



Übungsblatt 1

Software Engineering (für ST) (SoSe 2019)

Abgabe: Fr. 26.04.2019, 12:00 Uhr — Besprechung: Dienstag, 30.04.2019

- 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 1/) 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 Kritischer Pfad

In dieser Aufgabe sollen Sie ein Programm schreiben, das in der Lage ist, den kritischen Pfad in einem Netzplan (PERT-Chart) zu finden. Im ILIAS finden Sie bereits ein Java-Projekt, das die grundlegende Struktur des Programms vorgibt. Dieses Projekt müssen Sie nur noch ergänzen. Das Projekt enthält auch das Beispiel aus den Vorlesungsfolien als JUnit-Test. An diesem Test können Sie sehen, wie ein Netzplan initialisiert werden sollte. Beachten Sie, dass Ihre Implementierung auf beliebige Netzpläne anwendbar sein muss. Es genügt ausdrücklich nicht, nur diesen JUnit-Test zu erfüllen! Sie können den Test ergänzen oder weitere Tests hinzufügen.

Mit Ihrer Implementierung sollen Sie ein konkretes Problem lösen. Die folgende Tabelle enthält die Arbeitspakete (Aktivitäten) sowie deren geschätzte Aufwände und Abhängigkeiten.

Arbeitspaket (ID)	Aufwand (Tage)	Abhängigkeiten
A01	3	—
A02	4	—
A03	5	—
A04	4	A01
A05	9	A01
A06	4	A02
A07	2	A03
A08	4	A04
A09	2	A06
A10	3	A06
A11	3	A07
A12	2	A05, A08
A13	4	A10, A11
A14	7	A09, A12
A15	2	A09, A12
A16	4	A13, A15

Tabelle 1: Arbeitspakete eines Netzplans.

Hinweis: Das Lösen des Problems ohne Implementierung wird mit 0 Punkten bewertet. Die Lösung muss mit der Implementierung reproduzierbar sein.

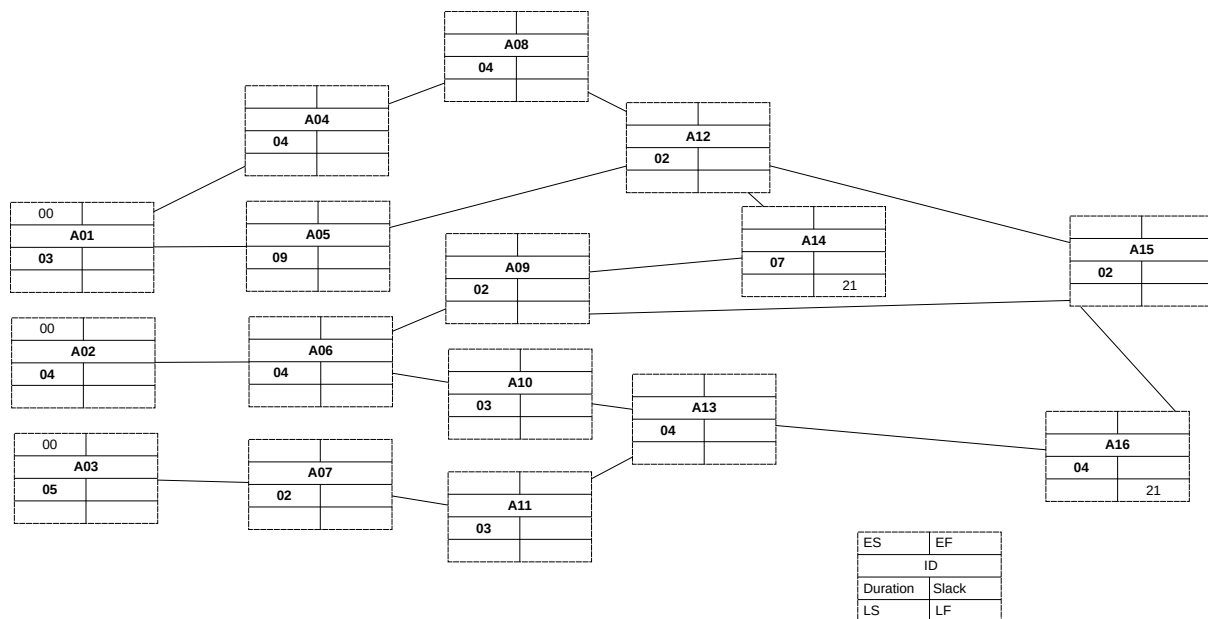


Abbildung 1: Pert-Chart des konkreten Problems, das mit der Implementierung gelöst werden soll.

- Um Ihre Implementierung besser überprüfen zu können, sollten Sie den Netzplan zuerst von Hand lösen. Sie können hierzu die Lücken in Abbildung 1 ausfüllen.
- Implementieren Sie nun die Methode *computeCriticalPath* in der Klasse *Project*. Versuchen Sie, private Methoden zu verwenden, um Ihren Code zu strukturieren. Verwenden Sie sinnvolle Bezeichner für Methoden und Variablen.
- Testen Sie Ihre Implementierung zuerst mit dem beigefügten JUnit-Test *LectureExampleTest*. Implementieren Sie außerdem das konkrete Problem (als JUnit-Test) und vergleichen Sie darin Ihre Lösung von Hand mit den Ergebnissen der Implementierung.

Aufgabe 2 Ressourcenmanagement

In dieser Aufgabe sollen Sie nun als Teamleiter einen Zeitplan für Ihr Entwicklerteam erstellen. Das zu bearbeitende Softwareprojekt setzt sich erneut aus den in Tabelle 1 dargestellten Arbeitspaketen zusammen. Beachten Sie auch dieses mal wieder die Abhängigkeiten der verschiedenen Arbeitspakete. Der Aufwand ist in Arbeitstagen für einen Entwickler angegeben.

Eine Studie Ihrer Firma hat ergeben, dass nicht zu viele Entwickler an einem Arbeitspaket arbeiten sollten, da ansonsten die Produktivität sinkt. Nach einer firmeninternen Richtlinie sollen daher über den gesamten Bearbeitungszeitraum für ein Arbeitspakets niemals mehr als zwei verschiedene Personen beteiligt sein.

Insgesamt stehen ihnen vier Mitarbeiter (inklusive Ihnen selbst) für die Entwicklung zur Verfügung. Allerdings hat ein Entwickler bereits fünf Tage Urlaub angemeldet. Er kann daher nicht an den Projekttagen 11–15 eingeplant werden.

- Entscheiden Sie, ob in diesem Fall eine kapazitätstreue oder eine termintreue Einsatzplanung notwendig ist. Begründen Sie ihre Entscheidung.
- Bestimmen Sie die minimale Anzahl an Tagen die theoretisch nötig ist, um alle Arbeitspakete abzuarbeiten. Erstellen Sie außerdem einen Auslastungsplan für die Mitarbeiter, der genau diese minimale Anzahl erfüllt. Dieser Plan soll sowohl die Abhängigkeiten der Arbeitspakete, als auch die firmeninterne Richtlinie berücksichtigen. Arbeitspakete müssen außerdem zusammenhängend abgearbeitet werden.
- Ihr Chef ist mit dem Abschlussdatum für das Projekt nicht einverstanden. Er möchte, dass das Projekt innerhalb von 10 Tagen abgeschlossen wird, also bevor Ihr Mitarbeiter Urlaub hat. Dafür ist Ihr Chef bereit, weitere Mitarbeiter dem Projekt zuzuteilen. Bestimmen Sie eine untere

Grenze für die minimale Anzahl an zusätzlichen Mitarbeitern, die für das Erreichen dieses Ziels zugeteilt werden müssen.

- (d) Ist der Plan des Chefs unter Einhaltung der Richtlinie der Firma durchführbar? Begründen Sie ihre Antwort. Für diese Teilaufgabe sollte besonders Abbildung 1 betrachtet werden.

Aufgabe 3 Gantt-Diagramm, Termin-Drift-Diagramm

Kurz vor dem Start Ihres Projekts wechselt die Führung in Ihrer Firma. Besonders qualifizierte Mitarbeiter sollen nun kleinere Projekte alleine bearbeiten. Sie sind einer davon. Das Projekt aus Tabelle 1 soll folglich neu geplant werden. Gleichzeitig regt Ihr neuer Chef eine Evaluierung der bisherigen Projekte an. Dafür sollen Sie ein altes Projekt eines Kollegen untersuchen und Verbesserungsvorschläge machen.

- (a) Erstellen Sie einen Terminplan für Ihr Projekt in Form eines Gantt-Charts für die Arbeitspakete aus Tabelle 1 bzw. Abbildung 1. Planen Sie hierbei Arbeitspakete so früh wie möglich ein. Das Gantt-Diagramm muss auch Abhängigkeiten zwischen den Arbeitspaketen darstellen. Außerdem müssen Verzögerungen ohne Einfluss auf das Ende des Projekts ebenfalls dargestellt werden!

Diese Aufgabe ist abhängig von Ergebnissen aus Aufgabe 1! Sie müssen das Problem des kritischen Pfads zumindest von Hand gelöst haben. Falls Sie diese Aufgabe nicht erfolgreich lösen konnten, so können Sie erfundene Werte für Startzeitpunkte und Verzögerungen verwenden, so weit diese nicht den gegebenen Abhängigkeiten widersprechen. In diesem Fall tragen Sie die erfundenen Werte auf jeden Fall in Abbildung 1 ein und geben Sie diese mit ab!

- (b) Für die Analyse des Projekts wird Ihnen u. a. ein Termin-Drift-Diagramm (Abbildung 2) zur Verfügung gestellt. Darin befinden sich fünf **Meilensteine**, die jeweils das Ende einer **Phase** der Entwicklung kennzeichnen: Spezifikation (S), Entwurf (E), Implementierung (I), Tests (T) und Abgabe (A). Die Projektphasen wurden strikt nach dem Wasserfall-Modell durchgeführt, das Projekt dauerte insgesamt 2 Monate und wurde ausschließlich von Ihrem Kollegen bearbeitet. Beantworten Sie dazu die folgenden Fragen:

- Wurde das Projekt rechtzeitig abgeschlossen?
- Welche **Meilensteine** wurden pünktlich/früher/später erreicht?
- Wie oft wurden **Meilensteine** jeweils verlegt?
- Welche **Phasen** waren länger/kürzer als geplant?
- Muss das Schätzverfahren verbessert werden?
- Stellen Sie mindestens zwei konkrete Vermutungen auf, was im Projekt nicht gut funktioniert hat. Betrachten Sie dazu die stärksten Abweichungen. Begründen Sie diese Vermutungen anhand des Diagramms!

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.
 - 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!

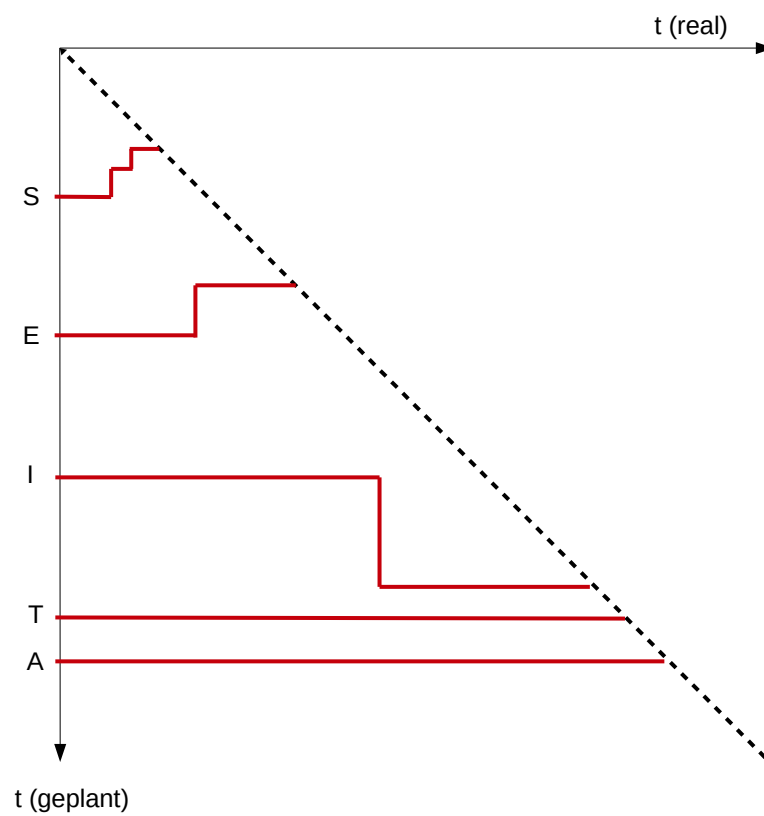


Abbildung 2: Das Termin-Drift-Diagramm zum abgeschlossenen Projekt Ihres Kollegen.