

# Beispielprüfung

## **iSAQB® Certified Professional for Software Architecture – Foundation Level (CPSA-F)®**

Antwortbogen 2021.2-rev5-DE-20210608



## 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 gelten folgende Hinweise:

- Die maximale Punktzahl pro Frage beträgt je nach Schwierigkeitsgrad oder Umfang 1-2 Punkte.
- 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 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 ausdrücklich untersagt, diese Prüfungsfragen in einer echten Prüfung zu verwenden.

## Fragen

### Frage 1

ID: Q-20-04-01

---

A-Frage: Wählen Sie eine Option aus 1 Punkt

---

Wie viele Definitionen des Begriffes "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

ID: Q-20-04-02

---

P-Frage: Wählen Sie die **drei besten** Aspekte aus 1 Punkt

---

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

ID: Q-17-13-01

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

Welche VIER 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 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.

### Frage 4

ID: Q-17-13-02

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

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?

- | 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. |

Things like *reasoning* or *alternatives* won't be contained in code, but need to be included in documentation, therefore not **all** parts of documentation can be extracted from source code.

## Frage 5

ID: Q-17-13-03

---

P-Frage: Wählen Sie die **vier besten** Aspekte aus 1 Punkt

---

Welche VIER 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

ID: Q-17-13-04

---

P-Frage: Wählen Sie die **drei** besten Aspekte aus 1 Punkt

---

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

ID: Q-17-13-05

K-Frage: Wählen Sie für jede Zeile "Richtig" oder "Falsch" aus. 1 Punkt

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

| 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.        |

## Frage 8

ID: Q-20-04-03

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

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

ID: Q-20-04-12

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

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?

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

## Frage 10

ID: Q-20-04-03

P-Frage: Wählen Sie die **drei** besten Optionen aus 2 Punkte

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.

**Explanation:** Concerning option (c): It's **not** our task to *reject* requirements just because they contain risks. We should identify and communicate those risks, but not reject such requirements.

## Frage 11

ID: Q-20-04-07

P-Frage: Wählen Sie die **drei** besten Optionen aus 2 Punkte

Sie sind als Architekt:in 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) Ihren Vorgesetzten technologische Updates zusätzlich zu den Geschäftsanforderungen vorschlagen

## Frage 12

ID: Q-21-05-01

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

Welche der folgenden Aussagen zu Architekturentscheidungen sind wahr, welche falsch?

- | 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.  |



### Frage 13

ID: Q-20-04-09

K-Frage: Wählen Sie für jede Zeile „richtig“ oder „falsch“ aus. 1 Punkt

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

| 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. |

### Frage 14

ID: Q-20-04-10

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

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

| 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.                                   |

Explanation:

Business stakeholder might very well have goals like performance, flexibility or security, which are

considered "architecture goals".

## Frage 15

ID: Q-20-04-11

---

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

---

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

- (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.

---

## Frage 16

ID: Q-20-04-19

---

P-Frage: Wählen Sie die **drei** am besten passenden Antworten aus 1 Punkt

---

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

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

## Frage 17

ID: Q-20-04-32

---

P-Frage: Wählen Sie die **drei** am besten passenden Antworten aus 1 Punkt

---

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

ID: Q-20-04-38

---

P-Frage: Wählen Sie die **drei** am häufigsten verwendeten Architektursichten 1 Punkt

---

Diverse Architekturentwicklungsmethoden schlagen einen sichtenbasierten Ansatz vor. Welche DREI der folgenden Sichten werden am häufigsten verwendet?

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

## Frage 19

ID: Q-20-04-22

---

|          |  |         |
|----------|--|---------|
| P-Frage: | Wählen Sie die <b>zwei</b> am besten passenden Antworten aus | 1 Punkt |
|----------|--|---------|

---

Sie dokumentieren einen Baustein einer Softwarearchitektur. Welche zwei 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

ID: Q-20-04-17

---

|          |  |         |
|----------|--|---------|
| P-Frage: | Wählen Sie die <b>zwei</b> am besten passenden Antworten aus | 1 Punkt |
|----------|--|---------|

---

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 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.

In most cases it is unrealistic to have *complete* requirements specification. Often it is enough to have an overview and know certain details (e.g. quality requirements).

## Frage 21

ID: Q-20-04-18

---

P-Frage: Wählen Sie die **drei** am besten passenden Antworten aus 1 Punkt

---

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

ID: Q-20-04-28

---

A-Frage: Wählen Sie eine Option aus 1 Punkt

---

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

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

**Frage 23****ID: Q-20-04-33**

---

|          |                            |         |
|----------|----------------------------|---------|
| A-Frage: | Wählen Sie eine Option aus | 1 Punkt |
|----------|----------------------------|---------|

---

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

- (a) Systeme mit harten Echtzeitanforderungen
  - (b) Regelbasierte Systeme
  - (c) Linnés-Systeme
  - (c) Sicherheitskritische Systeme
- 

**Frage 24****ID: Q-20-04-20**

---

|          |                            |         |
|----------|----------------------------|---------|
| A-Frage: | Wählen Sie eine Option aus | 1 Punkt |
|----------|----------------------------|---------|

---

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

ID: Q-20-04-21

K-Frage: Wählen Sie für jede Zeile „enge Kopplung“ oder „lose Kopplung“ aus. 1 Punkt

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

| 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.                                  |

## Frage 26

ID: Q-20-04-14

P-Frage: Wählen Sie die **zwei** am besten passenden Antworten aus 2 Punkte

Welche zwei 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?

- (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.



## Frage 27

ID: Q-20-04-15

K-Frage: Wählen Sie für jede Zeile „richtig“ oder „falsch“ aus. 2 Punkte

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                              |     |  |
|-------------------------------------|-------------------------------------|-----|--|
| <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. |

- Sometimes verbal communication needs to come first, there is no general rule.
- Feedback should not be restricted to written statements.

## Frage 28

ID: Q-20-04-37

K-Frage: Wählen Sie für jede Zeile „richtig“ oder „falsch“ aus. 2 Punkte

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

| wahr                                | falsch                              |     |  |
|-------------------------------------|-------------------------------------|-----|--|
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | (a) | Business Process Model & Notation (BPMN) sollte nur von Business-Analysten und nicht zur Architekturdokumentation 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.   |

- (d)                      Solange die Notation (z. B. mithilfe einer Legende) erläutert wird, kann eine beliebige Notation ausreichen, um Baueinstrukturen und Zusammenarbeit zu beschreiben.

## Frage 29

ID: Q-20-04-13

---

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

---

Welche zwei Architektursichten haben die beste praktische Anwendung für die Entwicklung von Softwarearchitekturen?

- (a) Pattern-Sicht.
- (b) Beobachtersicht (Observer-View).
- (c) Bausteinsicht (Komponentensicht).
- (d) Verteilungssicht.

## Frage 30

ID: Q-20-04-23

---

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

---

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 ("business context").

## Frage 31

ID: Q-20-04-24

P-Frage: Wählen Sie die **zwei** am besten passenden Gründe 1 Punkt

Die Dokumentation der Softwarearchitektur 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 innerhalb der gleichen Organisation erneut verwendet werden.
- (d) Querschnittskonzepte sollten von Spezialist:innen implementiert werden. Daher ist eine separate Dokumentation hilfreich.

## Frage 32

ID: Q-20-04-25

K-Frage: Wählen Sie für jede Zeile „richtig“ oder „falsch“ aus. 2 Punkte

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

- | 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. |

### Explanation

Regarding option (e), "identical behavior in all aspects": It's technically not feasible to have *identical* behavior, at least concerning latency, and response time.

A more detailed explanation can be found in the (rather famous) [Fallacies\\_of\\_distributed\\_computing](#)

### Frage 33

ID: Q-20-04-26

K-Frage: Wählen Sie für jede Zeile „richtig“ oder „falsch“ aus. 1 Punkt

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                              |     |   |
|-------------------------------------|-------------------------------------|-----|---|
| <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. |

Not *all* decisions need to be justified in writing - as the requirement for *written* documentation depends on the situation, the team, the system and other factors.

### Frage 34

ID: Q-20-04-31

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

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

| 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.                   |

- (f) Sicherstellung der gleichmäßigen Auslastung des Teams.

### Frage 35

ID: Q-20-04-30

---

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

---

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

| 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 – Einhaltung gesetzlicher Vorgaben ("Compliance"). |

### Frage 36

ID: Q-20-04-27

---

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

---

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

- (a) Durch entwickeln von UI-Prototypen.
- (b) Durch definieren expliziter Schnittstellen.
- (c) Durch erstellen, verfassen oder diskutieren von Szenarien.
- (d) Durch erstellen automatisierter Tests.
- (e) Durch erstellen eines Qualitätsbaums.

**Frage 37****ID: Q-20-04-28**

---

|          |  |          |
|----------|--|----------|
| P-Frage: | Wählen Sie die <b>vier</b> am besten passenden Antworten aus | 2 Punkte |
|----------|--|----------|

---

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

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

**Frage 38****ID: Q-20-04-29**

---

|          |  |          |
|----------|--|----------|
| P-Frage: | Wählen Sie die <b>zwei</b> am besten passenden Antworten aus | 2 Punkte |
|----------|--|----------|

---

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**

ID: Q-20-04-36

---

|          |  |         |
|----------|--|---------|
| P-Frage: | Wählen Sie die <b>drei</b> am besten passenden Antworten aus | 1 Punkt |
|----------|--|---------|

---

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

- (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.

**Explanation**

- Size can easily and reliably be measured when statically analyzing source code (lines-of-code metric is a reliable size metric)
  - change-rate and number-of-developers-per-component can reliably be measured when taking the version control history into account, which is perfectly feasibly with systems like git, subversion or similar tools that are widely used in development.
-