschäftsprozess, interne Struktur oder nicht-funktionales Qualitätsmerkmal
- Beispiele: stochastisches Testen unter Nutzung von statistischen Informationen über Ausfallraten (wie z.B. Zuverlässigkeitswachstumsmodelle) oder Systembenutzung (wie z.B. Benutzungsprofile), weitere Ausgangsmodelle: Geschäftsprozessmodelle, Zustandsmodelle

Testvorgehensweisen (3 von 4)

- **Methodische Vorgehensweise**

- systematische Nutzung vordefinierter Sets von Tests oder Testbedingungen, wie ...
 - einer Systematik von gängigen oder wahrscheinlichen Arten von Fehlerwirkungen,
 - einer Liste von wichtigen Qualitätsmerkmalen,
 - unternehmensweiten Look-and-Feel-Standards für mobile Anwendungen oder Webseiten.

- **Prozesskonforme (oder standardkonforme) Vorgehensweise**

- Analyse, Entwurf und Realisierung von Tests erfolgt auf der Grundlage von externen Vorschriften und Standards
- Beispiele: branchenspezifische Standards, Prozessdokumentation oder unternehmensspezifische Prozesse oder Standards

Testvorgehensweisen (4 von 4)

- **Angeleitete (oder beratende) Vorgehensweise**
 - das Testen wird durch Beratung, Anleitung und Anweisungen von Stakeholdern, Fachexperten oder Technologieexperten (auch von außerhalb des Testteams oder des Unternehmens) bestimmt
- **Leistungserhaltende Vorgehensweise**
 - durch den Wunsch motiviert, einen Rückgang bei den vorhandenen Leistungsfähigkeiten zu vermeiden
 - beinhaltet die **Wiederverwendung** vorhandener Testmittel, weitgehende **Automatisierung von Regressionstests** und die Nutzung von **Standardtestsuiten**.

Eine Kombination der unterschiedlichen Ansätze ist sinnvoll, z.B. ein risikoorientierter dynamischer Ansatz

Wovon hängt die Auswahl der Testvorgehensweise ab?

- Art und Komplexität des Produkts und des Geschäftsfelds, Technologie
- Risiko des Scheiterns des Projekts
- Risiken für das Produkt und Risiken für Personen, die Umwelt und das Unternehmen bei Produktfehlern bzw. -ausfällen
- Qualifikation und Erfahrung der Personen in den einzusetzenden Techniken, Werkzeugen und Methoden
- Testziele und Auftrag des Testteams
- Für den Entwicklungsprozess einzuhaltende Regularien

Empfehlungen:

Wählen Sie einen Mix von Vorgehensweisen aus, welches ein optimales Verhältnis zwischen Testkosten, verfügbaren Ressourcen und drohenden Fehlerkosten darstellt

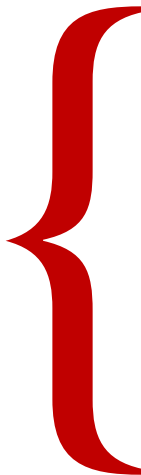


Folgende Fragen sollten Sie jetzt beantworten können

- Was ist der Zweck und Inhalt eines Testkonzepts?
- Wozu dienen im Testmanagement sogenannte Eingangskriterien?
- Wozu dienen im Testmanagement sogenannte Endekriterien?
- Welches sind typische Beispiele von Endekriterien?
- Welche Endekriterien für einen zustandsbezogenen Test gibt es?
- Warum sollte man Tests in einer separaten Umgebung durchführen?
- Warum priorisiert man Testfälle? Was sind Kriterien für eine Priorisierung?

Kapitel 5

Test- management



Testorganisation

Testplanung und **-schätzung**

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

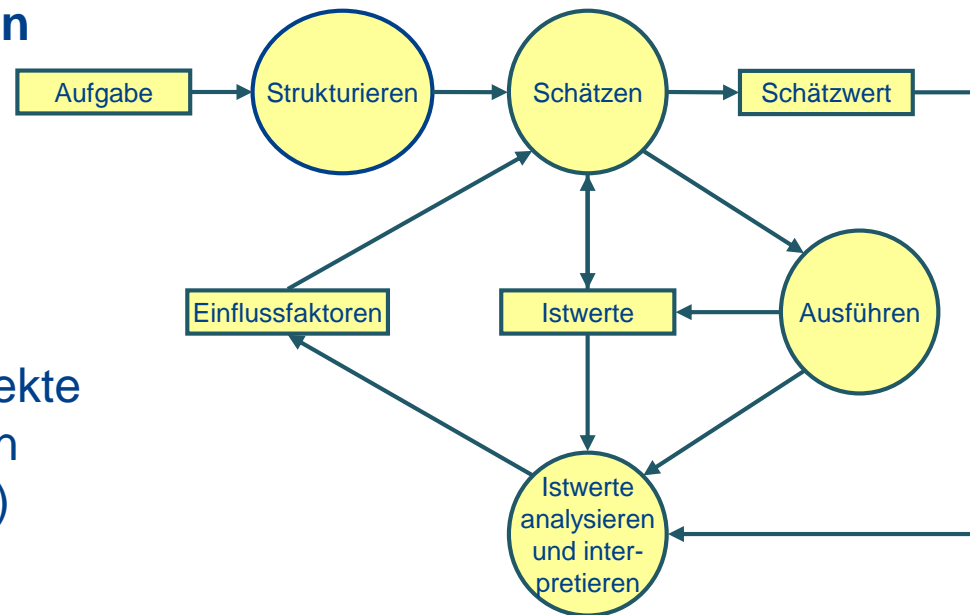
Konfigurationsmanagement

\pm Was beeinflusst den Testaufwand?



Schätzung des Testaufwands

- Voraussage über die Menge an testrelevanter Arbeit, die benötigt wird, um die Ziele des Testens für ein bestimmtes Projekt, ein Release oder eine Iteration zu erfüllen.
- Zwei Ansätze für die Schätzung des Testaufwands
 - **Expertenbasierte Verfahren**
durch Verantwortliche für diese Aufgaben oder externe Experten
 - **Metrikbasierte Verfahren**
auf Basis von Metriken früherer oder ähnlicher Projekte oder auf Basis von typischen Werten (Analogieschätzung)



Expertenbasierte Verfahren (1 von 2)

- **Einzelschätzung**

- Experte (oft der Testmanager oder erfahrene Tester) schätzt den Aufwand auf Basis der Aufgabenbeschreibung und seiner Erfahrung



- **Mehrfachbefragung**

- Es werden mehrere Schätzer, möglichst aus unterschiedlichen organisatorischen Bereichen, befragt und ein Mittelwert gebildet oder (bei größeren Unstimmigkeiten) diskutiert und ein Konsens gesucht



Expertenbasierte Verfahren (2 von 2)

- **Planungspoker**

- Teammitglieder schätzen den Aufwand, der ihrer Erfahrung nach notwendig ist, um eine Funktionalität bereitzustellen



- **Delphi-Schätztechnik**

- Formalisierte, schriftliche Befragung mehrerer Experten
- Zwei oder mehr Befragungsrunden mit Sitzungen zur Diskussion der Zwischenergebnisse der vorausgegangenen SchätZRunde

- **Schätzsitzung**

- Übertragung des Delphi-Prinzips auf eine einzige Gruppensitzung

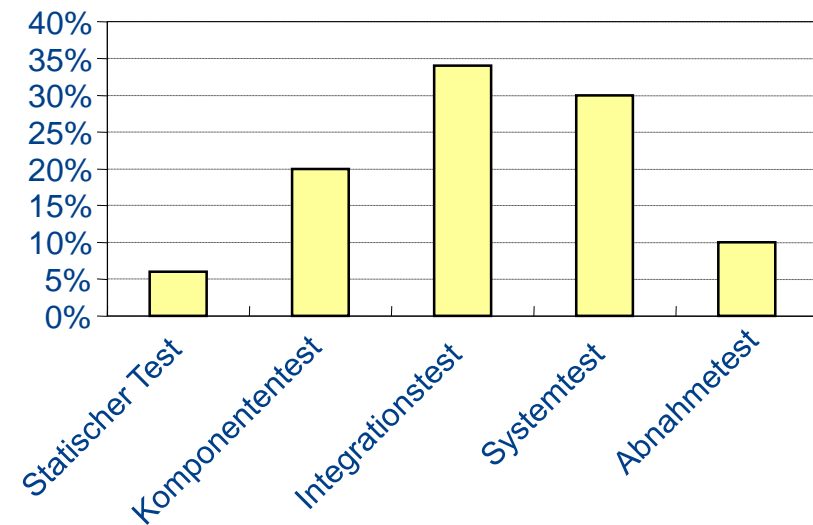
Bildquelle: <https://technology.amis.nl/2016/03/23/8-agile-estimation-techniques-beyond-planning-poker/#prettyPhoto>

Metrikbasierte Verfahren

- Vergleich des zu schätzenden Testprojekts mit einem oder mehreren bereits abgeschlossenen, ähnlichen Projekten
 - Gleiches oder ähnliches Anwendungsgebiet bzw. Aufgabenstellung
 - Gleiche oder ähnliche Produktgröße
 - Gleiche oder ähnliche Randbedingungen (z.B. Projektteam)
- Unterschiede in Aufgabenstellung und Realisierungsbedingungen werden vom Schätzer aufgrund seiner Erfahrung intuitiv berücksichtigt

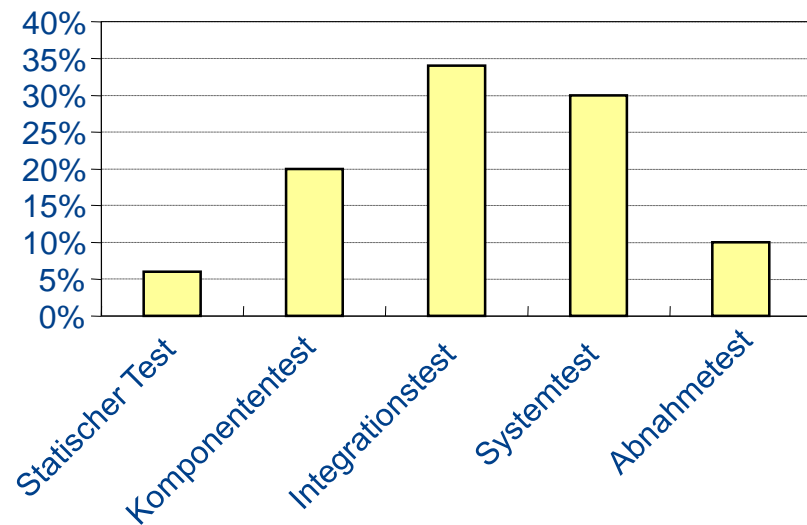
Prozentsatzmethode (1 von 2)

Aus früheren Projekten wird ermittelt, wie sich Aufwand und Projektzeit prozentual auf die einzelnen Testaktivitäten verteilen



- **Voraussetzung:**
 - Historische Daten
 - Einheitlicher Testprozess
- 2-stufige **Top-Down-Schätzung** des Aufwands
 - 1. Stufe: Schätzung des Testaufwands im Verhältnis zum Gesamtaufwand bzw. zum Entwicklungsaufwand
 - 2. Stufe: Verteilung der Testaufwände auf die Teststufen

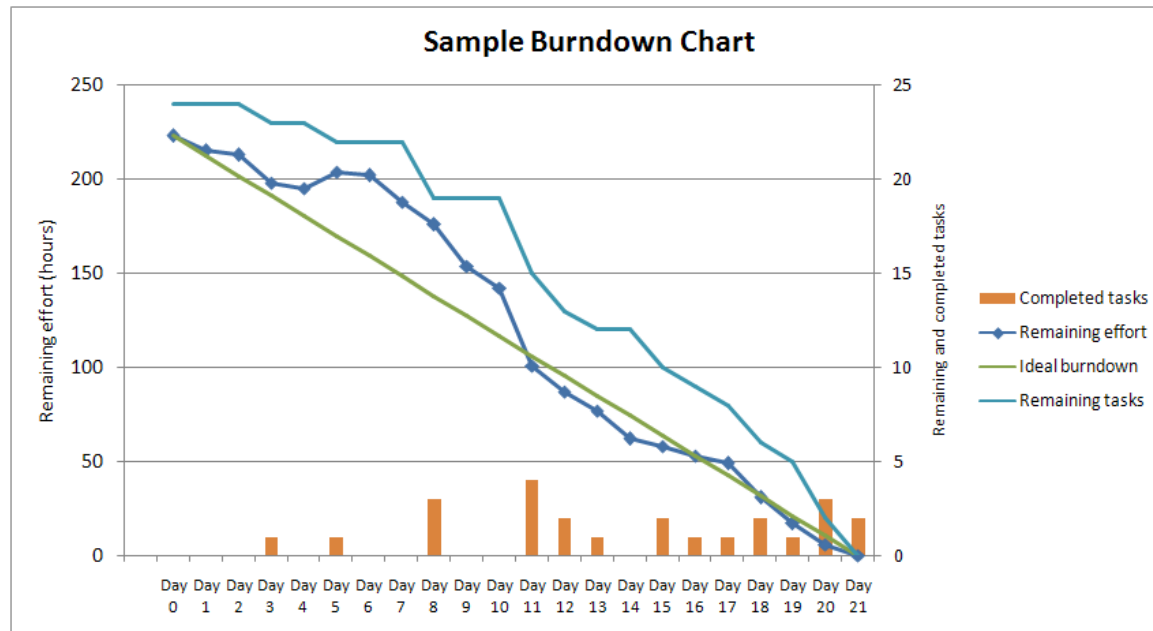
Prozentsatzmethode (2 von 2)



- Zwei Varianten für **Bottom-Up-Schätzungen**
 - Eine/mehrere Projektphasen sind vollständig abgeschlossen, der Ist-Aufwand wird errechnet und prozentual als Soll für die restlichen Phasen hochgerechnet
 - Für eine Phase/Aktivität (z.B. Programmierung) wird eine detaillierte Mikroschätzung durchgeführt und daraus wird auf das Gesamtprojekt geschlossen
- Zu Projektbeginn eher für Kontroll-Checks sinnvoll, da hohes Fehlerrisiko bei Hochrechnung aus niedrigen Anfangswerten besteht

Burndown Charts

- In der agilen Entwicklung sind Burndown-Charts Beispiele für ein metrikbasiertes Vorgehen.
- Der Aufwand wird bestimmt und dann zusammen mit der Velocity (Maß für Produktivität im agilen Projekt) des Teams genutzt, um die Menge an Arbeit zu ermitteln, die das Team in der nächsten Iteration erfüllen kann.



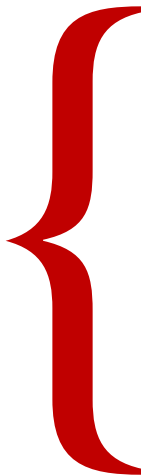


Folgende Fragen sollten Sie jetzt beantworten können

- Welche Faktoren beeinflussen den Testaufwand?
- Kennen Sie zwei konzeptionell unterschiedliche Verfahren zur Schätzung des Testaufwands?
- Stellen Sie sich vor Sie sind in einem Projekt als Testmanager eingesetzt. Kennen Sie min. 5 Aktivitäten der Testplanung, die sie im Projekt zu bewältigen haben?

Kapitel 5

Test- management



Testorganisation

Testplanung und -schätzung

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

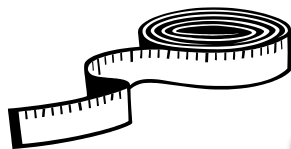
Konfigurationsmanagement

Testfortschrittsüberwachung (1 von 3)

Warum Testmetriken: Wo stehen wir? Wann sind wir fertig?

Testmetrik

Messbare Eigenschaft eines Testfalls, Testlaufs oder Testzyklus mit Angabe der zugehörigen Messvorschrift.



Metriken

Fehlerbasierte Metriken:
z.B. Zahl gefundener Fehlerzustände

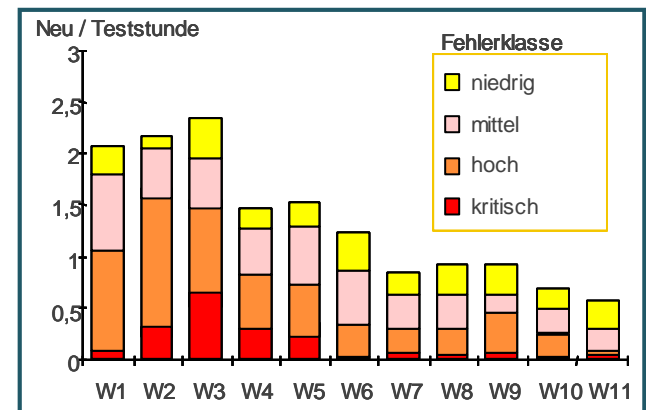
Testfallbasierte Metriken:
z.B. Anzahl geplanter Tests

Testobjektbasierte Metriken:
z.B. Codeüberdeckung

Kostenbasierte Metriken:
z.B. Kosten vs. Nutzen für das Auffinden
des nächsten Fehlers

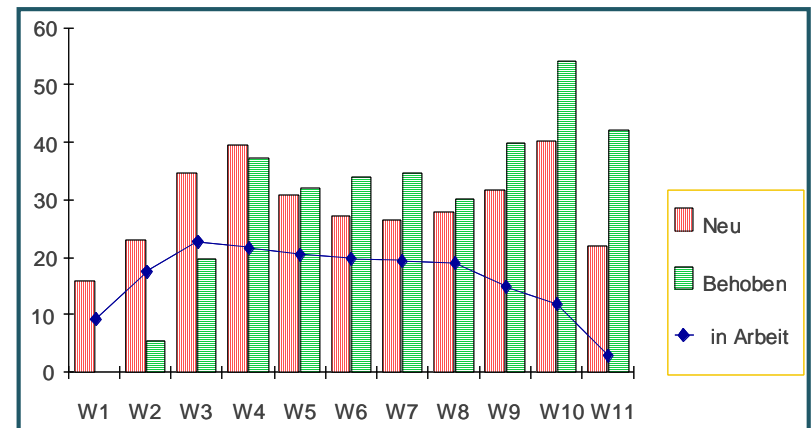
Testfortschrittsüberwachung (2 von 3)

- Überwachung und Erfolgskontrolle anhand objektiver Testmetriken und darauf aufbauender Endekriterien (Definition-of-Done im agilen Kontext)
- Dabei ist darauf zu achten, dass nur solche Metriken verwendet werden, die regelmäßig, zuverlässig und einfach zu messen sind, und solche deren Aussagekraft klar sind und wenig Interpretationsspielräume lassen
- Gebräuchliche Testmetriken sind u.a.
 - **Fehlerbasierte Metriken**
z.B. Fehlerdichte, gefundene und behobene Fehlerzustände, Fehleraufdeckungsrate, Ausfallrate und Nachtestergebnisse



Testfortschrittsüberwachung (3 von 3)

- **Testfallbasierte Metriken**, z.B.
 - Verhältnis der bereits spezifizierten/geplanten Testfälle
 - Anzahl der durchgeführten/nicht durchgeführten Testfälle
 - Anzahl der bestandenen/fehlgeschlagenen Testfälle
- **Testobjektbasierte Metriken**, z.B.
 - Testabdeckung der Anforderungen, User Stories, Abnahmekriterien der Risiken oder des Codes
- **Kostenbasierte Metriken**, z.B.
 - Vergleich der Kosten für das Auffinden des nächsten Fehlers und den nächsten Testdurchlauf
- Subjektives Vertrauen der Tester in das Produkt (nicht messbar!)



Standortbestimmung

- Die gemessenen Daten dienen zur Standortbestimmung und helfen, folgendes zu beantworten:
 - Fortschritt gegenüber dem geplanten Zeitplan und Budget
 - Gegenwärtige Qualität des Testobjekts
 - Angemessenheit der Testvorgehensweise
 - Effektivität der Testaktivitäten in Bezug auf die Ziele
 - Kann der Test beendet und das Produkt ausgeliefert werden?
- Auswahl der Kriterien zur Bestimmung des Testendes hängt von zu erfüllenden Qualitätsanforderungen und verfügbaren Testressourcen ab
- im Projekt geltenden Endekriterien werden im Testkonzept festgelegt
- Endekriterien müssen sich aus erhobenen Testmetriken berechnen lassen

Eingangs- und Endekriterien (1 von 3)

(Definition-of-Ready und Definition-of-Done)

- **Eingangskriterien** (in der agilen Entwicklung üblicherweise als **Definition-of-Ready** bezeichnet)
 - Vorbedingungen für die Durchführung einer bestimmten Testaktivität
 - Wenn Eingangskriterien nicht erfüllt dann wird die Aktivität vermutlich schwieriger, zeitaufwändiger, kostspieliger und risikoreicher
- **Endekriterien** (in der agilen Entwicklung üblicherweise **Definition-of-Done** genannt)
 - Bedingungen, damit eine Teststufe oder ein Set von Testfällen als abgeschlossen bezeichnet werden kann
- Eingangs- und Endekriterien sollten für jede Teststufe und jede Testart definiert sein und unterscheiden sich nach den Testzielen.



Was sind sinnvolle Eingangskriterien?



Eingangs- und Endekriterien (2 von 3)

(Definition-of-Ready und Definition-of-Done)

- Typische **Eingangskriterien** sind z.B.
 - Verfügbarkeit von testbaren Anforderungen, User-Stories und/oder Modellen (bspw., wenn eine modellbasierte Teststrategie verfolgt wird)
 - Verfügbarkeit von Testelementen, die die Endekriterien für vorangegangene Teststufen erfüllt haben
 - Verfügbarkeit einer Testumgebung
 - Verfügbarkeit der notwendigen Testwerkzeuge
 - Verfügbarkeit von Testdaten und anderen notwendigen Ressourcen

+/- Was sind sinnvolle Endekriterien?



Eingangs- und Endekriterien (3 von 3)

(Definition-of-Ready und Definition-of-Done)

- Typische **Endekriterien** sind z.B.
 - Geplante Tests wurden durchgeführt
 - Überdeckungsmaße sind erfüllt, z.B. Abdeckung von Code, Anforderungen, User Stories, Abnahmekriterien oder Risiko
 - Die Anzahl ungelöster Fehlerzustände bewegt sich innerhalb einer vereinbarten Grenze
 - Schätzung der (verbleibenden) Fehlerdichte oder Zuverlässigkeitsschätzung
 - Umfang der verbleibenden Risiken, wie z.B. erkannte aber nicht behobene Fehler oder fehlende Testüberdeckung in einzelnen Softwareteilen
 - Kosten, Zeit, z.B. Termin der Auslieferung oder Markteinführung
 - Die bewerteten Niveaus an Zuverlässigkeit, Performanz, Gebrauchstauglichkeit, Zugriffssicherheit und anderer relevanter Qualitätsmerkmale sind ausreichend

Testfortschrittsüberwachung

- Testfortschrittsüberwachung liefert Rückmeldung und Übersicht über Testaktivitäten.
- Die zur Messung der Endekriterien benötigte Information kann manuell oder automatisiert gesammelt werden.
- Metriken können ermittelt werden, um den Fortschritt im Zeitplan und die Einhaltung des Budgets zu beurteilen.
- Begrenzung der Testaktivitäten ohne Erreichen der Endekriterien?
 - ausgeschöpftes Budget
 - Ablauf der geplanten Zeitdauer
 - Drucks, das Produkt auf den Markt zu bringen
- Ende des Testens unter derartigen Umständen möglich, wenn Stakeholder und Fachverantwortliche das Risiko, ohne weitere Tests die Freigabe zu erteilen, geprüft und akzeptiert haben.

Testbericht (1 von 3)

- Zweck des Testberichts ist es, Informationen über die Testaktivitäten zusammenzufassen und zu kommunizieren
- Es gibt zwei Arten von Testberichten
 - **Testfortschrittsbericht**
=> Bericht für den Fortschritt während einer Testaktivität
 - **Testabschlussbericht**
=> Bericht am Ende einer Testaktivität, z.B. einer Teststufe
- Der Testmanager erstellt regelmäßig Testfortschrittsberichte
- Wenn die Endekriterien erreicht sind, erstellt er einen Testabschlussbericht auf Basis der letzten Fortschrittsberichte sowie weiterer Informationen

Testbericht (2 von 3)

- Inhalte eines Testberichts variieren in Abhängigkeit vom
 - Projekt
 - den organisatorischen Anforderungen
 - Softwareentwicklungslebenszyklus
- Beispiele
 - „regulatorisches Umfeld“: detaillierte, genaue Berichte
 - „agile Entwicklung“: Tasks Boards, Burndown-Charts, Stand-up-Meeting

Testbericht (3 von 3)

- Die Inhalte eines Testberichts sollten an die Zielgruppe des Berichts angepasst werden
- Beispiele:
 - Testteam: z.B. detaillierte Informationen über Fehlerarten und Fehlertrends
 - Management: Informationen auf abstrakter Ebene, KPIs zu Testfortschritt, Fehlern und Kosten
- ISO 29119-3 kennt die zwei Arten von Testberichten und beinhaltet für beide Arten Beispiele und Inhaltsstrukturen

Typische Inhalte eines Testberichts

- Status der Testaktivitäten und Fortschritt gegenüber dem Testkonzept
- Informationen darüber, was in der Testperiode vorgefallen ist
- Metriken über Fehlerzustände, Testfälle, Testüberdeckung, Aktivitätsfortschritt und Ressourcenverbrauch
- Stand der Tests und Produktqualität in Bezug auf Endekriterien oder Definition-of-Done
- Abweichungen vom Plan inkl. Abweichungen vom Zeitpunkt, Dauer und Aufwand von Testaktivitäten
- Faktoren, die den Fortschritt blockiert haben oder weiterhin blockieren
- Restrisiken
- Erzeugte wiederverwendbare Testarbeitsergebnisse

Teststeuerung (1 von 4)

- Bei Verzögerungen gegenüber der Projekt- oder Testplanung geeignete Gegenmaßnahmen einleiten, z.B.
 - Bei (neu) erkannten Risiken Änderung der Priorisierung der Tests (z.B. verspätete Lieferung von Softwareteilen)
 - Anpassung der zeitlichen Planung, wenn sich die Verfügbarkeit der Testumgebung verzögert
 - Korrigierte Fehlerzustände sind durch einen Entwickler nachzutesten, bevor die Software weiter verwendet wird

Teststeuerung (2 von 4)

- ggf. zusätzliche Testressourcen (Personal, Arbeitsplätze, Werkzeuge) anfordern und einzusetzen, um einen Rückstand gegenüber dem Testplan in den verbleibenden Zyklen aufzuholen
- Stehen keine zusätzlichen Ressourcen zur Verfügung, muss der Testplan angepasst werden, z.B.
 - Mit niedriger Priorität eingestufte Testfälle werden gestrichen
 - Eine weitere Möglichkeit ist, Testfälle, die in verschiedenen Varianten geplant sind, nur in einer Variante durchzuführen (z.B. werden Tests nur unter einem Betriebssystem ausgeführt, statt wie geplant unter verschiedenen)
 - Durch solche Anpassungen werden zwar einige interessante Tests nicht absolviert, aber die gesparten Ressourcen ermöglichen es, wenigstens die Testfälle hoher Priorität durchzuführen

Teststeuerung (3 von 4)

- Je nachdem wie gravierend die aufgedeckten Mängel und Probleme sind, ist die Testdauer zu verlängern, weil zusätzliche Testzyklen notwendig sind
 - Nach jedem Korrekturzyklus muss die geänderte Software erneut getestet werden
 - Kann zur Verschiebung der Markteinführung bzw. der Auslieferung des Softwareprodukts führen
- Wichtig ist, dass der Testmanager jede solche Planänderung dokumentiert und kommuniziert
 - Änderung am ursprünglichen Testplan bedeutet in aller Regel eine Erhöhung des Freigaberisikos
- Es ist Aufgabe des Testmanagers, dieses Risiko den Projektverantwortlichen zu jedem Zeitpunkt klar und offen darzulegen



Teststeuerung (4 von 4) : Zusammenfassung

- Teststeuerung beschreibt sämtliche Führungs- oder Korrekturmaßnahmen, die auf Grund gesammelter oder berichteter Informationen und Metriken ergriffen werden.
- Maßnahmen können jede Testaktivität betreffen und können jede andere Softwarelebenszyklusaktivität oder -aufgabe beeinflussen.
- Beispiele von Maßnahmen zur Teststeuerung sind:
 - Repriorisierung von Tests, wenn identifizierte Risiken auftreten
 - Änderung des Testzeitplans aufgrund der Verfügbarkeit der Testumgebung
 - Setzen einer Eingangskriterium mit der Maßgabe, dass Fehlerbehebungen durch einen Entwickler nachzutesten sind, bevor die Software an die nächste Teststufe übergeben wird

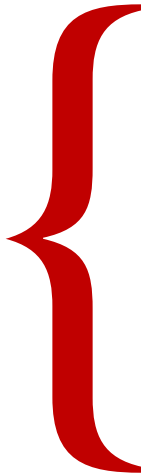


Folgende Fragen sollten Sie jetzt beantworten können

- Welche Arten bzw. Kategorien von Metriken zur Überwachung des Testfortschritts lassen sich unterscheiden?
- Welche Informationen soll ein Testfortschrittsbericht enthalten?
- Können Sie fehlerbasierte Metriken erklären?

Kapitel 5

Test- management



Testorganisation

Testplanung und -schätzung

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

Konfigurationsmanagement



Was ist ein Risiko?



Was ist ein Risiko?

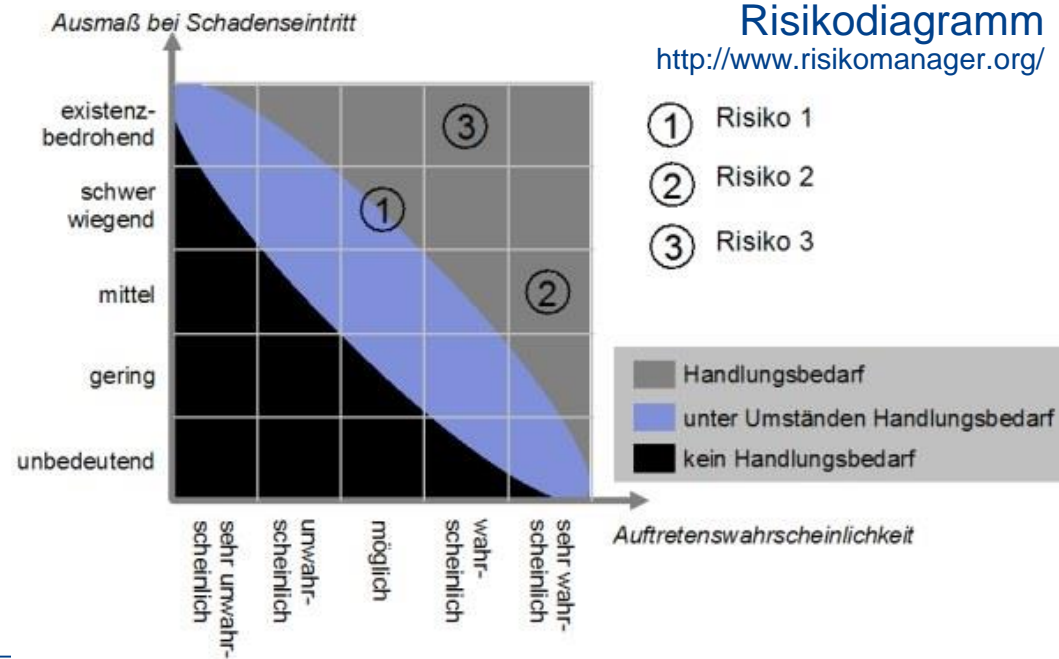
No Risk – No Fun!



- Risiko
 - Problem, das in der Zukunft eintreten könnte und unerwünschte Folgen hat; ein Faktor, der zu negativen Konsequenzen in der Zukunft führen könnte
- Die Höhe eines Risikos (R) ergibt sich aus der Eintrittswahrscheinlichkeit (W) und dem damit verbundenen Schadensausmaß (S):

$$R = W \cdot S$$

- Risiken unterteilen sich in
 - Projektrisiken
 - Produktrisiken/
Qualitätsrisiken



Was sind Risiken?

No Risk – No Fun!



- **Risiken** sind Probleme, die sowohl während der Entwicklung als auch bei dem Einsatz des Produkts möglicherweise eintreten können und unerwünschte Folgen haben.
- **Projektrisiken** beziehen sich auf Management und Steuerung eines (Test-) Projekts, z.B. Mangel an personellen Ressourcen, zu enge Zeitrahmen, sich ändernde Anforderungen, usw.
- **Produktrisiken** sind direkt auf das Software-Projekt bezogen und resultieren oft aus Qualitätsmängeln.

Projektrisiken

- Risiken bezüglich der Erreichung der Projektziele
- Projektrisiken können sowohl Entwicklungsaktivitäten als auch Testaktivitäten beeinflussen.
- In einigen Fällen sind Projektmanager für den Umgang mit allen Projektrisiken verantwortlich, aber es ist nicht ungewöhnlich, dass Testmanager die Verantwortung für testbezogene Projektrisiken tragen.



Was ist aktuell ein bekanntes Projektrisiko?





Beispiele für Projektrisiken (1 von 3)

- **Projektprobleme:**

- Verzögerungen können in der Lieferung, Aufgabenerfüllung oder Erfüllung von Endkriterien oder der Definition-of-Done auftreten.
- Nicht ausreichender Finanzierung durch ungenaue Schätzungen, Neuverteilung von Mitteln an höher priorisierte Projekte oder generelle Kosteneinsparungen im Unternehmen.
- Erheblichen Überarbeitungen durch späte Änderungen

- **Unternehmensprobleme:**

- Fähigkeiten, Schulungen und Mitarbeiter sind möglicherweise nicht ausreichend.
- Personalprobleme, die Konflikten und Probleme auslösen können
- Benutzer, Mitarbeiter der Fachabteilung oder Fachexperten sind unter Umständen aufgrund von gegenläufigen Geschäftsprioritäten nicht verfügbar.



Beispiele für Projektrisiken (2 von 3)

- **Politische Probleme**

- Kommunikationsprobleme mit Testern bzgl. ihrer Erfordernisse und Ergebnisse
- Nicht Nutzung von Erkenntnissen, die beim Testen oder in Reviews gefunden werden,
z.B. keine Umsetzung von erkannten Verbesserungen der Entwicklungs- und Testpraktiken
- Nicht angemessene Haltung gegenüber dem Testen allgemein oder nicht angemessene Erwartungen an das Testen,
z.B. Unterschätzung des Nutzens der Fehleraufdeckung durch das Testen

- **Lieferantenprobleme**

- Versagen einer dritten Partei
- Vertragsaspekte



Beispiele für Projektrisiken (3 von 3)

- **Technische Probleme**

- Beschreibung der Anforderungen ist zu ungenau
- Erfüllung der Anforderungen unter den gegebenen Randbedingungen hat zu viele Auslegungsmöglichkeiten
- Nicht rechtzeitig verfügbare Testumgebung
- Verspätete Lieferung von Werkzeugen
- Schwächen im Entwicklungsprozess, die Auswirkungen auf die Qualität von Projektergebnissen wie Entwurf, Code, Konfiguration, Testdaten und Testfälle haben
- Schlechtes Fehlermanagement können zu kumulierten Fehlerzuständen und anderen technischen Schulden führen

Produktrisiken

- Risiken bezüglich des Testobjekts (und seines Einsatzes)
 - Produkt könnte Geräten, Menschen oder Organisationen Schaden zufügen
 - Produkt ermöglicht bzw. unterstützt kriminelle Handlungen (Viren, ...)
- Wenn die Produktrisiken mit spezifischen Qualitätsmerkmalen eines Produkts in Verbindung stehen, werden Produktrisiken auch Qualitätsrisiken genannt



Beispiele für Produktrisiken

- Die Software führt vielleicht nicht die gewünschten Funktionen gemäß ihrer Spezifikation aus.
- Die Software führt vielleicht nicht die gewünschten Funktionen gemäß den Bedürfnissen von Benutzern, Kunden und/oder Stakeholdern aus.
- Rückmeldungen zum Benutzererlebnis könnten die Produkterwartungen nicht erfüllen.
- Eine bestimmte Berechnung könnte unter bestimmten Bedingungen nicht korrekt durchgeführt werden.
- Ein Regelkreisverfahren könnte nicht korrekt gebaut sein.
- Eine Systemarchitektur könnte einige nicht-funktionale Anforderungen nicht angemessen unterstützen.
- Die Antwortzeiten könnten für ein hochleistungsfähiges Abwicklungssystem nicht angemessen sein.



Beispiel: Risikoliste

Risikoliste		Projekt: CarConfig					Version: 1.8e	Vom: 15. Juni 2006		
	Risikobewertung						Risikostatus			
ID	Bezeichnung / Eigentümer	W	S	RPZ	Indikator	Trigger	Akt. wert	Maßnahmen	Verant- wortl.	Einge- leitet am
R7	Verlust der Konfig- Management- Datenbank / Sc	9	500	4500	Antwortzeit	5 s	1	Neuer Server	Mü	-
					Recovery	2/T	0,1			
R2	Verzögerung in der Entwicklung / Me	9	100	900	Meilenstein- verzug	14 T	7	Testplan angleichen	Sc	10.06.06
					Krankenstand	3	3	Testpersonal abordern	Wy	15.06.06
R4	Unzureichende Eingangsqualität der Testobjekte / Sc	9	10	90	Fail-Eing. Test	3/T	2	Code-Reviews intensivieren	Me	-
					Fehlerrate	15/T	12			
...		

Legende

W - Eintrittswahrscheinlichkeit (1 = vernachlässigbar, ... , 9 = sehr hoch)

S - Schadensausmaß (nichtlineare Abbildung auf (Geld-)Werte: z.B. 1, 10, 50, 100, 500, 1000)

RPZ - Risikoprioritätszahl ($RPZ = S * W$)

Indikator - Messgröße zur Ankündigung eines bevorstehenden Risikoeintritts

Trigger - Schwellwert mit Einleitung der Maßnahmen zur Risikobekämpfung
(z.B. Fail-Eing.Test - bei mehr als 3 fehlgeschlagenen Eingangstest pro Tag
sind als Maßnahme vermehrt Code-Reviews durchzuführen.)

Tabelle aus

Spillner, Roßner, Winter, Linz:

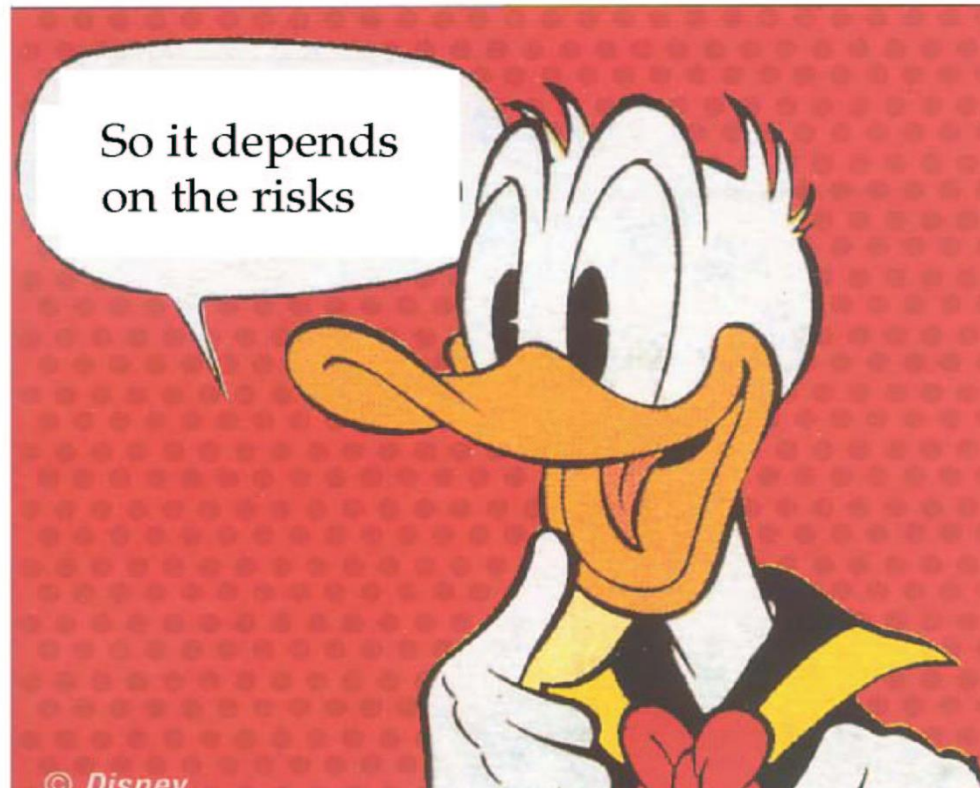
Praxiswissen Softwaretest – Testmanagement.
dpunkt verlag, Heidelberg, 2006, S. 191

Risikomanagement

- Risikomanagement ist die systematische Anwendung von Praktiken für die Aufgaben der Risikoidentifizierung, Risikoanalyse, Risikopriorisierung und Risikosteuerung zur Verhinderung von Fehlschlägen
- Aktivitäten:
 - Ermittlung des Risikokontextes
 - Risikoidentifikation
 - Risikoanalyse und -priorisierung
 - Risikosteuerung und -bewältigung
 - Risikoüberprüfung und -überwachung

Testmanagement: Risikoorientiertes Testen

How much is enough ??



Quelle (2013): <http://kehittaminen.turkuamk.fi/msuni/Testing/TestingCourse03-01-slides.pdf>

Prioritise tests so that
whenever you **stop testing**

you have done the **best testing**
in the **time available**.

Quelle (2013): <http://kehittaminen.turkuamk.fi/msuni/Testing/TestingCourse03-01-slides.pdf>

Risikoorientierter Testansatz

- Ansatz zur Reduzierung des Produktrisikos, beginnend mit Projektstart
- Identifikation von Produktrisiken und ihre Berücksichtigung bei der Steuerung der Testplanung, sowie der Erstellung und Durchführung von Tests
- Identifizierte Risiken bestimmen
 - die einzusetzenden Testverfahren und den Umfang der Tests
 - die einzelnen Teststufen und Testarten, die durchzuführen sind
 - die Priorisierung der Tests und Ressourcen mit dem Ziel, kritische Fehlerzustände so früh wie möglich zu finden
 - ob zusätzlich zum Testen weitere Tätigkeiten zur Risikoreduktion notwendig sind
- Risikoorientiertes Testen bietet Auskunft über verbleibende Risiken (Restrisiken), durch den Nachweis der Effektivität der Behebung von kritischen Fehlern

Risikoorientiertes Testen und Risikomanagement

- Beim risikoorientierten Testen werden das Wissen und die Kenntnisse der Stakeholder genutzt, um Risiken zu identifizieren und die erforderlichen Tests zum Behandeln der Risiken zu bestimmen
- Risikomanagement hilft die Wahrscheinlichkeit von Produkt- und Projekt-Problemen zu minimieren und bietet einen systematischen Ansatz für
 - die Beurteilung und regelmäßiger Neubewertung von Fehlerwirkungen, die auftreten können
 - die Entscheidung, welche Risiken behandelt werden müssen
 - die Umsetzung von Maßnahmen zur Behandlung bzw. Behebung dieser Risiken
- Zusätzlich kann Testen
 - die Identifikation neuer Risiken unterstützen
 - helfen festzulegen, welche Risiken reduziert werden sollen
 - die Unsicherheit bezüglich der Risiken verringern



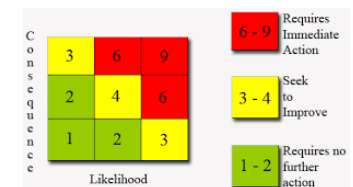
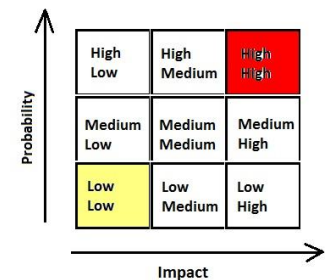
Beispiel: Risikoorientierte Testplanung

Risikoklasse	Endekriterien
hoch	100% Black-box Tests ausgeführt 95% fehlerfrei, kein Prio-0-Fehler Min. 85% Zweigüberdeckung
mittel	80% Black-box Tests ausgeführt, 80% fehlerfrei, max. ein Prio-0-Fehler, 75% Zweigüberdeckung
gering	70% Black-box Tests ausgeführt, 80% fehlerfrei, max. 3 Prio-0-Fehler, 50% Zweigüberdeckung



Risikoorientiertes Testen (Zusammenfassung)

- **Zielgerichtet testen**, indem Systemfunktionen je nach Risikostufe mit unterschiedlichen Testverfahren und „Testtiefen“ abgedeckt sowie bei der Fehlerbehebung gesondert betrachtet werden.
- **Priorisiert testen**, wobei Bereiche mit höherem Risiko bei der Testplanung eine höhere Priorität erhalten und entsprechend frühzeitig und intensiv getestet werden.
- In den Testberichten die **Restrisiken beziffern**, die bei einer Auslieferung der Software trotz Verkürzung des Tests oder Verzicht auf die Ausführung geplanter Tests verbleiben.



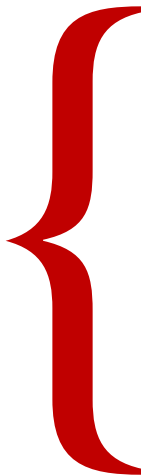


Folgende Fragen sollten Sie jetzt beantworten können

- Welches sind die Aktivitäten des Risikomanagements?
- Wie berechnet man ein Risiko?
- Was sind typische Produkt- und Projektrisiken?
- Welcher Ansatz wird durch ein risikoorientiertes Testen verfolgt?

Kapitel 5

Test- management



Testorganisation

Testplanung und -schätzung

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

Konfigurationsmanagement

Fehler- und Abweichungsmanagement

- gefundenen Mängel, Probleme und Fehlerwirkungen müssen zuverlässig und schnell korrigiert werden
 - Daher: gut funktionierendes Verfahren zur Übermittlung und Verwaltung der Fehlermeldungen
- Ein solches Verfahren wird als Fehlermanagement oder Abweichungsmanagement bezeichnet
- Auf diesen Prozess müssen sich alle am Fehlermanagement beteiligten Parteien einigen
 - Designer, Entwickler, Tester und Product Owner
 - In einigen Organisationen kann die Aufzeichnung und Nachverfolgung von Fehlerzuständen sehr informell sein.
- Dieses Verfahren beginnt nach Ende eines Testlaufs und umfasst
 - Testprotokoll
 - Fehlermeldung (allgemein: Abweichungsbericht)
 - Fehlerklassifikation
 - Fehlerstatus

Testprotokoll

- Auswertung der angefallenen Testprotokolle
 - Abweichungen zwischen vorausgesagtem Ergebnis und beobachtetem Istergebnis
 - Die Tester vergewissern sich, ob tatsächlich eine Abweichung zum vorausgesagtem Ergebnis vorliegt oder ob die Ursache ein falsch entworfener Testfall, eine fehlerhafte Testautomatisierung oder eine falsche Testausführung war
- Fehler dokumentieren
 - Liegt das Problem im Testobjekt, wird über die Beobachtung eine Fehlermeldung erstellt
 - Für jede im Testprotokoll enthaltene und vom Sollwert bzw. Sollverhalten abweichende Beobachtung
 - Duplikate von Fehlermeldungen vermeiden
- Ursachenanalyse ist Aufgabe der Entwickler
 - Die Tester müssen nicht die Ursache des gemeldeten Problems herausfinden

Fehlermeldung – Ziele

Fehlerberichte haben folgenden Ziele:

- **Entwicklern** und anderen Beteiligten **Informationen** über jedes aufgetretene unerwünschte Ereignis zur Verfügung zu stellen, um
 - spezifische Auswirkungen zu identifizieren,
 - das Problem mit minimalem Test zu reproduzieren und zu isolieren
 - und die möglichen Fehlerzustände wie erforderlich zu korrigieren oder anderweitig das Problem zu lösen
- **Testmanagern** eine Möglichkeit zu geben, die **Qualität der Arbeitsergebnisse** und die Auswirkungen auf das Testen nachzuverfolgen
- Ideen liefern für die **Verbesserung** der Entwicklung und des Testprozesses

Fehlermeldung – Fehlerdatenbank

- Üblicherweise ist eine zentrale Fehlerdatenbank vorhanden, in der alle Probleme, Mängel oder Fehlerwirkungen, die im Test oder auch später im Betrieb entdeckt werden, erfasst und verwaltet werden.
- Fehlermeldungen können von allen an der Entwicklung beteiligten Personen, aber auch von Kunden- und Anwenderseite eingebracht werden.
- Sie können sich auf Probleme in den getesteten Programm- (teil)en beziehen, aber auch auf Fehler oder Mängel in Spezifikationen, Benutzerhandbüchern oder anderen Dokumenten.
- Dokumentierte (und noch nicht in Testfällen abgeprüfte) Fehler sollten in neuen Testfällen und Testskripten resultieren, damit diese gleichen Fehler in Zukunft schnell gefunden werden.

Fehlerbericht - einheitliches Schema (1 von 2)

- jede Fehlermeldung muss nach einheitlichem Schema aufgebaut sein
 - Reibungslose Kommunikation
 - Meldungen statistisch auswertbar
- Dieses Schema legt der Testmanager (z.B. im Testkonzept) fest
- Typischerweise beinhaltet ein Fehlerbericht:
 - Kennung
 - Titel und kurze Zusammenfassung des gemeldeten Fehlerzustands
 - Datum des Fehlerberichts
 - Identifikation des Testelements und der Testumgebung
 - Phase des Entwicklungslebenszyklus, in der der Fehlerzustand beobachtet wurde
 - Beschreibung des Fehlerzustands inkl. Datenbank-Dumps, Protokolle, Bildschirmfotos, etc.
 - Erwartete Ergebnisse und Istergebnisse
 - ...

Fehlerbericht - einheitliches Schema (2 von 2)

- ...
- Umfang oder Auswirkungsgrad (Fehlerschweregrad) des Fehlerzustands
- Dringlichkeit/Priorität der Behebung
- Status des Fehlerberichts, z.B. offen, zurückgestellt, abgewiesen, Duplikat, Korrektur, Nachtest oder erledigt
- Schlussfolgerungen, Empfehlungen und Freigaben
- Allgemeine Probleme, z.B. wenn andere Bereiche durch die Korrektur der Abweichung beeinflusst werden könnten
- Änderungshistorie, welche Tätigkeiten von welchen Projektteammitgliedern zur Korrektur der Abweichung durchgeführt wurde
- Referenzen, einschließlich des Testfalls, der das Problem aufgedeckt hat

Fehlerbericht

- Wichtig ist, dass alle Informationen erfasst werden, die für das Reproduzieren und Lokalisieren des potenziellen Fehlers notwendig sind, sowie Informationen, die eine Auswertung der Produktqualität und des Korrekturfortschritts ermöglichen
- Unabhängig vom vereinbarten Schema gilt
 - Eine Meldung muss objektiv verfasst werden
 - Eine Meldung muss so abgefasst sein, dass der zuständige Entwickler mit minimalem Aufwand das Problem nachvollziehen und dessen Ursache finden kann
 - Die Reproduktion, Lokalisierung und Behebung eines Problems bedeuten für den Entwickler in der Regel nicht eingeplante, zusätzliche Arbeit; in dieser Situation neigen Entwickler dazu, jede unklare Fehlermeldung ohne weitere Analyse abzulehnen oder zurückzustellen



Beispiel: Fehlerklasse – Auswirkung des Fehlers

Klasse	Bedeutung
1	Systemabsturz mit ggf. Datenverlust; das Testobjekt ist in dieser Form nicht einsetzbar
2	Wesentliche Funktion ist fehlerhaft; Anforderung nicht beachtet oder falsch umgesetzt; das Testobjekt ist nur mit großen Einschränkungen einsetzbar
3	Funktionale Abweichung bzw. Einschränkung (»normale« Fehler); Anforderung fehlerhaft oder nur teilweise umgesetzt; System kann mit Einschränkungen genutzt werden
4	Geringfügige Abweichung; System kann ohne Einschränkung genutzt werden (Beseitigung hat mit dem nächsten Release zu erfolgen)
5	Schönheitsfehler (z.B. Mangel im Maskenlayout); System kann ohne Einschränkung genutzt werden (Beseitigung kann mit einem der nächsten Releases erfolgen)



Beispiel: Fehlerpriorität – Dringlichkeit für die Behebung

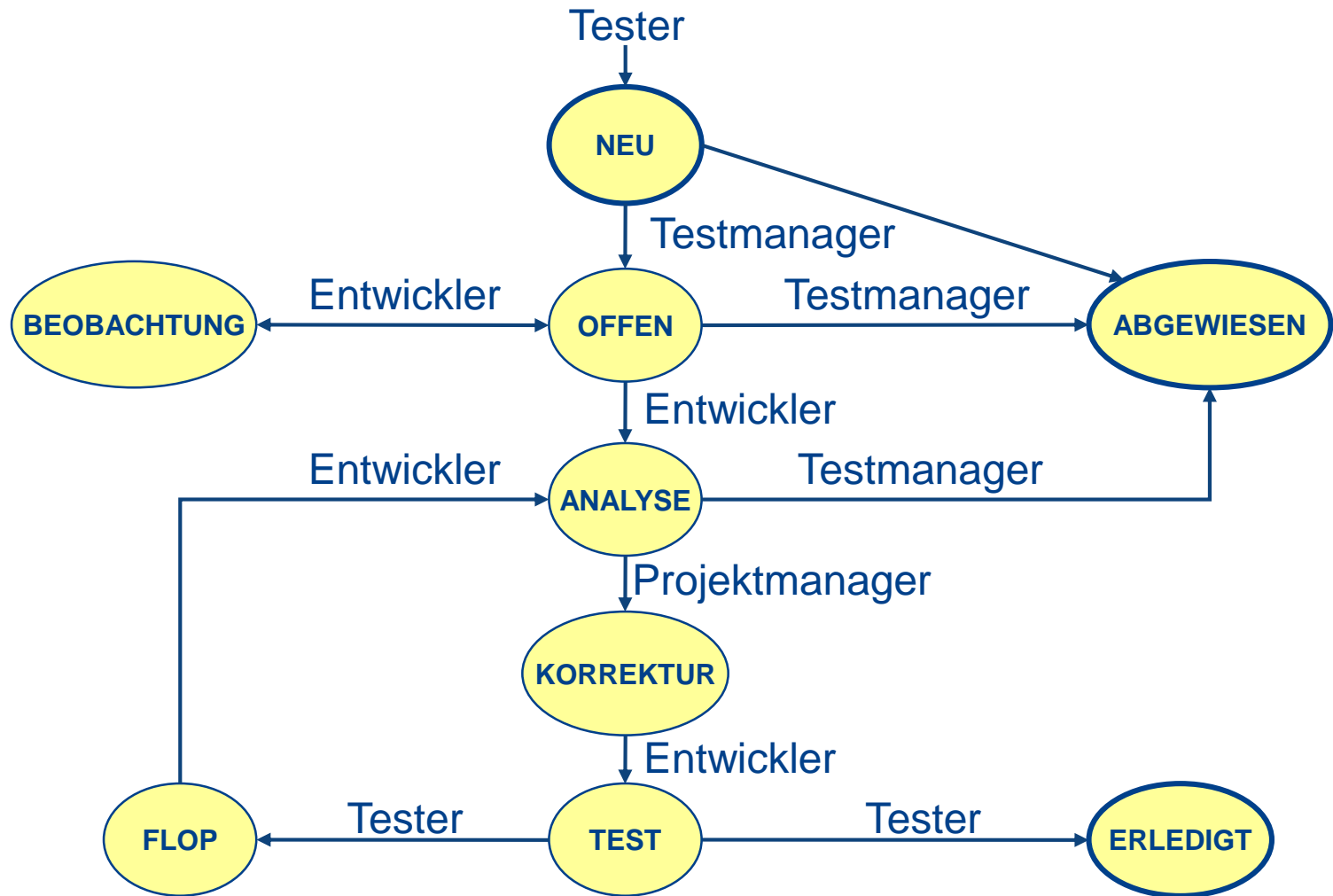
Priorität	Bedeutung
1 – PATCH	Der Arbeitsablauf beim Anwender ist blockiert oder die laufenden Tests können nicht fortgesetzt werden. Das Problem muss unmittelbar, ggf. Provisorisch behoben werden; ein Patch ist zu erstellen
2 – NÄCHSTE VERSION	Die Fehlerkorrektur erfolgt mit der nächsten regulären Produktversion oder der nächsten Testobjektlieferung
3 – GELEGENTLICH	Die Fehlerkorrektur erfolgt, sobald die betroffenen Systemteile ohnehin überarbeitet werden
4 – OFFEN	Korrekturplanung ist noch zu treffen

Fehlerstatus

- Testmanagement stellt sicher:
 - Fehler werden ordentlich erfasst und verwaltet
 - Fehlerzustände werden in Kooperation mit dem Projektmanagement zügig korrigiert
 - Korrekturen sind mit den neuen Versionen des Testobjekts verfügbar
- Hierzu ist eine kontinuierliche Verfolgung des Fehleranalyse- und Korrekturprozesses über alle Stadien notwendig
- Diese Verfolgung geschieht anhand des Fehlerstatus
- Dabei durchläuft jede Fehlermeldung eine Reihe festgelegter Status, welche die erstmalige Erfassung bis zur erfolgreichen Fehlerkorrektur beinhalten



Beispiel: Fehlerstatusmodell



Fehlerkorrektur - Erweiterungen

- Geforderte Änderungen in vielen Fällen keine »Fehlerkorrekturen«, sondern funktionale Erweiterungen (neue Features)
- Grenze zwischen »Fehlermeldung« und »Erweiterungswunsch« und die Beurteilung als »berechtigt« oder »unberechtigt« oft Ermessenssache
- Instanz im Fehlermanagement benötigt, die Änderungsanforderungen genehmigt oder ablehnt
- Änderungskontrollausschuss (**C**hange **C**ontrol **B**oard) und wird üblicherweise von Vertretern folgender Interessengruppen gebildet
 - Produktmanagement
 - Projektmanagement
 - Testmanagement
 - Kundenvertreter

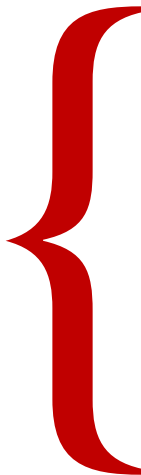


Folgende Fragen sollten Sie jetzt beantworten können

- Welche Daten soll eine Fehlermeldung enthalten?
- Was ist der Unterschied zwischen Fehlerpriorität und Fehlerklasse?
- Wozu wird ein Fehlerstatusmodell (im Testmanagement) benötigt?
- Wie sieht ein typisches Fehlerstatusmodell aus?

Kapitel 5

Test- management



Testorganisation

Testplanung und -schätzung

Testüberwachung und -steuerung

Risiken und Testen

Fehlermanagement

Konfigurationsmanagement

Konfigurationsmanagement



Zustand des Chaos, eine Orientierung ist kaum möglich.

besser



Zustand der Ordnung bzw. Strukturiertheit, eine Orientierung ist leicht möglich!

Konfigurationsmanagement (KM)

- Ziel:
Schaffung und Erhalt der Integrität (Aktualität, Vollständigkeit) der Software oder des Systems (Komponenten, Daten und Dokumentation, ...) sowie ihrer Beziehungen untereinander während des Projekt- und Produktlebenszyklus
- Ein Softwaresystem besteht aus vielen **einzelnen Bausteinen**, die zueinander passen müssen, damit das System als Ganzes funktioniert
- Von jedem dieser Bausteine entstehen im Laufe der Entwicklung des Systems neue, korrigierte oder verbesserte **Versionen** oder Varianten
- Da an diesem Prozess **mehrere Entwickler und Tester** parallel beteiligt sind, ist es alles andere als einfach, den Überblick zu behalten, welche Bausteine aktuell sind und zusammengehören, dies ist Aufgabe des Konfigurationsmanagement



Was sind typische Symptome eines unzureichenden KMs?



Typische Symptome von unzureichendem KM (1 von 2)

- Entwickler überschreiben gegenseitig ihre am Code oder anderen Dokumenten vorgenommenen Modifikationen
- gleichzeitiger Zugriff auf gemeinsame Codedateien nicht verhindert wird
- Integrationsarbeiten sind behindert,
 - weil unklar ist, welche Versionen des Codes einer bestimmten Komponente im Entwicklungsteam existieren und welche davon aktuell sind
 - weil unklar ist, welche Versionen verschiedener Komponenten zusammengehören und zu einem größeren Teilsystem integriert werden können
 - weil Compiler und andere Entwicklungswerkzeuge in unterschiedlichen Versionen eingesetzt werden

Typische Symptome von unzureichendem KM (2 von 2)

- **Fehleranalyse, Fehlerkorrektur und Regressionstest sind erschwert,**
 - weil unbekannt ist, wo und warum der Code einer Komponente gegenüber der Vorversion geändert wurde
 - weil unbekannt ist, aus welchen Codedateien ein bestimmtes integriertes Teilsystem (Objektcode) hervorgegangen ist
- **Tests und Testauswertung sind behindert,**
 - weil unklar ist, welche Testfälle zu welchem Versionsstand eines Testobjekts gehören
 - weil unklar ist, welche Testergebnisse ein Testlauf an einem Testobjekt einer bestimmten Version erbracht hat

Anforderungen an Konfigurationsmanagement

- **Versionsverwaltung:**

Katalogisieren, Speichern und Wiederabrufen von unterschiedlichen Versionen eines Konfigurationsobjekts (z.B. Version 1.0 und 1.1 einer Datei)

- **Konfigurationsverwaltung:**

Bestimmung und Verwaltung aller Dateien (Konfigurationsobjekte) in der jeweils passenden Version, die zusammen ein Teilsystem bilden (Konfiguration)

- **Statusverfolgung von Fehlern und Änderungen:**

Aufzeichnung von Problembereichen und Änderungsanforderungen und die Möglichkeit, deren Umsetzung an den Konfigurations-objekten nachzuvollziehen

- **Konfigurationsaudits:**

Wie wirksam ist das KM?



Folgende Fragen sollten Sie jetzt beantworten können

- Welche Anforderungen aus Sicht des Tests stellen sich an das Konfigurationsmanagement?
- Wie kann das Konfigurationsmanagement das Test (im Projekt) unterstützen?



Zusammenfassung (1 von 2)



- Entwicklungs- und Testaktivitäten sollen organisatorisch getrennt sein. Je klarer diese Trennung erfolgt, umso wirksamer kann getestet werden!
- Je nach Aufgabenstellung innerhalb des Testprozesses werden Mitarbeiter mit rollenspezifischen Testkenntnissen benötigt. Neben fachlichen Fähigkeiten ist auch soziale Kompetenz gefragt!
- Zu den Aufgaben des Testmanagers gehören Planung, Überwachung und Steuerung der einzelnen Testzyklen
- In dem Testkonzept beschreibt und begründet der Testmanager, wie er die anstehenden Testaufgaben lösen will, welchen Testumfang er plant, welche Werkzeuge eingesetzt werden sollen usw.



Zusammenfassung (2 von 2)



- Fehler und Mängel, die im Test übersehen werden, können hohe Fehlerkosten nach sich ziehen. Bei der Wahl der Teststrategie wird ein optimales Verhältnis zwischen Testkosten, verfügbaren Ressourcen und drohenden Fehlerkosten angestrebt
- Risikomanagement und risikoorientierter Test helfen, die kritischen Fehler frühzeitig zu finden und den Testaufwand zu optimieren
- Fehlermanagement und Konfigurationsmanagement bilden die Basis für einen effizienten Testprozess
- Fehlermeldungen müssen nach einem projektweit einheitlichen Schema erfasst und über alle Stadien des Fehleranalyse- und Korrekturprozesses verfolgt werden
- Normen und Standards enthalten Vorgaben und Empfehlungen zur fachgerechten Durchführung von Softwaretests. Eine Orientierung an solchen Standards macht auch dort Sinn, wo deren Einhaltung nicht verpflichtend vorgeschrieben ist



Folgende Fragen sollten Sie jetzt beantworten können

(1 von 2)

- Welche grundsätzlichen Modelle der Organisation des Testens lassen sich bzgl. Aufgabenteilung zwischen Entwicklung und Test unterscheiden?
- Welches sind die typischen Aufgaben der Rolle Testmanager?
- Welches sind die typischen Aufgaben der Rolle Tester?
- Welche Verfahren zur Schätzung des Testaufwands gibt es?
- Welche Arten von Metriken zur Überwachung des Testfortschritts lassen sich unterscheiden?
- Welche Informationen soll ein Teststatusbericht enthalten?
- Welches sind die Aktivitäten des Risikomanagements?
- Wie berechnet man ein Risiko?
- Was sind typische Produkt- und Projektrisiken?



Folgende Fragen sollten Sie jetzt beantworten können (2 von 2)

- Welche Daten soll eine Fehlermeldung enthalten?
- Was ist der Unterschied zwischen Fehlerpriorität und Fehlerklasse?
- Wozu wird ein Fehlerstatusmodell benötigt?
- Welche Anforderungen aus Sicht des Tests stellen sich an das Konfigurationsmanagement?
- Welche grundsätzlichen Arten von Normen und Standards lassen sich unterscheiden?



Muster-Prüfungsfragen

Testen Sie Ihr Wissen...



Frage 1

30. Welche der folgenden Aussagen beschreibt AM BESTEN die Aufgabenteilung zwischen Testmanager und Tester? [K1]

a)	Der Testmanager plant Testaktivitäten und wählt die Normen aus, die befolgt werden müssen, während der Tester die Werkzeuge und ihre Nutzungsregeln auswählt, die verwendet werden.	<input type="checkbox"/>
b)	Der Testmanager plant und steuert die Testaktivitäten, während der Tester die Tests entwirft und über das Testautomatisierungs-Framework entscheidet.	<input type="checkbox"/>
c)	Der Testmanager plant, überwacht und steuert die Testaktivitäten, während der Tester die Tests entwirft und über die Freigabe des Testobjekts entscheidet.	<input type="checkbox"/>
d)	Der Testmanager plant und organisiert die Testdurchführung und spezifiziert die Testfälle, während die Tester die Tests priorisieren und durchführen.	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 1 - Lösung

30. Welche der folgenden Aussagen beschreibt AM BESTEN die Aufgabenteilung zwischen Testmanager und Tester? [K1]

a)	Der Testmanager plant Testaktivitäten und wählt die Normen aus, die befolgt werden müssen, während der Tester die Werkzeuge und ihre Nutzungsregeln auswählt, die verwendet werden.	<input type="checkbox"/>
b)	Der Testmanager plant und steuert die Testaktivitäten, während der Tester die Tests entwirft und über das Testautomatisierungs-Framework entscheidet.	<input checked="" type="checkbox"/>
c)	Der Testmanager plant, überwacht und steuert die Testaktivitäten, während der Tester die Tests entwirft und über die Freigabe des Testobjekts entscheidet.	<input type="checkbox"/>
d)	Der Testmanager plant und organisiert die Testdurchführung und spezifiziert die Testfälle, während die Tester die Tests priorisieren und durchführen.	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 2

31. Welche der folgenden Metriken wäre am NÜTZLICHSTEN für die Überwachung während der Testdurchführung? [K1]

a)	Prozentualer Anteil der durchgeführten Testfälle	<input type="checkbox"/>
b)	Anzahl der durchschnittlich an der Testdurchführung beteiligten Tester	<input type="checkbox"/>
c)	Überdeckung der Anforderungen durch Code	<input type="checkbox"/>
d)	Prozentualer Anteil der bereits erstellten und gereviewten Testfälle	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 2 - Lösung

31. Welche der folgenden Metriken wäre am NÜTZLICHSTEN für die Überwachung während der Testdurchführung? [K1]

a)	Prozentualer Anteil der durchgeführten Testfälle	<input checked="" type="checkbox"/>
b)	Anzahl der durchschnittlich an der Testdurchführung beteiligten Tester	<input type="checkbox"/>
c)	Überdeckung der Anforderungen durch Code	<input type="checkbox"/>
d)	Prozentualer Anteil der bereits erstellten und gereviewten Testfälle	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 3

32. Welche ZWEI der folgenden Punkte können sich auf die (initiale) Testplanung auswirken und Teil davon sein? [K2]

a)	Budgeteinschränkungen	<input type="checkbox"/>
b)	Testziele	<input type="checkbox"/>
c)	Testprotokoll	<input type="checkbox"/>
d)	Ausfallrate	<input type="checkbox"/>
e)	Anwendungsfälle	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 3 - Lösung

32. Welche ZWEI der folgenden Punkte können sich auf die (initiale) Testplanung auswirken und Teil davon sein? [K2]

a)	Budgeteinschränkungen	<input checked="" type="checkbox"/>
b)	Testziele	<input checked="" type="checkbox"/>
c)	Testprotokoll	<input type="checkbox"/>
d)	Ausfallrate	<input type="checkbox"/>
e)	Anwendungsfälle	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 4

33. Welche der folgenden Listen enthält nur typische Endekriterien? [K2]

a)	Kennzahlen zu Zuverlässigkeit und Testüberdeckung, Testkosten, Zeitplan, Status über Fehlerbehebung und Restrisiken	<input type="checkbox"/>
b)	Kennzahlen zu Zuverlässigkeit und Testüberdeckung, Grad der Unabhängigkeit der Tester und Grad der Produktvollständigkeit	<input type="checkbox"/>
c)	Kennzahlen zu Zuverlässigkeit und Testüberdeckung, Testkosten, Verfügbarkeit einer Testumgebung, Zeit bis Markteinführung, Grad der Produktvollständigkeit	<input type="checkbox"/>
d)	Zeit bis Markteinführung, Restfehler, Qualifikation der Tester, Verfügbarkeit von testbaren Anforderungen, Testüberdeckung und Testkosten	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 4 - Lösung

33. Welche der folgenden Listen enthält nur typische Endekriterien? [K2]

a)	Kennzahlen zu Zuverlässigkeit und Testüberdeckung, Testkosten, Zeitplan, Status über Fehlerbehebung und Restrisiken	<input checked="" type="checkbox"/>
b)	Kennzahlen zu Zuverlässigkeit und Testüberdeckung, Grad der Unabhängigkeit der Tester und Grad der Produktvollständigkeit	<input type="checkbox"/>
c)	Kennzahlen zu Zuverlässigkeit und Testüberdeckung, Testkosten, Verfügbarkeit einer Testumgebung, Zeit bis Markteinführung, Grad der Produktvollständigkeit	<input type="checkbox"/>
d)	Zeit bis Markteinführung, Restfehler, Qualifikation der Tester, Verfügbarkeit von testbaren Anforderungen, Testüberdeckung und Testkosten	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 5

34. Welcher der folgenden Punkte ist NICHT in einem Testabschlussbericht enthalten? [K2]

a)	Definition der Kriterien für bestanden/fehlgeschlagen und die Ziele des Tests	<input type="checkbox"/>
b)	Abweichungen von der Testvorgehensweise	<input type="checkbox"/>
c)	Messung des tatsächlichen Fortschritts im Vergleich zu den Endekriterien	<input type="checkbox"/>
d)	Bewertung der Qualität des Testelements	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 5 - Lösung

34. Welcher der folgenden Punkte ist NICHT in einem Testabschlussbericht enthalten? [K2]

a)	Definition der Kriterien für bestanden/fehlgeschlagen und die Ziele des Tests	<input checked="" type="checkbox"/>
b)	Abweichungen von der Testvorgehensweise	<input type="checkbox"/>
c)	Messung des tatsächlichen Fortschritts im Vergleich zu den Endekriterien	<input type="checkbox"/>
d)	Bewertung der Qualität des Testelements	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 6

35. Das Projekt entwickelt einen „smarten“ Heizungsthermostat. Die Regelungsalgorithmen des Thermostats wurden als Matlab/Simulink-Modelle modelliert und laufen auf dem über das Internet angebundenen Server. Der Thermostat übernimmt die Vorgaben des Servers zur Ansteuerung der Heizungsventile.

Die Testmanagerin hat im Testkonzept folgende Festlegungen zur Teststrategie/Vorgehensweise festgelegt.

1. Der Abnahmetest für das Gesamtsystem wird als Erfahrungsbasierter Test durchgeführt.
2. Die Regelalgorithmen auf dem Server werden während der Implementierung mittels Continuous Integration getestet.
3. Der funktionale Test des Thermostats wird als risikoorientierter Test durchgeführt.
4. Die Absicherungstests von Daten / Kommunikation über das Internet erfolgen gemeinsam mit externen Security-Experten.

Welche vier gängigen Arten von Teststrategien/Vorgehensweisen hat die Testmanagerin dabei im Testkonzept umgesetzt? [K2]

a)	methodisch, analytisch, reaktiv und leistungserhaltend	<input type="checkbox"/>
b)	analytisch, modellbasiert, beratend und reaktiv	<input type="checkbox"/>
c)	modellbasiert, methodisch, analytisch und beratend	<input type="checkbox"/>
d)	leistungserhaltend, beratend, reaktiv und methodisch	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 6 - Lösung

35. Das Projekt entwickelt einen „smarten“ Heizungsthermostat. Die Regelungsalgorithmen des Thermostats wurden als Matlab/Simulink-Modelle modelliert und laufen auf dem über das Internet angebundenen Server. Der Thermostat übernimmt die Vorgaben des Servers zur Ansteuerung der Heizungsventile.

Die Testmanagerin hat im Testkonzept folgende Festlegungen zur Teststrategie/Vorgehensweise festgelegt.

1. Der Abnahmetest für das Gesamtsystem wird als Erfahrungsbasierter Test durchgeführt.
2. Die Regelalgorithmen auf dem Server werden während der Implementierung mittels Continuous Integration getestet.
3. Der funktionale Test des Thermostats wird als risikoorientierter Test durchgeführt.
4. Die Absicherungstests von Daten / Kommunikation über das Internet erfolgen gemeinsam mit externen Security-Experten.

Welche vier gängigen Arten von Teststrategien/Vorgehensweisen hat die Testmanagerin dabei im Testkonzept umgesetzt? [K2]

a)	methodisch, analytisch, reaktiv und leistungserhaltend	<input type="checkbox"/>
b)	analytisch, modellbasiert, beratend und reaktiv	<input checked="" type="checkbox"/>
c)	modellbasiert, methodisch, analytisch und beratend	<input type="checkbox"/>
d)	leistungserhaltend, beratend, reaktiv und methodisch	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 7

36. Welcher der folgenden Punkte kennzeichnet einen auf Metriken basierenden Ansatz für die Testaufwandsschätzung? [K2]

a)	Budget, das von einem früheren, ähnlichen Testprojekt verwendet wurde	<input type="checkbox"/>
b)	Übergreifende Erfahrung aus gesammelten Interviews mit Testmanagern	<input type="checkbox"/>
c)	Im Testteam abgestimmte Aufwandsschätzung für die Testautomatisierung	<input type="checkbox"/>
d)	Von den Fachexperten gesammelte durchschnittliche Kalkulationen	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 7 - Lösung

36. Welcher der folgenden Punkte kennzeichnet einen auf Metriken basierenden Ansatz für die Testaufwandsschätzung? [K2]

a)	Budget, das von einem früheren, ähnlichen Testprojekt verwendet wurde	<input checked="" type="checkbox"/>
b)	Übergreifende Erfahrung aus gesammelten Interviews mit Testmanagern	<input type="checkbox"/>
c)	Im Testteam abgestimmte Aufwandsschätzung für die Testautomatisierung	<input type="checkbox"/>
d)	Von den Fachexperten gesammelte durchschnittliche Kalkulationen	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019

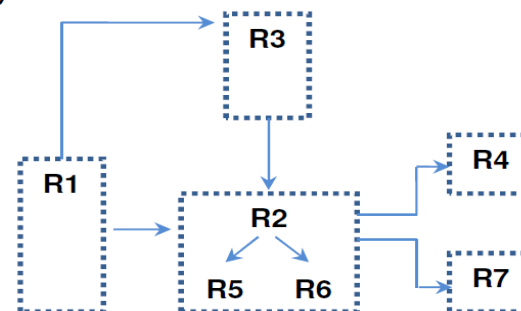


Frage 8

37. Als Testmanager verantworten Sie den Test folgender Anforderungen:

- R1 - Prozessanomalien
- R2 - Synchronisation
- R3 - Zulassung
- R4 - Problembearbeitung
- R5 - Finanzdaten
- R6 - Diagrammdaten
- R7 - Änderungen am Benutzerprofil

Notation: Logische Abhängigkeitsbeziehungen (A -> B bedeutet, dass B von A abhängig ist):



Wie würden Sie den Testausführungsplan, entsprechend den Abhängigkeiten der Anforderungen, aufbauen? [K3]

a)	R1 -> R3 -> R4 -> R7 -> R2 -> R5 -> R6	<input type="checkbox"/>
b)	R1 -> R3 -> R2 -> R4 -> R7 -> R5 -> R6	<input type="checkbox"/>
c)	R1 -> R3 -> R2 -> R5 -> R6 -> R4 -> R7	<input type="checkbox"/>
d)	R1 -> R2 -> R5 -> R6 -> R3 -> R4-> R7	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Frage 8 - Lösung

37. Als Testmanager verantworten Sie den Test folgender Anforderungen:

R1 - Prozessanomalien

R2 - Synchronisation

R3 - Zulassung

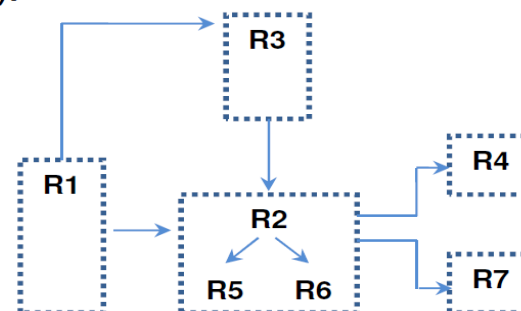
R4 - Problembearbeitung

R5 - Finanzdaten

R6 - Diagrammdaten

R7 - Änderungen am Benutzerprofil

Notation: Logische Abhängigkeitsbeziehungen (A -> B bedeutet, dass B von A abhängig ist):



Wie würden Sie den Testausführungsplan, entsprechend den Abhängigkeiten der Anforderungen, aufbauen? [K3]

a)	R1 -> R3 -> R4 -> R7 -> R2 -> R5 -> R6	<input type="checkbox"/>
b)	R1 -> R3 -> R2 -> R4 -> R7 -> R5 -> R6	<input type="checkbox"/>
c)	R1 -> R3 -> R2 -> R5 -> R6 -> R4 -> R7	<input checked="" type="checkbox"/>
d)	R1 -> R2 -> R5 -> R6 -> R3 -> R4-> R7	<input type="checkbox"/>

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board; 16.02.2019



Basiswissen Softwaretest Certified Tester

Exkurs 1: Fehlerkosten

HS@GTB
2019
Version 3.1



Nicht zu testen kann teuer sein

- Testen kann sehr aufwändig werden und kostenintensiv in einem Entwicklungsprojekt sein
- Fragen:
 - Wie viel Testaufwand ist für dieses Softwareprojekt angemessen?
 - Ab wann übersteigt der Aufwand den möglichen Nutzen?
- Um dies zu beantworten, muss betrachtet werden, welche Fehlerkosten ein Verzicht auf Prüfungen und Tests nach sich zieht
- Zwischen Fehlerkosten und Testkosten gilt es dann abzuwägen



Fehlerkosten (1 von 2)

- Werden Prüfungen und Tests im Umfang reduziert oder ganz eingespart, erhöht sich als Folge die Zahl der unentdeckten Fehlerzustände und Mängel
- Diese Fehlerzustände und Mängel verbleiben im Produkt und führen ggf. zu folgenden Kosten
 - Direkte Fehlerkosten
entstehen dem Kunden durch Fehlerwirkungen beim Betrieb des Softwareprodukts (Hersteller haftet evtl.),
z.B. Datenverlust, Fehlbuchung, Ausfall oder Schäden an Hardware oder Anlagenteilen, Personenschäden, Kosten durch Einspielen neuer Versionen
 - Indirekte Fehlerkosten
sind Kosten bzw. Umsatzverlust für den Hersteller, weil der Kunde mit dem Produkt unzufrieden ist,
z.B. Vertragsstrafen oder Minderung, erhöhter Aufwand für Kundenhotline und Support, Imageschaden, Verlust des Kunden



Fehlerkosten (2 von 2)

- Fehlerkorrekturkosten
entstehen dem Hersteller im Zuge der Fehlerkorrektur,
z.B. für Fehleranalyse und Korrektur, Regressionstests, erneute
Auslieferung und Installation, Nachschulung des Kunden, Verzug bei
Neuprodukten (wegen Bindung der Entwicklerkapazität im
Wartungsbereich), sinkende Konkurrenzfähigkeit

Fehler möglichst früh finden ...

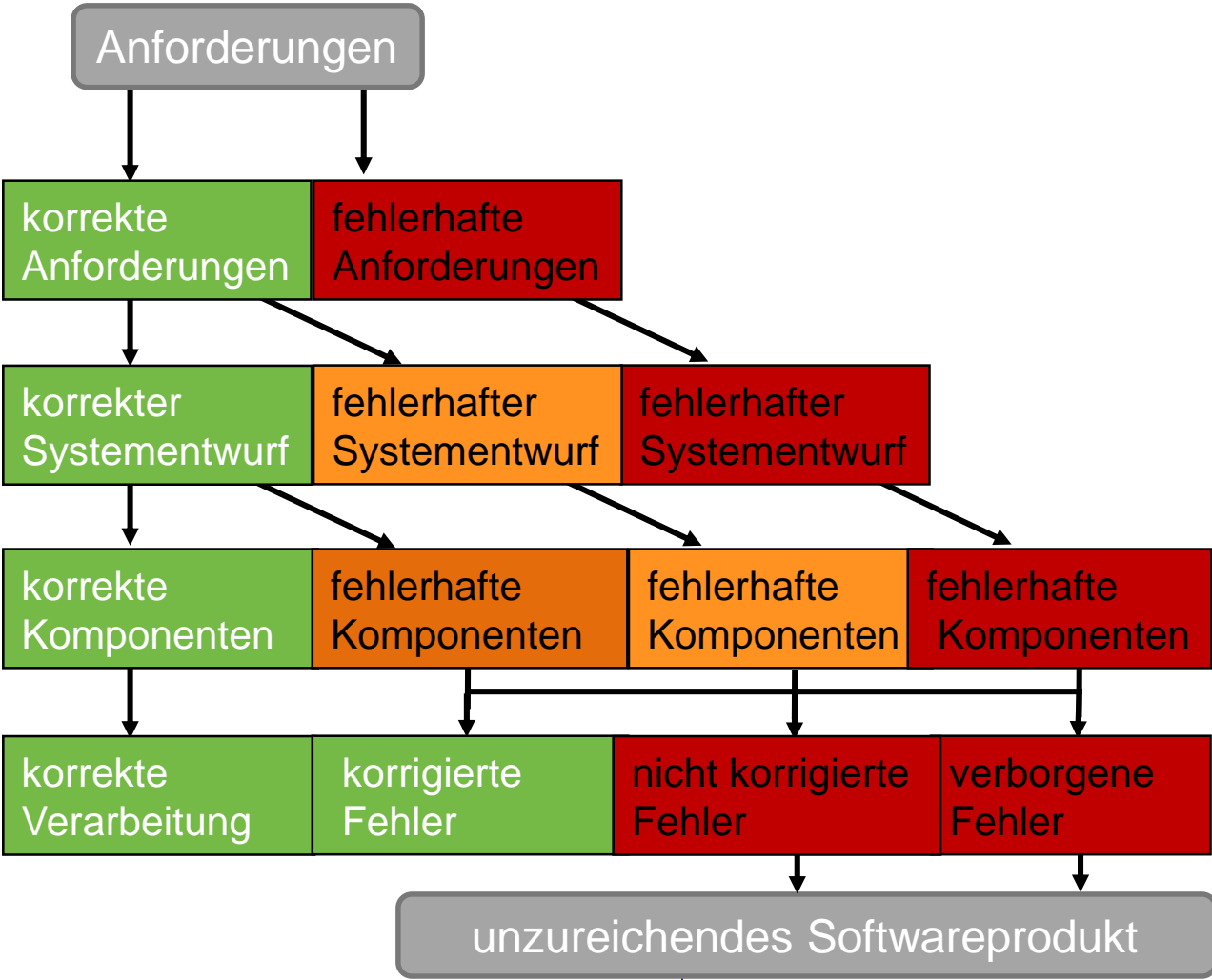
... denn Fehlerkosten steigen rapide über die Entwicklungsphasen an

- Ein Fehler, der sehr früh entsteht (z.B. ein Fehler in der Anforderungsdefinition), kann, solange er unentdeckt bleibt, in den anschließenden Entwicklungsphasen viele Folgefehler produzieren (Multiplikation des Effekts des ursprünglichen Fehlers)
 - Je später ein Fehler entdeckt wird, desto mehr Korrekturen sind notwendig. Unter Umständen müssen vorangegangene Phasen (Anforderungsdefinition, Design, Programmierung) zumindest teilweise wiederholt werden.
 - Zu wenig Tests resultieren in technischer Schuld, auf die im Lauf der Zeit Zinsen bezahlt werden müssen.
- Fehlerrisiko senken oder wenigstens begrenzen durch frühzeitige Prüfungen und Tests



Fehler und Folgefehler (1)

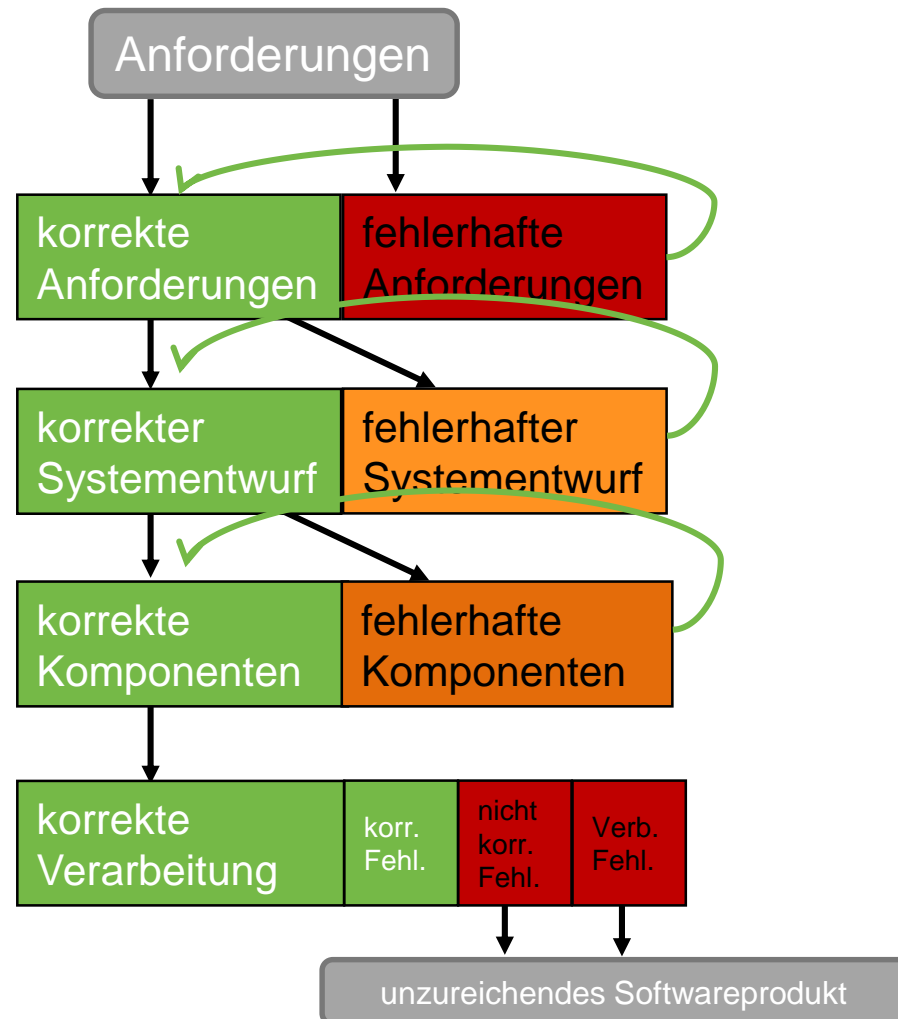
- Anforderungsdefinition, Fachkonzept
- Entwurf
- Realisierung
- Produktion





Fehler und Folgefehler (2) – Minimierung durch Korrektur

- Anforderungsdefinition, Fachkonzept
- Entwurf
- Realisierung
- Produktion





GTB

German Testing Board

Software. Testing. Excellence.



Basiswissen Softwaretest Certified Tester

Exkurs 2: Normen & Standards

HS@GTB
2019
Version 3.1



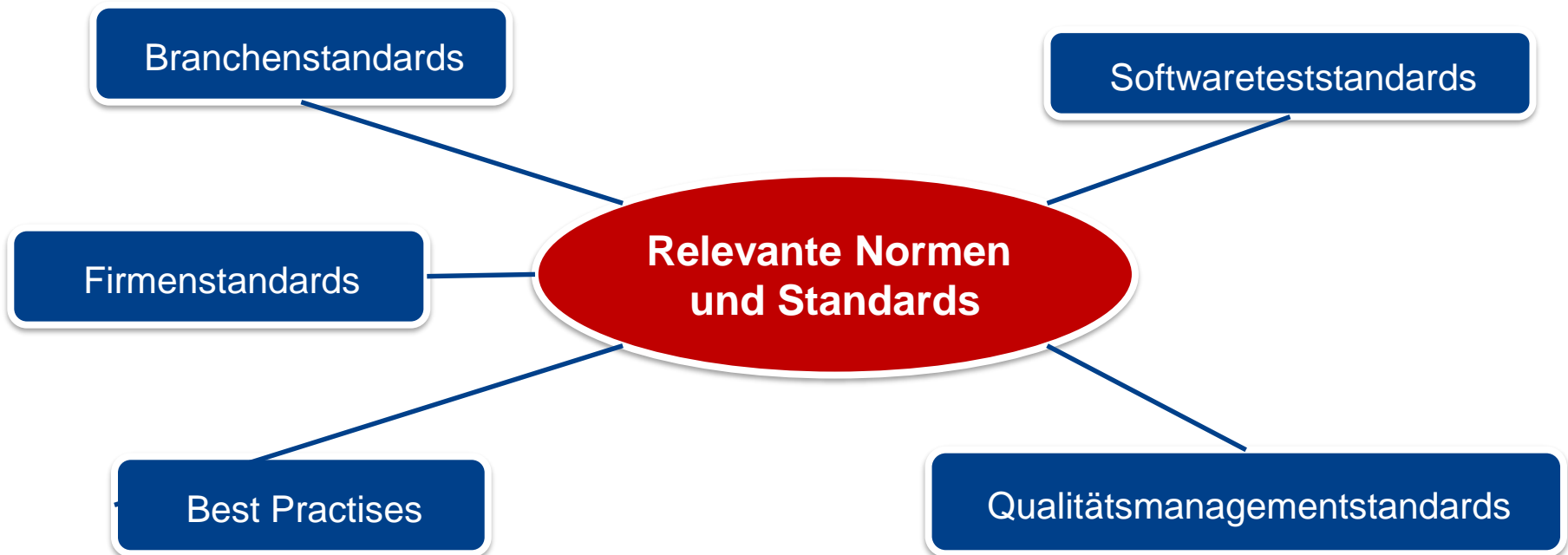
Exkurs: Relevante Normen und Standards (1 von 6)

- Zu den Aufgaben eines Qualitäts- oder Testmanagers gehört es festzustellen, welche Normen, Standards oder evtl. gesetzliche Richtlinien für das zu testende Produkt (Produktnormen) oder auch für das Projekt (Prozessnormen) relevant sind, und deren Einhaltung sicherzustellen
- Firmenstandards
 - Firmeninterne Richtlinien und Verfahrensanweisungen (des Herstellers und ggf. des Kunden), wie z.B. Qualitätsmanagement-handbuch, Rahmentestplan, Programmierrichtlinien
- Best Practises
 - Nicht standardisierte, aber fachlich bewährte Vorgehensweisen und Verfahren, die den Stand der Technik in einem Anwendungsgebiet darstellen



Exkurs: Relevante Normen und Standards (2 von 6)

Normen und Standards, die Randbedingungen vorgeben und den Stand der Technik definieren:





Exkurs: Relevante Normen und Standards (3 von 6)

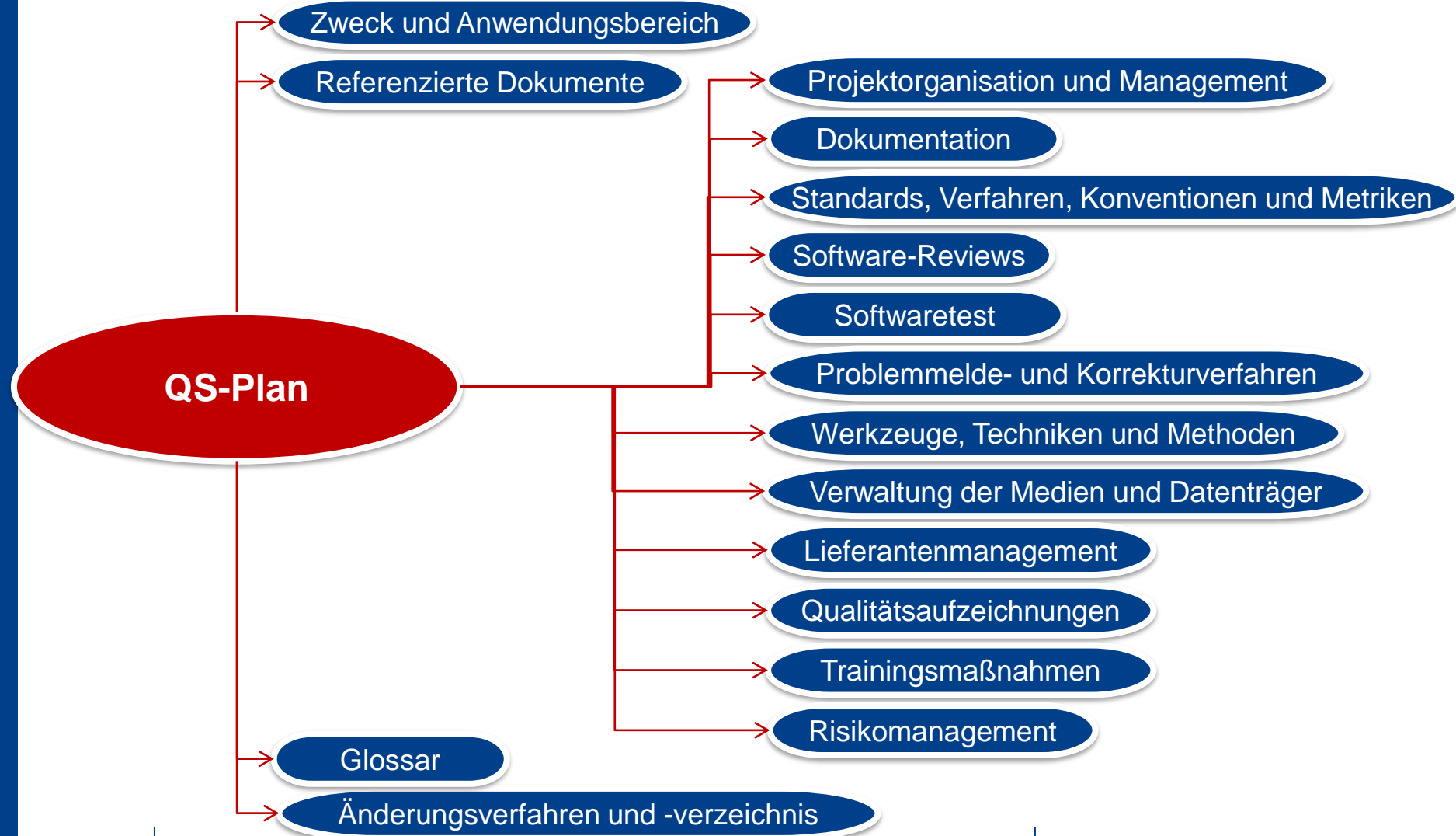
- Qualitätsmanagementstandards
 - Branchenübergreifende Standards, die Mindestanforderungen an Prozesse spezifizieren, ohne konkrete Anforderungen bezüglich der Umsetzung zu formulieren.
 - Bekanntes Beispiel ist [ISO 9000], die z.B. fordert, dass im Produktionsprozess (also auch im Spezialfall Softwareentwicklungsprozess) geeignete (Zwischen-)Prüfungen vorzusehen sind, ohne anzugeben, wann und wie diese zu erledigen sind

[ISO 9000] ISO 9000:2015
Qualitätsmanagementsysteme – Grundlagen und Begriffe.

- Ein weiteres Beispiel ist [IEEE 730] Quality Assurance Standard, siehe QS-Plan auf der nächsten Folie



Exkurs: IEEE 730 Qualitätssicherungsplan - Gliederung





Exkurs: Relevante Normen und Standards (4 von 6)

- Branchenstandards
 - Branchenspezifische Standards, beispielsweise
 - [DIN 60601-1-4] für Medizinprodukte,
 - [RTC-DO 178C] für Software in Flugzeugen,
 - [DIN EN 50128] für Bahn-Signalsysteme
 - die für eine bestimmte Produktkategorie oder ein Einsatzgebiet festlegen, in welchem Mindestumfang Tests durchzuführen oder nachzuweisen sind

[DIN 60601-1-4] DIN EN 60601-1-4, Ausgabe: 2001
Medizinische elektrische Geräte – Teil 1-4: Allgemeine Festlegungen für die Sicherheit; Ergänzungsnorm:
Programmierbare elektrische medizinische Systeme

[RTC-DO 178C]
RTC-DO Std 178C, Software Considerations in Airborne Systems and Equipment
Certification, RTCA, Inc., 2012

[DIN EN 50128]
DIN EN 50128:2012
Bahnanwendungen - Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Software
für Eisenbahnsteuerungs- und Überwachungssysteme



Exkurs: Relevante Normen und Standards (5 von 6)

- Softwareteststandards
 - Prozessstandards, die produktunabhängig festlegen, wie Softwaretests fachgerecht durchzuführen sind
 - Beispiele sind die Normen
 - [ISO 25010], [ISO 29119], [ISO 20246]

[ISO 25010] ISO/IEC 25010:2011

Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

[ISO 29119] ISO/IEC/IEEE 29119

Software and systems engineering -- Software testing

Part 1: Concepts and definitions – 2013

Part 2: Test processes – 2013

Part 3: Test documentation – 2013

Part 4: Test techniques – 2015

Part 5: Keyword-Driven Testing - 2016

[ISO 2024] ISO/IEC 20246:2017

Software and systems engineering -- Work product reviews



Exkurs: Relevante Normen und Standards (6 von 6)

- Technologiestandards für das Software Testen
 - Standards, die Technologien für die Automatisierung des Software Testens definieren
 - Beispiele sind die Normen
 - [ETSI TTCN-3] [OMG UTP], [IEEE 61671]

[ETSI TTCN-3]

ETSI Testing and Test Control Notation, v4.3.1, Juli 2011
ebenso als ITU-T Z.140 ff.

Part 1: TTCN-3 Core Language (CL)
Part 2: TTCN-3 Tabular Presentation Format (TFT)
Part 3: TTCN-3 Graphical Presentation Format (GFT)
Part 4: TTCN-3 Operational Semantics
Part 5: TTCN-3 Runtime Interface (TRI)
Part 6: TTCN-3 Control Interfaces (TCI)
Part 7: TTCN-3 Using ASN.1 with TTCN-3
Part 8: TTCN-3 The IDL to TTCN-3 Mapping
Part 9: TTCN-3 Using XML schema with TTCN-3
Part 10: TTCN-3 Documentation Comment Specification
Part 11: TTCN-3 Using JSON with TTCN-3

Extension Packages:

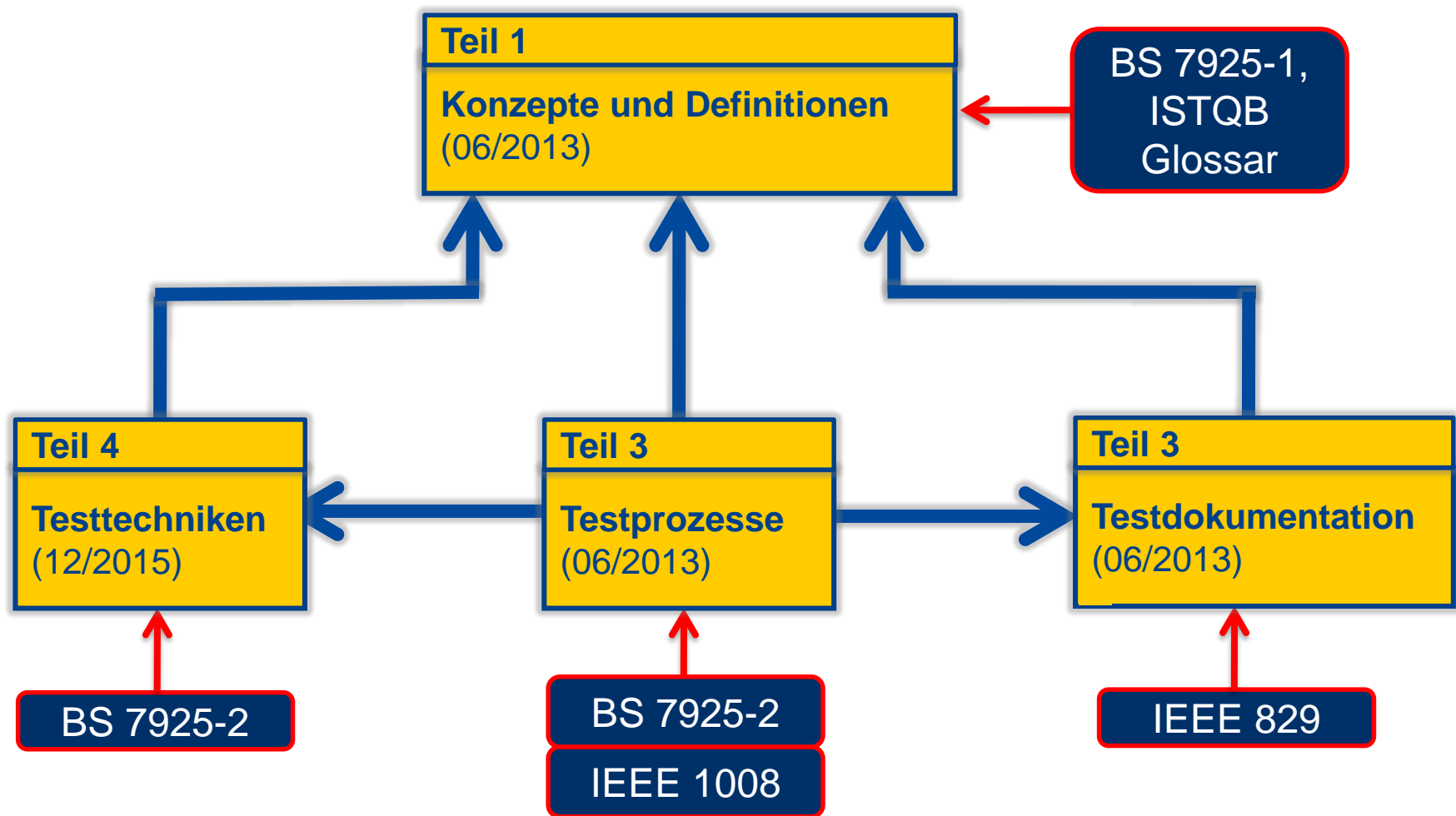
Configuration and Deployment Support, TTCN-3 Performance and Real-time Testing, Advanced Parameterization, Behaviour Types, Support of interfaces with continuous signals, Extended TRI, Extended Matching

[OMG UTP]

OMG UML Testing Profile, v.1.2, March 2013

[IEEE 61671]

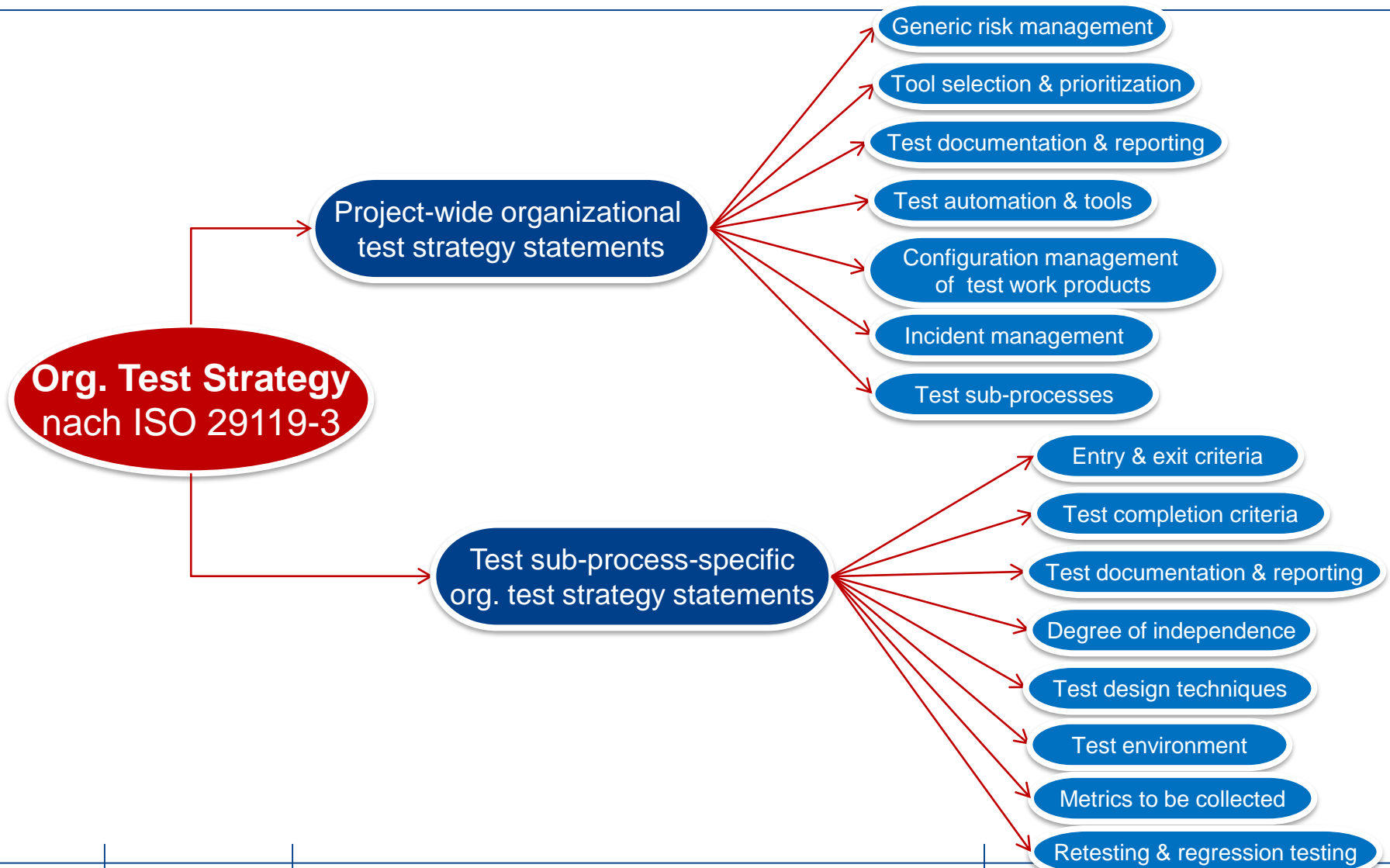
IEEE/IEC 67671 Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML - 2012
Part 2: Exchanging Instrument Description - 2012
Part 4: Exchanging Test Configuration - 2014
Part 5: Exchanging Test Adapter Description - 2015
Part 6: Exchanging Test Station Description - 2015





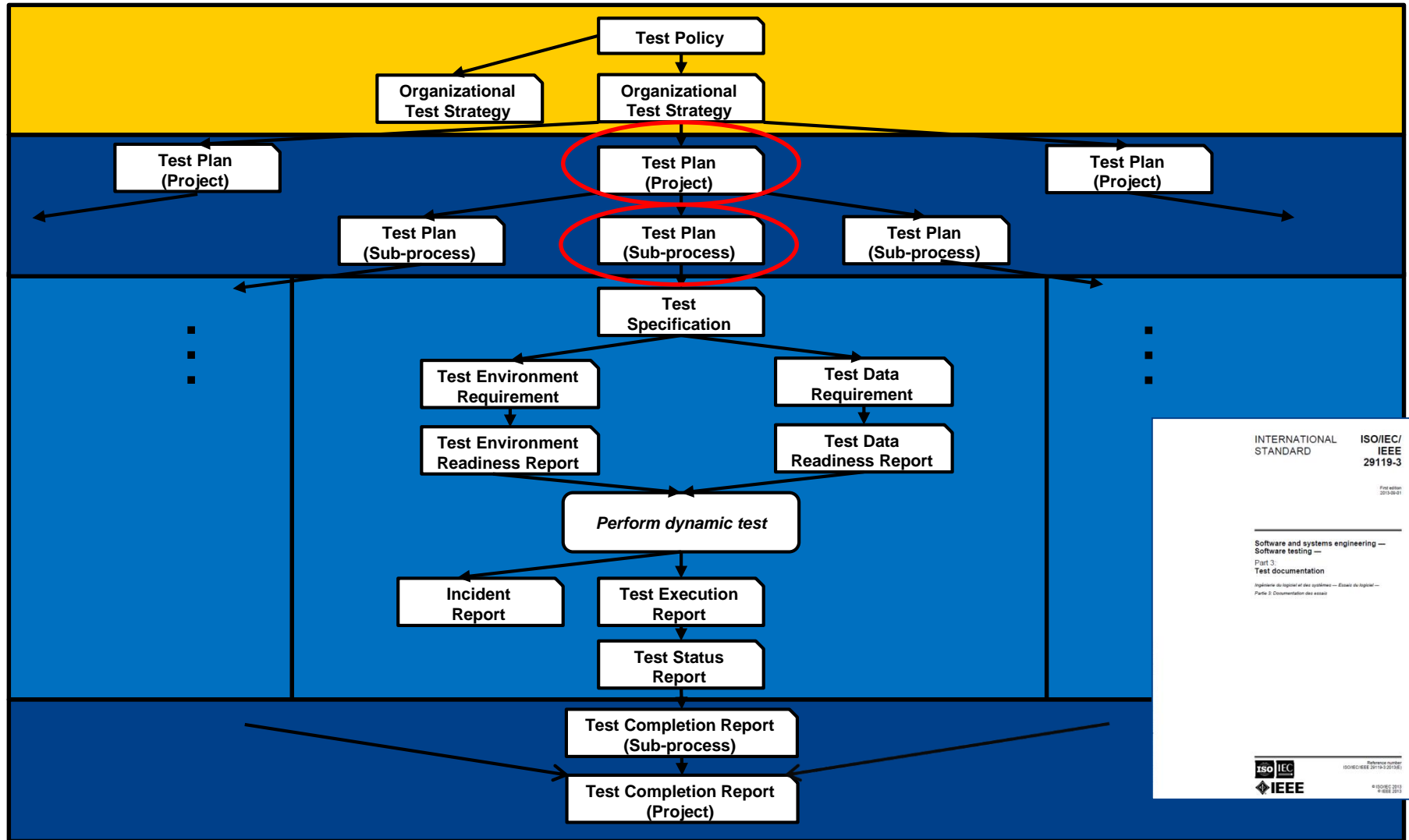


Exkurs: Teststrategie





ISO/IEC/IEEE 29119-3 Test Documentation



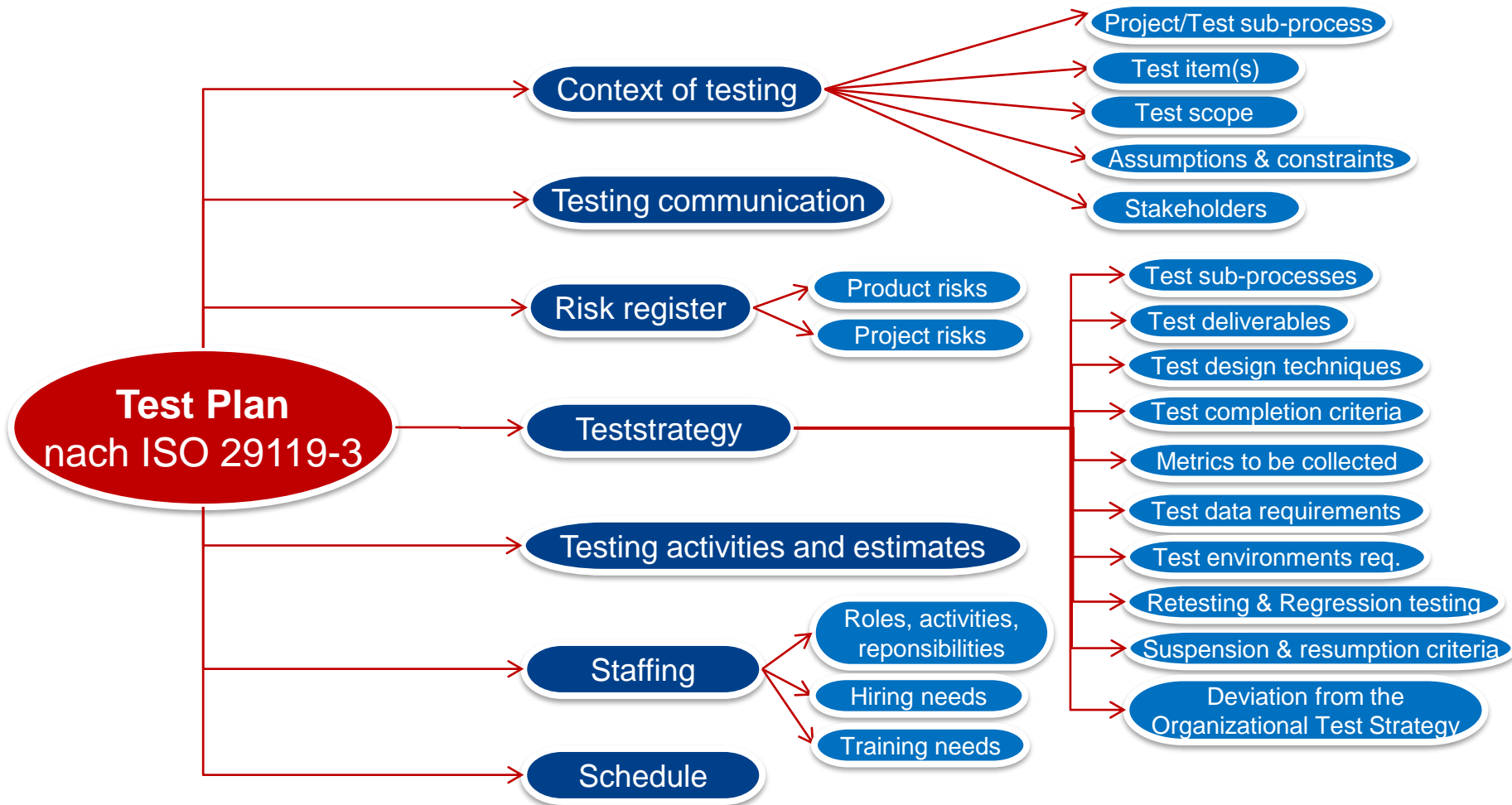


Exkurs: Testkonzept



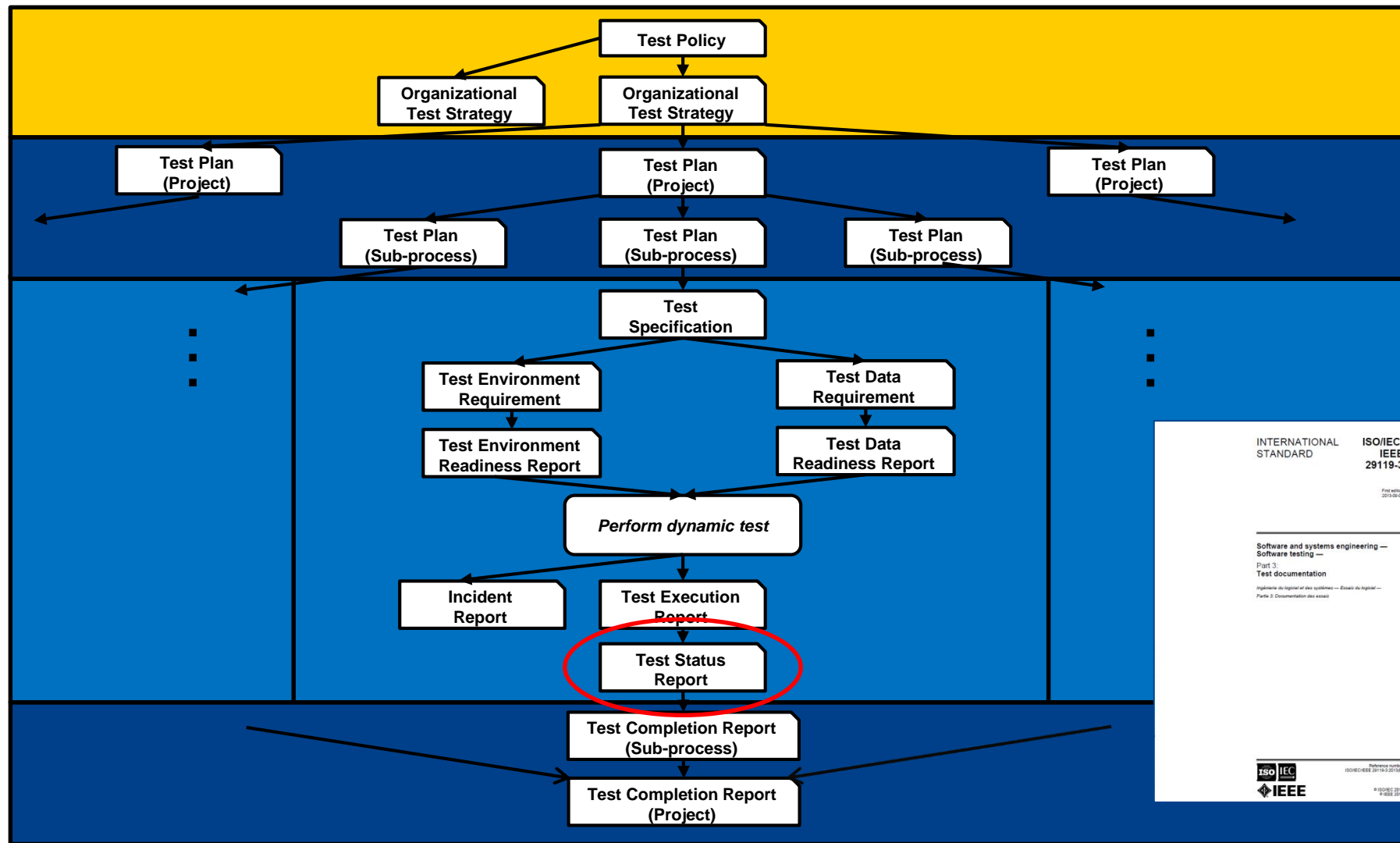


Exkurs: Testkonzept

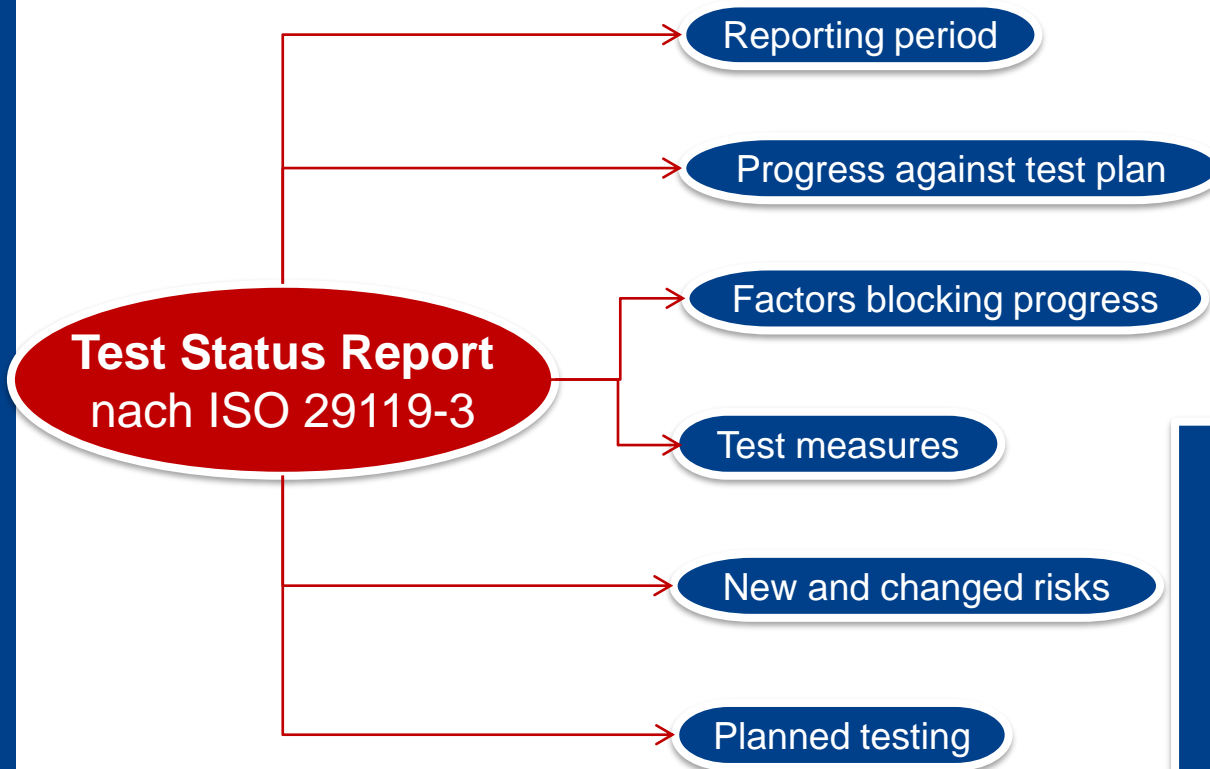




ISO/IEC/IEEE 29119-3 Test Documentation



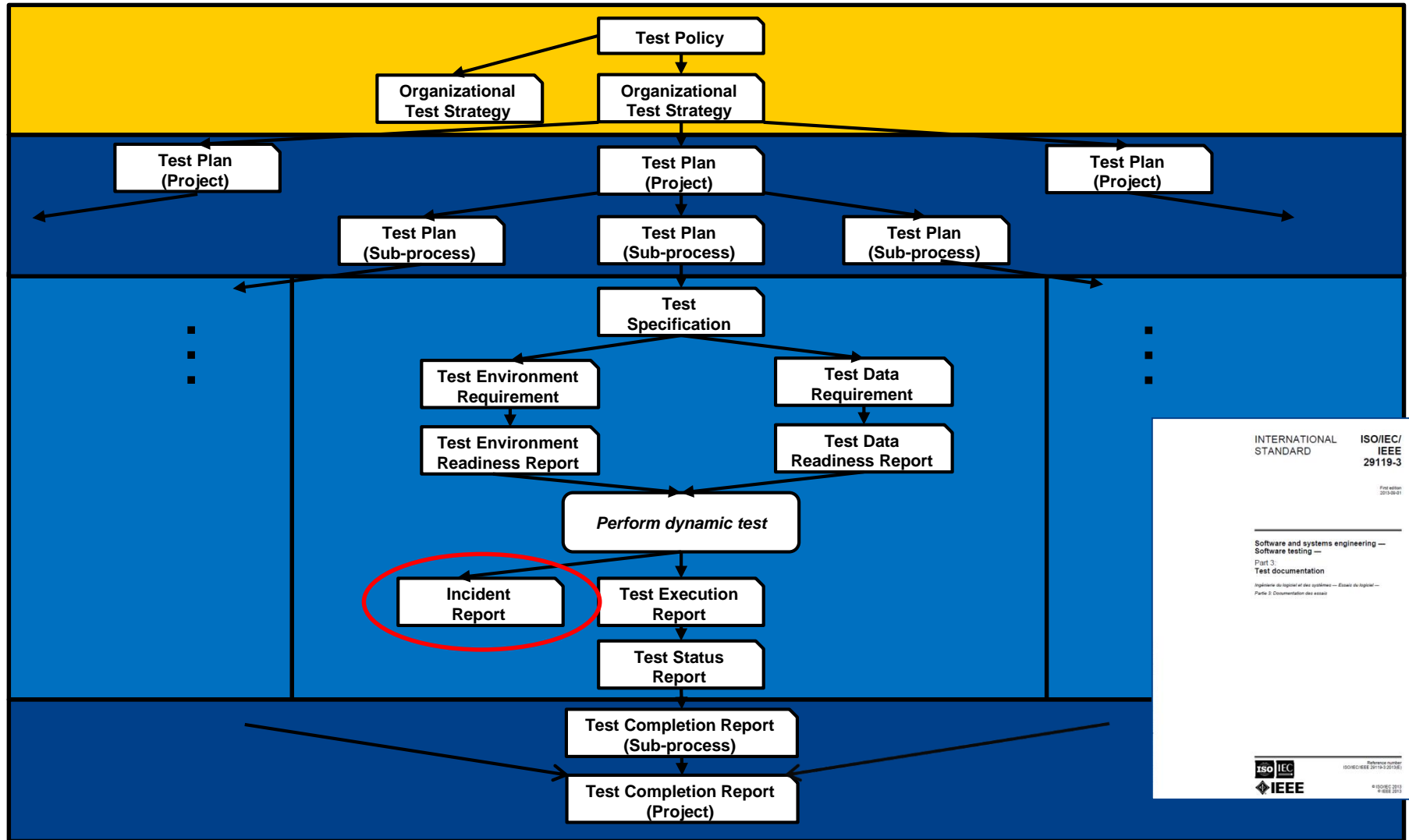
Exkurs: Testfortschrittsbericht



Glossar: Testfortschrittsbericht (engl. test progress report): Ein Dokument, das die Testaktivitäten und -ergebnisse zusammenfasst, und das in regelmäßigen Zeiträumen erstellt wird. Es berichtet über den Fortschritt der Testaktivitäten gegenüber einer definierten Vergleichsbasis (wie z.B. dem ursprünglichen Testkonzept) und kommuniziert Risiken und Alternativen, die eine Managemententscheidung erfordern.

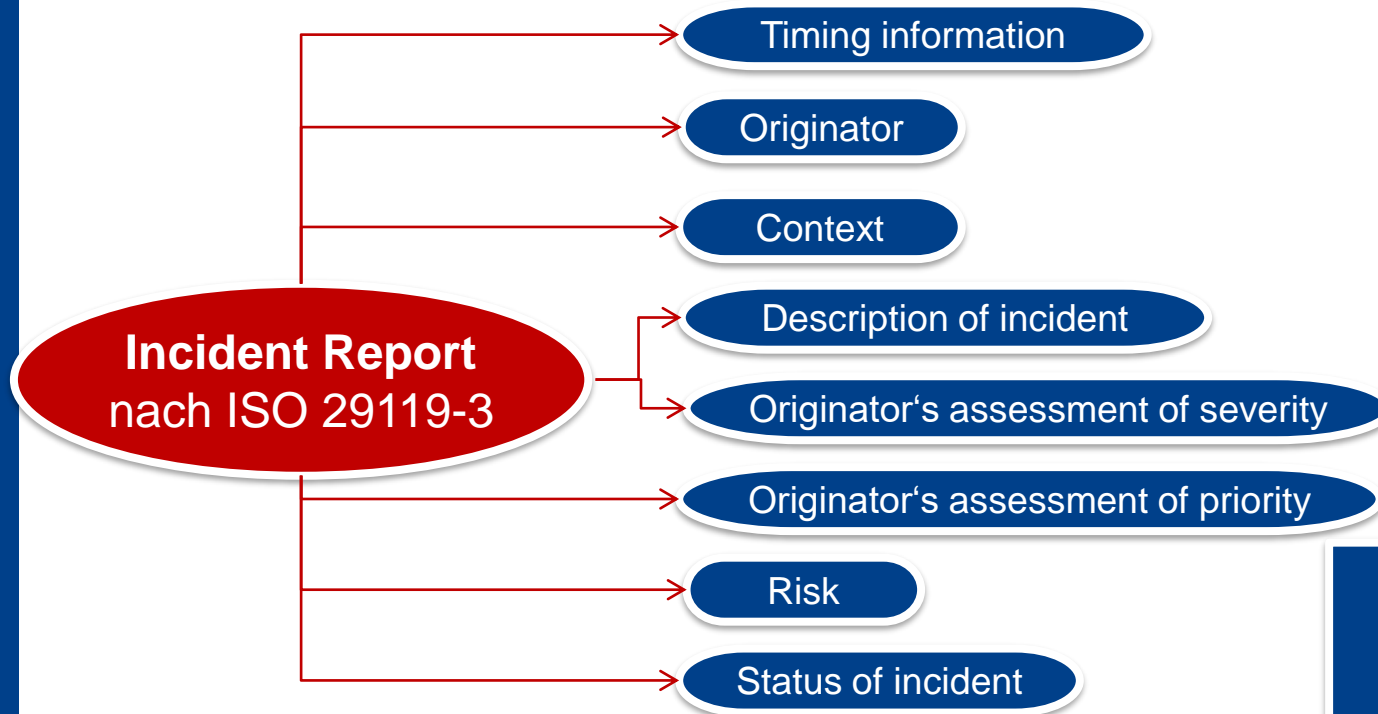


ISO/IEC/IEEE 29119-3 Test Documentation





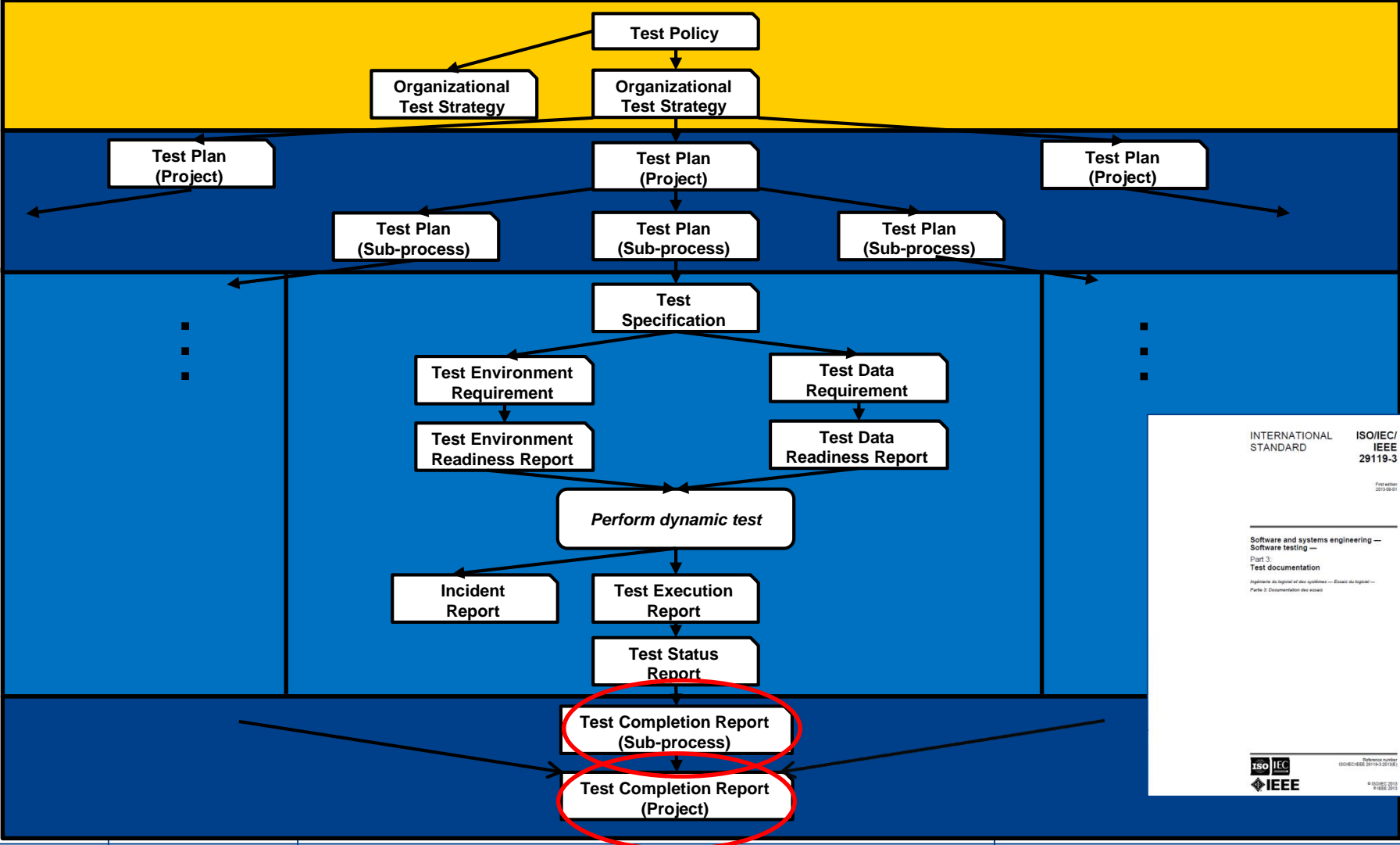
Exkurs: Fehlerbericht



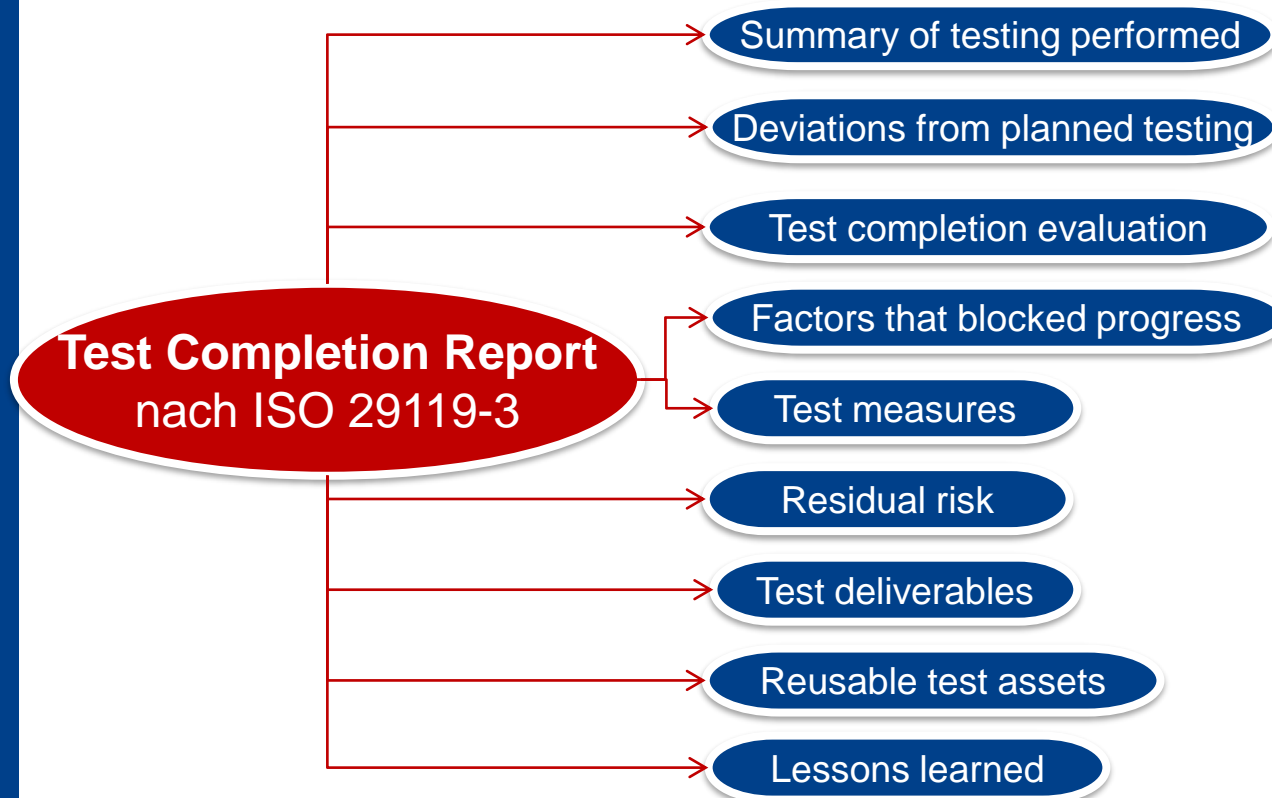
Glossar: Fehlerbericht (engl. incident report):
Die Dokumentation des Auftretens, der Art und des Status einer Abweichung.
[ISO 29119]



ISO/IEC/IEEE 29119-3 Test Documentation



Exkurs: Testabschlussbericht



Glossar:
Testabschlussbericht (engl. test summary report): Ein Dokument, welches die Testaktivitäten und -ergebnisse zusammenfasst. Es enthält eine Bewertung der durchgeführten Tests gegen definierte Endekriterien.
[nach IEEE 829]