# Software Engineering I; Projektarbeit Einführung

## Art der Veranstaltung

Software Engineering I im 4. Semester ist keine Vorlesung, sondern ein Software Praktikum, das in Form eines Projektes gestaltet ist.

## Zeitlicher Aufwand der Veranstaltung

Software Engineering ist im Vorlesungsplan verzeichnet mit 4 Stunden/Woche. Dies ist jedoch nur die Zeit, in der der betreuende Dozent dem Projektteam für Betreuung und Durchführung von Projekt-Reviews zur Verfügung steht.

Es wird von jedem beteiligten Studenten erwartet (und das Projektthema/-Volumen ist entsprechend so ausgelegt), dass er *im Mittel* einen reellen Zeitaufwand von etwa **10 Stunden/Woche** (Bis Ende des Projektes etwa 100 Stunden) in das Software-Praktikum aufwenden muss.

## Ziele der Veranstaltung

Prof. Jochen Ludewig (Univ. Stuttgart) charakterisiert in einem Aufsatz "Ziele der Softwaretechnik" [1] das Software Engineering als eine Technik, deren Ziel es ist, die ungeheure Komplexität des Prozesses der Erstellung, des Einsatzes und der Weiterentwicklung von Softwaresystemen durch Strukturierung und Reduktion beherrschbar zu machen.

Softwaretechnik ist in diesem Verständnis eine Ingenieurdisziplin. Wie in allen Ingenieurdisziplinen haben sich auch hier bestimmte organisatorische Vorkehrungen nach allen (empirisch belegten) Erfahrungen als sinnvoll erwiesen. Die Nichtanwendung dieses "State of the Art" ist ein handwerklicher Fehler und führt nicht nur langfristig zu Wettbewerbsnachteilen, sondern kann auch rechtliche oder finanzielle Konsequenzen haben.

Mit den verschiedenen Komponenten des Ausbildungszweigs "Softwaretechnik", heißt es weiter bei Prof. Ludewig, werden die folgenden vier Ziele verfolgt. Sie liegen auch der Ausrichtung unserer Veranstaltung zu Grunde.

* Studierende sollen erkennen, dass man größere Projekte nur mit einer **soliden Organisation** durchführen kann. Dazu gehören insbesondere auch verschiedene **Rollen**. Die Studierenden sollen den **Respekt** vor diesen Rollen erlernen und in die Lage versetzt werden, diese Rollen auch **selbst zu übernehmen**.
* Die Studierenden sollen die Bedeutung der **Kommunikation** (in allen Richtungen) erkennen und lernen, sich dieser Bedeutung entsprechend zu verhalten. Das schließt ein:
  + die systematische Planung, Vorbereitung, Durchführung und Auswertung von **Gesprächen und Besprechungen**, mit dem Kunden, aber auch innerhalb des Projektteams
  + die ordentliche technische **Dokumentation**,
  + die Ablieferung gut lesbarer **Beschreibungen**,
  + die gründlich vorbereitete **Präsentation** in Vorträgen und Demonstrationen.
* Es sind **Fachkenntnisse** zur Lösung der Probleme zu erwerben und anzuwenden.
* Die Arbeiten müssen **systematisch** und **handwerklich ordentlich** ausgeführt und **termingerecht** abgeschlossen werden. Dazu gehört insbesondere auch eine ständige **Zeitaufschreibung**.

Die bei praktisch allen Menschen gegebene Neigung, sich gegen diese Ziele aufzulehnen, impliziert, worauf Sie bei der Durchführung Ihres Projekts besonders achten sollten.

## Thema des Projektes

Für dieses Jahr ist als Thema ausgewiesen:

**Entwicklung eines Multi-User Dungeon Server   
(MUD-Service)**

***MUD-Service***

Ein Multi User Dungeon (Abkürzung: MUD) Spiel ist ein hauptsächlich textbasiertes Rollenspiel, das auf einem Server realisiert ist und in welchem Benutzer gleichzeitig nach einem Login eine Persona (auch Avatar genannt) annehmen, sich im Spiel bewegen und mit einander und der kreierten Spielwelt interagieren können. Die Spiele sind langlaufend. Spieler können sich nach Bedarf einloggen, beteiligen und wieder ausloggen.

„Als Charakter in einem MUD stehen dem Spieler verschiedene Möglichkeiten zur Interaktion mit anderen Spielern und mit den Objekten im MUD zur Verfügung. So gibt es eine Vielzahl von Kommunikationsmöglichkeiten zwischen den Spielern. In den meisten MUDs gibt es Räume, zwischen denen sich der Spieler bewegen kann. Diese Räume werden textlich beschrieben und können somit die Gestalt von Landschaften, Häusern oder z. B. auch Fahrzeugen annehmen. In den Räumen befinden sich Objekte (z. B. Taschen oder Beutel, Waffen, Rüstungen, Lebensmittel, …) und „Lebewesen“ (andere Spieler und computergenerierte Nicht-Spieler-Charakter/NPC), mit denen der Spieler interagieren kann.“

(<https://de.wikipedia.org/wiki/Multi_User_Dungeon>, Abruf 2.3.2020).

Ursprüngliche Spiele waren rein textbasiert und stellten märchenhafte Umgebungen im Stile J.R.R. Tolkien mit unterirdischen Räumen und magischen Lebewesen (Drachen, Orks, usw.) dar. Sie wurden auf Großrechnern über Terminals (VT100) gespielt. Später, in den frühen Jahren des Internets, wurden sie über einfache Internetverbindungsprotokolle (vor allem Telnet) gespielt. Heutzutage reicht ein Browser um die um etwas Grafik erweiterten Games spielen zu können. Eine Liste populäre MUDs findet sich z.B. unter <http://www.topmudsites.com/>. Es gibt diese Spiele auch als kommerzielle Brettspiele/Pen-and-paper-Games, z.B. das Dungeons and Dragons (offizielle Website: <https://dnd.wizards.com/>, die Basic Rules des Spiels sind downloadbar unter: <https://dnd.wizards.com/articles/features/basicrules>, Abruf 2.3.2020) und Pathfinder (offizielle Website <https://paizo.com/pathfinder>).

Ein MUD-Server muss in der Lage sein, ggf. mehrere MUD-Games zu hosten. Diese Games müssen eingerichtet (i.A. konfiguriert, nicht klassisch programmiert) werden können von einem sogenannten Dungeon Master, der danach dann den Spielverlauf übersehen und ggf. darin eingreifen kann. Zu konfigurieren sind solche Dinge als z.B. Personas (bzgl. u.a. Rasse, Klasse, Persönlichkeit, Equipment), Monster, Räumen, Objekten, und Aktionen (Bewegungen und sonstige). Wichtig ist auch die Realisierung diversen Formen der Kommunikation, zwischen teilnehmenden Spielern unter einander, sowie auch mit dem Dungeon Master.

Es gibt eine Vielzahl von MUD-Servern und MUD-Libraries, die als Vorbild für dieses Softwareprojekt dienen können. Benutzung dieser Ressourcen im Projekt sind allerdings allenfalls erlaubt nach eingängiger Diskussion und Abstimmung mit den Projektbetreuen

## Ablauf des Projektes

Das Projekt orientiert sich in der Methodologie an das Lehrbuch von Balzert [2]. Zusätzlich ist eine Vorprojektphase vorgesehen, in der Sie sich zunächst mit den Grundlagen des verlangten technischen Konzepts vertraut machen können.

Das Projekt gliedert sich grob in die folgenden Phasen:

* In einer **ersten Phase** sind die Anforderungen zu analysieren, zu spezifizieren und in den Dokumenten Lastenheft, Glossar und später im Pflichtenheft zu fixieren.
* In der **zweiten Phase** machen Sie sich im Rahmen eines Vorprojekts mit theoretischen und technischen Konzepten vertraut, die zur Erfüllung der Aufgabenstellung einzusetzen sind.
* In der **dritten Phase** wird die Modellierung ausgeführt, um schließlich
* In der **vierten Phase** das Modell zu implementieren und zu testen.

Wie Sie wissen, geht es hier explizit nicht um eine lineare Anordnung dieser Phasen: Vielmehr gibt es Iterationen, bzw. verläuft das Projekt nach dem Phasenkonzept von RUP (Rational Unified Process) [3] oder auch nach Agilen Vorgehen wie SCRUM [4].

Für Vorprojekt und Projekt sind begleitende **Produkt-Dokumentationen** entsprechend den Vorgaben in den Arbeitsblättern zu erstellen. Wichtige Etappen des Entwicklungsprozesses werden zu vorgegebenen Terminen mit Reviews abgeschlossen. Ein **Review** ist ein formelles Verfahren, in dem Dokumente nach vorherigem individuellem Studium während eines Präsenzmeetings der angegebenen Personen besprochen werden. Mehr zum Ablauf eines Reviews später in einem Arbeitsblatt.

## Werkzeugeinsatz

Im Folgenden wird davon ausgegangen, dass als objektorientiere Sprache Java zum Einsatz kommt. Dies ist nicht ausdrücklich Pflicht. Einsatz von anderen OO Sprachen und damit einhergehend anderen Technologiestacks sind mit Ihren Betreuern abzustimmen. Im Projekt ist eine IDE zu verwenden. Ebenso soll ein entsprechendes Testframework verwendet werden (z.B. bei Java JUnit). Für Analyse und Design des Programmes ist eine UML-Werkzeugumgebung vorzusehen. Alle relevanten Dokumente (Design, Code, sonstige Dokumentation sind in Git auf bitbucket.org oder gitlab.com zu verwalten. Genaueres ist mit den Betreuern am Anfang des Projektes festzulegen.

## Die Webseiten Ihres Projekts

Eine wichtige Komponente des Praktikums besteht darin, die eigene Arbeit zu dokumentieren und aus den Erfolgen und Fehlern anderer zu lernen.

Dies soll (in einer wenig reglementierten Weise) über die **Webseite des Projekts** realisiert werden. Dort sind deshalb die entsprechend den Vorgaben in den Arbeitsblättern zu erstellenden Materialien nach Ablauf des Abgabetermins **zu veröffentlichen**.

Darüber hinaus soll auf den Webseiten im Rahmen einer **Projekt-Dokumentation** der Fortgang der Arbeiten am Projekt dargestellt werden. Hierfür gibt es keine weiteren Vorgaben. Sie können sich z.B. an den Projektwebseiten von Open Source Projekten orientieren. Bedenken Sie, **dass die Webseite das Aushängeschild Ihrer Gruppe ist**, und ggf. als Projektkostprobe nach außerhalb der DHBW veröffentlicht wird.

## In der Gruppe zu besetzende Rollen

Zur arbeitsteiligen Umsetzung des Projekts sind in der Projektgruppe verschiedene Rollen zu besetzen:

* Die/der **Projektleiter/in** koordiniert die Arbeit der Gruppe und ist Ansprechpartner/in für den Kunden.
* Die/der **technische Assistent/in** ist für die Verwaltung der Gruppenressourcen verantwortlich.
* Die/der **Verantwortliche für Recherche** koordiniert die Anforderungsanalyse.
* Die/der **Verantwortliche für Modellierung** koordiniert die Modellierungsphase.
* Die/der **Verantwortliche für Tests** koordiniert die Erstellung eines Testkonzepts und überwacht dessen Umsetzung.
* Die/der **Verantwortliche für Implementierung** koordiniert den Prozess der Implementierung.
* Die/der **Verantwortliche für Qualitätssicherung und Dokumentation** koordiniert die Erstellung eines Dokumentations- und Qualitätssicherungskonzepts, überwacht dessen Umsetzung und die termingerechte Erstellung, Abnahme und Veröffentlichung der erforderlichen Dokumente.

Übernahme einer Verantwortlichkeit bedeutet, einen Aspekt der Projektarbeit so vorzubereiten, zu planen und zu organisieren, dass er zielgerichtet, mit möglichst geringem Aufwand, aber in guter Qualität von der Gruppe bewältigt werden kann. Dazu gehört es, sich mit den theoretischen Grundlagen dieses speziellen Aspekts langfristig vertraut zu machen, das dabei gewonnene Wissen in die Gruppe einzubringen (Schulung anderer, soweit relevant) und (in Zusammenarbeit mit dem Projektleiter) die Beachtung der Absprachen zu überwachen. Bei den Reviews wird bewertet, ob die Verantwortlichen ihre Rollen gerecht geworden sind.

## Rahmenplanung

Die Rahmenplanung entnehmen Sie bitte dem Dokument **02-Rahmenplan.xls.** Hierin ist im Groben einen möglichen zeitlichen Verlauf des Projekts verzeichnet. In dem Dokument finden Sie ebenso eine Auflistung einzelner Aufgabenblätter mit Abgabeterminen, die Sie im Laufe des Projektes zu bearbeiten haben.

**Diese Rahmenplanung stellt keinen Projektplan dar!** Insbesondere gibt es darin keine Angaben zu Humanressourceallokation. Ebenso wenig sind Arbeitspakete sinnvoll ausdefiniert, sondern allenfalls angedeutet. Es ist Aufgabe des Projektteams selbst einen Projektplan aufzustellen und fortzuschreiben. Ebenso ist ein Projektcontrolling unabdingbar.

## Aufwandserfassung

Für die qualifizierte Führung eines Projekts spielt auch die Aufwandsschätzung, -analyse und -erfassung eine wichtige Rolle, um die personellen Projektressourcen effizient und den jeweiligen Fähigkeiten angemessen einzusetzen. Für die Projekt-Arbeit soll deshalb ein Aufwandserfassungssystem eingeführt werden.

* Die auszuführenden Arbeiten werden in Teilaufgaben zerlegt und zugeordnet   
  (Verantwortlich: Projektleiter).
* Jedes Teammitglied führt selbst Buch über die aufgewendete Zeit und bewertet den Aufwand sowie die Schwierigkeit der Teilaufgabe auf einer Skala a bis e (a = viel zu hoch, b = zu hoch, c = angemessen, d = zu niedrig, e = viel zu niedrig). Dazu gehört natürlich auch die aufgewendete Zeit für Projekttreffen und ähnliche Abstimmungs- und Diskussionsrunden.
* Der Projektleiter sammelt diese Informationen wöchentlich ein, prüft sie im Team auf Plausibilität und erstellt den **wöchentlichen** **Aufwandsbericht**, aus dem ersichtlich wird, welche Teammitglieder für welche Teilaufgaben wie viel Zeit verwendet haben und wie die Arbeit eingeschätzt wurde.
* Ebenso soll aus jedem Wochenbericht die Wochengesamtaufwandsumme, sowie die kumulierten geleisteten Gesamtaufwände nach Projektmitgliedern klar hervorgehen. Ausgestaltung und Format dieser Kumulation/Aggregation sind mit dem Betreuer am Anfang des Projektes zu vereinbaren.
* Diese Berichte sind jeweils direkt nach Erstellung dem betreuenden Dozenten zu übermitteln, bzw. in einem geschützten Bereich der Projektwebsite zu veröffentlichen.
* Die Analysen der abzugebenden Aufwandsberichte sind so zu erstellen, dass sich die Berichtszeitintervalle nicht überschneiden.

Das Muster eines Aufwandberichtes für ein Teammitglied finden Sie in **03-Aufwanderfassung.xls**. Es obliegt den Projektleitern eine geeignete Form des Gesamtberichtes für die Betreuer zu erstellen und mit ihnen abzustimmen.

## Statusbericht

Das Projektteam soll am Ende jeder Woche einen kurzen (halbe Seite) **wöchentlichen Statusbericht** abgeben, bzw. in einem geschützten Bereich der Projektwebsite veröffentlichen. Form des Berichtes ist dem Team überlassen. Im Inhalt sollen kurz die in der Aufwanderfassung aufgelisteten Arbeiten erläutert bzw. kommentiert werden. Ebenso sind Abweichungen vom Projektplan zu berichten, zusammen mit evtl. geplante Aktivitäten oder Änderungen in Zusammenhang mit den Abweichungen.

## Bewertung

Im Rahmen von Software Engineering I sind die verschiedenen Phasen des Software-Entwicklungsprozesses termingerecht mit den im Rahmenplan genannten Aufgaben entsprechend den ausgegebenen Arbeitsblättern abzurechnen.

Pro Aufgabenblatt können maximal 5-35 Punkte erreicht werden, wobei in die Bewertung sowohl Qualität als auch Termintreue eingehen.

***Zu jedem Review (dazu später mehr) ist vom Projektleiter ein in der Gruppe abgestimmter Projektstatusbericht mit Ressourcenanalyse und eine neue Projektplanung für die folgende Periode abzugeben.***

**Die Leistung der gesamten Gruppe ist die Basis für die Bewertung. In den Projektstatusberichten und in den Reviews muss jedoch deutlich vom Projektteam herausgearbeitet werden, welchen Anteil die einzelnen Projektmitglieder an der Gesamtleistung haben, damit auf dieser Basis eine angepasste Individualbewertung möglich ist. Dazu ist bereits in der Planung auf die ausgewogene Verteilung der Arbeit in der Gruppe zu achten.**

Zum erfolgreichen Abschluss des Projekts müssen aus den Aufgabenblättern wenigstens **50 % der Punkte** erreicht sowie **alle Reviews** zu den angesetzten Terminen erfolgreich absolviert sein.

Achtung!!!

Lesen bzw. studieren Sie sämtliche Unterlagen, die Sie gerade bekommen haben:

* Dieses Dokument: 01-Einführung
* 02-Rahmenplan
* 03-Aufwanderfassung
* 04-Review
* Aufgabenblätter 1 bis 6 (inkl. Zusatzblatt zu Aufgabeblatt 3)

Insbesondere ist ein Studium der Aufgabenblätter direkt am Anfang des Projektes ***(und nicht erst, wenn sie laut Rahmenplan fällig sind!)*** äußerst ratsam. Etwaige Fragen zu den Inhalten bzw. Unklarheiten bzgl. der Interpretation des darin beschriebene sind umgehend und generell so früh wie möglich mit Ihren Betreuern zu klären.

Ihre Betreuer sind nur da, um Sie mit Rat zu unterstützen (und Ihre Arbeit zu den festgestellten Terminen zu beurteilen). Das Projekt ist Ihre Verantwortung! Sie sind vor allem selbst verantwortlich diesen Rat auch einzuholen. Und vor allem: diesen auch zu beherzigen!

Literaturverzeichnis

[1] Ludewig, Jochen, (2001): "Ziele der Softwaretechnik" in: Ludewig (Hrsg.) Praktische Lehrveranstaltungen im Studiengang Softwaretechnik: Programmierkurs, Software-Praktikum, Studienprojekte, Fachstudie. URL: <http://www.informatik.uni-stuttgart.de/studium/studierende/bsc-studiengaenge/softwaretechnik/studienprojekte/PraktLehr5.pdf> (letzter Zugriff: 05.03.2019).

[2] Balzert, Helmut; Liggesmeyer, Peter (2011): Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. 3. Aufl. Heidelberg: Spektrum Akademischer Verlag (Lehrbücher der Informatik).

[3] Essigkrug, Andreas; Mey, Thomas (2003): Rational Unified Process kompakt. Heidelberg: Spektrum Akad. Verl.

[4] Drähter, Rolf; Koschek, Holger; Sahling, Carsten (2013): Scrum – Kurz und Gut: O’Reilly (D-Punkt Verlag).