

Agile Softwareentwicklung in einem Open-Source-Projekt

Ziele der Lehrveranstaltung

- Erleben eines realitätsnahen Projektablaufs
- Teamarbeit und eigenverantwortliche Prozessgestaltung
- Zurechtfinden in einem existierenden, komplexen Produkt
- Umgang mit teils unklaren Anforderungen

Das Szenario

Fiktives Unternehmen

Hintergrund

- Unternehmen aus dem Mittelstand
- setzen seit einem Jahr Open-Source-Softwareprodukt ein
- Eigene Featurewünsche sollen implementiert werden
- Ein bis zwei Teams von Entwickler/innen werden für sechs Wochen dafür abgestellt



Ziele

- Pflege der neuen Features soll mittelfristig von der Open-Source-Gemeinde übernommen werden
- Entwickler/innen haben bisher nur theoretisches Scrum-Wissen, das soll nun praktisch vertieft werden

Das „Unternehmen“ aus Sicht der Studierenden

Arbeitsplatz

- Computer-Raum am Informatik-Institut für gesamte Projektdauer reserviert
- 40-Stunden-Woche
- Kernarbeitszeit von 10-16 Uhr

Kollegen

- SW-Entwickler/innen: Andere Studierende im Projekt
- Scrum Master & Product Owner: je WiMi aus der Arbeitsgruppe

Kontingente Ereignisse

- werden im Firmenkontext interpretiert, z.B.:

tatsächlich passiert	Umdeutung im Projekt
Dozent entdeckt ein ernstes Problem in Teilergebnissen der Studierenden.	Product Owner erhält eine erboste E-Mail von seinem Vorgesetzten und stürmt mit hochrotem Kopf in den Arbeitsraum.

Das Open-Source-Projekt

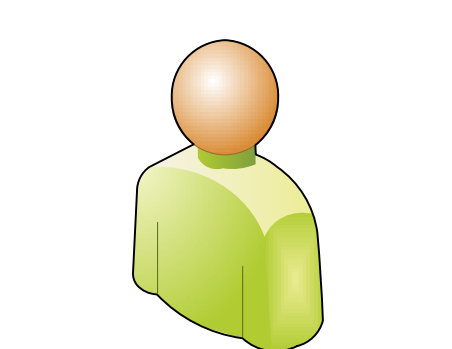
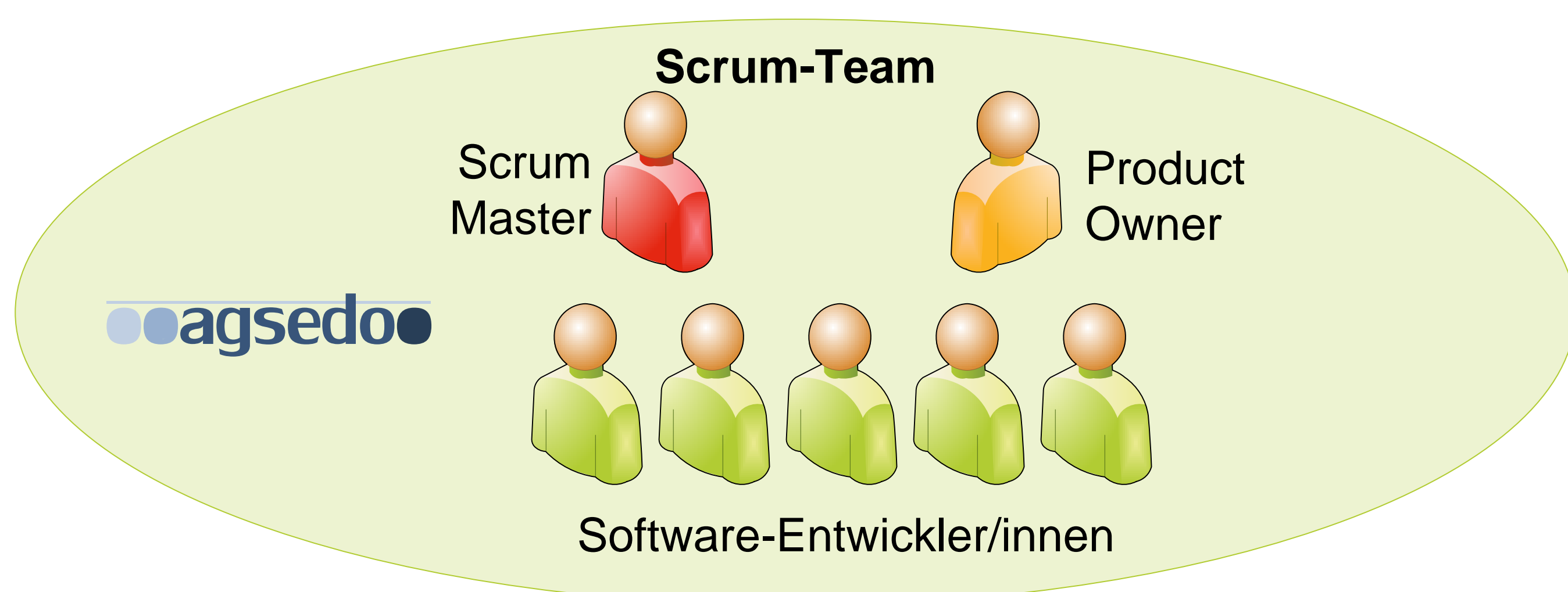
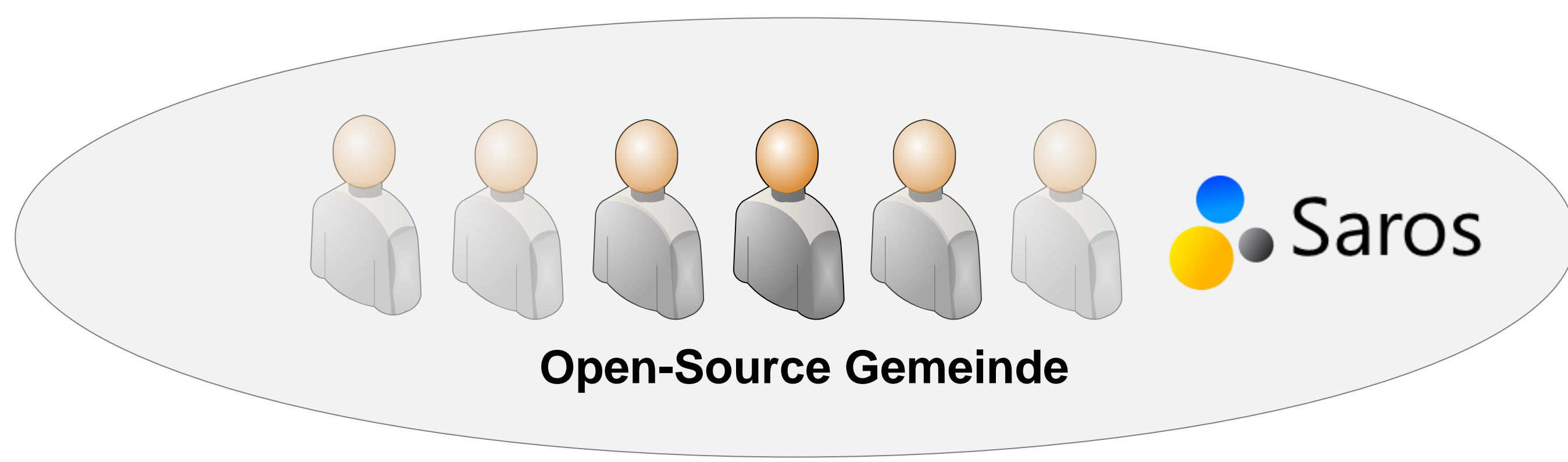
Das Projekt: Saros

- Saros (www.saros-project.org): Plugin für Entwicklungsumgebungen (Eclipse und IntelliJ IDEA)
- ermöglicht mehreren Entwickler/innen das gleichzeitige Bearbeiten von Quellcode

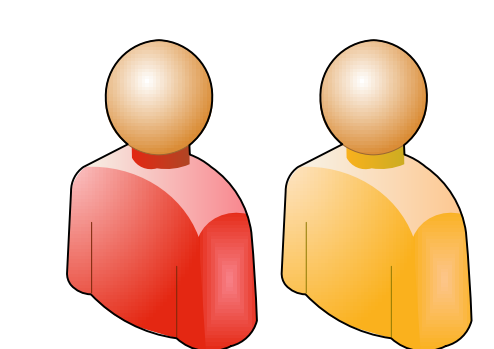


Open-Source-Prozesse allgemein

- Zu bedeutenden Teilen das Ergebnis von Freiwilligen
- Räumlich verteilt
- Asynchrone Kommunikation (Mailinglisten, Defektdatenbanken, ...)
- Lose Strukturen, meritokratische Entscheidungen
- Beisteuern von Verbesserung im Prinzip für jede/n möglich
- Aber: Projekt-eigener Prozess muss eingehalten werden, oft mit Code-Durchsichten



Student/in

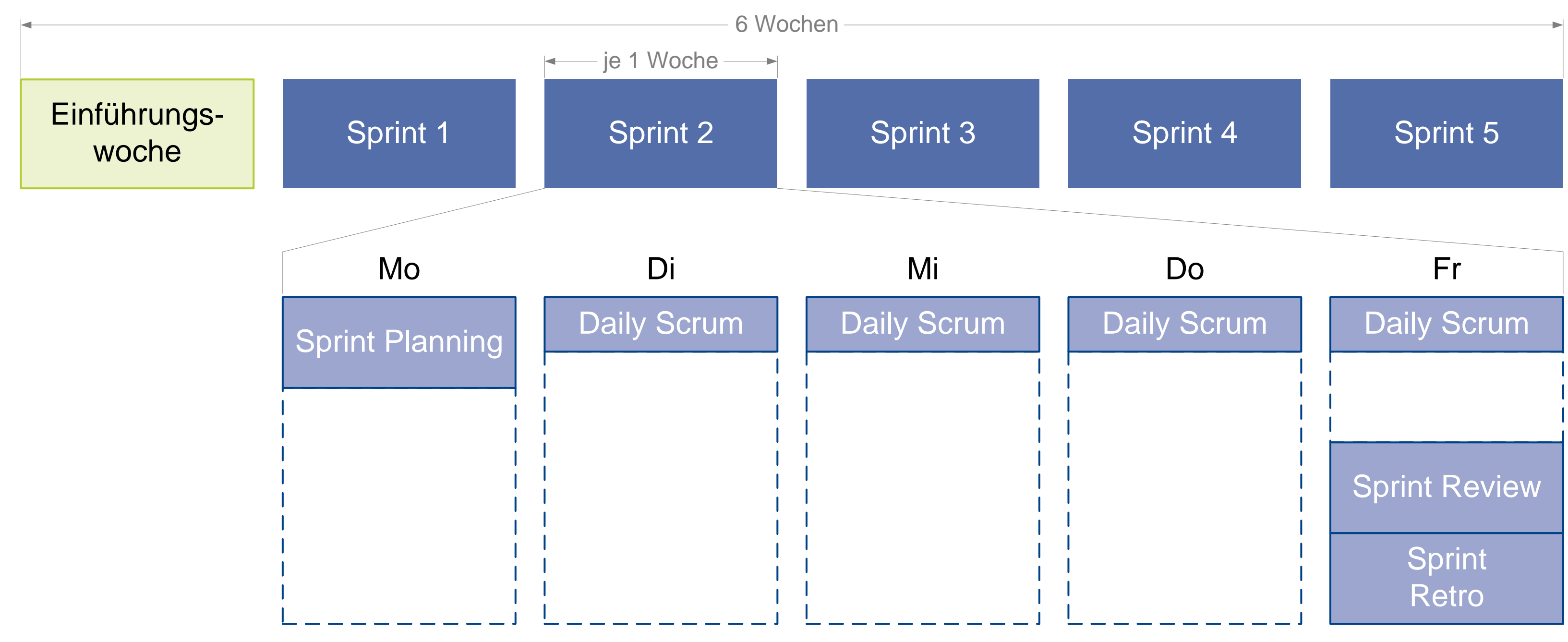


Wissenschaftliche Mitarbeiter

Der Projektablauf

Das Prozessmodell: Scrum

- Allgemein**
- Agiles Prozessframework
 - Gliederung des Prozesses in kurze Sprints
- Rollen**
- **Product Owner:** Anforderungslieferant, verwaltet Product Backlog (priorisierte Aufgaben für die Zukunft)
 - **Scrum Master:** Prozessunterstützung
 - **Development Team:** selbstorganisierte und eigenverantwortlich handelnde Entwickler/innen
- Zeremonien**
- **Sprint Planning:** Zielfestlegung für den Sprint (nicht darüber hinaus) durch das Scrum Team (nicht diktiert)
 - **Daily Scrum:** Planung für den nächsten Tag
 - **Sprint Review:** Auswertung des Sprint Erfolgs, Anpassung des Product Backlogs
 - **Sprint Retrospektive:** Verbesserung des Prozesses selbst



Der Prozess aus Sicht der Studierenden

- Anforderungen**
- Grobe Ideen stammen vom Product Owner, erhoffter Mehrwert für Unternehmen
 - Ausformulierung und Spezifikation gemeinsam im Scrum-Team (z.B. im Sprint Planning)
 - Weitere Diskussion und Abstimmung nach „außen“ (zum Open-Source-Projekt) ist Aufgabe der Studierenden

- Entwurf, Implementierung und Qualitätssicherung**
- Liegen in Hand der Studierenden
 - Aber: Letztlich müssen die Features von der Open-Source-Gemeinde angenommen werden, d.h. viel Kommunikation und Nachbesserungen
 - Qualitätsansprüche der Open-Source-Gemeinde als Stellvertreter für Endanwenderfeedback

- Prozessmanagement**
- In kleinen Dosen punktuell durch den Scrum Master
 - Strukturiert am Ende jedes Sprints (= wöchentlich) in der Sprint Retro