Familienname, Vorname: _____

Firmenadresse: _____

_____

_____

Telefon: _____

Fax: _____

E-Mail-Adresse: _____

Rechnungsanschrift: _____

_____

_____

Schulungsunternehmen: _____

Referent: _____

Foundation Level Probeprüfung

SET C (v2.1) – GTB Edition –

CTFL Syllabus Version v4.0

ISTQB® Certified Tester Foundation Level

Legal

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, (technical reviewer), Meile Posthuma, Gandhinee Rajkomar, Stuart Reid, Jean-François Riverin, Adam Roman, Lucjan Stapp, Stephanie Ulrich, Yaron Tsubery und Eshraka Zakaria.

Das Kernteam dankt dem Review-Team: Amanda Alderman, Alexander Alexandrov, Jürgen Beniermann, Rex Black, Young jae Choi, Nicola De Rosa, Klaudia Dussa-Zieger, Klaus Erlenbach, Joëlle Genois, Tamás Gergely, Dot Graham, Matthew Gregg, Gabriele Haller, Chinthaka Indikadahena, John Kurowski, Ine Lutterman, Isabelle Martin, Patricia McQuaid, Dénes 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 für ihre Vorschläge und Anregungen.

Revision History

Version	Datum	Bemerkungen
2.0	24.11.2024	Lokalisierte Version und interne Reviews
2.1	16.02.2025	FINAL nach Bearbeitung durch Lektorin

Einführung

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

1) Jede Einzelperson und jeder Schulungsanbieter kann diese Probeprüfung in einer Schulung verwenden, wenn ISTQB® als Quelle und Copyright-Inhaber der Probeprüfung anerkannt wird.

2) Jede Einzelperson oder Gruppe von Personen kann diese Probeprüfung als Grundlage für Artikel, Bücher oder andere abgeleitete Schriftstücke verwenden, wenn ISTQB® als Quelle und Copyright-Inhaber der Probeprüfung bestätigt wird.

3) Jedes vom ISTQB® anerkannte nationale Board kann diese Probeprüfung übersetzen und öffentlich zugänglich machen, wenn ISTQB® als Quelle und Copyright-Inhaber der Probeprüfung bestätigt wird.

4) Zu fast jeder Frage wird genau eine zutreffende Lösung erwartet. Bei den Ausnahmen wird explizit auf die Möglichkeit mehrerer Antworten hingewiesen.

Allgemeine Angaben zur Probeprüfung

Anzahl der Fragen: 40

Dauer der Prüfung: 60 Minuten

Gesamtpunktzahl: 40 (ein Punkt pro Frage)

Punktzahl zum Bestehen der Prüfung: 26 (oder mehr)

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

Frage 1	FL-1.1.1	K1	Punkt 1.0

Welche der folgenden Optionen ist ein typisches Testziel?

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® Foundation Level Syllabus V.4.0; Abschnitt 1.1.1):

a) FALSCH – Die Validierung der Übereinstimmung mit dokumentierten Anforderungen ist nicht korrekt, da sich die Validierung auf die Übereinstimmung mit den Anforderungen und Erwartungen der Benutzer bezieht, während sich die Verifizierung auf die Übereinstimmung mit spezifizierten Anforderungen bezieht. Daher wäre es richtig, "Validierung" durch "Verifizierung" zu ersetzen (siehe [CTFL 4.0], Abschnitt 1.1.1, 5. Aufzählungspunkt unter "Typische Testziele").

b) KORREKT – Das Hervorrufen von Fehlerwirkungen und das Identifizieren von Fehlerzuständen ist wahrscheinlich das häufigste Ziel dynamischer Tests (siehe [CTFL 4.0], Abschnitt 1.1.1, 2. Aufzählungspunkt unter "Typische Testziele").

c) FALSCH – Fehler zu initiieren und Ursachen zu identifizieren ist nicht korrekt, da Tester keine Fehler initiieren; sie versuchen, Fehler zu verursachen. Fehler werden typischerweise von Entwicklern gemacht (und können nicht wirklich initiiert werden) und führen zu Defekten, die Tester entweder direkt durch statisches Testen oder indirekt durch Fehler beim dynamischen Testen identifizieren. Die Identifizierung der Ursachen ist zwar nützlich, gehört aber zum Debugging, das eine vom Testen getrennte Aktivität ist (siehe [CTFL 4.0], Abschnitt 1.1.1, 2. Aufzählungspunkt partiell sowie Abschnitt 1.1.2, letzter Absatz).

d) FALSCH – Die Verifikation, dass das Testobjekt Erwartungen der Benutzer erfüllt, ist nicht korrekt, da sich die Verifikation sich darauf bezieht, ob die spezifizierten (dokumentierten) Anforderungen erfüllt werden, während sich die Validierung sich darauf bezieht, ob die Anforderungen und Erwartungen der Benutzer erfüllt werden. Daher ist es korrekt, "Verifikation" durch "Validierung" zu ersetzen (siehe [CTFL 4.0], Abschnitt 1.1.1, Letzter Aufzählungspunkt unter "Typische Testziele").

Frage 2	FL-1.1.2	K2	Punkt 1.0

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

Wählen Sie EINE Option! (1 von 4)

a)	Beim Testen treten Fehler auf, während beim Debuggen Fehler behoben werden.	☐
b)	Testen ist eine negative Aktivität, während Debuggen eine positive Aktivität ist.	☐
c)	Beim Testen werden Fehler gefunden, während beim Debuggen Fehler lokalisiert werden.	☒
d)	Beim Testen wird die Fehlerursache gefunden, während beim Debuggen die Fehlerursache behoben wird.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 1.1.2):

a) FALSCH – Dynamische Tests führen zu Fehlern (die dann lokalisiert und behoben werden). Der Zweck des Debuggens besteht jedoch darin, Fehler zu lokalisieren und zu beheben. Debugging behebt daher keine Fehler (siehe [CTFL 4.0], Abschnitt 1.1.2, 1. + 2. Absatz).

b) FALSCH – Sowohl das Testen als auch das Debugging tragen zur Verbesserung der Qualität des Testobjekts bei und sollten daher beide als positiv angesehen werden. Debugging wird im Allgemeinen als eine positive Aktivität angesehen, da es sich dabei um das Beheben von Fehlern handelt. Beim dynamischen Testen geht es darum, das Testobjekt absichtlich zum Scheitern zu bringen, weshalb es von manchen als negative Aktivität angesehen wird, aber das ist eine sehr enge Sichtweise (und nicht die, die normalerweise von Testern vertreten wird). Es gibt sowohl positive als auch negative Testfälle sind möglich. Positive Testfälle prüfen, ob das Testobjekt das tut, was es tun soll, während negative Testfälle prüfen, ob das Testobjekt nicht das tut, was es nicht tun soll (siehe [CTFL 4.0], Abschnitt 1.1.2, 1.+ 2. Absatz).

c) KORREKT – Beim Testen werden Fehler entdeckt, entweder direkt durch Beobachtung des Fehlers in Reviews (oder durch ein Werkzeug bei der statischen Analyse) oder indirekt durch die Ursache des Fehlers beim dynamischen Testen. Debugging ist eine vom Testen getrennte Aktivität (normalerweise von den Entwicklern durchgeführt), die sich mit der Lokalisierung von Fehlern (nur beim dynamischen Testen) und deren Behebung befasst (siehe [CTFL 4.0], Abschnitt 1.1.2, 1. und 2. Absatz).

d) FALSCH – Fehler werden typischerweise durch menschliches Versagen verursacht. Beim Testen werden Fehler entweder direkt durch statisches Testen oder indirekt durch Fehler beim dynamischen Testen gefunden, und beim Debuggen werden Fehler behoben. Beim Testen wird die Fehlerursache nicht gefunden und beim Debuggen wird die Fehlerursache nicht behoben (siehe [CTFL 4.0], Abschnitt 1.1.2, 3. Absatz).

Frage 3	FL-1.3.1	K2	Punkt 1.0

Der Trugschluss „keine Fehler“ bedeutet ein brauchbares System ist eines der Grundsätze des Testens. Welches der folgenden Beispiele zeigt, wie dieses Prinzip in der Praxis umgesetzt wird?

Wählen Sie EINE Option! (1 aus 4)

a)	Erklären, dass es nicht möglich ist, durch Testen die Abwesenheit von Fehlern zu beweisen.	☐
b)	Unterstützung der Endnutzer bei der Durchführung von Abnahmetests.	☒
c)	Sicherstellen, dass das gelieferte System keine Implementierungsfehler enthält.	☐
d)	Modifizieren von Tests, die keine Fehler verursachen, um sicherzustellen, dass nur wenige Fehler verbleiben.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 1.3.1):

Der Irrglaube, dass: „Keine Fehler“ ein brauchbares System bedeuten, beruht auf der Vorstellung das 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 dem Irrtum des „Abwesenheit von Fehlern“ widerspiegeln (siehe [CTFL 4.0], 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 4.0], Abschnitt 1.3, 7. Grundsatz, letzter Absatz).

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 Fehlerzuständen. Testen zeigt das Vorhandensein, und nicht die Abwesenheit von Fehlern“ besagt, dass Tests zwar das Vorhandensein von Fehlern im Testobjekt erkennen können, es nicht möglich ist, die Abwesenheit von Fehlerzuständen zu beweisen und damit deren Korrektheit zu garantieren (siehe [CTFL 4.0], 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. Dieser 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 bestätigt, dass das System den Bedürfnissen und Erwartungen der Benutzer entspricht (siehe [CTFL 4.0], Abschnitt 1.3, 5. Grundsatz).

Frage 4	FL-1.4.1	K2	Punkt 1.0

Welche der folgenden Testaktivitäten beinhaltet AM EHESTEN die Anwendung der Grenzwertanalyse und der Äquivalenzklassenbildung?

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 damit verbundenen Aufgaben erklären.

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 1.4.1):

Die Testanalyse wird im Folgenden beschrieben:

Um die Merkmale zu identifizieren, die getestet werden müssen, wird die Testbasis analysiert und als 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 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 4.0], Abschnitt 1.4.1, 5. Absatz: "Die Testrealisierung umfasst...").

b) KORREKT – Beim Testentwurf werden wahrscheinlich Testverfahren eingesetzt, um Testfälle aus Testbedingungen und Überdeckungselemente zu erstellen (siehe [CTFL 4.0], Abschnitt 1.4.1, 4. 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 4.0], Abschnitt 1.4.1, 6. 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 4.0], Abschnitt 1.4.1, 2. Absatz: "Die Testüberwachung umfasst...").

e) KORREKT – Die Testanalyse beinhaltet wahrscheinlich den Einsatz von Testverfahren, um Testbedingungen zu identifizieren (siehe [CTFL 4.0], Abschnitt 1.4.1, 3. Absatz: "Die Testanalyse umfasst...").

Frage 5	FL-1.4.3	K2	Punkt 1.0

Es werden folgende Testmittel angenommen:

1. **Testüberdeckungen**
2. **Änderungsanforderungen**
3. **Testausführungsplan**
4. **Priorisierte Testbedingungen**

und die folgenden Testaktivitäten

A. Testanalyse
B. Testentwurf
C. Testrealisierung
D. Testabschluss

Welche der folgenden Aussagen trifft AM BESTEN auf das durch die Aktivitäten erzeugte Testmittel zu?

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® Foundation Level Syllabus V.4.0; 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 4.0], Abschnitt 1.4.3, 4. Absatz: „Testanalyse ... (priorisierte) Testbedingungen (…)").

B. Testentwurf – priorisierte Testfälle, Testchartas, Überdeckungselemente (1), Testdatenanforderungen und Testumgebungsanforderungen (siehe [CTFL 4.0], Abschnitt 1.4.3, 5. Absatz: "… Überdeckungselemente ... ").

C. Testrealisierung – Testverfahren, automatisierte Testskripte, Testsuiten, Testdaten, Testausführungsplan (3) und Testumgebungselemente wie Stubs, Treiber, Simulatoren und Servicevirtualisierungen (siehe [CTFL 4.0], Abschnitt 1.4.3, 6. Absatz: "Testausführungspläne …").

D. Testabschluss – Testabschlussbericht, dokumentierte gewonnene Erkenntnisse, Verbesserungsmaßnahmen und Änderungsanforderungen (2) (als Elemente des Produkt-Backlogs (siehe [CTFL 4.0], Abschnitt 1.4.1, letzter Absatz: 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	Punkt 1.0

Welche der folgenden Aussagen über die verschiedenen Testrollen trifft AM EHESTEN zu?

Wählen Sie EINE Option! (1 aus 4)

a)	In der agilen Softwareentwicklung ist das Testmanagement hauptsächlich Aufgabe des Teams, während das Testen hauptsächlich Aufgabe einer Person außerhalb des Teams ist.	☐
b)	Die Testrolle ist hauptsächlich für die Testüberwachung und Teststeuerung verantwortlich, während die Testmanagementrolle hauptsächlich für die Testplanung und -durchführung verantwortlich ist.	☐
c)	In der agilen Softwareentwicklung werden Testmanagementaufgaben, die mehrere Teams betreffen, von einem Testmanager außerhalb des Teams übernommen, während einige Testmanagementaufgaben vom Team selbst durchgeführt werden.	☒
d)	Die Testmanagementrolle ist hauptsächlich für die Testanalyse und das Testentwurf verantwortlich, während die Testrolle hauptsächlich für die Testimplementierung und die Testdurchführung verantwortlich ist.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 4.0], 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 4.0], 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 der Testentwurf verantwortlich, obwohl es richtig ist zu sagen, dass die Rolle des Testens hauptsächlich für die Testimplementierung und die Testausführung verantwortlich ist (siehe [CTFL 4.0], Abschnitt 1.4.5.).

Frage 7	FL-1.5.2	K1	Punkt 1.0

Welcher der folgenden Punkte ist ein Vorteil des ganzheitlichen Teamansatzes (whole-team approach)?

Wählen Sie EINE Option! (1 aus 4)

a)	Teams ohne Tester	☐
b)	Verbesserte Teamdynamik	☒
c)	Mitglieder des Fachteams	☐
d)	Größere Teams	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 4.0.2], Abschnitt 1.5.2, „“Der Whole-Team Ansatz verbessert die Teamdynamik, (…)).“

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 4.0], 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 4.0], Abschnitt 1.5.2).

Frage 8	FL-1.5.3	K2	Punkt 1.0

Welche der folgenden Aussagen zur Unabhängigkeit des Testens ist zutreffend?

Wählen Sie EINE Option! (1 aus 4)

a)	Unabhängige Tester werden Fehler finden, weil sie eine andere technische Perspektive als die Entwickler haben, aber ihre Unabhängigkeit kann zu einer kontroversen Beziehung zu den Entwicklern führen.	☒
b)	Die Vertrautheit der Entwickler mit ihrem eigenen Code bedeutet, dass sie nur wenige Fehler darin finden. Aufgrund des gemeinsamen Software-Hintergrunds mit den Testern werden diese Fehler jedoch auch von den Testern gefunden.	☐
c)	Unabhängiges Testen erfordert Tester, die außerhalb des Entwicklungsteams und idealerweise außerhalb der Organisation stehen. Allerdings ist es für diese Tester schwer, die Anwendungsdomäne zu verstehen.	☐
d)	Tester außerhalb des Entwicklungsteams sind unabhängiger als Tester innerhalb des Teams, aber es ist wahrscheinlicher, dass die Tester innerhalb des Teams für Verzögerungen bei der Produktfreigabe verantwortlich gemacht werden.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 Release-Prozess wahrgenommen werden (siehe [CTFL 4.0], Abschnitt 1.5.3; 1. und 4. Absatz jeweils 1. Satz).

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 4.0], 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 Komponententest 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 4.0], 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 4.0], Abschnitt 1.5.3).

Frage 9	FL-2.1.2	K1	Punkt 1.0

Welche der folgenden Aussagen beschreibt AM BESTEN eine bewährte Testpraxis, die für alle Phasen des Softwareentwicklungslebenszyklus gilt?

Wählen Sie EINE Option! (1 aus 4)

a)	Für jede Teststufe gibt es eine entsprechende Phase im Softwareentwicklungsprozess.	☐
b)	Für jedes Testziel gibt es ein entsprechendes Entwicklungsziel.	☐
c)	Für jede Testaktivität gibt es eine entsprechende Benutzeraktivität.	☐
d)	Für jede Aktivität im Softwareentwicklungsprozess gibt es eine korrespondierende Testaktivität.	☒

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 2.1.2):

a) FALSCH – Die Qualitätskontrolle bezieht sich auf alle Entwicklungsaktivitäten, was bedeutet, dass das 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 4.0], Abschnitt 2.1.2, 1. Absatz: … Für jede Softwareentwicklungsaktivität gibt es eine entsprechende Testaktivität").

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 4.0], Abschnitt 2.1.2, Entwicklungsziel").

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 betreffen (siehe [CTFL 4.0], 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 4.0], Abschnitt 2.1.2, 1. Absatz, 1. Aufzählungspunkt).

Frage 10	FL-2.1.3	K1	Punkt 1.0

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

Wählen Sie EINE Option! (1 aus 4)

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

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 2.1.3):

a) FALSCH – Die komponententestgetriebene Entwicklung ist kein Beispiel für einen Test-First-Ansatz. Test-First-Ansätze wie Abnahmetestgetriebene Entwicklung (ATDD) sind auf spezifische Methoden zur Definition von Tests vor der Implementierung fokussiert (siehe [CTFL 4.0], 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 Abnahmetests die Benutzeranforderungen abbilden (siehe [CTFL 4.0], 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 4.0], Abschnitt 2.1.3).

d) KORREKT – Die Abnahmetestgetriebene Entwicklung (ATDD) ist ein bekanntes Beispiel für einen Test-First-Ansatz, bei dem Abnahmetests zur Definition der Anforderungen vor der Implementierung genutzt werden (siehe [CTFL 4.0], Abschnitt 2.1.3, 3. Absatz, 2. Satz: "Tests werden geschrieben, bevor der Teil der Anwendung entwickelt wird, der die Tests erfüllt.“).

Frage 11	FL-2.1.5	K2	Punkt 1.0

Welche der folgenden Aussagen beschreibt den Shift-Left-Ansatz AM BESTEN?

Wählen Sie EINE Option! (1 aus 4)

a)	Manuelle Aktivitäten werden mit Zustimmung der Entwickler automatisiert, um den um den Grundsatz „Frühes Testen spart Zeit und Geld“ zu unterstützen.	☐
b)	Wo es kosteneffektiv ist, werden Testaktivitäten auf einen früheren Zeitpunkt im Softwareentwicklungslebenszyklus (SDLC) verlagert, um die Gesamtqualitätskosten zu senken, indem die Anzahl der später im SDLC gefundenen Fehler reduziert wird.	☒
c)	Wenn freie Zeit verfügbar ist, müssen Tester Tests für Regressionstests automatisieren, beginnend mit Komponententests und Komponentenintegrationstests.	☐
d)	Wenn verfügbar, werden Tester darin geschult, Aufgaben zu Beginn des SDLC durchzuführen, so dass später im SDLC weitere Testaktivitäten automatisiert werden können.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 4.0], Abschnitt 2.1.5. 1. und letzter Absatz).

c) FALSCH – Obwohl automatisierte Komponententests und Komponentenintegrationstests für Regressionstests im Allgemeinen für Regressionstest 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 4.0], 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 4.0], Abschnitt 2.1.5).

Frage 12	FL-2.1.6	K2	Punkt 1.0

Welches der folgenden Ereignisse wird bei einer Retrospektive AM GERINGSTEN wahrscheinlich eintreten?

Wählen Sie EINE Option! (1 aus 4)

a)	Die Qualität zukünftiger Testobjekte wird durch die Identifizierung von Verbesserungen in den Entwicklungspraktiken verbessert.	☐
b)	Die Testeffizienz wird verbessert, indem die Konfiguration von Testumgebungen durch Automatisierung beschleunigt wird.	☐
c)	Das Verständnis der Endanwender für die Entwicklungs- und Testprozesse wird verbessert.	☒
d)	Automatisierte Testskripte werden durch das Feedback der Entwickler verbessert.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 4.0], Abschnitt 2.1.6).

c) KORREKT – Zu den Teilnehmern an Retrospektiven gehören typischerweise Tester, Entwickler, Architekten, Produktbesitzer und Business Analysten, aber Endbenutzer werden selten zu diesen Treffen eingeladen oder nehmen 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 4.0], 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 4.0], Abschnitt 2.1.6)

Frage 13	FL-2.2.1	K2	Punkt 1.0

Welche der folgenden Teststufen wird AM EHESTEN durchgeführt, wenn sich der Test auf die Validierung konzentriert und nicht von Testern durchgeführt wird?

Wählen Sie EINE Option! (1 aus 4)

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

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 in der Regel in ihrer Entwicklungsumgebung durchführen (siehe [CTFL 4.0], 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 in der Regel nicht von Testern durchgeführt, da diese Tests normalerweise von Entwicklern durchgeführt werden (siehe [CTFL 4.0], 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 4.0], 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 4.0], Abschnitt 2.2.1, 5. Absatz).

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 Verifikation (Überprüfung gegen Spezifikationen) konzentrieren und nicht auf Validierung (Überprüfung gegen Benutzeranforderungen).

Frage 14	FL-2.2.3	K2	Punkt 1.0

Die Software des Navigationssystems wurde aktualisiert und schlägt nun Routen vor, die gegen die Straßenverkehrsordnung verstoßen, z. B. das Befahren von Einbahnstraßen in falscher Richtung.

Welche der folgenden Aussagen beschreibt die durchzuführenden Tests AM BESTEN?

Wählen Sie EINE Option! (1 aus 4)

a)	Nur Fehlernachtests.	☐
b)	Fehlernachtests, dann Regressionstests.	☒
c)	Ausschließlich Regressionstests.	☐
d)	Regressionstests, dann Fehlernachtests.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 keine Fehler eingeführt oder aufgedeckt wurden (siehe [CTFL 4.0], 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 unveränderte Bereiche des Systems eingeführt oder aufgedeckt wurden (siehe [CTFL 4.0], Abschnitt 2.2.3, 2. + 3. 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 4.0], 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 4.0], 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	Punkt 1.0

Welcher der folgenden Beispielfehler identifiziert AM BESTEN Fehler, die durch statische Tests (statt durch dynamische Tests) gefunden werden könnten?

Wählen Sie ZWEI Optionen! (2 aus 5)

a)	Zwei verschiedene Teile der Designspezifikation stimmen aufgrund der Komplexität des Designs nicht überein.	☒
b)	Eine zu lange Antwortzeit führt dazu, dass Benutzer die Geduld verlieren.	☐
c)	Ein Fehler tritt auf, wenn das System eine Datei schreibt, während der Speicherplatz knapp wird.	☐
d)	Eine Variable wird deklariert, aber anschließend nie im Programm verwendet.	☒
e)	Das Programm benötigt zu viel Speicher, um einen Bericht zu erstellen.	☐

FL-3.1.3 (K2) Der Lernende kann statische Tests und dynamische Tests vergleichen und gegenüberstellen.

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 gefunden werden kann (siehe [CTFL 4.0], Abschnitt 4.1.4).

c) FALSCH – Speicherprobleme und Dateischreibvorgänge können nur durch dynamisches Testen entdeckt werden, da sie das tatsächliche Verhalten des Systems zur Laufzeit betreffen (siehe [CTFL 4.0], Abschnitt 4.1.4).

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 4.0], Abschnitt 6.1.1), (siehe [CTFL 4.0], 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 4.0], Abschnitt 4.1.4).

Fazit:

- **Die korrekten Antworten sind a) und d),** da diese Fehler durch statische Tests (z. B. Review oder statische Codeanalyse) identifiziert werden können.
- **Die falschen Antworten b), d.) und e)** betreffen Laufzeitverhalten und Systemressourcen und müssen durch dynamische Tests aufgedeckt werden.

Frage 16	FL-3.2.1	K1	Punkt 1.0

Welcher der folgenden Vorteile ist ein frühes und häufiges Stakeholder-Feedback?

Wählen Sie EINE Option! (1 aus 4)

a)	Änderungen der Anforderungen werden früher verstanden und umgesetzt.	☒
b)	Es stellt sicher, dass Stakeholder die Benutzeranforderungen verstehen.	☐
c)	Es ermöglicht den Produkt-Ownern, ihre Anforderungen so oft zu ändern, wie sie möchten.	☐
d)	Endbenutzern wird vor der Freigabe mitgeteilt, welche Anforderungen nicht umgesetzt werden.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 4.0], 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 4.0], 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 4.0], 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	Punkt 1.0

Unter Berücksichtigung der folgenden Reviewtypen:

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

und der folgenden Beschreibungen:

A. **Konzentriert sind auf Ziele wie Konsensfindung, Generierung neuer Ideen und Motivation der Autoren, sich zu verbessern.**
B. **Dient in erster Linie der Erkennung potenzieller Mängel und erfordert keine formelle Dokumentation.**
C. **Das Hauptziel besteht darin potenzielle Fehlerzustände zu erkennen und Metriken zu sammeln, um die Prozessverbesserung zu unterstützen.**
D. **Das Ziel ist es, die Reviewer zu schulen und gleichzeitig einen Konsens zu erzielen, neue Ideen zu generieren und potenzielle Mängel aufzudecken.**

Somit ist die Frage:

Welche der folgenden Aussagen trifft AM BESTEN auf die Reviewtypen und deren Beschreibungen zu?

Wählen Sie EINE Option! (1 von 4)

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

- **Das technische Review** (1) wird von technisch qualifizierten Reviewern durchgeführt und zielt darauf ab, einen Konsens zu erzielen, neue Ideen zu generieren und Autoren zur Verbesserung zu motivieren (**entspricht A**).
- Das **informelle Review** (2) hat das primäre Ziel, Anomalien zu erkennen, und erfordert keine formale Dokumentation (**entspricht D**).
- Die **Inspektion** (3) ist die formellste Art der Überprüfung, bei der der Schwerpunkt auf der Erkennung von Mängeln und der Erfassung von Metriken zur Verbesserung von Prozessen liegt (**entspricht C**).
- **Das Walkthrough** (4) wird vom Autor geleitet und umfasst Ziele wie die Schulung von Prüfern, die Erzielung eines Konsenses, die Generierung neuer Ideen und die Erkennung von Mängeln (**entspricht B**).

Daher:

a) FALSCH

b) KORREKT – Die richtige Antwort lautet: 1A, 2D, 3C, 4B

c) FALSCH

d) FALSCH

Daher:

a) FALSCH, weil

- **2B:** Informelles Review passt nicht zu Beschreibung B.
  - Informelle Reviews zielen in erster Linie auf die Identifizierung von Fehlerzuständen ab, ohne formale Dokumentation oder Konsensbildung (Beschreibung D passt besser).
- **4D:** Walkthroughs passen nicht zu Beschreibung D.
  - Walkthroughs sind autorengetrieben und haben vielseitige Ziele wie Konsensbildung, neue Ideen und Schulung der Gutachter (Beschreibung B passt besser).

Fazit: Die falschen Zuordnungen machen diese Option inkorrekt.

b) KORREKT – Die korrekte Übereinstimmung ist: 1A, 2D, 3C, 4B.

- **Technisches Review (1A):** Die Ziele des technischen Reviews umfassen die Konsensbildung und das Generieren neuer Ideen.
  Referenz: (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").
- **Informelles Review (2D):** Informelle Reviews dienen der Fehlererkennung ohne formal dokumentierte Ergebnisse.
  Referenz: (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").

- **Inspektion (3C):** Inspektionen haben das Hauptziel der Fehlererkennung und erfassen Metriken zur Prozessverbesserung.
  Referenz: (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").

- **Walkthrough (4B):** Walkthroughs umfassen Ziele wie Konsensbildung, Generierung neuer Ideen und die Aufklärung der Gutachter.
  Referenz: (siehe [CTFL 4.0], Abschnitt 3.2.4, 3. Absatz “Walkthrough”: "Ziele umfassen die Erzielung eines Konsenses” und die ”Schulung von Gutachtern").

c) FALSCH, weil

- **1B:** Technisches Review passt nicht zu Beschreibung B.

  Technische Reviews sind zielgerichtet und umfassen die Generierung neuer Ideen sowie die Motivation der Autoren. Beschreibung A ist zutreffender.

- **2C:** Informelles Review passt nicht zu Beschreibung C.

  Informelle Reviews beinhalten keine formellen Prozesse oder Metrikenerfassung, die für Beschreibung C typisch sind.

- **3D:** Inspektionen passen nicht zu Beschreibung D.

  Inspektionen sind die formellste Art des Reviews und erfassen Metriken für die Prozessverbesserung. Beschreibung C ist korrekter.

- **4A:** Walkthroughs passen nicht zu Beschreibung A.

  Walkthroughs sind autorengetrieben und verfolgen vielseitige Ziele, die besser durch Beschreibung B abgedeckt werden.

Fazit: Alle Zuordnungen in dieser Option sind inkorrekt.

d) FALSCH, weil

- **1C:** Technisches Review passt nicht zu Beschreibung C.

  Technische Reviews fokussieren sich auf Konsensbildung und die Motivation der Autoren. Beschreibung A ist zutreffender.

- **3A:** Inspektionen passen nicht zu Beschreibung A.

  Inspektionen sind formelle Reviews mit einem klaren Fokus auf Fehlererkennung und Metrikenerfassung, die besser zu Beschreibung C passen.

Fazit: Diese falschen Zuordnungen machen die Option inkorrekt.

Fazit:

- **Antwort b) ist korrekt**, da sie die richtige Zuordnung der Review-Typen zu ihren Beschreibungen trifft.

- **Antworten a), c) und d) sind falsch**, da sie entweder die Ziele oder den formalen Charakter der jeweiligen Review-Typen falsch interpretieren.

Frage 18	FL-3.2.5	K1	Punkt 1.0

Welcher der folgenden Faktoren trägt zu einem erfolgreichen Review bei?

Wählen Sie EINE Option! (1 aus 4)

a)	Sicherstellen, dass das Management als Gutachter teilnimmt.	☐
b)	Große Arbeitsprodukte in kleinere Teile aufteilen.	☒
c)	Setzen Sie die Bewertung durch den Gutachter als Ziel.	☐
d)	Ein Dokument pro Review planen.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Section 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 Review teilnehmen sollten (siehe [CTFL 4.0], Abschnitt 3.2.5).

b) KORREKT – Um erfolgreiche Reviews sicherzustellen, ist es wichtig, das Arbeitsergebnis in Teile aufzuteilen, die klein genug sind, um in einem angemessenen Zeitrahmen überprüft zu werden, um zu verhindern, dass Gutachter während einzelner Reviews oder Reviewbesprechungen den Fokus verlieren (siehe [CTFL 4.0], Abschnitt 3.2.5, 3. Aufzählungspunkt).

c) FALSCH – Um erfolgreiche Reviews zu gewährleisten, ist es wichtig, klare Ziele und messbare Abschlusskriterien zu definieren, ohne die “Bewertung der Teilnehmer” (siehe [CTFL 4.0], Abschnitt 3.2.5,).

d) FALSCH – Um sicherzustellen, dass Reviews erfolgreich sind, ist es wichtig, das Review 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 4.0], 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 Review-Praktiken widerspiegeln.

Frage 19	FL-4.1.1	K2	Punkt 1.0

Was ist der Hauptunterschied zwischen Black-Box- und erfahrungsbasierten Testverfahren?

Wählen Sie EINE Option! (1 von 4)

a)	Das Testobjekt.	☐
b)	Die Teststufe, auf dem das Testverfahren verwendet wird.	☐
c)	Die Testbasis.	☒
d)	Der Softwareentwicklungslebenszyklus (SDLC), in dem das Testverfahren verwendet werden kann.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 4.1.1):

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 4.0], Abschnitt 4.1).

b) FALSCH – Auf allen Teststufen können sowohl Black-Box-Testverfahren als auch erfahrungsbasierte Testverfahren eingesetzt werden (siehe [CTFL 4.0], 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 4.0], 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 4.0], Abschnitt 4.1.1).

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	Punkt 1.0

Sie testen einen PIN-Validator, der gültige PINs akzeptiert und ungültige PINs ablehnt. Eine PIN ist eine Folge on Ziffern. Eine PIN ist gültig, wenn sie aus vier Ziffern besteht, von denen mindestens zwei unterschiedlich sind.

Welche der folgenden ist ein Satz von Eingabetestdaten, der alle identifizierten Äquivalenzklassen abdeckt?

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 Äquivalenzklassenbildung zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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	Punkt 1.0

Ein Entwickler wurde gebeten, die folgende Geschäftsregel zu implementieren:

EINGABE: Wert (Ganzzahl)

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

ELSE schreibe „Wert OK“

Sie entwerfen die Testfälle mit Hilfe einer 2-Wert-Grenzwertanalyse.

Welcher der folgenden Sätze von Testeingaben erreicht DIE GRÖSSTE Überdeckung?

Wählen Sie EINE Option! (1 von 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® Foundation Level Syllabus V.4.0; 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-BVA 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 den BVA mit zwei Werten, sodass wir eine Überdeckung von 50 % erreichen. Es fehlen die benachbarten Grenzwerte 101 und 199, wodurch nur 50 % der Grenzwerte abgedeckt werden (siehe [CTFL 4.0], Abschnitt 4.2.2).

b) FALSCH – Nur 100 und 200 sind gültige Überdeckungselemente für den BVA mit zwei Werten. Die benachbarten Grenzwerte 101 und 199 werden nicht getestet, weshalb die Überdeckung ebenfalls nur 50 % beträgt (siehe [CTFL 4.0], Abschnitt 4.2.2).

c) FALSCH – Nur 100 und 101 sind gültige Überdeckungselemente für den BVA mit zwei Werten, ignorieren jedoch 199 und 200. Damit wird ebenfalls nur 50 % der möglichen Grenzwerte abgedeckt (siehe [CTFL 4.0], Abschnitt 4.2.2).

d) KORREKT – Nur 101, 199 und 200 sind gültige Überdeckungselemente für den 2-Wert-BVA, sodass wir eine Überdeckung von 75 % erreichen. Diese Option berücksichtigt die Anforderungen des 2-Wert-BVA am besten (siehe [CTFL 4.0], Abschnitt 4.2.2).

Fazit

- Die korrekte Option ist **d)**, da sie die größte Überdeckung der Grenzwerte erreicht.
- Die falschen Optionen sind a), b) und c), da sie entweder zu viele unnötige Werte oder nicht alle relevanten Grenzwerte abdecken.

Frage 22	FL-4.2.3	K3	Punkt 1.0

Sie arbeiten an einem Projekt zur Entwicklung eines Systems zur Analyse von Fahrprüfungsergebnissen. Sie wurden gebeten, Testfälle basierend auf der folgenden Entscheidungstabelle zu entwerfen.

	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 Testdaten zeigen, 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 Entscheidungstabellentests zur Ableitung von Testfällen anwenden.

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], Abschnitt 4.2.3, 3. Absatz).

b) FALSCH – Diese Kombination (T, F, T) entspricht nur der Regel R2, sodass kein Widerspruch zwischen mehreren Regeln besteht (siehe [CTFL 4.0], 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 4.0], Abschnitt 4.2.3)).

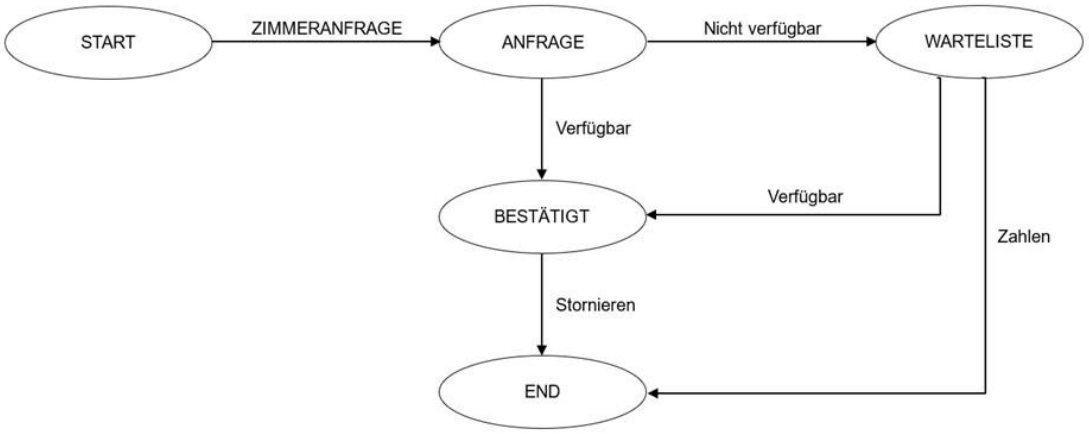
d) KORREKT – Diese Kombination (F, F, F) passt sowohl zu R2 als auch zu R3, aber R2 und R3 haben unterschiedliche Aktionen („Zusätzliche Fahrstunden anfordern“ und „Antrag auf Wiederholung der Prüfung“. Das zeigt einen Widerspruch zwischen den Regeln (siehe [CTFL 4.0], Abschnitt 4.2.3, 3.letzter Absatz: Widersprüche in den Anforderungen).

Fazit:

- Die korrekte Option ist **d)**, 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 basierend auf dem folgenden Zustandsdiagramm:



Wie viele Testfälle sind MINDESTENS erforderlich, um eine 100 %ige Überdeckung gültiger Übergänge zu erreichen?

Wählen Sie EINE Option! (1 of 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® Foundation Level Syllabus V.4.0; 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) → ENDE

TF2: START (Zimmeranfrage) → ANFRAGE (Nicht verfügbar) → WARTELISTE (Verfügbar) → BESTÄTIGT (Stornieren) → ENDE

TF3: START (Zimmeranfrage) → ANFRAGE (Nicht verfügbar) → WARTELISTE (Zahlen) → ENDE

Daher:

a) KORREKT

b) FALSCH

c) FALSCH

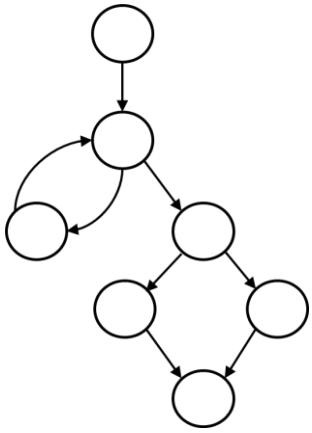
d) FALSCH

Fazit:

- Die korrekte Option ist **a)**, da drei Testfälle ausreichen, um alle gültigen Übergänge im Zustandsdiagramm abzudecken.

Frage 24	FL-4.3.2	K2	Punkt 1.0

Sie möchten Verzweigungstests auf den Code anwenden, der durch den folgenden Kontrollflussgraph dargestellt wird.



Wie viele Überdeckungselemente müssen Sie testen?

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® Foundation Level Syllabus V.4.0; Abschnitt 4.3.2):

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

Daher:

a) FALSCH

b) FALSCH

c) KORREKT

d) FALSCH

Frage 25	FL-4.3.3	K2	Punkt 1.0

Wie können White-Box-Tests zur Unterstützung von Black-Box-Tests nützlich sein?

Wählen Sie EINE Option! (1 aus 4)

a)	White-Box-Überdeckungsmaßnahmen können Testern bei der Bewertung von Black-Box-Tests hinsichtlich der durch diese Black-Box Tests erreichten Codeüberdeckung helfen.	☒
b)	Die White-Box-Überdeckungsanalyse kann Testern dabei helfen, nicht erreichbare Fragmente des Quellcodes zu identifizieren.	☐
c)	Zweigtests beinhalten Black-Box-Testverfahren, so dass das Erreichen einer vollständigen Zweigüberdeckung sicherstellt, dass jedes Black-Box-Testverfahren vollständig abgedeckt wird.	☐
d)	White-Box-Testverfahren können Überdeckungselemente für Black-Box-Testverfahren liefern.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 erhöhen (siehe [CTFL 4.0], 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.

☞ 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 4.0], Abschnitt 4.3.3).

c) FALSCH – Im Allgemeinen gibt es keine direkten (subsumierten) Beziehungen zwischen White-Box- und Black-Box-Verfahren (siehe [CTFL 4.0], 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 basierend auf der Spezifikation zu entwerfen. Daher besteht kein Zusammenhang zwischen den aus diesen beiden Testverfahren abgeleiteten Überdeckungselementen (siehe [CTFL 4.0], Abschnitt 4.3.3).

Frage 26	FL-4.4.1	K2	Punkt 1.0

Betrachten Sie die folgende Liste:

- **Korrekte Eingabe nicht akzeptiert**
- **Falsche Eingabe akzeptiert**
- **Falsches Ausgabeformat**
- **Durch Null teilen**

Welches Testverfahren wird HÖCHSTWAHRSCHEINLICH von dem Tester verwendet, der diese Liste beim Testen verwendet?

Wählen Sie EINE Option! (1 aus 4)

a)	Explorative Tests	☐
b)	Fehlerangriff	☒
c)	Checklistenbasiertes Testen	☐
d)	Grenzwertanalyse	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 4.4.1):

a) FALSCH – Beim explorativen Testen werden Testchartas 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 4.0], Abschnitt 4.4.1, 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 4.0], Abschnitt 4.4.1, 3. Absatz: Fehlerangriffe erfordern, dass “der 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 Checklisten-basierten 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 4.0], Abschnitt 4.4.1, allgemeiner Kontext: "Beim checklistenbasierten Testen wird eine Liste von Testbedingungen verwendet...").

d) FALSCH – Grenzwertanalyse basiert auf einer Analyse der Grenzwerte von Äquivalenzklassen. In der obigen Liste werden Äquivalenzklassen oder ihre Grenzen nicht erwähnt (siehe [CTFL 4.0], Abschnitt 4.4.1, allgemeiner Kontext: "Die Grenzwertanalyse basiert auf der Analyse von Äquivalenzklassen und deren Grenzen...").

Die korrekte Option ist **b)**, da die beschriebene Liste von Fehlerarten typisch für das Verfahren des Fehlerangriffs ist.

Frage 27	FL-4.4.3	K2	Punkt 1.0

Welche der folgenden Aussagen beschreibt AM BESTEN, wie der Einsatz checklistenbasierter Tests zu einer höheren Überdeckung führen kann?

Wählen Sie EINE Option! (1 aus 4)

a)	Checklistenelemente können mit einem ausreichend niedrigen Detaillierungsgrad definiert werden, sodass der Tester auf der Grundlage dieser Elemente detaillierte Testfälle implementieren und ausführen kann.	☐
b)	Checklisten können automatisiert werden. Jedes Mal, wenn eine automatisierte Testausführung die Checklistenpunkte abdeckt, führt dies zu einer zusätzlichen Überdeckung.	☐
c)	Jeder Checklistenpunkt sollte separat und unabhängig getestet werden, sodass die Elemente unterschiedliche Bereiche der Software abdecken.	☐
d)	Zwei Tester, die Tests auf der Grundlage derselben übergeordneten Checklistenelemente entwerfen und ausführen, führen die Tests normalerweise auf leicht unterschiedliche Weise durch.	☒

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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" (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 4.0], Abschnitt 4.4.3, letzter Satz).

Frage 28	FL-4.5.2	K2	Punkt 1.0

Welches der folgenden ist DAS BESTE Beispiel für ein szenarioorientiertes Akzeptanzkriterium?

Wählen Sie EINE Option! (1 aus 4)

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

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 4.0], 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 4.0], 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 verfasst sein, die sie verstehen. Dieser Satz wird für diese Stakeholder höchstwahrscheinlich unverständlich sein (siehe [CTFL 4.0], 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 szenariobasiertes Akzeptanzkriterium (siehe [CTFL 4.0], 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	Punkt 1.0

Sie verwenden abnahmetestgesteuerte Entwicklung und entwerfen Testfälle basierend auf der folgenden User-Story:

Als regulärer oder spezieller Benutzer möchte ich meine elektronische Etagenkarte für den Zutritt zu bestimmten Etagen nutzen können.

Akzeptanzkriterium:

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

AC2: Etage 4 ist nur für Spezialbenutzer zugänglich

AC3: Spezialbenutzer haben alle Zugriffsrechte von regulären Benutzern

---

Welcher Testfall ist DER SINNVOLLSTE, um AC3 zu testen?

Wählen Sie EINE Option! (1 von 4)

a)	Überprüfen Sie, ob ein regulärer Benutzer Zugang zu den Etagen 1 und 3 hat.	☐
b)	Stellen Sie sicher, dass ein regulärer Benutzer keinen Zugang zur Etage 4 hat.	☐
c)	Überprüfen Sie, ob ein Spezialbenutzer Zugang zur Etage 5 hat.	☐
d)	Überprüfen Sie, ob ein Spezialbenutzer Zugang zu den Etagen 1, 2 und 3 hat.	☒

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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. Aber 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 Spezialbenutzer Zugriff auf Etagen hat, die für einen regulären Benutzer zugänglich sind. Die korrekte Option ist **d)**, da dieser Testfall spezifisch die Zugriffsrechte eines Spezialbenutzers auf die regulären Etagen überprüft, wie es AC3 verlangt.

Frage 30	FL-5.1.1	K1	Punkt 1.0

Welcher der folgenden Punkte ist NICHT der Zweck eines Testkonzepts?

Wählen Sie EINE Option! (1 aus 4)

a)	Um Testdaten und erwartete Ergebnisse für Komponententests und Komponentenintegrationstests zu definieren.	☒
b)	Ein Endekriterium auf der Komponententestebene zu definieren, dass „100 % Anweisungsüberdeckung und 100 % Zweigüberdeckung erreicht werden müssen“.	☐
c)	Beschreiben, welche Felder der Testfortschrittsbericht enthalten soll und wie dieser Bericht aussehen soll.	☐
d)	Erläutern, warum Systemintegrationstests vom Test ausgeschlossen werden, obwohl die Teststrategie diese Teststufe erfordert.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 5.1.1):

a) KORREKT – Das Testkonzept kann Testdatenanforderungen (siehe [CTFL 4.0], 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 4.0], 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 4.0], 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 4.0], Abschnitt 5.1.1), Typische Inhalte eines Testkonzepts, 4. Aufzählungspunkt).

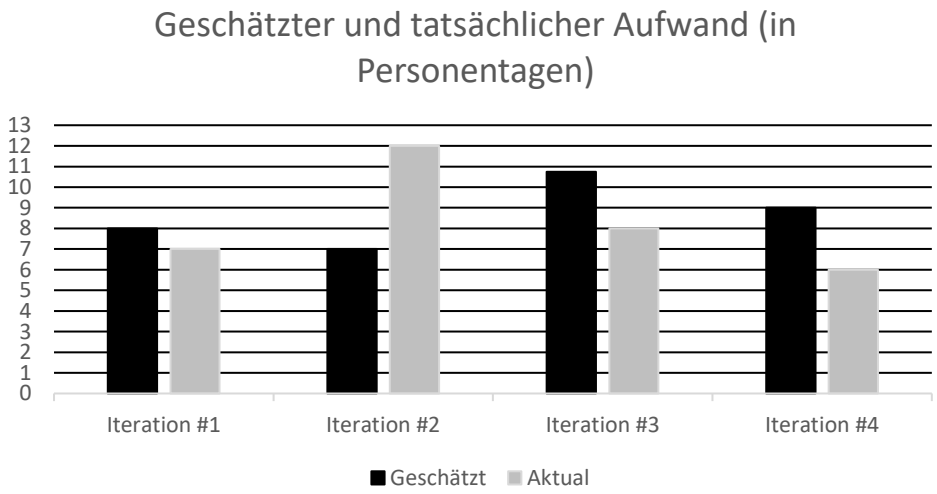
d) FALSCH – Einer der Zwecke eines Testkonzepts besteht darin, nachzuweisen, dass sich die Tests an die bestehende Testrichtlinie und Teststrategie halten, oder zu erklären, warum die Tests davon abweichen. Dies ist ein Beispiel zur Erläuterung der Abweichung hinsichtlich der Teststufen, die befolgt werden (oder nicht befolgt werden) (siehe [CTFL 4.0], Abschnitt 5.1.1, Typische Inhalte eines Testkonzepts, 2. Aufzählungspunkt).

Frage 31	FL-5.1.4	K3	Punkt 1.0

Zu Beginn jeder Iteration schätzt das Team den Arbeitsaufwand (in Personentagen), den es während der Iteration erledigen muss. Sei E(n) der geschätzte Arbeitsaufwand für Iteration n und A(n) der tatsächliche Arbeitsaufwand in Iteration n. Ab der dritten Iteration verwendet das Team das folgende Schätzmodell basierend auf Extrapolation:

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

Die Grafik zeigt den geschätzten und tatsächlichen Arbeitsaufwand 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® Foundation Level Syllabus V.4.0; 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

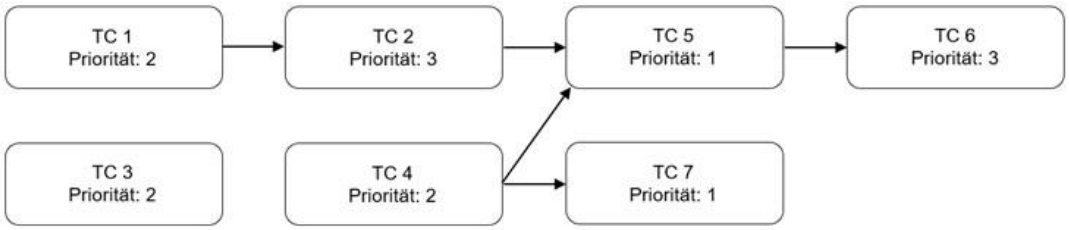
Die korrekte Option ist **c)**, da die Schätzung von 6,5 Personentagen mit der Formel und den historischen Daten übereinstimmt.

Frage 32	FL-5.1.5	K3	Punkt 1.0

Sie bereiten einen Testausführungsplan für die Ausführung von sieben Testfällen TC 1 bis TC 7 vor.

Die folgende Abbildung enthält die Prioritäten dieser Testfälle (1 = höchste Priorität, 3 = niedrigste Priorität).

Die Abbildung zeigt auch die Abhängigkeiten zwischen Testfällen anhand von Pfeilen. Beispielsweise bedeutet der Pfeil von TC 4 nach TC 5, dass TC 5 nur ausgeführt werden kann, wenn zuvor TC 4 ausgeführt wurde.



Welcher Testfall soll als sechster ausgeführt werde?

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® Foundation Level Syllabus V.4.0; Abschnitt 5.1.5):

Wir wollen Testfälle entsprechend ihrer 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 Zeitplan 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	Punkt 1.0

Was zeigt das Testpyramidenmodell?

Wählen Sie EINE Option! (1 aus 4)

a)	Dass Tests unterschiedliche Prioritäten haben können.	☐
b)	Dass Tests eine unterschiedliche Granularität haben können.	☒
c)	Dass Tests möglicherweise unterschiedliche Überdeckungskriterien erfordern.	☐
d)	Dass Tests von anderen Tests abhängen können.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Section 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 4.0], 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 4.0], 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 4.0], 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 4.0], Abschnitt 5.1.6).

Frage 34	FL-5.1.7	K2	Punkt 1.0

Welche Beziehung besteht zwischen den Testquadranten, Teststufen und Testtypen?

Wählen Sie EINE Option! (1 aus 4)

a)	Testquadranten stellen bestimmte Kombinationen von Teststufen und Testtypen dar und definieren deren Position im Softwareentwicklungslebenszyklus.	☐
b)	Testquadranten beschreiben den Grad der Granularität einzelner Testtypen, die auf jeder Testebene durchgeführt werden.	☐
c)	Testquadranten ordnen den Teststufen die durchführbaren Testarten zu.	☐
d)	Testquadranten gruppieren Teststufen und Testtypen nach mehreren Kriterien, z. B. der Ausrichtung auf bestimmte Stakeholder.	☒

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 5.1.7):

a) FALSCH – Testquadranten gruppieren Teststufen und Testarten getrennt nach mehreren Kriterien. Sie stellen keine Kombinationen von Teststufen und Testtypen dar und haben keinen Bezug zu einem Ort innerhalb eines Softwareentwicklungslebenszyklus. Sowohl Teststufen als auch Testtypen werden im Testquadrantenmodell separat behandelt (siehe [CTFL 4.0], Abschnitt 5.1.7).

b) FALSCH – Testquadranten gruppieren Teststufen und Testtypen nach mehreren Kriterien. Sie beschreiben nicht den Grad der Granularität der einzelnen Testtypen, die auf jeder Teststufe durchgeführt werden. Ein solches Modell hinsichtlich der Teststufen wird als Testpyramide bezeichnet (siehe [CTFL 4.0], Abschnitt 5.1.7).

c) FALSCH – Die Aussage ist falsch, da grundsätzlich jeder Testtyp auf jeder Teststufe durchgeführt werden kann (siehe [CTFL 4.0], Abschnitt 5.1.7).

d) KORREKT – Die Testquadranten gruppieren Teststufen, Testtypen, 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 4.0], Abschnitt 5.1.7, 1. Satz).

Frage 35	FL-5.2.3	K2	Punkt 1.0

Welches ist ein Beispiel dafür, wie die Produktrisikoanalyse die Gründlichkeit und den Umfang der Tests beeinflussen kann?

Wählen Sie EINE Option! (1 aus 4)

a)	Eine kontinuierliche Risikoüberwachung ermöglicht es uns, auftretende Risiken so früh wie möglich zu erkennen.	☐
b)	Die Risikoidentifizierung ermöglicht uns, risikomindernde Maßnahmen umzusetzen und das Risikoniveau zu senken.	☐
c)	Das bewertete Risikoniveau hilft uns bei der Wahl der Intensität der Tests.	☒
d)	Aus der Risikoanalyse ermöglicht uns die Ableitung von Überdeckungselementen.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; 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 Intensität und den Umfang der Tests für verschiedene Testobjekte festzulegen. Höhere Risiken erfordern intensivere Tests.

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

Frage 36	FL-5.3.2	K2	Punkt 1.0

Bei welchen der folgenden Aktivitäten im Testprozess werden Testfortschrittsberichte AM MEISTEN genutzt?

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® Foundation Level Syllabus V.4.0; Abschnitt 5.3.2):

a) FALSCH – Testfortschrittsberichte werden hauptsächlich während der Testüberwachung und Teststeuerung sowie beim Testabschluss verwendet, nicht während des Testentwurfs (siehe [CTFL 4.0], 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 dessen Abschlusskriterien erfüllt sind. **Dieser Bericht verwendet Informationen aus Testfortschrittsberichten** und anderen Daten (siehe [CTFL 4.0], Abschnitt 5.3.2, „**Der Testabschlussbericht basiert auf Testfortschrittsberichten** (…)).

c) FALSCH – Testfortschrittsberichte werden hauptsächlich während der Testüberwachung und Teststeuerung sowie beim Testabschluss verwendet, nicht während der Testanalyse (siehe [CTFL 4.0], 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 4.0], Abschnitt 5.3.2).

Frage 37	FL-5.4.1	K2	Punkt 1.0

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

Wählen Sie EINE Option! (1 aus 4)

a)	Alle Commits an das Repository werden eindeutig identifiziert und versioniert.	☐
b)	Alle Änderungen an den Elementen der Testumgebung werden verfolgt.	☐
c)	Alle Anforderungsspezifikationen werden in Testkonzepten eindeutig referenziert.	☐
d)	Alle identifizierten Fehlerzustände haben einen zugeordneten Status.	☒

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 5.4.1):

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 4.0], Abschnitt 5.4.1).

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 4.0], Abschnitt 5.4.1).

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 4.0], Abschnitt 5.4.1).

d) KORREKT – Das Zuweisen eines Status zu Fehlerzuständen ist eine Aufgabe des Fehlermanagements (siehe [CTFL 4.0], Abschnitt 5.5.1), nicht des Konfigurationsmanagements. Das Konfigurationsmanagement konzentriert sich auf die Kontrolle und Verfolgbarkeit von Artefakten und Änderungen (siehe [CTFL 4.0], Abschnitt 5.4.1).

Frage 38	FL-5.5.1	K3	Punkt 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 sollte zur Anmeldeseite weitergeleitet werden.

Tatsächliches Ergebnis: Die Anmeldeschaltfläche reagiert nicht, wenn daraufgeklickt wird.

Schweregrad: Hoch

Priorität: Dringend

Was sind DIE WICHTIGSTEN Informationen, die in diesem Bericht fehlen?

Wählen Sie EINE Option! (1 aus 4)

a)	Name des Testers und Datum des Berichts.	☐
b)	Angaben zur Testumgebung und ihre Versionsnummern.	☒
c)	Identifizierung des Testobjekts.	☐
d)	Auswirkungen auf die Interessen der Stakeholder.	☐

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 5.5.1):

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 4.0], Abschnitt 5.5.1, Fehlerbericht, 3. Aufzählungspunkt).

b) KORREKT – Angaben zur Testumgebung (z. B. der 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 4.0], Abschnitt 5.5.1, Fehlerbericht, 4.Aufzählungspunkt)

c) FALSCH – Das Testobjekt wurde im Bericht klar identifiziert (WebShop v0.99). Diese Information ist vorhanden und kein fehlender Bestandteil (siehe [CTFL 4.0], Abschnitt 5.5.1, Fehlerbericht, 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, sodass diese Information nicht fehlt (siehe [CTFL 4.0], Abschnitt 5.5.1, Fehlerbericht, 8.+ 9. Aufzählungspunkt).

Frage 39	FL-6.1.1	K2	Punkt 1.0

Werkzeuge aus welcher der folgenden Kategorien helfen bei der Organisation von Testfällen, erkannten Fehlern und dem Konfigurationsmanagement?

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® Foundation Level Syllabus V.4.0; Abschnitt 6.1.1):

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, das sie sich primär auf die Testdurchführung und Metriken konzentrieren (siehe [CTFL 4.0], Abschnitt 6.1.1).

b) FALSCH – Testentwurfs- und Testrealisierungswerkzeuge erleichtern die Erstellung von Testfällen, Testdaten und Testskripten [CTFL 4.0], Abschnitt 6.1.1, 4. 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 4.0], Abschnitt 6.1.1).

d) KORREKT – Testmanagementwerkzeuge erhöhen die Effizienz des Testprozesses, indem sie die Verwaltung des Softwareentwicklungslebenszyklus (SDLC), der Anforderungen, Tests, Fehler und des Konfigurationsmanagements erleichtern (siehe [CTFL 4.0], Abschnitt 6.1.1, 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)
- Konfigurationsmanagement (Versionierung von Testartefakten, Änderungsverfolgung).

Frage 40	FL-6.2.1	K1	Punkt 1.0

Welcher der folgenden Vorteile ist AM EHESTEN ein Vorteil der Testautomatisierung?

Wählen Sie EINE Option! (1 aus 4)

a)	Die Fähigkeit, Testfälle ohne Zugriff auf die Testbasis zu generieren.	☐
b)	Das Erreichen einer größeren Überdeckung durch eine objektivere Bewertung.	☐
c)	Die Erhöhung der verfügbaren Testausführungszeiten bei höherer Verarbeitungsleistung.	☐
d)	Die Vermeidung menschlicher Fehler durch größere Konsistenz und Wiederholbarkeit.	☒

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

Begründung (siehe ISTQB® Foundation Level Syllabus V.4.0; Abschnitt 6.2.1):

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 Reihenfolge ausgeführt (siehe [CTFL 4.0], Abschnitt 6.2.1, 2. Absatz, 2. Aufzählungspunkt).

Platz für Ihre Notizen:
(werden bei der Korrektur weder gelesen noch bewertet)

Platz für Ihre Notizen:
(werden bei der Korrektur weder gelesen noch bewertet)