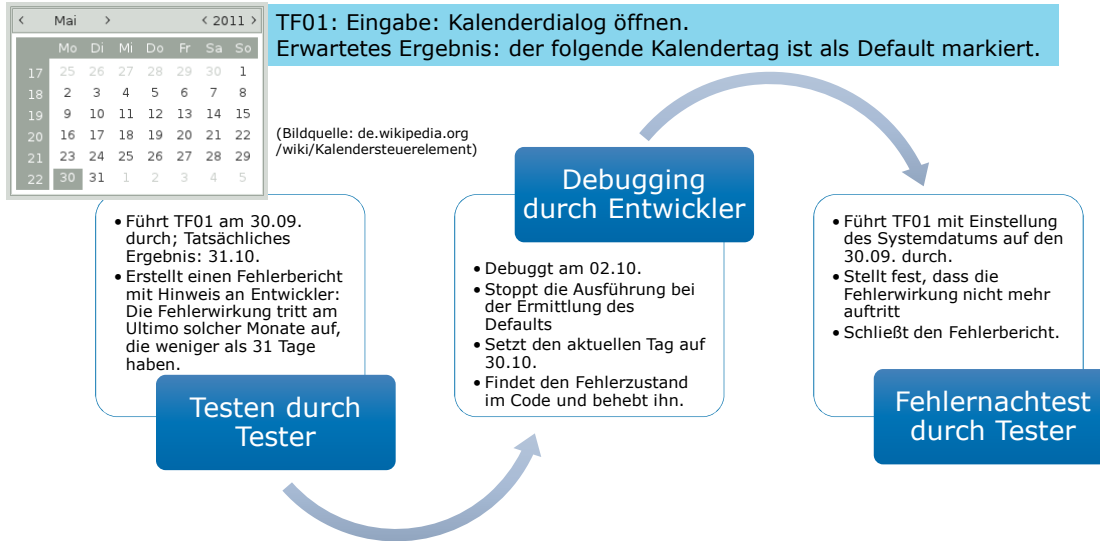




## Hinweise zur Beispielsammlung

- In diesem Dokument sind ergänzende Beispiele und Szenarien zu einzelnen inhaltlichen Aspekten der Lehrplan Kapitel beschrieben. Sie dienen dazu, die verschiedenen Lernziele des Lehrplans vertiefend mit den Teilnehmern diskutieren zu können.
- Die Beispiele sind bewusst nicht direkt in die übrigen Schulungsunterlagen integriert worden,
  - um sie unabhängig davon auf einem aktuellen Stand halten zu können, ohne die restlichen Schulungsunterlagen aktualisieren zu müssen,
  - um dem Dozenten die Möglichkeit zu geben, Beispiele flexibel in eine Schulung einbauen zu können,
  - um dem Dozenten Anstöße und Motivationen für das aktive Einbeziehen der individuellen Teilnehmererfahrungen zu erleichtern.
- Für jedes Beispiel werden im Notizbereich die zugehörigen Lernziele aus dem Lehrplan referenziert, es sind weitere Fragestellungen und Diskussionsaspekte beschrieben oder Lösungen zu Fragestellungen angegeben.

## 1.1.2: Testen und Debugging eines Kalenderdialogs



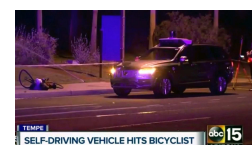
ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

3

## 1.2.1: Aktuelle IT-News

### Aktuelle Beispiele für Auswirkungen fehlerhafter Software

- 14 Millionen Facebook-Nutzer posten öffentlich, ohne es zu wollen (2018)
  - Quelle: <https://meedia.de/2018/06/08/software-fehler-14-millionen-facebook-nutzer-posteten-oeffentlich-ohne-es-zu-wollen/>
- Mazda ruft weltweit mehr als 100.000 Fahrzeuge zurück (2018)
  - Quelle: Westfälische Nachrichten, <https://www.wn.de/Welt/Wirtschaft>
- Softwarefehler im selbstfahrenden Auto führt zu tödlichem Unfall (2018)
  - Quelle: <https://www.handelsblatt.com/unternehmen/industrie/autonomes-fahren-softwarefehler-fuehrte-wohl-zu-toedlichem-uber-unfall/21254056.html?ticket=ST-8448644-USrZS0jE6ImaD2cDPQ1v-ap3>



ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

4

## 1.2.2: 3D-Drucker

1. Die Software zur Ansteuerung eines 3D-Druckers wurde nicht ausreichend getestet. Durch einen Softwarefehler ist der Druck nicht brauchbar, das Material verschenkt.
2. Es wurde zum Drucken falsches Material benutzt, der Druck ist deshalb nicht brauchbar, das Gerät muss gereinigt werden.

Fragen:

- Was lässt sich durch Testen beheben?
- Was durch andere Arten der Qualitätssicherung?
- Welche Rolle spielt die Qualitätssicherung im ersten Beispiel?
- Gäbe es eine Möglichkeit, auch im zweiten Beispiel durch Softwaretests die Qualität zu erhöhen?

## 1.2.3 und 1.2.4: Fehlerwirkungskette

Spezifikation:

Eintritt 20€



Rabatte:

Kinder unter 10 Jahre sind frei

Senioren über 65 erhalten 20% Rabatt

Junge Erwachsene zwischen 18 und 25 erhalten 25% Rabatt

Personen ab 45 Jahren erhalten 10% Rabatt



Welche Fehlhandlungen, Fehlerzustände und Fehlerwirkungen könnten sich aus dieser Spezifikation ergeben?

Welchen Rabatt würden sie für Personen im Alter von 18, 25 und 70 Jahren vorsehen?

Welche Grundursache führt zu diesen Fehlern?

### 1.3.1: Sieben Grundsätze

In einem Finanz-Projekt wird direkt nach der Erstellung der Anforderungen mit dem Design der Testfälle begonnen. Einige Tester, die zuvor in einem Automotive-Projekt gearbeitet haben, bekommen eine Einführung in das Testen von Finanzsoftware. Bei der Durchführung der Testfälle wird darauf geachtet sie immer wieder mit anderen Testdatensätzen durchzuführen. Nachdem im Bereich Zinsberechnung vermehrt Fehler aufgetreten sind, wurden hier weitere Testfälle erstellt um die Abdeckung zu erhöhen. Es wurde jede Anforderung mit mindesten zwei Testfällen getestet. Für Anforderungen mit einer hohen Priorität wurde zumindest die Entscheidungsabdeckung im Komponententest auf 100% festgelegt, auch wenn klar ist, das hiermit nicht alle Kombinationen durchgetestet und alle Fehler gefunden werden können. Es wurden insgesamt 135 kritische Fehler gefunden und behoben. Am Ende des Test sind alle Testfälle erfolgreich durchgeführt und alle Anforderungen erfüllt worden. Trotzdem gibt es aus der Produktion viel negatives Feedback der Anwender.

Finden Sie alle sieben Grundsätze in dieser Beschreibung?

### 1.4.1: Auswirkungen des Kontexts auf den Testprozess

Auf welche Aktivitäten im Testprozess (siehe Schaubild) haben die folgenden kontextabhängigen Faktoren einen Einfluss?

1. Die Spezifikation wird erst kurz vor der Testdurchführung an das Testteam geliefert.
2. Sie folgen einem Standard für sicherheitsrelevante Softwareprodukte in dem eine sehr detaillierte Berichterstattung gefordert wird.
3. Die Personen, welche die Testfälle durchführen sollen, werden erst kurz vor der Durchführungsphase benannt.
4. Sie haben einen sehr engen Zeitplan und sollen das bestmögliche herausholen.
5. Die Software wird agil entwickelt, und ist zu Beginn noch nicht vollständig spezifiziert
6. Die Software ist äußerst komplex
7. Das Projektteam besteht aus eher unerfahrenen Entwicklern und Testern
8. Sie müssen eine Standard-Testumgebungslandschaft nutzen

## 1.4.2: Die Testaktivitäten innerhalb des Testprozesses

In welcher Phase des Testprozesses finden die folgenden Aktivitäten statt?

1. Prüfung von Testergebnissen und -protokollen gegen spezifizierte Überdeckungskriterien
2. Vorbereitung von Testdaten und Sicherstellung, dass sie ordnungsgemäß in die Testumgebung geladen sind
3. Den Ansatz für das Erreichen der Testziele festlegen
4. Definition und Priorisierung der Testbedingungen für jedes Feature
5. Entwurf der Testumgebung und Identifizierung benötigter Infrastruktur und Werkzeugen
6. Vergleich der Istergebnisse mit den erwarteten Ergebnissen
7. Analyse der gewonnenen Erkenntnisse aus den abgeschlossenen Testaktivitäten
8. Erstellen eines Testabschlussberichts
9. Erstellen von Testsuiten aus den Testabläufen und automatisierten Testskripten
10. Entwurf und Priorisierung von Testfällen und Sets an Testfällen

## 1.4.3 (1): Arbeitsergebnisse des Testens

**Welches Arbeitsergebnis des Testens enthält jeweils die folgenden Texte?**

1. Als Testbasis dienen die User Stories aus dem Backlog, das Benutzerhandbuch aus Produktion, und die Risikoliste der Kundenbetreuer. Zu jedem Change Request werden Abnahmekriterien im Testmanagementsystem dokumentiert. Rückverfolgbarkeitsbeziehungen werden im Testmanagementsystem von den Abnahmekriterien zu den Testfällen gepflegt. Definition-of-done: Alle Abnahmekriterien sind getestet, mit maximal leichten Fehlern.
2. Es sind 12 Fehler offen, davon 2 kritisch, 5 schwer, und 5 leicht. Für morgen erwarten wir Fehlerbehebungen zum Nachtesten. Das Feature „Artikelauswahl“ ist dabei sehr fehleranfällig, hier sehen wir das Risiko, dass weitere schwere Fehler ohne zusätzliche Tests unentdeckt bleiben.
3. Für die Artikelauswahl soll mindestens getestet werden:
  - Suche mit 0, 1, oder mehreren Treffern;
  - Alle Filtermöglichkeiten einzeln, und einige in Kombination;
  - Artikel mit allen Lieferstatus (sofort lieferbar, nicht lieferbar, lieferbar in x Tagen)

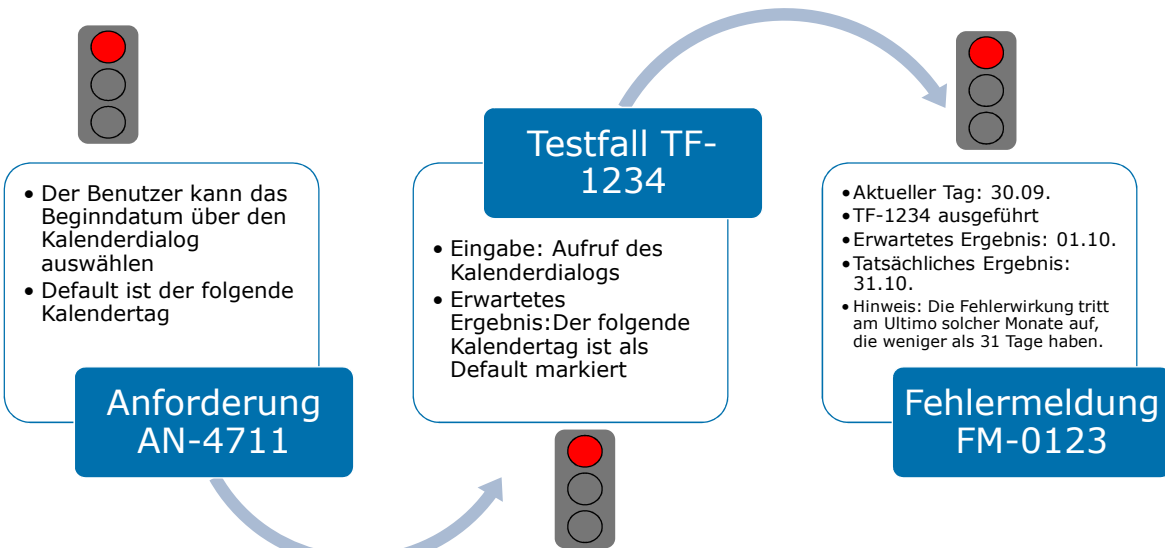
### 1.4.3 (2): Arbeitsergebnisse des Testens

4. Vorbedingungen: Benutzer hat die Artikelsuche aufgerufen;  
Eingabe: Artikelgruppe auswählen; Erwartetes Ergebnis: Die Liste von allen Pedelec Modellen wird angezeigt.  
Eingabe: Spaltenüberschrift Preis anklicken; Erwartetes Ergebnis: Die Liste wird aufsteigend nach Preis sortiert.  
Eingabe: Einen Artikel auswählen; Erwartetes Ergebnis: Die Details des Artikels werden angezeigt.  
Nachbedingung: Der Artikel kann bestellt werden, wenn er auf Lager ist.
5. Gebrauchte werden: mindestens drei Artikelgruppen; Pro Artikelgruppe mindestens drei Modelle mit unterschiedlichen Preisen.
6. Vorbedingungen herstellen: Registrierter Benutzer meldet sich im Onlineshop an und wählt die Artikelsuche. Prüfen ob der Warenkorb leer ist und ggf. löschen.  
Testfall 1 ausführen: vorhandenen Artikel suchen  
Testfall 5 ausführen: Artikeldetails anzeigen  
Testfall 12 ausführen: Artikel bestellen  
Aufräumen: Warenkorb leeren, Benutzer abmelden.

### 1.4.3 (3): Arbeitsergebnisse des Testens

7. Eine Auswahl an Testabläufen aus dem Regressions-Testset, so dass alle User Stories zum Testelement „Artikelsuche“ mindestens einmal ausgeführt sind.
8. Am Montag: Smoke Test, danach alle Artikelsuchen mit Priorität „hoch“;  
Am Dienstag: Artikelbestellungen, Artikelsuchen mit Priorität „mittel“; Tagesabschluss 1;  
Am Mittwoch: Artikelsuchen mit Priorität „niedrig“; Storno; Reklamationen; Tagesabschluss 2.

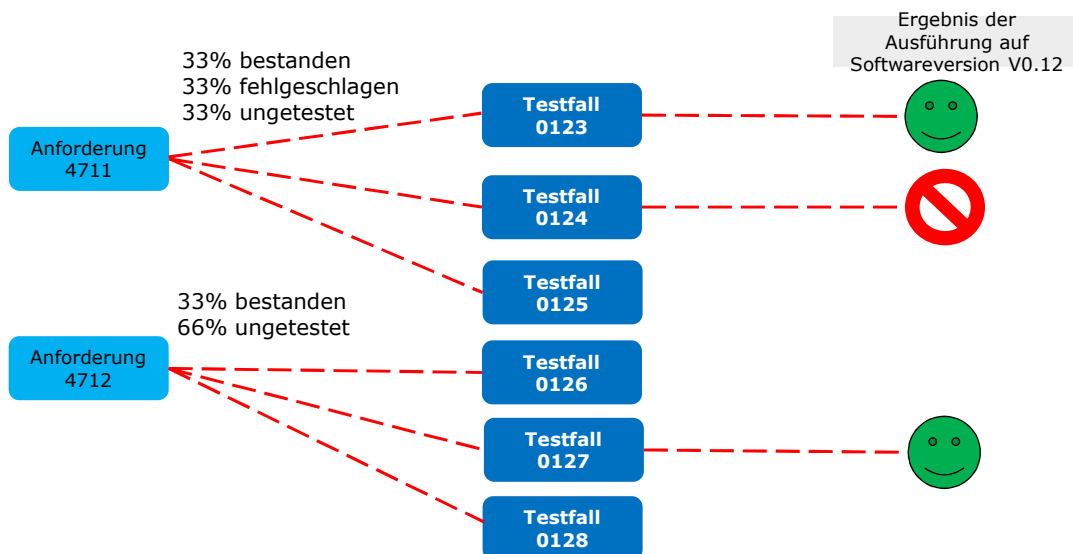
### 1.4.4 (1): Rückverfolgbarkeit



ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

13

### 1.4.4 (2): Verfolgbarkeit

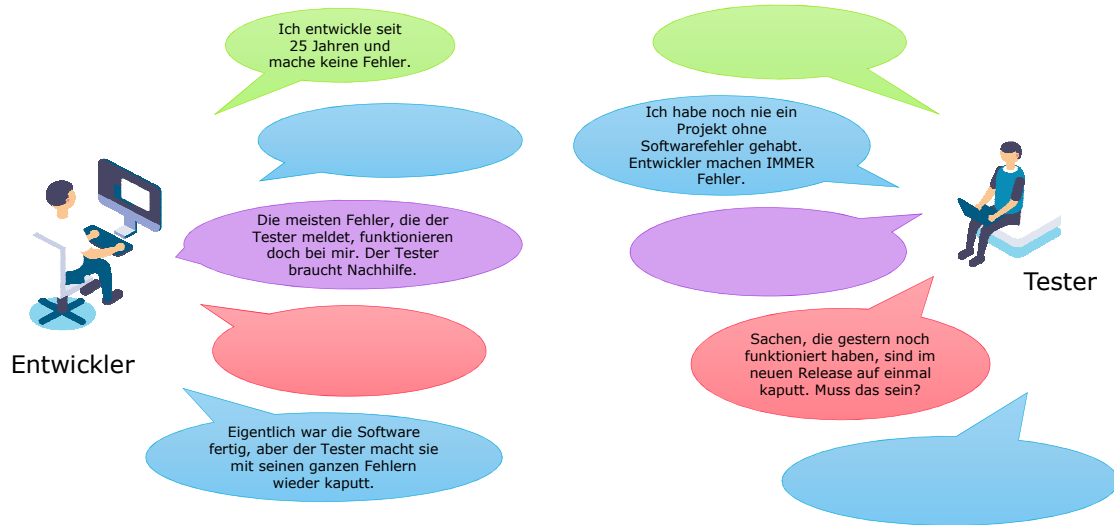


ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

14

## 1.5.2: Tester und Entwickler

Was entgegnet der Tester / Entwickler auf die Aussagen des jeweils anderen?



ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

15





### 2.1.1: „Komposit-V“ (kommerzielle Standardsoftware – COTS)

1. Die „Qbit Software AG“ entwickelt das kommerzielle Standardsoftwarepaket „Komposit-V“ für Komposit-Versicherungen
  - Entwicklungsansatz: Agile Vorgehensweise, mit wöchentlichen Iterationen und monatlichem Release an Kunden
  - Testaktivitäten:
    - Codeanalyse und -Review mit Fokus auf die Wartbarkeit
    - Komponententest
    - Integrationstest
    - Funktionaler Systemtest mit einer Pilotkonfiguration
    - Nicht-funktionale Tests für Kompatibilität (Browser, Datenbanksysteme usw.), Last- und Performanz, Installierbarkeit
2. Eine Versicherungsgesellschaft setzt „Komposit-V“ ein
  - Die Versicherung nutzt das Standardsoftwarepaket „Komposit-V“ und weitere Standardsoftware für Finanzbuchhaltung, Dokumentenmanagement, Personalverwaltung, BaFin-Berichte usw.
  - Der Lieferant von Komposit-V liefert jeden Monat ein neues Release, das von der IT-Abteilung konfiguriert wird.
  - Testaktivitäten:
    - Review der Konfiguration durch die IT-Abteilung
    - Systemtest: IT-Abteilung prüft die Konfiguration dynamisch
    - Systemintegrationstest: IT-Abteilung prüft die Schnittstellen von Komposit-V mit den Umsystemen
    - Benutzerabnahmetest: begründet das Vertrauen der Endbenutzer im Fachbereich
    - Betrieblicher Abnahmetest: Installierbarkeit, Benutzerverwaltung, IT-Sicherheit

### 2.2.1: Arten des Abnahmetests

1. Nach dem Befüllen der Datenbank wird ein Performanztest durchgeführt.
2. Es wird gegen vertraglich vereinbarte Abnahmekriterien geprüft.
3. Dem Product Owner wird im Sprint Review ein Demo vorgestellt.
4. Der Hersteller stellt eine App für einen begrenzten Nutzerkreis im Shop bereit.
5. Ein Systemadministrator testet das Backup und die Wiederherstellung.
6. Die Informationssicherheit wird anhand eines Penetrationstests überprüft.
7. „Friendly Users“ werden zu einem Test beim Softwarehersteller eingeladen.
8. Ein Medizinprodukt wird von der Gesundheitsbehörde abgenommen.
9. Die Bereitschaft des Systems in einer realen Betriebsumgebung wird validiert.
10. Das Anlegen, aktualisieren und löschen von Benutzeraccounts wird getestet.

## 2.2.1 (1): Teststufen

	Komponententest	Integrations-test	Systemtest	Abnahmetest
Ziele				
Testbasis				
Testobjekte				
Fehler				

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

19

## 2.2.1 (2): Teststufen

**Aufgabe: Ordnen Sie die Sätze den Feldern in der Tabelle zu.**

1. Datenbankmodule
2. Feinentwurf
3. Microservices
4. Verifizierung, ob die funktionalen und nicht-funktionalen Verhaltensweisen der Schnittstellen dem Entwurf und der Spezifikation entsprechen
5. Finden von Fehlerzuständen in der Komponente
6. Modelle des Systemverhaltens
7. Versagen des Systems bei der ordnungsgemäßen Arbeit in der Produktivumgebung
8. Sequenzdiagramme
9. Falsche Daten, fehlende Daten oder falsche Datenverschlüsselung
10. Vorschriften, rechtliche Verträge und Standards
11. Systemkonfiguration und Konfigurationsdaten
12. Validieren, ob das System vollständig ist und wie erwartet funktionieren wird
13. Datenflussprobleme
14. Verhindern, dass Fehlerzustände an höhere Teststufen oder in die Produktion weitergegeben werden
15. Wiederherstellungssysteme und Hot Sites
16. Systemworkflows erfüllen nicht die Fach- oder Benutzeranforderungen

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

20

### 2.3.1 (1): Testarten

	Komponententest	Integrationstest	Systemtest	Abnahmetest
Funktionaler Test				
Nicht-funktionaler Test				
White-Box-Test				

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

21

### 2.3.1 (2): Testarten

**Aufgabe: Ordnen Sie die Sätze den Feldern in der Tabelle zu.**

1. Abdecken der Geschäftsregeln
2. Prüfen von Feldeingaben für jede Komponente oder jedes Programm
3. Prüfen wie schnell Informationen zwischen den Komponenten übertragen werden
4. End-to-End Test des gesamten Systems
5. Abdeckung der Menüfunktionen und Navigation
6. Abdeckung der Modulaufrufe und Schnittstellen
7. Prüfen der Zuverlässigkeit des Systems
8. Prüfen der Robustheit und Wartungsfreundlichkeit, sowie Suche von Speicherlecks und Ressourcenproblemen
9. Anweisungsüberdeckung, Entscheidungsüberdeckung
10. Prüfen und sicherstellen ob es das tut, was es aus Kundensicht tun soll
11. Prüfen ob das System aus Kundensicht nutzbar ist
12. Prüfen des Austauschs von Informationen zwischen Modulen und Systemen.

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

22

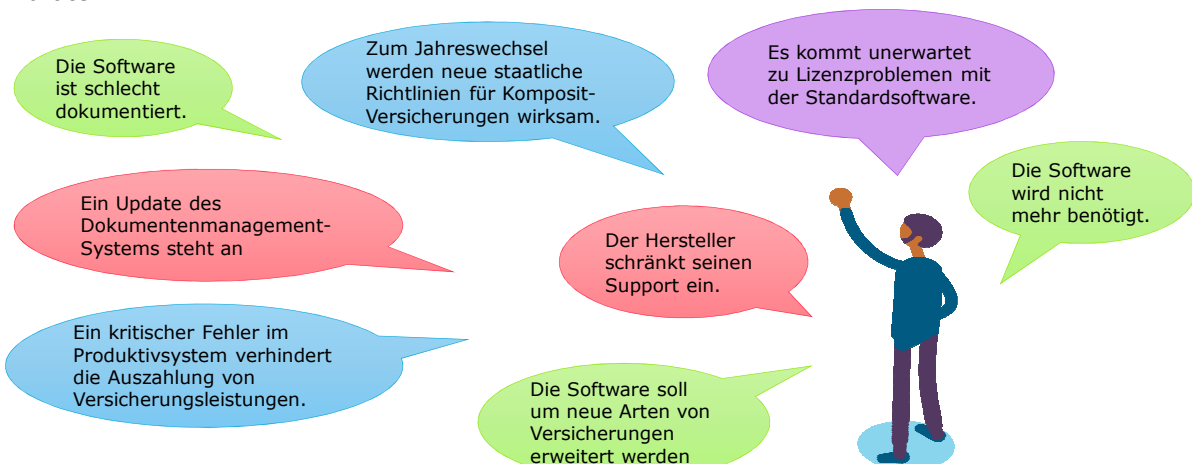
### 2.3.3: Fehlernachtest und Regressionstest

Die Entwicklung hat den Fehler FM-0123 zum folgenden Kalendertag in der Kalenderroutine behoben (vgl. Beispiel zu FL-1.4.4). Der Tester führt folgende Testfälle aus. Was davon ist Fehlernachtest bzw. Regressionstest?

Testfall	Vorbedingung	Eingabe/Aktion	Erwartetes Ergebnis	Nachbedingung
TF-1233 (bereits spezifiziert)	Aktueller Tag = 12.05.; Cursor auf Beginndatum	Aufruf des Kalenderdialogs	Default: 13.05.	Benutzer kann ein Datum $\geq$ 13.05. auswählen
TF-1234 (bereits spezifiziert)	Aktueller Tag = 30.09.; Cursor auf Beginndatum	Aufruf des Kalenderdialogs	Default: 01.10.	Benutzer kann ein Datum $\geq$ 01.10. auswählen
TF-2345 (neu)	Aktueller Tag = 28.02.; Cursor auf Beginndatum	Aufruf des Kalenderdialogs	Default: 01.03.	Benutzer kann ein Datum $\geq$ 01.03. auswählen
TF-2346 (neu)	Aktueller Tag = 30.09.; Cursor auf Antragsdatum	Aufruf des Kalenderdialogs	Default: 30.09.	Benutzer kann ein Datum $\leq$ 30.09. auswählen

### 2.4.1: Wartungsanlässe

In welchen Fällen ist eine Software-Wartung der Versicherungssoftware Komposit-V (siehe 2.1.1) anzuraten?

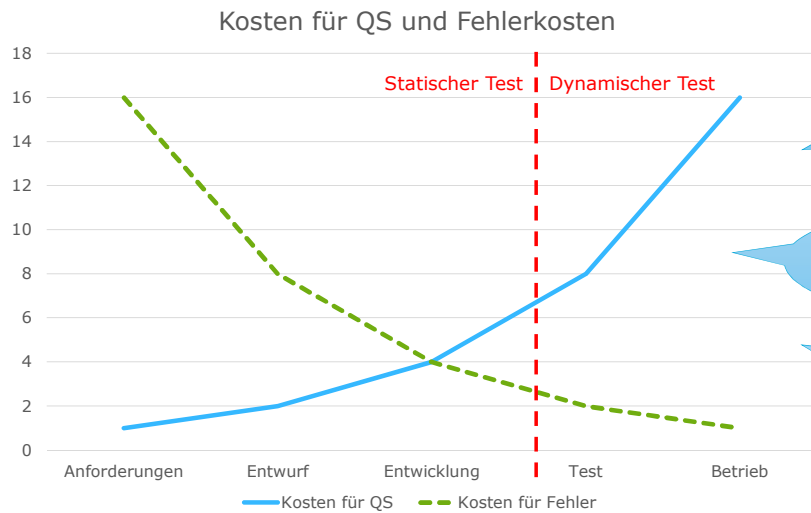


## 2.4.2: Auswirkungsanalyse

- Ein Unternehmen nutzt zahlreiche MS-Office Applikationen (z.B. Makros und Add-Ins) in Visual Basic, um seine Geschäftsprozesse zu unterstützen. Die Fachbereiche spezifizieren dazu die Anwendungsfälle. Die IT entwickelt und wartet die Applikationen.
- Das Unternehmen hat beschlossen, die MS-Office Plattform von der Version 10 (mit 32-Bit Adressierung) auf die Version O365 (64-Bit) zu ändern. Der Hersteller von MS-Office bietet ein Codeanalysewerkzeug an, das die Programmelemente in Visual Basic markiert, welche nur in der 32-Bit Version funktionieren und für 64-Bit angepasst werden müssen. (Z.B. wenn Adressen in 32-Bit langen Variablen gespeichert werden.)
- Die IT-Entwicklung führt die werkzeuggestützte Codeanalyse aus und passt die betroffenen Applikationen an. Sie klassifiziert die Applikationen in drei Kategorien: Unverändert / Leicht geändert / Wesentlich geändert.
- Die IT-Infrastruktur stellt für den Wartungstest eine Testumgebung mit der neuen MS-Office Version O365 zur Verfügung.
- Entwickler in der IT führen den Komponententest der angepassten Applikationen aus. Unabhängige Fachtester führen den Systemtest aus.
- Wie sollte die Codeanalyse diese Wartungstests beeinflussen?



### 3.1.2: Wert des statischen Tests



Ein im Entwurf entstandener Fehler wurde erst im Betrieb gefunden

Eine missverstandene Anforderung wird erst im Abnahmetest mit dem Kunden geklärt

Der Entwickler ignoriert Compiler Warnings und die Software ist nicht lauffähig

Da der Entwurf viel zu allgemein gehalten ist, programmieren die Entwickler, was sie wollen

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

27

### 3.1.3: Statisch oder dynamisch?

Ein System wird getestet, das die Forderungen des Unternehmens gegenüber Partnern verwaltet. Sind statische oder dynamische Verfahren besser geeignet, um die folgenden Fehlerzustände zu finden?

1. Die Anforderungen legen nicht fest, wie verfahren wird, wenn eine offene Forderung bei Insolvenz des Partners abgeschrieben werden soll.
2. Das im Anwendungsfall „Abschreibung“ spezifizierte Systemverhalten ist nicht korrekt implementiert.
3. Die Schnittstelle zwischen den Client- und Server-Komponenten enthält unnötig viele Daten.
4. Für die vielen und flexiblen Auswertungen ist die Datenbankstruktur ineffizient ausgelegt.
5. Die neue Auswertung des Forderungsbestands für die Aufsichtsbehörde ist beim erwarteten Datenvolumen innerhalb einer Stunde durchführbar.
6. Nach den Tuningmaßnahmen im Wartungsrelease funktioniert das System weiterhin korrekt, bei besseren Antwortzeiten.
7. Nach den vielen Änderungsgenerationen könnte das System unerreichbaren (toten) Code in erheblichem Umfang enthalten, der bei jedem Release teuer mitgewartet werden muss.
8. Das System belegt den Betrag der Abschreibung in der Schnittstelle zur Aufsichtsbehörde mit dem Betrag in einer Landeswährung, die von Euro abweichen kann, obwohl die Aufsichtsbehörde den Betrag in Euro erwartet.
9. Die Dialoge haben Sicherheitsschwachstellen gegen manipulierte Eingabedaten wie Überlängen oder eingebettete Programmanweisungen.
10. Die Rückverfolgbarkeit von Tests zu den Anforderungen sind nicht korrekt gepflegt und führen zur trügerischen Annahme, dass alle Akzeptanzkriterien erfolgreich getestet sind.

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

28

### 3.2.1: Reviewprozess

Die Fachabteilung hat ein wichtiges Dokument fertiggestellt. Zur Vorbereitung der Testentwicklung setzt der Reviewleiter ein 2-stündiges Ad-Hoc Review durch die Tester an, das in Form eines Walkthrough durchgeführt werden soll. Dazu wird das Dokument zunächst an alle Gutachter verteilt, die das Dokument daraufhin lesen sollen, um sich Fragen, Fehler oder Empfehlungen zu notieren. Anschließend wird das Dokument zurückgegeben an die Fachabteilung, die die gesammelten Befunde einarbeiten soll.

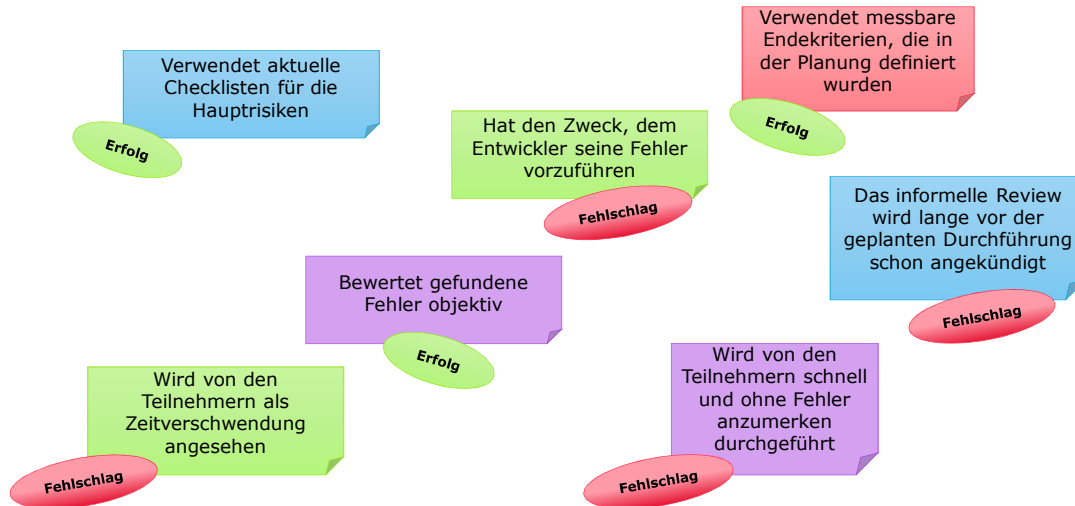
Welche Aktivitäten des Reviewprozesses sind im Text zu erkennen, welche Aktivität fehlt?

### 3.2.3: Reviewarten

1. Es soll ein Review für ein überschaubares Stück Code zu einer besonders kritischen Funktionalität einer Banking-Software durchgeführt werden.
2. Für umfangreiches Dokument benötigt der Autor schnelles Feedback
3. Kurz vor dem Urlaub möchte der Autor eines Fachkonzeptes die Bearbeitung und Fehlerkorrektur an einen Kollegen übergeben
4. In einem agilen Projekt sollen die erstellten Testfälle für unkritische Funktionen einem Review unterzogen werden.
5. Für eine Reihe von Geschäftsprozessen sollen Probeläufe „auf Papier“ durchgeführt werden, um potenzielle Fehler aufzudecken
6. Ein kritisches Dokument soll einem Review unterzogen werden, um über eine Grundursachenanalyse bereits entdeckter Fehler neue Fehlerzustände zu verhindern.
7. Für ein Dokument soll fachliches Feedback eingeholt werden, um die Qualität dieses Arbeitsergebnisses zu bewerten.

### 3.2.5: Erfolgreiche Reviews

Welches der Reviews wird wahrscheinlich erfolgreich sein, welches fehlschlagen?



ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

31





### 4.1.1: Black or White? Oder Erfahrungsbasiert?

1. Testbedingungen, Testfälle und Testdaten werden aus einer Testbasis abgeleitet
2. Eine Überdeckungsmessung ist in der Regel nicht möglich
3. Basieren auf einer Analyse der Architektur, dem Feinentwurf, der internen Struktur oder dem Code des Testobjekts
4. Konzentrieren sich auf die Eingaben und Ausgaben des Testobjekts
5. Testfälle können genutzt werden, um Lücken zwischen Anforderungen und Realisierung sowie Abweichungen von den Anforderungen zu erkennen
6. Testet die erwartete Nutzung der Software, ihrer Umgebung, mögliche Fehlerzustände und die Verteilung dieser Fehlerzustände
7. Die Testbasis besteht nur aus den Erfahrungen der Tester, Entwickler, Benutzer und anderer Stakeholder
8. Die Überdeckung wird an Hand der Ermittlungstechnik, die auf die Testbasis angewendet wird, gemessen

### 4.2.5: Anwendungsfallbasierter Test

Ordnen Sie die verschiedenen Bestandteile eines Anwendungsfalls (links) zu den Bestandteilen von Testfällen (rechts) in einer anwendungsfallbasierten Testsuite zu.

**Anwendungsfall:**

- i. Vorbedingungen
- ii. Nachbedingungen
- iii. Haupt- und Alternativszenarien
- iv. Aktion des Akteurs
- v. Wahrnehmbare Systemreaktion

**Testsuite:**

- 1) Testfälle
- 2) Vorbedingungen
- 3) Eingabewerte
- 4) Vorausgesagte Ergebnisse
- 5) Erwartete Nachbedingungen

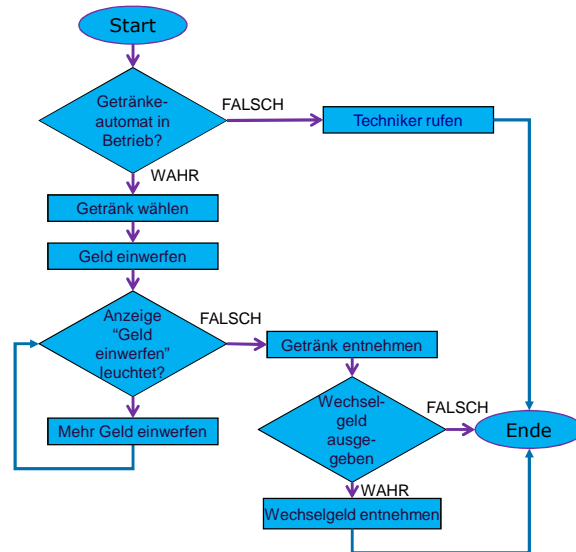
### 4.3.1: Anweisungstest

### 4.3.2: Entscheidungstest

```

IF (Getränkeautomat in Betrieb) THEN
    Getränk wählen
    Geld einwerfen
    WHILE (Anzeige "Geld einwerfen" leuchtet)
        Mehr Geld einwerfen
    ENDWHILE
    Getränk entnehmen
    IF (Wechselgeld ausgegeben) THEN
        Wechselgeld entnehmen
    ENDIF
ELSE
    Techniker rufen
ENDIF
  
```

Diskutieren Sie die Anweisungsüberdeckung und Entscheidungsüberdeckung, die jeweils durch beispielhafte Testfälle erreicht wird.



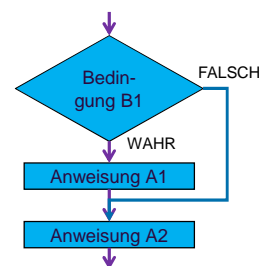
ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

35

### 4.3.3: Anweisungs- und Entscheidungsüberdeckung

Welche der folgenden Fehlerzustände in diesem Programmfragment werden durch welche Überdeckung aufgedeckt?  
 Welche dieser Fehlerzustände führen bei 100% Anweisungsüberdeckung bzw. 100% Entscheidungsüberdeckung mit Sicherheit zu Fehlerwirkungen?

1. Eine Ausführung von Anweisung A1 führt immer zum falschen Ergebnis.
2. Eine Ausführung von Anweisung A2 führt immer zum falschen Ergebnis.
3. Eine Ausführung von Anweisung A2 führt nur dann zum falschen Ergebnis, wenn A1 zuvor nicht ausgeführt wurde.
4. Ein Fehlerzustand in der Bedingung B1 führt dazu, dass sie immer nur auf FALSCH auswertet.
5. Ein Fehlerzustand in der Bedingung B1 führt dazu, dass sie immer nur auf WAHR auswertet.



ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

36

### 4.4.1: Intuitiv Testen

Sie entwickeln eine einfache Steuersoftware für einen 3D-Drucker. Diese kann folgende Steuerbefehle an den Druckkopf senden: Hoch/Runter, Rechts/Links, Vor/Zurück, eine bestimmte Menge Material abgeben und Pause. Die Steuerung des Druckers erfolgt durch ein einfaches Skript, das die Befehle listet. Weiterhin enthält das Skript Informationen über den Druckauftrag: Auftragsnummer, Name des Kunden und Preis des Produkts.

Der Drucker verwaltet intern die aktuelle Position des Druckkopfes und die aktuelle Menge an verfügbarem Druckmaterial. Falls ein Steuerbefehl nicht ausführbar ist, wird auf einem kleinen Display ein Fehler angezeigt, ansonsten wird die verbleibende Restzeit für das Druckprogramm berechnet und angezeigt.

Nach Fertigstellung des 3D-Drucks zeigt der Bildschirm die Informationen aus dem Druckauftrag (Auftragsnummer, Name des Kunden und Preis) an. Das Produkt kann dann entnommen werden.

Schildern Sie, wie Sie den 3D-Drucker intuitiv testen!

### 4.4.2: Explorativ Testen

Sie entwickeln eine einfache Steuersoftware für einen 3D-Drucker. Diese kann folgende Steuerbefehle an den Druckkopf senden: Hoch/Runter, Rechts/Links, Vor/Zurück, eine bestimmte Menge Material abgeben und Pause. Die Steuerung des Druckers erfolgt durch ein einfaches Skript, das die Befehle listet. Weiterhin enthält das Skript Informationen über den Druckauftrag: Auftragsnummer, Name des Kunden und Preis des Produkts.

Der Drucker verwaltet intern die aktuelle Position des Druckkopfes und die aktuelle Menge an verfügbarem Druckmaterial. Falls ein Steuerbefehl nicht ausführbar ist, wird auf einem kleinen Display ein Fehler angezeigt, ansonsten wird die verbleibende Restzeit für das Druckprogramm berechnet und angezeigt.

Nach Fertigstellung des 3D-Drucks zeigt der Bildschirm die Informationen aus dem Druckauftrag (Auftragsnummer, Name des Kunden und Preis) an. Das Produkt kann dann entnommen werden.

Schildern Sie, wie Sie den 3D-Drucker explorativ testen!

### 4.4.3: Mit Checklisten Testen

Sie entwickeln eine einfache Steuersoftware für einen 3D-Drucker. Diese kann folgende Steuerbefehle an den Druckkopf senden: Hoch/Runter, Rechts/Links, Vor/Zurück, eine bestimmte Menge Material abgeben und Pause. Die Steuerung des Druckers erfolgt durch ein einfaches Skript, das die Befehle listet. Weiterhin enthält das Skript Informationen über den Druckauftrag: Auftragsnummer, Name des Kunden und Preis des Produkts.

Der Drucker verwaltet intern die aktuelle Position des Druckkopfes und die aktuelle Menge an verfügbarem Druckmaterial. Falls ein Steuerbefehl nicht ausführbar ist, wird auf einem kleinen Display ein Fehler angezeigt, ansonsten wird die verbleibende Restzeit für das Druckprogramm berechnet und angezeigt.

Nach Fertigstellung des 3D-Drucks zeigt der Bildschirm die Informationen aus dem Druckauftrag (Auftragsnummer, Name des Kunden und Preis) an. Das Produkt kann dann entnommen werden.

Schildern Sie, wie Sie den 3D-Drucker mit einer Checkliste testen!



### 5.1.1: Unabhängiges Testen

Der Systemtest wird von einem Team an einem anderen Standort durchgeführt. Dabei werden unter anderem die folgenden Fehler gefunden:

1. Ein Kunde erhält bei einer bestimmten Kombination von Voraussetzungen einen falschen Rabatt zugewiesen.
2. In das Feld Vorname lassen sich nur 29 statt der gewünschten 30 Zeichen eingeben.
3. Umlaute im Namen werden ab einer bestimmten Stufe im Geschäftsprozess fehlerhaft angezeigt.

Hätten diese Fehler früher gefunden werden können und was wären die wesentlichen Vorteile dabei?

### 5.2.1: Testkonzept

Was ist der Zweck der folgenden Aussagen aus einem Testkonzept?

1. Es werden 3 Teststufen geplant, die von Entwicklern, Testern und Fachabteilung nacheinander durchzuführen sind.
2. Die Qualitätssicherung der Software wird für die vereinbarte Testüberdeckung der geforderten Funktionalitäten ca. 10.000 € kosten.
3. Die Software wird während des Testzeitraumes jeweils Montags, 8:00 für die Tester bereitgestellt.
4. Es werden Tests der Funktionalität, Performance, Benutzerfreundlichkeit und Sicherheit durchgeführt.
5. Testfälle werden nach den Testverfahren Äquivalenzklassenbildung, Entscheidungstabellentest und Entscheidungstest durchgeführt.
6. Während der Tests werden die verbrauchten Arbeitsstunden, die Anzahl der gefundenen Fehler, sowie die Anzahl der durchgeführten Testfälle gemessen.
7. Testdokumentation erfolgt nur innerhalb der dafür extra eingerichteten Software.

## 5.2.2 (1): Teststrategien

Die Qbit Bank migriert ihr Kontoführungssystem von einer Individualsoftware auf ein kommerzielles Standardsoftwarepaket. Dabei wurde das Standardsoftwarepaket analysiert, konfiguriert, Lücken gegenüber dem bestehenden Geschäftsprozessmodell der Qbit Bank identifiziert, und durch eigene Plug-Ins erweitert.

Das Testkonzept für die Migrationstests enthält die nachfolgenden Testvorgehensweisen.

Welche Arten von Teststrategien treffen AM MEISTEN für die jeweiligen Vorgehensweisen zu?

- 1. Die Transformationsprozesse der Daten von der alten zur neuen Struktur werden im Komponententest getestet. Für jede Transformationskomponente werden die Komplexität und die Kritikalität der Daten eingestuft, und daraus Risikostufen gebildet. Je nach Risikostufe muss folgende Überdeckung erreicht werden:*
  - *Niedrig: 100% der gültigen Äquivalenzklassen;*
  - *Mittel: 100% aller Äquivalenzklassen;*
  - *Hoch: 100% aller Äquivalenzklassen + 100% Entscheidungsüberdeckung im Code.*

## 5.2.2 (2): Teststrategien

- 2. Nach jedem Migrationslauf wird ein Geschäftsprozessstest durchgeführt. Getestet wird in der Testumgebung des Neusystems mit den migrierten Daten. Alle Aufgaben und Transitionen im Geschäftsprozessmodell der Qbit Bank sollen dabei überdeckt werden.*
- 3. Die Benutzungsoberfläche des Neusystems wird auf Konformität gegenüber den Look-and-Feel Standards der Qbit Bank statisch (Layouts) und dynamisch (Dialogführung) checklistenbasiert getestet.*
- 4. Das Bundesaufsichtsamt für Finanzwesen (BAFin) hat dieses Projekt als systemrelevant eingestuft. Deshalb muss die Testdokumentation den Standards des BAFin entsprechen. Insbesondere müssen die Testprotokolle durch Screenshots belegen, wer wann auf welcher Systemversion mit welchen Daten den Testfall durchgeführt hat.*
- 5. Im Benutzerabnahmetest validieren Experten aus dem Fachbereich sowie ausgewählte Key User das neue System. Aufgrund ihrer Expertise mit dem Geschäft der Qbit Bank wählen sie die zu testenden Abläufe und Konstellationen. Testkoordinatoren unterstützen sie bei der Erfassung von Testprotokollen und Fehlerberichten.*

### 5.2.2 (3): Teststrategien

6. *Die Datenmigration erfolgt inkrementell in Zyklen, angefangen mit unkritischen Sachkonten bis hin zu den komplexesten und geschäftskritischsten Konten der Großkunden. Bei jedem Zyklus werden die Testmittel (Testdaten, Testfälle, Testskripte usw.) der vorangehenden Zyklen wiederverwendet und weiter ausgebaut. Die neuen Testfälle eines Zyklus werden weitgehend automatisiert, und dem bestehenden Regressionstest zugefügt.*
7. *Das Neusystem enthält Leistungsmerkmale, welche im Altsystem nicht vorhanden waren und deren Benutzern noch nicht bekannt sind. Nach jedem Migrationszyklus werden diese Leistungen durch Experten aus dem Fachbereich und durch Key User explorativ getestet, um eventuelle dadurch entstehende Probleme (inkonsistente Daten, Inkompatibilität zu bestehenden Systemen etc.) zu erkennen.*

### 5.2.3 (1): Eingangs- und Endekriterien

Die Qbit Bank migriert ihr Kontoführungssystem von einer Individualsoftware auf ein kommerzielles Standardsoftwarepaket. Dabei wurde das Standardsoftwarepaket analysiert, konfiguriert, Lücken gegenüber dem bestehenden Geschäftsprozessmodell der Qbit Bank identifiziert, und durch eigene Plug-Ins erweitert.

Das Testkonzept sieht verschiedene Teststufen vor:

- *Die Transformationsprozesse der Daten von der alten zur neuen Struktur werden im Komponententest getestet.*
- *Im GUI-Test wird die Benutzungsoberfläche des Neusystems auf Konformität gegenüber den Look-and-Feel Standards der Qbit Bank statisch (Layouts) und dynamisch (Dialogführung) checklistenbasiert getestet.*
- *Nach jedem Migrationslauf wird ein Geschäftsprozessstest durchgeführt.*
- *Im Benutzerabnahmetest validieren Experten aus dem Fachbereich sowie ausgewählte Key User das neue System.*
- *Im betrieblichen Abnahmetest werden Performanz, IT-Sicherheit und Kompatibilität zur Systemlandschaft getestet.*

### 5.2.3 (2): Eingangs- und Endekriterien

*Welche Kriterien sind Eingangs- und welche sind Endekriterien? Für welche Teststufen sind sie relevant?*

- 1. Die Spezifikation für das Mapping der Daten vom Altsystem zum Neusystem ist verfügbar.*
- 2. Die vorgesehene Überdeckung der Transformationskomponente ist erreicht und es sind keine schweren Fehler offen.*
- 3. Das Geschäftsprozessmodell der Qbit Bank Kontoführung ist verfügbar.*
- 4. Die vorgesehene Überdeckung des Geschäftsprozessmodells ist erreicht, es sind keine schweren Fehler offen, und für alle mittelschweren Fehler gibt es zumindest ein akzeptiertes Workaround.*
- 5. Das Testmanagementwerkzeug ist für das Projekt konfiguriert.*
- 6. Der Rahmen für die Testautomatisierung ist aufgesetzt.*
- 7. Die Komponententests sind erfolgreich durchgeführt.*
- 8. Eine Testumgebung des Neusystems ist verfügbar.*
- 9. Die Daten des Testzyklus sind in die Testumgebung des Neusystems migriert.*
- 10. Die Geschäftsprozesstests sind erfolgreich durchgeführt.*

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

47

### 5.2.3 (3): Eingangs- und Endekriterien

- 11. Fachexperten bewerten das Restrisiko als gering.*
- 12. Das neue System ist auf einer im Sinne der IT-Sicherheit gehärteten Testumgebung installiert und die personenbezogenen Daten sind geladen.*
- 13. Die Checkliste für das GUI Look and Feel ist aktualisiert und vom Qualitätsmanagement freigegeben.*
- 14. Die Mengengerüste für das Datenvolumen und für die Geschäftstransaktionen sind ermittelt.*
- 15. Die gemessene Performanz des Neusystems mit dem Zielvolumen der Daten liegt innerhalb der spezifizierten Toleranzgrenzen.*

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

48



### 5.2.6: Metriken- oder Expertenbasiert?

Welche der folgenden Schätzverfahren sind eher metrikenbasiert und welche sind eher expertenbasiert?

1. Die Testbeteiligten besprechen die Testaufgaben. Jeder schätzt den Aufwand in Personentagen erst einmal alleine. Anschließend werden die Schätzungen verglichen. Differenzen werden besprochen, bis ein Konsens über die Anzahl Personentage erreicht wird. (Breitband-Delphi)
2. Bei einem Softwareentwicklungslebenszyklus nach dem Wasserfallmodell wird der Gesamtaufwand eines Projektes aufgrund der statistischen Daten im Unternehmen nach folgendem Verhältnis verteilt: 20% für Spezifikation, 50% für Entwicklung inkl. Komponententest, und 30% für den funktionalen Test. (Quelle: Tmap NEXT S. 452)
3. Ein externer Testberater untersucht die Mengengerüste und schätzt den Aufwand für den Performanztest aufgrund seiner Erfahrung in ähnlichen Projekten.
4. Der Umfang der Entwicklungsaufgabe wird in Function Points (nach IFPUG) vorgegeben. Der Testmanager errechnet den dazu nötigen Testumfang in Testpunkten nach den Formeln der Testpunktanalyse (TMap NEXT S. 456). Anschließend teilt er den Testumfang durch die gemessene durchschnittliche Produktivität der Tester (in Testpunkten pro Personentag), und schätzt somit den Aufwand in Personentagen.

### 5.3.2: Testberichte

Welchen Zweck und welche Zielgruppe haben die folgenden Aussagen aus einem Testbericht?

1. Im Testobjekt verbleiben nach Abschluss der Tests keine kritischen, aber 2 schwere Fehler. Das Projekt empfiehlt, die Software mit entsprechenden Workarounds in Betrieb zu nehmen.
2. Die Software enthält 3 hoch, 10 mittel und 2 niedrig priorisierte Fehler. Die hoch priorisierten Fehler sind zuerst zu beheben.
3. Der Teststatus ist grün. Es wurden 90% aller Testfälle durchgeführt.
4. Die Abnahmetester waren mit der Qualität der Software zufrieden.
5. Der Testzeitraum wurde um eine Woche verlängert.

## 5.4.1: Konfigurationsmanagement

Für den Test eines 3D-Druckers werden folgende Testmittel benötigt:

- Durchzuführender Testfall
- Konfigurations-Skript für den Drucker, das vor dem Test eingespielt werden muss
- Druckmaterial
- Testdaten, mit denen getestet werden soll
- Der Drucker selbst
- Die zu testende Version der Druckersoftware

Die Entwickler nutzen ein Versionsmanagement-System, in das sie ihren Code einpflegen. Auf dieser Basis wird täglich eine neue Version der Software automatisch erstellt.

Welchen Nutzen hat es für den Test, wenn das Versionsmanagement-System der Entwickler mitbenutzt wird? Welche Dinge sollten unter Konfigurationsmanagement gestellt werden?

## 5.5.2: Produkt- oder Projektrisiko?

**Welche der folgenden Probleme beschreiben Produkt- und welche Projektrisiken?**

1. Verzögerungen können in der Lieferung, Aufgabenerfüllung oder Erfüllung von Endkriterien oder der Definition-of-Done auftreten.
2. Tester könnten ihre Bedürfnisse und/oder die Testergebnisse nicht ausreichend kommunizieren.
3. Eine Systemarchitektur könnte einige nicht-funktionale Anforderungen nicht angemessen unterstützen.
4. Benutzer, Mitarbeiter der Fachabteilung oder Fachexperten sind unter Umständen aufgrund von gegenläufigen Geschäftsprioritäten nicht verfügbar.
5. Ein Regelkreisverfahren könnte nicht korrekt gebaut sein.
6. Ein Drittanbieter liefert möglicherweise ein notwendiges Produkt oder eine notwendige Dienstleistung nicht oder wird insolvent.
7. Vertragliche Probleme können Probleme für das Projekt auslösen.
8. Eine bestimmte Berechnung könnte unter bestimmten Bedingungen nicht korrekt durchgeführt werden.
9. Die Anforderungen könnten unter den gegebenen Beschränkungen nicht erfüllt werden.
10. Rückmeldungen zum Benutzererlebnis könnten die Produkterwartungen nicht erfüllen.
11. Die Software führt vielleicht nicht die gewünschten Funktionen gemäß den Bedürfnissen von Benutzern, Kunden und/oder Stakeholdern aus.
12. Späte Änderungen können zu erheblichen Überarbeitungen führen.
13. Datenkonvertierung, Migrationsplanung und ihre Werkzeugunterstützung könnte zu spät kommen.
14. Schlechtes Fehlermanagement und ähnliche Probleme können zu kumulierten Fehlerzuständen und anderen technischen Schulden führen.
15. Die Antwortzeiten könnten für ein hochleistungsfähiges Abwicklungssystem nicht angemessen sein.
16. Die Software führt vielleicht nicht die gewünschten Funktionen gemäß ihrer Spezifikation aus.

### 5.5.3: Produktrisikoaanalyse

Für eine Anwendung haben Sie mit dem Projektteam eine Risikoanalyse durchgeführt und Ihre Anforderungen dabei in verschiedene Risikoklassen eingeteilt.

Basierend auf der Risikoanalyse können Sie im Testkonzept beispielsweise folgende Vorgaben machen:

#### Komponententest

Risiko niedrig: 50% Anweisungsüberdeckung

Risiko mittel: 100% Anweisungsüberdeckung

Risiko hoch: 100% Entscheidungsüberdeckung

#### Systemtest

Risiko niedrig: Testen einiger zufälliger Werte

Risiko mittel: Äquivalenzklassenbildung

Risiko hoch: Äquivalenzklassenbildung inklusive Grenzwerteanalyse

ISTQB® CTFL Beispielsammlung | © 2019 Sogeti. All rights reserved.

53



## 6.1.1: Werkzeugtypen

Welche Bereiche werden durch die folgenden Werkzeuge unterstützt?

1. Ein Werkzeug, mit dem aus einem UML-Geschäftsprozessdiagramm automatisch Testfälle erstellt werden können, die eine spezifische Abdeckung erreichen.
2. Ein Werkzeug welches für die durchgeführten Testfälle ermittelt, welcher Grad der Anweisungsüberdeckung erreicht wurde.
3. Ein Werkzeug in dem Fehlermeldungen erfasst und zu den entsprechenden Testfällen und Anforderungen rückverfolgt werden können.
4. Ein Werkzeug mit dem mehrere Personen gleichzeitig ein Dokument betrachten und bearbeiten können.
5. Ein Werkzeug mit dem untersucht werden kann ob die Anwender mit der Anwendung gut umgehen können.
6. Ein Werkzeug mit dem die CPU-Auslastung während spezifischer Belastungen gemessen werden kann.



This message contains information that may be privileged or confidential and is the property of the Capgemini Group.  
Copyright © 2018 Sogeti. All rights reserved.



### About Sogeti

Sogeti is a leading provider of technology and engineering services. Sogeti delivers solutions that enable digital transformation and offers cutting-edge expertise in Cloud, Cybersecurity, Digital Manufacturing, Digital Assurance & Testing, and emerging technologies. Sogeti combines agility and speed of implementation with strong technology supplier partnerships, world class methodologies and its global delivery model, Rightshore®. Sogeti brings together more than 25,000 professionals in 15 countries, based in over 100 locations in Europe, USA and India. Sogeti is a wholly-owned subsidiary of Capgemini SE, listed on the Paris Stock Exchange.

Learn more about us at  
[www.sogeti.com](http://www.sogeti.com)