

Beispielprüfung

iSAQB[®] Certified Professional for Software Architecture – Foundation Level (CPSA-F[®])

Version: 2021.1-DE-rev6
basierend auf Lehrplan - Version 2019.2-DE; 11. März 2021



Erläuterungen zur Beispielprüfung Certified Professional for Software Architecture - Foundation Level (CPSA-F®)

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 39 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 Probepfung können 50,0 Punkte erreicht werden, zum Bestehen wären 30,0 Punkte erforderlich.

Grundsätzlich gilt: 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.

Die Multiple-Choice-Fragen der Beispielprüfung gliedern sich in drei Arten von Fragen:

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. Die erreichbare Punktzahl beträgt je nach Schwierigkeitsgrad 1-2 Punkte.

P-Fragen (Mehrfachauswahlfragen, „Pick“):

Wählen Sie zu einer Frage aus der Liste von Antwortmöglichkeiten die im Text vorgegebene Anzahl von 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. Die Punktzahl beträgt je nach Schwierigkeitsgrad 1-2 Punkte.

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. Die Punktzahl beträgt je nach Schwierigkeitsgrad 1-2 Punkte.

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 und 90 Minuten für Nicht-Muttersprachler. 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 jedoch ausdrücklich untersagt, diese Prüfungsfragen in einer echten Prüfung zu verwenden.

Frage 1

A-Frage: Wählen Sie eine Option aus.

1 Punkt

ID: Q-20-04-01

Wie viele Definitionen von „Softwarearchitektur“ gibt es?

- (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.

Frage 2

P-Frage: Wählen Sie die drei besten Aspekte aus.

1 Punkt

ID: Q-20-04-02

Welche **DREI** der folgenden Aspekte werden durch den Begriff „Softwarearchitektur“ abgedeckt?

- (a) Komponenten.
- (b) Querschnittskonzepte.
- (c) (interne und externe) Schnittstellen.
- (d) Datenbankschemata.
- (e) Hardware-Sizing.

Frage 3*P-Frage: Wählen Sie die vier besten Antworten aus.**2 Punkte*

ID: Q-17-13-01

Welche **VIERT** der folgenden Aussagen zu (Querschnitts-) Konzepten sind am zutreffendsten?

- (a) Durch die einheitliche Verwendung von Konzepten wird die Kopplung zwischen Bausteinen verringert.
- (b) Durch die Definition von geeigneten Konzepten wird Mustertreue der Architektur sichergestellt.
- (c) Eine einheitliche Ausnahmebehandlung wird am einfachsten erreicht, wenn die Architekten mit den Entwicklern 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.

Frage 4*K-Frage: Wählen Sie für jede Zeile „Geeignet“ oder „Nicht geeignet“ aus.**2 Punkte*

ID: Q-17-13-02

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

Geeignet Nicht geeignet

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

Frage 5*P-Frage: Wählen Sie die vier besten Optionen aus.**1 Punkt*

ID: Q-17-13-03

Welche **VIERT** der folgenden Techniken sind am besten zur Darstellung von Abläufen oder Laufzeitverhalten geeignet?

- (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.

Frage 6*P-Frage: Wählen Sie die drei besten Optionen aus.**1 Punkt*

ID: Q-17-13-04

Welche **DREI** der folgenden Grundsätze gelten für das Testen?

- (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.

Frage 7*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**1 Punkt*

ID: Q-17-13-05

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

Richtig Falsch

- (a) Durch die Befolgung des Prinzips „Information Hiding“ wird die Flexibilität für Änderungen erhöht.
- (b) Beim Information Hiding werden absichtlich Informationen vor Aufrufern oder Konsumenten des Bausteins verborgen.
- (c) Information Hiding erschwert die Unterscheidung zwischen Schnittstelle und Implementierung.
- (d) Information Hiding ist abgeleitet vom Ansatz der inkrementellen Verfeinerung entlang des Kontrollflusses.

Frage 8*P-Frage: Wählen Sie die zwei besten Optionen aus.**1 Punkt*

ID: Q-20-04-03

Was sind die **ZWEI** wichtigsten Ziele von Softwarearchitektur?

- (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.

Frage 9*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**1 Punkt*

ID: Q-20-04-12

Versetzen Sie sich in die Lage eines Softwarearchitekten für eine große und verteilte Geschäftsanwendung im Banken- oder Versicherungsbereich. Welche der folgenden Aussagen sind richtig und welche falsch?

Richtig Falsch

- (a) Ihre 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 der Hardware und Infrastruktur entworfen werden.

Frage 10*P-Frage: Wählen Sie die drei wichtigsten Aufgaben aus.**2 Punkte*

ID: Q-20-04-06

Was sind Ihre **DREI** wichtigsten Verantwortlichkeiten als Softwarearchitekt:in in Bezug auf Anforderungen?

- (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.

Frage 11*P-Frage: Wählen Sie die drei wichtigsten Aufgaben aus.**1 Punkt***ID: Q-20-04-07**

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

- (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) Ihrem Vorgesetzten technologische Updates zusätzlich zu den Geschäftsanforderungen vorschlagen.

Frage 12**ID: Q-20-04-08**

Frage entfernt, da nicht kompatibel mit CPSA-F Lehrplan.

Frage 13*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**1 Punkt***ID: Q-20-04-09**

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

Richtig Falsch

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

Frage 14 *K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.* 2 Punkte

ID: Q-20-04-10

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

Richtig Falsch

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

Frage 15 *P-Frage: Wählen Sie die zwei passendsten Antworten aus.* 1 Punkt

ID: Q-20-04-11

Was bedeutet die Regel „explizit, nicht implizit“ für die Architekturarbeit? Wählen Sie die **ZWEI** passendsten Antworten aus.

- (a) Architekten sollten rekursive Strukturen vermeiden und durch explizite Schleifen ersetzen.
- (b) Architekten sollten die Annahmen, die zu Entscheidungen führen, explizit machen.
- (c) Architekten sollten explizit auf Erläuterungen (d.h. Kommentare) in natürlicher Sprache für jeden Baustein bestehen.
- (d) Architekten sollten explizit auf schriftlichen oder zumindest mündlichen Begründungen für Entwicklungsaufwandschätzungen von ihrem Team bestehen.
- (e) Architekten sollten die Voraussetzungen für ihre Entscheidungen explizit darlegen.

Frage 16*P-Frage: Wählen Sie die drei passendsten Antworten aus.**1 Punkt*

ID: Q-20-04-19

Kreuzen Sie die **DREI** zutreffendsten Beispiele für typische Kategorien von Softwaresystemen an.

- (a) Batch-System.
- (b) Interaktives Onlinesystem.
- (c) Linnés System.
- (d) Eingebettetes Echtzeitsystem.
- (e) Integrationstestsystem.

Frage 17*P-Frage: Wählen Sie die drei in der Praxis am häufigsten vorkommenden Ansätze aus.**1 Punkt*

ID: Q-20-04-32

Es gibt zahlreiche Ansätze, die zu einer Softwarearchitektur führen. Welche **DREI** der folgenden Ansätze kommen in der Praxis am häufigsten vor?

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

Frage 18*P-Frage: Wählen Sie die drei am häufigsten verwendeten Sichten aus.**1 Punkt*

ID: Q-20-04-38

Viele Architekturentwicklungsmethoden sehen einen sichtenbasierten Ansatz vor. Welche **DREI** der folgenden Sichten werden am häufigsten verwendet?

- (a) Physische Datenbanksicht.
- (b) Kontextsicht.
- (c) Baustein-/Komponentensicht.
- (d) Testbasierte Sicht.
- (e) Konfigurationssicht.
- (f) Laufzeitsicht.

Frage 19*P-Frage: Wählen Sie die zwei am besten passenden Antworten aus. 1 Punkt*

ID: Q-20-04-22

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

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

Frage 20*P-Frage: Wählen Sie die zwei passendsten Antworten aus.**1 Punkt*

ID: Q-20-04-17

Welche Voraussetzungen müssen vor der Entwicklung einer Softwarearchitektur erfüllt sein? Wählen Sie die **ZWEI** passendsten Antworten aus.

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

Frage 21*P-Frage: Wählen Sie die drei passendsten Antworten aus.**1 Punkt***ID: Q-20-04-18**

Welche Faktoren können den Entwurf einer Softwarearchitektur beeinflussen? Wählen Sie die **DREI** passendsten Antworten aus.

- (a) Politische.
- (b) Organisatorische.
- (c) Technische.
- (d) Virtuelle.

Frage 22*A-Frage: Wählen Sie eine Antwort aus.**1 Punkt***ID: Q-20-04-28**

Welche der folgenden Eigenschaften lässt sich am ehesten durch eine Schichtenarchitektur verbessern?

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

Frage 23*A-Frage: Wählen Sie eine Antwort aus.**1 Punkt***ID: Q-20-04-33**

Für welche Art von System kann das Blackboard-Architekturmuster verwendet werden?

- (a) Harte Echtzeitsysteme.
- (b) Regelbasierte Systeme.
- (c) Linné-Systeme.
- (d) Sicherheitskritische Systeme.

Frage 24*A-Frage: Wählen Sie eine Antwort aus.**1 Punkt*

ID: Q-20-04-20

Welche Ziele versuchen Sie mit dem Dependency-Inversion-Prinzip zu erreichen?

- (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.

Frage 25*K-Frage: Wählen Sie für jede Zeile „Enge Kopplung“ oder „Lose Kopplung“ aus.**1 Punkt*

ID: Q-20-04-21

Was sind die Eigenschaften von enger (hoher) bzw. loser (niedriger) Kopplung?

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

Frage 26*P-Frage: Wählen Sie die zwei besten Antworten aus.**2 Punkte***ID: Q-20-04-14**

Welche **ZWEI** Aussagen zum „Don't repeat yourself“-Prinzip (DRY) sind richtig? (Anders gesagt: Was könnte passieren, wenn Teile des Quellcodes oder der Konfiguration im System mehrfach vorhanden sind?)

- (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 Schichtmuster ermöglicht eine konsistente Anwendung des DRY-Prinzips.

Frage 27*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**2 Punkte***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.

Richtig Falsch

- (a) Mündliche Kommunikation sollte schriftliche Dokumentation ergänzen.
- (b) Feedback zu Architekturentscheidungen sollte zur Sicherstellung der Nachverfolgbarkeit grundsätzlich schriftlich erfolgen.
- (c) Schriftliche Dokumentation sollte immer mündlicher Kommunikation vorausgehen.
- (d) Architekten sollten sich eine Variante (mündlich oder schriftlich) aussuchen und während der gesamten Entwicklung dabei bleiben.

Frage 28*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**2 Punkte*

ID: Q-20-04-37

Welche der folgenden Aussagen zu Notationen für Architektursichten sind richtig und welche falsch?

Richtig Falsch

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

Frage 29*P-Frage: Wählen Sie die zwei besten Antworten aus.**1 Punkt*

ID: Q-20-04-13

Welche Architektursichten haben eine praktische Anwendung für die Entwicklung von Softwarearchitekturen?

- (a) Pattern-Sicht.
- (b) Beobachtersicht (Observer-View).
- (c) Baustein- (oder Komponenten-) Sicht.
- (d) Verteilungssicht.

Frage 30*P-Frage: Wählen Sie die zwei passendsten Antworten aus.**1 Punkt***ID: Q-20-04-23**

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

- (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.

Frage 31*P-Frage: Wählen Sie die zwei besten Gründe aus.**1 Punkt***ID: Q-20-04-24**

Die Softwarearchitektur-Dokumentation sollte Beschreibungen der Querschnittskonzepte enthalten. Wählen Sie die **ZWEI** besten Gründe, warum die Dokumentation von Querschnittskonzepten sinnvoll ist.

- (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 mit der gleichen Organisation erneut verwendet werden.
- (d) Querschnittskonzepte sollten von einem Spezialisten implementiert werden. Daher ist eine separate Dokumentation hilfreich.

Frage 32*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**1 Punkt*

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.

Richtig Falsch

- (a) Die Verwendung der Schnittstellen sollte einfach zu erlernen sein.
- (b) Der Client-Code sollte angemessen leicht zu verstehen sein.
- (c) Eine Schnittstelle soll Zugriff auf eine umfassende Menge von Implementierungsdetails ermöglichen
- (d) Schnittstellenspezifikationen sollten funktionale und nichtfunktionale Aspekte enthalten.
- (e) Eine Schnittstelle soll die Implementierung so abstrahieren, dass eine Unterscheidung zwischen lokalen und entfernten Aufrufen nicht nötig ist.

Frage 33*K-Frage: Wählen Sie für jede Zeile „Richtig“ oder „Falsch“ aus.**1 Punkt*

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.

Richtig Falsch

- (a) Architektonische Entscheidungen können sich in der Struktur der Bausteine oder Komponenten niederschlagen.
- (b) Softwarearchitekten sollten alle Entwurfsentscheidungen schriftlich begründen.
- (c) Architektonische Entscheidungen können untereinander Wechselwirkungen haben.
- (d) Kompromisse zwischen zueinander in Konflikt stehenden Qualitätsanforderungen sollten explizite Entscheidungen sein.

Frage 34 *K-Frage: Wählen Sie für jede Zeile „typisch“ oder „untypisch“ aus.* 2 Punkte

ID: Q-20-04-31

Welche der folgenden Aussagen sind typische Gründe zur Führung einer (angemessenen) Architekturdokumentation und welche nicht?

typisch	untypisch	
<input type="checkbox"/>	<input type="checkbox"/>	(a) Unterstützung des Onboardings neuer Entwickler.
<input type="checkbox"/>	<input type="checkbox"/>	(b) Unterstützung der Testautomatisierung des Systems.
<input type="checkbox"/>	<input type="checkbox"/>	(c) Unterstützung der Arbeit von verteilten Teams.
<input type="checkbox"/>	<input type="checkbox"/>	(d) Unterstützung von späteren Verbesserungen des Systems.
<input type="checkbox"/>	<input type="checkbox"/>	(e) Einhaltung von gesetzlichen Vorgaben.
<input type="checkbox"/>	<input type="checkbox"/>	(f) Sicherstellung der gleichmäßigen Auslastung des Teams.

Frage 35 *K-Frage: Wählen Sie für jede Zeile „Konflikt“ oder „Kein Konflikt“ aus.* 1 Punkt

ID: Q-20-04-30

Welche der folgenden Eigenschaftspaare stehen üblicherweise miteinander in Konflikt und welche nicht?

Konflikt	Kein Konflikt	
<input type="checkbox"/>	<input type="checkbox"/>	(a) Verständlichkeit – Lesbarkeit.
<input type="checkbox"/>	<input type="checkbox"/>	(b) Benutzerfreundlichkeit – Sicherheit.
<input type="checkbox"/>	<input type="checkbox"/>	(c) Laufzeitkonfigurierbarkeit – Robustheit.
<input type="checkbox"/>	<input type="checkbox"/>	(d) Sicherheit – Compliance.

Frage 36*P-Frage: Wählen Sie die zwei besten Alternativen aus.**1 Punkt***ID: Q-20-04-27**

ISO 25010 enthält allgemeine Qualitätseigenschaften für Softwaresysteme. Wie können Qualitätsanforderungen zu diesen Eigenschaften konkretisiert werden? Wählen Sie die **ZWEI** besten Alternativen aus.

- (a) Durch die Entwicklung von UI-Prototypen.
- (b) Durch die Definition von expliziten Schnittstellen.
- (c) Durch die Diskussion oder Abfassung von Szenarien.
- (d) Durch die Erstellung automatisierter Tests.
- (e) Durch die Erstellung eines Qualitätsbaums.

Frage 37*P-Frage: Wählen Sie die vier besten Antworten.**1 Punkt***ID: Q-20-04-28**

Welcher der vier folgenden Punkte sind am besten geeignet, einer qualitative Analyse Ihrer Softwarearchitektur zu unterstützen?

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

Frage 38*P-Frage: Wählen Sie die zwei passendsten Anzeichen aus.**2 Punkte***ID: Q-20-04-29**

Sie versuchen, Ihre Architektur qualitativ zu analysieren. Was sind die **ZWEI** zutreffendsten Anzeichen für architektonische Problembereiche?

- (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.

Frage 39*P-Frage: Wählen Sie zwei Antworten aus.**1 Punkt***ID: Q-20-04-36**

Welche der folgenden Alternativen sind in Ihrer Softwarearchitektur schwieriger zu messen? Wählen Sie **ZWEI** Antworten aus.

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