

Übung 2: Projektplanung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Engineering
WS 2018/19 - Dr. Michael Eichberg

Abgabe

Die Übung wird als *sbt*-Projekt (<http://www.scala-sbt.org>) bereitgestellt. *sbt* kann verwendet werden, um Java (und Scala) Projekte einfach auszuführen und zu testen. Installieren Sie *sbt* auf Ihrem Rechner und stellen Sie sicher, dass ein Java SDK (min. Java 8) installiert ist. Überprüfen Sie, dass der Java Compiler *javac* auf der Kommandozeile ausführbar ist, auch wenn Sie sich nicht im Verzeichnis mit der ausführbaren Datei befinden. Wenn nicht, passen Sie Ihren *PATH* entsprechend an, Hinweise wie dies bei Ihrem Betriebssystem möglich ist, finden Sie im Internet.

IntelliJ erlaubt das Importieren von *sbt*-Projekten direkt, dafür muss allerdings das Scala Plugin installiert sein. Um ein Eclipse-Projekt zu erzeugen, kann *sbt eclipse* im Projektverzeichnis (das Verzeichnis, das die Datei *build.sbt* enthält) ausgeführt werden, danach kann das Projekt mit Datei > Importieren > Vorhandene Projekte in Arbeitsbereich importiert werden. Wenn Sie eine *main*-Methode geschrieben haben, können Sie Ihr Programm mit *sbt run* ausführen, Tests können Sie unter *src/test/java/* anlegen und mit *sbt test* ausführen.

Das Ausführen des Kommandos *sbt* im Projektverzeichnis startet den interaktiven Modus von *sbt*. Im interaktiven Modus können Kommandos ausgeführt werden, ohne jedes Mal *sbt* eingeben zu müssen.

Um Ihre Lösung abzugeben, melden Sie sich zunächst unter

<https://submission.st.informatik.tu-darmstadt.de/course/se18>

an und erzeugen Sie ein *submission token*. Wenn Sie den interaktiven Modus von *sbt* verwenden, führen Sie dann folgendes Kommando aus:

```
submit <ihreTUID> <submissionToken>
```

wobei *<ihreTUID>* Ihre eindeutige Identifikationsnummer an der TU Darmstadt (**nicht Ihre Matrikelnummer!**) und *<submissionToken>* das zuvor generierte Token ist. Geben Sie die spitzen Klammern nicht mit an, der Befehl sollte etwa so aussehen: *submit ab01cdef 01234567*. Wenn Sie nicht den interaktiven Modus verwenden, muss das Kommando in Anführungszeichen gesetzt werden, also *sbt "submit <ihreTUID> <submissionToken>"*. Sie können (innerhalb der Abgabefrist) beliebig oft eine Lösung einreichen, allerdings wird nur die zuletzt eingereichte Lösung bewertet. Die letzte Lösung, die ein Gruppenmitglied eingereicht hat, wird zur Bewertung für die ganze Gruppe herangezogen. Koordinieren Sie sich daher in Ihrer Gruppe, wer Ihre gemeinsame Lösung einreicht.

Stellen Sie sicher, dass Sie das zur Verfügung gestellte Template nutzen und die Namen und Signaturen der vorgegeben Klassen und Methoden nicht verändern sowie dass von Ihnen hinzugefügte Klassen und Methoden die geforderten Namen und Signaturen verwenden. Ändern Sie außerdem nichts an der vorgegebenen Datei *build.sbt*. Andernfalls wird das System Ihre Lösung nicht bewerten können. Beachten Sie, dass der Zugriff auf die Abgabepattform nur im **internen Netz der Universität** möglich ist. Für einen Zugriff von außerhalb benötigen Sie daher eine VPN-Verbindung.

Einführung

In dieser Übung werden Sie verschiedene Methoden anwenden, die der Projektplanung dienen. Bearbeiten Sie die folgenden fünf Aufgaben und erstellen Sie eine einzelne PDF-Datei mit Ihren Antworten. Legen Sie diese Datei in den Ordner *solution* in dem Template, das Ihnen zur Verfügung gestellt wurde, und verwenden Sie zur Abgabe wie oben angegeben *sbt submit*.

Für die Planung eines Softwareprojekts sind folgende Task (mit geschätzter Zeit) vorgesehen. Erstellen Sie entweder ein Activity Network oder ein Gantt-Diagramm, das die Projektplanung darstellt und bestimmen sie den kritischen Pfad.

Task	abhängig von	geschätzte Zeit	Milestone
Task 1	-	7 Tage	M1
Task 2	-	15 Tage	M2
Task 3	M1	6 Tage	M2
Task 4	M1	9 Tage	M2
Task 5	M2	3 Tage	Release
Task 6	M1	8 Tage	Release

Im Folgenden finden Sie zwei User Stories. Verwenden Sie in Ihrer Gruppe das in der Vorlesung vorgestellte "Planning Poker" um Story Points zu vergeben. Notieren Sie stichpunktartig, welche Story Points in welcher Runde vergeben wurden und wie diese begründet wurden.

User Story 1:

Name	Ausgabe in Präfixnotation
Beschreibung	Es steht eine Methode zur Verfügung, mit der bereits eingelesene boolesche Ausdrücke als Strings in Präfixnotation ausgegeben werden können.
Akzeptanzkriterium	Boolesche Ausdrücke können korrekt in Präfixnotation ausgegeben werden. Zwischen den Variablen und Operatoren steht jeweils ein Leerzeichen, kein Leerzeichen steht am Anfang und Ende des Strings.

User Story 2:

Name	Ausgabe in Infixnotation
Beschreibung	Es steht eine Methode zur Verfügung, mit der bereits eingelesene boolesche Ausdrücke als Strings in Infixnotation ausgegeben werden können.
Akzeptanzkriterium	Boolesche Ausdrücke können korrekt in Infixnotation mit optimaler Klammerung ausgegeben werden. Der innere Ausdruck einer Negation steht nur dann in Klammern, wenn sich um ein And oder Or handelt, ein Teilausdruck eines And steht nur dann in Klammern, wenn sich um ein Or handelt, Teilausdrücke eines Or stehen nie in Klammern. Um die Operatoren And und Or stehen Leerzeichen, nach Not steht kein Leerzeichen, ebenfalls nicht am Anfang und Ende des Strings, nach öffnenden oder vor schließenden Klammern.

- a) Implementieren Sie die oben angegebene User Story 1 (benutzen Sie das zur Verfügung gestellte Template, die Implementierung wird bewertet!) und notieren Sie die dafür benötigte Zeit. Berechnen Sie nun die Velocity:

$$Velocity = \frac{\text{geschätzte Story Points}}{\text{benötigte Zeit}}$$

Geben Sie die berechnete Velocity an sowie die Zeit, die Sie gemäß dieser Velocity für User Story 2 benötigen werden. Scheint Ihnen diese Zeit realistisch? Wenn nicht, geben Sie eine neue Schätzung für die Story Points für User Story 2 an und begründen Sie die Anpassung kurz.

- b) Implementieren Sie nun User Story 2 (im Template, auch diese Implementierung wird bewertet!) und bestimmen Sie erneut Ihre Velocity. Geben Sie die neue Velocity an. War ihre Schätzung realistisch?

Die folgende Tabelle listet die User Stories eines Softwareprojekts mit einer Iterationsdauer von 14 Tagen:

User Story	Story Points	Geplant für Iteration	Tatsächliche Zeit	Kundenpriorität
US 5	7	2	3 Tage	4
US 6	12	1	Nicht abgeschlossen	
US 8	4	2	2 Tage	
US 10	6	2	3 Tage	
US 11	7	2	Nicht abgeschlossen	5
US 14	6	2	4 Tage	
US 15	2	2	1 Tag	
US 16	3			
US 17	9			6
US 18	2			1
US 19	14			3
US 20	6			7
US 21	7			8
				2

- Berechnen Sie die Velocity der zweiten Iteration.
Bestimmen Sie anhand dessen die maximal realisierbaren Story Points für die kommende, dritte Iteration.
Geben Sie jeweils den vollständigen Rechenweg an.
- Wählen Sie unter Berücksichtigung der Kundenpriorität die User Stories für die kommende, dritte Iteration aus.
Begründen Sie ihre Auswahl.

Der folgende Text ist die Beschreibung eines Features für ein Kartenprogramm durch einen Kunden. Erstellen Sie daraus eine User Story.

„Nutzer sollen mit unserem Kartenprogramm auch Routen planen können. Wenn der Nutzer auf einen Punkt der Karte doppelt klickt, soll eine grüne Nadel an dieser Stelle erscheinen und der Cursor die Form einer roten Nadel annehmen. Wenn der Nutzer dann noch einmal klickt, wird die rote Nadel platziert und der eingebaute Routingalgorithmus mit den markierten Orten als Start- und Endpunkt gestartet. Die berechnete Route soll angezeigt werden. Außerdem muss die Länge der Route dargestellt werden. Kann keine Route gefunden werden, muss eine entsprechende Fehlermeldung angezeigt werden.“