

## Softwaretechnik

Seminar im Wintersemester 2020 / 2021

### Gruppe 21

Ender, Till - *Matrikelnummer:*  
Häußler, Jonathan - *Matrikelnummer:*  
Kalina, Joshua - *Matrikelnummer:*  
Kärst, Tobias - *Matrikelnummer:*  
Luo, Binbin - *Matrikelnummer:*

Technische Universität Ilmenau  
Deutschland  
16.10.2020

---

# 1 Seminar - Werkzeuge Softwareentwicklung

## Aufgabe 1 - Werkzeugkategorien

**Aufgabenstellung.** In der Softwaretechnik gibt es eine große Menge an Programmen und Systemen, die bei der Entwicklung eines Softwareprojekts verwendet werden.

- In welche Kategorien können die Werkzeuge unterteilt werden? Nennen Sie jeweils ein Beispiel.
- Welchen Zweck verfolgen Werkzeuge der einzelnen Kategorien?

### Lösung.

1. **Modellierungs-Tools** (UML) - Abstrakte Möglichkeit Software und Systeme (Architektur, Systemdesign) zu beschreiben ohne irrelevante Details zu beachten. Solche Abstraktionen haben folgenden Zweck: Dokumentation, Kommunikation, Spezifikation, Vertragsabschluss und Ausschreibung.
2. **Implementierungs-Tools** (Java, OOP, MongoDB) - Ermöglichen dem Programmierer mit Abstraktionen der Formen Programmiersprachen, Programmierparadigmen, Datenbanken und vieles mehr, das Modell technisch einfach zu realisieren über die Schnittstelle des Tools. Dabei können die Programmiersprachen erneut abstrahiert werden mittels Frameworks/Stacks, damit häufige Probleme nicht immer wieder erneut implementiert werden müssen (ReactJS).
3. **Entwicklungsplattformen/Editoren** (IntelliJ IDEA, Visual Studio Code) - Einfache Umgebungen (Texteditoren) helfen den Programmierer, Quelltext einer konkreten Programmiersprache zu formulieren oder Änderungen vorzunehmen. Auch spezielle GUI-Editoren sind dabei denkbar. Integrierte Entwicklungsumgebungen sind dabei viel komplexer und übernehmen mehrere Aufgaben dieser Liste gleichzeitig.
4. **Build-Tools** (Compiler) - Die konkret Implementierten Systeme müssen kompiliert oder interpretiert werden, damit ein Computersystem diese auch ausführen können
5. **Versionsverwaltung** (Git) - Speicherung und Verwaltung von Programmen mit mehreren Versionen um viele Backups zu erstellen und den Entwicklungsprozess zu dokumentieren.
6. **Programmverwaltung** (JUnit, Github) - In Programmen können Fehler auftauchen, die nicht einfach zu entdecken sind, weswegen Debugging-Tools benötigt werden. Um solche Fehler rechtzeitig zu entdecken, müssen die Programme getestet werden, da-

mit sichergestellt werden kann, dass das Programm das erwartete Verhalten aufweist. Diese Tests können automatisch vor jedem Hinzufügen zur Versionsverwaltung gesehenen, damit kein falscher Quelltext gespeichert wird (Deployment).

## Aufgabe 2 - Github

**Aufgabenstellung.** Machen Sie sich mit den Möglichkeiten zur Planung von Softwareprojekten am Beispiel von Github vertraut.

- (a) Welche Instrumente der Projektplanung bietet die Plattform Github und was sind deren wichtigste Funktionen und Eigenschaften?
- (b) Welche Personengruppen können am Prozess der Projektplanung beteiligt sein?

**Lösung.**

- (a) Neben allgemeinen Code-Verwaltungs-Features von git, bietet Github noch viele weitere Dienste, die unter anderem auch der Projektplanung dienen. In einem Github Repository besteht ebenfalls die Möglichkeit, Aufgabenlisten hinzuzufügen, welche dazu dienen dass Aufgaben geplant und der Fortschritt verfolgt werden können. Außerdem kann eine Wiki für das Repository erstellt werden um unter anderem eine Roadmap des Projektes zu erstellen oder die Software zu dokumentieren.
- (b) Am Prozess der Projektplanung kann im Prinzip jeder teilnehmen, falls dies nicht weiter restriktiert ist.

## Aufgabe 3 - Versionsverwaltung

**Aufgabenstellung.** Versionierung ist ein zentrales Element bei der konsistenten Gruppenarbeit an Projekten.

- (a) Welche Funktionen bietet eine Versionsverwaltung und welche Vorteile ergeben sich für die Entwicklung?
- (b) Was bedeuten die folgenden Begriffe im Kontext der Versionsverwaltung der GitSoftware?
  - Branch, Checkout, Pull-Request & Merge, Tags
- (c) Wie sollte die Versionierung während der Entwicklung verwendet werden?

**Lösung.**

- (a) Eine Versionsverwaltung ist ein System, welches die Aufgabe übernimmt Änderungen im Code/in den Daten/in Dokumenten zu erkennen und diese in unterschiedlichen Entwicklungsversio-

nen zu archivieren. Dabei sollte ein solches technisches System folgende Funktionen aufweisen:

- Protokollierung der Änderungen/Versionen erstellen - Verfolgt den Fortschritt des Projektes und erlaubt es nachzuziehen, wer und wann etwas verändert wurde
- Archivierung Programmzustände: Erstellt Versionen des Projekt zu sinnvollen Zeitpunkten
- Wiederherstellung/Backups: Möglichkeit auf eine ältere Version zurückzugreifen falls notwendig
- Entwicklungszweige: Möglichkeit für mehrere Programmierer auf unterschiedlichen Zweigen gleichzeitig zu arbeiten und später zusammenzufügen

(b)

**Hervorhebung (Branch).** Ein Branch ist eine Verzweigung/Abzweigung einer Version, um an einem dem Branch entsprechenden Feature zu arbeiten. Er repräsentiert also in anderen Worten eine unabhängige Entwicklungslinie, in der verschiedene Funktionen isoliert voneinander entwickelt werden können.

**Hervorhebung (Checkout).** Mit einem Checkout kann man zwischen zuvor erstellten Zweigen wechseln.

**Hervorhebung (Pull-Request).** Mit einem Pull-Request können unabhängige Entwickler oder Teammitglieder den für dieses Projekt Zuständigen Entwicklern mitteilen, dass ein Feature fertig entwickelt wurde. Diese können dann den Code überprüfen und freigeben, dass der Zustand mit einem anderen Branch merged (zusammengefügt) wird.

**Hervorhebung (Merge).** Dieser Befehl erlaubt es, zwei unabhängige Branches (Entwicklungslinien) in einen Einzigen zu überführen bzw. zusammenzufügen. Dabei wird darauf geachtet, dass keine Konflikte entstehen und falls doch, muss ein Entwickler manuell eingreifen.

**Hervorhebung (Tags).** Mit einem Tag kann ein bestimmter Zustand im Projektverlauf (eine Version) markiert werden können, da diese relevanter sind als andere. Typischerweise werden damit richtige Softwareversionen (Releases) gekennzeichnet, damit auf diese einfacher zugegriffen werden kann (Viele kleine commits, aber nur die wichtigsten kennzeichnen).

(c) Es sollten so viele Versionen wie möglich erstellt werden mit ein-

deutigen Beschreibungen, da somit ein genauer Überblick über das Projekt entsteht und eventuelle Fehler die sich eingeschlichen haben schneller identifiziert werden können.

## Aufgabe 4 - Entwicklungsumgebungen

**Aufgabenstellung.** Welche Funktionen bringt eine Entwicklungsumgebung (IDE) mit und wie integrieren sie sich in den Entwicklungsprozess?

**Lösung.**

**Hervorhebung** (IDE). Eine Software-Entwicklungsumgebung (engl. Integrated Development Environment) besteht aus einer strukturierten Menge integrierter Werkzeuge und Bausteine (Softwarewerkzeuge), die Entwickler bei der Softwareentwicklung bei anfallenden Tätigkeiten unterstützen sollen.

Die wichtigsten Funktionen sind dabei folgende: Quellcode-Editor (Programmierung mit Syntaxhervorhebung, Autovervollständigung, Syntax-Prüfung), Automatisierung lokaler Builds (Kompilierungsprozess) und ein Debugger (Programm manuell durchlaufen um Fehler zu finden). Dabei kann eine IDE jedoch alle möglichen Softwaretools beinhalten, denn das Ziel ist es die Effizienz des Entwicklers zu steigern.

Eine IDE steigert die Effizienz der Entwickler durch das Anbieten vieler integrierten Funktionen aus unterschiedlichen Bereichen (Dokumentation, Codeverwaltung, etc.). Sie können also (je nach IDE) in jedem Zustand des Entwicklungsprozess eine Abhilfe schaffen.

## Aufgabe 5 - Dokumentation

**Aufgabenstellung.** Dokumentation ist ein Teil des kompletten Entwicklungsprozesses.

- (a) Nennen Sie verschiedene Arten der Dokumentation. Welche Programme oder Hilfsmittel können dabei verwendet?
- (b) Welche Rolle spielen Modellierung und UML in diesem Prozess?

**Lösung.**

- (a)
  - Methodendokumentation - Grundlagen der Software (Modellierung, UML)
  - Programmiererdokumentation - Beschreibung Quelltext (Code-Editor mit Syntax Highlighting)
  - Installationsdokumentation - Erforderliche Hardware, Software, Prozeduren der Installation (Installationsprogramm)

- Benutzerdokumentation - Informationsmaterial für Endbenutzer
- Datendokumentation - Informationen über die Daten
- Entwicklungsdokumentation - Information einzelner Versionen, Changelogs (Git, Github)

(b) Durch ein Modell erstellt man eine Abstraktion eines Systems, wobei irrelevante Informationen vorerst ausgelassen werden, um sich auf die wichtigen Informationen zu konzentrieren. Dadurch wird der Grundstein des Projektes gelegt, welches nach und nach verfeinert werden können (Großes Projekt nach und nach verfeinern, da auf einmal zu komplex um zu überblicken). UML ist nun ein grafisches Beschreibungsmittel für die Aspekte des Softwareentwurf und können zur Spezifikation, Konstruktion, zum Entwurf, Visualisierung oder auch zur Dokumentation von Software genutzt werden; also eine allgemeine Beschreibung zur Modellierung von Systemen.

## 2 Seminar 2 - Anforderungsanalyse

### 2.1 Aufgabe 1 - Pflichtenheft

### 2.2 Aufgabe 2 - Anforderungsdefinition

ID	Bezeichner	Beschreibung	Priorität
NF01	Benzinverbrauchs- klassen	Es gibt 3 Benzinverbrauchsklassen. Jede Benzinklasse ist durch eine anders gefärbte Plakette gekennzeichnet.	Muss
NF02	Autos	Objekte, die sich mit gleichbleibender Geschwindigkeit und Richtung über den Bildschirm bewegen. Es sollen nicht alle Autos die gleiche Geschwindigkeit haben. Außerdem ist jedes Auto genau einer der drei Benzinverbrauchsklassen zugeordnet.	Muss
NF03	Abgaswolken	Jedes Auto stößt eine Abgaswolke aus. Die Größe/ Intensität einer Abgaswolke stimmt mit dem Benzinverbrauch des zugehörigen Autos überein.	Muss
NF04	Plakette	Jedes Auto ist durch eine farbige Plakette gekennzeichnet. Die Plakettenfarbe soll nicht immer mit der tatsächlichen Verbrauchsklasse des Autos übereinstimmen, damit der Anwender einige Farben erst richtig zuordnen muss.	Muss
NF05	Hindernisse	Es gibt einige Objekte (z.B. Bäume), die für kurze Zeit die Sicht auf ein Auto verdecken, wenn es aus Anwenderperspektive hinter dem Objekt entlang fährt.	Muss
NF06	Punktestand	Der aktuelle Punktestand soll gespeichert und oben auf dem Bildschirm angezeigt werden.	Muss
NF07	Leben des Spielers	Der Spieler startet das Spiel mit 3 Leben. Die verbliebenen Leben werden oben rechts auf dem Bildschirm angezeigt.	Muss
NF08	Level	Es soll unterschiedliche Level geben, die sich durch die Anzahl und Anordnung der Hindernisse, sowie die Anzahl und Geschwindigkeiten der Autos unterscheiden.	Muss

ID	Bezeichner	Beschreibung	Priorität
F01	Startbildschirm	Beim Start des Spiels soll ein Startbildschirm mit einer Spielanleitung angezeigt werden, bis der Spieler die linke Maustaste betätigt.	Muss
F02	Plakette ändern	Wird die Plakette eines Autos durch den Anwender angeklickt, soll sich die Farbe der Plakette ändern.	Muss
F03	Überprüfen der Plakettenzuweisung	Sobald ein Auto den Bildschirm überquert hat und den Rand "berührt", soll überprüft werden, ob die Plakettenfarbe mit der Verbrauchklasse des Autos übereinstimmt. Falls nicht, verliert der Spieler ein Leben.	Muss
F04	Spielende beim Verlust des letzten Lebens	Sobald der Spieler sein letztes Leben verliert, wird das Spiel beendet und der Punktestand angezeigt.	Muss
F05	Level beenden	Sobald der Spieler eine definierte Punktzahl erreicht hat, wird das aktuelle Level beendet und das nächste Level gestartet. Ist das letzte (3.) Level erfolgreich absolviert, wird das Spiel beendet und der Punktestand angezeigt.	Muss
F06	Spiel unterbrechen	Der Spieler soll die Möglichkeit haben, das Spiel über die ESC-Taste zu Pausieren. Es soll dann die Möglichkeit geben, das Spiel fortzusetzen, oder zu beenden.	Kann

## 2.3 Aufgabe 3 - Use-Case-Diagramme



### **3 Seminar 3 & 4 - Grobentwurf**

#### **3.1 Aufgabe 1 - Klassendiagramm Grobentwurf**

#### **3.2 Aufgabe 2 - Sequenzdiagramm**

## **4 Seminar 5 - Feinentwurf**

### **4.1 Aufgabe 1 - Klassendiagramm Feinentwurf**

### **4.2 Aufgabe 2 - Aktivitätsdiagramm**

## **5 Seminar 6 - Projektplanung**

### **5.1 Aufgabe 1 - Projektplanung**

### **5.2 Aufgabe 2 - Frameworks**

## **6 Zusatzaufgabe - Entwurfsmuster**

## **Abkürzungsverzeichnis**