



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

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

\_\_\_\_\_

\_\_\_\_\_

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

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

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

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

\_\_\_\_\_

\_\_\_\_\_

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

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

---

## Sample Exam Paper

---

**Verfasst von:**

**German Testing Board e. V. – Examination Panel**

**(SET A4Q\_TF4D\_Sample-Exam-Answers\_SetA\_v1.0\_DE)**

**A4Q TF4D Syllabus 2021 V.1.0 // Glossary**



## Einführung

Dies ist eine Probepfprüfung. Sie hilft den Kandidaten bei ihrer Vorbereitung auf die Zertifizierungsprüfung. Enthalten sind Fragen, deren Struktur, Aufbau und Format einer der regulären Prüfung ähnelt.

Die vorliegende Fassung der Probepfprüfungsfragen für den A4Q-TF4D wurde aus folgenden Quellen zusammengesetzt:

- der deutschsprachigen Lokalisierung des GTB des ISTQB® CTFL CORE 2018; V.3.1; SAMPLE EXAM SET A und SET B,
- der deutschsprachigen Lokalisierung des CTAL 2019 (V.3.0) SAMPLE EXAM PAPER,
- und weiteren ergänzenden Fragen erstellt von einer Arbeitsgruppe des GTB.

Die Probepfprüfung ist urheberrechtlich geschützt. Gemeinsame Inhaber der Urheber- und ausschließlichen Nutzungsrechte sind das International Software Testing Qualifications Board (ISTQB®) und das German Testing Board e. V. (GTB).

Es ist strengstens verboten, die Prüfungsfragen als Gegenstand einer Zertifizierungsprüfung zu nutzen, insbesondere sie für diese Zweck zu vervielfältigen, zu verbreiten oder öffentlich zugänglich zu machen.

Gestattet ist die Nutzung, insbesondere die Vervielfältigung, Verbreitung und öffentliche Zugänglichmachung der Probepfprüfung jeder Einzelperson für den Zweck der eigenen Vorbereitung auf die Zertifizierungsprüfung und sowie jedem nach dem ISTQB®-Standard akkreditierten Schulungsanbieter für den Zweck der Durchführung von Schulungen. Jedem vom ISTQB® anerkannten nationalen Board ist es darüber hinaus gestattet, die Probepfprüfung in andere Sprachen zu übersetzen und die Übersetzungen zu vervielfältigen, zu verbreiten und öffentlich zugänglich zu machen sowie Nutzungsrechte an der Probepfprüfung und deren Übersetzungen im Umfang der vorgenannten Bestimmungen einzuräumen. Jede Nutzung ist nur gestattet, wenn ISTQB® und GTB als Quelle und Copyright-Inhaber benannt werden.

Im Übrigen ist die Nutzung der Probepfprüfung – soweit sie nicht nach dem Gesetz über Urheberrechte und verwandte Schutzrechte vom 9. September 1965 (UrhG) erlaubt ist – nur mit ausdrücklicher Zustimmung von ISTQB® und GTB gestattet.

Die Verwendung der in diesem Werk enthaltenen Informationen erfolgt auf die alleinige Gefahr des Nutzers. GTB übernimmt insbesondere keine Gewähr für die Vollständigkeit, die technische Richtigkeit, die Konformität mit gesetzlichen Anforderungen oder Normen, sowie die wirtschaftliche Verwertbarkeit der Informationen. Es werden durch dieses Dokument keinerlei Produktempfehlungen ausgesprochen.

Die Haftung von GTB ist im Übrigen auf Vorsatz und grobe Fahrlässigkeit beschränkt.



---

## **Allgemeine Angaben zur Probepfprüfung:**

Anzahl der Fragen: 40

Dauer der Pfprüfung: 60 Minuten

Gesamtpunktzahl: 40 (ein Punkt pro Frage)

Punktzahl zum Bestehen der Pfprüfung: 26 (oder mehr)

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

---

## Fragen zum Thema "Grundlagen des Testens"

---

Frage 1	Keywords	K1	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen ist die korrekte Definition des Begriffes „Testfall“?**

**Wählen Sie genau EINE Antwort aus.**

a)	Teilmenge des Wertebereichs innerhalb einer Komponente oder eines Systems, für die aufgrund der Spezifikation erwartet wird, dass alle Werte gleichartig behandelt werden.	<input type="checkbox"/>
b)	Menge von Vorbedingungen, Eingaben, Aktionen, erwarteten Ergebnissen und Nachbedingungen, welche auf Basis von Testbedingungen entwickelt wurden.	<input checked="" type="checkbox"/>
c)	Arbeitsergebnis, welches während des Testprozesses erstellt wird und dazu gebraucht wird, um die Tests zu planen, zu entwerfen, auszuführen, auszuwerten und darüber zu berichten.	<input type="checkbox"/>
d)	Informationsquelle zur Ermittlung des erwarteten Ergebnisses, um es mit dem tatsächlichen Ergebnis eines Systems unter Test zu vergleichen.	<input type="checkbox"/>

Schlüsselwort: Testfall

Begründung: (A4Q TF4D Syllabus 2021; Glossar)

- a) FALSCH – Dies ist die Definition der Äquivalenzklasse aus dem Glossar.
- b) KORREKT – Basierend auf der Definition des Testfalls aus dem Glossar.**
- c) FALSCH – Dies ist die Definition des Testmittel aus dem Glossar.
- d) FALSCH – Dies ist die Definition des Testorakeles aus dem Glossar.

<b>Frage 2</b>	<b>FL-1.1.1</b>	<b>K1</b>	<b>Punkte 1.0</b>
----------------	-----------------	-----------	-------------------

**Welche der folgenden Aussagen beschreibt ein gültiges Ziel des Testens?**

**Wählen Sie genau EINE Antwort aus.**

a)	Der Test soll möglichst spät starten, damit die Entwicklung genug Zeit hatte, ein gutes Produkt zu erstellen.	<input type="checkbox"/>
b)	Es soll validiert werden, ob das Testobjekt so funktioniert, wie es die Benutzer und andere Stakeholder erwarten.	<input checked="" type="checkbox"/>
c)	Es soll nachgewiesen werden, dass alle möglichen Fehlerzustände identifiziert wurden.	<input type="checkbox"/>
d)	Es soll nachgewiesen werden, dass alle verbleibenden Fehlerzustände keine Fehlerwirkungen verursachen werden.	<input type="checkbox"/>

FL-1.1.1 (K1) Typische Ziele des Testens identifizieren können.

Begründung (TF4D Lehrplan 2021; V.1.0, Glossar)

- a) FALSCH – Widerspruch zu Grundsatz 3: „Frühes Testen spart Zeit und Geld.“ (siehe A4Q TF4D Lehrplan 2021; Abschnitt 1.3).
- b) KORREKT – Dies ist eines der Ziele des Testens (vgl. A4Q TF4D Lehrplan 2021; Abschnitt 1.1.1).**
- c) FALSCH – Grundsatz 2 besagt, dass vollständiges Testen unmöglich ist, und es kann nicht bewiesen werden, dass alle Fehlerzustände identifiziert wurden (siehe A4Q TF4D Lehrplan 2021; Abschnitt 1.3).
- d) FALSCH – Um eine Einschätzung treffen zu können, ob ein Defekt einen Fehler verursacht oder nicht, muss man den Fehlerzustand zunächst erkennen. Zu sagen, dass verbleibende Fehlerzustände keine Fehlerwirkungen verursachen, bedeutet implizit, dass alle Fehlerzustände gefunden wurden. Dies widerspricht erneut Grundsatz 2 (siehe A4Q TF4D Lehrplan 2021; Abschnitt 1.3).

Frage 3	FL-1.1.2	K2	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen beschreibt den Unterschied zwischen Testen und Debugging zutreffend?**

**Wählen Sie genau EINE Antwort aus.**

a)	Testen identifiziert die Ursache von Fehlerzuständen. Debugging analysiert die Fehlerzustände und schlägt Präventionsmaßnahmen vor.	<input type="checkbox"/>
b)	Dynamische Tests zeigen Fehlerwirkungen auf, die durch Fehlerzustände verursacht wurden. Debugging ist eine Entwicklungsaktivität, die Fehlerzustände beseitigt, die die Ursache von Fehlerwirkungen sind.	<input checked="" type="checkbox"/>
c)	Testen entfernt Fehlerwirkungen; Debugging entfernt dagegen Fehlerzustände, die Fehlerwirkungen verursachen.	<input type="checkbox"/>
d)	Dynamische Tests verhindern die Ursache von Fehlerwirkungen. Debugging entfernt die Fehlerwirkungen.	<input type="checkbox"/>

FL-1.1.2 (K2) Testen von Debugging unterscheiden können.

Begründung (A4Q TF4D Lehrplan 2021; V.1.0, Glossar)

- a) FALSCH – Durch Testen können nicht die Ursachen von Fehlerwirkungen identifiziert werden, sondern nur durch Debugging (siehe A4Q TF4D Lehrplan 2021, Abschnitt 1.1.2).
- b) KORREKT – Dynamisches Testen zeigt Fehlerwirkungen auf, die durch Fehlerzustände verursacht wurden. Durch Debugging können die Ursachen von Fehlerwirkungen analysiert und beseitigt werden (siehe A4Q TF4D Lehrplan 2021; Abschnitt 1.1.2).**
- c) FALSCH – Durch Testen werden keine Fehlerzustände und daraus folgende Fehlerwirkungen entfernt, sondern nur durch Debugging (siehe A4Q TF4D Lehrplan 2021; Abschnitt 1.1.2).
- d) FALSCH – Durch dynamische Tests können die Ursachen von Fehlerwirkungen (d. h. Fehlerzustände) nicht verhindert werden, sondern nur das Vorhandensein von Fehlerzuständen, die Fehlerwirkungen hervorrufen, nachgewiesen werden. (siehe A4Q TF4D Lehrplan 2021; Abschnitt 1.1.2 und Abschnitt 1.3; 1. Grundsatz).

<b>Frage 4</b>	<b>FL-1.2.1</b>	<b>K2</b>	<b>Punkte 1.0</b>
----------------	-----------------	-----------	-------------------

**Welcher ist ein wichtiger Grund für Testen im Softwareentwicklungsprozess?**

**Wählen Sie genau EINE Antwort aus.**

a)	Testen kann, durch seine Ergebnisse, als Werkzeug zur Leistungsbewertung der Entwickler benutzt werden.	<input type="checkbox"/>
b)	Testen kann helfen, mögliche Ausfälle der SW im Betrieb zu verhindern.	<input checked="" type="checkbox"/>
c)	Testen ist immer gesetzlich vorgeschrieben.	<input type="checkbox"/>
d)	Testen stellt stets sicher, dass alle Anforderungen vollständig korrekt erfüllt sind.	<input type="checkbox"/>

FL-1.2.1 (K2) Beispiele dafür geben können, warum Testen notwendig ist

Begründung

a) FALSCH – Dies ist keines der im A4Q TF4D Lehrplan 2021 Kapitel 1.2 genannten Gründe, im Gegensatz laut Lehrplan soll dies gerade nicht stattfinden

b) KORREKT – Lehrplan A4Q TF4D 2021 S 16 1.2

“Gründliches Testen von Komponenten oder Systemen und ihrer zugehörigen Dokumentation kann dabei helfen, das Risiko von Fehlerwirkungen zu reduzieren, die während des Betriebs auftreten können.”

c) FALSCH – Lehrplan A4Q TF4D 2021, S. 16: 1.2 Warum ist Testen notwendig?  
 “Darüber hinaus kann Softwaretesten auch notwendig sein, um vertragliche oder rechtliche Anforderungen oder branchenspezifische Standards zu erfüllen.” --> es kann, ist es aber nicht immer.

d) FALSCH – Lehrplan A4Q TF4D 2021, S. 16: 1.2.1 Der Beitrag des Testens zum Erfolg  
 “Historisch betrachtet ist es schon immer üblich gewesen, dass Software und Systeme in Betrieb genommen werden, obwohl Fehlerwirkungen infolge von Fehlerzuständen auftreten oder in anderer Weise die Bedürfnisse der Stakeholder nicht erfüllt werden. Allerdings kann der Einsatz geeigneter Testverfahren die Häufigkeit derartiger problematischer Inbetriebnahmen reduzieren, wenn diese Verfahren mit dem entsprechenden Grad an Testkompetenz, in den geeigneten Teststufen und zum richtigen Zeitpunkt im Softwareentwicklungslebenszyklus eingesetzt werden. “Durch das Testen wird die Häufigkeit reduziert, es ist aber nicht möglich, alle Probleme und Risiken vollständig abzudecken. (Siehe auch 1.3 Sieben Grundsätze des Testens: Vollständiges Testen ist nicht möglich)

Frage 5	FL-1.2.2	K2	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen beschreibt die Beziehung zwischen Testen und Qualitätssicherung?**

**Wählen sie genau EINE Antwort aus.**

a)	Testen ist Bestandteil der Qualitätssicherung	<input checked="" type="checkbox"/>
b)	Testen führt stets zu besseren Produkthanforderungen	<input type="checkbox"/>
c)	Frühes Testen im Entwicklungsprozess trägt nur wenig zur Qualitätssicherung bei	<input type="checkbox"/>
d)	Je mehr Testfälle ausgeführt werden desto höhere Qualität hat die Software	<input type="checkbox"/>



FL-1.2.2 (K2) Die Beziehung zwischen Testen und Qualitätssicherung beschreiben können und Beispiele dafür geben können, wie Testen zu höherer Qualität beiträgt

Begründung:

- a) **KORREKT – Da die Qualitätssicherung sich mit der korrekten Ausführung des gesamten Prozesses beschäftigt, unterstützt Qualitätssicherung korrektes Testen (vgl. Lehrplan A4Q TF4D 2021, Abschnitt 1.2.2)**
- b) **FALSCH –** Beispielsweise kann trotz gründlicher Tests aller spezifizierten Anforderungen und Beheben aller Fehlerzustände ein System erstellt werden, das schwer zu nutzen ist, das die Bedürfnisse und Erwartungen der Benutzer nicht erfüllt oder das geringwertigere Qualität hat als vergleichbare Systeme (vgl. Lehrplan A4Q TF4D 2021, Abschnitt 1.3)
- c) **FALSCH –** (vgl. Lehrplan A4Q TF4D 2021, Abschnitt 1.3)  
Um Fehlerzustände früh zu finden, sollten sowohl statische als auch dynamische Testaktivitäten so früh wie möglich im Softwareentwicklungslebenszyklus gestartet werden. Frühes Testen wird oft als Shift left bezeichnet. Frühes Testen im Softwareentwicklungslebenszyklus hilft dabei, kostenintensive Änderungen zu reduzieren oder vollständig zu vermeiden (siehe Abschnitt 3.1 Grundlagen des statischen Tests).
- d) **FALSCH –** (vgl. Lehrplan A4Q TF4D 2021, Abschnitt 1.3)  
Einige Unternehmen erwarten, dass Tester alle denkbaren Tests durchführen und alle denkbaren Fehlerzustände finden können, aber die Grundsätze 2 und 1 lehren uns, dass dies unmöglich ist. Des Weiteren ist es ein Trugschluss (d. h. ein Irrglaube), zu erwarten, dass allein das Finden und Beheben einer großen Anzahl von Fehlerzuständen den Erfolg eines Systems sicherstellen werde. Beispielsweise kann trotz gründlicher Tests aller spezifizierten Anforderungen und Beheben aller Fehlerzustände ein System erstellt werden, das schwer zu nutzen ist, das die Bedürfnisse und Erwartungen der Benutzer nicht erfüllt oder das geringwertigere Qualität hat als vergleichbare Systeme

<b>Frage 6</b>	<b>FL-1.2.3</b>	<b>K2</b>	<b>Punkte 1.0</b>
----------------	-----------------	-----------	-------------------

**Welches der folgenden Beispiele ist eine Fehlerwirkung in einem Tempomat eines Autos?**

**Wählen Sie genau EINE Antwort aus.**

a)	Der Entwickler des Systems hat vergessen, Variablen nach einem Ausschneiden und Einfügen umzubenennen.	<input type="checkbox"/>
b)	Nicht benötigter Code, der beim Rückwärtsfahren einen Alarm auslöst, wurde in das System aufgenommen.	<input type="checkbox"/>
c)	Das System hält die eingestellte Geschwindigkeit nicht mehr ein, wenn die Radiolautstärke erhöht oder verringert wird.	<input checked="" type="checkbox"/>
d)	Die System-Entwurfsspezifikation gibt die Geschwindigkeit falsch an.	<input type="checkbox"/>

FL-1.2.3 (K2) Zwischen Fehlhandlung, Fehlerzustand und Fehlerwirkung unterscheiden können

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Dies ist ein Beispiel für einen vom Entwickler begangenen Fehler (Fehlhandlung).
- b) FALSCH – Dies ist ein Beispiel für einen Fehlerzustand (etwas, das im Code falsch ist und einen Fehler verursachen kann).
- c) **KORREKT – Dies ist eine Abweichung von der erwarteten Funktionalität - ein Tempomat sollte nicht von der Laufstärkeinstellung des Radios betroffen sein.**
- d) FALSCH – Dies ist ein Beispiel für einen Defekt (Fehlerzustand in einer Spezifikation, die eine Fehlerwirkung verursachen kann, wenn gegen die Spezifikation anschließend implementiert wird).

Frage 7	FL-1.2.4	K2	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen ist eher ein Fehlerzustand als eine Grundursache fur einen Fehlerzustand in einem Fitness-Tracker?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Der Anforderungsmanager war mit der Domane des Fitnessstrainings nicht vertraut und ging zu Unrecht davon aus, dass die Benutzer die Herzschlag-Frequenz in Schlagen pro Stunde ablesen wollen.	<input type="checkbox"/>
b)	Der Tester des Smartphone-Interfaces war nicht im zustandsbasierten Testen geschult und hat daher einen signifikanten Fehler ubersehen.	<input type="checkbox"/>
c)	Eine vom Entwickler fur die GPS-Funktion fehlerhaft implementierte Konfigurationsvariable kann wahrend der Sommerzeit zu Standortproblemen fuhren.	<input checked="" type="checkbox"/>
d)	Die Designerin der Benutzeroberflache hat noch nie an tragbaren Geraten gearbeitet und missverstand deshalb die Auswirkungen von reflektiertem Sonnenlicht.	<input type="checkbox"/>

FL-1.2.4 (K2) Zwischen der Grundursache eines Fehlerzustands und seinen Auswirkungen unterscheiden konnen

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Die mangelnde Vertrautheit des Autors der Anforderungen mit der Fitnessdomane ist eine wesentliche Grundursache fur einen Fehlerzustand.
- b) FALSCH – Die fehlende Schulung des Testers im zustandsbasierten Testen war eine der wesentlichen Grundursachen fur den Fehler.
- c) **KORREKT – Die fehlerhaften Konfigurationsdaten stellen einen Fehlerzustand in der Software des Fitness-Trackers dar, der zu Fehlerwirkungen fuhren kann. Die Grundursache ist „Eine vom Entwickler fur die GPS-Funktion fehlerhaft implementierte Konfigurationsvariable. (V.3.0).“**
- d) FALSCH – Die mangelnde Erfahrung bei der Gestaltung von Benutzeroberflachen fur tragbare Gerate ist ein typisches Beispiel fur eine Grundursache fur einen Fehlerzustand.

Frage 8	FL-1.3.1	K2	Punkte 1.0
---------	----------	----	------------

**Ein Tester hat über einen Zeitraum von 5 Jahren Software-Applikationen auf mobilen Geräten einem Test unterzogen. Er hat sich einen großen Erfahrungsschatz im Testen von mobilen Applikationen angeeignet und erzielt in kürzerer Zeit bessere Ergebnisse als andere. Über einen längeren Zeitraum hat der Tester die existierenden automatisierten Testfälle nicht modifiziert und auch keine neuen Testfälle mehr erstellt. Dies führt dazu, dass durch Ausführung der Tests immer weniger Fehler gefunden werden. Welchen Grundsatz des Softwaretestens hat der Tester nicht beachtet?**

**Wählen Sie genau EINE Antwort aus.**

a)	Testen ist abhängig vom Umfeld	<input type="checkbox"/>
b)	Vollständiges Testen ist nicht möglich	<input type="checkbox"/>
c)	Wiederholungen haben keine Wirksamkeit	<input checked="" type="checkbox"/>
d)	Häufung von Fehlerzuständen	<input type="checkbox"/>

FL-1.3.1 (K2) Die sieben Grundsätze des Softwaretestens erklären können.

Begründung (Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Test ist abhängig vom Umfeld, egal, ob manuell oder automatisiert (vgl. Lehrplan A4Q TF4D 2021; Abschnitt 1.3; 6. Grundsatz), führt aber nicht dazu, dass - wie oben beschrieben - immer weniger Fehler aufgedeckt werden.
- b) FALSCH – Erschöpfendes vollständiges Testen ist unmöglich, egal wieviel Aufwand wir in den Test investieren (vgl. Lehrplan A4Q TF4D 2021; Abschnitt 1.3; 2. Grundsatz), führt aber nicht dazu, dass - wie oben beschrieben - immer weniger Fehler aufgedeckt werden.
- c) KORREKT – Ein Grundsatz (gem. Lehrplan A4Q TF4D 2021) besagt: "Vorsicht vor dem Pestizid-Paradoxon" bzw. Wiederholungen haben keine Wirksamkeit (vgl Lehrplan A4Q TF4D 2021; Abschnitt 1.3; 5. Grundsatz), dass eine Ausführung immer der gleichen Testfälle keine neuen Erkenntnisse mehr bringt. Um neue Fehlerzustände zu finden, müssen bestehende Tests möglicherweise verändert werden und neue Tests erstellt werden.**
- d) FALSCH – Oftmals ist eine "Häufung von Fehlerzuständen" (Lehrplan A4Q TF4D 2021; Abschnitt 1.3; 4. Grundsatz) in einer kleinen Anzahl von Modulen zu beobachten; das führt aber nicht dazu, dass - wie oben beschrieben - immer weniger Fehler aufgedeckt werden.

---

## Fragen zum Thema "Testen im Softwareentwicklungslebenszyklus"

---

Frage 9	Keywords	K1	Punkte 1.0
---------	----------	----	------------

**Welche der folgenden Aussagen ist eine korrekte Definition fur den Regressionstest?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Testen um festzustellen, ob in unveranderten Bereichen der Software Fehlerzustande eingebaut wurden.	<input checked="" type="checkbox"/>
b)	Testen der Auswirkungen einer geanderten Umgebung auf ein in Produktion laufendes System.	<input type="checkbox"/>
c)	Testen, das durch anderungen eines laufenden Systems veranlasst wird.	<input type="checkbox"/>
d)	Testen nach Behebung eines Fehlerzustands, um zu bestatigen, dass eine Fehlerwirkung nicht mehr auftritt.	<input type="checkbox"/>

Begrundung: (Lehrplan A4Q TF4D 2021, Glossar)

a) **KORREKT** – Basierend auf der Definition des Regressionstests aus dem Glossar .

b) FALSCH – Dies ist die Definition eines Wartungstests aus dem Glossar.

c) FALSCH – Dies ist die Definition eines Wartungstests aus dem Glossar.

d) FALSCH – Dies ist die Definition eines Fehlernachttests aus dem Glossar.

Frage 10	FL-2.3.1	K2	Punkte 1.0
----------	----------	----	------------

**Welcher der folgenden Tests ist ein White-Box-Testverfahren?**

**Wählen sie genau EINE Antwort aus.**

a)	Entscheidungstest	<input checked="" type="checkbox"/>
b)	Performancetest	<input type="checkbox"/>
c)	Code Review	<input type="checkbox"/>
d)	Äquivalenzklassenanalyse	<input type="checkbox"/>

FL-2.3.1 (K2) Funktionale, nicht-funktionale und White-Box-Tests vergleichen können

Begründung:

- a) **KORREKT** – Entscheidungstest ist ein White-Box-Testverfahren. (siehe Lehrplan A4Q TF4D 2021, Abschnitt 4.3 White-Box-Testverfahren).
- b) **FALSCH** – Performanztest hat das Ziel, die Performanz des Testobjekts zu bewerten (siehe Glossar), und ist deshalb eine Testart und kein Testverfahren. (siehe Lehrplan A4Q TF4D 2021. Abschnitt 2.2.2)
- c) **FALSCH** – Code-Review ist eine statische Testart, und gehört damit nicht zu den White-Box-Tests
- d) **FALSCH** – Äquivalenzklassenanalyse ist ein Black-Box-Verfahren (siehe Abschnitt 4.2 Black-Box-Testverfahren).

Frage 11	FL-2.3.3	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen vergleicht die Zwecke der Fehlernachtests und Regressionstests AM BESTEN miteinander?**

**Wählen Sie genau EINE Antwort aus.**

a)	Der Regressionstest stellt sicher, dass alle früher durchgeführten Tests immer noch korrekt laufen, während der Fehlernachtest sicherstellt, dass Korrekturen an einem Teil des Systems die anderen Teile nicht negativ beeinflussen.	<input type="checkbox"/>
b)	Der Fehlernachtest prüft, dass ein vorher gefundener Fehlerzustand korrigiert wurde, während der Regressionstest sicherstellt, dass die Korrektur keine anderen Teile des Systems negativ beeinflusst hat.	<input checked="" type="checkbox"/>
c)	Der Regressionstest stellt sicher, dass Korrekturen an einem Teil des Systems die anderen Teile nicht negativ beeinflussen, während der Fehlernachtest prüft, dass alle früher durchgeführten Tests immer noch die gleichen Ergebnisse produzieren.	<input type="checkbox"/>
d)	Der Fehlernachtest bestätigt, dass die Änderungen am System erfolgreich durchgeführt wurden, während der Regressionstest Tests durchführt, die vorher fehlgeschlagen sind, um sicherzustellen, dass sie jetzt korrekt funktionieren.	<input type="checkbox"/>

FL-2.3.3 (K2) Den Zweck von Fehlernachtests und Regressionstests vergleichen können

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Obwohl die Beschreibung des Regressionstests weitgehend richtig ist, ist die Beschreibung des Fehlernachtests (welcher bestätigen sollte, dass ein Fehlerzustand behoben wurde) falsch.
- b) KORREKT – Die Beschreibungen von Fehlernachtest und Regressionstest entsprechen sinngemäß Lehrplan A4Q TF4D 2021, Abschnitt 2.2.4.**
- c) FALSCH – Obwohl die Beschreibung des Regressionstests weitgehend richtig ist, ist die Beschreibung des Fehlernachtests (erneute Durchführung aller früher durchgeführten Tests) falsch. Der Zweck des Fehlernachtests ist zu prüfen, ob Tests, die früher fehlgeschlagen sind, jetzt korrekt funktionieren.
- d) FALSCH – Obwohl die Beschreibung des Fehlernachtests weitgehend richtig ist, ist die Beschreibung des Regressionstests (Tests durchzuführen, die vorher fehlgeschlagen sind) falsch. Das würde eher einer detaillierteren Beschreibung des Fehlernachtests entsprechen.

Frage 12	FL-2.4.1	K2	Punkte 1.0
----------	----------	----	------------

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

**Wählen Sie genau EINE Antwort aus.**

a)	Die Entscheidung, die Wartbarkeit der Software zu testen	<input checked="" type="checkbox"/>
b)	Die Entscheidung, das System nach der Migration auf einer neuen Betriebsplattform zu testen	<input type="checkbox"/>
c)	Die Entscheidung zu testen, ob archivierte Daten abgerufen werden können	<input type="checkbox"/>
d)	Die Entscheidung zum Testen nach "Hot Fixes"	<input type="checkbox"/>

FL-2.4.1 (K2) Auslöser für Wartungstests zusammenfassen können.

Begründung (nach Lehrplan A4Q TF4D 2021; Glossar)

- a) **KORREKT** – Dies ist ein Wartbarkeitstest und nicht ein Wartungstest. „... die meisten Wartbarkeitsfehler (können) nur durch statische Tests gefunden werden.“ (Lehrplan A4Q TF4D 2021, Kap. 3.1.3, letzter Absatz)
- b) **FALSCH** – Dies ist ein Auslöser für Wartungstests, siehe Lehrplan A4Q TF4D 2021, Kapitel 2.3.1: Betriebstests der neuen Umgebung, sowie der geänderten Software.
- c) **FALSCH** – Dies ist ein Auslöser für Wartungstests, siehe Lehrplan A4Q TF4D 2021, Kapitel 2.3.1
- d) **FALSCH** – Dies ist ein Auslöser für Wartungstests, siehe Lehrplan A4Q TF4D 2021, Kapitel 2.3.1.



<b>Frage 13</b>	<b>FL-2.4.2</b>	<b>K2</b>	<b>Punkte 1.0</b>
-----------------	-----------------	-----------	-------------------

**Welche der folgenden Aussagen beschreibt eine Aufgabe der Auswirkungsanalyse in der Wartung KORREKT?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Die Auswirkungsanalyse unterstutzt bei der Entscheidung, ob sich eine Fehlerkorrektur beim zu wartenden System lohnt.	<input checked="" type="checkbox"/>
b)	Die Auswirkungsanalyse identifiziert, wie Daten in das gewartete System zu migrieren sind.	<input type="checkbox"/>
c)	Die Auswirkungsanalyse unterstutzt bei der Entscheidung, welche Hot Fixes den meisten Nutzen fur den Benutzer haben.	<input type="checkbox"/>
d)	Die Auswirkungsanalyse unterstutzt die Ermittlung der Effektivitat neuer Wartungstestfalle.	<input type="checkbox"/>

FL-2.4.2 (K2) Den Einsatz der Auswirkungsanalyse im Wartungstest beschreiben konnen

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

a) **KORREKT** – Die Auswirkungsanalyse kann verwendet werden, um die Bereiche des Systems zu identifizieren, die von einer Fehlerbehebung betroffen sind. Daher kann das Ausma der Auswirkungen (z. B. notige Regressionstests) verwendet werden, um bei der Entscheidung zu helfen, ob die Veranderung vorgenommen werden sollte (Siehe Lehrplan A4Q TF4D 2021, Abschnitt 2.3.2).

b) **FALSCH** – Obwohl der Test der migrierten Daten Teil des Wartungstests ist (siehe Konvertierungstest), ermittelt die Auswirkungsanalyse nicht, wie dieser Test auszusehen hat.

c) **FALSCH** – Die Auswirkungsanalyse ermittelt, welche Teile eines Systems von einer anderung betroffen sind. Somit kann sie den Unterschied zwischen verschiedenen Hot Fixes in Hinsicht auf die Auswirkungen auf das System aufzeigen. Sie gibt aber keine Hinweise darauf, welchen Nutzen die anderungen fur den Benutzer haben.

d) **FALSCH** – Die Auswirkungsanalyse ermittelt, welche Teile eines Systems von einer anderung betroffen sind. Sie kann aber keine Anhaltspunkte zur Effektivitat von Testfallen liefern.

---

## Fragen zum Thema "Statischer Test"

---

Frage 14	Keywords	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen ist eine korrekte Definition für das Checklistenbasierte Review?**

**Wählen Sie genau EINE Antwort aus.**

a)	Ein Reviewverfahren, das entlang einer Liste an Fragen oder geforderten Eigenschaften geführt wird.	<input checked="" type="checkbox"/>
b)	Ein Review, das einem definierten Prozess folgt und ein formell dokumentiertes Ergebnis liefert.	<input type="checkbox"/>
c)	Eine Art statischer Test, bei dem ein Arbeitsergebnis oder -prozess von einer oder mehreren Personen bewertet wird, um Fehlerzustände zu erkennen oder Verbesserungen zu erzielen.	<input type="checkbox"/>
d)	Ein Reviewverfahren, bei dem ein Arbeitsergebnis aus der Perspektive unterschiedlicher Stakeholder beurteilt wird.	<input type="checkbox"/>

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

- a) **KORREKT** – Basierend auf der Definition des checklistenbasierten Reviews aus dem Glossar.
- b) FALSCH – Dies ist die Definition eines formalen Reviews aus dem Glossar.
- c) FALSCH – Dies ist die Definition eines Reviews allgemein aus dem Glossar.
- d) FALSCH – Dies ist die Definition eines rollenbasierten Reviews aus dem Glossar.

Frage 15	Keywords	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen ist eine korrekte Definition fur die zyklomatische Komplexitat?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Die maximale Anzahl von linear unabhangigen Pfaden durch ein Programm.	<input checked="" type="checkbox"/>
b)	Schwierigkeitsgrad, mit dem der Entwurf und/oder die interne Struktur einer Komponente oder eines Systems zu verstehen, zu warten und zu prufen ist.	<input type="checkbox"/>
c)	Die uberdeckung einer Sequenz von (N+1) Zustandsubergangen.	<input type="checkbox"/>
d)	Die uberdeckung samtlicher Ergebnisse von atomaren Teilbedingungen, die sich unabhangig voneinander auf das Gesamtergebnis der Entscheidung auswirken.	<input type="checkbox"/>

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

- a) **KORREKT** – Basierend auf der Definition der zyklomatischen Komplexitat aus dem Glossar.
- b) FALSCH – Dies ist die Definition der Komplexitat aus dem Glossar.
- c) FALSCH – Dies ist die Definition der N-Switch uberdeckung aus dem Glossar.
- d) FALSCH – Dies ist die Definition der modifizierten Bedingungs-/ Entscheidungsuberdeckung (MC/DC) aus dem Glossar.

Frage 16	Keywords	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen ist eine korrekte Definition für den paarweisen Integrationstest?**

**Wählen Sie genau EINE Antwort aus.**

a)	Eine Art von Integrationstest, der auf Paare von Komponenten abzielt, die entsprechend der Darstellung in einem Aufrufgraphen zusammenwirken.	<input checked="" type="checkbox"/>
b)	Eine Teststufe mit dem Schwerpunkt auf dem Zusammenwirken zwischen Komponenten oder Systemen.	<input type="checkbox"/>
c)	Eine Art von Integrationstest, bei dem alle Knoten, die mit einem bestimmten Knoten verbunden sind, die Grundlage für den Integrationstest sind.	<input type="checkbox"/>
d)	Testen, dessen Testelemente die Schnittstellen und das Zusammenwirken integrierter Komponenten sind.	<input type="checkbox"/>

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

a) **KORREKT** – Basierend auf der Definition des paarweisen Integrationstests aus dem Glossar.

b) FALSCH – Dies ist die Definition des Integrationstests aus dem Glossar.

c) FALSCH – Dies ist die Definition des Umgebungsintegrationstests aus dem Glossar.

d) FALSCH – Dies ist die Definition des Komponentenintegrationstests aus dem Glossar.

Frage 17	FL-3.2.4	K3	Punkte 1.0
----------	----------	----	------------

Sie wurden gebeten, an einem checklistenbasierten Review des folgenden Auszuges aus einer Anforderungsspezifikation eines Bibliothekssystems teilzunehmen:

**Bibliothekare können:**

1. Neue Nutzer registrieren.
2. Bücher von Nutzern zurücknehmen.
3. Mahngebühren für Nutzer erheben.
4. Neue Bücher mit deren ISBN, Autor und Titel dem System hinzufügen.
5. Bücher aus dem System löschen.
6. Systemrückmeldungen innerhalb von 5 Sekunden erhalten.

**Nutzer können:**

7. Maximal drei Bücher gleichzeitig ausleihen.
8. Die Historie ihrer ausgeliehen/reservierten Bücher anschauen.
9. Mit einer Mahngebühr wegen Nichtrückgabe eines Buches innerhalb von 3 Wochen belegt werden.
10. Systemrückmeldungen innerhalb von 3 Sekunden erhalten.
11. Ein Buch kostenfrei für maximal 4 Wochen ausleihen.
12. Bücher reservieren (falls sie ausgeliehen sind).

**Alle Benutzer (Bibliothekare und Nutzer):**

13. Können Bücher nach ISBN, Autor oder Titel suchen.
14. Können den Systemkatalog durchstöbern.
15. Das System soll innerhalb von 3 Sekunden auf Benutzeranfragen reagieren.
16. Die Benutzungsschnittstelle soll einfach zu bedienen sein.

Ihnen wurde der Checklisteneintrag zugewiesen, der ein Review der Spezifikation auf Inkonsistenzen zwischen einzelnen Anforderungen vorsieht (beispielsweise Konflikte zwischen Anforderungen).

*(Fortsetzung auf der nächsten Seite)*

**Welche folgenden Kombinationen weisen Inkonsistenzen zwischen Anforderungspaaren RICHTIG auf?**

**Wahlen Sie genau EINE Antwort aus.**

a)	6-10, 6-15, 7-12	<input type="checkbox"/>
b)	6-15, 9-11	<input checked="" type="checkbox"/>
c)	6-10, 6-15, 9-11	<input type="checkbox"/>
d)	6-15, 7-12	<input type="checkbox"/>

FL-3.2.4 (K3) Ein Reviewverfahren auf ein Arbeitsergebnis anwenden konnen, um Fehlerzustande zu finden

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

Zu betrachten werden die potenziellen Inkonsistenzen:

- 6-10 – Wenn Bibliothekare eine Systemruckmeldung innerhalb von 5 Sekunden erhalten sollen, dann ist das konsistent dazu, dass Nutzer eine Systemruckmeldung innerhalb von 3 Sekunden erhalten sollen.
- 6-15 – Wenn Bibliothekare eine Systemruckmeldung innerhalb von 5 Sekunden erhalten sollen, dann ist das inkonsistent dazu, dass alle Benutzer eine Systemruckmeldung innerhalb von 3 Sekunden erhalten sollen.
- 7-12 – Wenn sich Nutzer maximal 3 Bucher gleichzeitig ausleihen konnen, dann ist das konsistent dazu, dass sie sich Bucher reservieren konnen (falls sie ausgeliehen sind).
- 9-11 – Wenn ein Nutzer wegen Nichtruckgabe eines Buches innerhalb von 3 Wochen mit einer Mahngebuhr belegt wird, dann ist das inkonsistent dazu, dass Bucher kostenlos fur maximal 4 Wochen ausgeliehen werden durfen. Die zwei Fristen sind unterschiedlich.

Folglich ist Antwort b) korrekt.

Frage 18	TTA-3.2.1	K3	Punkte 1.0
----------	-----------	----	------------

Unten finden Sie den Pseudo-Code fur das Programm EASY:

```
00  program EASY
01  var1, var2, var3 : integer
02  easy: boolean
02  begin
03      read ( var2 )
04      read ( var1 )
05      read (easy)
06      If (easy = true) then
07          var3 = var2 + var1
08          print ( var3 )
09          if var1 = 5 then
10              print ( var1 )
11          else
12              print ( var1+1 )
13          endif
14          var2 = var2 + 1
15      else
16          var2 = 0
17          write ( "Wow - that was tricky!" )
18      endif
19  write ( var2 )
20  end program EASY
```

Was ist die zyklomatische Komplexitat des Programms?

Wahlen Sie genau EINE Antwort aus.

a)	2	<input type="checkbox"/>
b)	4	<input type="checkbox"/>
c)	1	<input type="checkbox"/>
d)	3	<input checked="" type="checkbox"/>

Begrundung:

a) FALSCH

b) FALSCH

c) FALSCH

d) **KORREKT** – Die zyklomatische Komplexitat bezieht sich auf die Anzahl der unabhangigen Pfade durch ein Programm. Im Programm Easy gibt es drei unabhangige Pfade:  
Pfad 1: (easy = false)  
Pfad 2: (easy = true, var1 = 5)  
Pfad 3: (easy = true, var1 not = 5)



Frage 19	TTA-3.2.2	K2	Punkte 1.0
----------	-----------	----	------------

**Nachfolgend sehen Sie den Pseudo-Code für ein Programm, das Verkaufsprovisionen berechnet und ausdrückt:**

```
00 program Provisionsberechnung
01 summe, nummer : integer
02 provision_max, provision_min : real
03 begin
04     read ( nummer )
05     while nummer ≠ -1 loop
06         summe = summe + nummer
07         read ( nummer )
08     endloop
09     if summe > 1000 then
10         provision_max = 100 + 0.2 * ( summe - 1000 )
11     else
12         provision_min = 0.15 * summe
13     endif
14     write ( „Die Provision dieses
15           Vertriebsmitarbeiters beträgt:“ )
16     write ( provision_max )
16 end program Provisionsberechnung
```

**Der Code enthält Datenflussanomalien in den Zeilen 6 und 12 (im Code fettgedruckt).**

**Welche Beispiele für Datenflussanomalien sind in diesen Zeilen zu finden?**

*(Fortsetzung auf der nächsten Seite)*

**Wahlen Sie genau EINE Antwort aus.**

a)	Zeile 6: der Variablen "summe" wird vor ihrer Verwendung kein Wert zugewiesen Zeile 12: Die Variable "provision_min" ist definiert, wird aber anschließend nicht verwendet	<input checked="" type="checkbox"/>
b)	Zeile 6: der Variablen "summe" wird ein ungultiger Wert zugewiesen Zeile 12: Die Variable "provision_min" wird vor ihrer Verwendung neu definiert	<input type="checkbox"/>
c)	Zeile 6: Variable "summe" ist auerhalb des Bereichs Zeile 12: der hart codierte" Wert "0.15" sollte nicht verwendet werden	<input type="checkbox"/>
d)	Zeile 6: die Variable "nummer" ist nicht definiert Zeile 12: die Variable "summe" wird vor ihrer Verwendung neu definiert	<input type="checkbox"/>

Begrundung:

a) **KORREKT – Anomalien:**

**summe: In Zeile 6 verwendet, bevor sie definiert wird (ku-Anomalie)**

**provision\_min: in Zeile 12 definiert und anschließend nicht verwendet (dk-Anomalie)**

b) FALSCH

c) FALSCH

d) FALSCH

<b>Frage 20</b>	<b>TTA-3.2.2</b>	<b>K2</b>	<b>Punkte 1.0</b>
-----------------	------------------	-----------	-------------------

**Welche der folgenden Aussagen uber die Datenflussanalyse ist zutreffend?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Eine Art der statischen Analyse, die auf einer Darstellung von eindeutigen Pfaden zur Ausfuhrung einer Komponente oder eines Systems basiert.	<input type="checkbox"/>
b)	Wenn eine Variable nochmal initialisiert wird, bevor sie zwischenzeitlich verwendet wurde, spricht man von einer dr-Anomalie.	<input type="checkbox"/>
c)	Mit der Datenflussanalyse lassen sich verdachtige Bereiche des Codes identifizieren, die zu moglichen Effizienzproblemen fuhren konnen.	<input checked="" type="checkbox"/>
d)	Die Datenflussanalyse sollte nach einem Codereview verwendet werden, um auszuschließen, dass Kontroll- und Datenflussanomalien unentdeckt bleiben.	<input type="checkbox"/>

TTA-3.2.2 (K2) Erklaren, wie die Datenflussanalyse verwendet wird, um zu ermitteln, ob der Programmcode Datenflussanomalien aufweist

Begrundung:

- a) FALSCH – Bei der Aussage handelt es sich um die Definition der Kontrollflussanalyse (vgl. Glossar).
- b) FALSCH – Die Aussage ist die Definition einer dd-Anomalie und nicht einer dr-Anomalie. Vgl. Lehrplan, Abschnitt 3.3.2, 6. Listenpunkt.
- c) **KORREKT – Bei der Datenflussanalyse werden solche Verwendungen hufig als verdachtig bzw. "fehlertrachtig" bezeichnet. Auch wenn diese Verwendung der Variablenzuweisung (z.B. Deklaration von Variablen ohne Verwendung) zulassig ist, kann dies spater zu Problemen mit der Effizienz des Programms fuhren.**
- d) FALSCH – Der Datenflussanalyse sollte VOR einem Codereview durchgefuhrt werden, um fehlertrachtige Stellen im Quelltext aufzufinden; die dann situationsbedingt im Codereview bewertet werden.

Frage 21	TTA-3.2.3	K3	Punkte 1.0
----------	-----------	----	------------

Nachfolgend sehen Sie den Pseudo-Code fur ein Programm namens TRICKY.

```
00  program TRICKY
01  var1, var2, var3 : integer
02  begin
03      read ( var2 )
04      read ( var1 )
05      while var2 < 10 loop
06          var3 = var2 + var1
07          var2 = 4
08          var1 = var2 + 1
09          print ( var3 )
10          if var1 = 5 then
11              print ( var1 )
12          else
13              print ( var1+1 )
14          endif
15          var2 = var2 + 1
16      endloop
17      write ( „Mensch, das war knifflig!“ )
18      write ( „Die Antwort lautet aber...“ )
19      write ( var2+var1 )
20  end program TRICKY
```

In welchem Bereich konnte der Einsatz der statischen Analyse die Wartbarkeit des Programms am besten verbessern?

Wahlen Sie genau EINE Antwort aus.

a)	Umstrukturierung des Codes	<input type="checkbox"/>
b)	Reduzieren der Programmkopplung	<input type="checkbox"/>
c)	Erhohen der Anzahl der Kommentare	<input checked="" type="checkbox"/>
d)	Verbessern der Einruckung des Codes	<input type="checkbox"/>

TTA-3.2.3 (K3) Möglichkeiten vorschlagen, wie die Wartbarkeit von Programmcode durch statische Analyse verbessert werden kann

Begründung:

- a) FALSCH – Der Code ist klar mit Steuerelementen strukturiert (z. B. Schleife, if-then-else). Es ist unwahrscheinlich, dass die statische Analyse Verbesserungen an der Kontrollstruktur identifizieren kann.
- b) FALSCH – Es sind keine globalen Variablen definiert und es werden keine anderen Programme aufgerufen. Die Kopplung ist kein Verbesserungsgebiet.
- c) KORREKT – Die statische Analyse kann mit Werkzeugunterstützung auch die Einhaltung von Programmierkonventionen und -richtlinien im vorhandenen Programmcode verifizieren (siehe Lehrplan A4Q TF4D 2021 3.3.3). Dazu gehört auch die Kommentierung. Da das Programm überhaupt keine Kommentare hat, würde dies als Bereich zur Verbesserung der Wartbarkeit des Codes hervorgehoben werden.
- d) FALSCH – Die statische Analyse kann Einrückungsregeln anwenden, aber im Fall des TRICKY-Programms ist eine ausreichende Einrückung vorhanden.

Frage 22	TTA-3.2.4	K2	Punkte 1.0
----------	-----------	----	------------

**Welche der folgenden Antworten beschreibt eine Moglichkeit der Verwendung von Aufrufgraphen zur Bestimmung der Anforderungen fur den Integrationstest?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Zur Ermittlung, wie viele Stellen in der Software eine Methode oder eine Funktion aufrufen.	<input type="checkbox"/>
b)	Zur Ermittlung, wie viele Stellen in der Software ein Modul oder eine Funktion aufrufen.	<input checked="" type="checkbox"/>
c)	Bei der Bestimmung der bedingten und bedingungslosen Modulaufrufe zur Performanzanalyse.	<input type="checkbox"/>
d)	Bei der Erkennung von Bereichen, die auf mogliche Speicherlecks untersucht werden sollten.	<input type="checkbox"/>

Begrundung:

- a) FALSCH – Dafur konnen Aufrufgraphen zwar verwendet werden, allerdings in Zusammenhang mit dem Komponententest, nicht mit dem Integrationstest (gema Lehrplan).
- b) KORREKT – Richtige Antwort gema Lehrplan A4Q TF4D 2021 Abschnitt 3.3.4**
- c) FALSCH – Die Bestimmung der bedingten und bedingungslosen Aufrufe kann im Integrationstest eingesetzt werden, aber ihre Verwendung fur die Performanzanalyse hat nichts mit Integration zu tun.
- d) FALSCH – Aufrufgraphen geben keinen Aufschluss daruber, welche Bereiche Speicherlecks enthalten konnen.

Frage 23	TTA-3.2.4	K2	Punkte 1.0
----------	-----------	----	------------

**Welche der folgenden Aussagen bezuglich Aufrufgraphen ist zutreffend?**

**Wahlen Sie genau eine Option aus! (1 aus 4)**

a)	Aufrufgraphen erlauben die Ermittlung der Verschachtelungstiefe der einzelnen Komponenten.	<input type="checkbox"/>
b)	Aufrufgraphen konnen verwendet werden, um Tests zu entwerfen, die ein bestimmte Module bzw. Funktionen aufrufen.	<input checked="" type="checkbox"/>
c)	Aufrufgraphen konnen bei der Planung der Komponententests verwendet werden, um die Komponenten zu identifizieren, die auf effektive Weise gleichzeitig zu integrieren sind.	<input type="checkbox"/>
d)	Aufrufgraphen erlauben die Identifikation von nicht ausgefuhrten Datenflussen der Komponente in einer Integration.	<input type="checkbox"/>

TTA-3.2.4 (K2) Den Einsatz von Aufrufgraphen fur die Bestimmung von Teststrategien fur den Integrationstest erklaren

Begrundung: (nach Lehrplan A4Q TF4D 2021)

- a) FALSCH – Die einzelnen Komponenten sind Knoten im Aufrufgraphen (siehe Lehrplan 3.3.4 erster Absatz). Deshalb kann ihre Verschachtelungstiefe nicht aus dem Graphen ermittelt werden.
- b) KORREKT – Aufrufgraphen konnen verwendet werden, um Tests zu entwerfen, die ein bestimmtes Modul oder System aufrufen. (siehe Lehrplan 3.3.4; dritter Absatz; erster Stichpunkt)**
- c) FALSCH – Die einzelnen Komponenten sind Knoten im Aufrufgraphen (siehe Lehrplan 3.3.4 erster Absatz). Aufrufgraphen werden sinnvollerweise nicht bei der Planung der Komponententests, sondern bei der Planung der Integrationstest verwendet.
- d) FALSCH – Die interne Struktur einer Komponente kann zwar durch einen Aufrufgraphen sichtbar gemacht werden, aber Datenflusse einer Komponente werden nicht ausgefuhrt.

---

## Fragen zum Thema "Testverfahren"

---

Frage 24	Keywords	K1	Punkte 1.0
----------	----------	----	------------

**Was ist checklistenbasiertes Testen?**

**Wählen Sie genau EINE Antwort aus.**

a)	Ein Testverfahren, bei dem Testfälle auf Basis des Wissens der Tester über frühere Fehler oder aus allgemeinem Wissen über Fehlerwirkungen abgeleitet werden.	<input type="checkbox"/>
b)	Ein Testverfahren, das auf einer Analyse der Spezifikation einer Komponente oder eines Systems basiert.	<input type="checkbox"/>
c)	Ein erfahrungsbasiertes Testverfahren, bei dem der erfahrene Tester z. B. eine Liste von Kontrollpunkten nutzt, welche beachtet, überprüft oder in Erinnerung gerufen werden müssen.	<input checked="" type="checkbox"/>
d)	Ein Testansatz, bei dem die Tester dynamisch Tests entwerfen und durchführen, basierend auf ihrem Wissen, der Erkundung des Testelements und dem Ergebnis früherer Tests.	<input type="checkbox"/>

FL-4.x (K1) Schlüsselbegriffe

Begründung: (nach Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Das ist die Definition für intuitive Testfallermittlung aus dem Glossar.
- b) FALSCH – Das ist die Definition für Black-Box-Testverfahren aus dem Glossar.
- c) **KORREKT – Basierend auf der Definition für checklistenbasiertes Testen aus dem Glossar.**
- d) FALSCH – Das ist (bis auf die Erwähnung des Testers) die Definition für exploratives Testen aus dem Glossar.



Frage 25	Keywords	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen beschreibt AM BESTEN exploratives Testen?**

**Wählen Sie genau EINE Antwort aus.**

a)	Eine Testvorgehensweise/ein Testansatz, bei der eine intensive Untersuchung des Hintergrunds des Testobjekts dazu genutzt wird, mögliche Schwachstellen zu identifizieren, die durch Testfälle untersucht werden.	<input type="checkbox"/>
b)	Eine Testvorgehensweise/ein Testansatz bei dem die Tester, basierend auf ihrem Wissen, der Erkundung des Testelements und dem Ergebnis früherer Tests, dynamisch Tests entwerfen und durchführen.	<input checked="" type="checkbox"/>
c)	Eine Testvorgehensweise/ein Testansatz, bei dem die Testaktivitäten – insbesondere Testanalyse und Testentwurf – als unterbrechungsfreie Sitzungen geplant werden, oft in Verbindung mit checklisten-basiertem Testen.	<input type="checkbox"/>
d)	Eine Testvorgehensweise/ein Testansatz, das auf der Erfahrung, dem Wissen und der Intuition des Testers basiert.	<input type="checkbox"/>

Schlüsselbegriff: exploratives Testen

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

a) FALSCH – Exploratives Testen wird meistens bei kurzfristigen Zeitvorgaben durchgeführt, daher sind intensive Untersuchungen des Hintergrunds des Testobjekts eher ungewöhnlich.

**b) KORREKT – Ein Testansatz, bei dem die Tests dynamisch entworfen und ausgeführt werden, basierend auf Wissen, der Erkundung des Testelements und den Ergebnissen früherer Tests. (Definition Glossar)**

c) FALSCH – Basiert auf der Definition des Glossars für sitzungsbasiertes Testen, jedoch wurde Testdurchführung durch Testanalyse ersetzt.

d) FALSCH – Basiert auf der Definition des Glossars für erfahrungsbasiertes Testen.

Frage 26	Keywords	K1	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen ist eine korrekte Definition fur den modifizierter Bedingungs-/Entscheidungstest?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Ein White-Box-Testverfahren, bei dem Testfalle so entworfen werden, dass sich Ergebnisse atomarer Teilbedingungen unabhangig voneinander auf das Gesamtergebnis der Entscheidung auswirken.	<input checked="" type="checkbox"/>
b)	Ein White-Box-Testverfahren, bei dem Testfalle im Hinblick auf die Ausfuhrung von Ergebniskombinationen von atomaren Bedingungen entworfen werden.	<input type="checkbox"/>
c)	Ein White-Box-Testverfahren, bei dem die Testfalle im Hinblick auf die Ausfuhrung von Bedingungs- und Entscheidungsergebnissen entworfen werden.	<input type="checkbox"/>
d)	Ein White-Box-Testverfahren, bei dem Testfalle im Hinblick auf die Ausfuhrung von Entscheidungsausgangen entworfen werden.	<input type="checkbox"/>

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

a) **KORREKT** – Basierend auf der Definition modifizierter Bedingungs- /Entscheidungstests aus dem Glossar.

b) FALSCH – Dies ist die Definition des Mehrfachbedingungstests aus dem Glossar.

c) FALSCH – Dies ist die Definition des Bedingungs- /Entscheidungstests aus dem Glossar.

d) FALSCH – Dies ist die Definition des Entscheidungstest aus dem Glossar.

Frage 27	FL-4.1.1	K2	Punkte 1.0
----------	----------	----	------------

Welche der folgenden Zuordnungen von Beschreibungen zu verschiedenen Kategorien von Testverfahren trifft AM BESTEN zu?

1. Uberdeckung wird auf Basis einer ausgewahlten Struktur des Testobjektes gemessen.
2. Verarbeitung innerhalb des Testobjekts wird uberpruft.
3. Tests basieren auf der Wahrscheinlichkeit von Fehlerzustanden und deren Verteilung.
4. Abweichungen von Anforderungen werden uberpruft.
5. User-Stories werden als Testbasis herangezogen.

Verwendete Notation fur die folgenden 4 Optionen:

- Black** – Black-Box-Testverfahren  
**White** – White-Box-Testverfahren  
**Erfahrung** – erfahrungsbasiertes Testverfahren

Wahlen Sie genau EINE Antwort aus.

a)	Black – 4, 5; White – 1, 2; Erfahrung – 3	<input checked="" type="checkbox"/>
b)	Black – 3; White – 1, 2; Erfahrung – 4, 5	<input type="checkbox"/>
c)	Black – 4; White – 1, 2; Erfahrung – 3, 5	<input type="checkbox"/>
d)	Black – 1, 3, 5; White – 2; Erfahrung – 4	<input type="checkbox"/>

FL-4.1.1 (K2) Die Eigenschaften, Gemeinsamkeiten und Unterschiede zwischen Black-Box-Testverfahren, White-Box-Testverfahren und erfahrungsbasierten Testverfahren erklären können

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

Die korrekte Verknüpfung der Beschreibungen zu den verschiedenen Kategorien von Testverfahren entsprechend des Lehrplans, Kapitel 4 ist:

- Black-Box-Testverfahren:  
Abweichungen von Anforderungen werden überprüft (4)  
User-Stories werden als Testbasis herangezogen (5)
- White-Box-Testverfahren:  
Überdeckung wird auf Basis einer ausgewählten Struktur des Testobjektes gemessen (1)  
Verarbeitung innerhalb des Testobjektes wird überprüft (2)
- erfahrungsbasiertes Testverfahren:  
Tests basieren auf der Wahrscheinlichkeit von Fehlerzuständen und deren Verteilung (3)

Folglich ist Antwort a) korrekt.

Frage 28	FL-4.2.1	K3	Punkte 1.0
----------	----------	----	------------

Ein Gerät zur Messung des täglichen Strahlungseinfalls für Pflanzen ermittelt einen Einstrahlungswert für Sonnenschein. Dieser ergibt sich aus der Kombination der Anzahl der Stunden, in denen eine Pflanze der Sonne ausgesetzt ist (unter 3 Stunden, 3 bis 6 Stunden, über 6 Stunden) und der durchschnittlichen Intensität des Sonnenscheins (sehr niedrig, niedrig, mittel, hoch). Die folgenden Testfälle existieren bereits:

	Dauer (Stunden)	Intensität	Einstrahlung
T1	1,5	sehr niedrig	10
T2	7,0	mittel	60
T3	0,5	sehr niedrig	10

Wie viele Testfälle müssen mindestens noch erzeugt werden, um eine vollständige Überdeckung ALLER GÜLTIGEN Eingabe-Äquivalenzklassen zu gewährleisten?

Wählen Sie genau EINE Antwort aus.

a)	1	<input type="checkbox"/>
b)	2	<input checked="" type="checkbox"/>
c)	3	<input type="checkbox"/>
d)	4	<input type="checkbox"/>

FL-4.2.1 (K3) Die Äquivalenzklassenbildung anwenden können, um Testfälle aus vorgegebenen Anforderungen abzuleiten

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

Die folgenden Äquivalenzklassen können identifiziert werden:

- Dauer
  1. Unter 3 Stunden
  2. 3 – 6 Stunden
  3. Über 6 Stunden
- Intensität
  4. sehr niedrig
  5. niedrig
  6. mittel
  7. hoch

Die existierenden Testfälle decken die folgenden gültigen Eingabe- Äquivalenzklassen ab:

T1	1,5 (1)	sehr niedrig (4)
T2	7,0 (3)	mittel (6)
T3	0,5 (1)	sehr niedrig (4)

Folglich sind die fehlenden gültigen Eingabe- Äquivalenzklassen (2), (5) und (7).

Sie können mit zwei zusätzlichen Testfällen abgedeckt werden, da (2) sowohl mit (5) als auch mit (7) kombiniert werden kann.

Folglich ist Antwort b) richtig.

<b>Frage 29</b>	<b>FL-4.2.2</b>	<b>K3</b>	<b>Punkte 1.0</b>
-----------------	-----------------	-----------	-------------------

**Eine Smart-Home-App misst die durchschnittliche Temperatur im Haus während der vergangenen Woche und gibt den Bewohnern basierend auf diesem Wert Informationen zur Umweltfreundlichkeit ihres Verhaltens. Das Feedback für die verschiedenen Durchschnittstemperaturbereiche (gerundet auf die nächsten ganzen °C) soll lauten:**

<b>Bis zu 10°C</b>	<b>Eiskalt!</b>
<b>11°C bis 15°C</b>	<b>Ganz schön schattig!</b>
<b>16°C bis 19°C</b>	<b>Cool!</b>
<b>20°C bis 22°C</b>	<b>Zu warm!</b>
<b>Über 22°C</b>	<b>Sauna!</b>

**Welches der folgenden Testsets liefert die höchste Überdeckung von Grenzwerten, wenn die Grenzwertanalyse ausschließlich unter Verwendung der Minimal- und Maximalwerte verwendet wird?**

**Wählen Sie genau EINE Antwort aus.**

a)	0°C,	11°C,	20°C,	22°C,	23°C	<input type="checkbox"/>
b)	9°C,	15°C,	19°C,	23°C,	100°C	<input type="checkbox"/>
c)	10°C,	16°C,	19°C,	22°C,	23°C	<input checked="" type="checkbox"/>
d)	14°C,	15°C,	18°C,	19°C,	21°C 22°C	<input type="checkbox"/>

FL-4.2.2 (K3) Die Grenzwertanalyse anwenden können, um Testfälle aus vorgegebenen Anforderungen abzuleiten

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

Bei den angegebenen Eingabe-Äquivalenzklassen ergibt das Zwei-Punkt-Grenzwertverfahren folgende 8 Überdeckungselemente:

10°C, 11°C, 15°C, 16°C, 19°C, 20°C, 22°C, 23°C.

Die Überdeckung durch die Optionen ist folglich jeweils:

- a) 4 von 8 (11, 20, 22 und 23).
- b) 3 von 8 (15, 19 und 23).
- c) 5 von 8 (10, 16, 19, 22 und 23).
- d) 3 von 8 (15, 19 und 22)

Folglich ist Option c) richtig.



<b>Frage 30</b>	<b>FL-4.2.3</b>	<b>K3</b>	<b>Punkte 1.0</b>
-----------------	-----------------	-----------	-------------------

**Den Beschaftigten einer Firma wird nur dann eine Jahrespremie ausbezahlt, wenn sie langer als ein Jahr im Unternehmen beschaftigt sind und ein Ziel erreichen, das vorher individuell vereinbart wurde.**

**Dieser Sachverhalt lasst sich in einer Entscheidungstabelle darstellen:**

Testfall-ID		T1	T2	T3	T4
Bedingung1	Beschaftigung langer als ein Jahr?	JA	NEIN	NEIN	JA
Bedingung2	Ziel vereinbart?	NEIN	NEIN	JA	JA
Bedingung3	Ziel erreicht?	NEIN	NEIN	JA	JA
Aktion	Auszahlung der Jahrespremie?	NEIN	NEIN	NEIN	JA

**Welcher der folgenden Testfalle beschreibt eine in der Praxis vorkommende Situation und fehlt in der oben aufgefuhrten Entscheidungstabelle?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Bedingung1 = JA, Bedingung2 = NEIN, Bedingung3 = JA, Aktion = NEIN	<input type="checkbox"/>
b)	Bedingung1 = JA, Bedingung2 = JA, Bedingung3 = NEIN, Aktion = JA	<input type="checkbox"/>
c)	Bedingung1 = NEIN, Bedingung2 = NEIN, Bedingung3 = JA, Aktion = NEIN	<input type="checkbox"/>
d)	Bedingung1 = NEIN, Bedingung2 = JA, Bedingung3 = NEIN, Aktion = NEIN	<input checked="" type="checkbox"/>

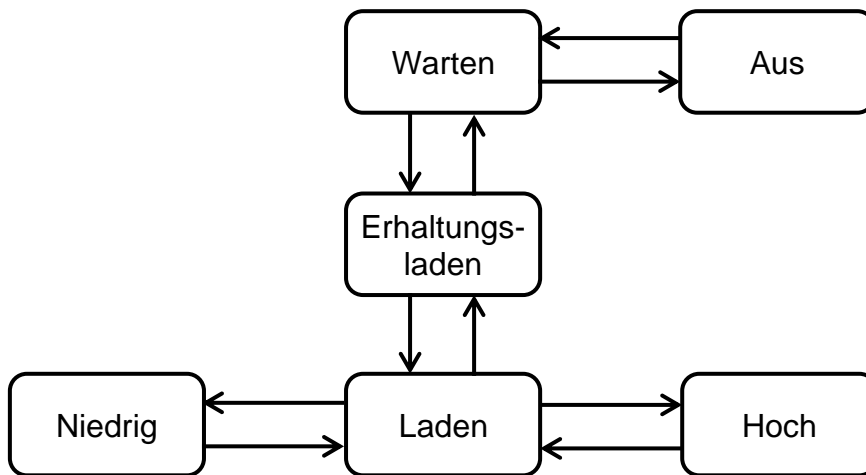
FL-4.2.3 (K3) Entscheidungstabellentests anwenden können, um Testfälle aus vorgegebenen Anforderungen abzuleiten.

Begründung (Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Wenn kein Ziel vereinbart wurde, kann das nicht vereinbarte Ziel auch nicht erreicht werden. Daher handelt es sich nicht um ein in der Praxis vorkommendes Szenario.
- b) FALSCH – Der Testfall ist fachlich falsch, da unter diesen Bedingungen keine Prämie gezahlt wird, weil das vereinbarte Ziel nicht erreicht wurde.
- c) FALSCH – Wenn kein Ziel vereinbart wurde, kann das nicht vereinbarte Ziel auch nicht erreicht werden. Daher handelt es sich nicht um ein in der Praxis vorkommendes Szenario. (Vgl. Antwort a)
- d) KORREKT – Der Testfall beschreibt die Situation, dass sowohl die zu kurze Beschäftigungszeit als auch das Nichterreichen des vereinbarten Ziels zur Nichtauszahlung der Prämie führen. Diese Situation kann in der Praxis vorkommen, fehlt aber in der Entscheidungstabelle.

Frage 31	FL-4.2.4	K3	Punkte 1.0
----------	----------	----	------------

Gegeben sei folgendes Zustandsubergangsdiagramm fur die Software eines Batterie-Ladegerates:



Welche der folgenden Zustandsubergangs-Sequenzen ergibt die hochste uberdeckung der Zustandsubergange?

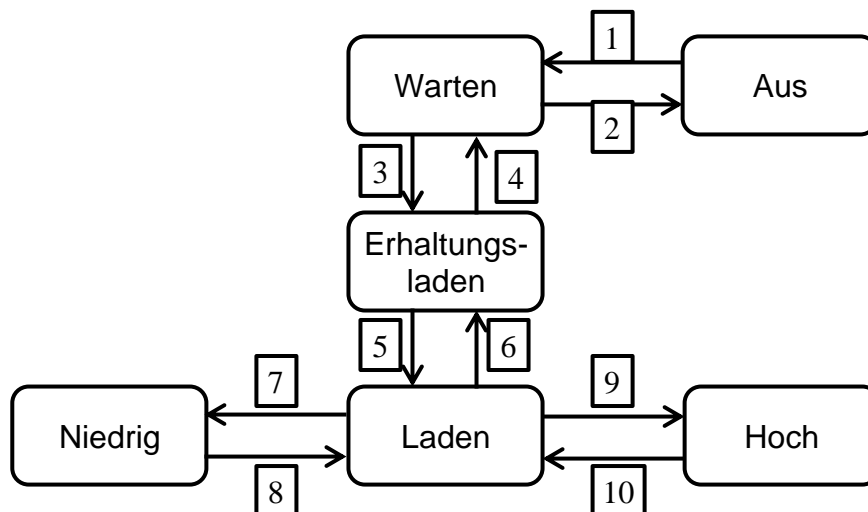
Wahlen Sie genau **EINE** Antwort aus.

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

FL-4.2.4 (K3) Zustandsubergangstests anwenden konnen, um Testfalle aus vorgegebenen Anforderungen abzuleiten

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

Im folgenden Zustandsdiagramm sind die Zustandsubergange nummeriert:



Die Sequenzen in den Antwortoptionen erreichen folgende uberdeckung der Zustandsubergange:

- a) FALSCH – Aus → (1) Warten → (2) Aus → (1) Warten → (3) Erhaltungsladen → (5) Laden → (9) Hoch → (10) Laden → (7) Niedrig. Das sind  $7/10=70\%$ .
- b) FALSCH – Warten → (3) Erhaltungsladen → (4) Warten → (2) Aus → (1) Warten → (3) Erhaltungsladen → (5) Laden → (7) Niedrig → (8) Laden. Das sind  $7/10=70\%$ .
- c) FALSCH – Hoch → (10) Laden → (7) Niedrig → (8) Laden → (6) Erhaltungsladen → (4) Warten → (3) Erhaltungsladen → (4) Warten → (3) Erhaltungsladen. Das sind  $6/10=60\%$ .
- d) **KORREKT – Warten → (3) Erhaltungsladen → (5) Laden → (9) Hoch → (10) Laden → (6) Erhaltungsladen → (4) Warten → (2) Aus → (1) Warten. Das sind  $8/10=80\%$ .**

**Antwortoption d) erreicht mit 80% die hochste uberdeckung der Zustandsubergange.**

Frage 32	FL-4.2.5	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen beschreibt AM BESTEN, wie Testfälle aus Anwendungsfällen entworfen werden?**

**Wählen Sie genau EINE Antwort aus.**

a)	Testfälle werden entworfen, um das im Anwendungsfall definierte grundlegende, Sonder- und Fehlerbehandlungs-Verhalten des Systems in Interaktion mit den Akteuren auszuführen.	<input checked="" type="checkbox"/>
b)	Testfälle werden entworfen, indem die vom Anwendungsfall betroffenen Komponenten identifiziert und Integrationstests erstellt werden, welche die Interaktionen dieser Komponenten ausführen.	<input type="checkbox"/>
c)	Testfälle werden entworfen, indem die Interaktionen der Akteure mit dem System analysiert werden, um sicherzustellen, dass die Benutzungsschnittstelle des Systems leicht bedienbar ist.	<input type="checkbox"/>
d)	Testfälle werden entworfen, mit denen alle Entscheidungspunkte im Geschäftsprozess des Anwendungsfalls ausgeführt werden, um eine	<input type="checkbox"/>

FL-4.2.5 (K2) Erklären können, wie man Testfälle aus einem Anwendungsfall ableitet

Begründung: (Lehrplan A4Q TF4D 2021; Glossar)

a) **KORREKT** – Im Lehrplan unter 4.2.5 Anwendungsfallbasierter Test steht: *Jeder Anwendungsfall definiert ein bestimmtes Verhalten, das ein Objekt in Zusammenarbeit mit einem oder mehreren Akteuren ausführen kann. Weiter unten ist zu lesen: Ein Anwendungsfall besteht aus mehreren möglichen Varianten seines grundlegenden Verhaltens, was u. a. Sonder- und Fehlerbehandlungen einschließt (Antwort- und Wiederherstellungsmechanismen des Systems nach Programmier-, Anwendungs- und Kommunikationsfehler, die z. B. zu Fehlermeldungen führen). Tests werden entworfen, um das definierte Verhalten nachzuweisen (grundlegendes, außergewöhnliches oder alternatives Verhalten und die Fehlerbehandlungsroutinen).*

b) **FALSCH** – Anwendungsfälle spezifizieren normalerweise Anforderungen und schließen daher nicht die Komponenten ein, die sie implementieren.

c) **FALSCH** – Anwendungsfallbasierte Tests führen zwar Interaktionen des Systems mit einem oder mehreren Akteuren aus. Aber sie konzentrieren sich auf die Funktionalität und betrachten nicht die leichte Bedienbarkeit der Benutzungsschnittstelle.

d) **FALSCH** – Tests decken zwar die Ablaufpfade des Anwendungsfalls ab, es geht jedoch nicht um eine Entscheidungsüberdeckung in diesen Pfaden, und bestimmt nicht in den Kontrollfluss im Geschäftsprozess.

Frage 33	FL-4.3.1	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Beschreibungen der Anweisungsuberdeckung ist korrekt?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Die Anweisungsuberdeckung ist ein Ma fur die Anzahl der Quellcodezeilen (ohne Kommentare), die im Test ausgefuhrt wurden.	<input type="checkbox"/>
b)	Die Anweisungsuberdeckung ist ein Ma fur den prozentualen Anteil der ausfuhrbaren Anweisungen im Quellcode, die im Test ausgefuhrt wurden.	<input checked="" type="checkbox"/>
c)	Die Anweisungsuberdeckung ist ein Ma fur den prozentualen Anteil der Quellcodezeilen (ohne Kommentare), die im Test ausgefuhrt wurden.	<input type="checkbox"/>
d)	Die Anweisungsuberdeckung ist ein Ma fur die Anzahl der ausfuhrbaren Anweisungen im Quellcode, die im Test ausgefuhrt wurden.	<input type="checkbox"/>

FL-4.3.1 (K2) Anweisungsuberdeckung erklaren konnen

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

Glossareintrag Anweisungsuberdeckung: Der Anteil der Anweisungen, die durch eine Testsuite ausgefuhrt wurden, bezogen auf alle Anweisungen.

Lehrplan A4Q TF4D 2021 Kapitel 4.3.1: Anweisungstests untersuchen die (potenziell) ausfuhrbaren Anweisungen im Code. Die uberdeckung wird an der Anzahl der im Test ausgefuhrten Anweisungen dividiert durch die Gesamtzahl aller ausfuhrbaren Anweisungen insgesamt im Testobjekt gemessen – ublicherweise als Prozentsatz dargestellt.

- a) FALSCH – Die Anweisungsuberdeckung bezieht sich auf durch Tests ausfuhrbare Anweisungen. In einer Zeile konnen mehrere solcher Anweisungen stehen, und eine Anweisung kann sich uber mehrere Zeilen erstrecken.
- b) **KORREKT – Der prozentuale Anteil der ausfuhrbaren Anweisungen ist der Anteil im Test ausgefuhrter Anweisungen bezogen auf alle Anweisungen, also (Anzahl durch die Testsuite ausgefuhrter Anweisungen / Anzahl alle Anweisungen) x 100%.**
- c) FALSCH – Die Anweisungsuberdeckung bezieht sich nicht auf Quellcodezeilen, sondern auf durch Tests ausfuhrbare Anweisungen, s. Begrundung zu a).
- d) FALSCH – Die Anweisungsuberdeckung bezieht sich nicht auf die absolute Anzahl der von der Testsuite ausgefuhrten Anweisungen, sondern auf deren Anteil bezogen auf alle ausfuhrbaren Anweisungen.

Frage 34	FL-4.3.1	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Aussagen ist eine Beschreibung für Anweisungsüberdeckung?**

**Wählen Sie genau EINE Antwort aus.**

a)	Es handelt sich um eine Metrik zur Berechnung und Messung des prozentualen Anteils der ausgeführten Testfälle.	<input type="checkbox"/>
b)	Es handelt sich um eine Metrik, die den Prozentsatz der ausgeführten Anweisungen im Code angibt.	<input checked="" type="checkbox"/>
c)	Es handelt sich um eine Metrik zur Berechnung und Messung der Anzahl von Anweisungen im Code, die durch Testfälle ausgeführt wurden, die keine Fehlerwirkung aufgedeckt haben.	<input type="checkbox"/>
d)	Es handelt sich um eine Metrik, die eine wahr/falsch-Bestätigung gibt, ob alle Anweisungen abgedeckt sind oder nicht.	<input type="checkbox"/>

FL-4.3.1 (K2) Anweisungsüberdeckung erklären können.

Begründung (Lehrplan A4Q TF4D 2021, V.3.1; Glossar)

a) FALSCH – Anweisungsüberdeckung misst den prozentualen Anteil der durch Testfälle ausgeführten (überdeckten) Anweisungen und hat keinen Bezug zur Zahl ausgeführten Testfälle.

b) KORREKT – Lehrplan A4Q TF4D 2021; Abschnitt 4.3.1: Der Anweisungstest führt die ausführbaren Anweisungen des Codes aus. Die Anweisungsüberdeckung wird als die Anzahl der von den Tests ausgeführten Anweisungen gemessen, geteilt durch die Gesamtzahl der ausführbaren Anweisungen im Testobjekt, normalerweise ausgedrückt in Prozent.

c) FALSCH – Die Abdeckung misst nicht bestanden/fehlgeschlagen bzw. Anweisungsüberdeckung berücksichtigt nicht, ob ein Test erfolgreich ausgeführt wurde.

d) FALSCH – Anweisungsüberdeckung ist eine Metrik, die Prozentwerte liefert und keine wahr/falsch-Aussage macht. Letzteres gilt nur für die Forderung nach 100% Anweisungsüberdeckung.

Frage 35	FL-4.3.2	K2	Punkte 1.0
----------	----------	----	------------

**Die folgende Aussage bezieht sich auf Entscheidungsuberdeckung:**

**"Wenn der Code nur aus einer einzigen IF-Anweisung (also keinen Schleifen oder CASE-Anweisungen) besteht und auch sonst durch den Test nicht geschachtelt aufgerufen wird, dann wird bei einem einzelnen Testfall, der ausgefuhrt wird, eine Entscheidungsuberdeckung von 50% erreicht."**

**Welcher der folgenden Aussagen ist zutreffend?**

**Wahlen Sie genau EINE Antwort aus.**

a)	Die Aussage ist wahr. Ein einzelner Testfall erzielt eine 100% Anweisungsuberdeckung und daher 50% Entscheidungsuberdeckung.	<input type="checkbox"/>
b)	Die Aussage ist wahr. Bei einem einzelnen Testfall ist der Entscheidungsausgang der IF-Anweisung entweder wahr oder falsch.	<input checked="" type="checkbox"/>
c)	Die Aussage ist falsch. Ein einzelner Testfall kann in diesem Fall nur eine Entscheidungsuberdeckung von 25% garantieren.	<input type="checkbox"/>
d)	Die Aussage ist falsch. Die Aussage ist zu weit gefasst. Sie kann abhangig von der getesteten Software richtig sein oder nicht.	<input type="checkbox"/>

FL-4.3.2 (K2) Entscheidungsuberdeckung erklaren konnen.

Begrundung: (nach Lehrplan A4Q TF4D 2021; Glossar)

- a) FALSCH – Wahrend die gemachte Aussage wahr ist, ist die Erklarung falsch; weil die Beziehung zwischen Anweisungs- und Entscheidungsuberdeckung falsch dargestellt ist. (vgl. Lehrplan A4Q TF4D 2021; Abschnitt 4.3.3)
- b) KORREKT – Da jeder Testfall dazu fuhrt, dass das Ergebnis der IF-Anweisung entweder WAHR oder FALSCH ist, haben wir definitiv 50% Entscheidungsuberdeckung erreicht. (vgl. Lehrplan A4Q TF4D 2021; Abschnitt 4.3.2, 2. Absatz)**
- c) FALSCH – Ein einzelner Testfall kann mehr als 25% Entscheidungs-uberdeckung erreichen; bei der obigen Aussage sind es 50% Entscheidungsuberdeckung (vgl. Begrundung zu Antwort b)
- d) FALSCH – Die obige Aussage ist konkret und immer wahr; weil durch jeden einzelnen Testfall immer eine Entscheidungsuberdeckung von 50% erreicht wird. (vgl. Begrundung zu Antwort b)



Frage 36	FL-4.3.2	K2	Punkte 1.0
----------	----------	----	------------

**Welche der folgenden Beschreibungen der Entscheidungsüberdeckung ist zutreffend?**

**Wählen Sie genau EINE Antwort aus.**

a)	Die Entscheidungsüberdeckung ist ein Maß für den prozentualen Anteil möglicher Pfade durch den Quellcode, die im Test ausgeführt wurden.	<input type="checkbox"/>
b)	Die Entscheidungsüberdeckung ist ein Maß für den prozentualen Anteil der Geschäftsabläufe durch die Komponente, die im Test ausgeführt wurden.	<input type="checkbox"/>
c)	Die Entscheidungsüberdeckung ist ein Maß für die „IF-Anweisungen“ im Quellcode, die im Test sowohl mit dem Ergebnis „WAHR“ als auch mit „FALSCH“ ausgeführt wurden.	<input type="checkbox"/>
d)	Die Entscheidungsüberdeckung ist ein Maß für den Anteil der Entscheidungsergebnisse, die im Test ausgeführt wurden.	<input checked="" type="checkbox"/>

FL-4.3.2 (K2) Entscheidungsuberdeckung erklaren konnen

Begrundung: (Lehrplan A4Q TF4D 2021; Glossar)

Lehrplan 4.3.2: Der uberdeckungsgrad wird gemessen anhand der Anzahl der Entscheidungsergebnisse, die durch die Tests ausgefuhrt werden, dividiert durch die Gesamtzahl an moglichen Entscheidungsergebnissen im Testobjekt – ublicherweise als Prozentsatz dargestellt.

- a) FALSCH – Ein Pfad durch den Quellcode ist ein moglicher Ablauf durch den Quellcode vom Eintrittspunkt zum Austrittspunkt, der eine Reihe von Entscheidungsergebnissen ausfuhren kann. Zwei verschiedene Pfade konnen bis auf einen die gleichen Entscheidungsergebnisse ausfuhren. Wird nur ein einziges Entscheidungsergebnis geandert, so wird ein anderer Pfad durchlaufen. Testfalle, die eine vollstandige Entscheidungsabdeckung erzielen, sind in der Regel eine winzige Teilmenge der Testfalle, die eine vollstandige Pfaduberdeckung erzielen wurden. In der Praxis haben die meisten nicht-trivialen Programme (und alle Programme mit unbeschrankten Schleifen, z. B. 'while'-Schleifen) eine potenziell unendliche Anzahl moglicher Pfade, sodass die Messung des abgedeckten Prozentsatzes praktisch unmoglich ist.
- b) FALSCH – Geschaftsablaufe konnen im anwendungsfallbasierten Test betrachtet werden. Sie werden jedoch nicht durch die Entscheidungsuberdeckung gemessen, auch wenn sie ein oder mehrere Entscheidungsergebnisse ausfuhren wurden.
- c) FALSCH – „IF-Anweisungen“ beinhalten zwar Entscheidungen, sind jedoch nicht die einzige Quelle fur Entscheidungen, da z. B. auch Schleifen oder „CASE-Anweisungen“ Entscheidungen beinhalten, aufgrund deren Ergebnis sie ausgefuhrt oder nicht ausgefuhrt werden.
- d) **KORREKT – S. Lehrplan-Auszug 4.3.2 oben.**

Frage 37	FL-4.3.3	K2	Punkte 1.0
----------	----------	----	------------

**Welche Aussage uber die Beziehung zwischen der Anweisungsuberdeckung und der Entscheidungsuberdeckung ist wahr?**

**Wahlen Sie genau EINE Antwort aus.**

a)	100% Entscheidungsuberdeckung schliet 100% Anweisungsuberdeckung ein.	<input checked="" type="checkbox"/>
b)	100% Anweisungsuberdeckung schliet 100% Entscheidungsuberdeckung ein.	<input type="checkbox"/>
c)	50% Entscheidungsuberdeckung schliet 50% Anweisungsuberdeckung ein.	<input type="checkbox"/>
d)	Entscheidungsuberdeckung kann nie 100% erreichen.	<input type="checkbox"/>

FL-4.3.3 (K2) Die Bedeutung von Anweisungs- und Entscheidungsuberdeckung erklaren konnen.

Begrundung (nach Lehrplan A4Q TF4D 2021, V.3.1; Glossar)

- a) **KORREKT** – Die Aussage ist wahr, weil die Ausfuhrung aller Entscheidungen zwangslaufig auch die Ausfuhrung aller Anweisungen bedingt. (siehe Lehrplan A4Q TF4D 2021; Abschnitt 4.3.3; 3. Absatz).
- b) **FALSCH** – Die Aussage ist falsch, weil die Ausfuhrung aller Anweisungen nicht zwangslaufig auch eine Ausfuhrung aller Entscheidungen bedingt (siehe Lehrplan A4Q TF4D 2021, Abschnitt 4.3.3; 3. Absatz).
- c) **FALSCH** – Die Aussage ist falsch, weil z. B. bei dem Code aus Aufgabe 21 in einem Zweig 3 Anweisungen und in dem anderen Zweig 1 Anweisung vorkommen kann - und sonst keine Anweisungen im Code vorkommen. Fuhrt nun ein Testfall den Zweig mit den 3 Anweisungen aus, ergibt sich 50% Entscheidungsuberdeckung und 75% Anweisungsuberdeckung, da 3 der 4 Anweisungen ausgefuhrt werden. (siehe Lehrplan A4Q TF4D 2021, Abschnitt 4.3.3).
- d) **FALSCH** – Die Aussage ist falsch, weil z. B. fur den Code von Aufgabe 21 zwei (2) Testfalle ausreichen, die einmal das Ergebnis der (einzigen) IF-Anweisung WAHR und einmal FALSCH machen. Dann liegt eine Entscheidungsuberdeckung von 2/2, also 100%, vor. (siehe Lehrplan A4Q TF4D 2021, Abschnitt 4.3.3).

Frage 38	TTA-2.2.1	K3	Punkte 1.0
----------	-----------	----	------------

**Betrachten Sie die vereinfachte Logik eines Teeautomaten:**

```
Teeautomat einschalten
IF ausreichend Wasser THEN
    Wasser kochen
    Tee hinzufugen
    Meldung "Milch?" anzeigen
    IF Milch = ja THEN
        Meldung "fettarm?" anzeigen
        IF fettarm = ja THEN
            Fettarme Milch hinzufugen
        ELSE
            Vollmilch hinzufugen
        ENDIF
    ENDIF
    Meldung "Zucker?" anzeigen
    IF Zucker = ja THEN
        Zucker hinzufugen
    ENDIF
    Umruhren
    3 Minuten warten
    Meldung "Bitte nehmen Sie Ihren Tee" anzeigen
ELSE
    Meldung "Bitte Wasser nachfullen" anzeigen
ENDIF
```

**Wie viele Testfalle wurden Sie entwerfen, um 100 % Anweisungsuberdeckung fur den Teeautomaten zu erzielen?**

**Wahlen Sie genau EINE Antwort aus.**

a)	3	<input checked="" type="checkbox"/>
b)	2	<input type="checkbox"/>
c)	5	<input type="checkbox"/>
d)	6	<input type="checkbox"/>

Begründung:

a) **KORREKT** – Die drei Testfälle sind durch die folgenden Eingaben definiert:

1. **Ausreichend Wasser, Milch, fettarm, Zucker**

2. **Ausreichend Wasser, Milch, nicht fettarm, Zucker oder kein Zucker**

3. **Nicht genügend Wasser**

b) FALSCH

c) FALSCH

d) FALSCH

Frage 39	TTA-2.3.1	K3	Punkte 1.0
----------	-----------	----	------------

Die vereinfachte Logik eines Programms wurde wie folgt spezifiziert:

```
Anweisung P
IF A THEN
    IF B THEN
        Anweisung Q
    ELSE
        Anweisung R
    ENDIF
ELSE
    Anweisung S
    IF C THEN
        Anweisung T
    ELSE
        Anweisung U
    ENDIF
ENDIF
Anweisung V
```

Wie viele Testfalle wurden Sie entwerfen, um 100%  
Entscheidungsuberdeckung zu erzielen?

Wahlen Sie genau EINE Antwort aus.

a)	2	<input type="checkbox"/>
b)	3	<input type="checkbox"/>
c)	4	<input checked="" type="checkbox"/>
d)	5	<input type="checkbox"/>

Begründung:

a) FALSCH

b) FALSCH

c) **KORREKT** – Die folgenden Bedingungen stellen sicher, dass alle Entscheidungsergebnisse getestet sind: 1) A, B 2) A, nicht B 3) nicht A, C 4) nicht A, nicht C

d) FALSCH

<b>Frage 40</b>	<b>TTA-2.4.1</b>	<b>K3</b>	<b>Punkte 1.0</b>
-----------------	------------------	-----------	-------------------

**Sie testen ein Blitzgerät für die Verkehrsüberwachung an einer Kreuzung. Es wurde festgelegt, dass ein Foto aufgenommen wird, wenn die Ampel rot ist (ROT) oder das Fahrzeug die Geschwindigkeit überschreitet (GESCHWINDIGKEIT) und die Vorderräder des Fahrzeugs die Linie überschreiten, die den Beginn der Kreuzung markiert (RÄDER).**

**Betrachten Sie die folgenden Sätze von Testwerten:**

1. ROT + GESCHWINDIGKEIT + RÄDER
2. ROT + GESCHWINDIGKEIT + keine RÄDER
3. ROT + keine GESCHWINDIGKEIT + RÄDER
4. ROT + keine GESCHWINDIGKEIT + keine RÄDER
5. kein ROT + GESCHWINDIGKEIT + RÄDER
6. kein ROT + GESCHWINDIGKEIT + keine RÄDER
7. kein ROT + keine GESCHWINDIGKEIT + RÄDER
8. kein ROT + keine GESCHWINDIGKEIT + keine RÄDER

**Angenommen, im Programmcode gilt folgende Logik:**

```

IF ((ROT ODER GESCHWINDIGKEIT) UND RÄDER) THEN
    FOTO AUFNEHMEN
ELSE
    KEIN FOTO AUFNEHMEN
    
```

**Unter Berücksichtigung dieser Informationen, welche Wertesätze ermöglichen die geringste Anzahl von Tests, um 100% modifizierte Bedingungs-/Entscheidungsüberdeckung zu erzielen?**

**Wählen Sie genau EINE Antwort aus.**

a)	1, 3 und 8.	<input type="checkbox"/>
b)	2 und 8.	<input type="checkbox"/>
c)	3, 4, 5 und 7.	<input checked="" type="checkbox"/>
d)	1, 5, 7 und 8.	<input type="checkbox"/>



Begründung:

- a) FALSCH – Dies deckt die Entscheidungsergebnisse ab, aber nicht die atomaren Bedingungen, die das Entscheidungsergebnis beeinflussen.
- b) FALSCH – Dies deckt die atomaren Bedingungen, die in das Entscheidungsergebnis einfließen, nicht ausreichend ab.

c) KORREKT – Diese Antwort liefert folgendes:

(W oder F) + W

(W oder F) + F

(F oder W) + W

(F oder F) + W

Damit werden alle Werte für die atomaren Bedingungen sowie alle Entscheidungsergebnisse mit der geringsten Anzahl von Tests getestet.

- d) FALSCH – Dies deckt die atomaren Bedingungen, die in das Entscheidungsergebnis einfließen, nicht ausreichend ab.



**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)



**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)



**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)



**Platz für Ihre Notizen:**

(werden bei der Korrektur weder gelesen noch bewertet)