

Staatsexamen 66116 / 2020 / Frühjahr

**Thema 2 / Teilaufgabe 1 / Aufgabe 1**

(Wissensfragen)

**Stichwörter:** Wasserfallmodell, Prozessmodelle, Kontinuierliche Integration (Continuous Integration), EXtreme Programming, Softwaremaße, SCRUM

- (a) Nennen Sie fünf Phasen, die im Wasserfallmodell durchlaufen werden sowie deren jeweiliges Ziel bzw. Ergebnis(-dokument).

**Anforderung** Lasten- und Pflichtenheft**Entwurf** Softwarearchitektur in Form von Struktogrammen, UML-Diagrammen etc.**Implementierung** Software**Überprüfung** überarbeitete Software**Einsatz und Wartung** erneut überarbeitete Software, verbesserte Wartung

- (b) Nennen Sie drei Arten der Softwarewartung und geben Sie jeweils eine kurze Beschreibung an.

**korrektive Wartung** Korrektur von Fehlern, die schon beim Kunden in Erscheinung getreten sind**präventive Wartung** Korrektur von Fehlern, die beim Kunden noch nicht in Erscheinung getreten sind**adaptive Wartung** Anpassung der Software an neue Anforderungen**perfektionierende Wartung** Verbesserung von Performance und Wartbarkeit und Behebung von technical debts <sup>a</sup><sup>a</sup><https://de.wikipedia.org/wiki/Softwarewartung>

- (c) Eine grundlegende Komponente des *Extreme Programming* ist „*Continuous Integration*“. Erklären Sie diesen Begriff und warum man davon einen Vorteil erwartet.

Die Software wird nach jeder Änderung (push) automatisch kompiliert und auch getestet. Man erwartet sich davon einen Vorteil, weil Fehler schneller erkannt werden und die aktuelle Version des Systems ständig verfügbar ist.

- (d) Nennen Sie zwei Softwaremetriken und geben Sie jeweils einen Vor- und Nachteil an, der im Projektmanagement von Bedeutung ist.

**Anzahl der Code-Zeilen****Vorteil:** Ist sehr einfach zu bestimmen.**Nachteil:** Entwickler können die Zeilenzahl manipulieren (zusätzliche Leerzeilen, Zeilen aufteilen), ohne dass sich die Qualität des Codes verändert.**Zyklomatische Komplexität****Vorteil:** Trifft eine Aussage über die Qualität des Algorithmus.**Nachteil:** Ist für Menschen nicht intuitiv zu erfassen. Zwei Codes, die dasselbe Problem lösen können die gleiche Zyklomatische Komplexität haben, obwohl der eine wesentlich schlechter zu verstehen ist (Spaghetticode!).

- (e) Nennen und beschreiben Sie kurz drei wichtige Aktivitäten, welche innerhalb einer Sprint-Iteration von Scrum durchlaufen werden.

**Sprint Planning** Im Sprint Planning werden zwei Fragen beantwortet:

- Was kann im kommenden Sprint entwickelt werden?
- Wie wird die Arbeit im kommenden Sprint erledigt?

Die Sprint-Planung wird daher häufig in zwei Teile geteilt. Sie dauert in Summe maximal 2 Stunden je Sprint-Woche, beispielsweise maximal acht Stunden für einen 4-Wochen-Sprint.

**Daily Scrum** Zu Beginn eines jeden Arbeitstages trifft sich das Entwicklerteam zu einem max. 15-minütigen Daily Scrum. Zweck des Daily Scrum ist der Informationsaustausch.

**Sprint Review** Das Sprint Review steht am Ende des Sprints. Hier überprüft das Scrum-Team das Inkrement, um das Product Backlog bei Bedarf anzupassen. Das Entwicklungsteam präsentiert seine Ergebnisse und es wird überprüft, ob das zu Beginn gesteckte Ziel erreicht wurde.

- (f) Erläutern Sie den Unterschied zwischen dem Product-Backlog und dem Sprint-Backlog.

**Product Backlog** Das Product Backlog ist eine geordnete Auflistung der Anforderungen an das Produkt.

**Sprint Backlog** Das Sprint Backlog ist der aktuelle Plan der für einen Sprint zu erledigenden Aufgaben.

- (g) Erläutern Sie, warum eine agile Entwicklungsmethode nur für kleinere Teams (max. 10 Personen) gut geeignet ist, und zeigen Sie einen Lösungsansatz, wie auch größere Firmen agile Methoden sinnvoll einsetzen können.

Bei agilen Entwicklungsmethoden finden daily scrums statt, die in der intendierten Kürze nur in kleinen Teams umsetzbar sind. Außerdem ist in größeren Teams oft eine Hierarchie vorhanden, die eine schnelle Entscheidungsfindung verhindert. Man könnte die große Firma in viele kleine Teams einteilen, die jeweils an kleinen (Teil-)Projekten arbeiten. Eventuell können die product owner der einzelnen Teams nach agilen Methoden zusammenarbeiten.