

Project Charter

Software Engineering 1

Project Charter

Software Engineering 1

Feld	Wert
Projektname	
Version	1.0
Status	Entwurf
Datum	<i>[Datum]</i>
Hochschule / Kurs	<i>[Hochschule]Software Engineering 1</i>

1. Projektübersicht

1.1 Projektzweck

Kurze Beschreibung des Projekts (2-4 Sätze): Was wird entwickelt? Welches Problem löst die Software? Für wen?

Beispiel: Im Rahmen des Moduls Software Engineering 1 wird eine Webanwendung zur Verwaltung von Studierenden-Lerngruppen entwickelt. Die Anwendung soll es Studierenden ermöglichen, Gruppen zu erstellen, beizutreten und Lernmaterialien zu teilen.

1.2 Projekthintergrund

Warum wird dieses Projekt durchgeführt? Welcher Bedarf oder welche Anforderung liegt zugrunde?

2. Ziele und Nicht-Ziele

2.1 Projektziele

Nr.	Ziel	Erfolgskriterien
Z-01	<i>[Ziel 1]</i>	<i>[Kriterium]</i>
Z-02	<i>[Ziel 2]</i>	<i>[Kriterium]</i>
Z-03	<i>[Ziel 3]</i>	<i>[Kriterium]</i>

2.2 Nicht-Ziele (Out of Scope)

Die folgenden Punkte sind **explizit nicht** Teil dieses Projekts:

- *[Was wird bewusst ausgeschlossen?]*
 - *[z. B. Mobile App, Admin-Backend, Mehrsprachigkeit]*
-

3. Stakeholder

Rolle	Name	Verantwortlichkeit
Auftraggeber / Betreuer	<i>[Dozent/in]</i>	Anforderungen, Abnahme, Bewertung
Projektleiter:in	<i>[Name]</i>	Koordination, Planung, Kommunikation
Entwickler:in	<i>[Dino]</i>	Implementierung, Testing
Entwickler:in	<i>[Luca]</i>	Implementierung, Dokumentation
Entwickler:in <i>(weitere)</i>	<i>[Lucas]</i> <i>[Alle]</i>	Implementierung, Architektur <i>[Rolle]</i>

4. Projektteam und Rollen

4.1 Teamstruktur

Bezeichnung	Details
Projektleitung	<i>[Name]</i> (Matrikel: <i>[Nr.]</i>)Schwerpunkt: <i>[z. B. Backend]</i>
Entwicklung 1	<i>[Name]</i> (Matrikel: <i>[Nr.]</i>)Schwerpunkt: <i>[z. B. Frontend]</i>
Entwicklung 2	<i>[Name]</i> (Matrikel: <i>[Nr.]</i>)Schwerpunkt: <i>[z. B. Datenbank]</i>
QA / Testing	<i>[Name]</i> (Matrikel: <i>[Nr.]</i>)Schwerpunkt: <i>[z. B. Testing, CI/CD]</i>

4.2 Kommunikation

Kanal	Zweck	Frequenz
<i>[z. B. Discord]</i>	Team-Kommunikation	taglich
<i>[z. B. GitHub Issues]</i>	Aufgabenverwaltung	kontinuierlich
<i>[z. B. Weekly Meeting]</i>	Fortschrittsbesprechung	wochentlich
<i>[z. B. E-Mail]</i>	Kommunikation mit Betreuer	bei Bedarf

5. Anforderungen (Überblick)

5.1 Funktionale Anforderungen

ID	Anforderung	Priorität
FA-01	[Funktionale Anforderung 1]	Hoch
FA-02	[Funktionale Anforderung 2]	Hoch
FA-03	[Funktionale Anforderung 3]	Mittel
FA-04	[Funktionale Anforderung 4]	Niedrig

5.2 Nicht-funktionale Anforderungen

ID	Anforderung	Kategorie
NFA-01	[z. B. Antwortzeit < 2 Sekunden]	Performance
NFA-02	[z. B. Responsives Design]	Usability
NFA-03	[z. B. HTTPS, Passwort-Hashing]	Sicherheit
NFA-04	[z. B. Deployment via Docker]	Betrieb

6. Technologie-Stack

Bereich	Technologie / Tool
Frontend	[z. B. React, Vue, HTML/CSS] Begründung: [Begründung]
Backend	[z. B. Spring Boot, Node.js, Django] Begründung: [Begründung]
Datenbank	[z. B. PostgreSQL, MySQL, MongoDB] Begründung: [Begründung]
Versionskontrolle	Git / GitHub (Standard, Kollaboration)
Projektmanagement	[z. B. GitHub Projects, Trello] Begründung: [Begründung]
CI/CD	[z. B. GitHub Actions] Begründung: [Begründung]
Kommunikation	[z. B. Discord, Teams] Begründung: [Begründung]

7. Zeitplan und Meilensteine

7.1 Projektphasen

Phase 1 - Planung & Analyse

- Anforderungserhebung
- Technologiewahl
- Project Charter

Phase 2 - Design & Architektur

- Systemarchitektur

- +-- UI/UX Mockups
- +-- Datenbankdesign

Phase 3 - Implementierung

- +-- Sprint 1: [Funktionen]
- +-- Sprint 2: [Funktionen]
- +-- Sprint 3: [Funktionen]

Phase 4 - Testing & QA

- +-- Unit Tests
- +-- Integrationstests
- +-- User Acceptance Tests

Phase 5 - Abgabe & Präsentation

- +-- Dokumentation finalisieren
- +-- Abschlusspräsentation

7.2 Meilensteinplan

Meilenstein	Beschreibung	Datum	Status
M-01	Project Charter abgeschlossen	[Datum]	Abgeschlossen
M-02	Anforderungsanalyse & Architektur	[Datum]	In Bearbeitung
M-03	Prototyp / MVP fertig	[Datum]	Offen
M-04	Feature-Complete	[Datum]	Offen
M-05	Testing abgeschlossen	[Datum]	Offen
M-06	Abgabe & Präsentation	[Datum]	Offen

8. Risikomanagement

ID	Risiko	W/A	Gegenmaßnahme
R-01	Ausfall eines Teammitglieds	M/H	Wissensteilung, Pair Programming
R-02	Technische Komplexität unterschätzt	M/H	Frühzeitige Spikes, Scope-Reduktion
R-03	Anforderungsänderungen	N/M	Klare Change-Request-Prozesse
R-04	Integrationsprobleme	M/M	Frühzeitige Integrationstests
R-05	[Projektspezifisches Risiko]	[W/A]	[Maßnahme]

Legende: W/A = Wahrscheinlichkeit/Auswirkung; H = Hoch, M = Mittel, N = Niedrig

9. Ressourcen und Constraints

9.1 Ressourcen

- **Teamgröße:** *[Anzahl]* Personen
- **Verfügbare Zeit pro Person:** ca. *[x]* Stunden/Woche
- **Gesamtprojektlaufzeit:** *[Startdatum]* - *[Enddatum]*
- **Budget:** kein monetäres Budget (studentisches Projekt)
- **Infrastruktur:** *[z. B. GitHub Free, lokale Entwicklung, Uni-Server]*

9.2 Rahmenbedingungen (Constraints)

- Die Abgabe erfolgt bis **[Datum]**
 - Der Technologie-Stack muss mit der Lehrveranstaltung kompatibel sein
 - Alle Teammitglieder müssen gleichmäßig zum Projekt beitragen (erkennbar in Git-History)
 - *[Weitere Vorgaben des Dozenten]*
-

10. Definition of Done (DoD)

Ein Feature gilt als **fertiggestellt**, wenn:

- Der Code ist implementiert und funktioniert lokal
 - Unit Tests sind geschrieben und bestehen
 - Der Code wurde von mindestens einem anderen Teammitglied reviewed (Pull Request)
 - Die Änderungen sind in den **main**-Branch gemergt
 - Die relevante Dokumentation wurde aktualisiert
 - Das Feature wurde manuell getestet
-

11. Abnahmekriterien

Das Projekt gilt als erfolgreich abgeschlossen, wenn:

1. Alle **Muss**-Anforderungen (Priorität: Hoch) vollständig implementiert sind
 2. Alle Unit- und Integrationstests bestehen
 3. Die Anwendung lauffähig demonstriert werden kann
 4. Die technische Dokumentation vollständig vorliegt
 5. Das Projekt erfolgreich präsentiert wurde
-

12. Unterschriften und Genehmigung

Mit ihrer Unterschrift bestätigen alle Beteiligten, dass sie den Inhalt dieses Project Charters gelesen haben und damit einverstanden sind.

Betreuer/in: _____ **Datum:** _____

Projektleiter/in: _____ **Datum:** _____

Teammitglied: _____ **Datum:** _____

Teammitglied: _____ **Datum:** _____

Teammitglied: _____ **Datum:** _____

Anhang

A. Glossar

Begriff	Definition
MVP	Minimum Viable Product - minimale, lauffähige Version des Produkts
Spike	Zeitlich begrenzter Forschungs- / Lernaufwand zur Risikoreduktion
DoD	Definition of Done - Kriterien, wann ein Feature als abgeschlossen gilt
<i>[Begriff]</i>	<i>[Definition]</i>

B. Referenzen

- Vorlesungsunterlagen Software Engineering 1, *[Hochschule]*, *[Jahr]*
- *[weitere Quellen, Literatur, Standards]*

C. Änderungshistorie

Version	Datum	Autor	Änderung
1.0	<i>[Datum]</i>	<i>[Name]</i>	Initiale Erstellung

Dieses Dokument wurde im Rahmen des Moduls Software Engineering 1 erstellt.