

Softwaretechnik-Praktikum

Vorlesung 1

Adrian Kunz

Dr. Jens Kosiol

Vorlesungsplan

- Einführung (Adrian)
 - Organisatorisches, Projektrollen und Ablauf
 - Git Flow
- Grundlagen und Theorie (Jens)
 - [Software in Teams](#)
 - Agile
 - Scrum
- Technologien (Adrian)
 - Git, Jira
 - Dagger, Retrofit, RxJava
 - JavaFX, fulibFx
 - Mocking

Organisatorisches

Ablauf



6-7 Vorlesungen

Dienstag 12:00 R1332

Freitag 12:00 R-1418

Falls wenige: Labor R1344

Aufnahmen auf [YouTube](#) [1]



16 Wochen

Teamprojekt

Bis 18. August

4 Releases

4 Präsentationen

Wöchentliche Teammeetings
und Hackathon



1

Abschlussturnier

vrsl. 19.-23. August

Voraussetzungen



Programmieren
& Modellieren



Grundbereich A

Eidl
TGI/DiLo/GEt
FGdl
LinAlg



Zeit

Rollenspiel: Kunde



CEO
N. Nolte



Dead Birds
Society



CTO
A. Kunz



Anforderungen



Server

Rollenspiel: Betreuer



Angry Bird
Consulting

- Henrik Eckhardt
- Simon Löser
- Giulio Pazzi
- Tom Reuter
- Sandro Schlusche

Rollenspiel: Teams



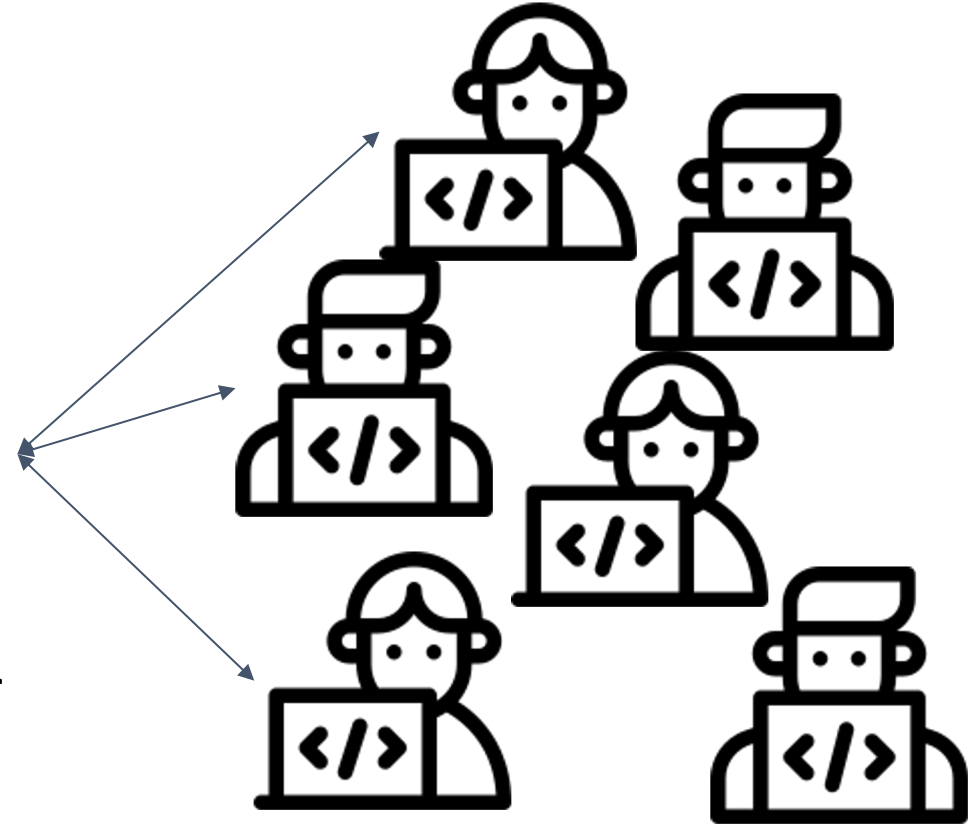
Kunde



Product Owner
"PO"






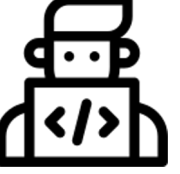
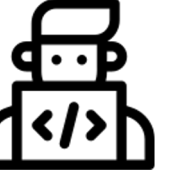



























Scrum Master
"SM"



































Entwickler

Rollenverteilung – Beispiel

Release	Alice	Bob	Charlie	Dani	Ezra	Felix	Gale	Helge
v1								
v2								
v3								
v4								

Rollenverteilung – Beispiel

Release	Alice	Bob	Charlie	Dani	Ezra	Felix	Gale	Helge
v1								
v2								
v3								
v4								

Prüfungsleistung – Führungsrollen



1 Dokumentation



1 Präsentation

Prüfungsleistung – Dauerhaft



Jira









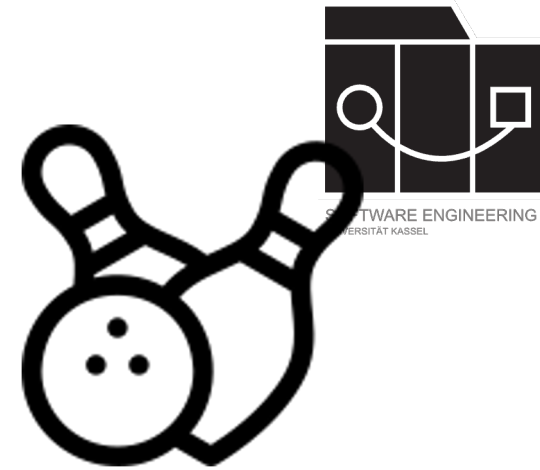
Code



Teamtreffen

Prüfungsleistung

	1x	Dokumentation	25%
	1x	Präsentation	25%
	16/17x	Teamtreffen	25%
	4x	Jira	12,5%
	4x	Code	12,5%
=====			
	25	Noten	100%



Strikes

- STP verlangt **konstante** Leistung über 16 Wochen
 - 8 CP / 240h nach Modulhandbuch
 - 240 h / 16 Wochen = **15 h/Woche**
 - 15 h - 1h Teamtreffen - 4h Hackathon = **10h Einzelarbeit**
- Betreuer schätzen den tatsächlichen Aufwand in Teamtreffen
- **Nicht zufriedenstellend => Strike!**
- **Drei Strikes => Sofortiger Ausschluss mit NB**

Projektrollen



Product Owner (PO)

Allgemeine Aufgaben

- Ansprechpartner für **Kundenkontakte**
- **Anforderungen** erfassen (Requirements Engineering)
- Schreiben und Pflegen von **User Stories**

Weitere Aufgaben im STP

- **Design** festlegen und **Assets** aussuchen
- **Benutzbarkeit** (UX) sicherstellen – von Wireframes bis zur fertigen Anwendung
- Anwendung aus Benutzersicht manuell testen, d.h. **Quality Assurance** (QA)



Scrum Master (SM)

Allgemeine Aufgaben

- Scrum-Meetings moderieren
- Tasks erstellen
- Stories, Tasks und Bug Tickets gerecht/strategisch auf Entwickler verteilen
- Technisches Verständnis über Codebase ("Big Picture")
 - Fragen zur Implementierung beantworten
 - Aufgaben detailliert formulieren
- Fertige Features durch Code Review absegnen und mergen

Weitere Aufgaben im STP

- Gute Programmierpraxis etablieren und prüfen (Code Style, Warnings, Duplikate, Kommentare, etc.)
- Entwicklerdokumentation pflegen (ReadMes, Changelogs, Pull Request Templates, ggf. Wiki, ...)



Entwickler

- Stories und ggf. Tasks/Bugs zeitlich **schätzen**
- Zugewiesene Aufgaben bearbeiten/**implementieren**
- **Tests schreiben** und aktuell halten
- **Pull Requests** erstellen
- Code Review **Kommentare** des SMs **umsetzen**
- Fehler finden (die PO übersehen hat) und **Bug Tickets** erstellen

Ablauf

1. Wireframes/Mockups

Anforderungen

Dead Birds Society Projekt Pioneers

Release 1 - Anforderungen

- User-Verwaltung
 - Account erstellen
 - Einloggen
- Lobby
 - Liste der (Online)-User anzeigen
 - Liste der Spiele anzeigen
 - Ausloggen
- Direktnachrichten
 - Darstellen von alten Nachrichten
 - Senden an andere User
 - Löschen eigener Nachrichten
 - Automatisches Anzeigen von neuen Nachrichten
- Spielverwaltung
 - Spiel erstellen
- Spiellansicht
 - Liste der Teilnehmenden anzeigen
 - Betretten mit Passwort
 - Bereit / nicht Bereit
- Technische Anforderungen
 - Dynamische Aktualisierung aller Informationen anhand von Events
 - Validierung aller Eingaben (siehe <https://pioneers.unika.de/api/v1>)
 - Testabdeckung 50% CO
- Fristen
 - Zeitraum: 25.04.2022 - 20.05.2022
 - Stories in Jira eingetragen bis 25.04.2022 08:00
 - Wireframes eingereicht bis 25.04.2022 08:00
 - Veröffentlichung der Version 1.0 am 20.05.2022
 - Kritische Patches spätestens am 22.05.2022 erlaubt
 - Dokumentation eingereicht bis 24.05.2022 16:00

Folgende optionale Features wurden vom Auftragnehmer vorgeschlagen und in den Leistungskatalog aufgenommen:

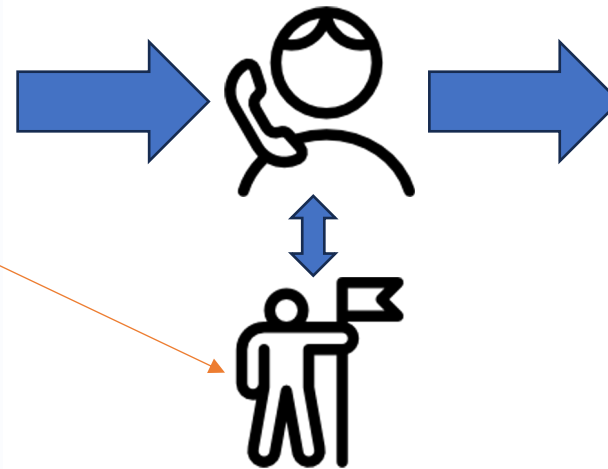
1.

Kassel, den 22.04.2022

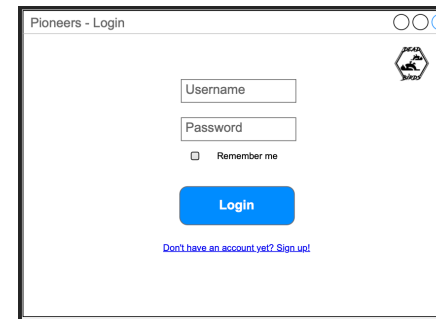
Product Owner Team X

Geschäftsführung Dead Birds Society Technikvorstand Dead Birds Society

Dead Birds Society - Geschäftsführung: Natasscha Nolte
Wilmshäbeler Allee 73, 34123 Kassel (Germany) • Phone: (+49) 561 - 804 8241 • Fax: (+49) 561 - 804 8242
Commerzbank BLZ 251 300 40 • SWIFT-BIC DRESDE333 • Kto.-Nr. 9680 891 00 • IBAN: DE84 15138 0040 0880 8910 0
Registergericht Kassel HRB 2110 • USt-ID-Nr.: DE 112 595 008

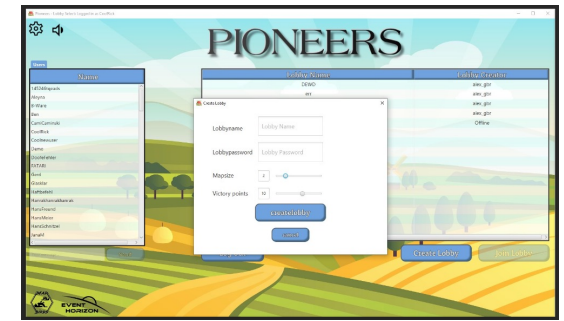


Wireframes



STP'22, Team A, R1

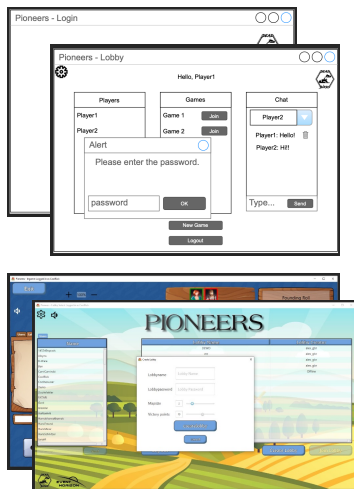
Mockups



STP'22, Team E, R3

2. Stories

Wireframes/Mockups



User Stories

Title: User wants to log in

Start: Alice has launched Pioneers. She sees the Login screen. She already has an account named "alice123" with password "hunter2".

Action: Alice enters her username "alice123" and her password "hunter2". She clicks "Log In".

Result: Alice is now logged in. She sees the main menu.

Title: User fails to log in (wrong password)

Start: Alice has launched Pioneers. She sees the Login screen. She already has an account named "alice123" with password "hunter2".

Action: Alice enters her username "alice123" and her password "hunter3". She clicks "Log In".

Result: The login screen shows an error message "Invalid credentials. Please try again".

Title: User fails to log in (no account)

Start: Bob has launched Pioneers. He does not have an account.

Action: Bob enters "bob" into the username field and "mypass" into the password field and presses enter.

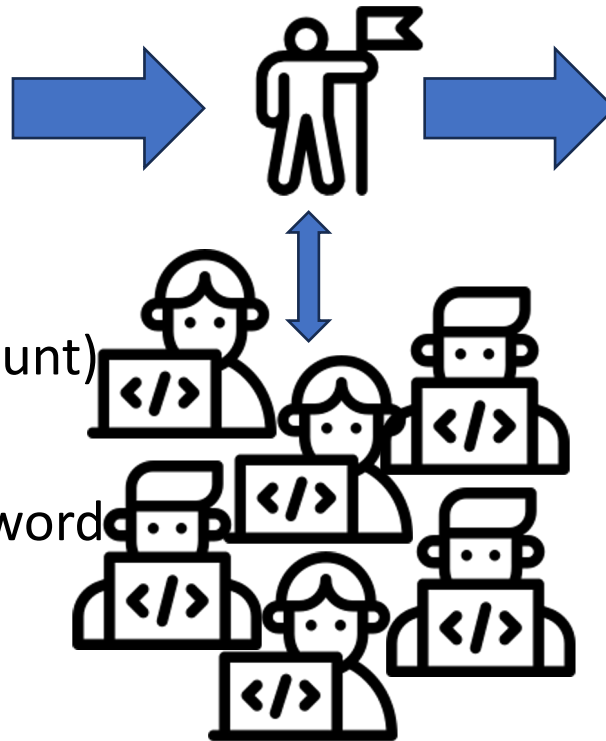
Result: B The login screen shows an error message "Invalid credentials. Please try again".

..... (+10-20 weitere)

3. Schätzung und Gruppierung

Stories (Tickets)

- #1 User wants to log in
- #2 User fails to log in (wrong password)
- #3 User fails to log in (no account)
- #4 User creates an account
- #5 User does not repeat password
- #6 User explores the lobby
- #7 User changes the language setting
- ...




Schätzung [Story Points]


- #1 User wants to log in [5] Login
- #2 User fails to log in (wrong password) [3] Login
- #3 User fails to log in (no account) [3] Login
- #4 User creates an account [8] Signup
- #5 User does not repeat password [1] Signup
- #6 User explores the lobby [8] Lobby

4. Aufgabenverteilung

Schätzung [Story Points]

#1 [5] Login	#9 [3] Settings	
#2 [3] Login	#10 [11] Game	
#3 [3] Login	#11 [3] Game	
#4 [8] Signup	#12 [5] Game	
#5 [1] Signup	#13 [1] Game	
#6 [8] Lobby	#14 [-] Core	
#7 [5] Settings	#15 [-] Core	
#8 [3] Lobby	#16 [-] Core	<div style="border: 1px solid orange; padding: 5px; display: inline-block;">Sprint 2</div>

Verteilung



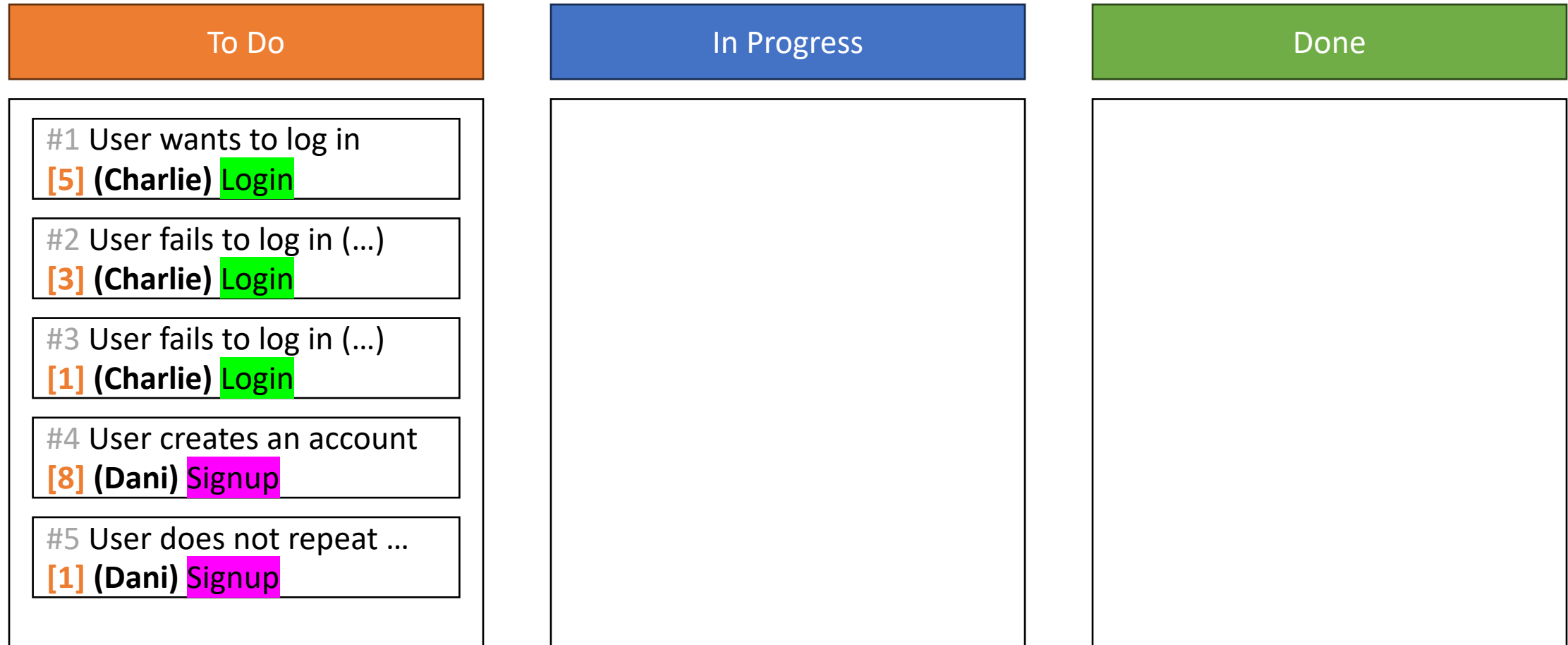
#1 [5]	#4 [8]	#6 [8]	#7 [5]	#14	#17 [3]
#2 [3]	#5 [1]	#8 [3]	#9 [3]	#15	#18 [5]
#3 [3]				#16	#19 [3]
=====	=====	=====	=====	=====	=====
Σ 11 SP	Σ 9 SP	Σ 11 SP	Σ 8 SP	Σ 0 SP	Σ 11 SP



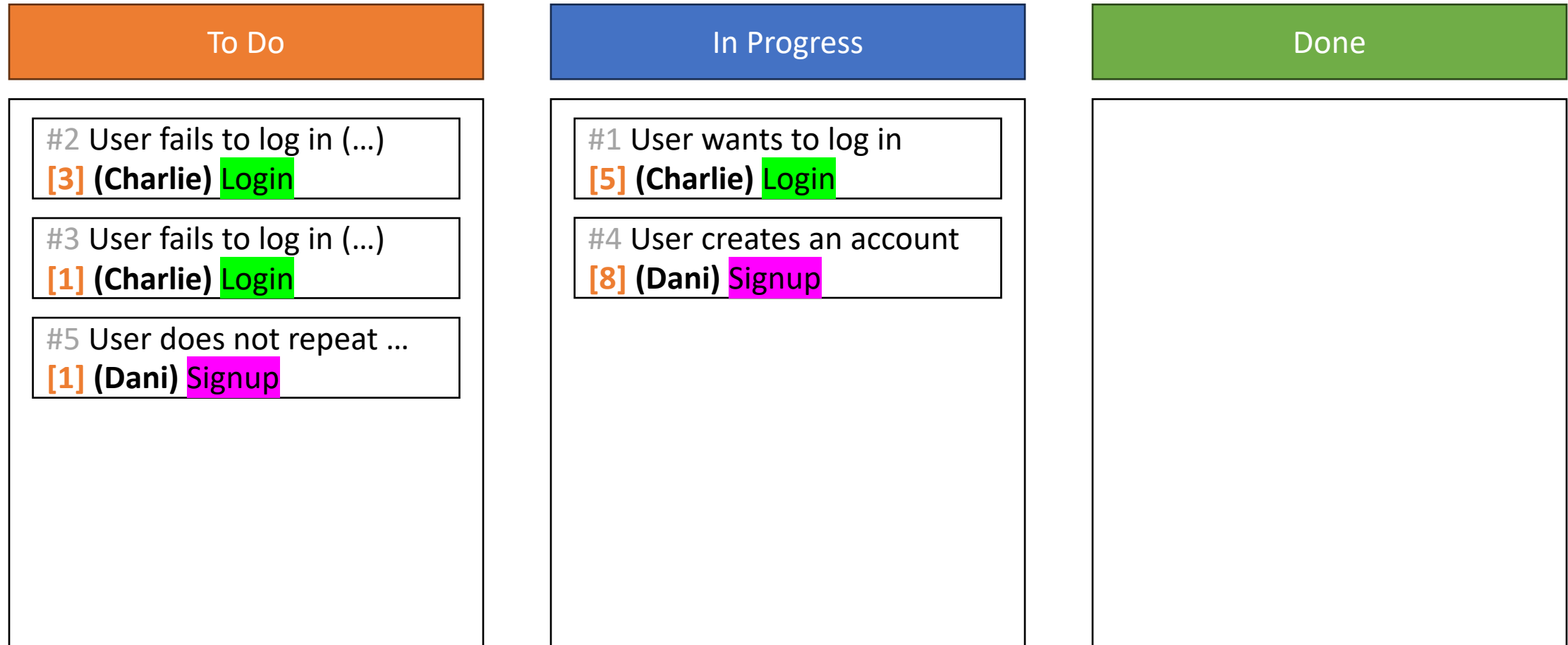
5. Priorisierung nach Abhängigkeiten

Woche 1		Woche 2	
#14 App Init Core	#15 Retrofit Core - Neue Endpunkte	#16 WebSocket Core - Reconnect - Bugfixes	
	#1 [5] Login - FXMLs - Logik	#2 [3] Login - Serveranbindg.	#3 [3] Login - Fehlerfall - Bugfixes
	#7 [5] Settings - FXMLs - Logik	#9 [3] Settings - Sprachwahl	- Übersetzungen
	#6 [8] Lobby - FXMLs - Logik	- Serveranbindg.	#8 [3] Lobby - Bugfixes - WS Updates
	#4 [8] Signup - FXMLs - Logik	#5 [1] Signup - Serveranbindg.	- Bugfixes

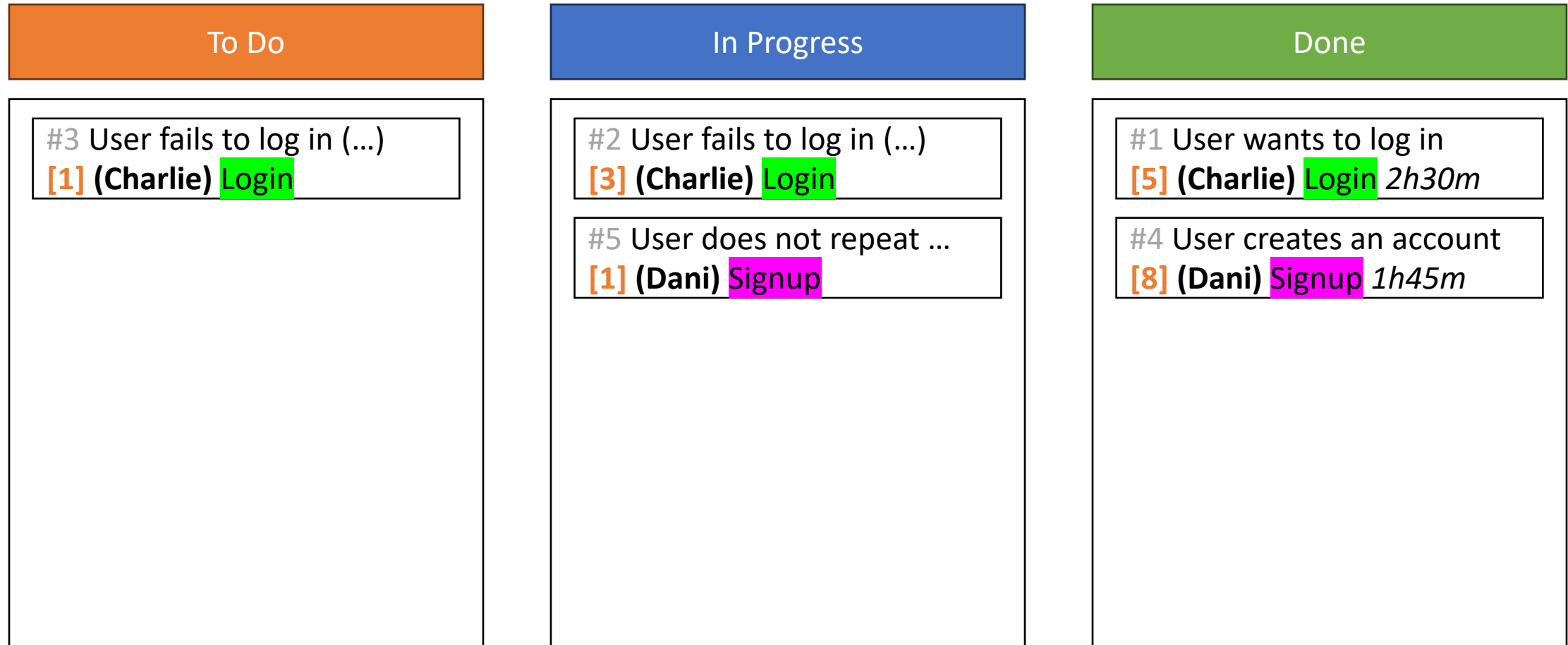
Kanban Board – Anfang Woche 1



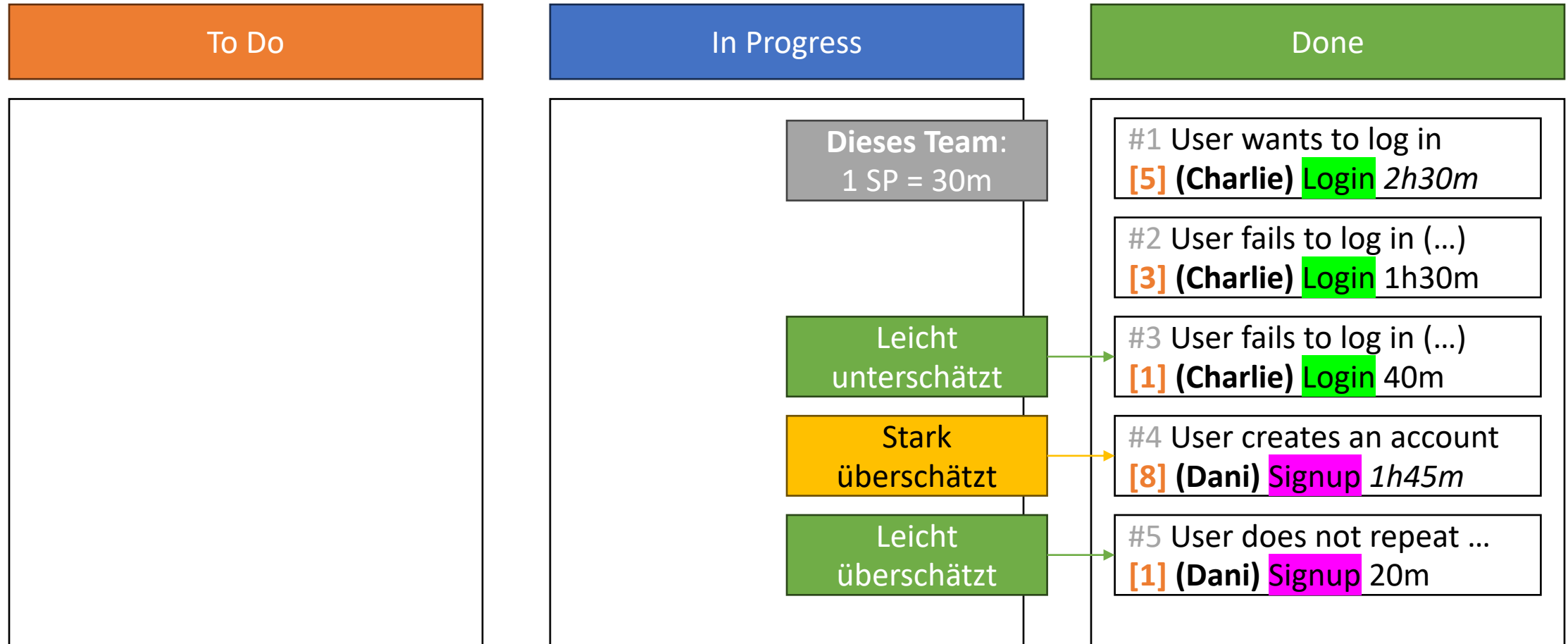
Kanban Board – Mitte Woche 1



Kanban Board – Ende Woche 1

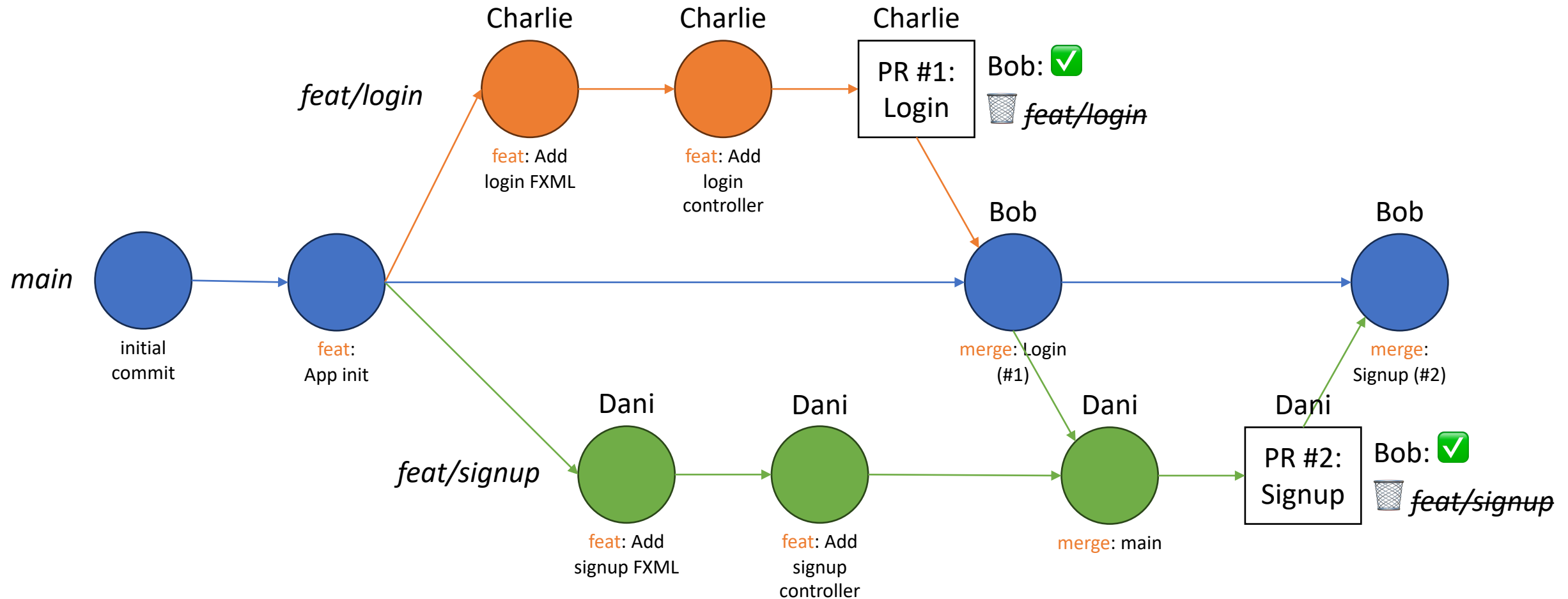


Kanban Board – Ende Sprint

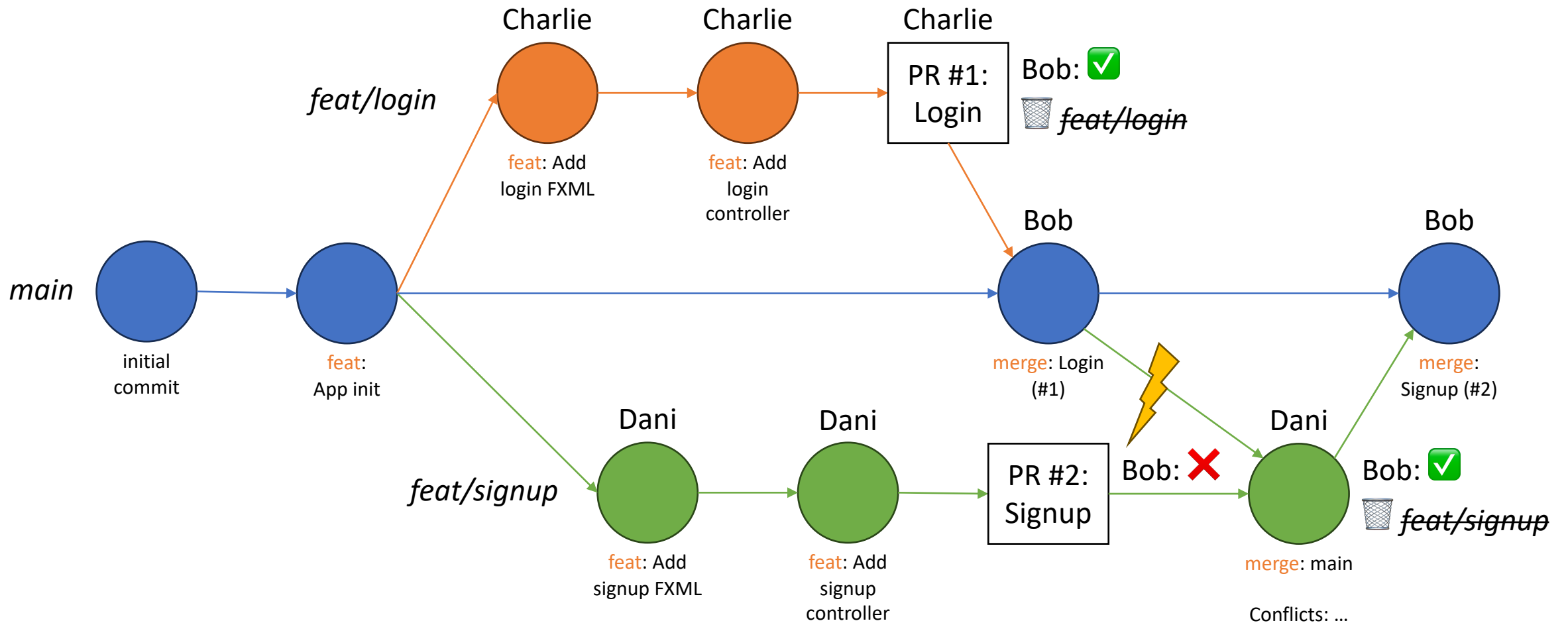


Git Flow

Git Flow: Feature Branches



Git Flow: Merge Konflikt



Git Flow: Conventional Commits

- **Muster:**
`<type>(<scope>): <title>`

`<description...>`
- **Beispiel:**
`feat(login): Add Remember Me`

 - Modified the login screen with a ...
- **Scope:** kann z.B. Gruppierung der Issues widerspiegeln
- **Titel:** aussagekräftig, aber nicht zu lang (ideal <50, max. 70 Zeichen)
- Weitere Informationen in Description
- **feat:** Neues Feature
- **fix:** Bugfixes, Verbesserungen
- **build:** Anpassungen am Build-Prozess (Jar, Assetverarbeitung, etc.)
- **chore:** Wartung am Repo, z.B. Dependencies aktualisieren
- **ci:** CI/CD-Pipeline-Anpassungen (GitHub Actions, ...)
- **docs:** JavaDocs, Readme, sonstige Dokumentation
- **style:** Code-Style, z.B. Reformat
- **refactor:** Umstrukturierung, Refactoring, Warnings beheben
- **perf:** Reine Performance-Verbesserungen
- **test:** Neue, angepasste oder gefixte Tests

<https://www.conventionalcommits.org/en/v1.0.0/>

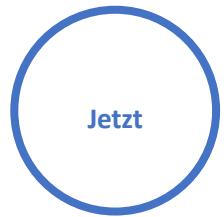
<https://github.com/angular/angular/blob/22b96b9/CONTRIBUTING.md#-commit-message-guidelines>

<https://cbea.ms/git-commit/>

Git Flow: Branches

- **main** (auch: **master**)
 - **Immer** lauffähig und grüne Tests
 - Nur der SM darf mergen
 - Niemand darf ohne PR committen/pushen (Branch Protection)
- **feat/...** – Feature Branch
- **fix/...** – Bugfix Branch
 - PRs sollten priorisiert reviewed und gemerged werden
- **docs/..., refactor/..., etc.** – sonstige Branches nach Konvention
- Manchmal Ticket-Nr., z.B. **feat/login-123**
- Branch gemerged -> **immer sofort löschen**

Zeitplan – die nächsten Tage



**Anmeldeformular
ausfüllen**

<https://forms.gle/H3kjfpDaM4j9N>
[Nor7](#)



Anmeldung beendet

Wir teilen die Teams ein
Jeder erhält eine **Email** mit:

- Teammitgliedern
- Betreuer
- Team-Discord-Link
- **Terminfindung**
- **Kennenlertreffen**



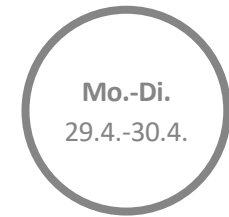
Terminfindung beendet

Betreuer teilen den Termin im
Team-Discord mit



Kennenlertreffen

Mit Team und Betreuern
Weitere Infos zum Ablauf



Kundentreffen

Beginn von Release 1
Anforderungsfindung,
Wireframes, Bonusfeatures
PO und 1 Entwickler

Alle Termine: <https://calendar.google.com/calendar/embed?src=50p736lrm2d3c82dfjdpph9du4%40group.calendar.google.com>