

Beispielprüfung

iSAQB® Certified Professional for Software Architecture –

Foundation Level (CPSA-F®)

International Software Architecture Qualification Board e. V.

Document version: 2025.2
Based on curriculum - version 2025.1-rev2, 20250427



Prüfungsregeln

Die vorliegende Prüfung ist eine Beispielprüfung, welche in Form und Umfang an die Zertifizierungsprüfung des Certified Professional for Software Architecture – Foundation Level (CPSA-F®) angelehnt ist. Sie dient der Veranschaulichung der echten iSAQB®-CPSA®-Prüfung sowie der entsprechenden Prüfungsvorbereitung.

Die Beispielprüfung besteht aus 38 Multiple-Choice-Fragen, welche je nach Schwierigkeitsgrad mit 1 bis 2 Punkten bewertet werden können. Es müssen zum Bestehen der Prüfung mindestens 60 Prozent erreicht werden. In dieser Probeprüfung können 52 Punkte erreicht werden, zum Bestehen wären 31,20 Punkte erforderlich.

Grundsätzlich gelten folgende Hinweise: Richtige Antworten ergeben Pluspunkte, falsche Antworten führen zu Punktabzug, jedoch nur in Bezug auf die jeweilige Frage. Führt die falsche Beantwortung einer Frage zu einem negativen Punktergebnis, so wird diese Frage mit insgesamt 0 Punkten bewertet. Falls Sie mehr Kreuze setzen als gefordert, erhalten Sie grundsätzlich null Punkte.

Es gibt in dieser Beispielprüfung (wie auch in der Originalprüfung) nur folgende drei Typen von Prüfungsfragen:

A-Fragen (Einfachauswahlfragen, „Auswahl“) Wählen Sie zu einer Frage aus der Liste von Antwortmöglichkeiten die einzig korrekte Antwort aus. Es gibt nur eine korrekte Antwort. Sie erhalten die angegebene Punktzahl für das Ankreuzen der korrekten Antwort.

P-Fragen (Mehrfachauswahlfragen, „Pick“) Wählen Sie zu einer Frage aus der Liste von Antwortmöglichkeiten die im Text vorgegebene Anzahl von zutreffenden oder korrekten Antworten aus. Kreuzen Sie maximal so viele Antworten an, wie im Einleitungstext verlangt werden. Sie erhalten für jede korrekte Antwort anteilig $1/n$ der Gesamtpunkte. Für jedes nicht-korrekte Kreuz wird $1/n$ der Punkte abgezogen.

K-Fragen (Klärungsfragen, „Kreuz“) Wählen Sie zu einer Frage die korrekte der beiden Optionen zu jeder Antwortmöglichkeit aus („richtig“ oder „falsch“ bzw. „zutreffend“ oder „nicht zutreffend“). Sie erhalten für jedes korrekt gesetzte Kreuz anteilig $1/n$ der Punkte. Nicht korrekt gesetzte Kreuze führen zum Abzug von $1/n$ der Punkte. Wird in einer Zeile KEINE Antwort ausgewählt, so gibt es weder Punkte noch Abzüge.

Zur genaueren Erläuterung der Fragetypen und Punkteverteilung stehen weitere Informationen unter der Prüfungsregeln des CPSA-F zur Verfügung.

Die Bearbeitungsdauer beträgt 75 Minuten für Muttersprachler:innen und 90 Minuten für Nicht-Muttersprachler:innen. Um eine möglichst authentische Prüfungsvorbereitung zu gewährleisten, sollte die Bearbeitungszeit eingehalten sowie auf jegliche Hilfsmittel (wie Seminarunterlagen, Bücher, Internet etc.) verzichtet werden.

Im Anschluss erfolgt die Auswertung der Prüfung mit Hilfe der Musterlösung. Sofern der iSAQB® e.V. als Quelle und Copyright-Inhaber angegeben wird, darf die vorliegende Beispielprüfung im Rahmen von Schulungen eingesetzt, zur Prüfungsvorbereitung genutzt oder unentgeltlich weitergegeben werden.

Es ist ausdrücklich untersagt, diese Prüfungsfragen in einer echten Prüfung zu verwenden.

1 – Id: Q-20-04-01

Wie viele Definitionen des Begriffes „Softwarearchitektur“ gibt es? [1 Punkt]

A-Frage: Bitte kreuzen Sie die richtige Antwort an.

- a) Genau eine für alle Arten von Systemen.
- b) Eine für jede Art von Softwaresystem (z. B. „eingebettet“, „Echtzeit“, „Entscheidungsunterstützung“, „Web“, „Batch“, ...).
- c) Ein Dutzend oder mehr unterschiedliche Definitionen.

2 – Id: Q-20-04-02

Welche der folgenden Aspekte werden durch den Begriff „Softwarearchitektur“ abgedeckt? [1 Punkt]

P-Frage: Wählen Sie aus den folgenden fünf Antworten die **drei** Antworten aus, die am besten passen.

- a) Komponenten
- b) Querschnittskonzepte
- c) (interne und externe) Schnittstellen
- d) Programmierkonventionen („coding conventions“)
- e) Hardware-Sizing

3 – Id: Q-17-13-01

Welche VIER der folgenden Aussagen zu (Querschnitts-) Konzepten [2 Punkte] sind am zutreffendsten?

P-Frage: Wählen Sie aus den folgenden sieben Antworten die **vier** Antworten aus, die am besten passen.

- a) Durch die einheitliche Verwendung von Konzepten wird die Kopplung zwischen Bausteinen verringert.
- b) Durch die Definition von geeigneten Konzepten wird die Mustertreue der Architektur sichergestellt.
- c) Eine einheitliche Ausnahmebehandlung (Exception Handling) kann man gut erreichen, indem die Architekt:innen mit den Entwickler:innen vor der Implementierung ein geeignetes Konzept vereinbaren.

- d) Für jedes Qualitätsziel sollte es ein explizit dokumentiertes Konzept geben.
- e) Konzepte sind ein Mittel zur Erhöhung der Konsistenz.
- f) Ein Konzept kann Einschränkungen für die Umsetzung vieler Bausteine definieren.
- g) Ein Konzept kann durch einen einzigen Baustein umgesetzt werden.

4 – Id: Q-17-13-02

Bei Ihrem Projekt arbeiten drei Architekt:innen und sieben Entwickler:innen an der Dokumentation der Softwarearchitektur. Welche Methoden eignen sich zur Gewährleistung einer konsistenten und zweckmäßigen Dokumentation und welche nicht? [1 Punkt]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Geeignet Nicht geeignet

- | | | |
|-------------------------------------|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Die/der leitende Architekt:in koordiniert die Erstellung der Dokumentation. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) Für die Dokumentation werden identische Vorlagen verwendet. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) Alle Teile der Dokumentation werden automatisch aus dem Quellcode extrahiert. |

5 – Id: Q-17-13-03

Welche der folgenden Techniken sind am besten zur Darstellung von Abläufen oder Laufzeitverhalten geeignet? [2 Punkte]

*P-Frage: Wählen Sie aus den folgenden acht Antworten die **vier** Antworten aus, die am besten passen.*

- a) Flussdiagramme
- b) Aktivitätsdiagramme
- c) Darstellung von Screenflows (Abfolge von Benutzerinteraktionen)
- d) Sequenzdiagramm
- e) Lineares Venn-Diagramm
- f) Nummerierte Liste aufeinanderfolgender Schritte
- g) Tabellarische Schnittstellenbeschreibung

h) Klassendiagramme

6 – Id: Q-17-13-04

Welche der folgenden Grundsätze gelten für das Testen? [1 Punkt]

P-Frage: Wählen Sie aus den folgenden fünf Antworten die **drei** Antworten aus, die am besten passen.

- a) Im Allgemeinen ist es nicht möglich, sämtliche Fehler eines Systems zu finden.
- b) Bei Komponenten mit vielen bekannten vorherigen Fehlern sind die Chancen für zusätzliche Fehler hoch.
- c) Durch ausreichendes Testen kann aufgezeigt werden, dass ein Programm fehlerfrei ist.
- d) Durch Testen kann nur die Existenz von Fehlern aufgezeigt werden, nicht jedoch ihre Abwesenheit.
- e) Die funktionale Programmierung erlaubt keine automatisierten Tests.

7 – Id: Q-17-13-05

Welche der folgenden Aussagen zum Entwurfsprinzip „Information Hiding“ sind richtig und welche falsch? [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Richtig Falsch

- | | | | |
|-------------------------------------|-------------------------------------|----|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) | Durch die Befolgung des Prinzips „Information Hiding“ wird die Flexibilität für Änderungen erhöht. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) | Beim Information Hiding werden absichtlich Informationen vor Aufrufern oder Konsumenten des Bausteins verborgen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) | Information Hiding erschwert das Bottom-Up Vorgehen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | d) | Information Hiding ist abgeleitet vom Ansatz der inkrementellen Verfeinerung entlang des Kontrollflusses. |

8 – Id: Q-20-04-03

Was sind die wichtigsten Ziele von Softwarearchitektur? [1 Punkt]

P-Frage: Wählen Sie aus den folgenden vier Antworten die **zwei** Antworten aus, die am besten passen.

- a) Verbesserung der Genauigkeit von Mustern in Struktur und Implementierung.
- b) Erreichung der Qualitätsanforderungen auf nachvollziehbare Weise.
- c) Ermöglichung von kosteneffizienten Integrations- und Abnahmetests des Systems.
- d) Ermöglichung eines grundlegenden Verständnisses der Strukturen und Konzepte für das Entwicklungsteam und andere Beteiligte.

9 – Id: Q-20-04-12

Stellen Sie sich vor, Sie sind Softwarearchitekt:in für eine große und verteilte Geschäftsanwendung im Banken- oder Versicherungsbereich. Welche der folgenden Aussagen sind für diese Situation richtig und welche falsch? [1 Punkt]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

- | | |
|-------------------------------------|--------------------------|
| Richtig | Falsch |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
- a) Die Architektur sollte so aufgebaut sein, dass häufig vorkommende Änderungen an den entsprechenden Geschäftsprozessen ohne umfangreiche Umstrukturierungen der Softwarearchitektur möglich sind.
 - b) Erforderliche Produktqualitäten sollten Ihre architektonischen Entscheidungen leiten.
 - c) Die Softwarearchitektur kann völlig unabhängig von Hardware und Infrastruktur entworfen werden.

10 – Id: Q-20-04-03

Was sind Ihre wichtigsten Verantwortlichkeiten als Softwarearchitekt:in in Bezug auf Anforderungen? [1 Punkt]

P-Frage: Wählen Sie aus den folgenden fünf Antworten die **drei** Antworten aus, die am besten passen.

- a) Unterstützung der Fachexpert:innen dabei, Qualitätsanforderungen konkret und explizit zu formulieren.
- b) Hilfe bei der Identifizierung von neuen Geschäftsmöglichkeiten anhand Ihres technischen Know-hows.
- c) Ablehnung von Anforderungen, die technische Risiken enthalten.
- d) Umformulierung aller Geschäftsanforderungen, so dass sie für Ihr Entwicklungsteam verständlich sind.

- e) Überprüfung der Anforderungen auf technische Machbarkeit.

11 – Id: Q-20-04-07

Sie sind als Architekt:in dafür zuständig, ein Altsystem entsprechend den laufenden Betriebsanforderungen am Laufen zu halten.
Was sind Ihre wichtigsten Aufgaben?

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **drei** Antworten aus, die am besten passen.*

- a) Verhandlung des Wartungsbudgets für Ihr Team
- b) Sicherstellung einer aktuellen Dokumentation des ausgelieferten Systems
- c) Analyse der Auswirkungen von neuen Anforderungen auf das aktuelle System
- d) Ermunterung der Teammitglieder, neue Programmiersprachen zu lernen
- e) Ihren Vorgesetzten technologische Updates zusätzlich zu den Geschäftsanforderungen vorschlagen

12 – Id: Q-21-05-01

Welche der folgenden Aussagen zu Architekturentscheidungen [2 Punkte] sind wahr, welche falsch?

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Wahr Falsch

- | | | | |
|-------------------------------------|-------------------------------------|----|--|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | a) | Architekturentscheidungen müssen nie aufgeschrieben werden, da diese bereits dem Entwicklungsteam bekannt sind. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) | Architecture Decision Records helfen dabei, Entscheidungen in ihrem Kontext nachzuvollziehen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) | Nachdem man sich für ein zentrales Framework (z.B. Persistenzframework) entschieden hat, darf diese Entscheidung nicht mehr geändert werden. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) | Qualitätsanforderungen helfen maßgeblich bei Architekturentscheidungen. |

13 – Id: Q-20-04-09

Geben Sie für jede der folgenden Aussagen an, ob sie richtig oder [2 Punkte] falsch ist.

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Richtig Falsch

- | | | |
|-------------------------------------|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Jede Iteration eines agilen Entwicklungsvorgehens kann Auswirkungen auf architektonische Grundsatzentscheidungen haben. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | b) Der Gesamtaufwand für Architekturarbeit ist bei iterativen Projekten wesentlich höher als bei Wasserfall-Projekten. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) Agile Projekte benötigen keine Architekturdokumente, da das Entwicklungsteam in täglichen Standup-Meetings Entscheidungen kommuniziert. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | d) Wenn Ihr System aus einer Reihe von Microservices besteht, ist kein zentrales Architekturdokument erforderlich, da jeder Service seine Technologien auswählen kann. |

14 – Id: Q-20-04-10

Geben Sie an, welche der folgenden Aussagen zu Projektzielen und Architekturzielen richtig und welche falsch sind. [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Richtig Falsch

- | | | |
|-------------------------------------|-------------------------------------|---|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Projektziele können funktionale Anforderungen sowie Qualitätsanforderungen umfassen. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) Architekturziele leiten sich von den Qualitätsanforderungen für das System oder Produkt ab. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) Stakeholder aus dem Business sollten sich auf Geschäftsziele konzentrieren und sich nicht mit Architekturzielen auseinandersetzen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | d) Zur Vermeidung von Konflikten sollten Geschäfts- und Architekturziele einander nicht überlappen. |

15 – Id: Q-20-04-11

Was bedeutet die Regel „explizit, nicht implizit“ für die Architekturarbeit? Wählen Sie die passendsten Antworten aus. [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Architekt:innen sollten rekursive Strukturen vermeiden und durch explizite Schleifen ersetzen.
- b) Architekt:innen sollten die Annahmen, die zu Entscheidungen führen, explizit machen.

- c) Architekt:innen sollten explizit auf Erläuterungen (d.h. Kommentare) in natürlicher Sprache für jeden Baustein bestehen.
- d) Architekt:innen sollten explizit auf schriftlichen oder zumindest mündlichen Begründungen für Aufwandschätzungen der Entwicklung von ihrem Team bestehen.
- e) Architekt:innen sollten die Voraussetzungen für ihre Entscheidungen explizit darlegen.

16 – Id: Q-20-04-32

Es gibt zahlreiche Ansätze, die zu einer Softwarearchitektur führen. [1 Punkt]
Welche der folgenden Ansätze kommen in der Praxis am häufigsten vor?

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **drei** Antworten aus, die am besten passen.*

- a) User-Interface Driven Design
- b) Domain-driven Design
- c) Sichtenbasierte Architekturentwicklung
- d) Bottom-up Design
- e) Mehrheitsentscheid

17 – Id: Q-20-04-38

Diverse Architekturentwicklungsmethoden schlagen einen sichtenbasierten Ansatz vor. Welche der folgenden Sichten werden am häufigsten verwendet? [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden sechs Antworten die **drei** Antworten aus, die am besten passen.*

- a) Physische Datenbanksicht
- b) Kontextsicht
- c) Bausteinsicht oder Komponentensicht
- d) Testbasierte Sicht
- e) Konfigurationssicht
- f) Laufzeitsicht

18 – Id: Q-20-04-22

Sie dokumentieren einen Baustein einer Softwarearchitektur. Welche Informationen sollten in seiner Black-Box-Beschreibung enthalten sein? [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden vier Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Öffentliche Schnittstellen.
- b) Verantwortlichkeiten des Bausteins.
- c) Interne Struktur des Bausteins.
- d) Spezifikation von Implementierungsdetails.

19 – Id: Q-20-04-17

Welche Voraussetzungen müssen vor der Entwicklung einer Softwarearchitektur erfüllt sein? Wählen Sie die passendsten Antworten aus. [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Die Anforderungsspezifikation für das System ist vollständig, detailliert und konsistent.
- b) Die wichtigsten Qualitätsanforderungen an das System sind bekannt.
- c) Die organisatorischen Randbedingungen sind bekannt.
- d) Die Programmiersprache wurde ausgewählt.
- e) Die Hardware für das Entwicklungsteam ist verfügbar.

20 – Id: Q-20-04-18

Welche Faktoren können den Entwurf einer Softwarearchitektur beeinflussen? Wählen Sie die passendsten Antworten aus. [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden vier Antworten die **drei** Antworten aus, die am besten passen.*

- a) Politische.
- b) Organisatorische.
- c) Technische.

- d) Virtuelle.

21 – Id: Q-20-04-28

Welche der folgenden Eigenschaften lässt sich am ehesten durch eine Schichtenarchitektur verbessern? [1 Punkt]

A-Frage: Bitte kreuzen Sie die richtige Antwort an.

- a) Laufzeiteffizienz (Performance).
- b) Flexibilität bei der Modifizierung oder Änderung des Systems.
- c) Flexibilität bei der Laufzeit (Konfigurierbarkeit).
- d) Nichtabstrebbarkeit.

22 – Id: Q-20-04-33

Für welche Entwurfsprobleme bietet das Pipes & Filter Pattern Lösungsansätze? [1 Punkt]

P-Frage: Wählen Sie aus den folgenden vier Antworten die **zwei** Antworten aus, die am besten passen.

- a) Verwaltung von globalem Anwendungszustand
- b) Strukturierung von IT-Systemen die Datenströme verarbeiten
- c) Entkopplung von Verarbeitungsschritten
- d) Entkopplung von zeitlichen Abhängigkeiten

23 – Id: Q-20-04-20

Welche Ziele versuchen Sie mit dem Dependency-Inversion-Prinzip zu erreichen? [1 Punkt]

A-Frage: Bitte kreuzen Sie die richtige Antwort an.

- a) Große Bausteine sollen nicht von kleinen Bausteinen abhängen.
- b) Komponenten sollen in der Lage sein, abhängige Komponenten leichter zu erstellen.
- c) Bausteine sollen nur über Abstraktionen voneinander abhängen.

24 – Id: Q-20-04-21

Was sind die Eigenschaften von enger (hoher) bzw. loser (niedriger) Kopplung? [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

- | | | |
|-------------------------------------|-------------------------------------|---|
| enge Kopplung | lose Kopplung | |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Bausteine können abhängige Bausteine direkt, d.h. ohne Umwege über Schnittstellen oder Abstraktionen, aufrufen. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) Bausteine verwenden gemeinsame komplexe Datenstrukturen. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | c) Bausteine verwenden eine gemeinsame Tabelle (für Schreib- und Leseoperationen) innerhalb einer relationalen Datenbank. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | d) Beim Baustein-Design haben Sie das Dependency-Inversion-Prinzip konsequent umgesetzt. |

25 – Id: Q-20-04-14

Welche Aussagen zum „Don't repeat yourself“-Prinzip (DRY) treffen am ehesten zu? Anders gesagt: Was könnte passieren, wenn Teile des Quellcodes oder der Konfiguration im System mehrfach vorhanden sind? [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **zwei** Antworten aus, die am besten passen.*

- a) DRY verringert die Sicherheit.
- b) Die strenge Einhaltung von DRY könnte zu höherer Kopplung führen.
- c) Die Komponenten des Systems mit redundantem Code können unabhängig voneinander verbessert werden.
- d) Die Einhaltung von DRY führt zu einer Verringerung der Angriffsvektoren in der IT-Sicherheit.
- e) Die Anwendung der Schichtenmuster ermöglicht eine konsistente Anwendung des DRY-Prinzips.

26 – Id: Q-20-04-15

Sie können Aspekte Ihrer Softwarearchitektur mündlich und/oder schriftlich kommunizieren. In welchem Zusammenhang stehen diese Möglichkeiten zueinander? Geben Sie für jede der folgenden Aussagen an, ob sie richtig oder falsch ist. [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Richtig Falsch

- | | | |
|-------------------------------------|-------------------------------------|---|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Mündliche Kommunikation sollte schriftliche Dokumentation ergänzen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | b) Feedback zu Architekturentscheidungen sollte zur Sicherstellung der Nachverfolgbarkeit grundsätzlich schriftlich erfolgen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) Schriftliche Dokumentation sollte immer mündlicher Kommunikation vorausgehen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | d) Architekt:innen sollten sich eine Variante (mündlich oder schriftlich) aussuchen und während der gesamten Entwicklung dabei bleiben. |

27 – Id: Q-20-04-37

Welche der folgenden Aussagen zu Notationen für Architektursichten sind richtig und welche falsch? [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Richtig Falsch

- | | | |
|-------------------------------------|-------------------------------------|--|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | a) Business Process Model & Notation (BPMN) sollte nur von Business-Analysten und nicht zur Architektdokumentation verwendet werden. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | b) UML-Verteilungsdiagramme sind die einzige Möglichkeit zur Dokumentation des Mappings der Softwarekomponenten in Bezug auf die Infrastruktur. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | c) UML-Paketdiagramme können zum Festhalten der Bausteinsicht der Softwarearchitektur verwendet werden. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) Solange die Notation (z. B. mithilfe einer Legende) erläutert wird, kann eine beliebige Notation ausreichen, um Bausteinstrukturen und Zusammenarbeit zu beschreiben. |

28 – Id: Q-20-04-13

Welche Architektursichten haben die beste praktische Anwendung [1 Punkt] für die Entwicklung von Softwarearchitekturen?

*P-Frage: Wählen Sie aus den folgenden vier Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Pattern-Sicht.
- b) Beobachtersicht (Observer-View).

- c) Bausteinsicht (Komponentensicht).
- d) Verteilungssicht.

29 – Id: Q-20-04-23

In der Kontextsicht können Sie einen geschäftlichen Kontext („business context“) und einen technischen Kontext verwenden. Wählen Sie die am besten passenden Antworten für den technischen Kontext aus. [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Der technische Kontext enthält die physischen Übertragungskanäle zwischen Ihrem System und der Umgebung.
- b) Der technische Kontext enthält die gesamte Infrastruktur, über die die Komponenten Ihres Systems verteilt werden.
- c) Der technische Kontext sollte die Hardware-Preisliste oder die Preisgestaltung von Cloud-Diensten, die als Infrastruktur für Ihre Architektur verwendet werden, enthalten.
- d) Der technische Kontext enthält Informationen zur gewählten Programmiersprache sowie allen zur Implementierung Ihrer Softwarearchitektur verwendeten Frameworks.
- e) Der technische Kontext enthält gegebenenfalls andere Elemente als der Geschäftskontext („business context“).

30 – Id: Q-20-04-24

Die Dokumentation der Softwarearchitektur sollte Beschreibungen der Querschnittskonzepte enthalten. Wählen Sie die besten Gründe, warum die Dokumentation von Querschnittskonzepten sinnvoll ist. [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden vier Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Querschnittskonzepte sollten sich auf die Fachdomäne konzentrieren und frei von technischen Informationen sein.
- b) Aspekte oder Konzepte, die in mehreren Teilen Ihrer Softwarearchitektur verwendet werden, sollten auf nicht redundante Weise beschrieben werden.

- c) Querschnittskonzepte können in weiteren Produkten innerhalb der gleichen Organisation erneut verwendet werden.
- d) Querschnittskonzepte sollten von Spezialist:innen implementiert werden. Daher ist eine separate Dokumentation hilfreich.

31 – Id: Q-20-04-25

Was sind Richtlinien für ein gutes Schnittstellen-Design? Kreuzen Sie an, welche der folgenden Aussagen richtig und welche falsch sind. [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

richtig falsch

- | | | |
|-------------------------------------|-------------------------------------|---|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Die Verwendung der Schnittstellen sollte einfach zu erlernen sein. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) Es soll möglich sein, angemessen verständlichen Client-Code für diese Schnittstelle zu schreiben. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) Eine Schnittstelle soll Zugriff auf eine umfassende Menge von Implementierungsdetails ermöglichen. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) Schnittstellenspezifikationen sollten funktionale und nichtfunktionale Aspekte enthalten. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | e) Lokale und entfernte („remote“) Aufrufe einer Schnittstelle sollen sich in allen Belangen identisch verhalten. |

32 – Id: Q-20-04-26

Eine Definition lautet: „Softwarearchitektur ist die Summe aller während der Entwicklung gefällten Entscheidungen“. Kreuzen Sie an, welche der folgenden Aussagen zu Architektur- /Designentscheidungen richtig und welche falsch sind. [2 Punkte]

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Richtig Falsch

- | | | |
|-------------------------------------|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) Architektonische Entscheidungen können sich in der Struktur der Bausteine oder Komponenten niederschlagen. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | b) Softwarearchitekt:innen sollten alle Entwurfsentscheidungen schriftlich begründen. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | c) Architektonische Entscheidungen können untereinander Wechselwirkungen haben. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) Kompromisse zwischen zueinander in Konflikt stehenden Qualitätsanforderungen sollten explizite Entscheidungen sein. |

33 – Id: Q-20-04-31

Welche der folgenden Aussagen sind typische Gründe zur [2 Punkte] Einführung einer Architekturdokumentation und welche nicht?

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

typisch nicht typisch

- | | |
|--|---|
| <input checked="" type="checkbox"/> <input type="checkbox"/> | a) Unterstützung des Onboardings neuer Entwickler:innen. |
| <input type="checkbox"/> <input checked="" type="checkbox"/> | b) Unterstützung der Testautomatisierung des Systems. |
| <input checked="" type="checkbox"/> <input type="checkbox"/> | c) Unterstützung der Arbeit von verteilten Teams. |
| <input checked="" type="checkbox"/> <input type="checkbox"/> | d) Unterstützung von späteren Verbesserungen des Systems. |
| <input checked="" type="checkbox"/> <input type="checkbox"/> | e) Einhaltung regulatorischer Vorgaben. |
| <input type="checkbox"/> <input checked="" type="checkbox"/> | f) Sicherstellung der gleichmäßigen Auslastung des Teams. |

34 – Id: Q-20-04-30

Welche der folgenden Eigenschaftspaare stehen üblicherweise [2 Punkte] miteinander in Konflikt und welche nicht?

K-Frage: Bitte ordnen Sie jede Antwort einer Kategorie zu.

Konflikt Kein Konflikt

- | | |
|--|--|
| <input type="checkbox"/> <input checked="" type="checkbox"/> | a) Verständlichkeit – Lesbarkeit |
| <input checked="" type="checkbox"/> <input type="checkbox"/> | b) Benutzerfreundlichkeit – Sicherheit |
| <input checked="" type="checkbox"/> <input type="checkbox"/> | c) Laufzeitkonfigurierbarkeit – Robustheit |
| <input type="checkbox"/> <input checked="" type="checkbox"/> | d) Sicherheit – Funktionale Korrektheit |

35 – Id: Q-20-04-27

ISO 25010 enthält allgemeine Qualitätseigenschaften für Softwaresysteme. Wie können Qualitätsanforderungen zu diesen Eigenschaften konkretisiert werden? [1 Punkt]

A-Frage: Bitte kreuzen Sie die richtige Antwort an.

- a) Durch Entwickeln von UI-Prototypen.
- b) Durch Definieren expliziter Schnittstellen.
- c) Durch Erstellen, Verfassen oder Diskutieren von Qualitätsszenarien.
- d) Durch Erstellen automatisierter Tests.

36 – Id: Q-20-04-28

Welche der folgenden Punkte sind am besten geeignet, die qualitative Bewertung einer Softwarearchitektur bzgl. der Erfüllung von Qualitätsanforderungen zu unterstützen? [2 Punkte]

*P-Frage: Wählen Sie aus den folgenden sechs Antworten die **vier** Antworten aus, die am besten passen.*

- a) Quantitative Abhängigkeitsanalyse.
- b) Architekturmodelle.
- c) Qualitätsszenarien.
- d) Teamgröße.
- e) Logdateien.
- f) Organigramm.

37 – Id: Q-20-04-29

Sie versuchen, Ihre Architektur qualitativ zu analysieren. Was sind die zutreffendsten Anzeichen für architektonische Problembereiche? [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **zwei** Antworten aus, die am besten passen.*

- a) Hohe Kopplung der Komponenten.
- b) Namen öffentlicher Methoden geben nicht deren Zweck wieder.
- c) Fehlende Kommentare.
- d) Häufung von Fehlern in bestimmten Bausteinen des Systems.
- e) Anzahl der Testfälle pro Komponente.

38 – Id: Q-20-04-36

Sie versuchen, ihre Architektur quantitativ zu untersuchen. Welche der folgenden Größen können Sie für Ihre Softwarearchitektur zuverlässig messen? Wählen Sie die am besten passenden Antworten aus. [1 Punkt]

*P-Frage: Wählen Sie aus den folgenden fünf Antworten die **drei** Antworten aus, die am besten passen.*

- a) Größe der Bausteine (z. B. Lines-of-Code).
- b) Änderungsrate des Quellcodes der Komponenten.
- c) Kohäsion der Architekturkomponenten.
- d) Sicherheitsstufe einer Komponente.
- e) Anzahl der Personen, die zu einer bestimmten Komponente beigetragen haben.