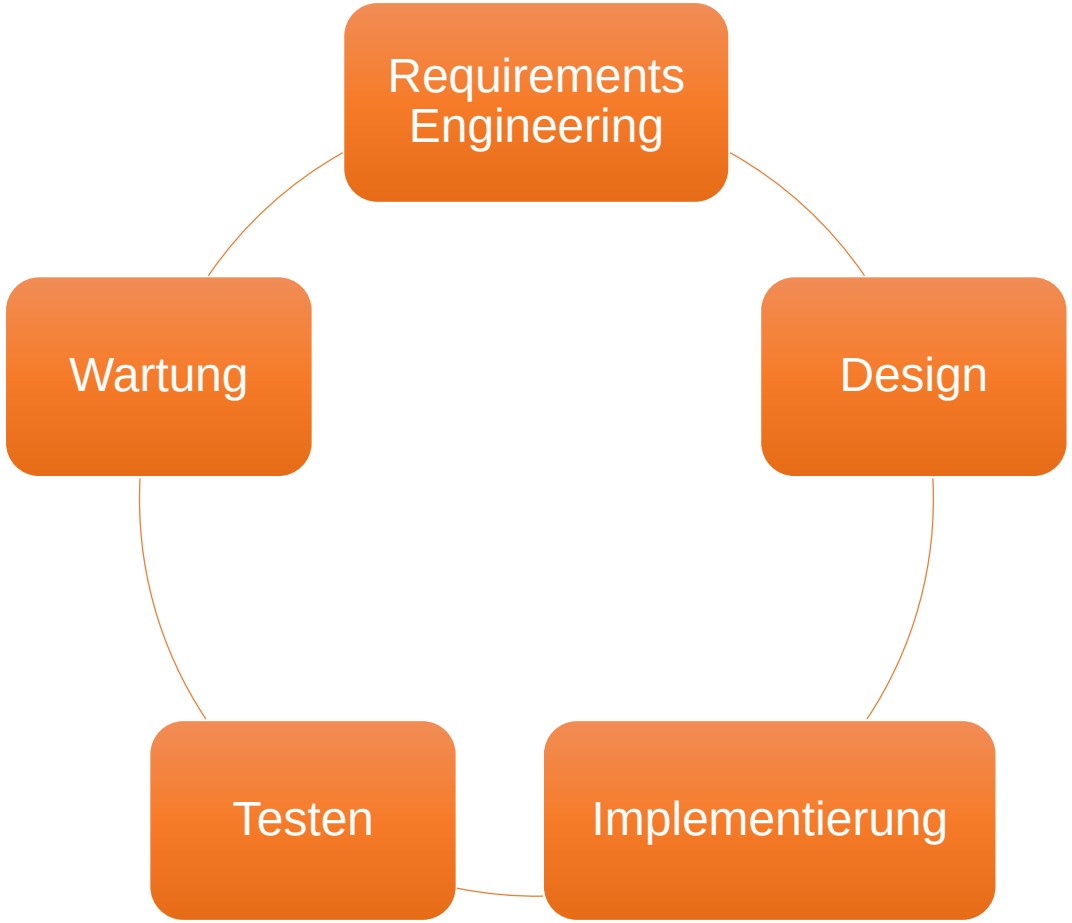
Übung 01

Requirement Engineering

http://www.hwk-chemnitz.de/fileadmin/user_upload/ausbildung/Pruefungen/Gesellenpruefungen/Fragezeichen.JPG

1. Software Life Cycle

Zählen Sie die Aktivitäten der Software Entwicklung auf, beschreiben Sie kurz deren Inhalt und nennen Sie den jeweils dabei entstehenden „Output“.

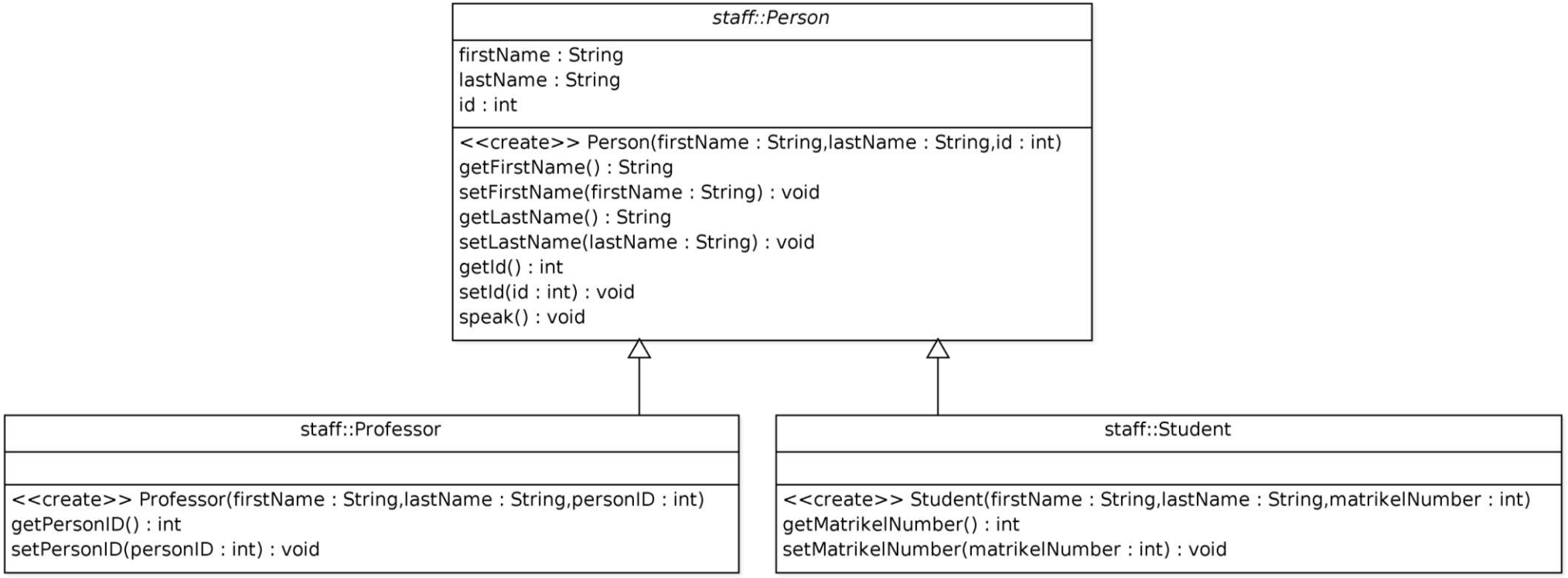


- **Requirement Engineering**
  - Die Anforderungen des Kunden werden ermittelt und im *Lastenheft* festgehalten (Kundensicht)
  - Die Anforderungen werden modelliert und spezifiziert (ggf. formal). Die Ergebnisse werden im *Pflichtenheft* hinterlegt. (Entwicklersicht)
- **Design**
  - Ein Lösungsansatz basierend auf den Ergebnissen der Analyse wird erstellt. Hierzu bietet es sich an, *UML* und ggf. *ER-Diagramme* zu erstellen
- **Implementierung**
  - Die Umsetzung des Designs in ausführbaren Quellcode
- **Testen / Validation**
  - Prüfen, ob die Implementierung die Zielsetzung aus den Requirements erfüllt
  - Erstellen von Testfällen, anhand derer ein Testreport zur Verfügung gestellt werden kann
- **Wartung**
  - Gewährleistung der kontinuierlichen Korrektur und Wartung der Implementierung, ggf. unter Berücksichtigung *geänderter* oder *neuer Anforderungen*

2. Universitäts-Verwaltungs-Programm

Für ein Universitäts-Verwaltungs-Programm sind vorerst nur die Komponenten Professoren und Studierende zu berücksichtigen.

- Beide Personengruppen haben je einen Vor- und Nachnamen, Professoren außerdem eine Personennummer, Studierende eine Matrikelnummer.
- Programmieren Sie die nötigen Java Klassen, und achten Sie dabei insbesondere auf einfache Erweiterbarkeit. Die Klassen sollen mindestens Getter und Setter für jede Membervariable und einen Konstruktur enthalten.



2. Universitäts-Verwaltungs-Programm

```
public abstract class Person {
    private String firstName;
    private String lastName;
    private int id;

    Person(String fn, String ls, int id) {
        this.firstName = fn;
        this.lastName = ls;
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String fn) {
        this.firstName = fn;
    }
    public String getLastName() {
        return lastName;
    }
    …
```

```
    …
    public void setLastName(String ln) {
        this.lastName = ln;
    }

    int getId() {
        return id;
    }

    void setId(int id) {
        this.id = id;
    }

    public void speak() {
        System.out.println("Hello, my name is "
        + firstName + " " + lastName + " ("
        + id + ").");
    }
}
```

2. Universitäts-Verwaltungs-Programm

```java
public class Student extends Person {

    public Student(String fn, String ln,
                   int mNr) {
        super(fn, ls, mNr);
    }

    public int getMatrikelNumber() {
        return super.getId();
    }

    public void setMatrikelNumber(int mNr) {
        super.setId(mNr);
    }
}
```

```java
public class Professor extends Person {

    public Professor(String fn, String ls,
                     int pID) {
        super(fn, ls, pID);
    }

    public int getPersonID() {
        return super.getId();
    }

    public void setPersonID(int pID) {
        super.setId(pID);
    }
}
```

2. Universitäts-Verwaltungs-Programm

```java
import staff.Professor;
import staff.Student;

public class Main {

    public static void main(String[] args) {
        Professor siegmund = new Professor("Norbert", "Siegmund", 123456);
        Student   philipp  = new Student("Philipp", "Seltmann", 654321);

        siegmund.speak();
        philipp.speak();
    }
}
```

```
> Hello, my name is Norbert Siegmund (123456).
> Hello, my name is Philipp Seltmann (654321).
```

3. Anforderungsbeschreibung: Volere – Snow Card

- Programmierer möchten ...
  - gerne wissen, warum eine bestimmte Requirement wichtig ist.
  - präzise Angaben haben, was damit gemeint ist.
- Tester benötigt...
  - Eine testbare Messung, die die Requirements widerspiegeln.
- Wirtschaftsanalysten benötigen ...
  - Wissen darüber, inwieweit eine Requirement andere Requieirements beeinflusst
- Wartungsarbeiter benötigen ...
  - Wissen darüber, wie ein neues Requirement andere schon implementierte Requirements beeinflusst

Volere

3. Anforderungsbeschreibung: Volere – Snow Card

Requirement #: Requirement Type: Event/Use Case #'s:

Description:

Rationale:

Originator:
Fit Criterion:

Customer Satisfaction: Customer Dissatisfaction:
Priority: Conflicts:
Supporting Materials:
History:

Volere

3. Anforderungsbeschreibung: Volere – Snow Card

Führen Sie eine Anforderungsbeschreibung nach Volere für jeweils eine funktionale und eine nicht funktionale Anforderung durch.

Funktionale Anforderung: Spezifiziert, **was** das System tun soll
Nicht Funktionale Anforderung: Spezifiziert, **wie** das System sich verhalten soll (Qualitätsattribute eines Systems)

Funktionale Anforderung	Nicht Funktionale Anforderung

• Anzeigen und Bearbeiten von Bildern
• Benutzer kann Bilddimensionen Abfragen
• Software soll das Manipulieren von Bildern erlauben
  • Drehen
  • Vergrößern / verkleinern (skalieren)
  • Neue Farbwerte im Bild setzen
  • Zusammenfügen zweier Bilder
	• Manipulationen sollen innerhalb von 0.2 Sekunden erfolgen

Volere

3. Anforderungsbeschreibung: Volere – Snow Card

Requirement #: 1

Requirement Type: funktional

Event/Use Case #'s: 1.1

Description:

Vergrößern und verkleinern von Bildern. (Skalieren)

Rationale:

Hauptfunktion eines Bildbearbeitungsprogramms

Originator: Mephisto GmbH

Fit Criterion:

Vergleich der Bilddimensionen vor und nach der Skalierung anhand des Skalierungsfaktors.

Customer Satisfaction: 3

Customer Dissatisfaction: 4

Priority: 4

Conflicts: -

Supporting Materials: Skullkopp02-scale-your-head.pdf

History: 28.10.2016, P. Seltmann, …

Volere

3. Anforderungsbeschreibung: Volere – Snow Card

Requirement #: 2

Requirement Type: Nicht funktional

Event/Use Case #'s: 1.2

Description:

Ausführung von Bildmanipulationen soll innerhalb von 0.2 Sekunden möglich sein

Rationale:

Gewährleisten eines natürlichem Workflow; Minimierung von Wartezeiten, …

Originator: Mephisto GmbH

Fit Criterion:

Zeitmessung verschiedener Bildmanipulationentestreihen. Erfüllt das Kriterium, wenn 95% der Manipulationen <= 0.2 Sekunden benötigen.

Customer Satisfaction: 2

Customer Dissatisfaction: 5

Priority: Hoch

Conflicts: -

Supporting Materials: -

History: 28.10.2016, P. Seltmann, …

Volere

http://www.hwk-chemnitz.de/fileadmin/user_upload/ausbildung/Pruefungen/Gesellenpruefungen/Fragezeichen.JPG