

# Software Engineering

## Prüfungsleistung: Projekt

Moritz Köhler

September 22, 2025

## Contents

<b>1</b>	<b>Vorbereitung/ Interview</b>	<b>2</b>
1.1	Kunde erzählt: . . . . .	3
1.2	Weitere Punkte aus dem Interview: . . . . .	3
1.3	Begriffe für das Begriffslexikon: . . . . .	4
<b>2</b>	<b>Geordnetes Interview</b>	<b>5</b>
2.1	Ist-Zustand . . . . .	5
2.2	Soll-Zustand . . . . .	5
2.3	Datenstruktur . . . . .	5
2.4	Achten auf: . . . . .	7
2.5	Quality of Live . . . . .	7
2.6	Für V.2 . . . . .	7
<b>3</b>	<b>Anforderungsanalyse</b>	<b>8</b>
3.1	Anforderungsdokument/ Lastenheft . . . . .	8
3.2	Begriffslexikon . . . . .	8
3.3	Spezifikation/ Pflichtenheft . . . . .	8
<b>4</b>	<b>Design</b>	<b>9</b>
4.1	XML . . . . .	9
4.1.1	TreeView . . . . .	9
4.1.2	View Exam . . . . .	9
4.1.3	View Chapter . . . . .	9
4.1.4	View Task . . . . .	10
4.2	PDF . . . . .	10

Hinweis: Diese Dokument umfasst die Vorbereitung und Mitschrift aus dem Kundeninterview und wurde mit den Grundlagen und Notizen der Anforderungsanalyse erweitert. Die Fertige und vollständige Anforderungsanalyse ist in einem Seperatem Dokument,

# 1 Vorbereitung/ Interview

Situation: Ich bin Software-Entwickler und soll ein Tool zur Erstellung von Klausuren aus Prüfungsaufgaben.

1. Wie liegen die einzelnen Prüfungsaufgaben vor? Pdf, Word, L<sup>A</sup>T<sub>E</sub>X oder andere. Soll eine eigene Datenstruktur für Prüfungsaufgaben erstellt werden?

Aktuell Libre Office mit allen Aufgaben

Der Datensatz soll neue Teilaufgaben erstellen können mit Text (keine Grafiken) kein Import nötig.

Grafik Import wäre Supper (v.2 erst mal nicht implementieren)

2. Sind Teilaufgaben/Unteraufgaben in einer Prüfungsaufgabe enthalten oder liegen diese einzeln vor.

Ja siehe unten.

3. Wie sollen die Klausuren aus den Prüfungsaufgaben generiert werden? Per Zufall, Händisch oder anders?

Wird vom Nutzer angegeben.

4. In welcher Reihenfolge sollen die Aufgaben erscheinen? Soll die Reihenfolge manuell bearbeitbar sein?

Die Kapitel werden angegeben.

5. Welches Ausgabemedium soll entstehen: Pdf, Digital oder Druck.

als PDF.

6. Sollen nach Aufgaben White Space für Lösungen sein? Wenn Ja: Blanko, Liniert, kariert, gepunktet? Oder sind diese schon in den Daten der Aufgaben hinterlegt?

Als Kasten mit der Größe abhängig von der Punktzahl. Aber noch anpassbar.

7. Bestehen die Daten auch aus Grafiken oder nur Text?

nur Text, Grafiken für v.2

8. Soll die Benotung in dem Tool möglich sein?

9. Wie soll die Punkteverteilung der Aufgaben geregelt werden.

Sind in den Teilaufgaben. Das Kapitel hat eine Zielpunktzahl.

10. Soll es ein Inhaltsverzeichnis der Klausur geben?

Ja im Deckblatt, nicht Teil der Anforderung.

11. Wie ist das mit einem Deckblatt.

Wird manuell vom Nutzer außerhalb vom Tool hinzugefügt.

12. Sollen die Fertigen Klausuren verschlüsselt werden? Wenn Ja wie und mit welchem Standard?

13. Sollen Korrekturen von z.B. Rechtschreibung in den Prüfungsaufgaben innerhalb des Tools möglich sein. (Aufwendig, da Textbearbeitung implementiert werden muss)

In der Datenerstellung.

14. Soll die Software eigenständig sein, oder z.B. Word ergänzen, z.B. als neuen Reiter.

Eigenständig, Lokal

15. Geht es bei Klausuren auch um anderer Prüfungsleistung wie Projekte.

16. Gibt es für die Prüfungsaufgaben irgendwelche Rechte? Dürfen manche nur ihre Eigenen Aufgaben sehen?

Es gibt nur einen Lokalen Nutzer

17. Wer darf die Software Nutzen? Brauchen wir eine Benutzerverwaltung oder kann die bisherige Login Funktion der Hochschule genutzt werden?

Open Source, Lokal

### 1.1 Kunde erzählt:

Eine Software als einzelperson lokal auf dem Rechner. Mit Zufallsgenerator (Eine Klausur keine Varianten). Klausuraufgaben und Teilaufgaben in einer Datensammlung.

Verschieden Betriebssystem als Java Anwendung. Soll Open Source sein mit Java FX.

Zwei generieren Buttons: Klausur und Probeklausur.

Es gibt eine Probeklausur, die als Leitlinie gilt.

### 1.2 Weitere Punkte aus dem Interview:

Eine Aufgabe (mit Titel als **Kapitel**) besteht aus: Nummeriert (automatisch), **Teilaufgabe** (Buchstabe). Eine Teilaufgabe ist die Aufgabe mit Textfeld, Punktezah, Musterlösung, Größe des White-Space (als Kasten) generieren durch Punktzahl. Teilaufgaben gehören zu einem Themengebiet. Musterlösung soll zusätzlich als Zweit PDF gedruckt werden.

Jede Teilaufgabe hat Kategorie (Schwer, Mittel, Einfach), Wie viele Aufgaben mit Welche Kapitel es gibt. Eine Teilaufgabe kann gekennzeichnet werden, das es nur für Probeklausur.

Variante an Teilaufgaben. Es soll nur eine Variante in der Generierung vorkommen. **Variante** die Teilaufgaben haben hier die gleicher Punktzahl. Ist aber Egal.

Eine Aufgabe soll aus Teilaufgaben eines Kapitels mit einer Zielpunktzahl. Die Schwierigkeit soll in dritteln erfolgen (aber mindestens eine Pro Schwierigkeit). Wenn das Dritteln nicht möglich ist, soll der Druck verhindert werden.

Wenn die Zielpunktzahl nicht möglich ist: Es sollen nur mögliche Zielpunkte zur Auswahl sein.

Versionierung der Teilaufgabe ist nicht notwendig.

Randplatz: 2cm

Kopf und Fußzeilen. Fußzeile mit Seitenzahl 1/x, beginnend mit 1.

Jedes Kapitel ist eine Aufgabe. 1 zu 1

Die Sprache ist auf English. Es soll aber möglich sein in Zukunft weitere Sprache der Benutzeroberfläche hinzuzufügen.

Probeklausur: Ist identisch

Code soll gut Dokumentiert sein (English)

Die Daten sollen als xml gespeichert werden. Einlesen und Laden für unterschiedliche Module.

Evtl. eine Vorschau des PDF.

Gesamtpunktzahl ist nicht relevant. Wird indirekt durch die Kapitel gesteuert.

In V.2 html fähiger Text

Reihenfolge der Teilaufgaben ist egal. Einfach nach Schwer

Benutzeroberfläche: einfach und übersichtlich mit Buttons, + hinzufügen, - löschen, exit, laden und speicher von xml (speichern unter). Jede Interaktion braucht einen Hover Text.

User Freie Software

### **1.3 Begriffe für das Begriffslexikon:**

1. Klausur
2. Kapitel (eine Aufgabe)
3. Aufgabe (Teilaufgaben)
4. Punktzahl (Wertebereich und Inkrement)
5. Teilaufgabe (Name, Punkte (0,5er Schritte), Schwierigkeit, Varianten)
6. Varianten (Frage, Lösung)
7. Schwierigkeitsgrad (Datentyp)
8. Zufallsgenerator
9. Probeklausur
10. Lösung

## 2 Geordnetes Interview

### 2.1 Ist-Zustand

Es gibt Altklausuren und Probeklausuren von zwei verschiedene Modulen in dem gewünschtem Format.

Hier gibt es Eine Aufgabe pro Kapitel und nur eine Auswahl an Kapitel der Vorlesung kommt in der Klausur vor. Diese sind Nummeriert von  $1 \dots n$

Jede Aufgabe besteht aus mehreren Teilaufgaben, welche eine Punktzahl haben. Diese sind von  $a \dots z$  Nummeriert und haben eine Antwortbox, welche nach der Punktzahl skaliert ist. Große Teilaufgaben haben auf den folgenden Seiten eine Ganzseitigen Antwortbox mit Zuordnung zu der Aufgabe.

Zu manchen Teilaufgaben gibt es noch Grafiken. Bei manchen Teilaufgaben gibt es vorgefertigten Text in der Antwortbox.

Das Deckblatt beinhaltet einen Überblick über die Bepunktung der Aufgaben und Teilaufgaben.

### 2.2 Soll-Zustand

Es soll eine Open Source Java FX Anwendung entstehen, welche eine zufalls-generierte Klausur oder Probeklausur als pdf genieren kann. Gleichzeitig soll mit der Endung "lsg" eine Lösung geniert werden, wo in den Antwortboxen die Musterlösung sichtbar ist.

Jeder Text in der Klausur ist reiner Fließtext.

Das Deckblatt, sowie Bilder und Formeln müssen noch nicht implementiert werden.

Die Formatierung soll so nahe wie möglich an den gegebenen Alt- und Probeklausuren sein. Randplatz 2cm, Kopf- und Fußzeilen und Seitenumerierung.

Die Daten für die Klausuraufgaben sollen pro Modul als xml gespeichert werden.

Die Sprache der Anwendung soll Englisch sein. Jedoch soll die Sprache der UI in Zukunft ergänzt werden. Hierzu sollen die Notwendigen Strukturen geschaffen werden.

Für jede Interaktion in der Software soll ein Hover-Text erscheinen. Dieser soll auch von der Sprachauswahl verändert werden.

Der Quelltext soll auch auf Englisch erfolgen. Somit sind Kommentare und Namensgebung auf Englisch.

### 2.3 Datenstruktur

Eine **Klausur** besteht aus einer Liste an Kapitel. Es soll ausgewählt werden

- Liste an Kapitel

Ein **Kapitel** hat eine Überschrift und beinhaltet eine Aufgabe.

- Überschrift

- Überschrift Kürzel
- genau eine Aufgabe (oder mehrere. Muss geklärt werden!)

#### **Fragen zu Kapiteln:**

Es gibt einen Widerspruch zu der Anzahl an Aufgaben pro Kapitel. Aus dem Interview ging eine 1 zu 1 mapping heraus und in den Probeklausuren hat ein Kapitel mehrere Aufgaben. Wechsel soll implementiert werden? Wenn die Software das kann, dann können Kapitel in Vorlesungen oder Lehrveranstaltungen gebündelt werden.

Eine **Aufgabe** hat entweder eine Teilaufgabe, welche keine weitere Nummerierung hat, oder mindestens 2 Teilaufgaben. Diese haben dann eine Nummerierung von a) bis z).

- Überschrift
- Liste an Teilaufgaben
- Gesamtpunktzahl

Eine **Teilaufgabe** besteht aus mindestens einer Variante, welche eine Fragestellung und evtl. eine Musterlösung jeweils als Fließtext hat.

Die Teilaufgabe lassen sich Zusätzlich in Schwierigkeitsgrade einteilen: Leicht, Mittel, Schwer. Das Verhältnis der Schwierigkeitsgrade von Teilaufgaben soll innerhalb einer Aufgabe 33% betragen.

Jede Teilaufgabe ist einem Scope zugeordnet. So kann es nur zu Generierung von Klausuren oder Probeklausuren verwendet werden. Bei bedarf kann eine Teilaufgabe auch für beides verwendet werden.

Die Teilaufgabe hat eine Punktzahl und eine Überschrift, welche in der UI und zur Generierung der Klausur angezeigt werden kann. Die Überschrift der Teilaufgabe taucht in dem generiertem pdf nicht mehr auf. Abhängig von der Punktzahl wird eine Größe der Antwortbox eingetragen.

- Überschrift
- Schwierigkeit (Leicht, Mittel, Schwer)
- Scope (Klausur, Probeklausur, Beides)
- Punktzahl
- Höhe der Antwortbox
- Liste an Varianten

#### **Variante**

- Fragestellung (Fließtext)
- Musterlösung (Fließtext)

## **2.4 Achten auf:**

Bei der Berechnung der Möglichkeiten an Teilaufgaben, sollte die Reihenfolge egal sein. So sind die Aufgaben ab gleichzustellen mit ba. Dies garantiert eine Gleichverteilung von Aufgabenkombination.

## **2.5 Quality of Live**

Passende Pagebreak nach Aufgaben. Hier kann dann auch noch Zusätzlich die Antwortboxen hochskaliert werden, sodass diese die gesamte Seite Füllen.

Page Preview bevor die Klausur gedruckt wird.

Bei Pdf gibt es auch interaktive Elemente, welche weiteren Text zum Vorschein bringen. Dies könnte evtl. für die Musterlösungen verwendet werden.

## **2.6 Für V.2**

Grafik Import für die Teilaufgaben oder Varianten.

Optional: Generierung des Deckblattes anhand der ausgewählten Kapitel und deren Aufgaben.

### 3 Anforderungsanalyse

Das Weitere sind Selbstgeschriebene Notizen. Die Eigentliche Anforderungsanalyse ist in einer Seperaten Datei.

#### 3.1 Anforderungsdokument/ Lastenheft

#### 3.2 Begriffslexikon

Begriff	Exam
Beschreibung	An Exam is composed of multiple Chapters and exported as an pdf-document
Bezeichnung	The Data which generates the pdf, are not saved in the tool
Unklarheiten	
Querverweise	Chapters, Operating System

Begriff	Chapter
Beschreibung	A Chapter is composed of a Heading and multiple Chapters are used to create an Exam. One Chapter represents a Task in the Exam.
Bezeichnung	A unique positive integer Number
Unklarheiten	
Querverweise	Exam, Task

Begriff	Task
Beschreibung	A Task is composed of multiple Subtasks and has a heading. This Heading is the Chapter Heading. One Task should have a goal score
Bezeichnung	Is Numbered from 1 to n.
Unklarheiten	
Querverweise	Chapter, Subtask, Score

Begriff	Subtask
Beschreibung	A Subtask is a text based task. It has a written question of plain text and a plain text answer which can be empty. It has a score of points. Each subtask also has a difficulty raiting. A Subtask may consist of multiple Variants, but always has at least one variant. In a Task, each Subtask is numbered alphabetically from a to z.
Bezeichnung	
Unklarheiten	
Querverweise	Score, Difficulty, Variant, Task

#### 3.3 Spezifikation/ Pflichtenheft



## 4 Design

### 4.1 XML

Wir haben verschiedene Views die alle von dem Tab XML erreichbar sind.

Diese Tab ist vertikal in zwei Hälften geteilt. Die Kleinere auf der Linken Seite ist ein TreeView, welcher die Struktur der .xml-Datei widerspiegelt:

1. exam
2. chapter
3. task
4. (variants) is not show, because it would be to cluttered.

Wenn es keine .xml Datei gibt, so ist in diesem TreeView nur das Exam, mit dem Name "Exam" und es gibt noch keine weiteren chapters oder tasks.

#### 4.1.1 TreeView

Dieser ist eine Strukturierte Navigation und wird mit einer Textsucher ergänzt. Diese zeigt alle Elemente an die den suchtext beinhalten. Wenn ein Blattelement den suchtext enthält, sollen alle Elternteile angezeigt werden. Die Suchfunktion fungiert also als ein Filter.

#### 4.1.2 View Exam

In diesem View lässt sich der Name des Exam ändern. So kann z.B. der Modulname verwendet werden. Dieser wird dann auch anstatt von "Exam" in dem TreeView angezeigt.

Als darzustellenden Informationen sind die Anzahl der Kapitel und wie viele (Diese Liste ist Alphabetisch sortiert):

- tasks and variants are in each chapter
- how many points these have in total
- and the distribution of points per difficulty

Aus dieser Ansicht soll es nicht möglich sein ein chapter zu löschen um Fehler zu vermeiden. Wenn auf ein bestimmtes chapter hier geklickt wird, so wird dies zum aktuellem View.

Es soll möglich sein ein neues chapter zu erstellen. Dies besteht einfach nur aus einem Namen und hat standardmäßig keine Aufgabe. (Ergänzend kann evtl. noch ein Kürzel ergänzt werden) Nach dem erstellen soll das neue chapter ausgewählt sein.

#### 4.1.3 View Chapter

In diesem View lässt sich der Name des Chapter ändern. Dieser wird dann auch anstatt von "Chapter" in dem TreeView angezeigt.

Als darzustellenden Informationen sind die Anzahl der Kapitel und wie viele:

- name of the chapter
- tasks and variants this chapter has
- how many points these have in total
- and the distribution of points per difficulty
- symbol: edit

Es soll möglich das gesamte chapter mit allen tasks unwiderrufflich zu löschen.

Es soll möglich sein neue Tasks zu erstellen. Ein Neuer Task wird immer mit einer Variant erstellt. Weitere Varianten können dann im Task selber erstellt werden. Nach dem erstellen einer Task bleibt der View des Chapters erhalten.

Unter den Edit und Create wird dann noch eine Liste an Task angezeigt. Diese haben als Name die abgekürzten question als Name und zeigt:

- sportend question of the first variant with ...
- amount of variants
- points for this task
- what difficulty it has
- and the visibility for the print as pdf (mock exam, exam of both)
- symbols: edit, delete

Aus dieser Liste soll es möglich sein die einzelnen tasks unwiderrufflich zu löschen. Wenn man diese bearbeiten möchte wird auf den task geklickt oder auf das bearbeiten Symbol.

#### 4.1.4 View Task

Es lassen sich die Notwendigen Attribute einer Task ändern:

- difficulty, choose between easy, medium and hard (one of three)
- scope (for printing): mock exam, exam (one or both)
- points (integer)

Darunter kommt eine Liste an allen Varianten der Frage. Diese haben folgende Felder:

- question (text)
- answer (text)
- symbol: edit, delete
- button: save changes

Jede Variante kann einzeln gelöscht werden und wenn die letzte Variante gelöscht wird, wird auch des Task gelöscht.

Die einzelnen Felder (question, answer) sollen auch bearbeitet werden können.

## 4.2 PDF