

Abschlussprüfung Winter 2022/23

Lösungshinweise



Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung (AO 2020)
1201



Planen eines
Softwareproduktes

Teil 2 der Abschlussprüfung

Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.).

Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1 = 100 – 92 Punkte
Note 3 = unter 81 – 67 Punkte
Note 5 = unter 50 – 30 Punkte

Note 2 = unter 92 – 81 Punkte
Note 4 = unter 67 – 50 Punkte
Note 6 = unter 30 – 0 Punkte

1. Aufgabe (28 Punkte)

a) 4 Punkte

Erwartet werden Beschreibungen wie z. B.:

- Netzwerkinfrastruktur der Standorte
- Bereitstellung der Dienste über zwei Rechenzentren
- Unterschiedliche Hardware
- Datenengpass (Bandbreitenproblem)
- Bereitstellung der Images über Serverinstallation

b) 4 Punkte

Erwartet werden Beschreibungen zu funktionalen Anforderungen wie z. B.:

- Standard-Office-Dienste
- Sichere Datenübertragung (über VPN-Punkte)
- Geeignete Hardware
- Leistungsfähige Netzinfrastruktur

Erwartet werden Beschreibungen zu nicht funktionalen Anforderungen wie z. B.:

- Performance
- Verfügbarkeit
- Speicherkapazität
- Stabilität der Services

ca) 2 Punkte

Ein möglicher Standard-Weg ist hier der Einsatz von Pipelines im Rahmen von Continuous Integration und Deployment (CI/CD). Die Pipelines können bei verschiedenen Ereignissen, z. B. dem Erstellen eines PRs oder dem Taggen eines Commits, ausgelöst werden.

cb) 4 Punkte

z. B. ausgehend von CI/CD:

Vorteile:

- Automatische Ausführung von Unit-Tests in einer gleichbleibenden Testumgebung vor Deployment möglich
- Automatische und einheitliche Bereitstellung auf verschiedene Umgebungen für Tests
- Nicht jeder Entwickler benötigt direkten Zugriff auf Staging- und Production-Systeme, da die Zugänge in der Pipeline hinterlegt sind.
- Ein Rollback ist i. d. R. mit wenigen Klicks/Befehlen möglich, in dem die jeweilige Pipeline mit dem alten Stand erneut ausgeführt wird.

d) 5 Punkte (1 Punkt für die Nennung der hybriden Verschlüsselung, 4 Punkte für die Erläuterung)

TLS (Transport Layer Security) arbeitet nach dem Prinzip der hybriden Verschlüsselung. Dabei werden im Rahmen des TLS-Handshakes zunächst die Public-Keys ausgetauscht, um die weitere Kommunikation asymmetrisch zu verschlüsseln. Nachdem sich Client und Server auf einen Shared-Key geeinigt haben, wird die weitere Verbindung symmetrisch verschlüsselt. Die hybride Verschlüsselung wird verwendet, da das asymmetrische Verfahren sehr rechenaufwendig ist und man somit die Vorteile aus beiden Verfahren (symmetrisch und asymmetrisch) kombinieren kann.

e) 3 Punkte

Mithilfe einer lokalen Certificate Authority können die Zertifikate der jeweiligen Dienste signiert werden. Die Software kann dann überprüfen, ob das verwendete Zertifikat von der lokalen Certificate Authority signiert und nicht widerrufen wurde.

f) 6 Punkte

Erwartet werden Erläuterungen bspw. zu den nachfolgenden Vor- und Nachteilen.

Vorteile:

- Zentrale Rechteverwaltung ermöglicht bessere Übersicht über erteilte Rechte.
- Nutzer müssen sich nicht verschiedene Benutzernamen und Passwörter merken -> geringere Gefahr, dass diese aufgeschrieben werden
- Bei Sicherheitsvorfällen kann schnell und an zentraler Stelle reagiert werden (z. B. Nutzer sperren, Passwörter ändern).

Nachteile:

- Mit einem Zugang erhalten Angreifer Zugriff auf verschiedenste Systeme.
- Abhängigkeit von der Verfügbarkeit des Identity Providers (bspw. bei Systemausfall)

2. Aufgabe (22 Punkte)

a) 6 Punkte

Webanwendungen bestehen typischerweise aus einem Webserver, welcher die Webanwendung hostet, einem Backend-Server und einer Datenbank.

Die Webanwendung fragt Daten bei dem Backend-Server an, welcher selbst die Daten in der Datenbank abfragt und dann an die Webanwendung zurückliefert.

ba) 1 Punkt

Scrum, ... (Andere agile Vorgehensmodelle sind ebenfalls gültig.)

bb) 6 Punkte

Die Anforderungen werden in den Sprints so umgesetzt, dass diese am Ende eines Sprints abgeschlossen sind. Es wird sich im Team jeden Tag im Daily-Scrum getroffen und kurz über die Arbeit am Sprint geredet. Nach einem Sprint wird dem Kunden der neue Stand gezeigt.

Bevor der neue Sprint beginnt, überlegt sich das Team, wie es seine Arbeitsweise verbessern kann.

- Daily-Scrum: Dauer ca. 15 Minuten, Inhalt: Was habe ich gestern gemacht, was möchte ich heute machen ...
- Sprint: Die Entwicklung wird in Iterationen organisiert, Länge sollte immer gleich sein, zwischen 2 und 4 Wochen.
- Die Liste der Anforderungen steht im Product Backlog. Jede Anforderung wird User-Story genannt.
- Scrum-Master: ist verantwortlich für die Sicherstellung, dass das Team operativ und produktiv arbeiten kann
- u. a.

Andere Elemente des entsprechend gewählten Vorgehensmodells sind auch korrekt.

ca) 3 Punkte

Bei Open-Source ist der Quelltext von Programmen oder Bibliotheken öffentlich einsehbar.

cb) 2 Punkte

Da Open-Source-Software mit Lizenzen versehen werden kann, darf man nicht automatisch jede Open-Source-Software einsetzen.

cc) 4 Punkte

Vorteil: Jeder kann nachvollziehen, was in der Software/Bibliothek gemacht wird. Man weiß also, wie die Software arbeitet.

Nachteil: Jeder kann den Quelltext lesen und eventuelle Schwachstellen zum Angreifen von Software nutzen, welche diese Open-Source-Software einsetzen.

4. Aufgabe (30 Punkte)

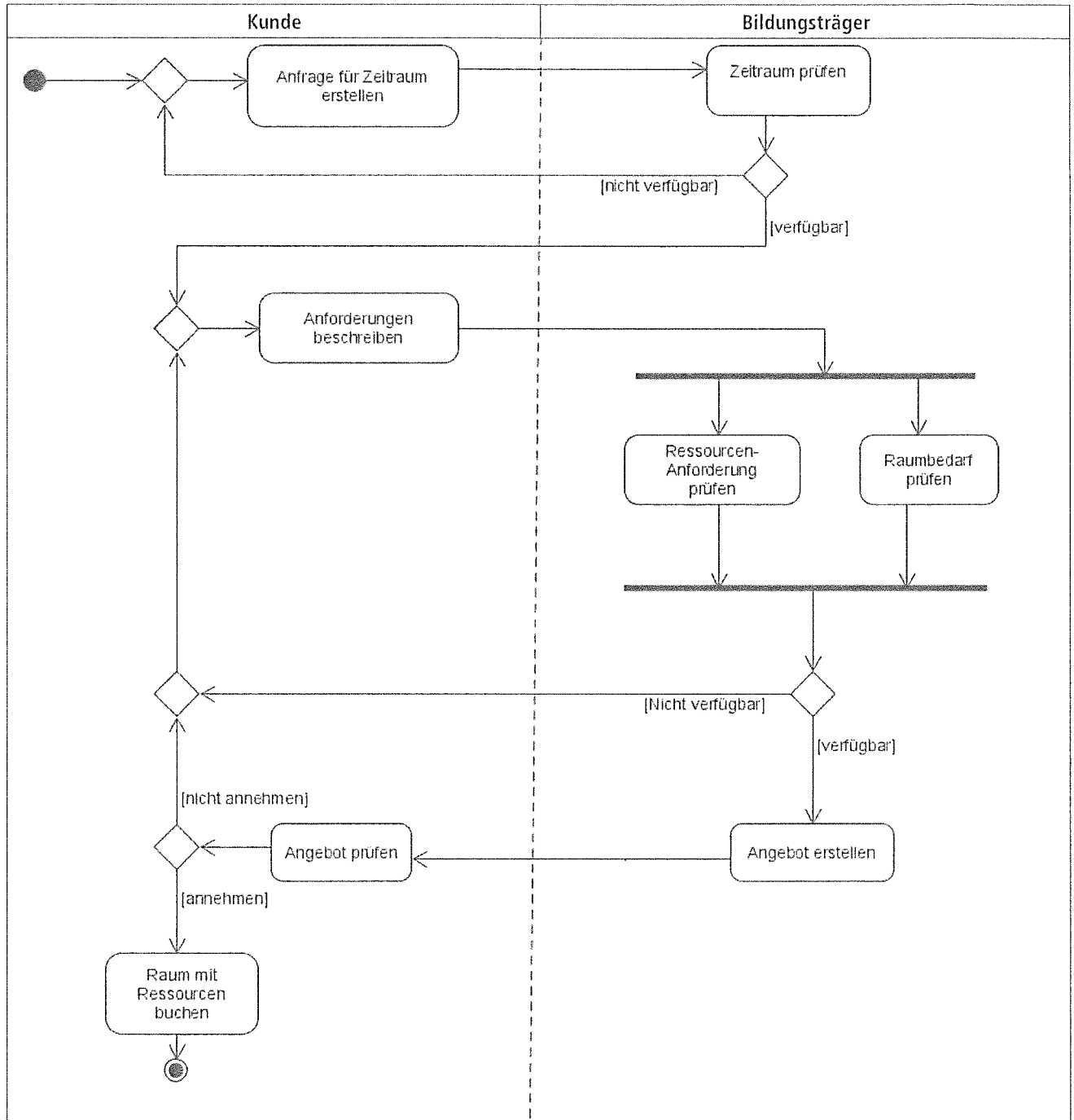
a) 14 Punkte

Vorschlag für die Punkteverteilung:

Je Aktivität 1 Punkt

Richtige Verteilung der Aktivitäten auf die Seiten 1 Punkt

Je Knoten 0,5 Punkte



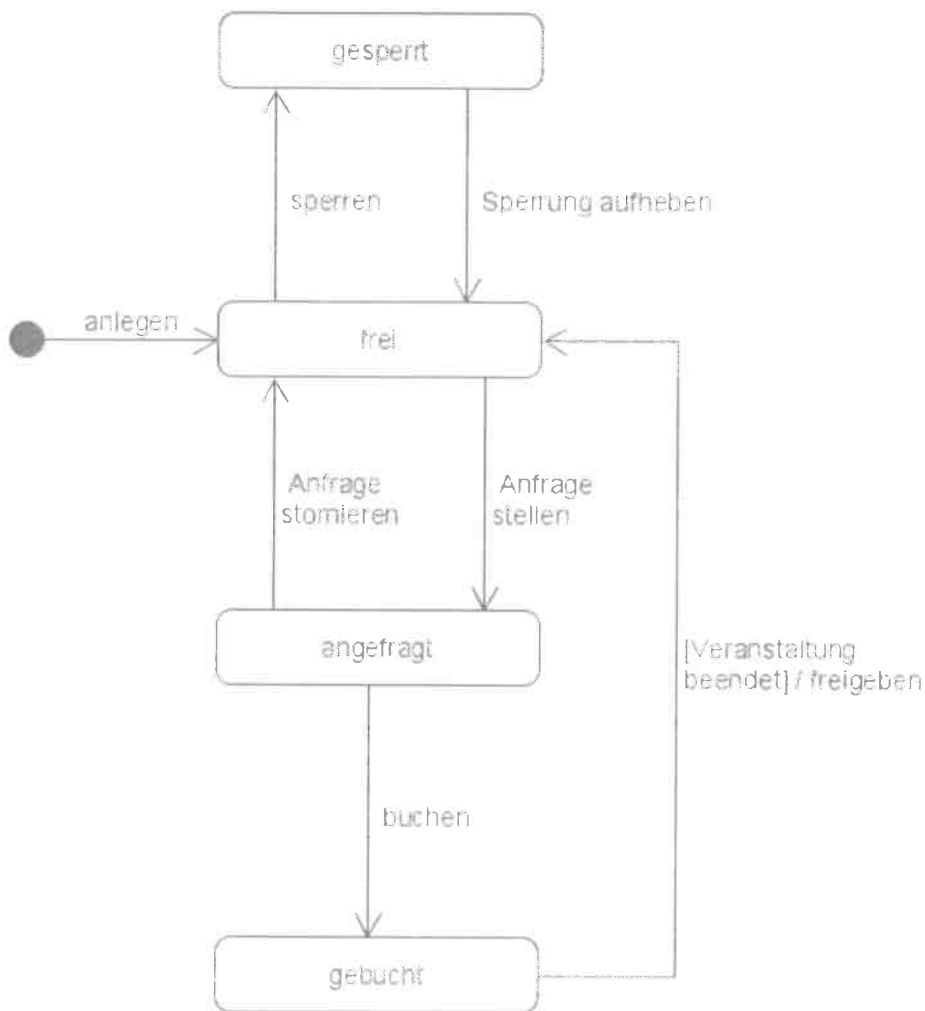
Andere Lösungen sind möglich.

b) 11 Punkte

Vorschlag für die Punkteverteilung:

Je Zustand 1 Punkt

Je Zustandsübergang 1 Punkt



Andere Lösungen sind möglich.

c) 5 Punkte, Entscheidung für eine Variante; 1 Punkt; je Argument 2 Punkte

Mögliche Argumente für NoSQL

- Flexible Datenstruktur, z. B. unterschiedliche Raumtypen mit unterschiedlichen Ausstattungen
- Datenmenge kann groß werden

Mögliche Argumente für relationale DB

- Transaktionssicherheit
- Datenkonsistenz immer gewährleistet

Andere Antworten sind möglich.