

## Software Engineering

# Übung 2: Projektplanung

## Lösung

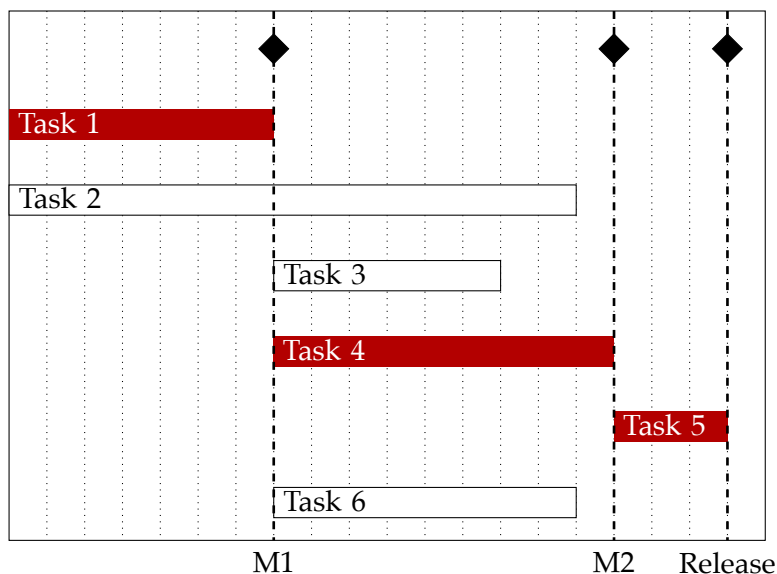
Patrick Elsen

Wintersemester 2018-2019

Technische Universität Darmstadt

### Problem 1: Activity Network / Gantt-Diagramm

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.



### Problem 2: Planning Poker

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.

Der Ausgabe in Präfixnotation haben wir eine Story Point gegeben. Dies sollte eine sehr leichte Implementation sein. Der Ausgabe als Infixnotation haben wir drei Story Points gegeben, weil wir

uns unsicher sind, wie viel Aufwand dafür benötigt wird, und weil wir mit Operatorpräzedenzen hantieren müssen.

### Problem 3: Velocity

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.

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

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.

Ich habe ca. 10 Minuten gebraucht, um die Funktionalität zu implementieren. Ich konnte alles größtenteils von der Implementation von `toPostfixString()` kopieren. Ich war sehr schockiert, dass uns keine Tests für die vorgegebenen Methoden geliefert wurden, also habe ich meine Tests von der vorherigen Abgabe kopiert. Dabei habe ich gemerkt, dass ich eine andere Operatorenreihenfolge in meiner Abgabe benutzt hatte als in dieser, also musste ich einige Tests anpassen. Ich denke, dass ich für die Funktionalität von User Story 2 etwas mehr Zeit brauche, da ich nicht einfach Code kopieren kann, sondern eigene Logik implementieren muss.

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?

### Problem 4: Iterationsplanung

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.

### Problem 5: User Story

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.“