

## Software Engineering Projekt

## eCourse

# Projekthandbuch

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

**3819525, 4679471, 5247876**

**6499003, 6504782, 7182188**

**7750470, 8538336, 9654562**

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>Tabellenverzeichnis</b>	<b>1</b>
<b>1 Ansprechpartner und Organisationsstruktur</b>	<b>2</b>
<b>2 Ausgangssituation, Projektkontext und Projektziele</b>	<b>3</b>
2.1 Ausgangssituation und Problembeschreibung . . . . .	3
2.2 zeitlicher Projektkontext . . . . .	3
2.3 sachlicher Projektkontext . . . . .	3
2.4 Projektziele . . . . .	3
<b>3 Ablagestruktur (Dokumente)</b>	<b>5</b>
<b>4 Projektkommunikationsplan</b>	<b>6</b>
<b>5 Projektplan</b>	<b>7</b>
5.1 Projektverknüpfungen . . . . .	7
5.2 Eskalationspfade . . . . .	7
5.3 Projektablauf . . . . .	7
5.4 Meilensteinplan . . . . .	8
<b>6 Risikomanagement</b>	<b>10</b>
<b>7 Projektabschluss und Bewertung des Projekterfolges</b>	<b>12</b>
7.1 Erfolgsmessung . . . . .	12
7.2 Reflektion/Lessons Learned . . . . .	12
7.3 Planung Nachprojektphase/Restaufgaben . . . . .	12
<b>8 Wartungsphasen</b>	<b>13</b>

# Abbildungsverzeichnis

1.1	Organigramm für das Projekt eCourse . . . . .	2
5.1	Eskalationspfad im Projekt eCourse . . . . .	7
5.2	Scrum Prozess . . . . .	9

# Tabellenverzeichnis

4.1	Projektkommunikationsplan . . . . .	6
6.1	Risiken im Projekt eCourse . . . . .	11

# 1 Ansprechpartner und Organisationsstruktur

Die Zuständigkeiten und die allgemeine personelle Organisationsstruktur sind dem Organigramm in Abbildung 1.1 zu entnehmen.

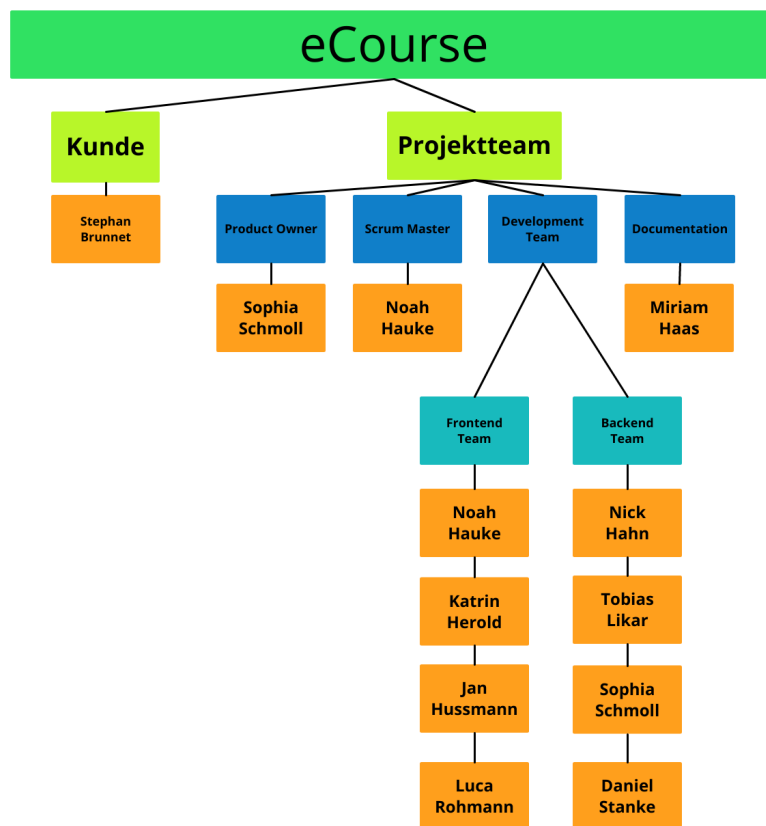


Abbildung 1.1: Organigramm für das Projekt eCourse

Alle externe Anfragen an das Projektteam gehen über den Projektansprechpartner. Diese Rolle übernimmt Miriam Haas. Als Kontaktart wird eine E-Mail an diese Adresse bevorzugt.

## **2 Ausgangssituation, Projektkontext und Projektziele**

### **2.1 Ausgangssituation und Problembeschreibung**

Das Projekt erstellt eine neue Software. Es fließen keine Vorprojekte ein und das Projekt kann auch nicht auf eine bestehende Struktur aufbauen.

Dadurch muss zu Beginn des Projektes eine neue Architektur aufgebaut werden.

### **2.2 zeitlicher Projektkontext**

Das Projekt startet am 18.03.2021 und endet am 20.05.2021.

### **2.3 sachlicher Projektkontext**

Gefordert ist eine datenbankgestützte Webapplikation. Ebenfalls in die Anwendung soll eine Rechtesteuerung integriert sein.

### **2.4 Projektziele**

- Die Programmierung einer lauffähigen Software
- Die Erstellung der erforderlichen Projektdokumente
  - Lastenheft
  - Pflichtenheft
  - Architekturdokument
  - Codierungsrichtlinien und Konfigurationsmanagement

- Planung von Betrieb und Wartung
- Dokumentation und Benutzeranleitung
- Getestete Software mit einem Nachweis der Testabdeckung

# 3 Ablagestruktur (Dokumente)

Alle zum Projekt zugehörigen Dokumente werden mit eindeutiger Benennung unter Einhaltung der vorgegebenen Ablagestruktur im GitHub-Repository abgelegt.

Alle zum Projekt gehörenden Dokumente befinden sich im Ordner „SWE\_Doc“. Innerhalb dieses Ordners ist die Dokumentationsstruktur nochmals aufgespalten. Dokumente, die das Frontend betreffen, finden sich im Ordner „Frontend“, Dokumente, die das Backend betreffen, befinden sich im Ordner „Backend“. Alle weiteren Dokumente befinden sich in separaten Ordnern, die den Namen des entsprechenden Dokuments tragen.



## 4 Projektkommunikationsplan

Zur internen Kommunikation innerhalb des Projektes wird die Plattform Discord verwendet. Dort wurde ein eigens für dieses Projekt bestimmter Server erstellt, auf welchem die Kommunikation stattfindet.

Alle Besprechungen, die das gesamte Development Team betreffen werden im Kanal „General“ abgehalten. Dazu zählt auch schriftliche Kommunikation via Chat.

Sind einzelne Inhalte nur für eine bestimmte Gruppe interessant, kann auf die anderen, entsprechend gekennzeichneten Kanäle ausgewichen werden.

In Tabelle 4.1 sind alle regelmäßigen Projektmeetings aufgeführt. Zu diesen Meetings wird eine vollzählige Anwesenheit des Development Teams erwartet.

Bezeichnung des Meetings	Inhalte	Häufigkeit und Dauer
Daily Scrum	Besprechung was getan wurde und was noch getan wird	jeden Montag von 9.30 bis 10.00 und jeden Mittwoch von 15.00 bis 15.30
Sprint Review	Besprechung der im Sprint geleisteten Arbeit	jeden Donnerstag direkt nach der SWE-Vorlesung
Sprint Retrospektive	Besprechung des Ablaufs des letzten Sprints	jeden Donnerstag direkt nach dem Sprint Review
Sprint Planning	Planung des nächsten Sprints	jeden Donnerstag direkt nach der Sprint Retrospektive

Tabelle 4.1: Projektkommunikationsplan

# 5 Projektplan

## 5.1 Projektverknüpfungen

Das Projektteam hat sich dazu entschlossen das Projekt über GitHub zu verwalten. Dort sind in einem Kanban-Board alle Tasks des aktuellen Sprints und der abgeschlossenen Sprints zu finden. Alle Tasks die eine konkrete Arbeit erfordern sind einem Issue zugeordnet. Dieses Issue wird dann mit dem, den Task erfüllenden Pull-Request verbunden. Im Issue ist auch das zuständige Projektteam für diese Aufgabe gekennzeichnet. Für eine Übersicht der Projektverknüpfungen wird daher auf das GitHub-Repository und darin auf die Issue-Liste und das Kanban-Board verwiesen.

## 5.2 Eskalationspfade

Da dieses Projekt in keine gewöhnliche unternehmensinterne hierarchische Organisation eingebettet ist, können keine weitreichenden Eskalationspläne erstellt werden. In Abbildung 5.1 ist dennoch ein kurzer graphischer Weg zum Umgang mit Problemen dargestellt. Dieser zeigt einen Weg auf, über den es möglich ist teamintern mit Problemen umzugehen.

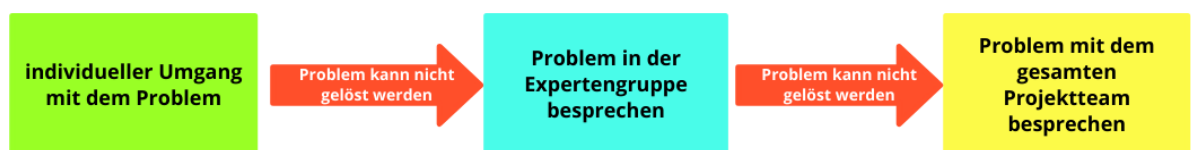


Abbildung 5.1: Eskalationspfad im Projekt eCourse

## 5.3 Projektablauf

Die Arbeit am Projekt startet am 18.03.2021 und endet am 20.05.2021. Teamintern wurde festgelegt, dass die Projektorganisation agil mittels Scrum erfolgen soll.

Dies sieht vor, dass das Projekt in einzelne Entwicklungsstufen unterteilt ist, welche Iterationen genannt werden.

Zu Beginn wurden für den Scrum Prozess die drei Scrum-Rollen im Team verteilt. Die genaue Aufteilung der Rollen ist der Abbildung 1.1 zu entnehmen.

In Abbildung 1.1 ist ebenfalls zu sehen, dass das Development Team nochmals aufgeteilt ist, in ein Frontend und ein Backend Team. Diese Maßnahme wurde durchgeführt, um für die beiden großen Themengebiete jeweils eine Expertengruppe zu bilden. Dadurch muss nicht jeder Entwickler mit allen Projektdetails vertraut sein, sondern nur mit seinem spezielleren Themengebiet. Ebenfalls wird dadurch die Verteilung von Tasks einfacher, da die Tasks dann in erster Linie dem Expertenteam zugeordnet wird und dort dann intern die Tasks noch weiter aufgeteilt bzw. verteilt werden können.

Die Dokumentation des Projektes wurde separat an ein Teammitglied ausgelagert. Dies gibt den Development-Team die Möglichkeit sich vollständig um die Implementierung neuer Features zu kümmern.

Der Scrum-Prozess wird im Team nach folgenden Regeln gestaltet:

- Jeder Sprint dauert genau eine Woche. Die Sprintlänge ergibt sich daraus, dass das Projekt sehr dynamisch ist und durch die beiden Expertenteams ein hohes Maß an Kommunikation erforderlich ist. Außerdem erlaubt die kurze Sprintdauer eine hohe Abhängigkeit zwischen den Tasks der beiden Gruppen.
- Daily Scrums finden jeden Montag und Mittwoch statt. Eigentlich werden Daily Scrums an jedem Arbeitstag durchgeführt, allerdings ist es in diesem Projekt so, dass die einzelnen Teammitglieder nicht zu 100% an diesem Projekt arbeiten. Dadurch werden nicht an jedem Arbeitstag neue Projektergebnisse generiert, über die gesprochen werden könnte.
- Der iterative Scrum-Prozess ist in Abbildung 5.2 kurz dargestellt.

### 5.4 Meilensteinplan

Wie im vorherigen Unterkapitel 5.3 bereits erläutert, wird dieses Projekt agil entwickelt. Dadurch ergeben sich im Projekt keine Meilensteine.

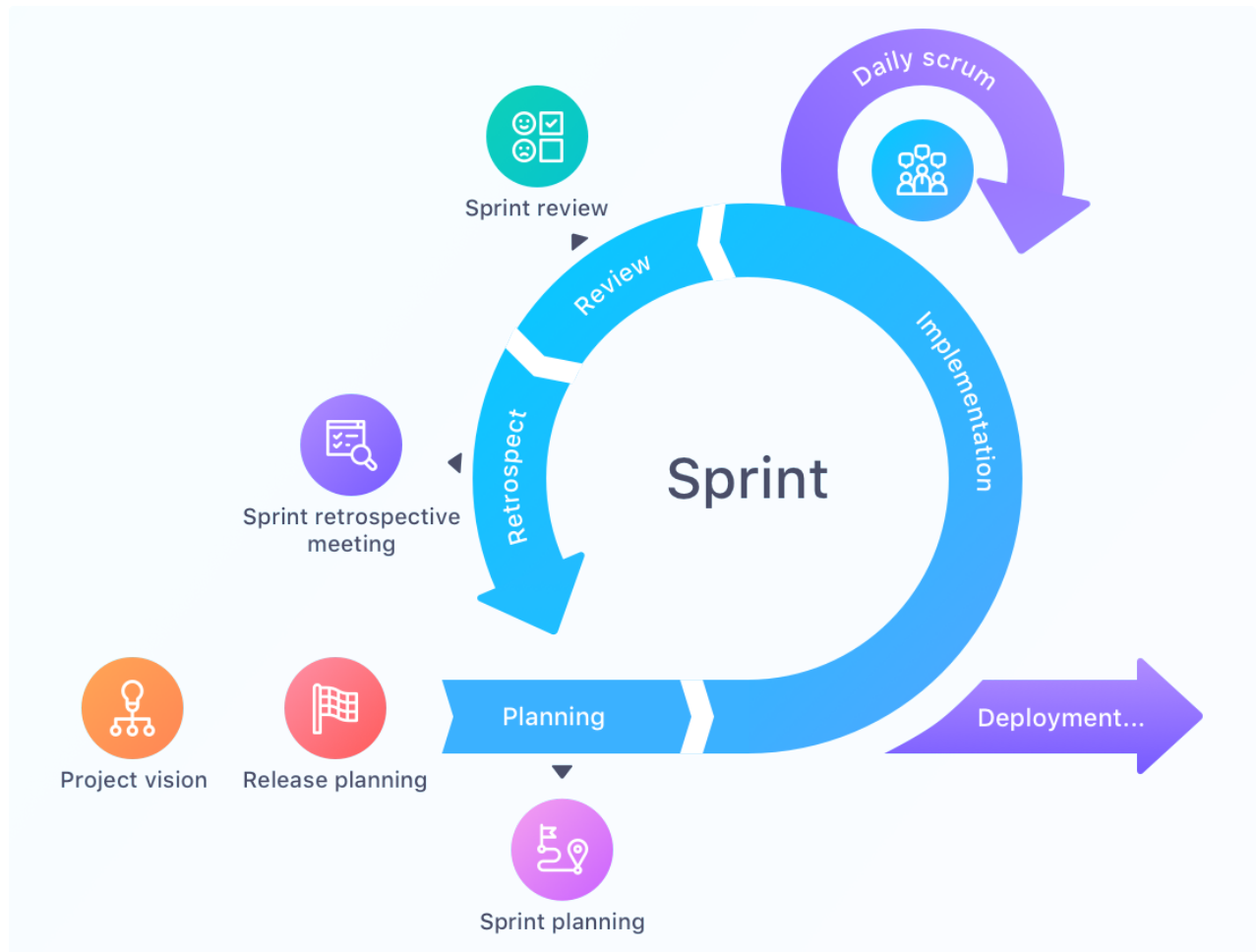


Abbildung 5.2: Scrum Prozess

## 6 Risikomanagement

Zur Ermittlung potenzieller Risiken werden zu Projektbeginn mit dem ganzen Team die relevanten Risiken des Projekts identifiziert. Zur Einstufung der Risiken wurde im Team die Eintrittswahrscheinlichkeit des Risikos sowie die dazugehörige Auswirkung geschätzt. Dies führte zu den Ergebnissen in Tabelle 6.1. Im Zuge der Besprechung der Risiken wurden auch Möglichkeiten zur Vermeidung dieser Risiken diskutiert. Diese finden sich ebenfalls in Tabelle 6.1. Um die Risiken im Projekt klein zu halten wird über die gesamte Projektdauer versucht die in Tabelle 6.1 aufgeführten Risikovermeidungsstrategien umzusetzen.

## 6 Risikomanagement

Ursache	Ausmaß	Risiko	Auswirkung	Chance	Vermeidung
Anforderungen	Produkt	Ungenügende Anforderungsanalyse	ernst	hoch	
		Im Produkteinsatz stellt sich heraus dass mit wachsender Benutzerzahl Einbrüche hinsichtlich der Performance einhergehen. Als Folge können nicht alle Funktionen Effektiv und effizient genutzt werden . Die Nutzvorteile sind eingeschränkt.	tolerierbar	hoch	
Extern	Produkt	Höhere Gewalt beeinflusst das Projekt	tolerierbar	gering	
Kommunikation	Team	Probleme werden nicht offen angesprochen	ernst	mittel	Offenheit im Team fördern
		Probleme bei der Programmierung werden zu lange nicht angesprochen	ernst	mittel	Offenheit im Team fördern
		Fehlende Dokumente	ernst	mittel	Dokumente als Tasks in Sprints aufnehmen
Management	Produkt	unrealistische Zeitplanung	ernst	hoch	Puffer einplanen
		fehlende Dokumentation	ernst	hoch	Team zur Dokumentation anhalten
		vergessene und nachträglich hinzugefügte Tasks	ernst	mittel	Daily Scrums
		zu viel Kommunikation	gering	gering	Zeitmanagement in Meetings
	Team	schlechter Informationsfluss	ernst	gering	Offenheit im Team fördern
		Unklare Befugnisse: Es ist unklar wer die Autorität hat ein Projektziel durchzusetzen	tolerierbar	gering	Kommunikation, DailyScrums
Produktumfang	Produkt	Komplexität wird unterschätzt	ernst	mittel	
		Im Zeitverlauf über die Entwicklungsphasen wird keine klare Linie verfolgt sondern es gibt ständig neue und wechselnde Wünsche nach Funktionen; die an die SW gestellten Anforderungen werden kontinuierlich verändert.	niedrig	gering	Kommunikation Einhaltung des Lasten- und Pflichtenheftes
		Nicht umsetzbares Design	ernst	mittel	Kommunikation
Systemarchitektur	Produkt	Ungenügende Spezifikationen von Schnittstellen	ernst	hoch	Anforderungsanalyse
		Komponenten arbeiten nicht zusammen	ernst	mittel	
		Nur teilweise erfahren SW-Entwickler	tolerierbar	hoch	
		Komponenten laufen nicht stabil	ernst	mittel	
		Probleme mit Projektmanagementsystem	ernst	gering	Alternativen bereithalten
	Stakeholder	Sicherheitslücken	ernst	mittel	Saubere Implementierung
		Testumgebung für Integrationstests sind nicht verfügbar	ernst	gering	Frühzeitig erkennen
Team	Produkt	Nur teilweise erfahren SW-Entwickler	ernst	hoch	Pairprogramming
		Mangelhafte Zeitplanung	tolerierbar	mittel	Kommunikation
		Implementierung unnötiger Eigenschaften und keine Entwicklung einer der Anforderung entsprechenden Eigenschaft	mittel	gering	Offenheit im Team fördern
		Nicht genügend Entwickler	ernst	mittel	Scope
		Austausch/ Wegfall von Personal (Exmatrikulation o. ä.)	mittel	gering	
		Einsatz neuer unbekannter Tools	tolerierbar	hoch	Wiki
		Mitarbeiter müssen sich in Tools einarbeiten	tolerierbar	hoch	Kommunikation
		Mangelnde Leistung Engagement	ernst	mittel	Kommunikation
		Negative Einstellung gegenüber des Projektes	ernst	gering	Offenheit im Team fördern
	Team	Geringe Motivation	ernst	mittel	Zeitplan einhalten

Tabelle 6.1: Risiken im Projekt eCourse

# 7 Projektabschluss und Bewertung des Projekterfolges

Im Nachfolgenden wird der Erfolg des Projektes sowie das Verbesserungspotential für kommende Projekte festgehalten.

## 7.1 Erfolgsmessung

Zielerreichung inhaltlich

Gefordert:

- ja...

Ja was wir halt getan haben

## 7.2 Reflektion/Lessons Learned

Team/Zusammenarbeit:

Projektmanagement:

Sonstiges:

## 7.3 Planung Nachprojektphase/Restaufgaben

Da für dieses Projekt eine harte Deadline gesetzt wurde, können von diesem Projektteam keine weiteren Restaufgaben erledigt werden.

Allerdings ist es für ein Folgeprojekt möglich noch weitere Features in die Software zu implementieren.

Hier noch weitere Features für Folgeprojekte einfügen

## 8 Wartungsphasen

Wie bereits in Kapitel 7 erwähnt wurde diesem Projekt eine harte Deadline gesetzt. Dadurch ist es diesem Projektteam nicht möglich die Software auf viele Jahre hin zu warten. Allerdings kann ein Wartungsplan dem Dokument „Planung von Betrieb und Wartung“ entnommen werden.