



# Übungsblatt 5

Software Engineering (SoSe 2018)

Abgabe: Fr. 30.06.2018, 12:00 Uhr — Besprechung: Montag, 02.07.2018

- Bitte lösen Sie die Übungsaufgabe in **Gruppen von 4 Studenten** und wählen **EINEN** Studenten aus, welcher die Lösung im ILIAS als **PDF** (Ordner **Abgaben/Blatt 5/**) als **Gruppenabgabe** (unter Angabe aller Gruppenmitglieder) einstellt.
- Für schriftliche Aufgaben erstellen Sie **EINE** PDF-Datei. Die Projekte zu den Programmieraufgaben können Sie als **ZIP**-Datei hinzufügen. Bitte erstellen Sie ein **Titelblatt**, welches die Namen der Studenten, die Matrikelnummern und die E-Mail-Adressen enthält. Im Quellcode fügen Sie diese Informationen bitte als Kommentar hinzu.
- Benennen Sie die Dateien nach dem folgenden Schema:  
**SE[Blattnummer]-[Nachnamen der Teammitglieder].[pdf oder zip]**.

## Aufgabe 1 Maße und Skalentypen

Betrachten Sie die folgenden Maße und geben Sie an, auf welchen Skalentypen (Absolutskala, Nominalskala, Ordinalskala, Intervallskala, Rationalskala) diese gültig sind. Begründen Sie jeweils ausführlich Ihre Wahl.

- Geschwindigkeit von Schiffen in Knoten
- Einwohnerzahl einer Stadt
- Position eines Hauses in der Straße repräsentiert durch die Hausnummer
- Komplexität eines Programms als Anzahl Codezeilen
- Qualität einer App im Google Play Store gemäs Kundenrezensionen (1–5)
- Unit-test coverage als Qualität eine Software

## Aufgabe 2 Tarantula

In dieser Aufgabe sollen Sie eine Möglichkeit zum Auffinden von Fehlern in Programm-Code von Hand durchführen. Eines der in der Vorlesung vorgestellten Verfahren ist Tarantula. Die für das Verfahren notwendigen Daten können Sie den folgenden Tabellen entnehmen.

Für den Programm-Code wurden insgesamt fünf Testfälle mit den folgenden Parameter-Eingaben durchgeführt und eine abschließende Bewertung des Resultats erstellt:

Test	String text	int number	boolean useBreaks	Soll-Ergebnis = Ist-Ergebnis
Test1	"test"	-2	false	Nein
Test2	"test"	0	false	Ja
Test3	"123"	0	true	Ja
Test4	"test"	2	false	Nein
Test5	"test"	2	true	Ja

Die Testfälle erreichen unterschiedliche Code-Abdeckungen. Jede abgedeckte Zeile ist mit einem X markiert. Code-Zeilen mit schließenden Klammern wurden mit vorigen Zeilen vereint, um die Zahl der zu betrachtenden Code-Zeilen zu reduzieren:

Nr..	Code	Test1	Test2	Test3	Test4	Test5
1	StringBuffer buffer = new StringBuffer();	X	X	X	X	X
2	if(number < 0){	X	X	X	X	X
3	throw new IllegalStateException(); }	X				
4	else if(number == 0){		X	X	X	X
5	return ""; }		X	X		
6	while(number>0){				X	X
7	buffer.append(text);				X	X
8	if(useBreaks){				X	X
9	buffer.append("\n"); }					X
10	else{				X	X
11	buffer.append(" "); }				X	
12	number--; }				X	X
13	if(useBreaks){				X	X
14	buffer.delete(buffer.length()-1, buffer.length()); }					X
15	else{				X	X
16	buffer.delete(buffer.length()-2, buffer.length()); }				X	
17	return buffer.toString();				X	X

- (a) Berechnen Sie für jede Code-Zeile den Verdächtigkeits-Wert nach dem Tarantula-Verfahren. Geben Sie sowohl den Ergebnis-Wert, als auch den Rechenweg an!
- (b) In welchen Code-Zeilen liegt mit hoher Wahrscheinlichkeit der Grund für die fehlgeschlagenen Testfälle?

### Aufgabe 3 Konfigurationsmanagement anhand von Kubernetes

In vielen Umfragen wird Kubernetes als eines der Top-5-Open-Source-Projekte genannt. Ihre Aufgabe ist es, das GitHub-Repository zu erforschen (<https://github.com/kubernetes/kubernetes>) und die folgenden Fragen bezüglich des Konfigurationsmanagements zu beantworten:

- Geben Sie eine Zusammenfassung des Projekts in Bezug auf was es tut, was es ist und wie groß es ist in verschiedene Metriken (z.B. LOC, d. h. Lines of Code)
- Ermitteln Sie den Prozess, der beschreibt, wie Entwickler zum Projekt beitragen können.
- Stellen Sie sich vor, Sie hätten eine der Kubernetes-Komponenten repariert. Wie sieht der Prozess von dem Moment an aus, in dem Sie Ihre Änderungen committen möchten, bis Ihre Änderung in einer Version enthalten ist?
  - Zeichnen Sie ein Diagramm, das den Prozess visualisiert.
  - Markieren Sie alle automatisierten und manuellen Schritte in Ihrem Diagramm.
  - Was ist ein Pull-Request?
  - Wo finden sich Schritte zur Qualitätssicherung?
  - Welches Versionierungsschema für Releases wird verwendet?

### Hinweise zu den Übungen:

- Durch die Teilnahme am Übungsbetrieb können Sie sich bis zu **3 Bonuspunkte für die Klausur** verdienen.
- Bedingungen:
  - Während des Semesters darf maximal eine Abgabe im ILIAS ausgelassen werden.
  - Während des Semesters darf pro Person max. eine Hörsaalübung ausgelassen werden.
  - Jede Gruppe muss im Laufe des Semesters eine Aufgabe in der Hörsaalübung präsentieren.
  - Jede Präsentation wird mit folgender Skala bewertet:
    - \* 3 Punkte: Korrekt
    - \* 2 Punkt: Sinnvoll aber fehlerhaft
    - \* 1 Punkte: Sinnlos/falsch oder fehlt (Basispunkt für Präsentation)

**Viel Erfolg!**