Inhaltsverzeichnis

[1. Einleitung 5](#_Toc520787437)

[2. Testobjekte 6](#_Toc520787438)

[3. Prüfkriterien 7](#_Toc520787439)

[4. Qualitätsmerkmale 8](#_Toc520787440)

[5. Testziel 9](#_Toc520787441)

[6. Tools 10](#_Toc520787442)

[6.1. JIRA mit Zephyr (Empfehlung) 10](#_Toc520787443)

[7. Testarten 11](#_Toc520787444)

[8. Teststrategie 12](#_Toc520787445)

[9. Voraussetzungen für Testbeginn 13](#_Toc520787446)

[10. Testendekriterien 14](#_Toc520787447)

[11. Testabbruchkriterien 15](#_Toc520787448)

[12. Dokumentation 16](#_Toc520787449)

[13. Testumgebung 17](#_Toc520787450)

[14. Verantwortlichkeiten 18](#_Toc520787451)

[15. Testplan 19](#_Toc520787452)

[16. Risiken 20](#_Toc520787453)

[17. Anhang I 21](#_Toc520787454)

[18. Abbildungsverzeichnis 22](#_Toc520787455)

[19. Tabellenverzeichnis 23](#_Toc520787456)

# Einleitung

Durch das Testkonzept soll der Umfang und der Ansatz des Testens beschrieben werden und mit allen Beteiligten abgestimmt werden, sodass die Testumgebung und andere Anforderungen wie Testwerkzeuge und Testdaten frühzeitig mit berücksichtigt und abgestimmt werden können.

Dieses Konzept kann ganz unterschiedliche Ausprägungen haben.

In diesem Kapitel wird das Projekt kurz beschrieben, z.B.

* Warum wird die Anwendung benötigt?
* Was wird in der Anwendung bearbeitet?
* Gibt es Schnittstellen?
* Vorgehensweise des Projekts

# Testobjekte

In diesem Kapitel werden die Testobjekte definiert. Sämtliche Testaktivitäten müssen sich auf Testobjekte beziehen. Ein Test wird stets auf einem Testobjekt durchgeführt.

Solche Testobjekte können sein:

* Anwendungsfälle
* User Story
* Anwendungsfunktion
* Komponente

Das „kleinste“ Testobjekt ist somit der kleinste Teil des Gesamtverfahrens, welcher im Rahmen der Tests betrachtet wird. Dabei ist zu berücksichtigen, dass die Wahl eines feingranularen Testobjektes zu genaueren Aussagen hinsichtlich des QS-Status befähigt, aber auch die Aufwände zur QS steigen lässt. Große Objekte sind sehr viel schwerer handhabbar und lassen sich aus Sicht der QS schwerer beschreiben.

# Prüfkriterien

Ein Prüfkriterium ist eine konkrete Prüfung der Software, welche im Rahmen der Abnahmetests durchgeführt werden soll. Sie ergibt sich in der Regel aus einer Anforderung. In Ausnahmen kann sie sich aus zusätzlichen Details der Systemspezifikation, oder aus technischen Quellen wie Systementwurf oder Systemhandbuch ergeben.

Für die Durchführung der Prüfung einer Prüfspezifikation werden Testabläufe spezifiziert. Sie enthalten eine Reihe von Prüfkriterien, welche der gleichen Kombination aus Testart und Testobjekt zugeordnet sind.

Der Unterschied zwischen den Prüfkriterien und den aus ihnen erzeugten Testabläufen ist es, dass in den Prüfkriterien steht, WAS getestet werden soll, während in den Testabläufen steht, WIE es geschieht.

Zur Erstellung der Prüfkriterien auf Basis der Anforderungen erzeugt man pro Anforderung eine Liste von Kriterien, mit welchen man die vollständige Anforderung oder einen Teil dieser Anforderung im Rahmen eines Tests prüfen möchte.

# Qualitätsmerkmale

In diesem Kapitel werden die Qualitätsmerkmale beschrieben, die dem System zu Grunde liegen.

Empfohlen werden z.B. die Qualitätsmerkmale nach ISO 25010

* Funktionalität (Angemessenheit, Richtigkeit, Vollständigkeit)
* Sicherheit (Vertraulichkeit, Integrität, Nachweisbarkeit, Ordnungsmäßigkeit, Authentizität)
* Zuverlässigkeit (Reife, Fehlertoleranz, Wiederherstellbarkeit, Verfügbarkeit)
* Benutzbarkeit (Verständlichkeit, Erlernbarkeit, Bedienbarkeit)
* Effizienz (Ressourcenverbrauch, Zeitverhalten, Kapazität)
* Wartbarkeit (Analysierbarkeit, Modularität, Wiederverwendbarkeit, Änderbarkeit, Testbarkeit)
* Übertragbarkeit (Anpassbarkeit, Installierbarkeit, Austauschbarkeit)

Es wird auch begründet, warum bestimmte Qualitätsmerkmale nicht getestet werden sollen, bzw. ob sie durch andere Tests geprüft werden.

# Testziel

Die Tests sollen sicherstellen, dass die in ISO 25010 genannten Qualitätsziele eingehalten werden. Diese Qualitätsziele werden durch folgende Testziele operationalisiert:

Beispiele für Testziele:

* Die Funktionalität der Anwendung muss sichergestellt werden:
* Es dürfen keine bekannten kritischen Fehler mehr bestehen
* 80% der Prüfkriterien müssen durch Testfälle abgedeckt sein und dürfen keine bekannten Fehler enthalten
* Die Prüfkriterien der Risikoklasse A1 müssen zu 100% durch Testfälle abgedeckt sein.

# Tools

In diesem Kapitel werden die Tools beschrieben, die in dem Projekt zum Einsatz kommen.

## JIRA mit Zephyr (Empfehlung)

Zur Dokumentation der Testfälle, Testsets und Testberichte wird JIRA/Zephyr eingesetzt.

Defects werden in JIRA angelegt und nachverfolgt.

# Testarten

Welche Testarten werden eingesetzt? Hier werden einige Beispiele aufgeführt, die beliebig ergänzt werden können. Es kann sinnvoll sein, für spezielle Tests, z.B. Last- und Performance, eigene Testkonzepte zu erstellen.

Modultest, Systemtest, Systemintegrationstest werden häufig durch die Entwickler durchgeführt.

Funktionaler Test:

* Sicherstellung der Funktionalität der Anwendung
* Wie werden die Testfälle abgeleitet? Prüfkriterien?

Benutzbarkeitstest:

* GUI-Komponenten mit Style Guide abgleichen
* Benutzbarkeit durch den Fachbereich abprüfen lassen

Regressionstest

* Regressionstests sollten nach Möglichkeit automatisiert werden

Nachtest

* Jedes Bug-Fixing muss durch einen Nachtest geprüft werden
* Ist der Nachtest erfolgreich, kann der Bug geschlossen werden
* Sollte der Nachtest fehlschlagen, ist der Bug an die Entwickler zurückzuweisen
* Sowohl ein erfolgreicher als auch ein fehlgeschlagener Nachtest wird im Bug kommentiert.

# Teststrategie

Zunächst geht es darum, die Teststrategie zu definieren. Ein wesentlicher Aspekt dabei ist eine Festlegung, wie intensiv getestet werden soll. Das hängt von unterschiedlichen Faktoren ab, beispielsweise dem verfügbaren Budget für das Testen, dem Risiko, der Komplexität des zu entwickelnden Systems, der Menge der Teams oder auch dem Wissen über Testaktivitäten.

Die Teststrategie muss für jedes Projekt neu erarbeitet werden. Folgende Checkliste soll dabei helfen,

* Kompatibilität zum eingeschlagenen Vorgehensmodell
* geringer Implementierungsaufwand
* Verwendung von Ergebnistypen aus dem SWE-Prozess
* Integration vorhandener Daten und Reports
* Einbeziehung von Expertenwissen
* Methodenvielfalt
* skalierbare Teststrategie und -aufwand
* Reduktion bestehender Produkt- und Projektrisiken
* Lieferung aussagekräftiger Testberichte
* Begründung von Qualitätssicherungsmaßnahmen

Für komplexere Anwendungen empfiehlt sich die Teststrategie des [risikobasierten Testens](https://wiki.zssi.ivbb.bund.de/display/SWEQM/Risikobasierter%2BTest).

Beim risikobasierten Testen wird versucht die Menge von Testfällen gering zu halten, in dem man nur Testfälle erstellt/ausführt, die große Risikofaktoren prüfen. Dazu werden Testobjekte in Risiko-Komplexität-Klassen eingeordnet. Zur Bestimmung der Komplexität können verschiedene Methoden, wie zum Beispiel Expertenbefragungen, Use Case Points Methode oder Auswertung der Defects eingesetzt werden. Zur Bestimmung des Risikos können zum Beispiel Experteninterviews oder Fragebögen genutzt werden.

Für Testobjekte der Risikoklassen A1, A2, B1 und B2 werden umfangreiche Testfälle spezifiziert, die sowohl positive als auch negative Systemreaktionen abprüfen. Für Testobjekte der Risikoklassen A3, B3, C1, C2 und C3 werden nur Testfälle spezifiziert, die positive Systemreaktionen abprüfen.

# Voraussetzungen für Testbeginn

Um mit einem Test beginnen zu könne, müssen bestimmte Voraussetzungen erfüllt sein.

Dies können z.B. sein:

* Die benötigten Funktionen werden bis zum Testbeginn in einem testbaren Zustand bereitgestellt. Es erfolgt ein Annahmetest.
* Die Testumgebung(en) stehen in der benötigten Qualität zur Verfügung.
* Die entsprechenden User müssen im den Testsystemen angelegt und berechtigt sein.
* Die benötigten Testdaten müssen strukturiert, konsistent und in vollem Umfang im System vorhanden sein.
* Ressourcen stehen ausreichend zu den erforderlichen Terminen zur Verfügung.
* Die Tester haben zu Beginn der Tests ausreichende Kenntnisse, um die Testfälle anhand der ihnen zur Verfügung gestellten Unterlagen möglichst selbstständig bearbeiten zu können.

# Testendekriterien

Die Kriterien für das Testende müssen im Vorfeld definiert werden, um zu wissen, wann das Qualitäts- und Testziel erreicht wurde.

Dies können z.B. sein:

* Die Erreichung eines definierten Testabdeckungsgrades
* Die Anzahl der verbliebenen Fehler ist geringer als die definierte Grenze
* Alle Kern-Tests sind bestanden

Als Testendekriterium die abgelaufene Zeit oder einen erreichten Termin zu wählen sollte vermieden werden, da es sich bei Testendekriterien um ein Qualitätsziele handelt und alle nachfolgenden Teststufen durch eine mögliche schlechte Qualität behindert werden oder im schlimmsten Fall nicht durchgeführt werden können.

# Testabbruchkriterien

Ist eine Fortsetzung des Tests wegen gravierender Mängel nicht sinnvoll, kann der Test abgebrochen und zu einem späteren Zeitpunkt wieder fortgesetzt werden.

Gründe dafür können z.B. folgende sein:

* Fehlende oder instabile Testumgebungen an mehr als drei aufeinanderfolgenden Werktagen.
* Ein Fehler der Klasse Kritisch vorliegt, der das gesamte Testteam an mehr als drei aufeinanderfolgenden Tagen daran hindert zu Testen.

# Dokumentation

Folgende Dokumentationen müssen erstellt werden, um in regelmäßigen Abständen die nötige Transparenz hinsichtlich des Test- und Testprozessfortschritts schaffen zu können.

Beispiele:

* Die Dokumentation der Testfälle, Testzyklen und Testausführungen findet in JIRA/Zephyr statt.
* Defects werden in JIRA erfasst und nachverfolgt.
* Qualitätsberichte
* Testberichte

Falls Jira nicht eingesetzt wird muss ein Testnachweis erfolgen, der folgende Mindestanforderungen enthält:

* Wer hat den Test durchgeführt (Person, Testumgebung, Rolle, Profil)
* Was wurde getestet
* Wann hat der Test stattgefunden
* Welche Zielsetzung/Aufgabe hat der Testfall
* Mit welchem Ergebnis hat der Test geendet ( sowohl positiv als auch negativ)
* Verweis auf das Defect Management und die verbliebenen Fehler

# Testumgebung

In diesem Kapitel wird/werden die Testumgebung/en beschrieben.

Nach der ISO 29119-3 werden folgende Elemente erwartet:

* ID
* Beschreibung
* Verantwortlichkeit
* Zeitraum für den Bedarf

Mehr Vorgaben sind nicht möglich, da Testumgebungen in Umfang und Komplexität äußerst unterschiedlich sein können.

Das Testkonzept sollte aber darlegen:

* welche Anforderungen an die Testplattform, z.B. in Form von Hardware, Betriebssystem und sonstiger Basissoftware, bestehen
* in welchem Umfang entsprechen Ressourcen bereitzustellen oder zu beschaffen sind
* welche sonstigen Voraussetzungen, z.B. Referenzdatenbanken oder Konfigurationsdaten, für den Test erforderlich sind
* über welche Schnittstellen das Testobjekt angesprochen wird, bzw. seine Reaktion überprüft wird
* ob Monitoring- und Protokollumgebungen notwendig sind
* welche Ansätze zur Testautomatisierung möglich sind
* ob zur Beschaffung der notwenigen Testdaten besondere Maßnahmen erforderlich sind (Datenschutzaspekte)

# Verantwortlichkeiten

In diesem Kapitel werden die Verantwortlichen genannt. Das können z.B. die Ansprechpartner des Fachbereiches sein, der Projektleiter, der Testmanager/Tester

# Testplan

Übersicht der Testphasen und Meilensteine einfügen.

# Risiken

In diesem Kapitel werden die Risiken erläutert, die verhindern könnten, dass das Projekt rechtzeitig realisiert wird oder die Software nicht in der gewünschten Qualität fertig gestellt wird.

# Anhang I

# Abbildungsverzeichnis

[Abbildung 1 - Mustergrafik 6](#_Toc282528989)

# Tabellenverzeichnis