

Übungen zu Software Engineering

Wintersemester 2022/23

Blatt 1

Übungsbetrieb

Dienstags nach der Vorlesung wird ein Aufgabenblatt hochgeladen, das bis einschließlich Donnerstag (letztes Testat um 11:30 Uhr) der darauffolgenden Woche zu bearbeiten ist. Die Aufgabenblätter finden Sie auf der Veranstaltungswebseite in Stud.IP unter dem Reiter Vips.

Übungen finden Freitags von 10:15 - 11:45 Uhr (Ort: 93/E33) und von 12:15 - 13:45 Uhr (Ort: 93/E06) mit dem gleichen Inhalt statt. In den Übungen werden das neue Aufgabenblatt besprochen und die Lösungen zum alten Aufgabenblatt diskutiert.

Testatbetrieb

Begleitend zur Veranstaltung finden wöchentliche, 30-minütige Testate bei den Tutoren der Veranstaltung statt. Das aktuelle Aufgabenblatt ist bis zum Testattermin zu bearbeiten und dem jeweiligen Tutor, in Präsenz (Ort: 93/E11) zum entsprechenden Zeitpunkt zu erklären.

Achten Sie bei der Bearbeitung der Aufgaben darauf, diese *vor* dem Testat einmal auf einem Uni-Rechner getestet zu haben, beispielsweise in Raum 93/E11, der zum Bearbeiten der Aufgaben zur Verfügung steht.

Die Testate erfolgen in Zweierteams, zu denen Sie sich von Mittwoch, 26.10. 10:00 Uhr, bis Freitag, 28.10. 14:00 Uhr anmelden können. Die Anmeldung erfolgt in Stud.IP unter dem Reiter *Terminvergabe*. Zwei Studierende in der gleichen Testatgruppe bilden ein Zweierteam. Der angegebene Tutor einer Testatgruppe kann aufgrund der Auslastverteilung für die Tutoren jedoch noch wechseln. Der von Ihnen gewählte Testattermin bleibt in der Regel über das gesamte Semester bestehen, wobei Änderungen in besonderen Fällen vorbehalten bleiben.

Bearbeitung der Aufgaben

Laden Sie die Abgabe Ihrer Lösungen in Stud.IP unter dem Reiter *Vips* unter dem jeweiligen Wochenblatt. Beachten Sie dabei, dass das pro Aufgabenblatt nur eine Datei hochgeladen werden soll (vorzugsweise eine zip Datei). Der Name der Datei soll folgendes Schema haben: <RZ-Kürzel-Partner-1>-<RZ-Kürzel-Partner-2>. Ein Neu-Upload ersetzt die vorherige Datei. Zur Bewertung wird immer die zuletzt hochgeladene Datei während des Testats genutzt. Achten Sie darauf, dass beide Testatpartner die Abgabe vorzeigen können, falls einer der Testatpartner nicht teilnehmen kann. Sie können den Upload ihres Testatpartners nicht einsehen.

Lösen Sie sämtliche Aufgaben immer schriftlich. Quellcode oder Speicherstände der verschiedenen Entwicklungswerkzeuge können Sie auf das Testatverwaltungstool vor dem Testat hochladen. Zeichnungen sollen digital im .PNG, .JPEG oder .PDF Format im Testatverwaltungstool hochgeladen werden.

Dabei können Sie die Zeichnungen entweder direkt digital erstellen oder diese erst handschriftlich erstellen und ein (gut lesbares!) Foto Ihrer Zeichnung hochladen.

Alle Programmieraufgaben sind in der Programmiersprache *Java* abzugeben. Nutzen Sie dabei Java Version 17. Kommentieren Sie Ihren gesamten Quellcode javadoc-konform und achten Sie auf geeignete Klassen-, Methoden- und Variablennamen. Bei der Formatierung des Quellcodes können Sie sich nach den Java-Code-Conventions von Oracle (Sun) richten (<https://www.oracle.com/technetwork/java/codeconvtoc-136057.html>). Machen Sie außerdem von der Java API Gebrauch.

Fragen, Antworten und Ankündigungen

Um Fragen oder Probleme untereinander und mit den Tutoren diskutieren zu können, steht auf der Veranstaltungsseite in Stud.IP ein Forum zur Verfügung.

ECTS-Vergabe

Um die Zulassung zur Klausur zu erhalten, müssen alle bis auf eins der ausgegebenen Übungsblätter erfolgreich bearbeitet (mindestens 50% der Punkte) und dem Tutor präsentiert werden. Zum Abschluss der Veranstaltung entscheidet eine Klausur über die Scheinvergabe.

Krankheitsfälle

Sollten Sie aus gesundheitlichen Gründen nicht an einem Testat teilnehmen können, müssen Sie bis spätestens zum nächsten Testat Ihrem Tutor ein ärztliches Attest einscannen und per E-Mail zukommen lassen. Sollte solch ein Attest ausbleiben, wird Ihr fehlendes Erscheinen als 0 Punkte für dieses Testat gewertet. Ab dem Dritten verpassten Testat gelten für ein Attest dieselben Anforderungen wie für einen Prüfungsrücktritt. Die Regelungen hierzu können Sie unter folgendem Link einsehen: https://www.mathinf.uni-osnabrueck.de/fachbereich/pruefungsamt/hinweise_zum_pruefungsruecktritt.html

Beachten Sie, dass ihr Testatpartner auch bei einer Krankschreibung von Ihnen wie gewöhnlich am Testat teilnehmen muss. Er oder sie benötigt weiterhin mindestens 50% der Punkte auf dem Aufgabenblatt zum Bestehen des Testats.

Alte Klausurzulassungen

Falls Sie bereits in einem vorherigen Semester die Klausurzulassung erworben haben, so bleibt diese grundsätzlich bestehen. Sie müssen also nicht noch einmal am Testatbetrieb teilnehmen. Sollten Sie freiwillig noch einmal am Testatbetrieb teilnehmen wollen, so erlischt Ihre alte Klausurzulassung ab dem zweiten Testat.

Aufgabe 1.1: Grundlagen: Software-Krise und Software Engineering (20 Punkte)

Bearbeiten Sie folgende Teilaufgaben:

1. Geben Sie einen kurzen Überblick über die Ursachen, die zur Software-Krise geführt haben.
2. Erläutern Sie die Bestrebungen, die als Gegenmaßnahmen gegen die Auswirkungen der Software-Krise unternommen werden.

Aufgabe 1.2: Aktuelle Beispiele für das Scheitern von Software-Projekten (20 Punkte)

In der Vorlesung haben Sie mehrere Beispiele für große Software-Projekte kennengelernt, die entweder ganz gescheitert sind oder in mindestens einem gravierenden Punkt ihre Anforderungen nicht erfüllt haben. Recherchieren Sie ein weiteres Beispiel für solche Software-Projekte und erklären Sie Ihrem Tutor folgende Teilaufgaben:

1. Um was für ein Projekt handelt es sich?
2. Wann ist das Projekt gestartet, wann ist es fertiggestellt oder aufgegeben worden? Alternativ: Seit wann wurde das System im Laufenden Betrieb eingesetzt und wie lange dauerte es, bis der Anforderungsmangel erkannt wurde?
3. Was genau ist schief gegangen in dem Projekt (Anforderungen nicht erfüllt, Kostenrahmen überschritten, etc.)?
4. Was sind die Ursachen für das Scheitern des Projekts?
5. Welche Schritte hätten unternommen werden können um das Scheitern zu verhindern? Sofern für Ihr gewähltes Beispiel keine Experteneinschätzungen vorliegen, überlegen Sie sich selber Möglichkeiten.

Die Aufgaben sollen zudem schriftlich in Fließtext oder Stichpunkten abgegeben werden. Berücksichtigen Sie bei Ihrer Recherche nur Projekte, deren Scheitern ab dem Jahr 2012 publik geworden ist. Eine Möglichkeit ist es z. B. die Problematik rund um die Boeing 737 MAX zu betrachten.

Aufgabe 1.3: Objektorientierter Entwurf (60 Punkte)

Das Zahlenrätsel Sudoku besteht aus einem Spielfeld mit 9x9 Feldern, die teilweise mit je einer Ziffer zwischen 1 und 9 gefüllt sind. Aufgabe des Spielers ist es, die übrigen leeren Felder mit Ziffern zwischen 1 und 9 zu füllen, so dass in jeder Zeile, jeder Spalte und jedem Unterfeld alle Ziffern zwischen 1 und 9 genau einmal vorkommen. Ein Unterfeld besteht aus 3x3 Feldern. Es gibt 9 disjunkte Unterfelder in einem Sudoku.

Entwerfen Sie die Geschäftslogik des Sudoku-Spiels als UML-Klassendiagramm, so dass sie in Java implementiert werden kann. Es sollen keine primitiven Datentypen oder Arrays genutzt werden, d.h. auch die Sudoku-Ziffern sollen als Klasse repräsentiert werden. Gehen Sie dafür davon aus, dass mehrwertige Beziehungen mit Listen umgesetzt werden. Modellieren Sie nicht nur das Spielfeld, seine Zeilen, Spalten und Unterfelder, sondern auch den Spieler und die Züge des Spielers in einem Spiel. Jeder Spielzug sollte eine Referenz auf den vorherigen Spielzug besitzen, so dass Züge rückgängig gemacht werden können. In einem Spielzug soll zudem die Zahl, die gesetzt wurde, sowie das Spielfeld referenziert sein.

Für die Modellierung können Sie ein Programm Ihrer Wahl verwenden. Beispiele hierfür sind DIA (<http://dia-installer.de/>) oder auch bestimmte Webseiten, wie z. B. <https://app.diagrams.net/>. Auf die Angabe von Methoden und Sichtbarkeiten können Sie im gesamten Diagramm verzichten. Geben Sie aber bei jeder Beziehung Kardinalitäten an. Machen Sie das Diagramm außerdem navigierbar.