

# Erstklausur Software Engineering

WiSe18/19

Zeit: 100 Minuten (+ 5 Minuten Verlängerung)

*PDF Version 3*

Gesamtpunkte: 100

erlaubte Hilfsmittel: nicht programmierbarer Taschenrechner & beidseitig handschriftlich beschriebenes A4 Blatt

Punkteverteilungen sind aus dem Gedächtnis, ggf. anders

## **1 Aufgabe 1 - Metriken [20]**

### **1.1 Softwaremetriken**

Nennen sie 3 Motivationen von Software Metriken.

### **1.2 Lines Of Code**

Erklären sie die folgenden Metriken:

- LoC
- SLoC
- LLoc

### **1.3 Spiralmodell**

Erklären sie das Spiralmodell.

## 2 Aufgabe 2 - COCOMO [15]

Sie sind in einer Firma, welches autonomes Fahren entwickelt. Es setzt sich aus 3 Modulen zusammen, welche intensive zusammenarbeiten um Ergebnisse zu erzielen. Der erste Teil schreibt Code mit der Länge von 300000 Delivered Source Instructions. Das zweite Modul erbringt ca 100000 DSI und das dritte ca 150000 DSI.

Die Programmierer des Unternehmens sind in der Programmiersprache gut erfahren. Außerdem muss das Produkt sehr sehr zuverlässig sein, aufgrund des Anwendungsbereichs des autonomen Fahrens. Verwenden sie das intermediate "Verfahren für Berechnung.

### 2.1 M

Berechnen sie den Kostenfaktor **M**. Runden sie auf 2 Nachkommastellen.

### 2.2 E

Berechnen sie den Projektaufwand für das gegebene Unternehmen.

### 2.3 T

Berechnen sie die Projektdauer für das gegebene Unternehmen.

### 2.4 H

Berechnen sie die Anzahl der benötigten Mitarbeiter **H**.

### 2.5 P

Berechnen sie die Produktivität.

### 3 Use Cases [5+10]

Es soll ein Use Case Diagramm für ein soziales Netzwerk entwickelt werden. Nutzern ist es möglich Beiträge zu verfassen und diese sich anzusehen. Zudem können Nachrichten als besonders gut markiert werden.

Wird eine Nachricht als besonders gut markiert, so wird der Verfasser benachrichtigt. Zudem wird er ebenfalls über E-Mail benachrichtigt, wenn er E-Mail Benachrichtigungen angeschaltet hat.

Zudem gibt es die besondere Rolle "Moderator", welchen es möglich ist Beiträge zu verschieben und zu löschen.

#### 3.1 Use Case Diagramm

Erstellen sie ein Use Cases Diagramm für das soziale Netzwerk.

#### 3.2 Use Cases

Ihnen sind zwei Formulare gegeben. Füllen sie diese passend für zwei unterschiedliche Use Cases aus und ergänzen sie falls nötig sinnvoll.

Akteure	
Beschreibung	
Trigger	
Vorbedingung	
Nachbedingung	
normal Flow	
alternative Flow	
Exceptions	

Akteure	
Beschreibung	
Trigger	
Vorbedingung	
Nachbedingung	
normal Flow	
alternative Flow	
Exceptions	

## **4 Zustandsdiagramm [10]**

Sie sollen mittels eines Zustandsdiagramms den Ablauf einer Werkstatt modellieren. Der Vorgang ist in 4 Phasen aufgeteilt: Vormontage, Hauptmontage, Batteriemontage, Endmontage.

In der ersten Phase wird die Karosserie zusammengebaut. Die Haupt- und Batteriephase finden gleichzeitig statt. In der Hauptphase wird der Motor in die Karosserie eingelassen. In der Batteriephase werden mehrere Batterien in das Auto eingelassen. Jede Batterie wird einzeln eingebaut. Nachdem eine Batterie eingelassen wurde, wird gecheckt, ob nun genug Batterien eingebaut wurden, falls ja werden diese verkabelt.

In der letzten Phase wird die Karosserie lackiert und der Prozess beendet.

## 5 JAUS [7+8]

Zeigen sie für die gegebenen Terme welchen Typ diese haben mittels eines Herleitungsbaums. Ist eine Typbestimmung nicht möglich, zeigen sie das mithilfe eines Herleitungsbaums.

Gehen sie von folgendem Raum aus  $A = \{\emptyset, u : int, v : boolean, w : int, x : int\}$ .

### 5.1 a)

$$((43 + w) + v)$$

### 5.2 b)

$$((u + w) + (16 + x))$$

## 6 JUnit [3+3+9]

```
1 import java.util.List;
3 import java.util.ArrayList;
4
5 public class A {
6     private List<String> s;
7
8     public A() {
9         s = new ArrayList<String>();
10        s.add("Bob");
11        s.add("Alice");
12        s.add("Eve");
13    }
14
15    public String s(B b) {
16        int t = b.t();
17        String r = "Hello ";
18        for(String z : s) {
19            boolean x = b.f(t, 5);
20            if (x) {
21                r += z;
22            }
23        }
24        return r;
25    }
26 }
```

```
1 public interface B {
2     int t(); //Komplexe Berechnung
3
4     boolean f(int a, int b); //Komplexer Algorithmus
5 }
```

### 6.1 Fehler in A

Welchen Fehler finden sie in A?

Erklären sie warum dieser Fehler hinderlich ist beim Testen.

### 6.2 Behebung Fehler

Wie behebt man den, in der obigen Aufgabe beschriebenen, Fehler, ohne die Funktionalität des Programmes zu verändern mit möglichst wenig Aufwand.

Schreiben sie ihren Code nieder.

### 6.3 Test

Schreiben sie, wie in der Vorlesung beschrieben, einen JUnit 5 Test, welcher testen soll, ob der Aufruf von s in A "Hello Bob Alice Eve" ausgibt.

*[Kommentar: JUnit 4 war in der Vorlesung dran, nicht JUnit 5, ich habe Sasha darauf aufmerksam gemacht. Er meinte er wird für die Verwendung von JUnit 4 keine Punkte abziehen.]*

## 7 IBAN Klasse [10]

Erstellen sie eine Klasse, welche den europäischen Standard der IBAN implementiert. Ihre Implementierung muss nur für deutsche IBANs funktionieren.

Beispiel IBAN: DE 01 5932855 1876587925

Eine IBAN setzt sich aus folgenden Teilen zusammen:

- Landesvorwahl: **DE**
- Prüfziffern: **01**
- Bankleitzahl: **5932855**
- Kontonummer: **1876587925**

Ihre Implementierung sollte möglichst wenig redundante Daten speichern.

*Anmerkung: mit "hamburger".substring(4, 8) erhält man "urge".*