Familienname, Vorname: _____

Firmenadresse: _____

_____

_____

Telefon: _____

Fax: _____

E-Mail-Adresse: _____

Rechnungsanschrift: _____

_____

_____

Schulungsunternehmen: _____

Referent: _____

Foundation Level – Probeprüfung

SET C v2.3.0 – GTB-Edition –

CTFL-Lehrplan Version v4.0

ISTQB® Certified Tester Foundation Level

Urheberrecht

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 kann dieses Dokument übersetzen.

Verantwortlich für dieses Dokument 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 (technischer Reviewer), Meile Posthuma, Gandhinee Rajkomar, Stuart Reid, Jean-François Riverin, Adam Roman, Lucjan Stapp, Stephanie Ulrich, Yaron Tsubery und Eshraka Zakaria.

Für seine Vorschläge und Anregungen dankt das Kernteam dem Reviewteam: 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 Medzihradszky, 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.

Änderungsübersicht

Version	Datum	Bemerkungen
2.0	24.11.2024	Lokalisierte Version und interne Reviews
2.1	16.02.2025	Final nach Bearbeitung durch Lektorin und Änderungen für CTFL 4.0.2 errata Konformität
2.2.1	06.03.2025	Korrektur der Begründung der Frage 17
2.3.0	08.06.2025	Wording der deutschsprachigen Fragen verbessert

Einführung

Dies ist eine Probeprüfung. Sie hilft den Kandidaten bei ihrer Vorbereitung auf die Zertifizierungsprüfung. Enthalten sind Fragen, deren Format der regulären ISTQB®/ GTB-Prüfung Certified Tester Foundation Level 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)

Frage 1	FL-1.1.1	K1	Punkte 1.0

Welche der folgenden Optionen stellt ein typisches Testziel dar?

Wählen Sie EINE Option! (1 aus 4)

a)	Validieren, ob die dokumentierten Anforderungen erfüllt sind.	☐
b)	Fehlerwirkungen provozieren und Fehlerzustände identifizieren.	☒
c)	Fehler initiieren und deren Ursachen identifizieren.	☐
d)	Verifikation, ob das Testobjekt die Erwartungen der Benutzer erfüllt.	☐

FL-1.1.1 (K1) Der Lernende kann typische Testziele identifizieren.

Begründung (siehe ISTQB®-Lehrplan V4.0, Abschnitt 1.1.1):

a) FALSCH – Die Validierung der Übereinstimmung mit dokumentierten Anforderungen ist nicht korrekt. „Validierung“ bezieht sich auf die Übereinstimmung mit den Anforderungen und Erwartungen der Benutzer, während „Verifizierung“ die Übereinstimmung mit den spezifizierten Anforderungen beschreibt. (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.1, 5. Aufzählungspunkt).

b) KORREKT – Das gezielte Hervorrufen von Fehlerwirkungen und die Identifikation von Fehlerzuständen ist ein zentrales Ziel dynamischer Tests (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.1, 2. Aufzählungspunkt).

c) FALSCH – Fehler werden nicht durch Tester „initiiert“, sondern entstehen typischerweise durch menschliche Fehler (z.B. von Entwicklern). Tester versuchen, Fehlerwirkungen aufzudecken. Die Ursachenanalyse ist Bestandteil des Debuggings und damit nicht Teil des eigentlichen Testens (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.1, 2. Aufzählungspunkt partiell sowie Abschnitt 1.1.2, letzter Absatz).

d) FALSCH – Verifikation bezieht sich auf die Prüfung, ob spezifizierte (dokumentierte) Anforderungen erfüllt sind. Die Erwartungen der Benutzer werden hingegen im Rahmen der Validierung überprüft. Daher ist hier der Begriff „Verifikation“ fehl am Platz (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.1, letzter Aufzählungspunkt).

Frage 2	FL-1.1.2	K2	Punkte 1.0

Welche der folgenden Aussagen beschreibt den Unterschied zwischen Testen und Debugging AM BESTEN?

Wählen Sie EINE Option! (1 aus 4)

a)	Beim Testen wird die Ursache von Fehlerzuständen identifiziert, während beim Debugging die Fehlerzustände analysiert und behoben werden.	☐
b)	Testen ist eine negative Aktivität, während Debugging eine positive Aktivität ist.	☐
c)	Beim Testen werden Fehler gefunden, während beim Debugging Fehler behoben werden.	☒
d)	Beim Testen wird die Fehlerursache gefunden, während beim Debugging die Fehlerursache behoben wird.	☐

FL- 1.1.2 (K2) Der Lernende kann Testen von Debugging unterscheiden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.1.2):

a) FALSCH – Beim Testen werden nicht die Ursachen von Fehlerzuständen identifiziert. Der Zweck des Debuggings besteht hingegen darin, Fehler zu analysieren und zu beheben (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.2, 1. und 2. Absatz).

b) FALSCH – Sowohl das Testen als auch das Debugging tragen zur Qualitätsverbesserung des Testobjekts bei und sollten daher beide als betrachtet werden. Debugging gilt allgemein als positiv, da es Fehler behebt. Zwar kann dynamisches Testen als negativ erscheinen, weil es das Testobjekt „zum Scheitern bringt“, doch dies ist eine einseitige Betrachtung. Testen umfasst sowohl positive Testfälle (Überprüfung des Sollverhaltens) als auch negative Testfälle (Überprüfung des Fehlverhaltens) (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.2, 1. und 2. Absatz).

c) KORREKT – Beim Testen werden Fehler gefunden - direkt (z.B. bei Reviews oder statischer Analyse) oder indirekt (durch das Auslösen von Fehlerwirkungen im dynamischen Test). Debugging ist eine vom Testen getrennte Aktivität, meist durch Entwickler durchgeführt, und umfasst sowohl die Lokalisierung als auch die Behebung von Fehlern (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.2, 1. und 2. Absatz).

d) FALSCH – Fehlerursachen werden beim Testen nicht ermittelt und beim Debugging nicht automatisch nicht automatisch behoben. Fehler entstehen typischerweise durch menschliches Versagen. Testen dient der Entdeckung von Fehlerwirkungen, Debugging der Analyse und Behebung von Defekten. Die Aussage stellt eine unzulässige Vereinfachung dar (siehe CTFL-Lehrplan V4.0, Abschnitt 1.1.2, 3. Absatz).

Frage 3	FL-1.3.1	K2	Punkte 1.0

Ein Grundsatz des Testens lautet:

„Der Trugschluss, dass ein fehlerfreies System auch automatisch brauchbar ist, muss vermieden werden.“

Welches der folgenden Beispiele zeigt, wie dieser Grundsatz in der Praxis berücksichtigt wird?

Wählen Sie EINE Option! (1 aus 4)

a)	Darauf hinweisen, das Testen niemals die die Abwesenheit von Fehlern beweisen kann.	☐
b)	Die Endnutzer bei der Durchführung von Abnahmetests unterstützen.	☒
c)	Sicherstellen, dass das gelieferte System keine Implementierungsfehler enthält.	☐
d)	Test verändern, die keine Fehler finden, um die Fehlerdichte zu verringern.	☐

FL-1.3.1 (K2) Der Lernende kann die sieben Grundsätze des Testens erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.3):

Der Irrglaube, dass „keine Fehler“ ein brauchbares System bedeuten, beruht auf der Vorstellung, dass die Sicherstellung der Korrektheit gemäß den Anforderungen (d. h. die Überprüfung der Abwesenheit von Implementierungsfehlern) keine Garantie für die Zufriedenheit der Benutzer mit dem System ist. Um dieses Problem anzugehen, muss auch überprüft werden, ob das System die Bedürfnisse und Erwartungen der Benutzer erfüllt, die Geschäftsziele erreicht und konkurrierende Systeme übertrifft.

a) FALSCH – Der Grundsatz „Testen zeigt das Vorhandensein, nicht die Abwesenheit von Fehlerzuständen“ bedeutet, dass das Testen zwar das Vorhandensein von Fehlern im Testobjekt nachweisen kann, dass es aber nicht möglich ist, die Abwesenheit von Fehlerzuständen nachzuweisen und damit die Korrektheit des Testobjekts zu gewährleisten. Daher würde die Erläuterung, dass es durch Tests nicht möglich ist, die Abwesenheit von Fehlern durch Testen nachzuweisen, teilweise diesem Grundsatz Rechnung tragen und nicht den Irrtum der „Abwesenheit von Fehlern“ widerspiegeln (siehe CTFL-Lehrplan V4.0, Abschnitt 1.3, 1. Grundsatz).

b) KORREKT – Durch die Unterstützung des Endbenutzers bei der Durchführung von Abnahmetests sollte es möglich sein, zu validieren, ob das System die Bedürfnisse und Erwartungen der Benutzer erfüllt (siehe CTFL-Lehrplan V4.0, Abschnitt 1.3, 7. Grundsatz, letzter Satz).

c) FALSCH – Es kann nicht sichergestellt werden, dass keine Implementierungsfehler im gelieferten System verbleiben, da der Grundsatz „Testen zeigt das Vorhandensein, nicht die Abwesenheit von Fehlern“ besagt, dass Tests zwar das Vorhandensein von Fehlern im Testobjekt erkennen können, es aber nicht möglich ist, die Abwesenheit von Fehlerzuständen zu beweisen und damit deren Korrektheit zu garantieren (siehe CTFL-Lehrplan V4.0, Abschnitt 1.3, 1. Grundsatz).

d) FALSCH – Die Modifikation von Tests, die keine Fehler verursachen, um sicherzustellen, dass nur wenige Fehler verbleiben, ist eine Möglichkeit, dem Prinzip „Tests nutzen sich ab“ zu begegnen. Bei diesem Grundsatz geht es um die Idee, dass die Wiederholung identischer Tests an unverändertem Code wahrscheinlich keine neuen Fehler aufdeckt und daher eine Änderung der Tests unerlässlich ist, sie bestätigt aber nicht, dass das System den Bedürfnissen und Erwartungen der Benutzer entspricht (siehe CTFL-Lehrplan V4.0, Abschnitt 1.3, 5. Grundsatz).

Frage 4	FL-1.4.1	K2	Punkte 1.0

Bei welchen der folgenden Testaktivitäten werden typischerweise die Testverfahren der Grenzwertanalyse und der Äquivalenzklassenbildung angewendet?

Wählen Sie ZWEI Optionen! (2 aus 5)

a)	Testrealisierung	☐
b)	Testentwurf	☒
c)	Testdurchführung	☐
d)	Testüberwachung	☐
e)	Testanalyse	☒

FL-1.4.1 (K2) Der Lernende kann die verschiedenen Testaktivitäten und die damit verbundenen Aufgaben erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.4.1):

Die Testanalyse wird wie folgt beschrieben:

Um die Merkmale zu identifizieren, die getestet werden müssen, wird die Testbasis analysiert und Testbedingungen identifiziert, die dann zusammen mit den damit verbundenen Risiken priorisiert werden. Die systematische Identifizierung von Testbedingungen als Überdeckungselemente erfordert häufig den Einsatz von Testverfahren sowohl während der Testanalyse als auch als Teil des Testentwurfs.

Aus der obigen Beschreibung ist ersichtlich, dass Testverfahren häufig in der Testanalyse und beim Testentwurf verwendet werden. Grenzwertanalyse und Äquivalenzklassenbildung sind Testverfahren.

a) FALSCH – Die Testrealisierung erfordert wahrscheinlich nicht den Einsatz von Testverfahren, da es sich hauptsächlich um die Zusammenstellung von Testfällen zu Testprozeduren handelt, während Testverfahren Testfälle erstellen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.1, 6. Absatz: "Die Testrealisierung umfasst ...").

b) KORREKT – Beim Testentwurf werden wahrscheinlich Testverfahren eingesetzt, um Testfälle aus Testbedingungen und Überdeckungselemente zu erstellen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.1, 5. Absatz: "Der Testentwurf umfasst ...").

c) FALSCH – Die Testdurchführung erfordert wahrscheinlich nicht den Einsatz von Testverfahren, da sie sich hauptsächlich mit der Ausführung von Testprozeduren (und damit Testfällen) befasst, während Testverfahren Testfälle erstellen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.1, 7. Absatz: "Die Testdurchführung umfasst ...").

d) FALSCH – Bei der Testüberwachung werden wahrscheinlich keine Testverfahren eingesetzt. Bei der Testüberwachung geht es hauptsächlich um die laufenden Kontrollen, um sicherzustellen, dass der Plan eingehalten wird, während Testverfahren Testfälle erstellen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.1, 3. Absatz: "Die Testüberwachung umfasst ...").

e) KORREKT – Die Testanalyse beinhaltet wahrscheinlich den Einsatz von Testverfahren, um Testbedingungen zu identifizieren (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.1, 4. Absatz: "Die Testanalyse umfasst ...").

Frage 5	FL-1.4.3	K2	Punkte 1.0

Gegeben sind folgende Testaktivitäten:

A. Testanalyse

B. Testentwurf

C. Testrealisierung

D. Testabschluss

Und folgende mögliche Testergebnisse (Testmittel):

1. Testüberdeckungen

2. Änderungsanforderungen

3. Testausführungsplan

4. Priorisierte Testbedingungen

Welche der folgenden Aussagen beschreibt die Zuordnung von Testaktivität zu erzeugtem Testmittel AM ZUTREFFENDSTEN? Wählen Sie EINE Option! (1 aus 4)

a)	1B, 2D, 3C, 4A	☒
b)	1B, 2D, 3A, 4C	☐
c)	1D, 2C, 3A, 4B	☐
d)	1D, 2C, 3B, 4A	☐

FL-1.4.3 (K2) Der Lernende kann Testmittel, die die Testaktivitäten unterstützen, unterscheiden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.4.3):

Betrachten wir jede der aufgeführten Testaktivitäten und ihre Testergebnisse:

A. Testanalyse – priorisierte Testbedingungen (4) (z. B. Akzeptanzkriterien) und Fehlerberichte für in der Testbasis identifizierte Fehler (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.3, 3. Aufzählungspunkt: "... (priorisierte) Testbedingungen (…)").

B. Testentwurf – priorisierte Testfälle, Test-Chartas, Überdeckungselemente (1), Testdatenanforderungen und Testumgebungsanforderungen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.3, 4. Aufzählungspunkt: "… Überdeckungselemente ...").

C. Testrealisierung – Testverfahren, automatisierte Testskripte, Testsuiten, Testdaten, Testausführungsplan (3) und Testumgebungselemente wie Stubs, Treiber, Simulatoren und Service-Virtualisierungen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.3, 5. Aufzählungspunkt: "Testausführungspläne …").

D. Testabschluss – Testabschlussbericht, dokumentierte gewonnene Erkenntnisse, Verbesserungsmaßnahmen und Änderungsanforderungen (2) (als Elemente des Produkt-Backlogs (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.3, letzter Aufzählungspunkt: "… Änderungsanträge ...").

Daher:

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

b) FALSCH

c) FALSCH

d) FALSCH

Frage 6	FL-1.4.5	K2	Punkte 1.0

Welche der folgenden Aussagen über Testrollen in der agilen Softwareentwicklung trifft AM ZUTREFFENSTEN zu?

Wählen Sie EINE Option! (1 aus 4)

a)	In agilen Teams liegt das Testmanagement hauptsächlich beim Team, während das Testen überwiegend von externen Spezialisten durchgeführt wird.	☐
b)	Die operative Rolle des Testens übernimmt vor allem die Testüberwachung und –steuerung, während das Testmanagement für Planung und Durchführung zuständig ist.	☐
c)	In der agilen Entwicklung übernimmt ein externer Testmanager teamübergreifende Aufgaben, während das Team selbst einige Testmanagementaufgaben intern umsetzt.	☒
d)	Das Testmanagement ist hauptsächlich für Testanalyse und Testentwurf verantwortlich, während die operative Rolle des Testens vor allem Implementierung und Ausführung übernimmt.	☐

FL-1.4.5 (K2) Der Lernende kann die verschiedenen Rollen beim Testen vergleichen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.4.5):

a) FALSCH – Obwohl es richtig ist zu sagen, dass bei der agilen Softwareentwicklung einige der Testmanagementaufgaben möglicherweise vom agilen Team selbst übernommen werden können, liegt die Testverantwortung nicht primär bei einer einzelnen Person außerhalb des Teams. Stattdessen ist es wahrscheinlicher, dass die Tests von verschiedenen Teammitgliedern durchgeführt werden, die dem Whole-Team-Ansatz folgen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.5).

b) FALSCH – Die Rolle des Testmanagements umfasst hauptsächlich Aktivitäten im Zusammenhang mit der Testplanung, der Testüberwachung und Teststeuerung sowie dem Testabschluss. Obwohl diese Aussage teilweise richtig ist, ist es falsch zu sagen, dass die Rolle des Testmanagements hauptsächlich für die Testüberwachung und Teststeuerung verantwortlich ist (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.5).

c) KORREKT – In der agilen Softwareentwicklung können einige der Testmanagementaufgaben vom agilen Team selbst übernommen werden. Bei Testaktivitäten, die mehrere Teams innerhalb einer Organisation betreffen, können diese Aufgaben jedoch auch von Testmanagern außerhalb des Entwicklungsteams durchgeführt werden (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.5, 2. Absatz).

d) FALSCH – Die Rolle des Testmanagements umfasst in erster Linie Aktivitäten im Zusammenhang mit der Testplanung, der Teststeuerung sowie dem Testabschluss, während die Rolle des Testens hauptsächlich für die technischen und fachlichen Aspekte des Testens verantwortlich ist, wie z. B. Testanalyse, Testentwurf, Testimplementierung und Testausführung. Daher ist die Rolle des Testmanagements normalerweise nicht für die Testanalyse und den Testentwurf verantwortlich, obwohl es richtig ist zu sagen, dass die Rolle des Testens hauptsächlich für die Testimplementierung und die Testausführung verantwortlich ist (siehe CTFL-Lehrplan V4.0, Abschnitt 1.4.5).

Frage 7	FL-1.5.2	K1	Punkte 1.0

Was ist ein typischer Vorteil des ganzheitlichen Teamansatzes (Whole-Team-Ansatz) in agilen Projekten?

Wählen Sie EINE Option! (1 aus 4)

a)	Das Team arbeitet ohne dedizierte Tester.	☐
b)	Die Teamdynamik wird verbessert.	☒
c)	Nur Mitglieder des Fachbereichs sind für das Testen zuständig.	☐
d)	Teams werden durch zusätzliche Rollen vergrößert.	☐

FL-1.5.2 (K2) Der Lernende kann die Vorteile des Whole-Team-Ansatzes wiedergeben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.5.2):

a) FALSCH – Der Whole-Team-Ansatz bedeutet nicht, dass Teams ohne Tester arbeiten. Stattdessen spielen Tester eine wesentliche Rolle, indem sie ihr Fachwissen mit dem Team teilen, Teststrategien erarbeiten und die Qualitätssicherung unterstützen. Sie arbeiten eng mit Entwicklern und Business-Vertretern zusammen, um Abnahmetests und Testautomatisierung zu optimieren (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.2).

b) KORREKT – Der Whole-Team-Ansatz fördert eine bessere Teamdynamik durch stärkere Zusammenarbeit und effektivere Kommunikation. Dies führt zu einer effizienteren Arbeitsweise, da Synergien durch den gezielten Einsatz unterschiedlicher Kompetenzen entstehen (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.2, 2. Absatz).

c) FALSCH – Der Whole-Team-Ansatz erfordert keine spezifischen Fachteams oder stark spezialisierte Rollen. Stattdessen wird Wert daraufgelegt, dass alle Teammitglieder flexibel Aufgaben übernehmen, sofern sie über die notwendigen Kompetenzen verfügen. Der Vorteil liegt in der Verteilung von Wissen und Verantwortung auf das gesamte Team (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.2).

d) FALSCH – Die Größe eines Teams ist kein entscheidender Faktor für den Whole-Team-Ansatz. Es gibt keine allgemeine Empfehlung, dass größere Teams besser funktionieren. Vielmehr kommt es darauf an, wie effektiv ein Team zusammenarbeitet, unabhängig von seiner Größe (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.2).

Frage 8	FL-1.5.3	K2	Punkte 1.0

Welche der folgenden Aussagen zur Unabhängigkeit des Testens trifft AM EHESTEN zu?

Wählen Sie EINE Option! (1 aus 4)

a)	Unabhängige Tester entdecken häufig Fehler, weil sie mit einer anderen Perspektive als die Entwickler testen – diese Unabhängigkeit kann jedoch zu Spannungen im Team führen.	☒
b)	Entwickler finden nur wenige Fehler in ihrem eigenen Code, und Tester mit ähnlichem Hintergrund sind beim Finden derselben Fehler ebenfalls eingeschränkt.	☐
c)	Unabhängiges Testen ist nur durch externe Tester außerhalb des Teams oder der Organisation möglich, was jedoch die Kenntnis der Fachdomäne erschwert.	☐
d)	Tester außerhalb des Teams gelten als unabhängiger, während interne Tester eher für Projektverzögerungen verantwortlich gemacht werden.	☐

FL-1.5.3 (K2) Der Lernende kann die Vor- und Nachteile des unabhängigen Testens unterscheiden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 1.5.3):

a) KORREKT – Unabhängige Tester sind effektiver in der Fehlerfindung, da sie eine andere Perspektive und Voreingenommenheit als Entwickler haben. Dies kann jedoch zu Kommunikationsproblemen und einer angespannten Beziehung zwischen Testern und Entwicklern führen. Zudem können unabhängige Tester als Engpass im Releaseprozess wahrgenommen werden (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.3, 1. und 4. Absatz).

b) FALSCH – Die Vertrautheit eines Entwicklers mit dem Code bedeutet nicht, dass er selten Fehler darin findet, sondern vielmehr, dass er viele Fehler in seinem eigenen Code effizient finden kann. Und der Grund dafür, dass Tester und Entwickler unterschiedliche Arten von Fehlern finden, ist normalerweise nicht, dass Entwickler und Tester den gleichen Hintergrund haben, sondern dass Entwickler einen anderen Hintergrund haben als Tester (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.3).

c) FALSCH – Tests können auf verschiedenen Ebenen der Unabhängigkeit durchgeführt werden, von keiner Unabhängigkeit des Autors bis hin zu einer sehr hohen Unabhängigkeit für Tester von außerhalb der Organisation. In den meisten Projekten werden mehrere Unabhängigkeitsstufen verwendet: Entwickler führen Komponententests und Komponentenintegrationstests durch, das Testteam führt System- und Systemintegrationstests durch und Unternehmensvertreter führen Abnahmetests durch. Tester können also Teil des Entwicklungsteams sein und müssen nicht von außerhalb der Organisation kommen. Die Kenntnis der Anwendungsdomäne variiert von Fall zu Fall und hängt nicht vom Grad der Unabhängigkeit ab (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.3).

d) FALSCH – Tests können auf verschiedenen Ebenen der Unabhängigkeit durchgeführt werden, von keiner Unabhängigkeit des Autors bis hin zu einer sehr hohen Unabhängigkeit für Tester von außerhalb der Organisation, wobei Tester von außerhalb des Entwicklungsteams im Allgemeinen unabhängiger sind als Tester innerhalb des Teams. Es gibt jedoch mehr Gründe zu der Annahme, dass Tester außerhalb des Teams wahrscheinlich stärker von den Entwicklern isoliert sind und daher eher für Verzögerungen bei der Produktfreigabe verantwortlich gemacht werden (siehe CTFL-Lehrplan V4.0, Abschnitt 1.5.3).

Frage 9	FL-2.1.2	K1	Punkte 1.0

Welche der folgenden Aussagen beschreibt eine bewährte Testpraxis, die in allen Softwareentwicklungs-Lebenszyklen gilt?

Wählen Sie EINE Option! (1 aus 4)

a)	Jeder Teststufe ist eine bestimmte Entwicklungsstufe direkt zugeordnet.	☐
b)	Jedes Testziel entspricht einem bestimmten Entwicklungsziel.	☐
c)	Jede Testaktivität steht in direktem Zusammenhand mit einer Benutzeraktivität.	☐
d)	Zu jeder Softwareentwicklungsaktivität gibt es eine passende Testaktivität.	☒

FL-2.1.2 (K1) Der Lernende kann gute Praktiken für das Testen, die für alle Softwareentwicklungslebenszyklen gelten, wiedergeben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 2.1.2):

a) FALSCH – Die Qualitätskontrolle bezieht sich auf alle Entwicklungsaktivitäten, was bedeutet, dass es für jede Softwareentwicklungsaktivität eine entsprechende Testaktivität gibt. Allerdings versuchen wir hier, Teststufen mit Entwicklungsstufen gleichzusetzen, und obwohl wir wissen, was mit „Teststufen“ gemeint ist, gibt es kein gemeinsames Verständnis des Begriffs „Entwicklungsstufe“ (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.2, 1. Aufzählungspunkt).

b) FALSCH – Zu jeder Softwareentwicklungsaktivität gehört eine entsprechende Testaktivität. Die Testziele sind jedoch sehr unterschiedlich. Beispielsweise könnte ein Testziel darin bestehen, sicherzustellen, dass ein Testobjekt eine vertragliche Anforderung erfüllt, wonach vor der Auslieferung eine bestimmte Art von Tests durchgeführt werden muss. In diesem Fall gibt es keinen Grund für ein entsprechendes Entwicklungsziel (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.2).

c) FALSCH – Die Qualitätskontrolle gilt für alle Entwicklungsaktivitäten, d. h., für jede Softwareentwicklungsaktivität gibt es eine entsprechende Testaktivität. Diese Symmetrie gilt jedoch nicht für Tests und Benutzeraktivitäten. Beispielsweise ist es bei einigen Systemen schwierig, die Endbenutzer überhaupt zu identifizieren. Außerdem konzentrieren sich einige Testaktivitäten auf Entwickler (z. B. Tests auf einfache Wartbarkeit), was keinen Benutzeraspekt betrifft (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.2).

d) KORREKT – Die Qualitätskontrolle bezieht sich auf alle Entwicklungsaktivitäten, was bedeutet: “Für jede Softwareentwicklungsaktivität gibt es eine entsprechende Testaktivität” (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.2, 1. Aufzählungspunkt).

Frage 10	FL-2.1.3	K1	Punkte 1.0

Welcher der folgenden Ansätze ist ein Beispiel für eine Test-First-Entwicklung?

Wählen Sie EINE Option! (1 aus 4)

a)	Komponententestgetriebene Entwicklung	☐
b)	Integrationstestgetriebene Entwicklung	☐
c)	Systemtestgetriebene Entwicklung	☐
d)	Akzeptanztestgetriebene Entwicklung	☒

FL-2.1.3 (K1) Der Lernende kann die Beispiele für Test-First-Ansätze in der Entwicklung wiedergeben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 2.1.3):

a) FALSCH – Die komponententestgetriebene Entwicklung ist kein Beispiel für einen Test-First-Ansatz. Test-First-Ansätze wie akzeptanztestgetriebene Entwicklung (ATDD) sind auf spezifische Methoden zur Definition von Tests vor der Implementierung fokussiert (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.3).

b) FALSCH – Integrationstestgetriebene Entwicklung ist kein korrektes Beispiel für einen Test-First-Ansatz. Test-First-Ansätze wie ATDD konzentrieren sich darauf, Tests vor der Implementierung zu erstellen, wobei die Akzeptanztests die Benutzeranforderungen abbilden (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.3).

c) FALSCH – Systemtestgetriebene Entwicklung ist ebenfalls kein Beispiel für einen Test-First-Ansatz. Test-First-Ansätze umfassen Methoden wie ATDD, die explizit auf die Definition von Tests vor der Implementierung abzielen (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.3).

d) KORREKT – Die Akzeptanztestgetriebene Entwicklung (ATDD) ist ein bekanntes Beispiel für einen Test-First-Ansatz, bei dem Akzeptanztests zur Definition der Anforderungen vor der Implementierung genutzt werden (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.3, 3. Absatz, 2. Aufzählungspunkt: "Tests werden geschrieben, bevor der Teil der Anwendung entwickelt wird, der die Tests erfüllt.“).

Frage 11	FL-2.1.5	K2	Punkte 1.0

Welche der folgenden Aussagen beschreibt den Shift-Left-Ansatz im Softwaretesten AM ZUTREFFENDSTEN?

Wählen Sie EINE Option! (1 aus 4)

a)	Manuelle Tests werden nach Zustimmung der Entwickler automatisiert, um Zeit und Kosten zu sparen.	☐
b)	Testaktivitäten werden möglichst früh im Entwicklungsprozess eingeplant, um Fehler frühzeitig zu erkennen und Qualitätskosten zu senken.	☒
c)	Tester automatisieren Regressionstests, sobald Zeit dafür zur Verfügung steht – beginnend mit Komponententests.	☐
d)	Tester werden geschult, um frühe Aktivitäten zu übernehmen, sodass spätere Tests besser automatisiert werden können.	☐

FL-2.1.5 (K2) Der Lernende kann Shift-Left erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 2.1.5):

a) FALSCH – Die Praktiken des Shift-Left-Testens zielen darauf ab, mehr Testaktivitäten in den frühen Phasen des Entwicklungslebenszyklus zu implementieren und den SDLC so darzustellen, als würde er sich von links nach rechts bewegen (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.5).

b) KORREKT – Shift-Left betont, wie wichtig es ist, mit dem Testen früher im Softwareentwicklungslebenszyklus (SDLC) zu beginnen. Die Implementierung von Shift-Left-Tests erfordert zusätzliche Schulungen sowie einen höheren Aufwand und höhere Kosten in den frühen Phasen des SDLC, aber die Gesamteinsparungen sollten höher sein (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.5. 1. und 3. Absatz).

c) FALSCH – Obwohl automatisierte Komponententests und Komponentenintegrationstests im Allgemeinen für Regressionstests nützlich sind, liegt die Erstellung dieser Tests normalerweise in der Verantwortung der Entwickler, und wenn ein Continuous Integration/Continuous Delivery-(CI/CD-)Ansatz verfolgt wird, werden diese Tests zusammen mit dem Code ausgeliefert. In einigen Situationen kann der Tester Tests für Regressionstests und manchmal sogar für Komponententests und Komponentenintegrationstests automatisieren. Dies ist jedoch nicht Teil eines Shift-Left-Ansatzes, der Tests früher im SDLC verschiebt (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.5).

d) FALSCH – Die Ausbildung von Testern für die Durchführung von Aufgaben zu einem frühen Zeitpunkt im SDLC würde einen Shift-Left-Ansatz unterstützen, indem die Bedeutung des früheren Beginns des Testens im SDLC betont wird. Die Automatisierung anderer Testaktivitäten, die später im SDLC durchgeführt werden sollen, ist jedoch nicht Teil eines Shift-Left-Ansatzes (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.5).

Frage 12	FL-2.1.6	K2	Punkte 1.0

Welches der folgenden Ereignisse ist im Rahmen einer Retrospektive AM WENIGSTEN wahrscheinlich?

Wählen Sie EINE Option! (1 aus 4)

a)	Die Qualität künftiger Testobjekte wird verbessert, weil Schwächen in den Entwicklungspraktiken erkannt werden.	☐
b)	Die Testeffizienz steigt, weil Testumgebungen künftig automatisiert werden.	☐
c)	Endanwender gewinnen ein besseres Verständnis für Entwicklungs- und Testprozesse.	☒
d)	Entwickler geben Rückmeldung, durch die automatisierte Testskripte optimiert werden.	☐

FL-2.1.6 (K2) Der Lernende kann den Einsatz von Retrospektiven als Mechanismus zur Prozessverbesserung erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 2.1.6):

a) FALSCH – Ein Ziel von Retrospektiven ist es, potenzielle Prozessverbesserungen zu identifizieren, die, wenn sie in die Praxis umgesetzt werden, zu einer höheren Qualität zukünftiger Ergebnisse des Entwicklungsprozesses (Testobjekte) führen sollten. Dies ist daher wahrscheinlich das Ergebnis einer Retrospektive (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.6).

b) FALSCH – Einer der Vorteile von Retrospektiven für das Testen ist eine höhere Testeffizienz durch Prozessverbesserungen. Dies ist daher wahrscheinlich das Ergebnis einer Retrospektive (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.6).

c) KORREKT – Zu den Teilnehmern an Retrospektiven gehören typischerweise Tester, Entwickler, Architekten, Produktbesitzer und Businessanalysten, aber Endbenutzer werden selten zu diesen Treffen eingeladen oder nehmen selten daran teil – und es ist auch unwahrscheinlich, dass sie Berichte von diesen Treffen erhalten. Daher ist es sehr unwahrscheinlich, dass sie durch Retrospektiven mehr über die Entwicklungs- und Testprozesse erfahren und verstehen (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.6, 1. Absatz).

d) FALSCH – Ein Vorteil von Retrospektiven für das Testen ist die Verbesserung der Qualität der Testware (einschließlich automatisierter Testskripte) durch gemeinsame Reviews mit den Entwicklern. Dies ist daher wahrscheinlich das Ergebnis einer Retrospektive (siehe CTFL-Lehrplan V4.0, Abschnitt 2.1.6).

Frage 13	FL-2.2.1	K2	Punkte 1.0

In welcher der folgenden Teststufen steht die Validierung im Vordergrund, und sie wird typischerweise nicht von Testern durchgeführt?

Wählen Sie EINE Option! (1 aus 4)

a)	Komponententest	☐
b)	Komponentenintegrationstest	☐
c)	Systemintegrationstest	☐
d)	Abnahmetest	☒

FL-2.2.1 (K2) Der Lernende kann die verschiedenen Teststufen unterscheiden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 2.2.1):

a) FALSCH – Komponententests (auch Unit-Tests genannt) beinhalten das isolierte Testen einzelner Komponenten und sind in der Regel eine Verifikation gegen eine Spezifikation und keine Validierung gegen Benutzeranforderungen. Diese Tests werden in der Regel nicht von Testern durchgeführt, da die Entwickler diese Tests normalerweise in ihrer Entwicklungsumgebung durchführen (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.1).

b) FALSCH – Das Testen der Komponentenintegration umfasst das Testen der Schnittstellen und Interaktionen zwischen Komponenten und ist in der Regel eine Verifikation gegen eine Spezifikation und nicht eine Validierung anhand von Benutzeranforderungen. Diese Tests werden normalerweise nicht von Testern, sondern von Entwicklern durchgeführt (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.1).

c) FALSCH – Bei Systemintegrationstests werden die Schnittstellen zu anderen Systemen und externen Diensten untersucht. Dabei handelt es sich in der Regel um eine Überprüfung anhand einer Spezifikation und nicht um eine Validierung anhand von Benutzeranforderungen. Diese Art von Tests wird auch am häufigsten von Testern durchgeführt (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.1).

d) KORREKT – Abnahmetests dienen der Überprüfung, ob das System die fachlichen Anforderungen des Benutzers erfüllt und für den Einsatz bereit ist. Im Idealfall werden diese Tests von den Endbenutzern durchgeführt (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.1, 5. Aufzählungspunkt).

Fazit:

- **Antwort d) ist korrekt**, da Abnahmetests sich direkt auf die Validierung von Benutzeranforderungen beziehen und typischerweise nicht von Testern, sondern von Endbenutzern oder deren Stellvertretern durchgeführt werden.
- **Antworten a), b) und c) sind falsch**, da sie sich auf die Verifikation (Überprüfung gegen Spezifikationen) konzentrieren und nicht auf die Validierung (Überprüfung gegen Benutzeranforderungen).

Frage 14	FL-2.2.3	K2	Punkte 1.0

Nach einem Software-Update schlägt das Navigationssystem z.B. Routen durch Einbahnstraßen in falscher Richtung vor – ein klarer Verstoß gegen Verkehrsregeln.Welche der folgenden Maßnahmen sind in diesem Fall AM SINNVOLLSTEN?

Wählen Sie EINE Option! (1 aus 4)

a)	Nur Fehlernachtests	☐
b)	Fehlernachtests gefolgt von Regressionstests	☒
c)	Nur Regressionstests	☐
d)	Regressionstests gefolgt von Fehlernachtests	☐

FL-2.2.3 (K2) Der Lernende kann Fehlernachtests von Regressionstests unterscheiden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 2.2.3):

a) FALSCH – Es ist ein Fehlernachtest erforderlich, um zu überprüfen, ob die Aktualisierungen zu einer korrekten Implementierung geführt haben. Es wäre jedoch sinnvoll, einen Regressionstest durchzuführen, um sicherzustellen, dass keine Fehler in unveränderten Bereichen des Systems eingeführt oder aufgedeckt wurden (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.3).

b) KORREKT – Durch Fehlernachtests wird überprüft, ob die Aktualisierungen zu einer korrekten Implementierung geführt haben. Anschließend werden Regressionstests durchgeführt, um sicherzustellen, dass keine Fehler in unveränderten Bereichen des Systems eingeführt oder aufgedeckt wurden (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.3, 2. und 4. Absatz).

c) FALSCH – Regressionstests sollten verwendet werden, um sicherzustellen, dass durch die Aktualisierung keine Fehler in unveränderten Bereichen des Systems eingeführt oder aufgedeckt wurden. Es ist jedoch auch erforderlich, Fehlernachtests durchzuführen, um zu überprüfen, ob die Aktualisierungen zu einer korrekten Implementierung geführt haben (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.3).

d) FALSCH – Durch Fehlernachtests (Bestätigungstests) wird überprüft, ob die Aktualisierungen zu einer korrekten Implementierung geführt haben. Regressionstests werden durchgeführt, um sicherzustellen, dass keine Fehler in unveränderten Bereichen des Systems eingeführt oder aufgedeckt wurden. Wenn jedoch ein Fehlernachtest durchgeführt wird (d. h., wenn ein Update getestet werden muss), geht er dem Regressionstest voraus (siehe CTFL-Lehrplan V4.0, Abschnitt 2.2.3).

Fazit:

- **Antwort b) ist korrekt**, da Fehlernachtests zuerst durchgeführt werden müssen, gefolgt von Regressionstests.
- **Antworten a), c) und d) sind falsch**, da sie entweder Regressionstests überbetonen oder die Reihenfolge der Tests nicht korrekt wiedergeben.

Frage 15	FL-3.1.3	K2	Punkte 1.0

Welche der folgenden Fehlerarten können typischerweise durch statische Tests – im Gegensatz zu dynamischen Tests – erkannt werden?

Wählen Sie ZWEI Optionen! (2 aus 5)

a)	Widersprüche zwischen verschiedenen Teilen der Entwurfsspezifikation	☒
b)	Längere Antwortzeiten führen zur Frustration der Benutzer	☐
c)	Ein Fehler tritt auf, wenn beim Speichern einer Datei der Speicherplatz knapp wird	☐
d)	Eine im Code deklarierte Variable wird nicht verwendet	☒
e)	Das System benötigt beim Erstellen eines Berichts übermäßig viel Speicher	☐

FL-3.1.3 (K2) Der Lernende kann statischen Test und dynamischen Test vergleichen und gegenüberstellen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 3.1.3):

Unter Berücksichtigung jedes der aufgeführten Fehlerbeispiele:

a) KORREKT – Zwei verschiedene Teile der Entwurfsspezifikation stimmen aufgrund der Komplexität des Entwurfs nicht überein – dies ist ein Beispiel für einen Spezifikationsfehler, der Inkonsistenzen, Mehrdeutigkeiten, Widersprüche, Auslassungen, Ungenauigkeiten und Duplikate umfasst, die am einfachsten durch statische Tests gefunden werden können (siehe CTFL-Lehrplan V4.0, Abschnitt 3.1.3, 2. Absatz, 2. Aufzählungspunkt).

b) FALSCH – Eine Antwortzeit ist zu lang und lässt den Benutzer die Geduld verlieren – dies ist ein Beispiel für einen Antwortzeitfehler (Laufzeitfehler), der in der Praxis nur durch Ausführen des Programms und Messen der Antwortzeit gefunden werden kann, was am einfachsten durch dynamisches Testen festgestellt werden kann (siehe CTFL-Lehrplan V4.0, Abschnitt 3.1.3, 1. Absatz, 5. Aufzählungspunkt).

c) FALSCH – Speicherprobleme und Dateischreibvorgänge können nur durch dynamisches Testen entdeckt werden, da sie das tatsächliche Verhalten des Systems zur Laufzeit betreffen (siehe CTFL-Lehrplan V4.0, Abschnitt 3.1.3, 1. Absatz, 2. Aufzählungspunkt).

d) KORREKT – Eine Variable wird deklariert, dann aber nie im Programm verwendet – dies ist ein Beispiel für einen Programmierfehler, der Variablen mit undefinierten Werten, nicht deklarierte Variablen, doppelten oder nicht zugänglichen Code und übermäßige Codekomplexität umfasst und am einfachsten durch statische Tests gefunden werden kann (siehe CTFL-Lehrplan V4.0, Abschnitt 3.1.1 und Abschnitt 3.1.3, 2. Absatz, 3. Aufzählungspunkt).

e) FALSCH – Die Menge an Speicher, die das Programm benötigt, um einen Bericht zu erstellen, ist zu groß – dies ist ein Beispiel für einen Performance-Fehler, der in der Praxis nur durch Ausführen des Programms und Messen des verwendeten Speichers gefunden werden kann, was am einfachsten durch dynamische Tests festgestellt werden kann (siehe CTFL-Lehrplan V4.0, Abschnitt 3.1.3, 1. Absatz, 5. Aufzählungspunkt).

Fazit:

Die Antworten a) und d) sind korrekt, da diese Fehler durch statische Tests (z. B. Review oder statische Codeanalyse) identifiziert werden können.

Die Antworten b), d) und e) sind falsch, sie betreffen Laufzeitverhalten und Systemressourcen und müssen durch dynamische Tests aufgedeckt werden.

Frage 16	FL-3.2.1	K1	Punkte 1.0

Welcher der folgenden Punkte ist ein typischer Vorteil von frühem und häufigem Stakeholder-Feedback?

Wählen Sie EINE Option! (1 aus 4)

a)	Änderungen an Anforderungen werden frühzeitig erkannt und können schneller umgesetzt werden.	☒
b)	Geschäftsinteressenten erhalten eine Schulung zu den Benutzeranforderungen.	☐
c)	Product Owner können ihre Anforderungen beliebig oft ändern.	☐
d)	Endnutzer werden vor der Freigabe über nicht umgesetzte Anforderungen informiert.	☐

FL-3.2.1 (K2) Der Lernende kann Vorteile eines frühzeitigen und häufigen Stakeholder-Feedbacks erkennen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 3.2.1):

a) KORREKT – Es kann von großem Nutzen sein, frühzeitig und häufig im Softwareentwicklungsprozess Feedback von Stakeholdern einzuholen. Es erleichtert die frühzeitige Kommunikation potenzieller Qualitätsprobleme, kann Missverständnisse über Anforderungen verhindern und stellt sicher, dass Änderungen der Stakeholder-Anforderungen schneller verstanden und umgesetzt werden (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.1, 2. Absatz, 1. Satz).

b) FALSCH – Das Feedback stammt von Stakeholdern, und es ist unwahrscheinlich, dass die Bereitstellung von Feedback ihr Verständnis für die eigenen Benutzeranforderungen verbessert (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.1).

c) FALSCH – Es kann von großem Nutzen sein, frühzeitig und häufig im Softwareentwicklungsprozess Feedback von Stakeholdern einzuholen. Es erleichtert die frühzeitige Kommunikation potenzieller Qualitätsprobleme, kann Missverständnisse über Anforderungen verhindern und stellt sicher, dass Änderungen der Stakeholder-Anforderungen schneller verstanden und umgesetzt werden. Da Änderungen der Anforderungen jedoch schneller verstanden und umgesetzt werden können, bedeutet dies nicht, dass unbegrenzte Änderungen der Anforderungen gefördert werden (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.1).

d) FALSCH – Das Feedback stammt von Stakeholdern und umfasst nicht die Kommunikation mit ihnen. Zu den Mitteilungen an Endbenutzer könnte gehören, dass sie vor der Freigabe darüber informiert werden, welche Anforderungen nicht umgesetzt werden. Idealerweise sollte dies jedoch überhaupt nicht geschehen (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.1).

Fazit:

- **Antwort a) ist korrekt**, da Stakeholder-Feedback hilft, Änderungen früher zu verstehen und umzusetzen.
- **Antworten b), c) und d) sind falsch**, da sie Aspekte ansprechen, die nicht direkt durch Stakeholder-Feedback verbessert oder ermöglicht werden.

Frage 17	FL-3.2.4	K2	Punkte 1.0

Ordnen Sie die folgenden Reviewarten den entsprechenden Beschreibungen zu:

1. **Technisches Review**
2. **Informelles Review**
3. **Inspektion**
4. **Walkthrough**

A. **Konzentriert sind auf Ziele wie Vertrauensaufbau, Generierung neuer Ideen und Schulung von Gutachtern unter Leitung des Autors.**

B. **Dient hauptsächlich der Erkennung potenzieller Mängel und erfordert keine formelle Dokumentation.**

C. **Ziel ist es, möglichst viele Fehler zu finden und Metriken für die Prozessverbesserung zu erheben.**

D. **Zielt darauf ab, durch qualifizierte Gutachter einen Konsens zu erreichen, Entscheidungen zu treffen und neue Ideen zu entwickeln.**

Welche der folgenden Optionen beschreibt die korrekte Zuordnung AM BESTEN?

Wählen Sie EINE Option! (1 von 4)

a)	1A, 2B, 3C, 4D	☐
b)	1D, 2B, 3C, 4A	☒
c)	1B, 2C, 3D, 4A	☐
d)	1C, 2D, 3A, 4B	☐

FL-3.2.4 (K2) Der Lernende kann verschiedene Arten von Reviews vergleichen und gegenüberstellen.

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 3.2.4):

Bei Betrachtung der einzelnen aufgeführten Review-Typen:

1. Das technische Review – Diese Art der Begutachtung wird von technisch qualifizierten Gutachtern unter der Leitung eines Moderators durchgeführt. Die Ziele bestehen darin, einen Konsens zu erzielen und Entscheidungen über technische Probleme zu treffen, während gleichzeitig die Qualität bewertet und Vertrauen in das Arbeitsprodukt aufgebaut wird, neue Ideen generiert werden, Autoren motiviert und befähigt werden, sich zu verbessern, und Anomalien aufgedeckt werden (siehe [CTFL 4.0], Abschnitt 3.2.4, 3. Absatz “Technisches Review”: "Die Ziele ... sind die Erzielung eines Konsenses ... aber auch ... die Entwicklung neuer Ideen").

Zuordnung: 1D

2. Das Informelle Review – Das primäre Ziel besteht darin, Anomalien zu erkennen. Der Prozess ist nicht definiert und erfordert keine formale Dokumentation (siehe [CTFL 4.0], Abschnitt 3.2.4, 3. Absatz “Informelles Review: Das Hauptziel ist die Aufdeckung von Anomalien” (Erkennung potenzieller Fehlerzustände) “und erfordert keine formalen, dokumentierten Ergebnisse").

Zuordnung: 2B

3. Die Inspektion ist die formellste Art eines Reviews und folgt dem generischen Reviewprozess. Das Hauptziel besteht darin, die meisten Anomalien zu finden. Zu den weiteren Zielen gehören die Bewertung der Qualität und die Schaffung von Vertrauen in das Arbeitsprodukt, die Motivation und Befähigung der Autoren zur Verbesserung sowie die Erfassung von Kennzahlen, die zur Verbesserung des Softwareentwicklungslebenszyklus (SDLC) einschließlich des Inspektionsprozesses verwendet werden können. Der Autor kann nicht als Leiter oder Protokollführer eines Reviews fungieren (siehe [CTFL 4.0], Abschnitt 3.2.4, 3. Absatz “Inspektion”: "Das Hauptziel ... die maximale Anzahl von Anomalien zu finden ...” und es werden “Metriken gesammelt").

Zuordnung: 3C

4. Walkthrough – Diese Art von Review wird vom Autor geleitet und dient verschiedenen Zielen, wie z. B. der Bewertung der Qualität und dem Aufbau von Vertrauen in das Arbeitsprodukt, der Schulung von Gutachter, der Konsensfindung, der Generierung neuer Ideen, der Motivation und Befähigung von Autoren zur Verbesserung und der Erkennung von Anomalien. Gutachter können vor dem Walkthrough ein individuelles Review durchführen, dies ist jedoch nicht zwingend erforderlich (siehe [CTFL 4.0], Abschnitt 3.2.4, 3. Absatz “Walkthrough”: "Ziele umfassen die Erzielung eines Konsenses” und die ”Schulung von Gutachtern").

Zuordnung: 4A

Somit:

a) FALSCH

b) KORREKT

c) FALSCH

d) FALSCH

Frage 18	FL-3.2.5	K1	Punkte 1.0

Welcher der folgenden Faktoren trägt typischerweise zum Erfolg eines Reviews bei?

Wählen Sie EINE Option! (1 aus 4)

a)	Das Management wird als Gutachter in alle Reviews eingebunden.	☐
b)	Große Arbeitsprodukte werden in kleinere, überschaubare Einheiten aufgeteilt.	☒
c)	Ziel des Reviews ist ausschließlich die Bewertung durch den Gutachter.	☐
d)	Für jedes Review wird genau ein Dokument eingeplant.	☐

FL-3.2.5 (K1) Der Lernende kann die Faktoren, die zu einem erfolgreichen Review beitragen, wiedergeben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 3.2.5):

a) FALSCH – Um erfolgreiche Reviews zu gewährleisten, ist es wichtig, die Unterstützung des Managements für den Reviewprozess sicherzustellen. Das bedeutet jedoch nicht, dass sie als Reviewer teilnehmen sollten (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.5).

b) KORREKT – Um erfolgreiche Überprüfungen sicherzustellen, ist es wichtig, das Arbeitsergebnis in Teile aufzuteilen, die klein genug sind, um in einem angemessenen Zeitrahmen überprüft zu werden. Damit wird verhindert, dass Prüfer während einzelner Überprüfungen oder Überprüfungsbesprechungen den Fokus verlieren (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.5, 3. Aufzählungspunkt).

c) FALSCH – Um erfolgreiche Überprüfungen zu gewährleisten, ist es wichtig, klare Ziele und messbare Abschlusskriterien zu definieren, ohne die “Bewertung der Teilnehmer” vorzunehmen (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.5, 1. Aufzählungspunkt).

d) FALSCH – Um sicherzustellen, dass Überprüfungen erfolgreich sind, ist es wichtig, die Überprüfung in kleinere Abschnitte zu unterteilen, um zu verhindern, dass Gutachter während einzelner Reviews oder Reviewsitzungen den Fokus verlieren. Sie sollten also NICHT planen, ein Dokument pro Review abzudecken (siehe CTFL-Lehrplan V4.0, Abschnitt 3.2.5).

Fazit:

- **Antwort b) ist korrekt**, da das Aufteilen großer Arbeitsprodukte die Effizienz und Qualität von Reviews verbessert.
- **Antworten a), c) und d) sind falsch**, da sie entweder den falschen Fokus setzen oder ineffiziente Reviewpraktiken widerspiegeln.

Frage 19	FL-4.1.1	K2	Punkte 1.0

Was ist der zentrale Unterschied zwischen Black-Box-Verfahren und erfahrungsbasierten Testverfahren?

Wählen Sie EINE Option! (1 aus 4)

a)	Das getestete Objekt	☐
b)	Die Teststufe, auf der das Testverfahren angewendet wird.	☐
c)	Die zugrunde liegende Testbasis	☒
d)	Die Phase im Softwareentwicklungslebenszyklus (SDLC), in der das Testverfahren eingesetzt wird.	☐

FL-4.1.1 (K2) Der Lernende kann Black-Box-Testverfahren, White-Box-Testverfahren und erfahrungsbasierte Testverfahren unterscheiden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.1):

a) FALSCH – In den meisten Fällen können für dieselben Testobjekte sowohl Black-Box-Testverfahren als auch erfahrungsbasierte Testverfahren eingesetzt werden. Die Wahl des Testverfahrens hängt nicht primär vom Testobjekt ab (siehe CTFL-Lehrplan V4.0, Abschnitt 4.1).

b) FALSCH – Auf allen Teststufen können sowohl Black-Box-Testverfahren als auch erfahrungsbasierte Testverfahren eingesetzt werden (siehe CTFL-Lehrplan V4.0, Abschnitt 4.1).

c) KORREKT – Black-Box-Testverfahren (auch spezifikationsbasierte Verfahren genannt) basieren auf einer Analyse des spezifizierten Verhaltens des Testobjekts ohne Bezug auf seine interne Struktur. Die Testbasis ist also in der Regel eine Spezifikation. Erfahrungsbasierte Testverfahren nutzen das Wissen und die Erfahrung von Testern effektiv für den Entwurf und die Implementierung von Testfällen. Dies bedeutet, dass der Tester beim Entwerfen von Tests die Spezifikation möglicherweise überhaupt nicht verwenden darf (siehe CTFL-Lehrplan V4.0, Abschnitt 4.1 und Abschnitt 2.2.2).

d) FALSCH – Erfahrungsbasierte Testverfahren können Fehler erkennen, die mit Black-Box- und White-Box-Testverfahren möglicherweise übersehen werden. Daher ergänzen erfahrungsbasierte Testverfahren Black-Box-Testverfahren und White-Box-Testverfahren, und sowohl Black-Box-Testverfahren als auch erfahrungsbasierte Testverfahren können in allen SDLCs verwendet werden (siehe CTFL-Lehrplan V4.0, Abschnitt 4.1).

Fazit:

- **Antwort c) ist korrekt**, da die **Testbasis** den entscheidenden Unterschied zwischen Black-Box- und erfahrungsbasierten Testverfahren ausmacht.
- **Antworten a), b) und d) sind falsch**, da sie sich auf Faktoren beziehen, die nicht ausschlaggebend für die Unterscheidung dieser Testverfahren sind.

Frage 20	FL-4.2.1	K3	Punkte 1.0

Ein PIN-Validator akzeptiert nur gültige PINs und lehnt ungültige ab.

Eine PIN gilt als gültig, wenn sie aus genau vier Ziffern besteht, von denen mindestens zwei unterschiedlich sind.

Welche der folgenden Datensätze deckt alle relevanten Äquivalenzklassen ab (gültige und ungültige PINs)?

Wählen Sie EINE Option! (1 aus 4)

a)	112, 1111, 1234, 123456	☒
b)	1, 123, 1111, 1234	☐
c)	12, 112, 1112, 11112	☐
d)	1, 111, 1111, 11111	☐

FL-4.2.1 (K3) Der Lernende kann die Äquivalenzklassenbildung zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.2.1):

Die Äquivalenzklassen für gültige und ungültige PINs sind:

1. **Gültige PINs**:
   - Eine PIN mit **genau vier Ziffern**, von denen **mindestens zwei unterschiedlich sind** (z. B. **1234, 1123, 5678**).
2. **Ungültige PINs**:
   - PINs mit **weniger als vier Ziffern** (z. B. **1, 12, 123**).
   - PINs mit **mehr als vier Ziffern** (z. B. **123456, 11111**).
   - PINs mit **vier identischen Ziffern** (z. B. **1111, 2222**).

a) KORREKT – 112, 1111, 1234, 123456
- **112** → Ungültig (weniger als vier Ziffern)
- **1111** → Ungültig (vier gleiche Ziffern)
- **1234** → Gültig (vier Ziffern, mindestens zwei verschieden)
- **123456** → Ungültig (mehr als vier Ziffern)

Diese Option ist korrekt, weil alle Äquivalenzklassen abgedeckt werden:

- **Weniger als vier Ziffern** (112)
- **Vier gleiche Ziffern** (1111)
- **Gültige PIN (vier Ziffern, mind. zwei verschieden)** (1234)
- **Mehr als vier Ziffern** (123456)

b) FALSCH – 1, 123, 1111, 1234
- **1** → Ungültig (weniger als vier Ziffern)
- **123** → Ungültig (weniger als vier Ziffern)
- **1111** → Ungültig (vier gleiche Ziffern)
- **1234** → Gültig

Hier fehlt die Äquivalenzklasse **"mehr als vier Ziffern"**.

c) FALSCH – 12, 112, 1112, 11112
- **12** → Ungültig (weniger als vier Ziffern)
- **112** → Ungültig (weniger als vier Ziffern)
- **1112** → Gültig (vier Ziffern, mind. zwei verschieden)
- **11112** → Ungültig (mehr als vier Ziffern)

Hier fehlt die Äquivalenzklasse **"vier gleiche Ziffern" (z. B. 1111)**.

d) FALSCH – 1, 111, 1111, 11111
- **1** → Ungültig (weniger als vier Ziffern)
- **111** → Ungültig (weniger als vier Ziffern)
- **1111** → Ungültig (vier gleiche Ziffern)
- **11111** → Ungültig (mehr als vier Ziffern)

Diese Option enthält **keine einzige gültige PIN**.
Die Äquivalenzklasse der gültigen PINs (z. B. 1234) fehlt vollständig.

Frage 21	FL-4.2.2	K3	Punkte 1.0

Gegeben ist die folgende Geschäftsregel:

EINGABE: Wert (Ganzzahl)

WENN (Wert ≤ 100 ODER Wert ≥ 200) DANN schreibe „Wert falsch“

ELSE schreibe „Wert OK“

Sie möchten Testfälle mit Hilfe einer Grenzwertanalyse mit jeweils zwei Werten pro Grenze entwerfen.

Welcher der folgenden Mengen von Testwerten deckt die relevanten Grenzwerte AM BESTEN ab?

Wählen Sie EINE Option! (1 aus 4)

a)	100, 150, 200, 201	☐
b)	99, 100, 200, 201	☐
c)	98, 99, 100, 101	☐
d)	101, 150, 199, 200	☒

FL-4.2.2 (K3) Der Lernende kann die Grenzwertanalyse zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.2.2):

Die Äquivalenzpartitionen sind: {…, 99, 100}, {101, 102, …, 198, 199}, {200, 201, …}.

Somit gibt es 4 Grenzwerte: 100, 101, 199 und 200.

Bei der 2-Werte-Grenzwertanalyse gibt es für jeden Grenzwert zwei Überdeckungselemente (den Grenzwert und seinen nächsten Nachbarn, der zur angrenzenden Partition gehört). Da die nächsten Nachbarn auch Grenzwerte in der angrenzenden Partition sind, gibt es nur vier Überdeckungselemente.

Daher:

a) FALSCH – Nur 100 und 200 sind gültige Überdeckungselemente für die Grenzwertanalyse mit zwei Werten, so dass wir eine Überdeckung von 50 % erreichen. Es fehlen die benachbarten Grenzwerte 101 und 199, wodurch nur 50 % der Grenzwerte abgedeckt werden (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.2).

b) FALSCH – Nur 100 und 200 sind gültige Überdeckungselemente für die Grenzwertanalyse mit zwei Werten. Die benachbarten Grenzwerte 101 und 199 werden nicht getestet, weshalb die Überdeckung ebenfalls nur 50 % beträgt (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.2).

c) FALSCH – Nur 100 und 101 sind gültige Überdeckungselemente für die Grenzwertanalyse mit zwei Werten, ignoriert werden jedoch 199 und 200. Damit werden ebenfalls nur 50 % der möglichen Grenzwerte abgedeckt (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.2).

d) KORREKT – Nur 101, 199 und 200 sind gültige Überdeckungselemente für die 2-Wert-Grenzwertanalyse, so dass wir eine Überdeckung von 75 % erreichen. Diese Option berücksichtigt die Anforderungen der 2-Wert-Grenzwertanalyse am besten (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.2).

Fazit:

Die Antwort d) ist korrekt, da sie die größte Überdeckung der Grenzwerte erreicht.

Die Antworten a), b) und c) sind falsch, da sie entweder zu viele unnötige Werte oder nicht alle relevanten Grenzwerte abdecken.

Frage 22	FL-4.2.3	K3	Punkte 1.0

Gegeben ist ein System zur Analyse von Fahrprüfungsergebnissen. Die folgenden Regeln wurden in einer Entscheidungstabelle formuliert:

	R1	R2	R3
C1: Erster Prüfungsversuch?	-	-	F
C2: Theoretische Prüfung bestanden?	T	F	-
C3: Praktische Prüfung bestanden?	T	-	F
Führerschein ausstellen?	X		
Zusätzliche Fahrstunden anfordern?			X
Antrag auf Wiederholung der Prüfung?		X	

Welche Kombination von Eingabedaten zeigt, dass die Entscheidungstabelle widersprüchliche Regeln enthält?

Wählen Sie EINE Option! (1 aus 4)

a)	C1 = T, C2 = T, C3 = F	☐
b)	C1 = T, C2 = F, C3 = T	☐
c)	C1 = T, C2 = T, C3 = T und C1 = F, C2 = T, C3 = T	☐
d)	C1 = F, C2 = F, C3 = F	☒

FL-4.2.3 (K3) Der Lernende kann den Entscheidungstabellentest zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.2.3):

a) FALSCH – Diese Kombination (T, T, F) entspricht keiner Regel (R1, R2 oder R3). Das zeigt, dass es sich um eine Auslassung handelt, aber nicht um einen Widerspruch (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.3, 3. Absatz).

b) FALSCH – Diese Kombination (T, F, T) entspricht nur der Regel R2, so dass kein Widerspruch zwischen mehreren Regeln besteht (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.3).

c) FALSCH – Beide Kombinationen (T, T, T) und (F, T, T) stimmen nur mit einer Spalte/Regel R1 überein. Es gibt daher keinen Widerspruch zwischen den Regeln (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.3).

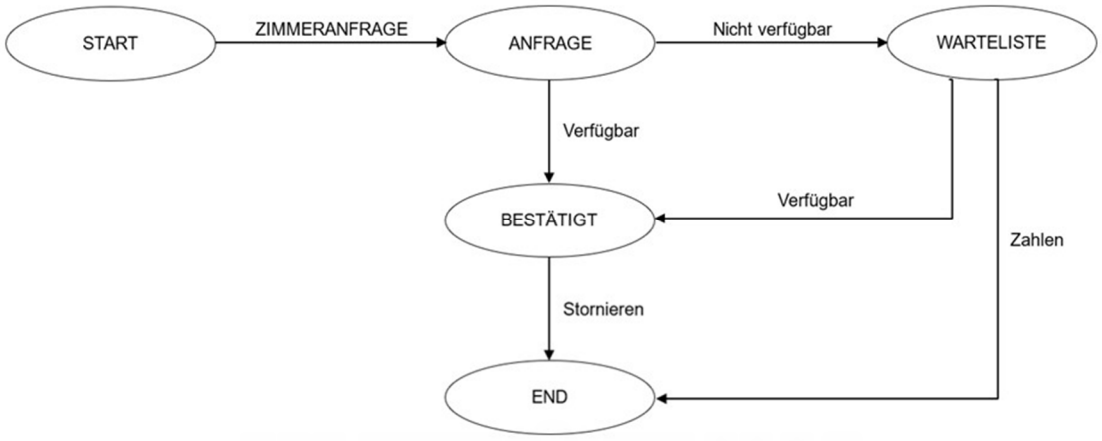
d) KORREKT – Diese Kombination (F, F, F) passt sowohl zu R2 als auch zu R3, aber R2 und R3 haben unterschiedliche Aktionen („Zusätzliche Fahrstunden anfordern“ und „Antrag auf Wiederholung der Prüfung“). Das zeigt einen Widerspruch zwischen den Regeln (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.3, letzter Absatz: „… Widersprüche in den Anforderungen …“).

Fazit:

Die Antwort d) ist korrekt, da diese Kombination einen Widerspruch zwischen den Regeln R2 und R3 in der Entscheidungstabelle zeigt.

Frage 23	FL-4.2.4	K3	Punkt 1.0

Sie entwerfen Testfälle auf der Grundlage des folgenden Zustandsübergangsdiagramm:



Wie viele Testfälle sind MINDESTENS erforderlich, um eine vollständige Überdeckung aller gültiger Übergänge zu erreichen?

Wählen Sie EINE Option! (1 aus 4)

a)	3	☒
b)	2	☐
c)	5	☐
d)	6	☐

FL-4.2.4 (K3) Der Lernende kann den Zustandsübergangstest zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.2.4):

Die folgenden drei Übergänge:

„ANFRAGE -> BESTÄTIGT“
„WARTELISTE -> BESTÄTIGT“
„WARTELISTE -> END“

können nicht im selben Testfall vorkommen, was darauf hindeutet, dass mindestens drei Testfälle erforderlich sind. Alle anderen Übergänge können in Kombination mit einem oder mehreren dieser drei Übergänge auftreten, daher benötigen wir mindestens drei Testfälle.

Tatsächlich sind nur drei Sequenzen möglich:

TF1: START (Zimmeranfrage) -> ANFRAGE (Verfügbar) -> BESTÄTIGT (Bezahlen) -> END
TF2: START (Zimmeranfrage) -> ANFRAGE (Nicht verfügbar) -> WARTELISTE (Verfügbar) -> BESTÄTIGT (Stornieren) -> END
TF3: START (Zimmeranfrage) -> ANFRAGE (Nicht verfügbar) -> WARTELISTE (Zahlen) -> END

Daher:

a) KORREKT

b) FALSCH

c) FALSCH

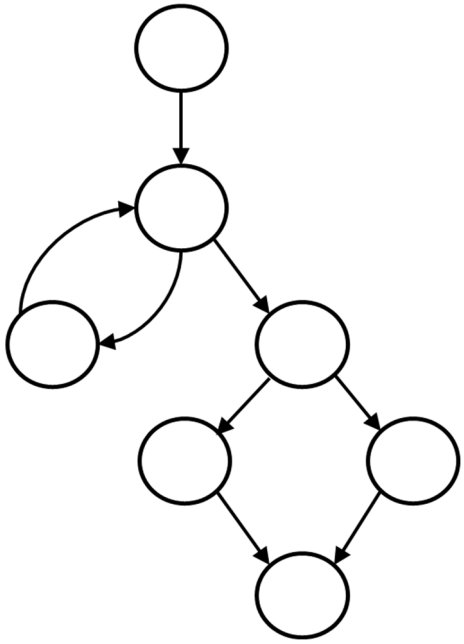
d) FALSCH

Fazit:

Die Antwort a) ist korrekt, da drei Testfälle ausreichen, um alle gültigen Übergänge im Zustandsübergangsdiagramm abzudecken.

Frage 24	FL-4.3.2	K2	Punkte 1.0

Sie sollen einen Verzweigungstest für den folgenden Programmcode durchführen, dessen Ablauf im Kontrollflussdiagramm dargestellt ist.



Wie viele Entscheidungen bzw. Zweige müssen getestet werden, um vollständige Verzweigungsüberdeckung zu erreichen?

Wählen Sie EINE Option! (1 aus 4)

a)	2	☐
b)	4	☐
c)	8	☒
d)	7	☐

FL-4.3.2 (K2) Der Lernende kann den Zweigtest erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.3.2):

Beim Zweigtest sind die Überdeckungselemente Zweige, die durch die Kanten eines Kontrollflussdiagramms dargestellt werden. Es gibt 8 Kanten im Kontrollflussdiagramm.

Daher:

a) FALSCH

b) FALSCH

c) KORREKT

d) FALSCH

Frage 25	FL-4.3.3	K2	Punkte 1.0

Wie können White-Box-Tests zur Verbesserung von Black-Box-Tests beitragen?

Wählen Sie EINE Option! (1 aus 4)

a)	White-Box-Überdeckungsanalysen helfen, die Codeabdeckung durch Black-Box-Tests zu bewerten.	☒
b)	White-Box-Analysen ermöglichen es, nicht erreichbare Teile des Codes zu identifizieren.	☐
c)	Zweigtests basieren auf Black-Box-Verfahren, daher deckt vollständige Zweigüberdeckung alle Black-Box-Tests ab.	☐
d)	White-Box-Verfahren liefern direkt anwendbare Testelemente für Black-Box-Tests.	☐

FL-4.3.3 (K2) Der Lernende kann den Wert des White-Box-Tests erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.3.3):

a) KORREKT – Die reine Durchführung von Black-Box-Tests liefert kein Maß für die tatsächliche Codeüberdeckung. White-Box-Überdeckungsmessungen bieten eine objektive Messung der Überdeckung und liefern die notwendigen Informationen, um die Generierung zusätzlicher Tests zu ermöglichen, um diese Überdeckung zu erhöhen und anschließend das Vertrauen in den Code zu stärken (siehe CTFL-Lehrplan V4.0, Abschnitt 4.3.3, 3. Absatz).

   - Black-Box-Tests basieren auf der Spezifikation und können nicht messen, wie viel Code tatsächlich getestet wurde.
   - White-Box-Tests liefern Codeüberdeckungsmetriken, die helfen, Lücken in Black-Box-Tests aufzudecken.
   - Durch die Kombination beider Testarten kann eine bessere Testabdeckung erreicht werden.

   Daraus folgt: Diese Option ist korrekt, da sie den Zusammenhang zwischen Codeüberdeckung (White Box) und funktionalen Tests (Black Box) richtig beschreibt.

b) FALSCH – Diese Aussage ist korrekt, hat aber nichts mit Black-Box-Tests zu tun (siehe CTFL-Lehrplan V4.0, Abschnitt 4.3.3).

c) FALSCH – Im Allgemeinen gibt es keine direkten (subsumierten) Beziehungen zwischen White-Box- und Black-Box-Verfahren (siehe CTFL-Lehrplan V4.0, Abschnitt 4.3.3).

d) FALSCH – White-Box-Testverfahren werden verwendet, um Tests basierend auf dem Testobjekt selbst zu entwerfen, während Black-Box-Testverfahren verwendet werden, um Tests auf Basis der Spezifikation zu entwerfen. Daher besteht kein Zusammenhang zwischen den aus diesen beiden Testverfahren abgeleiteten Überdeckungselementen (siehe CTFL-Lehrplan V4.0, Abschnitt 4.3.3).

Frage 26	FL-4.4.1	K2	Punkte 1.0

Ein Tester verwendet beim Testen die folgende Liste möglicher Fehlerzustände:

- **Korrekte Eingaben werden nicht akzeptiert**
- **Ungültige Eingaben werden akzeptiert**
- **Falsches Ausgabeformat**
- **Division durch null**

Welches Testverfahren wird dabei am WAHRSCHEINLICHSTEN eingesetzt?

Wählen Sie EINE Option! (1 aus 4)

a)	Exploratives Testen	☐
b)	Fehlerangriff	☒
c)	Checklistenbasiertes Testen	☐
d)	Grenzwertanalyse	☐

FL-4.4.1 (K2) Der Lernende kann die intuitive Testfallermittlung erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.4.1):

a) FALSCH – Beim explorativen Testen werden Test-Chartas verwendet, keine Liste möglicher Mängel/Ausfälle. Obwohl explorative Tests den Einsatz anderer Testverfahren beinhalten können, ist in diesem Fall ein Fehlerangriff die wahrscheinlichste Option (siehe CTFL-Lehrplan V4.0, Abschnitt 4.4.2, 2. Absatz: Beim explorativen Testen verwendet der Tester "eine Test-Charta mit Testzielen, um das Testen zu steuern").

b) KORREKT – Dies ist eine Liste möglicher Fehler. Fehlerangriffe sind ein methodischer Ansatz zur Implementierung der Fehlerschätzung und erfordern, dass der Tester eine Liste möglicher Fehler, Defekte und Ausfälle erstellt oder erfasst und Tests entwirft, die mit den Fehlern verbundene Defekte identifizieren, die Defekte aufdecken oder die Fehler verursachen Misserfolge (siehe CTFL-Lehrplan V4.0, Abschnitt 4.4.1, 3. Absatz: Fehlerangriffe erfordern, dass "Tester eine Liste möglicher Fehlhandlungen, Fehlerzustände und Fehlerwirkungen erstellen ...").

c) FALSCH – Der Tester verwendet eine Checkliste mit Elementen, um seine Tests zu unterstützen. Sowohl beim Erraten von Fehlern als auch beim checklistenbasierten Testen werden solche Listen verwendet. Allerdings handelt es sich hier um eine Liste möglicher Fehler und nicht um Testbedingungen. Das WAHRSCHEINLICHSTE Testverfahren ist daher ein Fehlerangriff, der sich auf Fehler, Defekte und Ausfälle konzentriert (siehe CTFL-Lehrplan V4.0, Abschnitt 4.4.3, 1. Absatz: "Beim checklistenbasierten Test entwirft, implementiert und führt ein Tester Tests aus, um Testbedingungen aus einer Checkliste abzudecken.").

d) FALSCH – Die Grenzwertanalyse basiert auf einer Analyse der Grenzwerte von Äquivalenzklassen. In der obigen Liste werden Äquivalenzklassen oder ihre Grenzen nicht erwähnt (siehe CTFL-Lehrplan V4.0, Abschnitt 4.2.2, 1. Absatz: "Die Grenzwertanalyse ist ein Testverfahren, das auf der Überprüfung der Grenzen von Äquivalenzklassen basiert.").

Die Antwort b) ist korrekt, da die beschriebene Liste von Fehlerarten typisch für das Verfahren des Fehlerangriffs ist.

Frage 27	FL-4.4.3	K2	Punkte 1.0

Welche der folgenden Aussagen beschreibt AM BESTEN, wie der Einsatz checklistenbasiertes Testen zu höherer Testüberdeckung führen kann?

Wählen Sie EINE Option! (1 aus 4)

a)	Checklistenelemente können grob definiert werden, sodass Tester daraus eigenständig detaillierte Testfälle ableiten und ausführen.	☐
b)	Durch die Automatisierung von Checklistenpunkten wird bei jeder Testausführung automatisch eine höhere Überdeckung erzielt.	☐
c)	Jeder Checklistenpunkt sollte einzeln getestet werden, damit verschiedene Bereiche der Software abgedeckt werden.	☐
d)	Zwei Tester, die dieselben Checklistenpunkte nutzen, setzen sie meist unterschiedlich um – das führt zu ergänzender Überdeckung.	☒

FL-4.4.3 (K2) Der Lernende kann den checklistenbasierten Test erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.4.3):

a) FALSCH – Es stimmt zwar, dass der Tester anhand der Checkliste detaillierte Testfälle implementieren und ausführen kann, es wird jedoch nicht erklärt, wie dies zu einer erhöhten Überdeckung führen würde.

b) FALSCH – “Checklisten sollten keine Elemente enthalten, die automatisch geprüft werden können" (siehe CTFL-Lehrplan V4.0, Abschnitt 4.4.3, 3. Satz), d. h., Checklistenelemente sollten nicht automatisiert werden. Aber selbst, wenn dies der Fall ist, führen die automatisierten Testskripte die Tests immer auf die gleiche Weise aus, was in der Regel keine erhöhte Überdeckung zur Folge hat.

c) FALSCH – Es stimmt, dass jeder Checklistenpunkt separat und unabhängig getestet werden sollte. Dies wirkt sich jedoch auf die Testausführungsreihenfolge aus und hat keinen Einfluss auf die erreichte Überdeckung und führt daher nicht zu einer erhöhten Überdeckung.

d) KORREKT – Wenn es sich bei den Checklisten um ein hohes Niveau handelt, ist es wahrscheinlich, dass bei den tatsächlichen Tests eine gewisse Variabilität auftritt, was möglicherweise zu einer größeren Überdeckung, aber einer geringeren Wiederholbarkeit führt. Wenn zwei Tester einer Checkliste mit hochrangigen Elementen folgen, kann jeder von ihnen unterschiedliche Testdaten, Testschritte usw. verwenden. Auf diese Weise wird ein Tester wahrscheinlich einige Bereiche abdecken, die der andere Tester nicht abdeckt, und dies führt zu einer erhöhten Überdeckung (siehe CTFL-Lehrplan V4.0, Abschnitt 4.4.3, letzter Satz).

Frage 28	FL-4.5.2	K2	Punkte 1.0

Welches der folgenden Akzeptanzkriterien (Abnahmekriterien) ist das typische Beispiel für ein szenarioorientiertes Kriterium?

Wählen Sie EINE Option! (1 aus 4)

a)	Die Anwendung muss es Nutzenden ermöglichen, ihr Konto sowie alle zugehörigen Daten auf Anfrage zu löschen.	☐
b)	Wenn ein Kunde einen Artikel in den Warenkorb legt und zur Kasse geht, soll er aufgefordert werden, sich anzumelden oder ein Konto zu erstellen, sofern dies noch nicht geschehen ist.	☒
c)	IF (contain(product(23).Name, cart.products())) THEN return FALSE.	☐
d)	Die Website muss den Anforderungen der ICT Accessibility 508 Standards entsprechen und sicherstellen, dass alle Inhalte für Menschen mit Behinderungen zugänglich sind.	☐

FL-4.5.2 (K2) Der Lernende kann die verschiedenen Möglichkeiten zum Schreiben von Akzeptanzkriterien einordnen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.5.2):

a) FALSCH – Dieses Akzeptanzkriterium beschreibt, welche Regeln oder Vorschriften das System einhalten muss (in diesem Fall das Recht auf Vergessen werden). Dies ist ein Beispiel für ein regelorientiertes Akzeptanzkriterium (siehe CTFL-Lehrplan V4.0, Abschnitt 4.5.2).

b) KORREKT – Dieses Akzeptanzkriterium beschreibt ein Beispielszenario, das vom System realisierbar sein muss. Dies ist ein Beispiel für ein szenarioorientiertes Akzeptanzkriterium (siehe CTFL-Lehrplan V4.0, Abschnitt 4.5.2). Laut ISTQB-Lehrplan sind szenarioorientierte Akzeptanzkriterien typischerweise in einem "Gegeben/Wenn/Dann"-Format beschrieben und orientieren sich an der verhaltensgetriebenen Entwicklung (BDD).

c) FALSCH – Dieser Satz ähnelt eher einer Codezeile, die eine Geschäftsregel implementiert. Akzeptanzkriterien sollten in Zusammenarbeit mit Unternehmensvertretern verfasst werden und daher in einer Sprache formuliert sein, die sie verstehen. Dieser Satz wird für diese Stakeholder höchstwahrscheinlich unverständlich sein (siehe CTFL-Lehrplan V4.0, Abschnitt 4.5.2).

d) FALSCH – Dieses Akzeptanzkriterium beschreibt, welche Regeln oder Vorschriften das System einhalten muss und wie die Einhaltung sichergestellt wird. Daher handelt es sich hier um ein Beispiel für ein regelorientiertes Akzeptanzkriterium und nicht um ein szenarioorientiertes Akzeptanzkriterium (siehe CTFL-Lehrplan V4.0, Abschnitt 4.5.2).

Fazit:

- **Antwort b) ist korrekt**, da sie ein konkretes Nutzungsszenario beschreibt, das überprüft werden kann.
- **Antworten a) und d) sind falsch**, da sie regelorientierte Kriterien sind.
- **Antwort c) ist falsch**, da sie eine technische Implementierung darstellt, die für Stakeholder nicht verständlich ist.

Frage 29	FL-4.5.3	K3	Punkte 1.0

Sie arbeiten mit abnahmetestgetriebener Entwicklung und entwerfen Testfälle auf Grundlage der folgenden User-Story:

> ***Als regulärer oder spezieller Benutzer möchte ich meine elektronische Etagenkarte nutzen können, um Zugang zu bestimmten Etagen zu erhalten.***

Akzeptanzkriterien:

AC1: Reguläre Benutzer haben Zugang zu den Etagen 1 bis 3.

AC2: Etage 4 ist ausschließlich für spezielle Benutzer zugänglich.

AC3: Spezielle Benutzer verfügen über alle Zugriffsrechte regulärer Benutzer.

Welcher Testfall eignet sich AM BESTEN, um das Akzeptanzkriterium AC3 zu testen?

Wählen Sie EINE Option! (1 aus 4)

a)	Ein regulärer Benutzer erhält Zugang zu den Etagen 1 und 3.	☐
b)	Ein regulärer Benutzer hat keinen Zugang zur Etage 4.	☐
c)	Ein spezieller Benutzer versucht, auf Etage 5 zuzugreifen.	☐
d)	Ein spezieller Benutzer erhält Zugang zu den Etagen 1, 2 und 3.	☒

FL-4.5.3 (K3) Der Lernende kann die abnahmetestgetriebene Entwicklung (ATDD) zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 4.5.3):

a) FALSCH – Wir möchten überprüfen, ob spezielle Benutzer die Rechte von regulären Benutzern haben. Daher müssen wir die Zugriffsrechte für einen speziellen Benutzer testen, nicht für einen regulären Benutzer.

b) FALSCH – Wir möchten überprüfen, ob spezielle Benutzer die Rechte von regulären Benutzern haben. Daher müssen wir die Zugriffsrechte für einen speziellen Benutzer testen, nicht für einen regulären Benutzer.

c) FALSCH – In den Akzeptanzkriterien ist keine Etage 5 beschrieben. Die Testfälle sollten den Umfang der User-Story nicht erweitern. Selbst wenn wir gerne einen Negativtest durchführen würden, steht dieser Test nicht in direktem Zusammenhang mit AC3.

d) KORREKT – Auf diese Weise können wir prüfen, ob ein spezieller Benutzer Zugriff auf Etagen hat, die für einen regulären Benutzer zugänglich sind. Die korrekte Option ist **d)**, da dieser Testfall spezifisch die Zugriffsrechte eines speziellen Benutzers auf die regulären Etagen überprüft, wie es AC3 verlangt.

Frage 30	FL-5.1.1	K1	Punkte 1.0

Welcher der folgenden Punkte gehört NICHT zum Zweck eines Testkonzepts?

Wählen Sie EINE Option! (1 aus 4)

a)	Definition konkreter Testdaten und erwarteter Ergebnisse für Komponenten- und Integrationstests	☒
b)	Festlegung eines Endekriteriums auf Komponententestebene mit dem Ziel 100 % Anweisungs- und Zweigüberdeckung	☐
c)	Beschreibung der Inhalte und Struktur von Testfortschrittsberichten	☐
d)	Begründung, warum eine in der Teststrategie vorgesehene Teststufe (z. B. Systemintegration) nicht durchgeführt wird	☐

FL-5.1.1 (K2) Der Lernende kann Beispiele zu Zweck und Inhalt eines Testkonzepts geben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.1.1):

a) KORREKT – Das Testkonzept kann Testdatenanforderungen (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.1, Typische Inhalte eines Testkonzepts, 6. Aufzählungspunkt) als Teil des Testansatzes enthalten, jedoch nicht die detaillierten Testdaten für Testfälle. Testdaten sind Teil der Testfälle, nicht des Testkonzepts. Außerdem ist es in der Regel nicht möglich, solche Daten bei der Testplanerstellung zu definieren, da nicht genau bekannt ist, wie die Komponenten aussehen werden (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.1).

b) FALSCH – Einer der Zwecke eines Testkonzepts besteht darin, sicherzustellen, dass die durchgeführten Testaktivitäten die festgelegten Kriterien erfüllen, indem es Eingangs- und Endekriterien einbezieht. Ein Beispiel für solche Kriterien für die Komponententestebene sind die Codeüberdeckungskriterien (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.1, Typische Inhalte eines Testkonzepts, 6. Aufzählungspunkt).

c) FALSCH – Dokumentationsvorlagen sind typische Inhalte eines Testkonzepts. Dies trägt dazu bei, die Kommunikation zwischen den Beteiligten zu erleichtern, indem eine Standardmethode für die Kommunikation oder Berichterstattung definiert wird (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.1, Typische Inhalte eines Testkonzepts, 4. Aufzählungspunkt).

d) FALSCH – Einer der Zwecke eines Testkonzepts besteht darin, nachzuweisen, dass sich die Tests an die bestehende Testrichtlinie und Teststrategie halten, oder zu erklären, warum die Tests davon abweichen. Dies ist ein Beispiel zur Erläuterung der Abweichung hinsichtlich der Teststufen, die befolgt werden (oder nicht befolgt werden) (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.1, Typische Inhalte eines Testkonzepts, 2. Aufzählungspunkt).

Frage 31	FL-5.1.4	K3	Punkte 1.0

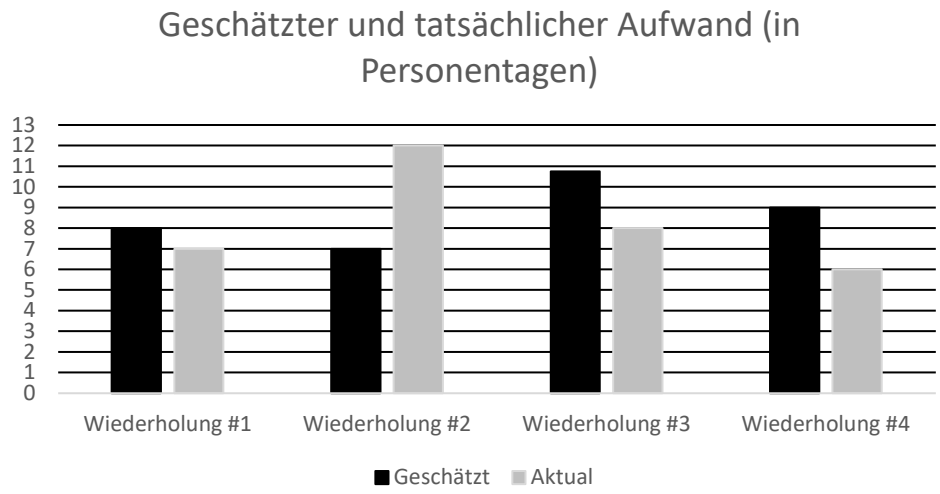
Zu Beginn jeder Iteration schätzt das Team den Arbeitsaufwand (in Personentagen), den es während dieser Iteration leisten wird.

Bezeichnet sei E(n) **als der geschätzte und** A(n) **als der tatsächlich geleistete Aufwand in Iteration n.**

Ab der dritten Iteration verwendet das Team folgendes Extrapolationsmodell zur Aufwandsschätzung:

$$\frac{3 * A(n-1) + A(n-2)}{4}$$

Eine Grafik zeigt die geschätzten und tatsächlichen Werte für die ersten vier Iterationen.



Wie hoch ist der geschätzte Arbeitsaufwand für Iteration Nr. 5?

Wählen Sie EINE Option! (1 aus 4)

a)	10,5 Personentage	☐
b)	8,25 Personentage	☐
c)	6,5 Personentage	☒
d)	9,4 Personentage	☐

FL-5.1.4 (K3) Der Lernende kann Schätzverfahren zur Berechnung des erforderlichen Testaufwands anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.1.4):

Die Prüfungsfrage basiert auf der **Extrapolation**, einem Verfahren zur Schätzung des Testaufwands. Laut dem Lehrplan wird bei der Extrapolation der für die verbleibende Arbeit erforderliche Aufwand durch Anwendung eines mathematischen Modells auf historische Daten angenähert.

Aus der Grafik ergibt sich:

A(4)=6 und A(3)=8 (die letzten beiden grauen Kästchen)

Aus der Formel erhalten wir:

E(5) = (3*A(4) + A(3)) / 4 = (3*6+8) / 4 = 26 / 4 = 6,5 Personentage

Daher:

a) FALSCH

b) FALSCH

c) KORREKT

d) FALSCH

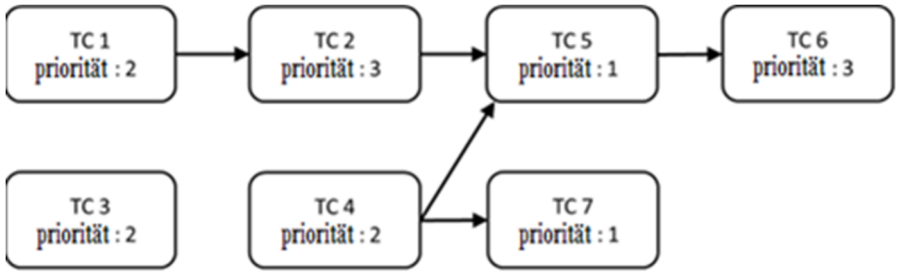
Die Antwort c) ist korrekt, da die Schätzung von 6,5 Personentagen mit der Formel und den historischen Daten übereinstimmt.

Frage 32	FL-5.1.5	K3	Punkte 1.0

Sie erstellen einen Testausführungsplan für sieben Testfälle (TC 1 bis TC 7).

Eine Abbildung zeigt:
– Die Priorität jedes Testfalls (1 = höchste, 3 = niedrigste)
– Abhängigkeiten zwischen den Testfällen in Form gerichteter Pfeile.

Ein Testfall darf nur ausgeführt werden, wenn alle vorgelagerten (abhängigen) Testfälle bereits ausgeführt wurden.



Welcher Testfall sollte gemäß diesen Vorgabe als sechster ausgeführt werden?

Wählen Sie EINE Option! (1 aus 4)

a)	TC 3	☒
b)	TC 5	☐
c)	TC 6	☐
d)	TC 2	☐

FL-5.1.5 (K3) Der Lernende kann die Priorisierung von Testfällen anwenden.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.1.5):

Wir wollen Testfälle entsprechend ihren Prioritäten ausführen, müssen aber auch die Abhängigkeiten berücksichtigen.

Wenn wir nur Prioritäten berücksichtigen, möchten wir zuerst TC 5 und TC 7 (höchste Priorität), dann TC 1, TC 3 und TC 4 und schließlich TC 2 und TC 6 (niedrigste Priorität) ausführen.

Um jedoch TC 7 auszuführen, müssen wir zuerst TC 4 ausführen.

Um TC 5 auszuführen, müssen wir TC 4 und TC 2 ausführen, aber TC 2 wird durch TC 1 blockiert, das vor TC 2 ausgeführt werden sollte.

Um also Testfälle der Priorität 1 so früh wie möglich auszuführen, sollten die ersten fünf Testfälle sein: TC 4 – TC 7 – TC 1 – TC 2 – TC 5.

Als Nächstes müssen wir TC 3 ausführen, da es eine höhere Priorität als TC 6 hat.

Somit wird der vollständige Testausführungsplan TC 4 – TC 7 – TC 1 – TC 2 – TC 5 – TC 3 – TC 6 sein.

Der sechste Testfall wird also TC 3 sein.

Daher:

a) KORREKT

b) FALSCH

c) FALSCH

d) FALSCH

Frage 33	FL-5.1.6	K1	Punkte 1.0

Was zeigt das Testpyramidenmodell?

Wählen Sie EINE Option! (1 aus 4)

a)	Tests können unterschiedliche Prioritäten haben.	☐
b)	Tests haben unterschiedliche Granularität.	☒
c)	Für verschiedene Tests gelten unterschiedliche Überdeckungskriterien.	☐
d)	Tests können voneinander abhängig sein.	☐

FL-5.1.6 (K1) Der Lernende kann die Konzepte der Testpyramide wiedergeben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.1.6):

a) FALSCH – Das Testpyramidenmodell beschreibt keine Testprioritäten. Es konzentriert sich auf die Hierarchie und Granularität von Tests, nicht auf ihre Priorisierung (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.6).

b) KORREKT – Das Testpyramidenmodell zeigt, dass Tests in unterschiedlichen Ebenen mit unterschiedlicher Granularität ausgeführt werden. Es illustriert die Granularität von Tests, z. B. von feinkörnigen Unit-Tests bis zu grobkörnigeren End-to-End-Tests (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.6, 1. Satz).

c) FALSCH – Überdeckungskriterien sind nicht Teil des Fokus des Testpyramidenmodells. Dieses Modell beschäftigt sich mit Testebenen und Granularität, nicht mit den Überdeckungskriterien der Tests (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.6).

d) FALSCH – Das Testpyramidenmodell beschreibt keine Abhängigkeiten zwischen Tests. Es konzentriert sich auf die Struktur und Hierarchie von Testebenen (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.6).

Frage 34	FL-5.1.7	K2	Punkte 1.0

Welche Beziehung besteht zwischen den Testquadranten, Teststufen und Testarten?

Wählen Sie EINE Option! (1 aus 4)

a)	Testquadranten kombinieren bestimmte Teststufen mit Testarten und ordnen sie einem festen Abschnitt im Softwareentwicklungslebenszyklus zu.	☐
b)	Testquadranten definieren die Granularität von Testarten, die auf unterschiedlichen Teststufen angewendet werden.	☐
c)	Testquadranten ordnen jeder Teststufe eine spezifische Menge an Testarten zu.	☐
d)	Testquadranten gruppieren Teststufen und Testarten nach Kriterien wie der Zielgruppe oder dem Testfokus.	☒

FL-5.1.7 (K2) Der Lernende kann die Testquadranten und ihre Beziehungen zu Teststufen und Testarten zusammenfassen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.1.7):

a) FALSCH – Testquadranten gruppieren Teststufen und Testarten getrennt nach mehreren Kriterien. Sie stellen keine Kombinationen von Teststufen und Testarten dar und haben keinen Bezug zu einem Ort innerhalb eines Softwareentwicklungslebenszyklus. Sowohl Teststufen als auch Testarten werden im Testquadrantenmodell separat behandelt (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.7).

b) FALSCH – Testquadranten gruppieren Teststufen und Testarten nach mehreren Kriterien. Sie beschreiben nicht den Grad der Granularität der einzelnen Testarten, die auf jeder Teststufe durchgeführt werden. Ein solches Modell hinsichtlich der Teststufen wird als Testpyramide bezeichnet (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.7).

c) FALSCH – Die Aussage ist falsch, da grundsätzlich jede Testart auf jeder Teststufe durchgeführt werden kann (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.7).

d) KORREKT – Die Testquadranten gruppieren Teststufen, Testarten, Aktivitäten, Testverfahren und Arbeitsprodukte in der agilen Softwareentwicklung. In diesem Modell können Tests geschäftsorientiert oder technologieorientiert sein. Tests können das Team unterstützen (d. h. die Entwicklung leiten) oder das Produkt kritisieren (d. h. sein Verhalten anhand der Erwartungen messen). Die Kombination dieser beiden Standpunkte bestimmt die vier Quadranten (siehe CTFL-Lehrplan V4.0, Abschnitt 5.1.7, 1. Satz).

Frage 35	FL-5.2.3	K2	Punkte 1.0

Wie kann eine Produktrisikoanalyse den Umfang und die Gründlichkeit von Tests beeinflussen?

Wählen Sie EINE Option! (1 aus 4)

a)	Durch kontinuierliche Risikoüberwachung können neue Risiken frühzeitig erkannt werden.	☐
b)	Die Risikoidentifikation unterstützt die Planung risikomindernder Maßnahmen.	☐
c)	Das bewertete Risikoniveau bestimmt die erforderliche Testtiefe und -überdeckung.	☒
d)	Die Risikoanalyse liefert die Grundlage für die Auswahl technischer Testobjekte.	☐

FL-5.2.3 (K2) Der Lernende kann den möglichen Einfluss der Produktrisikoanalyse auf Gründlichkeit und Umfang des Testens erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.2.3):

a) FALSCH – Die Risikoüberwachung ist Teil der Risikosteuerung, nicht der Risikoanalyse. Sie dient der Identifizierung neuer Risiken oder der Überprüfung bereits erkannter Risiken während des Testprozesses.

b) FALSCH – Die Risikoidentifizierung selbst ermöglicht es uns nicht, Maßnahmen zur Risikominderung umzusetzen. Die Abhilfemaßnahmen bzw. Maßnahmen zur Risikominderung werden in der Risikosteuerung definiert.

c) KORREKT – Die Risikoanalyse bewertet die Risiken nach Wahrscheinlichkeit und Auswirkung. Diese Bewertung hilft dabei, die Gründlichkeit und den Umfang der Tests für verschiedene Testobjekte festzulegen. Höhere Risiken erfordern gründlichere Tests.

d) FALSCH – Überdeckungselemente werden durch Testverfahren (z. B. Äquivalenzklassen oder Grenzwertanalyse) definiert, nicht durch die Risikoanalyse.

Frage 36	FL-5.3.2	K2	Punkte 1.0

Bei welcher der folgenden Aktivitäten im Testprozess werden Testfortschrittsberichte AM HÄUFIGSTEN verwendet?

Wählen Sie EINE Option! (1 aus 4)

a)	Testentwurf	☐
b)	Testabschluss	☒
c)	Testanalyse	☐
d)	Testplanung	☐

FL-5.3.2 (K2) Der Lernende kann Zweck, Inhalt und Zielgruppen von Testberichten zusammenfassen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.3.2):

a) FALSCH – Testfortschrittsberichte werden hauptsächlich während der Testüberwachung und Teststeuerung sowie beim Testabschluss verwendet, nicht während des Testentwurfs (siehe CTFL-Lehrplan V4.0, Abschnitt 5.3.2).

b) KORREKT – Ein Testabschlussbericht wird während des Testabschlusses erstellt, wenn ein Projekt, eine Teststufe oder eine Testart abgeschlossen ist und wenn im Idealfall die Abschlusskriterien erfüllt sind. Dieser Bericht verwendet Informationen aus Testfortschrittsberichten und anderen Daten (siehe CTFL-Lehrplan V4.0, Abschnitt 5.3.2, 3. Absatz).

c) FALSCH – Testfortschrittsberichte werden hauptsächlich während der Testüberwachung und Teststeuerung sowie beim Testabschluss verwendet, nicht während der Testanalyse (siehe CTFL-Lehrplan V4.0, Abschnitt 5.3.2).

d) FALSCH – Testfortschrittsberichte sind ein Werkzeug zur Steuerung und Überwachung der Tests. Sie werden nicht während der Testplanung verwendet, da diese Berichte zu diesem Zeitpunkt noch nicht existieren (siehe CTFL-Lehrplan V4.0, Abschnitt 5.3.2).

Frage 37	FL-5.4.1	K2	Punkte 1.0

Welche der folgenden Aussagen stellt KEIN Beispiel dafür dar, wie das Konfigurationsmanagement das Testen unterstützt?

Wählen Sie EINE Option! (1 aus 4)

a)	Alle Änderungen im Repository werden eindeutig identifiziert und versioniert.	☐
b)	Änderungen an der Testumgebung werden dokumentiert und nachvollziehbar gemacht.	☐
c)	Anforderungen sind in Testdokumenten eindeutig referenziert.	☐
d)	Fehlerberichte sind mit einem Bearbeitungsstatus versehen.	☒

FL-5.4.1 (K2) Der Lernende kann eine mögliche Unterstützung des Testens durch das Konfigurationsmanagement zusammenfassen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.4):

a) FALSCH – Das Konfigurationsmanagement stellt sicher, dass alle Konfigurationselemente, einschließlich Testelemente und Softwarekomponenten, eindeutig identifiziert, versionskontrolliert und nachverfolgt werden. Dies erleichtert die Reproduzierbarkeit von Testergebnissen, indem man zu einer bestimmten Version zurückkehren kann (siehe CTFL-Lehrplan V4.0, Abschnitt 5.4).

b) FALSCH – Das Konfigurationsmanagement verwaltet Änderungen an Testumgebungen und ermöglicht es, frühere Versionen einer Umgebung wiederherzustellen, falls eine neue Änderung zu unerwarteten Problemen führt. Dadurch können Tester konsistente Testbedingungen gewährleisten und Regressionen vermeiden (siehe CTFL-Lehrplan V4.0, Abschnitt 5.4).

c) FALSCH – Das Konfigurationsmanagement stellt sicher, dass alle identifizierten Dokumentationen (z. B. Anforderungsspezifikationen) und Softwareelemente in der Testdokumentation (z. B. Testkonzepte) eindeutig referenziert werden (siehe CTFL-Lehrplan V4.0, Abschnitt 5.4).

d) KORREKT – Das Zuweisen eines Status zu Fehlerzuständen ist eine Aufgabe des Fehlermanagements (siehe CTFL-Lehrplan V4.0, Abschnitt 5.5), nicht des Konfigurationsmanagements. Das Konfigurationsmanagement konzentriert sich auf die Kontrolle und Verfolgbarkeit von Artefakten und Änderungen (siehe CTFL-Lehrplan V4.0, Abschnitt 5.4).

Frage 38	FL-5.5.1	K3	Punkte 1.0

Betrachten Sie den folgenden Fehlerbericht für eine webbasierte Einkaufsanwendung:

Anwendung: WebShop v0.99

Fehler: Login-Button funktioniert nicht

Schritte zum Reproduzieren:

Starten Sie die Website

Klicken Sie auf den Login-Button

Erwartetes Ergebnis: Der Benutzer wird zur Anmeldeseite weitergeleitet.
Tatsächliches Ergebnis: Die Anmeldeschaltfläche reagiert nicht.

Schweregrad: Hoch

Priorität: Dringend

Welche wichtigen Informationen fehlen in diesem Bericht?

Wählen Sie EINE Option! (1 aus 4)

a)	Name des Testers und Datum des Berichts.	☐
b)	Angaben zur Testumgebung mit Versionsinformationen.	☒
c)	Identifizierung des Testobjekts.	☐
d)	Beschreibung der Auswirkungen für betroffene Stakeholder.	☐

FL-5.5.1 (K3) Der Lernende kann einen Fehlerbericht erstellen.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 5.5):

a) FALSCH – Diese Informationen sind zwar hilfreich, aber weniger entscheidend als Details zur Testumgebung, da diese für die Fehlerreproduktion und -behebung wichtiger sind (siehe CTFL-Lehrplan V4.0, Abschnitt 5.5, 3. Absatz, 3. Aufzählungspunkt).

b) KORREKT – Angaben zur Testumgebung (z. B. Browser, Betriebssystem, Softwareversion) sind essenziell, um Fehler zu reproduzieren und zu beheben. Ohne diese Informationen haben Entwickler Schwierigkeiten, das Problem in derselben Umgebung nachzustellen und die Ursache zu identifizieren (siehe CTFL-Lehrplan V4.0, Abschnitt 5.5, 3. Absatz, 4. Aufzählungspunkt).

c) FALSCH – Das Testobjekt wurde im Bericht klar identifiziert (WebShop v0.99). Diese Information ist vorhanden und kein fehlender Bestandteil (siehe CTFL-Lehrplan V4.0, Abschnitt 5.5, 3. Absatz, 4. Aufzählungspunkt).

d) FALSCH – Die Auswirkungen wurden im Bericht angegeben, und zwar durch die Bewertung des Schweregrads ("hoch") und die Priorität („dringend“). Diese Angaben geben bereits eine Einschätzung der Relevanz für Stakeholder, so dass diese Information nicht fehlt (siehe CTFL-Lehrplan V4.0, Abschnitt 5.5, 3. Absatz, 8. und 9. Aufzählungspunkt).

Frage 39	FL-6.1.1	K2	Punkte 1.0

Welche der folgenden Werkzeugkategorien unterstützt die Verwaltung von Testfällen, Fehlerberichten und Konfigurationsdaten? Wählen Sie EINE Option! (1 aus 4)

a)	Werkzeuge für Testdurchführung und Testüberdeckung.	☐
b)	Werkzeuge für Testentwurf und Testrealisierung.	☐
c)	Fehlermanagementwerkzeuge	☐
d)	Testmanagementwerkzeuge	☒

FL-6.1.1 (K2) Der Lernende kann eine mögliche Unterstützung des Testens durch verschiedene Arten von Testwerkzeugen erklären.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 6.1):

a) FALSCH – Testdurchführungswerkzeuge und Testüberdeckungswerkzeuge dienen der automatisierten Ausführung von Testfällen sowie der Messung der Testabdeckung (z. B. Codeabdeckung). Diese Werkzeuge unterstützen jedoch nicht direkt das Fehlermanagement oder das Konfigurationsmanagement, da sie sich primär auf die Testdurchführung und Metriken konzentrieren (siehe CTFL-Lehrplan V4.0, Abschnitt 6.1).

b) FALSCH – Testentwurfs- und Testrealisierungswerkzeuge erleichtern die Erstellung von Testfällen, Testdaten und Testskripten (siehe CTFL-Lehrplan V4.0, Abschnitt 6.1, 3. Aufzählungspunkt), unterstützen jedoch nicht beim Fehlermanagement und beim Konfigurationsmanagement.

c) FALSCH – Fehlermanagementwerkzeuge sind speziell für das Erfassen, Verfolgen und Verwalten von Fehlerberichten konzipiert. Sie unterstützen nicht bei der Organisation von Testfällen oder beim Konfigurationsmanagement (siehe CTFL-Lehrplan V4.0, Abschnitt 6.1).

d) KORREKT – Testmanagementwerkzeuge erhöhen die Effizienz des Testprozesses, indem sie die Verwaltung des Softwareentwicklungslebenszyklus (SDLC), der Anforderungen, Tests, Fehler und der Konfiguration erleichtern (siehe CTFL-Lehrplan V4.0, Abschnitt 6.1, 1. Aufzählungspunkt). Testmanagementwerkzeuge sind speziell darauf ausgelegt, den gesamten Testprozess zu unterstützen. Sie ermöglichen die Verwaltung von

- Testfällen (Erstellung, Organisation, Verknüpfung von Anforderungen),
- Fehlern (Verknüpfung mit Tests, Priorisierung, Statusverfolgung),
- Konfigurationen (Versionierung von Testartefakten, Änderungsverfolgung).

Frage 40	FL-6.2.1	K1	Punkte 1.0

Welcher der folgenden Punkte ist ein typischer Vorteil der Testautomatisierung?

Wählen Sie EINE Option! (1 aus 4)

a)	Automatisierte Generierung von Testfällen ohne Bezug zur Testbasis	☐
b)	Erhöhte Überdeckung durch objektive Bewertung von Testergebnissen	☐
c)	Längere Testlaufzeiten durch höhere Verarbeitungsleistung	☐
d)	Reduzierung menschlicher Fehler durch konsistente und wiederholbare Testausführung	☒

FL-6.2.1 (K1) Der Lernende kann die Nutzen und Risiken von Testautomatisierung wiedergeben.

Begründung (siehe ISTQB®-Lehrplan CTFL V4.0, Abschnitt 6.2):

a) FALSCH – Testfälle ohne Zugriff auf die Testbasis zu generieren, ist nicht möglich. Die Generierung von Testfällen durch Tester oder Werkzeuge erfordert Zugriff auf die Testbasis.

b) FALSCH – Testautomatisierung ermöglicht eine objektivere Bewertung der Überdeckung, trägt aber nicht direkt zur Erhöhung der Überdeckung bei. Die Überdeckung kann nur durch Schreiben und Ausführen zusätzlicher Testfälle erhöht werden.

c) FALSCH – Diese Aussage ist widersprüchlich. Höhere Verarbeitungsleistung reduziert typischerweise die Testausführungszeiten, was einen Vorteil darstellt, nicht die Erhöhung der Ausführungszeiten.

d) KORREKT – Ein entscheidender Vorteil der Testautomatisierung ist die Konsistenz und Wiederholbarkeit der Tests, die menschliche Fehler minimiert. Automatisierte Tests werden immer mit derselben Genauigkeit und in der gleichen Reihenfolge ausgeführt (siehe CTFL-Lehrplan V4.0, Abschnitt 6.2, 2. Absatz, 2. Aufzählungspunkt).

Platz für Ihre Notizen:

(Sie werden bei der Korrektur weder gelesen noch bewertet)

Platz für Ihre Notizen:

(Sie werden bei der Korrektur weder gelesen noch bewertet)