

## Aufgabe 2

### A Allgemein

**Aufgabe 1** Im Software Engineering geht es vor allem darum qualitativ hochwertige Software zu entwickeln.

**Aufgabe 2** Software Engineering ist gleichbedeutend mit Programmieren.

### B Vorgehensmodelle

**B1** Die Erhebung und Analyse von Anforderungen sind nicht Teil des Software Engineerings.

**B2** Agile Methoden eignen sich besonders gut für die Entwicklung komplexer und sicherer Systeme in verteilten Entwicklerteams.

**B3** Das Spiralmodell ist ein Vorläufer sogenannter Agiler Methoden.

### C Anforderungserhebung

**C1** Bei der Anforderungserhebung dürfen in keinem Fall mehrere Erhebungstechniken (z. B. Workshops, Modellierung) angewendet werden, weil sonst Widersprüche in Anforderungen zu Vorschein kommen könnten.

**C2** Ein Szenario beinhaltet eine Menge von Anwendungsfällen.

**C3** Nicht-funktionale Anforderungen sollten, wenn möglich, immer quantitativ spezifiziert werden.

### D Architekturmuster

**D1** Schichtenarchitekturen sind besonders für Anwendungen geeignet, in denen Performance eine wichtige Rolle spielt.

**D2** Das Black Board Muster ist besonders für Anwendungen geeignet, in denen Performance eine wichtige Rolle spielt.

**D3** „Dependency Injection“ bezeichnet das Konzept, welches Abhängigkeiten zur Laufzeit reglementiert.

### E UML

**E1** Sequenzdiagramme beschreiben Teile des Verhaltens eines Systems.

**E2** Zustandsübergangsdiagramme beschreiben das Verhalten eines Systems.

**E3** Komponentendiagramme beschreiben die Struktur eines Systems.

### F Entwurfsmuster

**F1** Das MVC Pattern verursacht eine starke Abhängigkeit zwischen Datenmodell und Benutzeroberfläche.

**F2** Das Singleton Pattern stellt sicher, dass es zur Laufzeit von einer bestimmten Klasse höchstens ein Objekt gibt.

**F3** Im Kommando Entwurfsmuster (engl. „Command Pattern“) werden Befehle in einem sog. Kommando-Objekt gekapselt, um sie bei Bedarf rückgängig zu machen.

**G** Testen

**G1** Validation dient der Überprüfung von Laufzeitfehlern.

**G2** Testen ermöglicht sicherzustellen, dass ein Programm absolut fehlerfrei ist.

**G3** Verifikation dient der Überprüfung, ob ein System einer Spezifikation entspricht.