

Klausur

Name:

[REDACTED]

Matrikelnr.:

[REDACTED]

Die Klausur besteht aus 4 Aufgaben. Verschaffen Sie sich einen kurzen Überblick über die Aufgaben. Beginnen Sie mit der Aufgabe, die Ihnen am ehesten ein Erfolgserlebnis zu bringen scheint.

- Bei Verständnisfragen heben Sie bitte den **Arm**; ich bemühe mich dann um eine Klärung.
- Es sind **keine Hilfsmittel** zugelassen.
- Bei **Täuschungsversuchen** wird Ihre Klausur mit 0 Punkten gewertet!
- Schreiben Sie **leserlich**! Nicht lesbare oder unklare Teile werden mit 0 Punkten bewertet.
- Der vorgesehene **Platz hinter den Aufgaben** sollte für die Antwort **ausreichen**. Schreiben Sie zur Not (und nur zur Not) auf der Rückseite weiter. Zusätzliche Blätter sind nicht erlaubt.

Viel Erfolg !

1	2	3	4	Summe
55	40	50	60	205

14P

Aufgabe 1 (Verständnisfragen) bitte kurz und präzise beantworten:

- a) Was ist die wichtigste Aufgabe eines Projektleiters?

Kommunikation mit allen Beteiligten und auf die unterschiedlichen Bedürfnisse eingehen können (von Entwicklern, Kunde, Vorstand, ...). Hierzu gute Gesprächsführung nötig, d.h. Zielorientierung, ankluge Fragen stellen, aktiv Zuhören.

- b) Was ist der Unterschied zwischen einem Problem und einem Risiko?

Ein Risiko ist ein mögliches Problem, welches mit einer gewissen Wahrscheinlichkeit eintreten kann.
Ein Problem ist eingetreten und „aktuell“.

- c) Nennen sie ein Qualitätsmerkmal eines Programms (z.B. aus ihrem Praktikumsprojekt) mit einer für genau das angegebene Merkmal relevanten Metrik.

Qualitätsmerkmal: Dokumentation der Implementierung / Kodedokumentation




Metrik: Anteil des LOC, welcher Kommentartext ist.

- d) Beschreiben Sie
- kurz**
- ein Anwendungsszenario, in dem es Nachteile bei der Verwendung des Composite-Pattern geben kann und erläutern Sie einen Nachteil.

Anwendungsszenario:

Behälter ist „Vase“ „Geometrische Form“
Bestandteile sind Quadrat, Rechteck, Kreis, Kreis

Nachteil:

Typprüfung zur Laufzeit, weil sich z.B. die ^{Fläche} Seitenlänge eines Quadrats anders berechnet als die ^{Fläche} Seitenlänge eines Rechtecks.
Fläche eines  
Kreis 

- e) Nennen Sie eine typische Methode zur Durchführung der Aufwandsschätzung in IT-Projekten und erläutern Sie diese
- kurz**
- .

Macro-schätzung

Es gibt ein/wenige Vergleichsprojekt(e). Aus diesen werden Daten gesammelt bzgl. des Zeit- und Budgetaufwands und eine Cost-Datenbank erstellt. Das aktuell geplante Projekt wird mit ~~den~~ ^{den} bzw. die Bestandteile werden mit den gesammelten Daten verglichen und aus möglichst ähnlichen Aufgaben von früher werden Schätzungen abgeleitet, Aufgaben, die 1:1 gleich sind, geht es quasi nicht, deshalb muss mit Faktoren geschätzt werden zur Anpassung.

Aufgabe 2 (Metriken):

Geben Sie den Quellcode eines Programms (in Java oder Pseudocode) an, das eine McCabe-Zahl von genau 4 hat.

Programmstart

```
for 1 to 10 do
  print 1
endfor
```

```
for 1 to 10 do
  print 1
endfor
```

```
for 1 to 10 do
  print 1
endfor
```

Programmende

Nur Einführung wäre 1
Pro Schleife + 1

40

Aufgabe 3 (Projektmanagement - Projektstrukturplan):

Ein Projektstrukturplan (PSP) kann baumartig dargestellt werden.

a) Was wird in die Blätter des PSP eingetragen?

Arbeitspakete, bearbeitbar von Teil des Teams in definierter Zeit und klarem Umfang. 10

b) Was wird in die nicht-Blätter des PSP eingetragen?

Oberaufgaben, „Aufgabemengen“, die eine größere Granularität als ein Arbeitspaket haben und nicht in dieser Größe „als ein“ bearbeitet werden. 10

c) Stellen Sie eine „Methode“ dar, wie man die Struktur eines PSP für ein gegebenes Projekt ableiten kann.

(Für Ihre Methode bitte aufschreiben: Was brauchen Sie als „Daten“ für die Methode und was machen Sie damit, um zum PSP zu kommen?)

Zum Zeitpunkt der PSP-Erstellung befindet man sich i. d. R. in der Initialisierungsphase, d. h. der Kunde hat das Angebot angenommen. Wenn gar keine Arbeit geleistet wurde ist der Umfang des Projekts klar, ein Sinn von „Arbeit“, was soll in dem Projekt und insbesondere auch „was nicht“. Die Aufgaben/ Anforderungen an die SW sind auf jeden Fall Daten für die Methode. Außerdem müssen die Ressourcen als Daten einfließen, d. h. Zeit, Budget, Personal (mit welchem Fachwissen?). 30

Die Anforderungen werden dann in große Kategorien unterteilt wie „UI“, „Datenbank“. Neben der Entwicklung fließen auch Dinge aus dem Projektmanagementtrichter als Daten ein, wie z. B. wann es und wie oft mit dem Kunde ein Release möchte. Man erstellt ein PSP, indem man möglichst grobgranular die anfallenden Aufgaben auflistet. Mit den oben genannten Daten bietet sich eine Unterteilung in „Projektmanagement“ und „Entwicklung“ an. Pro Ebene gilt es nun immer feingranuläre Unterteilungen (z. B. nach „Entwicklung“ dann UI, Frontend und Backend) bis man so feingranular genug ist, um ein Arbeitspaket zu erhalten, d. h. von der nach unten entsteht ein breiter werdender Baum, in dem alle Unterknoten eines Oberknotens die „Aufgabemenge“ des Oberknotens erfüllen. „Oberaufgabe“ des Oberknotens lösen. Bei den Arbeitspaketen werden nun die Ressourcen zugeordnet, die ja auch als Daten übergeben wurden, d. h. wie macht die Aufgabe, wie lange dauert das, wie teuer. → PSP fertig

Aufgabe 4: (Qualität und Konfigurationsmanagement)

a) Erläutern Sie einen Vorteil von Whitebox/Glassbox-Tests gegenüber Blackbox-Tests und einen Nachteil.

Vorteil:

Whitebox-Test hat Team-, Zweig- und Anweisungüberdeckung, d.h. „jeder mögliche Weg“ innerhalb der Komponente wird einmal (mindestens) begangen. Dadurch kann überprüft werden, ob die innere Logik einer Komponente richtig rechnet, während beim Blackbox-Test ein richtiges Ergebnis auch durch einen inneren Fehler trotzdem zufällig erzeugt werden sein könnte. 10

Nachteil:

Erstellung erst nach / während Implementierung möglich, da Innenleben ^{Durchsuchbarkeit} ~~schon~~ ^{wird}, bei Blackbox schon schon bei Spezifikation der Klasse / Komponente. 20

b) Welche Teile umfasst das Konfigurationsmanagement?
(Teile benennen und kurz erläutern)

Release Management

Zuständig für das Ausliefern lauffähiger Konfigurationen (Software) an den Kunden z.B.) 30

Build Management

Zuständig für das (Zusammen-) Bauen lauffähiger Software(-Einheiten)/Komponenten (z.B. mithilfe von Tools wie Maven oder Gradle) 20

Change Management

Zuständig für das Durchführen erfolgreicher Änderungen an der Software und insbesondere für Änderungswünsche vom Kunden. 10

Version Management

Zuständig für die Versionierung von Software- (und Dokumenten) Einheiten und Dokumenten → Nachvollziehbarkeit, Rückverfolgung möglich