

**Familienname, Vorname:** \_\_\_\_\_

**Firmenadresse:** \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Telefon:** \_\_\_\_\_

**Fax:** \_\_\_\_\_

**E-Mail-Adresse:** \_\_\_\_\_

**Rechnungsanschrift:** \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Schulungsunternehmen:** \_\_\_\_\_

**Referent:** \_\_\_\_\_

---

**Foundation Level Probepfprüfung**  
**SET B (v1.3.1c) – GTB edition –**

---

---

**CTFL Syllabus Version v4.0**

---

**ISTQB® Certified Tester Foundation Level**

## Legal

Copyright © 2023 International Software Testing Qualifications Board (im Folgenden ISTQB® genannt). Alle Rechte vorbehalten.

Die Autoren übertragen das Urheberrecht an das International Software Testing Qualifications Board (im Folgenden ISTQB® genannt). Die Autoren (als derzeitige Urheberrechtsinhaber) und das ISTQB® (als zukünftiger Urheberrechtsinhaber) haben sich auf die folgende Nutzungsbedingung geeinigt:

Jedes ISTQB®-Mitgliedsboard darf dieses Dokument übersetzen.

Verantwortlich für die ursprüngliche Fassung dieses Dokuments ist die ISTQB® Examination Working Group.

ISTQB® Working Group EXAM 2023

## Danksagung

Dieses Dokument wurde von einem Kernteam des ISTQB® erstellt: Laura Albert, Wim de Coutere, Arnika Hryszko, Gary Mogyorodi (technical reviewer), Meile Posthuma, Gandhinee Rajkomar, Stuart Reid, Jean-François Riverin, Adam Roman, Lucjan Stapp, Stephanie Ulrich, Yaron Tsubery und Eshraga Zakaria.

Das Kernteam dankt dem Review-Team: Amanda Alderman, Alexander Alexandrov, Jürgen Beniermann, Rex Black, Young jae Choi, Nicola De Rosa, Klaudia Dussa-Zieger, Klaus Erlenbach, Joëlle Genois, Tamás Gergely, Dot Graham, Matthew Gregg, Gabriele Haller, Chinthaka Indikadahena, John Kurowski, Ine Lutterman, Isabelle Martin, Patricia McQuaid, Dénes Medzihradzsky, Blair Mo, Gary Mogyorodi, Jörn Münzel, Markus Niehammer, Ingvar Nordström, Fran O'Hara, Raul Onisor, Dénes Orosz, Arnd Pehl, Horst Pohlmann, Nishan Portoyan, Ale Rebon Portillo, Stuart Reid, Ralf Reissing, Liang Ren, Jean-Francois Riverin, Lloyd Roden, Tomas Rosenqvist, Murian Song, Szilard Szell, Giancarlo Tomasig, Joanne Tremblay, François Vaillancourt, Daniel van der Zwan, André Verschelling und Paul Weymouth für ihre Vorschläge und Anregungen.

## Revision Historie

Version	Datum	Bemerkungen
1.3	30.05.2024	BETA_01 Version
1.3.1	30.06.2024	BETA_02 Version
1.3.1c	21.07.2024	Final GTB edition

## Einführung

Dies ist eine Probeprüfung. Sie hilft den Kandidaten bei ihrer Vorbereitung auf die Zertifizierungsprüfung. Enthalten sind Fragen, deren Format der regulären ISTQB®/GTB Certified Tester Foundation Level Prüfung ähnelt. Es ist strengstens verboten, diese Prüfungsfragen in einer echten Prüfung zu verwenden.

- 1) Jede Einzelperson und jeder Schulungsanbieter kann diese Probeprüfung in einer Schulung verwenden, wenn ISTQB® als Quelle und Copyright-Inhaber der Probeprüfung anerkannt wird.
- 2) Jede Einzelperson oder Gruppe von Personen kann diese Probeprüfung als Grundlage für Artikel, Bücher oder andere abgeleitete Schriftstücke verwenden, wenn ISTQB® als Quelle und Copyright-Inhaber der Probeprüfung bestätigt wird.
- 3) Jedes vom ISTQB® anerkannte nationale Board kann diese Probeprüfung übersetzen und öffentlich zugänglich machen, wenn ISTQB® als Quelle und Copyright-Inhaber der Probeprüfung bestätigt wird.
- 4) Zu fast jeder Frage wird genau eine zutreffende Lösung erwartet. Bei den Ausnahmen wird explizit auf die Möglichkeit mehrerer Antworten hingewiesen.

## **Allgemeine Angaben zur Probeprüfung:**

Anzahl der Fragen: 40

Dauer der Prüfung: 60 Minuten

Gesamtpunktzahl: 40 (ein Punkt pro Frage)

Punktzahl zum Bestehen der Prüfung: 26 (oder mehr)

Prozentsatz zum Bestehen der Prüfung: 65 % (oder mehr)

**Feedback zur vorliegenden Probeprüfung als Ganzes (40 Fragen) oder zu einzelnen Fragen wurden in den deutschsprachigen BETA-Versionen des SET B im Zeitraum März - Juni 2024 abgegeben von:** Jörn Münzel, Stephan Weissleder, Horst Pohlmann, Marc-Florian Wendland, Ecaterina Irina Manole, Jessica Heymann (Sogeti), Jürgen Beniermann (Sogeti und GTB), Sabine Gschwandtner (imbus), Markus Thaler (Qytera), Sabine Gschwandtner (imbus), Daniel Moretz (WAMECON Academy), Sören Schmock (ITGAIN), Joachim Schulz (sepp.med), Arnd Prehl (imbus) und Paul Müller (Software Quality Lab).

An der englischsprachigen Fassung waren beteiligt bzw. an den geänderten Fragen zur englischsprachigen Fassung: Ecaterina Irina Manole, Horst Pohlmann und Debbie Friedenber

Frage 1	FL-1.2.1	K2	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen beschreibt am besten, warum Testen im Softwareentwicklungslebenszyklus notwendig ist?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Dynamisches Testen ist die einzige Möglichkeit, die Qualität eines Testobjekts zu bewerten.	<input type="checkbox"/>
b)	Das Testen stellt sicher, dass die Benutzer die Bedürfnisse der Entwickler verstehen und nachvollziehen können.	<input type="checkbox"/>
c)	Testen wird ausschließlich durchgeführt, um regulatorische Standards zu erfüllen.	<input type="checkbox"/>
d)	Testen hilft Fehlerzustände aufzudecken und damit die Qualität des Testobjekts zu verbessern.	<input checked="" type="checkbox"/>

FL-1.2.1 (K2) Der Lernende kann Beispiele geben, warum Testen notwendig ist [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 1.2.1)

a) FALSCH – Dynamisches Testen ist zwar ein wichtiges Mittel zur Qualitätsbewertung, aber nicht das einzige Mittel. Es gibt auch andere Mittel wie z. B. Code-Reviews und Inspektionen.

b) FALSCH – Diese Aussage ergibt keinen Sinn. Die Benutzer müssen nicht die Bedürfnisse der Entwickler nachvollziehen können. Vielmehr wird durch Testen sichergestellt, dass die Bedürfnisse der Benutzer während des Entwicklungszyklus berücksichtigt werden (siehe Abschnitt 1.2.1, 3. Absatz)

c) FALSCH – Obwohl Testen dazu beitragen kann, regulatorischen Anforderungen zu entsprechen, hebt der CTFL 4.0 Syllabus hervor, dass Testen auch anderen wichtige Funktionen erfüllt, wie die Bewertung der Qualität und die Unterstützung von Freigabeentscheidungen im SDLC.

**d) KORREKT – Der CTFL 4.0 Syllabus erklärt, dass Testen Fehlerzustände identifiziert, die durch Debugging behoben werden können, was implizit zu einer höheren Qualität des Testobjekts beiträgt.**

Frage 2	FL-1.2.2	K1	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen ist zutreffend?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Qualitätssicherung ist ein korrektiver Ansatz	<input type="checkbox"/>
b)	Testen ist eine Form der Qualitätssteuerung	<input checked="" type="checkbox"/>
c)	Testen ist ein Synonym für Qualitätssteuerung	<input type="checkbox"/>
d)	Qualitätssteuerung ist ein präventiver Ansatz	<input type="checkbox"/>

FL-1.2.2 (K1) Der Lernende kann die Beziehung zwischen Testen und Qualitätssicherung wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Section 1.2.2)

- a) FALSCH – Qualitätssicherung ist kein korrigierender, sondern ein präventiver Ansatz (siehe [CTFL 4.0], Abschnitt 1.2.2, 3. Absatz, 1. Satz).
- b) KORREKT – Testen ist eine Form der Qualitätssteuerung (siehe [CTFL 4.0], Abschnitt 1.2.2, 1. Absatz, letzter Satz).**
- c) FALSCH – Testen und Qualitätssteuerung sind keine Synonyme. Testen ist eine der wichtigsten Formen der Qualitätssteuerung, andere (nicht testende) Formen sind formale Methoden wie Modellprüfung und Korrektheitsnachweis oder Simulation und Prototyping (siehe [CTFL 4.0], Abschnitt 1.2.2, 2. Absatz, letzter Satz).
- d) FALSCH – Qualitätssteuerung ist kein präventiver Ansatz, sondern ein korrigierender Ansatz (siehe [CTFL 4.0], Abschnitt 1.2.2, 2. Absatz, 1. Satz).

Frage 3	FL-1.3.1	K2	Punkte 1.0
---------	----------	----	------------

Ein Tester hat über einen Zeitraum von 5 Jahren Softwareanwendungen auf mobilen Geräten getestet. Über einen langen Zeitraum hat der Tester die bestehenden Testfälle nicht verändert und auch keine neuen Testfälle erstellt. Bei neueren Versionen der mobilen Plattform sind mehr Fehlerwirkungen von den Nutzern gemeldet worden.

Welchen Grundsatz des Testens hat der Tester nicht beachtet?

Wählen Sie EINE Option! (1 aus 4)

a)	Testen ist abhängig vom Umfeld	<input type="checkbox"/>
b)	Vollständiges Testen ist nicht möglich	<input type="checkbox"/>
c)	Tests nutzen sich ab	<input checked="" type="checkbox"/>
d)	Häufung von Fehlerzuständen	<input type="checkbox"/>

FL-1.3.1 (K2) Der Lernende kann die sieben Grundsätze des Testens erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

- a) FALSCH – Das Testen ist abhängig vom Umfeld bzw. dem Kontext, egal, ob manuell oder automatisiert (siehe [CTFL 4.0], Abschnitt 1.3; 6. Grundsatz). Das ändert jedoch nichts an der Tatsache, dass – wie oben beschrieben – immer weniger Fehler aufgedeckt werden.
- b) FALSCH – Vollständiges Testen ist, außer in trivialen Fällen, unmöglich, egal wie viel Aufwand in das Testen investiert wird (siehe [CTFL 4.0], Abschnitt 1.3; 2. Grundsatz). Dies führt aber nicht dazu, dass – wie oben beschrieben – immer weniger Fehler aufgedeckt werden.
- c) KORREKT – Der Grundsatz “Tests nutzen sich ab” (siehe [CTFL 4.0], Abschnitt 1.3, 5. Grundsatz) besagt, dass die Wiederholung unveränderter Tests beim Auffinden von Fehlerzuständen zunehmend ineffektiv wird („Wiederholungen haben keine Wirksamkeit“). Um neue Fehlerzustände zu finden, müssen bestehende Tests und Testdaten modifiziert und neue Tests erstellt werden.
- d) FALSCH – Grundsätzlich gilt, dass Fehlerzustände oft gehäuft in wenigen Komponenten auftreten (siehe [CTFL 4.0], Abschnitt 1.3; 4. Grundsatz). Dies steht jedoch in keinem Zusammenhang damit, dass – wie oben beschrieben – immer weniger Fehler aufgedeckt werden.

Frage 4	FL-1.4.3	K2	Punkte	1.0
---------	----------	----	--------	-----

**Betrachten Sie das folgende Testmittel:**

<b>Test Charta</b>	<b>#04.018</b>	<b>Sitzungsdauer: 1h</b>
<b>Erforsche</b>	<b>die Registrierungsseite</b>	
<b>Mit</b>	<b>verschiedenen fehlerhaften Eingabesätzen</b>	
<b>Zu entdecken</b>	<b>Fehler beim Registrierungsvorgang bei fehlerhaften Eingaben</b>	

**In welcher Testaktivität wird dieses Testmittel erstellt?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Testplanung	<input type="checkbox"/>
b)	Testüberwachung und -steuerung	<input type="checkbox"/>
c)	Testanalyse	<input type="checkbox"/>
d)	Testentwurf	<input checked="" type="checkbox"/>



FL-1.4.3 (K2) Der Lernende kann Testmittel, die die Testaktivitäten unterstützen, unterscheiden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0, Abschnitt 1.4.3)

- a) FALSCH – Unter Testplanung versteht man die Erstellung oder Aktualisierung eines Testkonzepts. Ein Testkonzept ist ein Dokument, das den Umfang, die Vorgehensweise, die Ressourcen und den Zeitplan der geplanten Testaktivitäten beschreibt. Eine Testcharta ist kein Testkonzept, sondern ein flexibleres und informelleres Dokument, das eine Testsitzung beim explorativen Testen anleitet (siehe [CTFL 4.0], Abschnitt 1.4.3).
- b) FALSCH – Die Testüberwachung und -steuerung besteht darin, den Status der Testaktivitäten zu überprüfen, Abweichungen von den geplanten oder erwarteten Ergebnissen festzustellen und über den Testfortschritt und die Qualität zu berichten. Eine Testcharta ist kein Bericht oder eine Statusaktualisierung, sondern ein Dokument, das festlegt, was zu untersuchen ist, wie es zu untersuchen ist und wonach bei explorativen Tests zu suchen ist (siehe [CTFL 4.0], Abschnitt 1.4.3).
- c) FALSCH – Unter Testanalyse versteht man die Analyse der Testbasis (z. B. Anforderungen, Design, Architektur usw.) und die Definition dessen, was getestet werden soll, in Form von Testbedingungen. Eine Testbedingung ist ein Element oder Ereignis einer Komponente oder eines Systems, das durch einen oder mehrere Testfälle verifiziert werden kann. Eine Testcharta ist keine Liste von Testbedingungen, sondern ein Dokument, das den Umfang, die Ziele und die Herangehensweise einer Testsitzung für exploratives Testen definiert (siehe [CTFL 4.0], Abschnitt 1.4.3).
- d) KORREKT – Der Testentwurf ist die Ableitung und Spezifikation von Testfällen aus Testbedingungen. Ein Testfall ist ein Satz von Eingaben, Vorbedingungen, Aktionen (falls anwendbar), erwarteten Ergebnissen und Nachbedingungen, der für ein bestimmtes Testziel oder eine bestimmte Testbedingung entwickelt wird. Eine Testcharta ist eine Form des Testentwurfs, die festlegt, was getestet werden soll, wie getestet werden soll und wonach beim explorativen Testen gesucht werden soll (siehe [CTFL 4.0]). Bei dem obigen Testmittel handelt es sich um eine Testcharta. Testchartas sind das Ergebnis der Testentwicklung (siehe [CTFL 4.0], Abschnitt 1.4.3). Die Test Charta #04.018 beschreibt den Testfall, bei dem die Registrierungsseite mit verschiedenen fehlerhaften Eingabesätzen erforscht wird, um Fehler beim Registrierungsverfahren bei fehlerhafter Eingabe zu entdecken. Dieser Schritt ist entscheidend, um sicherzustellen, dass das Testobjekt korrekt funktioniert und alle relevanten Fehlerzustände abgedeckt sind.

Folglich ist d) KORREKT.

Frage 5	FL-1.4.2	K2	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aspekte beeinflusst AM EHESTEN die Durchführung von Tests für ein bestimmtes Testobjekt?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Die durchschnittliche Erfahrung des Marketingteams des Unternehmens	<input type="checkbox"/>
b)	Das Wissen der Benutzer, dass ein neues System für sie entwickelt wird	<input type="checkbox"/>
c)	Die Anzahl der Jahre an Testerfahrung der Mitglieder des Testteams	<input checked="" type="checkbox"/>
d)	Die Organisationsstruktur der Benutzer der zu entwickelnden Anwendung	<input type="checkbox"/>

FL-1.4.2 (K2) Der Lernende kann die Auswirkungen des Kontexts auf den Testprozess erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 1.4.2)

- a) FALSCH – Es ist sehr unwahrscheinlich, dass das durchschnittliche Erfahrungsniveau des Marketingteams (von dem die meisten im Marketing tätig sind) einen Einfluss darauf hat, wie die Tests für ein bestimmtes Testobjekt durchgeführt werden. Ebenso ist es unwahrscheinlich, dass das Team an der Testdurchführung beteiligt ist, wenn es sich um einen Abnahmetest handelt. Eine Einflussnahme des Marketings kann durch spezielle Anforderungen aus den Marktbedürfnissen erfolgen und somit ggfs. Tests priorisieren, jedoch ohne konkreten Einfluss auf eine Testdurchführung (siehe [CTFL 4.0], Abschnitt 1.4.2, 3. Aufzählungspunkt).
- b) FALSCH – Es ist nicht ableitbar, dass das Wissen der Benutzer darüber, dass ein neues System für sie entwickelt wird, einen Einfluss auf die Testdurchführung hat. Ein Einfluss der Benutzer kann durch spezielle Bedürfnisse oder Erwartungen Einfluss auf Anforderungen haben und somit ggfs. Tests priorisieren, aber ohne konkrete Auswirkung auf eine Testdurchführung (siehe [CTFL 4.0], Abschnitt 1.4.2, 1. Aufzählungspunkt)
- c) KORREKT – Die Anzahl der Jahre an Testerfahrung der Mitglieder des Testteams trägt dazu bei, die Fähigkeiten und das Wissen (z. B. über verschiedene Tools und Fehlertypen) zu bestimmen, die die Teammitglieder beim Testen anwenden und somit nutzen werden (siehe [CTFL 4.0], Abschnitt 1.4.2, 2. Aufzählungspunkt).
- d) FALSCH – Die Organisationsstruktur der Benutzer des zu entwickelnden Systems ist eher unbekannt und wird sich normalerweise nicht auf die Testdurchführung auswirken. Falls eine Zielorganisation bekannt ist, kann das Auswirkungen auf ein Rollenmodell und somit auf die Anforderungen haben, aber nicht direkt auf die Testdurchführung. Eine Organisation, die Einfluss hat, ist die Organisation des Entwicklungs- und Testteams. Das ist aber nicht Inhalt der Option (siehe [CTFL 4.0], Abschnitt 1.4.2, 6. Aufzählungspunkt).

Frage 6	FL-1.4.4	K2	Punkte 1.0
---------	----------	----	------------

**Beachten Sie die folgenden Testaktivitäten:**

- 1. Auswählen von Regressionstests**
- 2. Bewertung der Vollständigkeit der Testdurchführung**
- 3. Identifizieren, welche User Stories offene Fehlerberichte haben**
- 4. Bewertung, ob die Anzahl der Tests für jede Anforderung mit dem Grad des Produktrisikos vereinbar ist**

**Betrachten Sie die folgenden Möglichkeiten, wie die Verfolgbarkeit beim Testen helfen kann:**

- A. Verbessern der Verständlichkeit von Teststatusberichten, um den Status der Elemente der Testbasis einzubeziehen**
- B. Testaktivitäten nachvollziehbarer/prüfbarer machen**
- C. Bereitstellung von Informationen zur Beurteilung der Prozessqualität**
- D. Analysieren der Auswirkungen von Änderungen**

**Welche der folgenden Aussagen passt am besten zur aufgeführten Testaktivität und wie kann "die Verfolgbarkeit" diese Aktivität jeweils unterstützen?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	1D, 2B, 3C, 4A	<input type="checkbox"/>
b)	1B, 2D, 3A, 4C	<input type="checkbox"/>
c)	1D, 2C, 3A, 4B	<input type="checkbox"/>
d)	1D, 2B, 3A, 4C	<input checked="" type="checkbox"/>

FL-1.4.4 (K2) Der Lernende kann die Bedeutung der Pflege der Verfolgbarkeit erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

Verfolgbarkeit (engl. traceability) hilft bei:

- Auswahl von Regressionstests im Hinblick auf die Analyse der Auswirkungen von Änderungen (1D)
- Bewertung der Vollständigkeit der Testausführung, die das Testen nachvollziehbarer macht (2B)
- Identifizieren, welche User-Stories offene Fehlerberichte haben, was die Verständlichkeit von Teststatusberichten verbessert, um den Status von Testbasis-Elementen (3A) einzubeziehen
- Bewertung, ob die Anzahl der Tests für jede Anforderung mit dem Grad des Produktrisikos übereinstimmt, was Informationen zur Bewertung der Testprozessqualität liefert, d. h. Ausrichtung des Testaufwands am Risiko (4C)

Daher ist die Option d) **KORREKT**.

Frage 7	FL-1.5.1	K2	Punkte 1.0
---------	----------	----	------------

Sie sind Teil eines Testteams, das an der Entwicklung eines Steuerungssystems eines Hubschraubers beteiligt ist. Kürzlich wurde ein erfahrener Hubschrauberpilot als Tester für das Testteam eingestellt. Welche positiven Auswirkungen werden die allgemeinen Kompetenzen des neuen Testers vermutlich AM EHSTEN auf das Testteam haben?

Wählen Sie EINE Option! (1 aus 4)

a)	Die Anwendung von 3-Wert-Grenzwertanalyse für einen gründlicheren Testentwurf im Systemtest.	<input type="checkbox"/>
b)	Inkonsistenzen und Ungenauigkeiten in den fachlichen Anforderungen werden effektiv aufgedeckt.	<input checked="" type="checkbox"/>
c)	Einsatz eines Werkzeugs zur Automatisierung von Zustandsübergangstests.	<input type="checkbox"/>
d)	Testergebnisse werden konstruktiver und defensiver an die Entwickler kommuniziert.	<input type="checkbox"/>

FL-1.5.1 (K2) Der Lernende kann Beispiele für die allgemeinen Kompetenzen geben, die für das Testen erforderlich sind [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 1.5.1)

- a) FALSCH – Die Fähigkeit zur Anwendung von Testverfahren wie die 3-Wert-Grenzwertanalyse zählt zur Kategorie „Testwissen“ (siehe 1. Aufzählungspunkt). Es geht aus der Frage nicht hervor, dass der neue Tester auch Testwissen mitbringt.
- b) KORREKT – Das Wissen in der Anwendungsdomäne, das der neue Tester gemäß der Frage auf jeden Fall mitbringt, ist sicherlich hilfreich, wenn es um die effektive Aufdeckung von Inkonsistenzen und Ungenauigkeiten geht. Dieses Wissen fällt in die Kategorie „Wissen in der Anwendungsdomäne“ (siehe 6. Aufzählungspunkt). Ein Tester mit Erfahrung als Pilot ist effektiver im Review der fachlichen Anforderungen.
- c) FALSCH – Die Fähigkeit ein Werkzeug für den automatisierten, zustandsbasierten Test einzusetzen zählt zur Kategorie „Testwissen“ (siehe 1. Aufzählungspunkt). Es geht aus der Frage nicht hervor, dass der neue Tester auch Testwissen mitbringt.
- d) FALSCH – Eine konstruktivere und defensivere Kommunikation von Testergebnissen zählt zur Kategorie „Gute Kommunikationsfähigkeit“ (siehe 3. Aufzählungspunkt). Es geht aus der Frage nicht hervor, dass der neue Tester über gute Kommunikationsfähigkeiten verfügt.

<b>Frage 8</b>	<b>FL-1.5.2</b>	<b>K1</b>	<b>Punkte 1.0</b>
----------------	-----------------	-----------	-------------------

**Welche der folgenden Aussagen beschreibt einen Vorteil des Whole-Team-Ansatzes (Whole Team Approach)?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Er ermöglicht es den Teammitgliedern, jederzeit jede Rolle zu übernehmen.	<input type="checkbox"/>
b)	Es wird nur ein Team benötigt, um das gesamte Entwicklungsprojekt zu unterstützen.	<input type="checkbox"/>
c)	Tester können isoliert arbeiten, ohne die Entwickler oder die Fachbereichsvertreter mit testspezifischen Informationen abzulenken.	<input type="checkbox"/>
d)	Oft ergeben sich Synergien, von denen das gesamte Team und das Projekt profitieren.	<input checked="" type="checkbox"/>

FL-1.5.2 (K1) Der Lernende kann die Vorteile des Whole-Team-Ansatzes wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 1.5.2)

- a) FALSCH – Der Whole-Team-Ansatz basiert auf der Idee der interdisziplinären Teams (siehe [CTFL 4.0], Abschnitt 1.5.2, 2. Absatz), d. h., ist ein Teammitglied befähigt für eine Aufgabe, kann er diese übernehmen. Dies bedeutet jedoch nicht, dass ein Teammitglied zu jeder Zeit jede Rolle übernehmen kann. In der Regel übernimmt eine Person nur die Rolle bzw. die Aufgaben der Rolle, für die sie kompetent ist. Eine Aufgabe auszuführen, für die man nicht geeignet ist, hilft nicht dem Team und dem Projekt.
- b) FALSCH – Der Whole-Team-Ansatz bezieht sich auf die Arbeitsweise eines einzelnen Teams (typischerweise in der agilen Softwareentwicklung) (siehe [CTFL 4.0], Abschnitt 1.5.2, 1. und 2. Absatz). Der Ansatz impliziert nicht, dass nur ein Whole-Team für ein komplettes Projekt benötigt wird.
- c) FALSCH – Der Whole-Team-Ansatz basiert auf der Idee, Synergien zwischen den Teammitgliedern zu schaffen, indem wichtige Informationen zwischen den Mitgliedern geteilt werden. Dies fördert die Zusammenarbeit und schafft ein gemeinsames Verständnis. Tester sollten daher unbedingt testspezifische Informationen mit den Teammitgliedern teilen (siehe [CTFL 4.0], Abschnitt 1.5.2, 3. Absatz, letzter Satz).
- d) **KORREKT** – Durch die optimale Nutzung der unterschiedlichen Fähigkeiten der einzelnen Teammitglieder fördert Whole-Team-Approach die Teamdynamik, eine gute Kommunikation und Zusammenarbeit und erzeugt eine Teamsynergie, von der das gesamte Projekt profitiert (siehe [CTFL 4.0], Abschnitt 1.5.2, 2. Absatz).

Frage 9	FL-2.1.1	K2	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen über die Wahl des Softwareentwicklungslebenszyklus und dessen Beziehung zum Testen ist KORREKT?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	In der agilen Softwareentwicklung ersetzt die Automatisierung von Systemtests die Notwendigkeit von Regressionstests.	<input type="checkbox"/>
b)	Wenn ein sequentielles Entwicklungsmodell verwendet wird, wird das dynamische Testen typischerweise in einer späteren Phase des Lebenszyklus durchgeführt.	<input checked="" type="checkbox"/>
c)	Wenn ein iteratives Entwicklungsmodell verwendet wird, werden Komponententests typischerweise manuell von Entwicklern durchgeführt.	<input type="checkbox"/>
d)	Wenn ein inkrementelles Entwicklungsmodell verwendet wird, werden statische Tests in frühen Inkrementen und dynamische Tests in späteren Inkrementen durchgeführt.	<input type="checkbox"/>



FL-2.1.1 (K2) Der Lernende kann die Auswirkungen des gewählten Softwareentwicklungslebenszyklus auf das Testen erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 2.1.1)

- a) FALSCH – Bei der agilen (iterativ-inkrementellen) Softwareentwicklung werden in jeder Iteration lauffähige Ergebnisse erzeugt und können ausgeliefert werden. Die häufige Bereitstellung von Inkrementen erfordert somit auch ausreichende Regressionstests, möglichst auf allen eingesetzten Teststufen. Eine umfassende Testautomatisierung der Regressionstests erleichtert das Vorgehen. Der System- und Abnahmetest von neuen Features erfolgt eher erfahrungsbasiert und manuell (siehe [CTFL 4.0], Abschnitt 2.1.1, 3. und 4. Absatz). Daher ist nur ein automatisierter Systemtest nicht die korrekte Umsetzung des agilen Softwareentwicklungsmodells in Bezug auf das Testen.
- b) KORREKT – Wenn ein sequenzielles Entwicklungsmodell verwendet wird, steht in den frühen Phasen des Lebenszyklus kein ausführbarer Code zur Verfügung, so dass in dieser Phase statische Tests (z. B. Verifikationen) durchgeführt werden. Später im Lebenszyklus, wenn Code zur Ausführung verfügbar ist, sind die Durchführungen dynamischer Tests möglich (siehe [CTFL 4.0], Abschnitt 2.1.1, 2. Absatz). Es ist jedoch zu beachten, dass die Vorbereitung für dynamische Tests früh im Lebenszyklus der Softwareentwicklung Sinn macht (3.Grundsatz des Testens).
- c) FALSCH – Wenn ein iteratives Entwicklungsmodell verwendet wird, können Komponententests durchaus in jeder Iteration verwendet werden (siehe [CTFL 4.0], Abschnitt 2.1.1, 3. Absatz). Ob die Komponententests dann eher manuell oder automatisiert oder gemischt (neue plus Regressionstests) sind und von Entwicklern oder Testern durchgeführt werden, ist eine projektspezifische Entscheidung (Kosten-Nutzen, Effizienz). Das ist nicht der Einfluss des Softwareentwicklungsmodells.
- d) FALSCH – In den meisten inkrementellen Entwicklungsmodellen werden lauffähige und auslieferfähige Ergebnisse in jedem Inkrement erstellt, so dass sowohl statische als auch dynamische Tests auf allen Teststufen für ein erstelltes Inkrement sinnvoll sind (siehe [CTFL 4.0], Abschnitt 2.1.1, 3. Absatz) Die Verteilung von statischen Tests auf frühe Inkremente und dynamische Tests auf späte Inkremente ist somit nicht korrekt.

Frage 10	FL-2.1.2	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Optionen ist eine gute Testpraktik, die unabhängig vom gewählten Modell des Softwareentwicklungslebenszyklus?**

**Wählen Sie die BESTE Option! (1 aus 4)**

a)	Tester sollten die Arbeitsergebnisse einer Entwicklungsphase in der nächsten Entwicklungsphase überprüfen	<input type="checkbox"/>
b)	Tester sollten mit der Überprüfung der Arbeitsergebnisse einer Softwareentwicklungsaktivität beginnen, sobald Entwürfe verfügbar sind	<input checked="" type="checkbox"/>
c)	Tester sollten Arbeitsergebnisse einer Softwareentwicklungsaktivität nur im Rahmen der Testanalyse und des Testentwurfs überprüfen	<input type="checkbox"/>
d)	Tester sollten Arbeitsergebnisse prüfen, sobald sie zur Nutzung freigegeben sind	<input type="checkbox"/>

FL-2.1.2 (K1) Der Lernende kann gute Praktiken für das Testen, die für alle Lebenszyklen der Softwareentwicklung gelten, wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 2.1.2)

- a) FALSCH – Tester sollten beginnen Arbeitsergebnisse zu überprüfen, sobald Entwürfe verfügbar sind, um frühzeitiges Testen im Rahmen eines Shift-Left-Ansatzes bzw. des 3. Grundsatz des Testens unabhängig vom gewählten SDLC zu ermöglichen (siehe [CTFL 4.0], Abschnitt 2.1.2, 4. Aufzählungspunkt, sowie Abschnitte 2.1.5 und 1.3). Würde bis zu einer nächsten Entwicklungsphase gewartet, könnte das zu unnötigen Fehlerzuständen in abgeleiteten Ergebnissen führen und somit zu unnötiger Entwicklungs- (und Test-) Arbeit (siehe [CTFL 4.0], Abschnitt 1.3).
- b) KORREKT – Tester sollten beginnen Arbeitsergebnisse zu überprüfen, sobald Entwürfe verfügbar sind, um frühzeitiges Testen im Rahmen eines Shift-Left-Ansatzes unabhängig vom gewählten SDLC zu ermöglichen und unnötige Folgefehler zu vermeiden (siehe [CTFL 4.0], Abschnitt 2.1.2, 4. Aufzählungspunkt, sowie Abschnitte 2.1.5 und 1.3).
- c) FALSCH – Die Testanalyse analysiert die Testbasis, um testbare Merkmale zu identifizieren und die zugehörigen Testbedingungen zu bestimmen. Der Testentwurf umfasst dann die Ausarbeitung der Testbedingungen zu Testfällen (siehe [CTFL 4.0], Abschnitt 1.4.1, 4. und 5. Absatz). Die Überprüfung eines Entwicklungsergebnisses, unabhängig vom SDLC, dient der Qualitätssteuerung und sorgt für die Vermeidung von Folgefehlern (siehe [CTFL 4.0], Abschnitt 2.1.2, 1. Aufzählungspunkt) und sollte somit VOR der Testaktivität der Testanalyse durchgeführt sein. Das gilt unabhängig davon, dass Testanalyse und -entwurf in der gleichen Entwicklungsphase beginnen sollten (siehe [CTFL 4.0], Abschnitt 2.1.2, 3. Aufzählungspunkt).
- d) FALSCH – Tester sollten Arbeitsergebnisse prüfen, sobald Entwürfe verfügbar sind, um frühzeitiges Testen im Rahmen eines Shift-Left-Ansatzes unabhängig vom gewählten SDLC zu ermöglichen (siehe [CTFL 4.0], Abschnitt 2.1.2, 4. Aufzählungspunkt, sowie Abschnitt 2.1.5). Das Warten bis zur Freigabe zur Verwendung bedeutet, dass die Mängel, die während der Überprüfung durch den Tester festgestellt werden, im freigegebenen Arbeitsergebnis noch enthalten sind und zu Folgefehlern in darauf basierenden und parallel begonnenen Entwicklungsaktivitäten führen können.

Frage 11	FL-2.1.3	K1	Punkte 1.0
----------	----------	----	------------

**Welches der folgenden Beispiele ist ein Test-First-Ansatz für die Entwicklung?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Testgetriebene Entwicklung	<input checked="" type="checkbox"/>
b)	Überdeckungsgetriebene Entwicklung	<input type="checkbox"/>
c)	Qualitätsgetriebene Entwicklung	<input type="checkbox"/>
d)	Feature-getriebene Entwicklung	<input type="checkbox"/>

FL-2.1.3 (K1) Der Lernende kann die Beispiele für Test-First-Ansätze in der Entwicklung wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 2.3.1)

- a) **KORREKT** – Testgetriebene Entwicklung (TDD: Test Driven Development) ist ein bekanntes Beispiel für einen Test-First-Ansatz in der Entwicklung (siehe [CTFL 4.0], Abschnitt 2.1.1, 2. Absatz).
- b) **FALSCH** – Überdeckungsgetriebene Entwicklung ist kein Beispiel für einen Test-First-Entwicklungsansatz. Das Testen kennt die Überdeckung (Glossar: Der Grad, zu dem bestimmte Überdeckungselemente von einer Testsuite ausgeführt wurden, ausgedrückt in Prozent.) als Ansatz um einen Status zu ermitteln. Dies hat aber keinen Bezug zu dem Zeitpunkt, wann getestet wird.
- c) **FALSCH** – Qualitätsgetriebene Entwicklung ist kein Beispiel für einen Test-First-Ansatz bei der Entwicklung. Jede Entwicklung sollte ein klares Bild haben, welche Eigenschaften (Qualitätsmerkmale) das Zielsystem erfüllen soll. Dies hat aber keinen Bezug zu dem Zeitpunkt, wann getestet wird.
- d) **FALSCH** – Feature-getriebene Entwicklung (FDD: feature-driven development) ist kein Beispiel für einen Test-First-Entwicklungsansatz, sondern eine agile, inkrementelle Softwareentwicklungsmethodik, die auf der Bereitstellung von Features basiert (siehe Glossar).

Frage 12	FL-2.1.4	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen trifft auf DevOps zu?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Um Releases zu beschleunigen wird kontinuierliche Integration eingesetzt, die es den Entwicklern ermöglicht, Code schnell und ohne Komponententests auszuliefern.	<input type="checkbox"/>
b)	Um Systeme schneller zu aktualisieren und freigeben zu können, nutzt die DevOps-Lieferkette Automatisierung, um zeitaufwändige manuelle Regressionstests zu reduzieren.	<input checked="" type="checkbox"/>
c)	Um die Kluft zwischen Entwicklung und Betrieb zu verringern, priorisiert das Testen mit einem Shift-Right-Ansatz den Abnahmetest parallel zur kontinuierlichen Auslieferung.	<input type="checkbox"/>
d)	Um eine größere Synergie zwischen Testern, Entwicklern und Betrieb zu schaffen, werden die Tests vollständig automatisiert, so dass keine manuellen Tests mehr erforderlich sind.	<input type="checkbox"/>

FL-2.1.4 (K2) Der Lernende kann die möglichen Auswirkungen von DevOps auf das Testen zusammenfassen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 2.1.4)

- a) FALSCH – DevOps ermöglicht es den Teams, qualitativ hochwertigen Code über eine DevOps-Auslieferungskette (Delivery Pipeline) schneller zu erstellen, zu testen und freizugeben. Doch es gehört weiterhin dazu, dass Entwickler auch die zugehörigen Komponententests liefern: kontinuierliche Integration (CI) fördert einen Shift-Left-Ansatz beim Testen, indem Entwickler dazu angehalten werden, qualitativ hochwertigen Code zusammen mit Komponententests und statischer Analyse bereitzustellen (siehe [CTFL 4.0], Abschnitt 2.1.4, 1. Absatz und 2. Aufzählungspunkt).
- b) KORREKT – DevOps verbessert das Testen auf verschiedene Weise durch Automatisierung in der Auslieferungskette, die den Bedarf an sich wiederholenden manuellen Tests reduziert; durch Erhöhung des Umfangs und der Bandbreite der automatisierten Regressionstests wird das Risiko einer Regression minimiert (siehe [CTFL 4.0], Abschnitt 2.1.4, 5. und 6. Aufzählungspunkt).
- c) FALSCH – DevOps fördert einen Kulturwandel innerhalb eines Unternehmens, um die Kluft zwischen Entwicklung (einschließlich Testen) und Betrieb zu überbrücken und gleichzeitig ihre jeweilige Aufgabe gleichwertig zu behandeln (siehe [CTFL 4.0], Abschnitt 2.1.4, 1. Absatz). Dazu zählt aber nicht ein Shift-Right-Ansatz, sondern eher ein Shift-Left-Ansatz (frühes Testen) und auch nicht der Abnahmetest parallel zu einer produktiven Auslieferung. Der Abnahmetest bleibt Voraussetzung für eine produktive Freigabe und Auslieferung.
- d) FALSCH – Automatisierte Prozesse wie Continuous Integration/Continuous Delivery (CI/CD) in DevOps ermöglichen stabile Testumgebungen und reduzieren den Bedarf an sich wiederholenden manuellen Tests durch die Automatisierung (z. B. automatisierte Regressionstests) (siehe [CTFL 4.0], Abschnitt 2.1.4, 3., 5. und 6. Aufzählungspunkt). Obwohl DevOps ein hohes Maß an automatisierten Tests mit sich bringt, sind manuelle Tests – insbesondere aus Benutzerperspektive – weiterhin erforderlich (siehe [CTFL 4.0], Abschnitt 2.1.4, letzter Absatz).

Frage 13	FL-2.2.1	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Maßnahmen werden im Rahmen von Systemtests am EHESTEN durchgeführt?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	End-to-End-Tests der IT-Sicherheit eines Kreditmanagementsystems durch ein unabhängiges Testteam.	<input checked="" type="checkbox"/>
b)	Test der Zusammenwirkens eines Geldwechselsystems mit einer externen Bank oder mit dem System einer externen Bank.	<input type="checkbox"/>
c)	Beta-Test einer Lernplattform durch die Trainer eines Schulungsanbieters.	<input type="checkbox"/>
d)	Test der Interaktion zwischen der Benutzeroberfläche und der Datenbank eines Personalverwaltungssystems.	<input type="checkbox"/>

FL-2.2.1 (K2) Der Lernende kann die verschiedenen Teststufen unterscheiden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 2.2.1)

- a) **KORREKT** – Systemtests untersuchen das Verhalten und die Fähigkeiten des gesamten Systems und umfassen nicht-funktionale Tests von Qualitätsmerkmalen, einschließlich Testen der IT-Sicherheit. Der Systemtest wird häufig von einem unabhängigen Testteam auf der Grundlage von Systemspezifikationen durchgeführt (siehe [CTFL 4.0], Abschnitt 2.2.1, 3. Aufzählungspunkt).
- b) **FALSCH** – Diese Option beschreibt eine Maßnahme des Systemintegrationstests. Dieser untersucht die Schnittstellen zu anderen Systemen und externen Diensten (siehe [CTFL 4.0], Abschnitt 2.2.1, 4. Aufzählungspunkt).
- c) **FALSCH** – Beta-Tests können im Rahmen eines Abnahmetests durchgeführt werden (siehe [CTFL 4.0], Abschnitt 2.2.1, 5. Aufzählungspunkt). Der Beta-Test ist eine Art Abnahmetest, der an einem zur Testumgebung des Entwicklers externen Standort durch Akteure außerhalb der Herstellerorganisation durchgeführt wird (siehe Glossar).
- d) **FALSCH** – Diese Option beschreibt eine Maßnahme des Komponentenintegrationstests. Beim Testen der Komponentenintegration werden die Schnittstellen und Interaktionen zwischen den Komponenten eines Systems getestet, z. B. zwischen der Benutzeroberfläche und der Datenbank (siehe [CTFL 4.0], Abschnitt 2.2.1, 2. Aufzählungspunkt).

Frage 14	FL-2.3.1	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Entscheidungen sollte KEIN Auslöser für Wartungstests sein?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Die Entscheidung, die Wartbarkeit der Software zu testen	<input checked="" type="checkbox"/>
b)	Die Entscheidung, das System nach der Migration auf eine neue Betriebsplattform zu testen	<input type="checkbox"/>
c)	Die Entscheidung, die Wiederherstellbarkeit archivierter Daten nach Außerbetriebnahme zu testen	<input type="checkbox"/>
d)	Die Entscheidung zu testen, nachdem ein "Hotfix" auf die Produktivversion aufgespielt wurde	<input type="checkbox"/>

FL-2.3.1 (K2) Der Lernende kann Wartungstest und dessen Auslöser zusammenfassen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

- a) **KORREKT** – Die Wartbarkeit ist ein Qualitätsmerkmal, das nicht von der Ausführung des Codes abhängt (siehe [CTFL 4.0], Abschnitt 3.1.3, 1. Absatz, 5. Aufzählungspunkt). Sie ist daher KEIN Auslöser für Wartungstests, die auf Änderungen im Code oder in der Umgebung als dynamische Tests durchgeführt werden (siehe [CTFL 4.0], Abschnitt 2.3, 3. Absatz inkl. Aufzählungspunkte).
- b) **FALSCH** – Dies ist ein Auslöser für Wartungstests (siehe [CTFL 4.0], Abschnitt 2.3, 3. Absatz, 2. Aufzählungspunkt). Nach einer Plattformmigration ist es wichtig, die Funktionalität und Interoperabilität der Software zu überprüfen.
- c) **FALSCH** – Dies ist ein Auslöser für Wartungstests (siehe [CTFL 4.0], Abschnitt 2.3, 3. Absatz, 3. Aufzählungspunkt). Dies ist relevant, um sicherzustellen, dass archivierte Daten ordnungsgemäß wiederhergestellt werden können.
- d) **FALSCH** – Dies ist ein Auslöser für Wartungstests (siehe [CTFL 4.0], Abschnitt 2.3, 3. Absatz, 1. Aufzählungspunkt). Nach einem Hotfix ist es wichtig, die Auswirkungen auf die bestehende Funktionalität zu überprüfen.



Frage 15	FL-3.1.2	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen beschreibt am BESTEN den Einsatz von statischen Test?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Statisches Testen kann Fehlerzustände aufdecken, die durch dynamisches Testen nicht gefunden werden können.	<input checked="" type="checkbox"/>
b)	Fehlerzustände im Code können durch dynamische Tests effizienter gefunden werden als durch statische Tests.	<input type="checkbox"/>
c)	Der statische Test kann erst in einer späten Phase des SDLC durchgeführt werden.	<input type="checkbox"/>
d)	Um den statischen Test so effizient wie möglich zu gestalten, sollten so wenig Stakeholder wie möglich involviert sein.	<input type="checkbox"/>

FL-3.1.2 (K2) Der Lernende kann den Wert statischer Tests erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

- a) **KORREKT – weil es Fehlerzustände gibt, die nur durch statische Tests erkannt werden (siehe [CTFL 4.0], Abschnitt 3.1.2, 1. Absatz, letzter Satz).**
- b) FALSCH – Statische Tests können Fehlerzustände effizienter, d. h. früher und i. d. R. mit weniger Ressourcen aufdecken als dynamische Tests (siehe [CTFL 4.0], Abschnitt 3.1.2, 4. Absatz, 1. Satz).
- c) FALSCH – Statische Tests können bereits in den frühen Phasen des SDLC eingesetzt werden (siehe [CTFL 4.0], Abschnitt 3.1.2, 2. Absatz, 3. Satz).
- d) FALSCH – Ein Vorteil statischer Tests, insbesondere von Reviews, ist die Verbesserung der Kommunikation und des gegenseitigen Verständnisses zwischen den Stakeholdern. Daher sollte eine große Anzahl von Stakeholdern in statische Tests einbezogen werden (siehe [CTFL 4.0], Abschnitt 3.1.2, 2. Absatz, letzter Satz).

Frage 16	FL-3.2.1	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen beschreibt einen Vorteil von frühem und häufigem Stakeholder-Feedback?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Es ermöglicht dem Projektmanagement, weniger produktive Entwickler frühzeitig zu identifizieren.	<input type="checkbox"/>
b)	Es ermöglicht Projektmanagern, ihre Interaktionen mit Stakeholdern zu reduzieren	<input type="checkbox"/>
c)	Es erleichtert die frühzeitige Kommunikation potenzieller Qualitätsprobleme	<input checked="" type="checkbox"/>
d)	Endbenutzer verstehen besser, warum sich die Bereitstellung der Anwendung verzögert	<input type="checkbox"/>

FL-3.2.1 (K1) Der Lernende kann die Vorteile eines frühen und häufigen Stakeholder-Feedbacks identifizieren [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 3.2.1)

Ein frühzeitiges und häufiges Feedback der Stakeholder dient der Kommunikation, um Missverständnissen über Anforderungen vorzubeugen und um potentielle Probleme aus dem Weg zu räumen (siehe [CTFL 4.0], Abschnitt 3.2.1, 1. und 2. Absatz). Daher:

- a) FALSCH – Es geht nicht um die Bewertung von Personen und ihren Leistungen.
- b) FALSCH – Eine Reduzierung der Interaktionen durch den Projektmanager wird kein Vorteil sein, da dadurch ggfs. wichtiges Feedback verloren geht oder zu spät kommt (siehe [CTFL 4.0], Abschnitt 3.2.1, 1. und 2. Absatz).
- c) **KORREKT – Die frühzeitige und häufige Einholung von Feedback von den am Softwareentwicklungsprozess Beteiligten kann sehr vorteilhaft sein, da sie die frühzeitige Kommunikation potenzieller Qualitätsprobleme erleichtert, Missverständnisse bezüglich der Anforderungen vermeiden hilft und sicherstellt, dass Änderungen der Anforderungen der Beteiligten schneller verstanden und umgesetzt werden (siehe [CTFL 4.0], Abschnitt 3.2.1, 1. und 2. Absatz).**
- d) FALSCH – Es ist ein möglicher Vorteil des Feedbacks und der Kommunikation, Terminprobleme zu vermeiden und nicht sie zu begründen, z. B. durch die Klärung von Anforderungen oder Priorisierung von Wünschen (siehe [CTFL 4.0], Abschnitt 3.2.1, 1. Absatz).

Frage 17	FL-3.2.2	K2	Punkte 1.0
----------	----------	----	------------

Die folgenden Aufgaben des Review-Prozesses werden beschrieben:

1. Die zu bewertenden Qualitätsmerkmale und die Endkriterien werden festgelegt
2. Das Arbeitsergebnis wird allen Beteiligten zugänglich gemacht
3. Anomalien im Arbeitsergebnis werden identifiziert
4. Anomalien werden analysiert und diskutiert

Ordnen Sie diese Aufgabenbeschreibungen den folgenden Review-Aktivitäten zu:

- A. Individuelles Review
- B. Reviewbeginn
- C. Planung
- D. Kommunikation und Analyse

Welche Zuordnung von Aufgabe zu Aktivität ist KORREKT?

Wählen Sie EINE Option! (1 aus 4)

a)	1B, 2C, 3D, 4A	<input type="checkbox"/>
b)	1B, 2D, 3C, 4A	<input type="checkbox"/>
c)	1C, 2A, 3B, 4D	<input type="checkbox"/>
d)	1C, 2B, 3A, 4D	<input checked="" type="checkbox"/>

FL-3.2.2 (K2) Der Lernende kann die Aktivitäten des Reviewprozesses zusammenfassen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 3.2.2)

Unter Berücksichtigung jeder der aufgeführten Aufgabenbeschreibungen:

1. Die zu bewertenden Qualitätsmerkmale und die Endkriterien werden festgelegt = Planung (C): Festlegen des Reviewumfangs, des Zwecks, der zu überprüfenden Arbeitsergebnisses, der **zu bewertenden Qualitätsmerkmale**, die zu berücksichtigenden Bereiche, **der Endkriterien**, der unterstützenden Informationen sowie der Standards, des Aufwands und Zeitrahmen (siehe [CTFL 4.0], Abschnitt 3.2.2, 1. Aufzählungspunkt).
2. Alle Teilnehmer haben Zugang zum Arbeitsergebnis = Reviewbeginn (B): Sicherstellen, dass alle Teilnehmer vorbereitet sind, um mit dem Review zu starten. Dazu gehört auch, dass jeder Teilnehmer **Zugang zu dem zu prüfenden Arbeitsergebnis hat**, seine Rolle und Verantwortlichkeiten versteht und alles erhält, was er für die Durchführung des Reviews benötigt (siehe [CTFL 4.0], Abschnitt 3.2.2, 2. Aufzählungspunkt).
3. Anomalien werden im Arbeitsprodukt identifiziert = Individuelles Review (A): Bewertung der Qualität des Arbeitsprodukts, **Identifizierung** und Protokollierung **von Anomalien**, Empfehlungen und Fragen mit Hilfe von Reviewverfahren wie checklistenbasiertem oder szenariobasiertem Review (siehe [CTFL 4.0], Abschnitt 3.2.2, 3. Aufzählungspunkt).
4. Anomalien werden analysiert und diskutiert = Kommunikation und Analyse (D): Da es sich bei einer festgestellten Anomalie nicht unbedingt um einen Fehlerzustand handelt, **wird jede Anomalie analysiert und besprochen**, ihr Status, ihr Eigentum und die erforderlichen Maßnahmen werden festgelegt und Review-Entscheidungen getroffen, normalerweise in einer Reviewsitzung (siehe [CTFL 4.0], Abschnitt 3.2.2, 4. Aufzählungspunkt).

Daher:

- a) FALSCH
- b) FALSCH
- c) FALSCH

**d) KORREKT – Die korrekte Kombination ist: 1C, 2B, 3A, 4D**

Frage 18	FL-3.2.3	K1	Punkte 1.0
----------	----------	----	------------

Der generische Review-Prozess kennt unter anderem die folgenden Rollen:

1. Protokollant
2. Reviewleiter
3. Moderator
4. Manager

Im Rahmen von Reviews können diese Rollen folgende Verantwortlichkeiten übernehmen:

- A. Sorgt für die effektive Durchführung von Review-Sitzungen, einschließlich einer geschützten Review-Umgebung
- B. Zeichnet Review-Informationen auf, z. B. Entscheidungen und neue Anomalien, die während der Review-Sitzung gefunden wurden
- C. Entscheidet, was geprüft werden soll und stellt Ressourcen wie Personal und Zeit für das Review zur Verfügung
- D. Übernimmt die Gesamtverantwortung für das Review, z. B. die Organisation, wann und wo das Review stattfindet

Welche der folgenden Zuordnungen von Rollen zu Verantwortlichkeiten ist KORREKT?

Wählen Sie EINE Option! (1 aus 4)

a)	1A, 2B, 3D, 4C	<input type="checkbox"/>
b)	1A, 2C, 3B, 4D	<input type="checkbox"/>
c)	1B, 2D, 3A, 4C	<input checked="" type="checkbox"/>
d)	1B, 2D, 3C, 4A	<input type="checkbox"/>

FL-3.2.3 (K1) Der Lernende kann die bei der Durchführung von Reviews den Hauptrollen zugewiesenen Verantwortlichkeiten wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 3.2.3)

Die aufgeführten Rollen übernehmen folgende Verantwortungen:

1. Protokollant – verantwortlich für das Sammeln von Anomalien der Gutachter und die Aufzeichnung von Reviewinformationen, wie z. B. getroffene Entscheidungen und neue Anomalien, die während des Reviews gefunden werden. (B) (siehe [CTFL 4.0], Abschnitt 3.2.3, 4. Aufzählungspunkt)
2. Reviewleiter – übernimmt die Gesamtverantwortung für das Review, z. B. die Entscheidung wer daran teilnimmt, und die Organisation, wann und wo das Review stattfindet. (D) (siehe [CTFL 4.0], Abschnitt 3.2.3, 6. Aufzählungspunkt)
3. Moderator – verantwortlich dafür, dass die Reviewsitzungen effektiv ablaufen, einschließlich Mediation, Zeitmanagement und einer geschützten Reviewumgebung, in der jeder frei sprechen kann. (A) (siehe [CTFL 4.0], Abschnitt 3.2.3, 3. Aufzählungspunkt)
4. Manager – verantwortlich für die Entscheidung, was überprüft werden soll, und für die Bereitstellung von Ressourcen wie Personal und Zeit für das Review. (C) (siehe [CTFL 4.0], Abschnitt 3.2.3, 1. Aufzählungspunkt)

Daher:

a) FALSCH

b) FALSCH

**c) KORREKT – Die korrekte Kombination ist: 1B, 2D, 3A, 4C**

d) FALSCH

Frage 19	FL-4.1.1	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen beschreibt KORREKT die Zuordnung von Verfahrensgrundlagen zu Entscheidungstabellen- und Zweigtests?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Beim Entscheidungstabellentest werden die Testfälle aus den Entscheidungsergebnissen im Code abgeleitet. Beim Zweigtest werden die Testfälle aus der Kenntnis des Kontrollflusses des Testobjekts abgeleitet.	<input type="checkbox"/>
b)	Beim Entscheidungstabellentest werden die Testfälle aus der Spezifikation abgeleitet, die die Geschäftslogik beschreibt. Beim Zweigtest basieren die Testfälle auf der Antizipation potenzieller Fehler im Quellcode.	<input type="checkbox"/>
c)	Beim Entscheidungstabellentest werden die Testfälle aus der Kenntnis des Kontrollflusses des Testobjekts abgeleitet. Beim Zweigtest werden die Testfälle aus der Spezifikation abgeleitet, die die Geschäftslogik beschreibt.	<input type="checkbox"/>
d)	Beim Entscheidungstabellentest werden die Testfälle unabhängig vom Zustand der Software abgeleitet. Beim Zweigtest können die Testfälle erst nach dem Entwurf oder der Implementierung des Codes erstellt werden.	<input checked="" type="checkbox"/>

FL-4.1.1 (K2) Der Lernende kann Black-Box-Testverfahren, White-Box-Testverfahren und erfahrungsbasierte Testverfahren unterscheiden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.1)

Grundlagen:

- **Black-Box-Testverfahren** (auch spezifikationsbasierte Verfahren genannt) basieren auf einer Analyse des spezifizierten Verhaltens des Testobjekts ohne Kenntnis der internen Struktur. Daher werden die Testfälle unabhängig von der Implementierung der Software erstellt.
  - **White-Box-Testverfahren** (auch als strukturbasierte Verfahren bekannt) basieren auf einer Analyse der internen Struktur und Verarbeitung des Testobjekts. Da die Testfälle vom Entwurf der Software abhängig sind, können sie erst nach dem Entwurf oder der Implementierung des Testobjekts erstellt werden.
  - **Erfahrungsbasierte Testverfahren** nutzen das Wissen und die Erfahrung von Testern effektiv für den Entwurf und die Implementierung von Testfällen.
  - Der **Entscheidungstablentest** gehören zu den Black-Box-Testverfahren (siehe [CTFL 4.0], Abschnitt 4.2.3)
  - Der **Zweigtest** gehören zu den White-Box-Testverfahren (siehe [CTFL 4.0], Abschnitt 4.3.2)
- a) FALSCH – Der Entscheidungstablentest ist ein Black-Box-Testverfahren, d. h. spezifikationsbasiert und nicht strukturbasiert – die Testfälle basieren NICHT auf den Entscheidungen im Code. Diese Aussage ist somit nicht korrekt. Beim Zweigtest werden die Testfälle aus der Kenntnis des Kontrollflusses des Testobjekts abgeleitet. Diese Aussage ist grundsätzlich korrekt.
- b) FALSCH – Beim Entscheidungstablentest werden die Testfälle aus der Spezifikation abgeleitet, die die Geschäftslogik beschreibt. Diese Aussage ist grundsätzlich korrekt. Die Antizipation potenzieller Fehler wird bei der intuitiven Testfallermittlung (ein erfahrungsbasiertes Testverfahren) und nicht beim Zweigtest (ein strukturbasiertes Testverfahren) verwendet. Diese Aussage ist somit nicht korrekt.
- c) FALSCH – Basiert ein Testfall auf der Kenntnis des Kontrollflusses des Testobjekts, handelt es sich um ein White-Box-Testverfahren. Das Testen von Entscheidungstabellen basiert in der Regel auf einer Analyse der Geschäftslogik und ist ein Black-Box-Testverfahren. Diese Aussage ist somit nicht korrekt. Beim Zweigtest werden Testfälle nicht aus der Spezifikation abgeleitet – dies würde zu einem Black-Box-Testverfahren gehören. Diese Aussage ist somit nicht korrekt.
- d) **KORREKT – Entscheidungstablentests sind ein Black-Box-Testverfahren und basieren auf einer Analyse des spezifizierten Verhaltens des Testobjekts ohne Bezug auf seine interne Struktur. Die Testfälle sind daher unabhängig von der Implementierung der Software. Der Zweigtest ist ein White-Box-Testverfahren, d. h. die Testfälle basieren auf einer Analyse der internen Struktur und Verarbeitung des Testobjekts. Da die Testfälle davon abhängen, wie die Software entworfen und codiert wird, können sie erst nach dem Entwurf oder der Implementierung des Testobjekts erstellt werden.**



Frage 20	FL-4.2.1	K3	Punkte 1.0
----------	----------	----	------------

Die Kunden der Autowaschanlage TestWash haben Karten, auf denen die Anzahl der bisher gekauften Wäschen gespeichert ist. Der Anfangswert des Zählers ist 0. Bei der Einfahrt in die Waschanlage erhöht das System die Zahl den Zähler auf der Karte um eins. Dieser Wert stellt die Anzahl der aktuellen Wäschen dar. Anhand der Anzahl der Wäschen entscheidet das System, welchen Rabatt der Kunde erhält.

Für jede zehnte Wäsche gewährt das System einen Rabatt von 10 % und für jede zwanzigste Wäsche gewährt das System einen weiteren Rabatt von 40 % (d. h. insgesamt 50 % Rabatt).

Welche der folgenden Eingaben (bezogen auf die als Anzahl der Wäschen) erreicht die höchste Überdeckung der Äquivalenzklassen?

Wählen Sie EINE Option! (1 aus 4)

a)	19, 20, 30	<input checked="" type="checkbox"/>
b)	11, 12, 20	<input type="checkbox"/>
c)	1, 10, 50	<input type="checkbox"/>
d)	10, 29, 30, 31	<input type="checkbox"/>

FL-4.2.1 (K3) Der Lernende kann Äquivalenzklassenbildung zur Ableitung von Testfällen ableiten [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.2.1)

- a) **KORREKT** – 19 deckt die Äquivalenzklasse „Kein Rabatt“, 20 die Äquivalenzklasse „50 % Rabatt“ und 30 die Äquivalenzklasse „10 % Rabatt“ ab. Diese drei Werte decken alle drei gültigen Äquivalenzklassen ab.
- b) **FALSCH** – 11 und 12 decken die Äquivalenzklasse „kein Rabatt“ ab, während 20 die Äquivalenzklasse „50 % Rabatt“ abdeckt. Die Äquivalenzklasse „10 % Rabatt“ ist nicht abgedeckt. Es werden somit nur zwei Äquivalenzklassen abgedeckt
- c) **FALSCH** – 1 deckt die Äquivalenzklasse „kein Rabatt“ ab, während 10 und 50 die Äquivalenzklassen „10 % Rabatt“ abdecken. Die Äquivalenzklasse „50 % Rabatt“ ist nicht abgedeckt. Es werden somit nur zwei Äquivalenzklassen abgedeckt.
- d) **FALSCH** – 29 und 31 decken die Äquivalenzklasse „kein Rabatt“ ab, während 10 und 30 die Äquivalenzklasse „10 % Rabatt“ abdecken. Die Äquivalenzklasse „50 % Rabatt“ ist nicht abgedeckt. Es werden somit nur zwei Äquivalenzklassen abgedeckt.

Frage 21	FL-4.2.2	K3	Punkte 1.0
----------	----------	----	------------

Ein Weinlagerungssystem verwendet ein Kontrollgerät, das die Temperatur (T) des Weinlagers (gemessen in °C, gerundet auf das nächste Grad) misst und den Benutzer alarmiert, wenn die optimale Temperatur über- oder unterschritten wird:

- Wenn  $11 \leq T \leq 13$ , meldet das System: "optimale Temperatur"
- Wenn  $T < 11$  ist, meldet das System: "Die Temperatur ist zu niedrig!"
- Wenn  $T > 13$  ist, meldet das System: "Die Temperatur ist zu hoch!"

Sie verwenden die 3-Wert-Grenzwertanalyse, um das erwartete Verhalten des Reglers zu überprüfen. Die Testeingabe ist eine vom Gerät gelieferte Temperatur in °C.

Welche Testeingaben ergeben eine 100%ige Überdeckung?

Wählen Sie EINE Option! (1 aus 4)

a)	11, 12, 13	<input type="checkbox"/>
b)	9, 13, 15	<input type="checkbox"/>
c)	9, 10, 11, 12, 13, 14, 15	<input checked="" type="checkbox"/>
d)	10, 11, 12, 13, 14	<input type="checkbox"/>

FL-4.2.2 (K3) Der Lernende kann die Grenzwertanalyse zur Ableitung von Testfällen anwenden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0, Abschnitt 4.2.2)

Es gibt drei Äquivalenzklassen: ÄK1: {..., 9, 10}, ÄK2: {11 ... 13}, und ÄK 3: {14, 15 ...}.

Um eine 100%ige Überdeckung bei der 3-Wert-Grenzwertanalyse zu erreichen, müssen wir alle Überdeckungselemente testen, einschließlich der identifizierten Grenzwerte und ihrer Nachbarn (siehe [CTFL 4.0]).

Die Grenzwerte sind 10, 11, 13 und 14 (wobei negative werte sowie werte außerhalb der Wertebereichsgrenzen nicht betrachtet wurden). Bei der 3-Wert-Grenzwertanalyse müssen nun die Grenzwerte sowie ihre beiden Nachbarn verwendet werden:

- Für ÄK1: 9, **10**, 11
- Für ÄK2: 11, **12**, 13 (hier ist die Besonderheit, dass die Werte 11 und 13 sowohl für den oberen als auch unteren Grenzwert der Nachbar ist. Diese Werte müssen daher nur einmal verwendet werden in einem Testfall).
- Für ÄK3 : 13, **14**, 15

a) FALSCH – Das würde der 3-Wert-Grenzwertanalyse für ÄK2 entsprechen

b) FALSCH – Das entspräche einer reinen Äquivalenzklassenbildung, da immer nur ein beliebiger Wert aus der jeweiligen Äquivalenzklasse verwendet wird

c) **KORREKT – Dies entspricht den oben aufgeführten identifizierten Werten für die 3-Wert-Grenzwertanalyse**

d) FALSCH – Das entspricht der 2-Wert-Grenzwertanalyse für alle Äquivalenzklassen

Frage 22	FL-4.2.3	K3	Punkte 1.0
----------	----------	----	------------

Die folgende Entscheidungstabelle enthält die Regeln zur Bestimmung des Risikos für Arteriosklerose (Arterienverkalkung) auf der Grundlage der Cholesterin- und Blutdruckwerte des Patienten.

	Regel 1	Regel 2	Regel 3	Regel 4	Regel 5
<b>Bedingungen</b>					
<b>Cholesterin (mg/dl)</b>	$\leq 124$	$\leq 124$	125 – 200	125 – 200	$\geq 201$
<b>Blutdruck (mmHg)</b>	$\leq 140$	$> 140$	$\leq 140$	$> 140$	–
<b>Aktion</b>					
<b>Risikoniveau</b>	sehr gering	gering	mittel	hoch	sehr hoch

Sie haben die Testfälle mit den folgenden Testdaten entworfen:

TC1: Cholesterin = 125 mg/dl - Blutdruck = 141 mm Hg

TC2: Cholesterin = 200 mg/dl - Blutdruck = 201 mm Hg

TC3: Cholesterin = 124 mg/dl - Blutdruck = 201 mm Hg

TC4: Cholesterin = 109 mg/dl - Blutdruck = 200 mm Hg

TC5: Cholesterin = 201 mg/dl - Blutdruck = 140 mm Hg

Welche Überdeckung der Entscheidungstabelle wird durch diese Testfälle erreicht?

Wählen Sie EINE Option! (1 aus 4)

a)	40 %	<input type="checkbox"/>
b)	60 %	<input checked="" type="checkbox"/>
c)	80 %	<input type="checkbox"/>
d)	100 %	<input type="checkbox"/>

FL-4.2.3 (K3) Der Lernende kann Entscheidungstabellentests zur Ableitung von Testfällen anwenden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.2.3)

Die Entscheidungstabelle besteht aus fünf Spalten. Jeder Testfall deckt eine dieser Spalten ab.

TC1 und TC2 decken beide Regel 4 ab

TC3 und TC4 decken beide Regel 2 ab

TC5 deckt Regel 5 ab

Diese fünf Testfälle decken also drei von fünf Spalten ab und erreichen eine Überdeckung von  $(3/5) \cdot 100 \% = 60 \%$  (siehe [CTFL 4.0], Abschnitt 4.2.3, 5. Absatz).

Folglich ist Option b) die KORREKTE Option.

Daher:

a) FALSCH

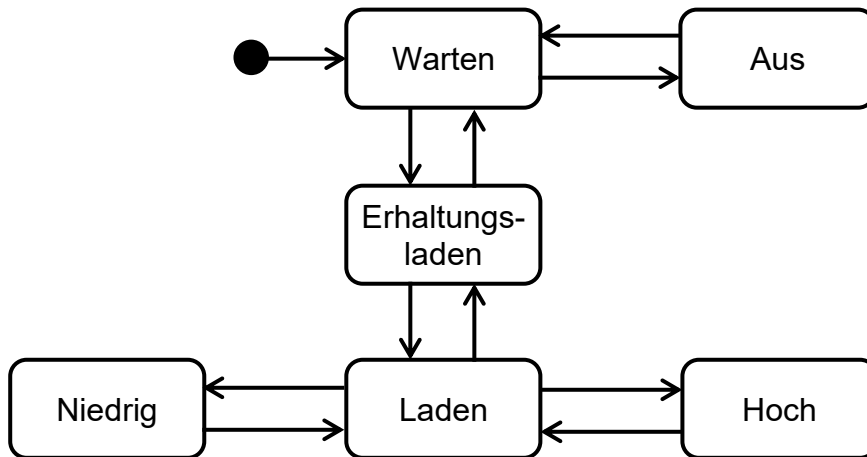
**b) KORREKT**

c) FALSCH

d) FALSCH

Frage 23	FL-4.2.4	K3	Punkte 1.0
----------	----------	----	------------

Gegeben sei folgendes Zustandsübergangsdiagramm für die Software eines Batterieladegerätes:



Welcher der folgenden Testfälle enthält sowohl gültige als auch ungültige Übergänge?

Wählen Sie EINE Option! (1 aus 4)

a)	Start → Warten → Aus → Warten → Erhaltungsladen → Warten	<input type="checkbox"/>
b)	Start → Warten → Erhaltungsladen → Laden → Hoch → Laden	<input type="checkbox"/>
c)	Start → Warten → Erhaltungsladen → Laden → Niedrig → Laden	<input type="checkbox"/>
d)	Start → Warten → Aus → Warten → Laden → Niedrig → Laden	<input checked="" type="checkbox"/>

FL-4.2.4 (K3) Der Lernende kann Zustandsübergangstests zur Ableitung von Testfällen anwenden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

Das Zustandsübergangsdiagramm zeigt 10 gültige Übergänge. Der Übergang Aus → Hoch, welcher von Testfall D ausgeführt wird, ist nicht in dem Zustandsdiagramm enthalten und daher ein ungültiger Übergang (siehe [CTFL 4.0], Abschnitt 4.2.4):

- a) FALSCH – In diesem Testfall werden nur gültige Übergänge ausgeführt, da er den Zustand gemäß dem Diagramm durchläuft.
- b) FALSCH – In diesem Testfall werden nur gültige Übergänge ausgeführt, da er den Zustand gemäß dem Diagramm durchläuft.
- c) FALSCH – In diesem Testfall werden nur gültige Übergänge ausgeführt, da der den Zustand gemäß dem Diagramm durchläuft.
- d) **KORREKT** – In diesem Testfall wird der ungültige Übergang *Aus → Hoch* repräsentiert ausgeführt.

**Daher ist Testfall d) derjenige, der neben gültigen Übergängen auch einen ungültigen Übergang aufweist.**



Frage 24	FL-4.3.1	K2	Punkte 1.0
----------	----------	----	------------

Sie führen zwei Testfälle, T1 und T2, für denselben Code aus. Testfall T1 erreicht eine Anweisungsüberdeckung von 40 % und Testfall T2 erreicht eine Anweisungsüberdeckung von 65 %.

Welche der folgenden Aussagen ist aufgrund der obigen Informationen KORREKT?

Wählen Sie EINE Option! (1 aus 4)

a)	Die Testsuite, bestehend aus den Testfällen T1 und T2, erreicht eine Anweisungsüberdeckung von 105 %.	<input type="checkbox"/>
b)	Mindestens eine Anweisung wurde sowohl von T1 als auch von T2 ausgeführt.	<input checked="" type="checkbox"/>
c)	Mindestens 5 % der Anweisungen im getesteten Code sind nicht ausführbar.	<input type="checkbox"/>
d)	Die aus den Testfällen T1 und T2 bestehende Testsuite erreicht eine Zweigüberdeckung von 100 %.	<input type="checkbox"/>

FL-4.3.1 (K2) Der Lernende kann den Anweisungstest erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.3.1)

- a) FALSCH – Die Überdeckung wird immer als Prozentsatz der gesamten Anzahl ausführbarer Anweisungen im Code definiert. Sie kann daher nicht größer als 100 % sein (siehe [CTFL 4.0], Abschnitt 4.3.1, 1. Absatz).
- b) KORREKT – Wenn die von T1 und T2 ausgeführten Anweisungen disjunkt wären, würde die Überdeckung der Testsuite {T1, T2} 105 % betragen, was unmöglich ist (siehe Antwort a). Daher müssen mindestens 5 % der ausführbaren Anweisungen sowohl von T1 als auch von T2 ausgeführt worden sein.
- c) FALSCH – Die Anweisungsüberdeckung sagt nichts über die Menge der nicht ausführbaren Anweisungen im Code aus, denn sie zeigt nur die bisher ausgeführten Anweisungen an. Ob bisher nicht ausgeführter Code ein nicht erreichbarer und somit nicht ausführbarer Code ist, kann nur durch Analysen ermittelt werden.
- d) FALSCH – Selbst, wenn eine Testsuite eine 100%ige Anweisungsüberdeckung erreicht, bedeutet dies nicht automatisch, dass eine 100%ige Zweigüberdeckung erreicht wurde. Hierfür ist eine Messung der Zweigüberdeckung notwendig (siehe [CTFL 4.0], Abschnitt 4.3.1, 2. Absatz sowie Abschnitt 4.3.2, 4. Absatz).

Frage 25	FL-4.3.2	K2	Punkte 1.0
----------	----------	----	------------

Die Formel für die Zweigüberdeckungsmetrik ist definiert als  $ZÜ = (X / Y) * 100 \%$ .

Was bedeuten X und Y in dieser Formel?

Wählen Sie **EINE** Option! (1 aus 4)

a)	X = Anzahl der von den Testfällen ausgeführten Entscheidungsergebnisse Y = Gesamtzahl der Entscheidungsergebnisse im Code	<input type="checkbox"/>
b)	X = Anzahl der durch die Testfälle ausgeführten bedingten Zweige Y = Gesamtzahl der Zweige im Code	<input type="checkbox"/>
c)	X = Anzahl der von den Testfällen ausgeführten Zweige Y = Gesamtzahl der Zweige im Code	<input checked="" type="checkbox"/>
d)	X = Anzahl der von den Testfällen ausgeführten bedingten Zweige Y = Gesamtzahl der Entscheidungsergebnisse im Code	<input type="checkbox"/>

FL-4.3.2 (K2) Der Lernende kann den Zweigttest erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.3.2)

Der Zweigttest ist eine White-Box-Testverfahren, bei dem die Überdeckungselemente Zweige sind. Ein Zweig ist ein Kontrollübergang zwischen zwei Knoten im Kontrollflussdiagramm, das die möglichen Sequenzen aufzeigt, in denen Quellcodeanweisungen im Testobjekt ausgeführt werden. Jede Kontrollübergang kann entweder bedingungslos (d. h. geradliniger Code) oder bedingt (d. h. ein Entscheidungsergebnis) sein. Die Messgröße der Zweigüberdeckung erfolgt als Anzahl der durch die Testfälle ausgeführten Zweige dividiert durch die Gesamtzahl der Zweige im Code und wird in Prozent ausgedrückt.

Daher:

- a) FALSCH – Ein Entscheidungsergebnis ist ein bedingter Zweig. Beim Zweigttest zählt X nicht nur bedingte, sondern auch unbedingte Zweige. Daher ist die Aussage zu X nicht korrekt, es fehlen die unbedingten Zweige. Analog gilt dies auch für die Aussage zu Y.
- b) FALSCH – Die Zweigttest umfasst nicht nur bedingte, sondern auch unbedingte Zweige (siehe a).
- c) **KORREKT – Die Zweigüberdeckung wird gemessen als die Anzahl der Zweige, die von den Testfällen mindestens einmal ausgeführt wurden (X), dividiert durch die Gesamtanzahl aller Zweige im Code (Y) und als Prozentsatz ausgedrückt ( $ZÜ = X / Y * 100 \%$ ).**
- d) FALSCH – Sowohl X als auch Y zählen nur die bedingten Zweige und berücksichtigen nicht die unbedingten Zweige (siehe a).

Frage 26	FL-4.4.2	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen liefert die BESTE Begründung für den effektiven Einsatz explorativer Tests?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Die bestehende Teststrategie fordert, dass die Tester Black-Box-Testverfahren verwenden.	<input type="checkbox"/>
b)	Die Spezifikation ist in einer formalen Sprache geschrieben, die von einem Werkzeug verarbeitet werden kann.	<input type="checkbox"/>
c)	Die Tester sind Mitglieder eines agilen Teams und verfügen über gute Programmierkenntnisse.	<input type="checkbox"/>
d)	Die Tester haben Erfahrung in der Anwendungsdomäne und gute analytische Fähigkeiten.	<input checked="" type="checkbox"/>

FL-4.4.2 (K2) Der Lernende kann den explorativen Test erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.4.2)

Explorative Tests sind dann sinnvoll, wenn es nur wenige oder unzureichende Spezifikationen gibt oder wenn die Tests unter erheblichem Zeitdruck durchgeführt werden müssen. Explorative Tests eignet sich auch als Ergänzung zu anderen, eher formalen Testverfahren. Explorativer Test ist effektiver, wenn der Tester über Erfahrungen und Fach- bzw. Anwendungskennnissen verfügt und ein hohes Maß an analytischen Fähigkeiten, Neugier und Kreativität besitzt.

Daher:

- a) FALSCH – Exploratives Testen ist kein Black-Box-Testverfahren, daher hat die Aussage keinen Bezug zu einer Begründung von einem explorativen Test.
- b) FALSCH – Explorative Tests sind nützlich, wenn die Spezifikationen unvollständig oder unzureichend sind. Eine formal beschriebene und durch Werkzeuge verarbeitbare Spezifikation begründet eher den Einsatz einer statischen Analyse und automatisierten Tests und nicht den Einsatz explorativer Tests.
- c) FALSCH – Agile Teams und agile Entwicklung kann sinnvoll einen explorativen Test für neue Features einsetzen, jedoch haben Programmierkenntnisse nichts mit explorativem Testen zu tun bzw. unterstützen nicht die effiziente Durchführung. Daher ist dies keine Begründung.
- d) **KORREKT – Explorative Tests ist effektiver, wenn die Tester Erfahrungen haben sowie über Anwendungskennnisse und ein hohes Maß an analytischen Fähigkeiten, Neugier und Kreativität verfügen.**

Frage 27	FL-4.4.3	K2	Punkte 1.0
----------	----------	----	------------

**Welches ist das BESTE Beispiel für eine Testbedingung bei der Verwendung von checklistenbasierten Tests?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	„Der Entwickler hat bei der Implementierung des Codes eine Fehlhandlung gemacht.“	<input type="checkbox"/>
b)	„Die erreichte Anweisungsüberdeckung ist größer als 85 %“	<input type="checkbox"/>
c)	„Das Programm erfüllt die funktionalen und nicht-funktionalen Anforderungen korrekt“	<input type="checkbox"/>
d)	„Die Fehlermeldungen des Systems sind für die Benutzer verständlich.“	<input checked="" type="checkbox"/>

FL-4.4.3 (K2) Der Lernende kann den checklistenbasierten Test erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.4.3)

- a) FALSCH – Checklisten sollten zu prüfende Testbedingungen enthalten und keine allgemeinen Vermutungen. Das Entwickler Fehlhandlungen begehen, kann eine Erfahrung sein. Um daraus ein oder mehrere Elemente als Testbedingungen für eine Checkliste zu machen, sollte diese Erfahrung konkretisiert werden: welche Fehlhandlungen passieren öfters und welche Fehlerzustände bzw. welche Fehlerwirkungen können konkret daraus als Testbedingungen abgeleitet werden (siehe [CTFL 4.0], Abschnitt 4.4.3, 1. Absatz).
- b) FALSCH – Checklisten sollten keine Elemente enthalten, die besser als Endekriterien geeignet sind. Dies ist ein Beispiel für ein Endekriterium (siehe [CTFL 4.0], Abschnitt 4.4.3, 1. Absatz).
- c) FALSCH – Checklisten sollten zu prüfende Testbedingungen enthalten. Bei dieser Option handelt es sich um eine sehr allgemeine Forderung, die praktisch das grundsätzliche Ziel des Testens beschreibt und keine konkrete Testbedingung. Es sollte möglich sein, jedes Element einzeln und direkt zu prüfen (siehe [CTFL 4.0], Abschnitt 4.4.3, 2. Absatz).
- d) **KORREKT – Dies ist ein Beispiel für eine Testbedingung, die von einem Tester auf Basis seines Wissens und seiner Erfahrung, was dem Benutzer wichtig ist, überprüft werden kann (siehe [CTFL 4.0], Abschnitt 4.4.3, 1. Absatz).**

Frage 28	FL-4.5.2	K2	Punkte 1.0
----------	----------	----	------------

**Betrachten Sie das folgende Abnahmekriterium (Akzeptanzkriterium) für eine User Story, das aus der Perspektive eines Online-Shop-Inhabers geschrieben wurde.**

Angenommen, der Benutzer ist eingeloggt ist befindet sich auf der Startseite:

**Wenn** der Benutzer auf die Schaltfläche „Artikel hinzufügen“ klickt,

**Dann** sollte das Formular „Artikel anlegen“ erscheinen,

**Und** der Benutzer sollte in der Lage sein, einen Namen und einen Preis für den neuen Artikel einzugeben.

**In welchem Format ist dieses Abnahmekriterium geschrieben?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Regelorientiert	<input type="checkbox"/>
b)	Szenario-orientiert	<input checked="" type="checkbox"/>
c)	Produktorientiert	<input type="checkbox"/>
d)	Prozessorientiert	<input type="checkbox"/>

FL-4.5.2 (K2) Der Lernende kann die verschiedenen Möglichkeiten zum Schreiben von Abnahmekriterien einordnen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 4.5.2)

- a) FALSCH – Das regelorientierte Format umfasst z. B. Verifizierungslisten mit Aufzählungspunkten oder tabellarische Formen von Eingabe-Ausgabe-Zuordnungen (Input-Output-Zuordnung), in denen die zu befolgenden Regeln angezeigt werden (siehe [CTFL 4.0], Abschnitt 4.5.2, 3. Absatz, 2. Aufzählungspunkt). Gegeben/Wenn/Dann ist ein szenario-orientiertes Format, das ein zu verifizierendes Szenario beschreibt.
- b) KORREKT – Das Abnahmekriterium nutzt das Gegeben/Wenn/Dann-Format, das szenario-orientiert ist (siehe [CTFL 4.0], Abschnitt 4.5.2, 3. Absatz, 1. Aufzählungspunkt).**
- c) FALSCH – Es gibt kein „produktorientiertes“ Format für Abnahmekriterien, das allgemeingültig definiert ist.
- d) FALSCH – Es gibt kein „prozessorientiertes“ Format für Abnahmekriterien, das allgemeingültig definiert ist.

Frage 29	FL-4.5.3	K3	Punkte 1.0
----------	----------	----	------------

Bitte betrachten Sie die folgende User Story:

**„Als Trainer einer Fußballmannschaft möchte ich die Spielberechtigungsliste im DFBNET für einen Spieltag abrufen können, damit ich den Kader für den nächsten Spieltag zusammenstellen kann.“**

**Welcher Testfall eignet sich am BESTEN für eine Abnahmetest-getriebene Entwicklung der User Story?**

**Wählen Sie eine Option! (1 aus 4)**

a)	Login als Trainer ins DFBNET; wähle die nächsten Spieltage für meine Mannschaft aus; lade die Spielberechtigungslisten.	<input type="checkbox"/>
b)	GEGEBEN: Ich bin als Trainer im DFBNET mit meiner Trainerkennung angemeldet UND GEGEBEN: Ich habe den nächsten Spieltag ausgewählt, WENN ich „Spielberechtigungsliste laden“ auswähle, DANN wird mir eine Liste der spielberechtigten Spieler für den nächsten Spieltag angezeigt	<input checked="" type="checkbox"/>
c)	Login als Mannschaftenverantwortlicher; wähle den nächsten Spieltag aus; lade Spielberechtigungsliste; entferne Spieler die nicht spielberechtigt sind.	<input type="checkbox"/>
d)	GEGEBEN: Ich habe die nächsten Spieltage für meine Mannschaft ausgewählt WENN ich einen Spieltag auswähle UND WENN ich die Spielberechtigungsliste für diesen Spieltag lade, DANN sollen mir alle spieleberechtigten Spieler für diesen Spieltag angezeigt werden.	<input type="checkbox"/>



FL-4.5.3 (K3) Der Lernende kann die abnahmetestgetriebene Entwicklung (ATDD) zur Ableitung von Testfällen anwenden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

Begründung der Testfälle

- a) FALSCH – weil NICHT GEEIGNET – da die User Story nicht korrekt wiedergeben wird. In diesem Testfall wird eine Liste von Spieltagen abgerufen und für alle diese Spieltage werden die Spielberechtigungslisten geladen. Dies wird aber in der User Story nicht gefordert bzw. spezifiziert. Dort soll nur die Spielberechtigungsliste für einen Spieltag geladen werden.
- b) KORREKT – weil am besten GEEIGNET – bei diesem Testfall wird nach dem Login als Trainer die Spielberechtigungsliste für den nächsten Spieltag geladen und angezeigt. Dies entspricht der Spezifikation der User Story. Diese Option ist am besten geeignet, da sie die spezifischen Schritte für die Abnahme der Funktionalität klar definiert und die Akzeptanzkriterien direkt in den Testfall integriert.**
- c) FALSCH – weil NICHT GEEIGNET – Die User Story spezifiziert eindeutig, dass der Trainer die Spielberechtigungsliste für einen Spieltag einsehen möchte, um seine Mannschaft zusammenzustellen. Von einem Mannschftsverantwortlichen ist nicht die Rede.
- d) FALSCH – weil NICHT GEEIGNET – Dieser Testfall prüft zwar die in der User Story spezifizierte Funktionalität in angemessener Weise, berücksichtigt aber nicht, dass der Benutzer als Trainer am DFBNET angemeldet ist. Da die geforderte Funktionalität explizit auf die Rolle des Trainers zugeschnitten ist, ist der Testfall nicht am besten als Abnahmetest der User Story geeignet.

Frage 30	FL-5.1.3	K2	Punkte 1.0
----------	----------	----	------------

Ihr Team folgt einem Prozess, der eine kontinuierliche Integrations- und Auslieferungspipeline (CI/CD) mit einem Shift-Left-Ansatz verwendet. Die ersten drei Schritte dieses Prozesses sind:

- (1) Entwicklung und Bereitstellung des Codes
- (2) Übergabe des Codes an das Versionskontrollsystem und Integration des Codes in den Branch „Test“.
- (3) Durchführen des automatisierten Komponententests des übergebenen Codes

Welches der folgenden Kriterien eignet sich am BESTEN als Eingangskriterium für Schritt (2) dieser Pipeline?

Wählen Sie EINE Option! (1 aus 4)

a)	Die statische Analyse meldet für den übermittelten Code keine Fehler und keine Warnungen mit hohem Schweregrad.	<input checked="" type="checkbox"/>
b)	Die Versionskontrolle meldet keine Konflikte beim Kompilieren und Integrieren des Codes in den „Test“-Zweig.	<input type="checkbox"/>
c)	Die Komponententests sind kompiliert und stehen lauffähig für den „Test“-Zweig zur Verfügung.	<input type="checkbox"/>
d)	Die Anweisungsüberdeckung des Komponententests beträgt mindestens 80 %.	<input type="checkbox"/>

FL-5.1.3 (K2) Der Lernende kann Eingangskriterien und Endekriterien vergleichen und gegenüberstellen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.1.3)

- a) **KORREKT** – Der CI-CD-Ansatz zusammen mit dem Shift-Left-Ansatz fördert das frühe Testen, so dass Entwickler angehalten sind, ihren hochwertigen Code zusammen mit den Komponententests und der statischen Analyse bereitzustellen (siehe [CTFL 4.0], Abschnitt 2.1.4 und 2.1.5). Daher sind die Ergebnisse der statischen Analyse ein sinnvolles und messbares Eingangskriterium vor der Integration des Codes in die CI-CD-Pipeline, vergleichbar einem Smoke-Test (siehe [CTFL 4.0], Abschnitt 5.1.3, 2. Absatz)
- b) **FALSCH** – Dies kann überprüft werden, nachdem Schritt (2) ausgeführt wurde, da die Meldung von Konflikten erst nach der Kompilierung und Einbindung des Codes erfolgen kann.
- c) **FALSCH** – Dies ist als Eingangskriterium für Schritt (3) besser geeignet, da es keine Relevanz für Schritt (2) hat.
- d) **FALSCH** – Dies ist als Endekriterium für Schritt (3) besser geeignet.

Frage 31	FL-5.1.4	K3	Punkte 1.0
----------	----------	----	------------

Sie möchten den Testaufwand für ein neues Projekt mit Hilfe einer auf Kennzahlen basierenden Schätzung abschätzen. Sie berechnen das Verhältnis von Testaufwand zu Entwicklungsaufwand, indem Sie die Durchschnittsdaten sowohl für den Entwicklungs- als auch für den Testaufwand aus vier historischen Projekten verwenden, die dem neuen Projekt ähnlich sind. Die Tabelle zeigt unten diese historischen Daten.

Projekt	Entwicklungskosten (€)	Testaufwand (€)
P1	800.000	40.000
P2	1.200.000	130.000
P3	600.000	70.000
P4	1.000.000	120.000

Der geschätzte Entwicklungsaufwand für das neue Projekt beträgt 800.000 EURO. Wie hoch schätzen Sie den Testaufwand in diesem Projekt ein?

Wählen Sie EINE Option! (1 aus 4)

a)	€40.000	<input type="checkbox"/>
b)	€80.000	<input checked="" type="checkbox"/>
c)	€81.250	<input type="checkbox"/>
d)	€82.500	<input type="checkbox"/>

FL-5.1.4 (K3) Der Lernende kann Schätzverfahren zur Berechnung des erforderlichen Testaufwands anwenden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.1.4)

Das genutzte Schätzverfahren basiert auf Verhältniszahlen, die sich aus den Durchschnittswerten historischer und ähnlicher Projekte als Kennzahlen ergibt (siehe [CTFL 4.0], Abschnitt 5.1.4, 3. Absatz).

Der durchschnittliche Entwicklungsaufwand betrug 900.000 EURO (Summe: 3.600.000 EURO) und der durchschnittliche Testaufwand 90.000 EURO (Summe: 360.000 EURO), berechnet aus den vier Projekten.

Das durchschnittliche Verhältnis von Testaufwand zu Entwicklungsaufwand beträgt 1:10 (90.000 €: 900.000 €), d. h. der Testaufwand betrug im Durchschnitt 10 % des Entwicklungsaufwands.

Wenn also ein Entwicklungsaufwand von 800.000 EURO erwartet wird, kann der Testaufwand wie folgt geschätzt werden:

$$10 \% * 800.000 \text{ €} = 0,1 * 800.000 \text{ €} = 80.000 \text{ €}.$$

Folglich:

- a) FALSCH
- b) KORREKT**
- c) FALSCH
- d) FALSCH

Frage 32	FL-5.1.5	K3	Punkte 1.0
----------	----------	----	------------

Sie wurden gebeten eine optimale, risikobasierte Ausführungsreihenfolge der folgenden Testfälle festzulegen, die bereits priorisiert und auf mögliche Abhängigkeiten hin untersucht wurden:

Testfall-ID	Priorität	Abhängig von
T1	3	-
T2	1	T1
T3	3	T2
T4	3	T2
T5	1	T3
T6	2	T4

Priorität 1 ist dringlicher als Priorität 2 usw.

Welche der folgenden Testabläufe berücksichtigt die oben genannten Abhängigkeiten und Prioritäten?

Wählen Sie **EINE** Option! (1 aus 4)

a)	T1 → T2 → T4 → T5 → T3 → T6	<input type="checkbox"/>
b)	T1 → T2 → T3 → T4 → T5 → T6	<input type="checkbox"/>
c)	T1 → T2 → T4 → T3 → T5 → T6	<input type="checkbox"/>
d)	T1 → T2 → T3 → T5 → T4 → T6	<input checked="" type="checkbox"/>

FL-5.1.5 (K3) Der Lernende kann Priorisierung von Testfällen anwenden [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

- a) FALSCH – T5 ist von T3 abhängig, da heißt, dass T3 vor T5 ausgeführt werden muss.
- b) FALSCH – T5 ist dringlicher als T4, so dass nach Ausführung von T3 als nächstes T5 ausgeführt werden sollte.
- c) FALSCH – T5 ist dringlicher als T4, so dass T4 nach T3 und T5 ausgeführt werden sollte, nicht davor.
- d) **KORREKT – Dieser Testausführungsplan berücksichtigt die Abhängigkeiten und Prioritäten korrekt. Der Testfall mit der höchsten Priorität (T5) benötigt T3. Dadurch wird T3 früher ausgeführt, als der gleich priorisierte Testfall T4. T4 wird nach T5 ausgeführt, damit abschließend T6 ausgeführt werden kann.**

Frage 33	FL-5.1.7	K2	Punkte 1.0
----------	----------	----	------------

**Welches der folgenden Elemente wird im Testquadrantenmodell dem Testquadranten Q1 („technologieorientiert“ und „Unterstützung des Teams“) zugeordnet?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Gebrauchstauglichkeitstests	<input type="checkbox"/>
b)	Smoke-Tests	<input type="checkbox"/>
c)	Benutzerabnahmetests	<input type="checkbox"/>
d)	Komponentenintegrationstests	<input checked="" type="checkbox"/>

FL-5.1.7 (K2) Der Lernende kann die Testquadranten und ihre Beziehungen zu Teststufen und Testarten zusammenfassen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.1.7)

- a) FALSCH – Usability-Tests sind geschäftsorientierte Tests, die das Produkt kritisieren (Q3).
- b) FALSCH – Smoke-Tests sind Tests, die das Produkt kritisch betrachten (Q4).
- c) FALSCH – Benutzerakzeptanztests sind geschäftsorientierte Tests, die das Produkt kritisieren (Q3).
- d) KORREKT – Komponentenintegrationstests sind technologieorientierte Tests, die das Team (die Entwicklung) unterstützen (Q1).**



Frage 34	FL-5.2.4	K2	Punkte 1.0
----------	----------	----	------------

**Welche Aussage im Zusammenhang mit dem Risikomanagement beschreibt die Beziehung zwischen Produktrisiko und Testplanung NICHT genau?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Die potenziellen Auswirkungen von IT-Sicherheitsschwachstellen wurden als signifikant hoch bewertet, so dass das Endkriterium für den IT-Sicherheitstest auf 99 bestandene Testfälle erhöht wurde.	<input type="checkbox"/>
b)	Die geforderte Qualität des Netzwerkmoduls ist unklar, was zu weiteren Risikoanalysen in diesem Bereich führen wird.	<input checked="" type="checkbox"/>
c)	Die Benutzer hatten Probleme mit der Benutzungsschnittstelle des bestehenden Systems, so dass zusätzliche Gebrauchstauglichkeitstest-Tests für das Nachfolgesystem geplant sind.	<input type="checkbox"/>
d)	Die Ladezeit von Webseiten ist entscheidend für den Erfolg der neuen Website, daher wird ein Experte für Performanztests in das Projekt einbezogen.	<input type="checkbox"/>

FL-5.2.4 (K2) Der Lernende kann mögliche Maßnahmen, die als Reaktion auf analysierte Produktrisiken ergriffen werden können, erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0)

- a) FALSCH – Das Thema IT-Sicherheitsmängel wird als hohes Produktrisiko eingestuft und als Maßnahme zur Risikominderung wird das Thema IT-Sicherheitstests inklusive eines hohen Überdeckungsgrades gewählt. Dies ist eine sinnvolle Maßnahme im Sinne der Risikominderung (siehe [CTFL 4.0], Abschnitt 5.2.4, 2. Absatz und 4. Aufzählungspunkt).
- b) KORREKT – Wenn die Qualität eines Moduls unklar ist, ist eine zusätzliche Risikoanalyse notwendig, aber dies ist keine Aufgabe der Risikosteuerung. Die Testplanung sollte darauf abzielen, die Unsicherheiten zu beseitigen und die Qualität sicherzustellen (siehe [CTFL 4.0], Abschnitt 5.2.4).
- c) FALSCH – Da die Anwender Probleme mit der Benutzungsschnittstelle des bisherigen Systems hatten, wird das Produktrisiko für die Gebrauchstauglichkeit der neuen Benutzungsschnittstelle als sehr hoch eingestuft, was dazu geführt hat, dass weitere Gebrauchstauglichkeitstests geplant sind. Diese Maßnahme zur Risikominderung (mehr Tests im geeigneten Testverfahren, höhere Überdeckung) ist sinnvoll und somit macht diese Aussage daher im Rahmen des Risikomanagements sinnvoll (siehe [CTFL 4.0], Abschnitt 5.2.4, 2. Absatz und 4. Aufzählungspunkt).
- d) FALSCH – Da die Zeit, die zum Laden von Webseiten benötigt wird, als entscheidend für den Erfolg der neuen Website eingestuft wurde, ist die Performanz der Website als Risiko eingestuft. Die Auswahl von Testern mit dem richtigen Maß an Erfahrung und Fähigkeiten, die für einen bestimmten Risikotyp geeignet sind, ist eine sinnvolle Risikominderungsmaßnahme (siehe [CTFL 4.0], Abschnitt 5.2.4, 2. Absatz und 1. Aufzählungspunkt). Der Einsatz eines Performanztest-Experten ist daher im Rahmen des Risikomanagements sinnvoll.

Frage 35	FL-5.3.1	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Kennzahlen ist eine Produktqualitätsmetrik?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Mittlere Betriebsdauer bis zum Ausfall (Mean time to Failure)	<input checked="" type="checkbox"/>
b)	Anzahl der aufgedeckten Fehlerzustände	<input type="checkbox"/>
c)	Anforderungsüberdeckung	<input type="checkbox"/>
d)	Fehlerdichte	<input type="checkbox"/>

FL-5.3.1 (K1) Der Lernende kann die beim Testen verwendeten Metriken wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.3.1)

- a) **KORREKT** – Produktqualitätsmetriken messen Qualitätsmerkmale. Die mittlere Zeit bis zum Ausfall misst die Reife und ist somit eine Kennzahl für die Produktqualität (siehe Lehrplan CTFL v4.0, Abschnitt 5.3.1, Aufzählungspunkt 3).
- b) **FALSCH** – Dies ist ein Beispiel für eine Fehlerzustandsmetrik, keine Produktqualitätsmetrik (siehe [CTFL 4.0], Abschnitt 5.3.1, Aufzählungspunkt 4).
- c) **FALSCH** – Dies ist ein Beispiel für eine Überdeckungsmetrik, keine Produktqualitätsmetrik (siehe [CTFL 4.0], Abschnitt 5.3.1, Aufzählungspunkt 6).
- d) **FALSCH** – Dies ist ein Beispiel für eine Fehlerzustandsmetrik, keine Produktqualitätsmetrik (siehe [CTFL 4.0], Abschnitt 5.3.1, Aufzählungspunkt 4).

Frage 36	FL-5.3.3	K2	Punkte 1.0
----------	----------	----	------------

Stellen Sie sich vor, Sie sind Teil eines agilen Teams mit Sitz in Nordamerika. Ihr Team entwickelt ein Produkt für einen Kunden mit Sitz in Europa. Das Team arbeitet nach dem DevOps-Ansatz und verwendet eine Pipeline für kontinuierliche Integration und kontinuierliche Bereitstellung für die Entwicklung.

Welche der folgenden Kommunikationsmethoden wäre angesichts der geografischen Entfernung und des agilen Charakters des Projekts am wenigsten effektiv, um den Testfortschritt an den Kunden zu übermitteln?

**Wählen Sie EINE Option! (1 aus 4)**

a)	Persönliche Treffen(von Angesicht zu Angesicht)	<input checked="" type="checkbox"/>
b)	Interaktive Dashboards	<input type="checkbox"/>
c)	E-Mail-Aktualisierungen	<input type="checkbox"/>
d)	Videokonferenzen	<input type="checkbox"/>

FL-5.3.3 (K2) Der Lernende kann Beispiele geben, wie man den Teststatus kommunizieren kann [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.3.3)

Laut Lehrplan ist nur die direkte Kommunikation von Angesicht zu Angesicht über verschiedene Zeitzonen hinweg nicht immer möglich (siehe [CTFL v4.0], Abschnitt 5.3.3, Absatz 2, Satz 2. Daher ist Option a) korrekt).

- a) **KORREKT** – Der Kunde befindet sich an einem anderen, weit entfernten Ort und in einer anderen Zeitzone, so dass es schwierig sein kann, persönlich zu kommunizieren (siehe [CTFL v4.0], Abschnitt 5.3.3, Absatz 2, Satz 2).
- b) FALSCH – Dashboards sind in der Regel für jeden Nutzer jederzeit verfügbar, so dass der Zeitzonenunterschied die Kommunikation nicht so sehr behindert wie die verbale Kommunikation von Angesicht zu Angesicht.
- c) FALSCH – Obwohl der Zeitunterschied zwischen Europa und Amerika mehrere Stunden beträgt und dies einige Unannehmlichkeiten mit sich bringen kann, ist er sicherlich nicht so groß wie bei der Kommunikation von Angesicht zu Angesicht.
- d) FALSCH – Videokonferenzen sind ein praktisches Kommunikationsmittel. Obwohl die Kommunikation zwischen Europa und Amerika während der Arbeitszeit normalerweise erfordert, dass sich eine Partei in sehr früh oder sehr spät verbindet, ist dies nicht so unangenehm wie die verbale Kommunikation von Angesicht zu Angesicht.

Frage 37	FL-5.4.1	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen beschreibt ein Beispiel dafür, wie Konfigurationsmanagement (KM) das Testen unterstützt?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Über die Versionsnummer der Testumgebung und die gespeicherten Beziehungen kann das KM-Werkzeug die Versionsnummern der in dieser Umgebung verwendeten Bibliotheken, Platzhalter und Treiber abrufen.	<input checked="" type="checkbox"/>
b)	Die Änderung von Baselines wird durch KM-Werkzeuge flexibel und pragmatisch unterstützt, wenn die Tester dies aufgrund unerwarteter Ereignisse während der Testdurchführung für notwendig erachten.	<input type="checkbox"/>
c)	Das Konfigurationsmanagement unterstützt die Verfolgung von Testskripten und Testfällen. Testergebnisse werden hingegen durch das Fehlermanagement verwaltet.	<input type="checkbox"/>
d)	Im Konfigurationsmanagement fasst komplexe Konfigurationselemente in einer Baseline zusammen. Um diese als Basis zu etablieren, können Tester später nicht mehr auf eine frühere Baseline zurückgreifen.	<input type="checkbox"/>

FL-5.4.1 (K2) Der Lernende kann mögliche Unterstützung des Testens durch das Konfigurationsmanagement zusammenfassen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.4.1)

- a) **KORREKT** – Für ein komplexes Konfigurationselement (z. B. eine Testumgebung) zeichnet KM die Elemente auf, aus denen es besteht, ihre Beziehungen und Versionen (siehe [CTFL 4.0], Abschnitt 5.4, Absatz 2, Satz 2).
- b) **FALSCH** – Für die Änderung von Baselines ist ein formaler Änderungskontrollprozess notwendig (siehe [CTFL 4.0], Abschnitt 5.4, Absatz 2, Satz 2).
- c) **FALSCH** – Das Konfigurationsmanagement unterstützt auch die Verfolgung von Testergebnissen (siehe [CTFL 4.0], Abschnitt 5.4, Absatz 1).
- d) **FALSCH** – Es ist möglich, zu einer früheren Baseline zurückzukehren (siehe [CTFL 4.0], Abschnitt 5.4, Absatz 3, Satz 2). Das ist z. B. auch notwendig, um frühere Ergebnisse nachvollziehen zu können.

Frage 38	FL-5.5.1	K3	Punkte 1.0
----------	----------	----	------------

**Sie testen eine Sortierfunktion, die eine Reihe von Zahlen als Eingabe erhält und dieselben Zahlen in aufsteigender Reihenfolge sortiert zurückgibt.**

**Das Protokoll der Testausführung sieht wie folgt aus:**

---

Environment configuration: sort function build 2.002.2182, test case set: TCS-3, # of TCs: 5  
 Test run ID: 736  
 Start 12:43:21.003

12:43:21.003	Execution of TC1.	Input: 3.	Output: 3.	Result: passed
12:43:21.003	Execution of TC2.	Input: 3, 11, 6, 5.	Output: 3, 5, 6, 11.	Result: passed
12:43:21.004	Execution of TC3.	Input: 8, 7, 3, 7, 1.	Output: 1, 3, 7, 8.	Result: failed
12:43:21.005	Execution of TC4.	Input: -2 -2 -2 -3 -3.	Output: -3, -2.	Result: failed
12:43:21.005	Execution of TC5.	Input: 0, -2, 0, 3, 4, 4.	Output: -2, 0, 3, 4.	Result: failed

End 12:43:21.005  
 Total time of test cycle: 0:00:00.002

---

**Welche der folgenden Beschreibungen ist die BESTE Fehlerbeschreibung, die in einem Fehlerbericht verwendet werden kann?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Das System kann mehrere Zahlensätze nicht sortieren. Referenz: TC3, TC4, TC5.	<input type="checkbox"/>
b)	Das System scheint Duplikate beim Sortieren zu ignorieren. Referenz: TC3, TC4, TC5.	<input checked="" type="checkbox"/>
c)	Das System kann keine negativen Zahlen sortieren. Referenz: TC4, TC5.	<input type="checkbox"/>
d)	TC3, TC4 und TC5 sind fehlerhaft (doppelte Eingabedaten) und sollten korrigiert werden.	<input type="checkbox"/>

FL-5.5.1 (K3) Der Lernende kann einen Fehlerbericht erstellen [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 5.5.1)

- a) FALSCH – Obwohl der Satz wahr ist, bietet er dem Entwickler keinen großen Mehrwert. Die beobachteten Abweichungen lassen eine detailliertere Analyse zu. Daher ist Option b) zu bevorzugen.
- b) KORREKT – Aus den Testergebnissen geht hervor, dass das System Duplikate ignoriert und die Liste ohne Berücksichtigung der Wiederholungen sortiert, bzw. die duplizierten Werte nur einmal zurückgibt. Dies ist wahrscheinlich die Ursache für Fehler in TC3, TC4, TC5. Solche Informationen können dem Entwickler helfen, den Fehler zu finden und ihn effizient zu beheben.
- c) FALSCH – Das System versagt nicht beim Sortieren negativer Zahlen. Das sieht man auch daran, dass die negativen Zahlen korrekt am Anfang der Reihe einsortiert werden. Das Problem besteht vielmehr darin, Duplikate zu ignorieren.
- d) FALSCH – Die Testfälle TC3, TC4 und TC5 schlagen fehl, aber wir kennen keine Fehler in den Testfällen. In der Spezifikation ist nicht beschrieben, dass Duplikate nicht verwendet werden dürfen bzw. dass alle Zahlen unterschiedlich sein müssen.



Frage 39	FL-6.1.1	K2	Punkte 1.0
----------	----------	----	------------

**Berücksichtigen Sie die folgenden Beschreibungen:**

- 1. Unterstützung der Verfolgung von Arbeitsabläufen**
- 2. Kommunikation erleichtern**
- 3. virtuelle Maschinen**
- 4. Unterstützung von Auswertungen bei Reviews**

**und die folgenden Kategorien von Testwerkzeugen:**

- A. Statische Testwerkzeuge**
- B. Werkzeuge zur Unterstützung der Skalierbarkeit und Standardisierung der Bereitstellung**
- C. DevOps-Werkzeuge**
- D. Werkzeuge für die Zusammenarbeit**

**Welche der folgenden Optionen ordnet die Beschreibungen und Kategorien am BESTEN einander zu?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	1A, 2B, 3C, 4D	<input type="checkbox"/>
b)	1B, 2D, 3C, 4A	<input type="checkbox"/>
c)	1C, 2D, 3B, 4A	<input checked="" type="checkbox"/>
d)	1D, 2C, 3A, 4B	<input type="checkbox"/>

FL-6.1.1 (K2) Der Lernende kann mögliche Unterstützung des Testens durch verschiedene Arten von Testwerkzeugen erklären [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 6.1.1)

Unter Berücksichtigung der einzelnen aufgeführten Werkzeugkategorien und ihrer Beschreibungen:

- A. Statische Testtools – unterstützen den Tester bei der Durchführung von Überprüfungen und statischen Analysen (4) – siehe [CTFL 4.0], Absatz 6.1.1, Aufzählungspunkt 2
- B. Tools, die Skalierbarkeit und Bereitstellungsstandardisierung unterstützen – zum Beispiel virtuelle Maschinen, Container-Tools (3) – siehe [CTFL 4.0], Absatz 6.1.1, Aufzählungspunkt 8
- C. DevOps-Tools – unterstützen die DevOps-Bereitstellungspipeline, Workflow-Tracking, automatisierte Build-Prozesse, kontinuierliche Integration/kontinuierliche Bereitstellung (CI/CD) (1) – siehe [CTFL 4.0], Absatz 6.1.1, Aufzählungspunkt 6
- D. Tools für die Zusammenarbeit – erleichtern die Kommunikation (2) – siehe [CTFL 4.0], Absatz 6.1.1, Aufzählungspunkt 7

Daher:

a) FALSCH

b) FALSCH

**c) KORREKT – Die richtige Übereinstimmung ist: 1C, 2D, 3B, 4A**

d) FALSCH

<b>Frage 40</b>	<b>FL-6.2.1</b>	<b>K1</b>	<b>Punkte 1.0</b>
-----------------	-----------------	-----------	-------------------

**Welcher der folgenden Vorteile trifft AM EHESTEN auf die Testautomatisierung zu?**

**Wählen Sie EINE Option! (1 aus 4)**

a)	Die Testautomatisierung ermöglicht auch die Messung komplexerer Überdeckungskriterien.	<input checked="" type="checkbox"/>
b)	Durch die Testautomatisierung wird ein Teil die Verantwortung für das Testen dem Werkzeuganbieter übertragen.	<input type="checkbox"/>
c)	Die Testautomatisierung macht kritisches Denken bei der Analyse von Testergebnissen überflüssig.	<input type="checkbox"/>
d)	Testautomatisierung generiert Testfälle auf Systemebene aus einer Analyse des Programmcodes.	<input type="checkbox"/>

FL-6.2.1 (K1) Der Lernende kann die Nutzen und Risiken von Testautomatisierung wiedergeben [CTFL 4.0]

Begründung: (siehe ISTQB® Foundation Level Syllabus v4.0; Abschnitt 6.1.2)

- a) **KORREKT** – Die Testautomatisierung kann Messungen bereitstellen, die zu kompliziert sind, um von Menschen abgeleitet zu werden, z. B. White-Box-Testüberdeckungsmaße für alle Codes außer den trivialsten (siehe Lehrplan CTFL v4.0, Abschnitt 6.2, Aufzählungspunkt 3).
- b) **FALSCH** – Bei der Verwendung von Testwerkzeugen wird die Verantwortung für das Testen NICHT mit dem Hersteller des Werkzeugs geteilt, da der Anbieter nicht am Testen beteiligt ist und es in der Verantwortung des Testers liegt. Die einzige mögliche Verantwortung, die dem Tool-Anbieter übertragen werden kann, ist die, dass das Tool nicht wie erwartet funktioniert und falsche Testergebnisse liefert (siehe [CTFL 4.0], Abschnitt 6.2, Aufzählungspunkt 11).
- c) **FALSCH** – Der Tester muss bei der Analyse von Anomalien in den Testergebnissen kritisch sein, um deren wahrscheinliche Ursache zu ermitteln (siehe [CTFL 4.0], Abschnitt 6.2, Aufzählungspunkt 10).
- d) **FALSCH** – Weder Tester noch Werkzeuge können Testfälle einfach aus einer Analyse des Programmcodes generieren, da der Code die Implementierung darstellt und keine Informationen über die erwarteten Ergebnisse liefert, die aus einem anderen Teil der Testbasis, z. B. der Designspezifikation, stammen müssen (siehe [CTFL 4.0], Abschnitt 6.2, Aufzählungspunkt 7).

**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)

**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)

**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)

**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)