

# Anforderungen an die Projektarbeit in Software Engineering 2

Stand: 1.3.2024 FS24

Der benotete Leistungsnachweis im Modul «Software Engineering 2» besteht aus der hier beschriebenen Projektarbeit, welche als Einzelarbeit durchgeführt wird. Die formulierten Anforderungen stellen gleichzeitig die Bewertungskriterien dar, nach denen die Leistungen beurteilt werden.

## Das Wichtigste in Kürze

- Es wird ein Prototyp einer Web-Applikation mit Frontend, Backend und Datenbank entworfen und realisiert. Dabei setzen die Studierenden eine eigene Idee um. Die Dozierenden dürfen jedoch nach Abgabe der Projektbeschreibung Änderungen verlangen.
- Die Bewertung berücksichtigt den Projektinhalt (50%), die Dokumentation (25%) und die Präsentation (25%).

## Termine

- **2. Semesterwoche:** Einführung Projektarbeit.
- **Ende 4. Semesterwoche (17.3.24):** Abgabe der Projektbeschreibung.
- **5. und 6. Semesterwoche:** Pitch der Projektidee in der KK.
- **7. bis 14. Semesterwoche:** Unterstützung bei der Projektarbeit während der KK nach Bedarf.
- **Ende 14. Semesterwoche (26.5.24):** Abgabe der schriftlichen Arbeit und der Implementation. Die Details zur Abgabe sind am Ende dieses Dokuments beschrieben.
- **Kalenderwoche 23 2024 (3.6.-7.6.24):** Präsentation des Projekts, Abgabe der Präsentation. Alternativ darf in Ausnahmefällen auch in Semesterwoche 14 präsentiert werden.

**Wichtig:** Du hast für dieses Projekt 12 Wochen Zeit. Es gibt keine Erstreckung der Abgabefrist für Krankheiten von weniger als zwei Wochen Dauer. Längere Krankheiten müssen bis spätestens zwei Wochen vor Abgabefrist gemeldet und belegt werden. Plane dein Projekt so, dass du es rechtzeitig abgeben kannst. Nicht abgegebene Projekte werden mit 1.0 bewertet.

## Anforderungen an die Projektbeschreibung

Die Projektbeschreibung umfasst die folgenden Elemente:

- Ausgefüllte Explore-, Create- und Evaluate-Boards (md Format im Ordner /doc) ✓
- Use-Case Diagramm (drawio.svg Format im Ordner /doc/figures) ✓

Die Projektbeschreibung wird mit pass/fail bewertet. Die Abgabe erfolgt durch Abgabe der GitHub-URL via Moodle. David (DavidZhaw) muss beim Repo als Collaborator hinzugefügt werden.

## Anforderungen an den Pitch

Der Pitch dauert drei Minuten und basiert auf dem Inhalt der Projektbeschreibung. Der Pitch soll sich an mögliche Investoren oder Institutionen richten, welche das Projekt unterstützen oder finanzieren könnten. Die Mitstudierenden geben Feedback. Der Pitch wird bewertet (20% von Teil Präsentation, 5% der Gesamtwertung). Slides dürfen verwendet werden. Slides mit mehr als 5 Wörtern sind verboten. Inhaltlich gelten die gleichen Anforderungen wie beim PackMe Projekt (siehe Woche 3).

## Technische Anforderungen an das Projekt

- Die Anwendung ist eine Applikation basierend auf dem Spring Boot Framework. ✓
- Die Anwendung verfügt über ein Frontend basierend auf Svelte. ✓
- Die Anwendung verwendet MongoDB auf Atlas für die Persistenz. ✓
- Die Anwendung ist auf Azure App Service deployed. ✓
- Die Anwendung enthält Unit Tests (mit JUnit) für die Klassen der Business-Logik, mindestens ein Test verwendet Mockito sinnvoll. ✗ noch machen!
- Die Anwendung enthält JUnit Integrationstests für möglichst alle Endpoints basierend auf SpringBootTest und MockMvc. ✗ noch machen!
- Die Anwendung enthält separate Ordner für Modelle, Controller, Repository und Services. ✓
- Die Anwendung verfügt über mindestens zwei authentifizierte Benutzerrollen mit unterschiedlichen Berechtigungen. ✓
- Die Anwendung verfügt über einen Login basierend auf Auth0. Gewisse Teile der Applikation sind öffentlich, andere Teile können nur mit einem gültigen Login erreicht werden. ✓
- Die Anwendung greift via Backend auf mindestens ein Drittsystem zu<sup>1</sup>. ✗ noch machen!
- Das Datenmodell enthält mindestens drei Entitätstypen mit Beziehungen. Die Entitäten werden in der DB gespeichert. Mindestens eine Entität durchläuft mehrere Zustände. ✓
- Die Anwendung hat Endpoints für CRUD-Operation und Endpoints für Services. ✓

## Anforderungen an das Vorgehen und an die verwendeten Tools

- Das Deployment auf Azure App Service erfolgt mit Dockerfile und GitHub Action. Erfolgreiche Logs der Deployments liegen im GitHub-Repository vor. ✓
- Für das Backend liegen Reports der JUnit Tests sowie der Testabdeckung vor. ✗
- Die Testabdeckung wird mittels JaCoCo gemessen und ist möglichst hoch. ✓
- Für die Anwendung wird mit Postman ein Mock-Backend erstellt. ✗ noch machen!
- Die API (Endpoints) ist mit Postman dokumentiert und veröffentlicht. ✓
- Die Anwendung wird auf einem privaten GitHub-Repository entwickelt. ✓
- Anforderungen für den Umgang mit GitHub:
  - In GitHub wird ein SCRUM-Board erstellt und verwendet. ✓
  - Die Umsetzung erfolgt in mehreren Sprints. Die Sprints sind in GitHub als Iterations abgebildet. Die Sprints haben eine sinnvolle Dauer. noch machen!
  - Arbeitspakete sind in GitHub als Issues dokumentiert. Issues haben eine Beschreibung und überprüfbare Anforderungen. Es werden mindestens drei unterschiedliche Labels verwendet. Issues sind einem Sprint oder dem Backlog zugeordnet. Issues durchlaufen im SCRUM-Board die Spalten Ready, In Progress und Done. Abgeschlossene Issues haben den Zustand Closed. noch machen!

---

<sup>1</sup> Die Verwendung eines Iframes oder Widget genügt nicht. Auth0 gilt nicht als Drittsystem.

## Optionale Anforderungen

Durch das Umsetzen der folgenden Anforderungen können die Erwartungen übertroffen werden:

- Codeanalyse mit SonarQube. ✗
- Komplexes Datenmodell. ✓
- Komplexes Frontend. ✓
- Komplexe Benutzerverwaltung. ✓
- Komplexe Abfragen auf der Datenbank. ✓
- Detaillierte Dokumentation auf GitHub (Issues etc.). ✓
- Es werden mehrere Branches sinnvoll verwendet. ✗
- End-to-End Tests der Anwendung. **noch machen!**
- Weitere Erweiterungen können direkt mit den Dozierenden vereinbart werden.

## Anforderungen an die schriftliche Dokumentation

Die Dokumentation befindet sich im GitHub-Repo im Ordner /doc. Die Dokumentation besteht aus fünf Dokumenten im MD-Format – eine Datei pro Kapitel. Bilder sind im Ordner /doc/figures abgelegt. Die Bilder sind in die MD-Dateien eingebunden. Die Kapitel sowie der zugehörige Inhalt sind unten aufgeführt.

- **Einleitung:**
  - Finale Version von Explore-, Create- und Evaluate-Board ✓
  - Erkenntnisse aus dem Pitch (bezogen auf Projekthalt)
- **Anforderungen:**
  - Use-Case Diagramm und Use-Case Beschreibung ✓
  - Fachliches Datenmodell (ER-Modell) mit Erläuterungen ✓
  - Prozessmodell (BPMN-Diagramm) mit Erläuterungen ✓
  - Mockup oder Skizze des UIs ✓
- **Implementation:**
  - Beschreibung des Frontends mit Screenshots der fertigen Applikation ✓
  - Klassendiagramm mit dem technischen Datenmodell (DTOs, DAOs) und deren Verwendungszweck ✓
  - Aufgaben und Funktionen eingebundener Drittsysteme ✗
- **Testing:**
  - Beschreibung der Modul- und Integrationstests ✗ **noch machen!**
  - Beschreibung der End-to-End Tests (falls vorhanden) ✗ **noch machen!**
  - Erkenntnisse aus der Analyse mit SonarQube (falls vorhanden) ✗
- **Fazit**
  - Stand der Implementation, Diskussion der Ergebnisse ✓
  - Nächste Schritte ✓
  - Persönliches Fazit ✓

# Abgabe der schriftlichen Arbeit und der Implementation

Am Semesterende müssen die Dozierenden

- als Collaborators des GitHub-Repos (Repo->Settings->Collaborators) **und**
- als Collaborators beim GitHub-Project (Project->Settings->Manage Access)

hinzugefügt werden (David: DavidZhaw, Adrian: mosazhaw, Max: mmeisterhans, Benjamin: bkuehnis)

Über Moodle müssen die folgenden Dokumente abgegeben werden:

- Zip-Ordner des GitHub-Repos (Download von GitHub)
- Zip-Ordner mit dem Namen jacoco.zip, enthält das Verzeichnis «target\site\jacoco» mit den generierten Reports der Testabdeckung
- Zip-Ordner mit dem Namen surefire-reports.zip, enthält das Verzeichnis «target\surefire-reports» mit einer \*.txt-Datei pro Testklasse
- URL der deployten Applikation
- URL des Mock-Backends
- URL der API-Dokumentation (Endpoints)
- URL des GitHub-Repos
- URL von SonarQube (falls vorhanden)
- Liste der umgesetzten optionalen Anforderungen

## Anforderungen an die Präsentation

Die Abschlusspräsentation dauert 10 Minuten. Anschliessend werden die Studierenden durch die Dozierenden während 10 Minuten zur Arbeit befragt. Folgende Elemente werden während der Präsentation erwartet:

- Ziel der Anwendung: Problem, angesprochene Nutzergruppe, Lösungsansatz
- Demonstration der Anwendung (Live, kein Video, mindestens zwei Rollen)
- Fazit (Projektstand und persönliches Fazit)