

# Softwareprojekt 24/25 - Einzel

Institut für Softwaretechnik und Programmiersprachen  
Sabrina Böhm, Valentin Kolb, Matthias Tichy



universität  
**uulm**

## Blatt 3 Vorbereitung & Start Implementierung

Abgabe 22. November 2024 bis 18:00 Uhr

Auf diesem Blatt wird die Implementierung des Spiels *Starfighter Alliance* vorbereitet, sowie erste Anforderungen umgesetzt. Um den Entwicklungsprozess zu versionieren wird weiterhin euer Repository in Gitlab ([gitlab.uni-ulm.de](https://gitlab.uni-ulm.de)) verwendet, welches schon bei Blatt 1 und Blatt 2 verwendet wurde. Für das Projektmanagement soll Gitlab verwendet werden.

### Aufgabe 1 Pflichtenheft

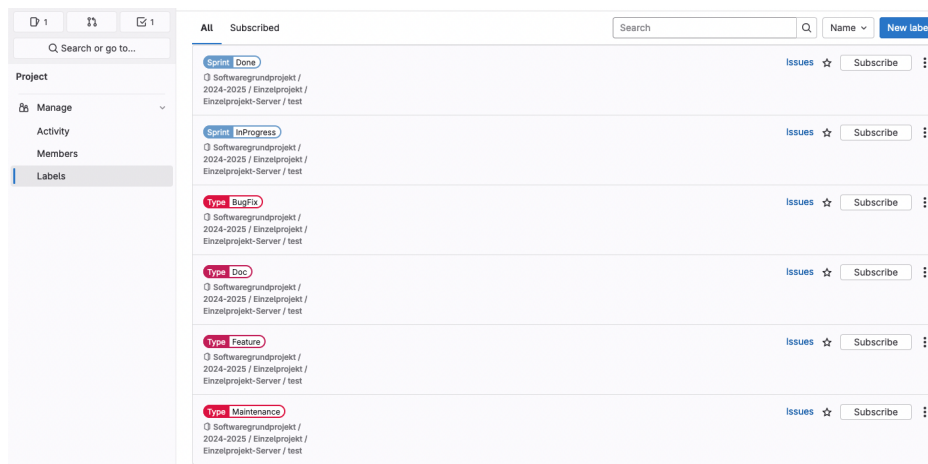
Im Übergang zur Implementierungsphase soll das Pflichtenheft finalisiert werden. Im Pflichtenheft sollen alle bisher erstellten Beschreibungstexte und Diagramme aus den ersten zwei Übungsblättern enthalten sein. Damit das Ergebnis ein in sich geschlossenes, konsistentes Dokument ist, müssen entsprechend noch an der ein oder anderen Stelle eine Einführung (siehe Todos im Template) oder Erklärungen zu den verwendeten Darstellungsformen (Diagramme usw.) verfasst werden, sofern dieses in den vorangegangenen Übungsblättern nicht detailliert umgesetzt wurde.

### Aufgabe 2 Vorbereitung Entwicklungsprozess

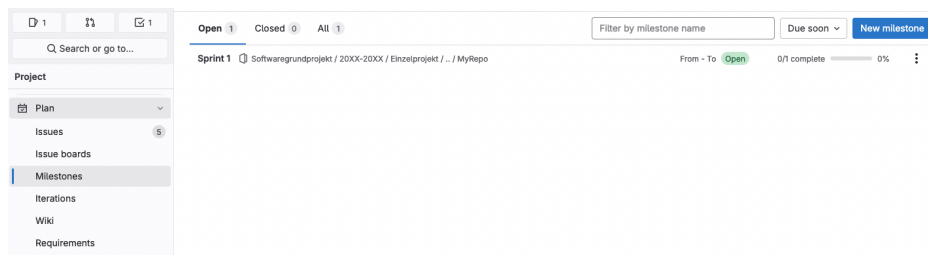
Um den Entwicklungsprozess gut zu strukturieren soll im Gitlab alles dokumentiert werden. Dafür soll euer Repository entsprechend aufgesetzt werden.

Als erstes sollen für die Sprints entsprechende Labels angelegt werden. Dies kann in der linken Seitenleiste unter **Manage > Labels** gemacht werden. Danach sollen die Labels **Sprint::In Progress** und **Sprint::Done** angelegt werden. Mit dem Doppelpunkt werden Sublabels angelegt.

Es steht euch frei beliebige andere Labels hinzuzufügen um euren Entwicklungsprozess zu strukturieren.

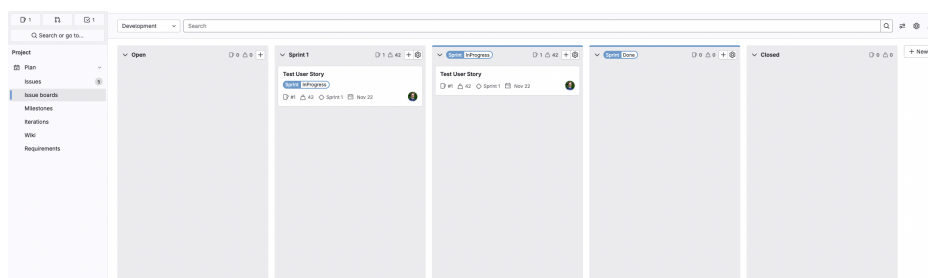


Füge außerdem noch für das erste Blatt unter **Plan > Milestones** einen Meilenstein hinzu.



Sobald die Labels und der erste Meilenstein angelegt wurden, kann das Issue Board erstellt werden. Dies ist unter **Plan > Issue Boards** möglich. Im Development Board (auswählbar über ein Dropdown neben der Search Bar) soll nun zusätzlich zu der Open und Closed Column für den ersten Sprint sowie jedes der zwei Label eine neue Column erstellt werden.

Nun sollte das Board wie folgt aussehen:



Dabei ist das Board folgendermaßen aufgebaut:

- **Open** Alle offenen Issues die keinem Meilenstein zugeordnet sind (Backlog)

- **Sprint 1** Alle Issues die für den aktuellen Sprint ausgewählt sind. Für die weiteren Meilensteine werden hier dann entsprechende Spalten eingefügt.
- **Sprint::InProgress** Alle Issues, an denen gerade gearbeitet wird.
- **Sprint::Done** Alle Issues, die in dem aktuellen Sprint abgeschlossen wurden.
- **Closed** Nachdem ein Sprint abgeschlossen wurde, werden die Issues aus der **Sprint::Done** Spalte in Closed verschoben um sie als endgültig abgeschlossen zu markieren.

Da wir im Softwareprojekt zur Entwicklung **Feature-Branches** nutzen, muss noch ein **dev**-Branch angelegt werden, von dem aus Feature-Branches erstellt werden können. Lege dazu diesen Branch an.

### Aufgabe 3 GitLab Issues

Als nächstes sollen alle auf Blatt 1 definierten Anforderungen durch Issues im Gitlab abgebildet werden. Dies könnt ihr auf deutsch oder englisch machen. Hier erwarten wir mindestens 30 Issues.

In der Softwareprojekt Vorlesung am 07.11. wurde SCRUM vorgestellt. Die Issues sollen in Form von User Stories erstellt werden. Dazu muss verpflichtend das im Gitlab hinterlegte **UserStoryTemplate** genutzt werden (siehe "Choose a template" über dem Feld für die Beschreibung, beim Erstellen eines Issues). Insofern werden im folgenden die Begriffe Issues und User Stories als Synonyme verwendet. Zusätzlich müssen die Issues jetzt noch in Sprints (Gitlab Meilensteine) unterteilt werden. Auf den Blättern 3 bis 5 ist angegeben, welche Funktionalitäten bei der jeweiligen Abgabe implementiert sein muss.

Wie die Issues umgesetzt werden sollen, wird im Folgenden anhand eines Beispiels erklärt.

ID	FA1
TITEL:	Start Screen
BESCHREIBUNG:	Zu Beginn des Spiels soll ein Start Bildschirm angezeigt werden.
BEGRÜNDUNG:	Der spielenden Person soll nach öffnen des Spiel ein Startbildschirm angezeigt werden. Von diesem Bildschirm aus soll die spielende Person mit einem Button das Spiel starten können.
ABHÄNGIGKEITEN:	Button, ...

Diese Anforderung lässt sich in mehrere Issues unterteilen:

- Startbildschirm
- Wechseln vom Start Screen zum Game Screen
- ...

Nun wollen wir beispielhaft zwei Issues im Gitlab anlegen. Technische Details, die die Umsetzung des Issues betreffen, können als Task/ToDo angelegt werden (siehe “Child items”). (Die Issue Beschreibung etc. wurde aus Platzgründen auf den Screenshots entfernt).

The screenshot shows a GitLab issue titled "Start Screen" created by Valentin Kolb. The issue description is "As a player, I want to see a start screen when I open the game, so that I know the game has launched successfully." It has 0 votes and 0 comments. A "Child items" section shows one item: "Button" (type: Task) linked to "Sprint 1". The "Linked items" section is empty. The "Activity" section shows two updates: "Valentin Kolb changed milestone to %Sprint 1 just now" and "Valentin Kolb added 'Type: Feature' label just now". The sidebar on the right shows metadata: 0 Assignees, 0 Epics, 1 Label (Type: Feature), 1 Milestone (Sprint 1), 0 Iterations, 0 Weights, 0 Due dates, 0 Time tracking, 0 Health status, 0 Confidentiality, and 1 Participant.

Da das zweite Issue nur abgeschlossen werden kann, wenn das erste Issue und die Button-Task umgesetzt sind, können diese beiden Aufgaben als **Linked items** hinzugefügt werden. Das **Is blocked by** zeigt die Beziehung an.

The screenshot shows a GitLab issue titled "Start Game" created by Valentin Kolb. The issue description is "As a player, I want to start the game by pressing a button on the start screen, so that I can begin playing when I'm ready." It has 0 votes and 0 comments. The "Child items" section is empty. The "Linked items" section shows two items: "Start Screen #7" (type: Task) and "Button #8" (type: Task), both linked to "Sprint 1". The sidebar on the right shows metadata: 0 Assignees, 0 Epics, 1 Label (Type: Feature), 1 Milestone (Sprint 1), 0 Iterations, 0 Weights, 0 Due dates, 0 Time tracking, 0 Health status, 0 Confidentiality, and 1 Participant.

Da bei beiden Issues der Meilenstein **Sprint 1** verlinkt wurde erscheinen diese nun auch in der entsprechenden Spalte im Board.

## Entwicklungsprozess

Folgende Bestandteile des Entwicklungsprozesses sind verpflichtend und müssen genutzt werden:

- Tracken des Vorschritts von Issues durch Labels
- Verlinken von Issues in Commits

- Meilensteine um die einzelnen Sprints zu dokumentieren
- **Conventional Commits**
- Mergen von Feature-Banches in dev und mergen von dev in main durch entsprechende Merge-Requests mit dem dafür bereitgestelltem Template

**Hinweise:**

Für dieses Blatt sowie Blatt 4 und 5 müssen unter **Issues > Milestones** Meilensteine angelegt werden. Die Issues müssen dann entsprechend einsortiert werden. Es empfiehlt sich dieses Blatt in einen Sprint (einen Meilenstein), Blatt 4 in zwei Sprints (zwei Meilensteine) und Blatt 5 in einen Sprint (einen Meilenstein) aufzuteilen.

Für die meisten IDE's gibt es Conventional Commits Plugins, die diese erzwingen und entsprechend formatieren.

Issues lassen sich wie folgt in Commits referenzieren: `git commit -m "... message ... Ref #8 ..."`

Es dürfen gerne weitere Funktionalitäten genutzt werden, wie z.B. Gitlab Tasks (<https://docs.gitlab.com/ee/user/tasks.html>) zum Unterteilen von Issues, bzw. User Stories in Implementierungsdetails. Gitlab bietet außerdem viele Möglichkeiten den Entwicklungsprozess gut nachzuverfolgen, die wir im Gruppenprojekt auch nutzen werden. Dazu zählen zum Beispiel Time Tracking ([https://docs.gitlab.com/ee/user/project/time\\_tracking.html](https://docs.gitlab.com/ee/user/project/time_tracking.html)) oder spezifischere Labels (<https://docs.gitlab.com/ee/user/project/labels.html>).

**Aufgabe 4 Start Implementierung**

Im Gitlab <https://gitlab.uni-ulm.de/softwaregrundprojekt/2024-2025/einzelprojekt/templates> findet ihr drei verschiedene Templates für die Implementierung des Spiels. Du kannst dir eins davon aussuchen und dich damit vertraut machen oder ein eigenes Projekt aufbauen.

Entscheide dich für ein Template und clone es in dein Repository oder erstelle ein eigenes Projekt, sodass folgendes vorhanden ist:

- Startscreen mit Button um in den Gamescreen zu gelangen
- Gamescreen mit Raumschiff

Nach dem initialen Commit vom Laden des Templates in euer Repository sollt ihr zusätzlich zum Startscreen und Gamescreen noch den **Endscreen** erstellen.

Für die Bearbeitung jedes Issues soll ab jetzt ein neuer **Feature-Branch** erstellt werden. Dies ist auch direkt über das Gitlab Webinterface in einem betrachteten Issue möglich. Das Mergen des jeweiligen Branches in Dev und/oder Main soll über einen in gitlab gestellten Merge Request erfolgen (unter Verwendung des Templates für die Merge Request Beschreibung).

Nachdem ihr den Endscreen implementiert habt, sollt ihr diesen mit geeigneter Commit Message (Conventional Commit mit Referenzieren des Issues für den Endscreen) in euer Repository pushen.