

## HAUSAUFGABE NR. 1

### Zeitschätzung, Zeitplanung und Teamarbeit

16. November 2020  
Softwaretechnik 2020/2021

#### Individuelle Beiträge der Teammitglieder:

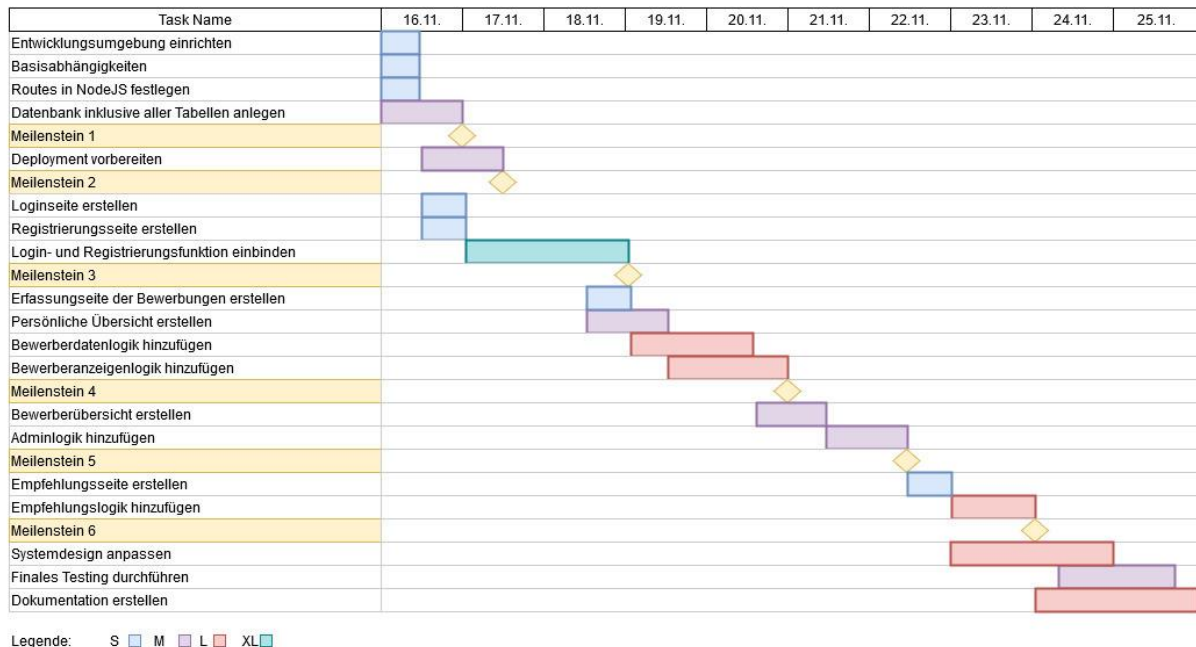
Teammitglied	Beitrag
Yassine Hammou (YH)	<ul style="list-style-type: none"><li>• Datenzulieferung Risikofaktoren und Prozessplan</li><li>• Ausformulierung des Systementwurfs</li></ul>
Cedric Meyer-Piening (CM)	<ul style="list-style-type: none"><li>• Initiale Strukturierung</li><li>• Initiale Strukturierung</li><li>• Datenzulieferung Prozessplan</li><li>• Textproduktion Issues auf GitHub</li><li>• Erstellung des Gantt-Diagramms</li></ul>
Niklas Metje (NM)	<ul style="list-style-type: none"><li>• Datenzulieferung Risikofaktoren und Prozessplan</li><li>• Initiale Strukturierung</li></ul>
Caspar Laurenz Einhorn (CE)	<ul style="list-style-type: none"><li>• Initiale Strukturierung</li><li>• Datenzulieferung Prozessplan</li></ul>
Max Weßling (MW)	<ul style="list-style-type: none"><li>• Initiale Strukturierung</li><li>• Datenzulieferung Risikofaktoren und Prozessplan</li><li>• Textproduktion Zeitplan, Prozessplan und Risikofaktoren</li></ul>

## Inhaltsverzeichnis

<b>1. Zeitplan .....</b>	<b>3</b>
<b>2. Prozessplan .....</b>	<b>4</b>
Teaminterne Kommunikation .....	4
Aufgabenverwaltung .....	4
Kollaboration .....	5
Qualitätssicherung .....	5
Technologie-Stack .....	5
<b>3. Risikofaktoren .....</b>	<b>5</b>

## 1. Zeitplan

Das folgende Gantt-Diagramm gibt den Zeitplan für das Entwicklungsvorhaben des Teams wieder.



Das Team hat sich zur Zeiteinschätzung der einzelnen Aufgaben der Kleidungsgrößen von S (zwei Stunden) bis XL (ein Tag und zwei Stunden) bedient, wobei jeder Größensprung einem Zeitaufwand von zwei Stunden entspricht. Mit dem Erstellen der Login-Seite im Frontend als Referenz für eine Aufgabe der Größe S hat das Team die anderen Einschätzungen bei einem Planning Poker diskutiert. Hierfür hat jedes Teammitglied seine Einschätzung des Aufwandes jeder Aufgabe abgegeben. Auf Basis dieser Einschätzungen wurde im Team der geschätzte Aufwand jeder Aufgabe diskutiert und eine Einigung erzielt. Das Ergebnis dieses Planning Pokers findet sich in den farbigen Markierungen der Aufgaben im Gantt-Diagramm wieder.

Aufgabe	Zuständig
Entwicklungsumgebung einrichten	NM
Basisabhängigkeiten	NM
Routes in NodeJS festlegen	NM
Datenbank inklusive aller Tabellen anlegen	NM
Deployment vorbereiten	NM
Loginseite erstellen	CM
Registrierungsseite erstellen	CE
Login- und Registrierungsfunktion einbinden	YH
Erfassungsseite der Bewerbungen erstellen	CM
Persönliche Übersicht erstellen	CM
Bewerberdatenlogik hinzufügen	MW
Bewerberanzeigenlogik erstellen	YH
Bewerberübersicht erstellen	YH
Adminlogik hinzufügen	NM
Empfehlungsseite erstellen	MW
Empfehlungslogik hinzufügen	MW

---

Systemdesign anpassen	CE
Finales Testing durchführen	YH
Dokumentation erstellen	MW

Eine weitere Übersicht der Aufgaben findet sich auf Kanban-Boards in GitHub unter [diesem Link](#). Jede eingetragene Aufgabe erhält zusätzlich eine Information über den geschätzten Zeitaufwand der Aufgabe sowie über die Person, die mit der Bearbeitung der Aufgabe betraut wurde.

Insgesamt sollen während der Umsetzung des Projekts sechs Meilensteine erreicht werden.

Meilenstein Nr.	Ergebnis
1	Fertig initialisiertes NodeJS sowie eine eingerichtete PostgreSQL-Datenbank
2	Abgeschlossene Vorbereitung für ein plattformunabhängiges Deployment
3	Fertiggestelltes Login-System
4	Fertiggestellter Nutzerbereich der Bewerberrolle
5	Fertiggestellter Nutzerbereich der Mitarbeiterrolle
6	Fertiggestellte Seite zum Einreichen von Empfehlungsschreiben

## 2. Prozessplan

### Teaminterne Kommunikation

Für die Kommunikation innerhalb des Teams hat das Team feste Kommunikationswege vereinbart. So findet drei mal pro Woche ein Zoom-Meeting statt, an dem alle Teammitglieder teilnehmen. Diese Meetings dienen in erster Linie dazu, gemeinsam den Projektfortschritt zu verfolgen und gemeinsam projektbezogene Entscheidungen zu treffen. Außerdem bieten die Meetings eine Plattform dafür, Probleme jeglicher Art im gesamten Team zu diskutieren.

Neben diesen drei festen Treffen pro Woche werden außerdem Pair-Programming-Sessions, in denen sich zwei Teammitglieder gemeinsam um bei der Entwicklung auftretende Probleme kümmern, nach Bedarf angesetzt.

Für die Kommunikation zwischen den Zoom-Meetings wird der private Slack-Channel des Teams genutzt. Von der Nutzung von WhatsApp für projektbezogene Kommunikation sieht das Team ab.

### Aufgabenverwaltung

Um die Zusammenarbeit im Team zu koordinieren verwendet das Team GitHub zur Aufgabenverwaltung. Die zu erledigenden Aufgaben werden in GitHub angelegt und in ein Kanban-Board einsortiert. Dort werden die Aufgaben zur Erledigung einem oder mehreren Teammitgliedern zugewiesen und auf dem Kanban-Board verschoben, sobald sich der Status der Aufgabe (beispielsweise in *erledigt*) ändert. Die Aufgabenzuweisung wird hierbei so gestaltet, dass jede Datei zu einem Zeitpunkt immer nur von einem Teammitglied bearbeitet wird.

## Kollaboration

Für das gemeinsame Arbeiten am Quellcode des Projektes verwendet das Team GitHub als Repository in Verbindung mit dem Software-Versionierungssystem Git. Jedes Teammitglied setzt Änderungen am System in einem eigenen Branch um und gliedert diese Änderungen erst in den master-Branch ein, sobald diese lauffähig sind. Das gemeinsame Bearbeiten von Textdokumenten, Präsentationen oder Tabellenkalkulationen findet über Google Docs statt. Dabei entstandene Dokumente werden in einem gemeinsam genutzten Google Drive-Ordner abgelegt.

## Qualitätssicherung

Um eine hohe Qualität des Codes sicherzustellen, wird jede Änderung des Quellcodes vor dem Mergen in den Master-Branch des Projektes von mindestens zwei Teammitgliedern gegengeprüft. Zuerst pusht das Teammitglied, das die Änderung am Code vorgenommen hat, diese Änderungen erst, sobald der Code als lauffähig angesehen wird. Anschließend wird ein Pull Request erstellt und die Änderungen werden von einem weiteren Teammitglied geprüft. Erweist sich der Code als lauffähig und sind alle Tests bestanden, so kann der Code in den Master-Branch eingegliedert werden.

## Technologie-Stack

Das Team setzt das System mit NodeJS für das Backend, PostgreSQL als Datenbankmanagementsystem sowie HTML, CSS und JavaScript für das Frontend um. Die empfohlene Entwicklungsumgebung für dieses Projekt ist Visual Studio Code, die Wahl obliegt jedoch letztendlich dem einzelnen Teammitglied.

## 3. Risikofaktoren

Das Team hat vier wesentliche Risikofaktoren identifiziert, die, sofern sie unbeobachtet blieben, die Qualität des Produktes negativ beeinflussen könnten. Auf alle vier Risikofaktoren wird im Folgenden näher eingegangen.

Der **sehr heterogene Wissensstand** innerhalb des Teams könnte hinderlich für eine ausgeglichene Aufgabenverteilung und somit für die erfolgreiche Zusammenarbeit im Team sein. Da im Team lediglich ein Mitglied über fundierte Erfahrung mit dem von uns im Backend verwendeten Framework NodeJS aufweisen kann, wird es hier zu einer Herausforderung, die bei der Backend-Programmierung anfallenden Aufgaben unter den Teammitgliedern aufzuteilen und eine angemessene Aufgabenverteilung sicherzustellen. Um dieser Herausforderung zu begegnen, bewertet das Team die im Backend anfallenden Aufgaben hinsichtlich ihrer Komplexität und verteilt sie so, dass Mitglieder ohne Erfahrung mit NodeJS Aufgaben übernehmen, für die eine Einarbeitung in der knapp bemessenen Zeit bis zur Abgabefrist möglich ist. Darüber hinaus stellen die im Prozessplan bereits vorgestellten Pair-Programming-Sessions eine Möglichkeit dar, bei der Backend-Programmierung von der Erfahrung anderer Teammitglieder zu profitieren und besonders komplexe Probleme gemeinsam anzugehen.

Ein weiterer Risikofaktor für das Gelingen des Projektes stellt die **teaminterne Kommunikation** dar. Gerade nicht stattfindende Kommunikation kann dazu führen, dass Aufgaben falsch, redundant oder gar nicht erledigt werden. Insbesondere das simultane Bearbeiten derselben Dateien aufgrund

von mangelhafter Kommunikation kann durch entstehende Merge-Konflikte für Schwierigkeiten beim Einhalten des Zeitplanes sorgen. Um dem zu begegnen, stützt das Team sich auf regelmäßige Zoom-Meetings sowie auf einen Slack-Channel als Möglichkeit für unkomplizierten und direkten Austausch. Darüber hinaus wird der aktuelle Stand aller Aufgaben auf GitHub mittels Kanban-Boards transparent nachverfolgt. Das Zuweisen von Aufgaben in GitHub sorgt zudem dafür, dass redundante Arbeiten zweier Personen an derselben Aufgabe vermieden werden.

Abgesehen von organisatorischen Hindernissen gilt es, Risiken technischer Natur ebenfalls zu erkennen und zu minimieren. Ein wesentlicher Punkt zur Vermeidung technischer Risiken stellt der **Schutz des Systems vor unberechtigten Zugriffen** dar. Alle Nutzer des Systems sollen nur Zugriff auf die Daten erhalten, die für sie bestimmt sind. Dies gilt insbesondere für die Bewerber: sie sollen nicht in der Lage sein, Daten anderer Bewerber einzusehen oder gar Entscheidungen über die Zulassungen einzutragen. Dies vermeidet das Team, indem es eine konsequente und nutzerbezogene Rechtevergabe umsetzt.

Des Weiteren ist sicherzustellen, dass **lokale Datenverluste** (bspw. eine ausfallende Festplatte) nicht zu großen Hindernissen für das Einhalten des Zeitplans werden. Das Team trägt hierfür Sorge, indem jedes Mitglied seinen Projektfortschritt am Quellcode des Systems in regelmäßigen und kurzen Abständen in das GitHub-Repository pusht. So wird vermieden, dass der Fortschritt ganzer Arbeitstage bei einem Datenverlust verloren geht und die Arbeit zu Lasten des Zeitplanes erneut erledigt werden muss.