

# Woche 2

# Projektmanagement / Sprint 1

Modul 324

*In software, we rarely have meaningful requirements. Even if we do, the only measure of success that matters is whether our solution solves the **customer's shifting idea of what their problem is**. -- [Jeff Atwood](#), Gründer von [StackOverflow](#)*

# Agile Vorgehensmethoden<sup>9</sup>

- Von grosser Beweglichkeit zeugend; regsam und wendig

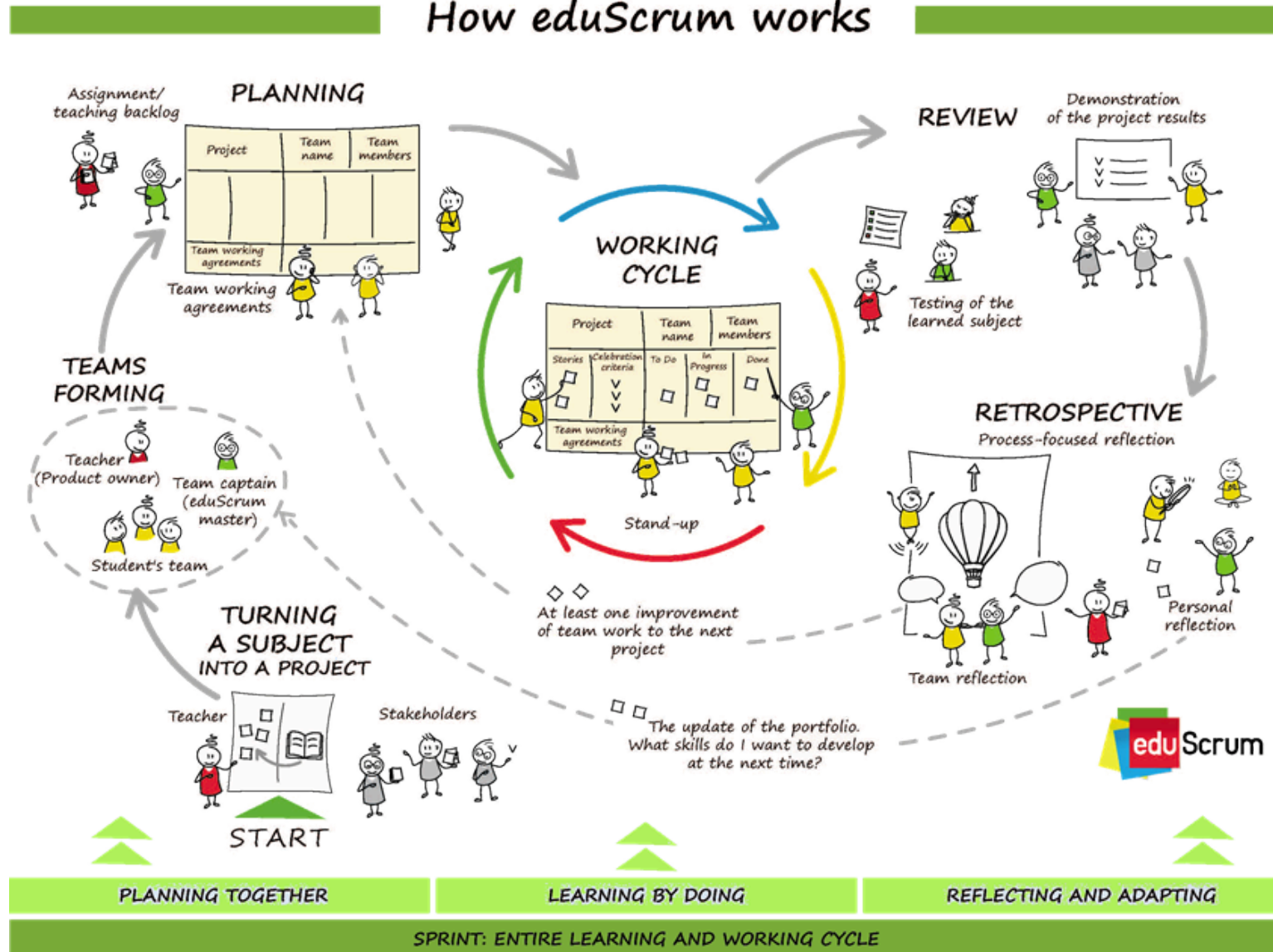
## Inkrementell / Iterativ

- schrittweise erfolgend, aufeinander aufbauend
- sich schrittweise in wiederholten Rechengängen der exakten Lösung annähernd

---

9. Duden | [agil](#), [iterativ](#), [inkrementell](#)

# How eduScrum works



# Sprint

## Start

- **GitHub Issues:** Aufgaben beschreiben und ausarbeiten

## Ende





- **Fachgespräch:** Sprint-Review
- **Persönliche Reflexion:** Retrospektive auf den Sprint

## Pro Modultag

- **10' Stand-up:** Planen, austauschen, Aufgaben verteilen
- **Pull-Requests / Commits:** Aufgaben ausführen

# Sprint 1 - Kickoff

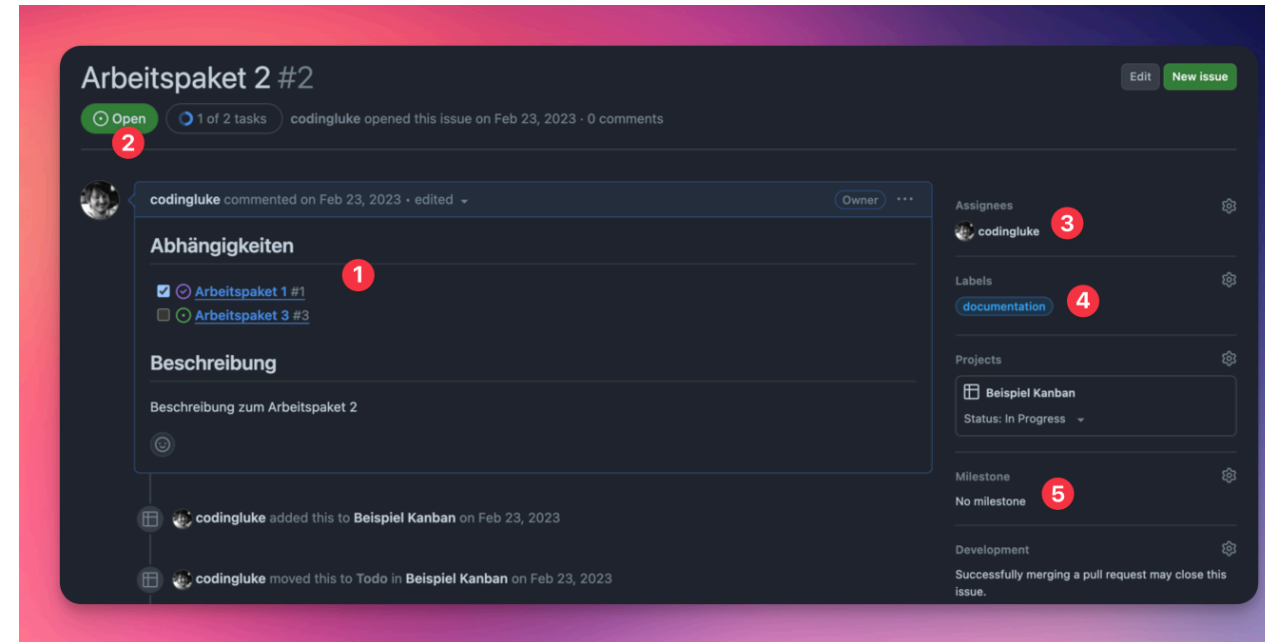
Der Sprint 1 hat zum Ziel, dass sich die Gruppe kennenlernt, zusammen ein Projekt definiert wird und die Grundlage gesetzt wird, um das Projekt umzusetzen.

-  Gruppen kennenlernen
-  Projekt finden
-  GitHub kennenlernen
-  Arbeitsgrundlage schaffen

# GitHub Issues (Arbeitspakete)

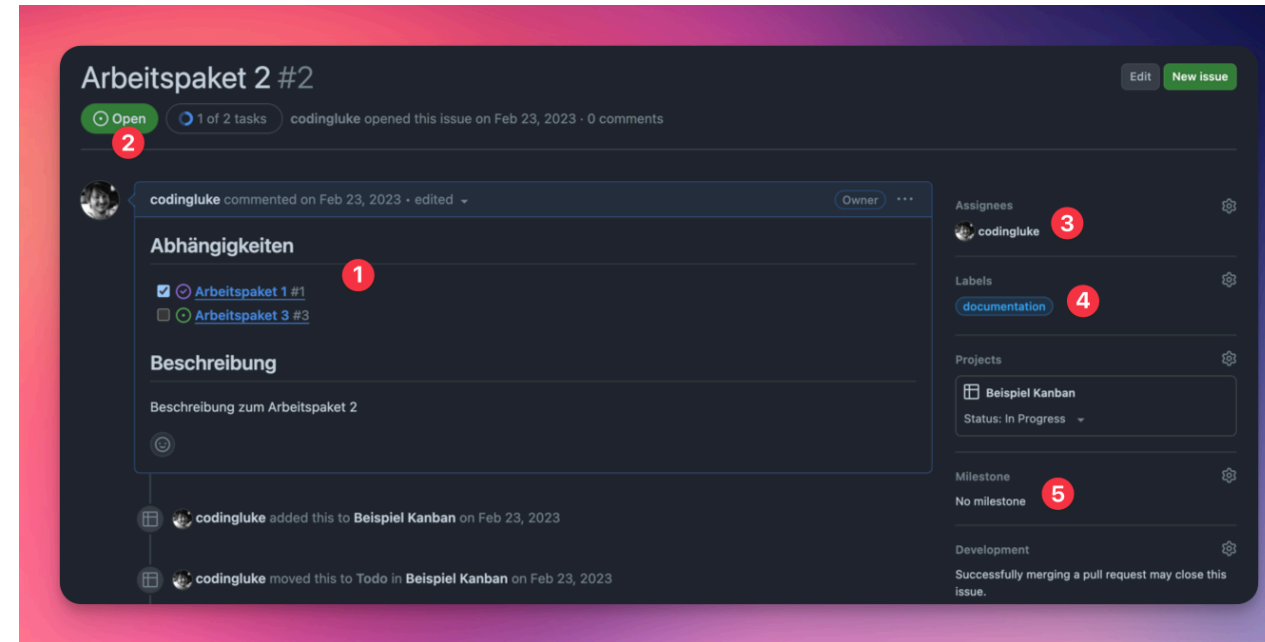
- haben **Abhängigkeiten** zu anderen Arbeitspakete.
- müssen eine klare "**Definition of Done**" besitzen.
- sollten nur **ein Feature** beschreiben.
- müssen nach Fertigstellung an einen **Pull-Requests** geknüpft sein

💡 Wenn ein "und" im Titel steht, kann es meistens Aufgesplittet werden.



# GitHub Issues (*Epics*)

- Beschrieben ein grösseres Feature
- **Zeigen auf Arbeitspakete** welche den Epic umsetzen.
- Dienen als **Gruppierung** von Arbeitspaketen
- **Löst selbst die Aufgabe nicht**, sondern verlinkt zu Issues

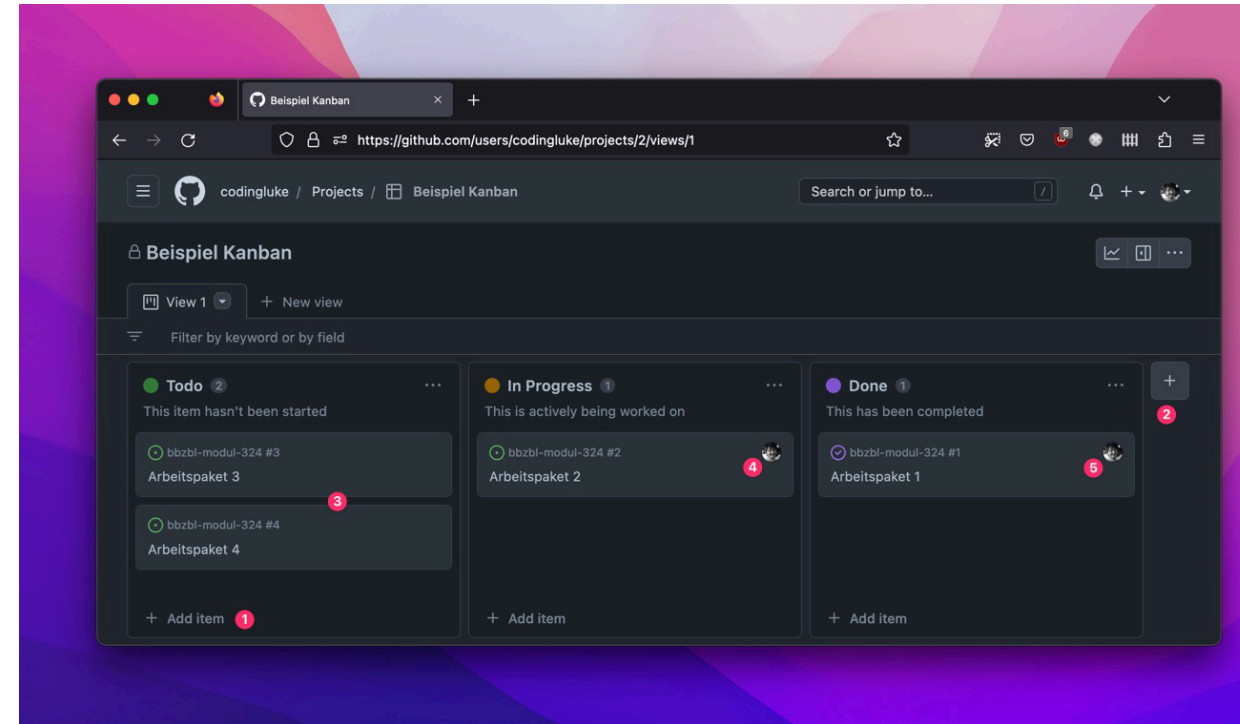




# GitHub Projekt

- Tool um Arbeitspakete (GitHub Issues) zu strukturieren
- GitHub Issues brauchen kein Projekt, es macht es aber übersichtlicher

💡 *Sind nicht einem Repository untergeordnet (leider!)*



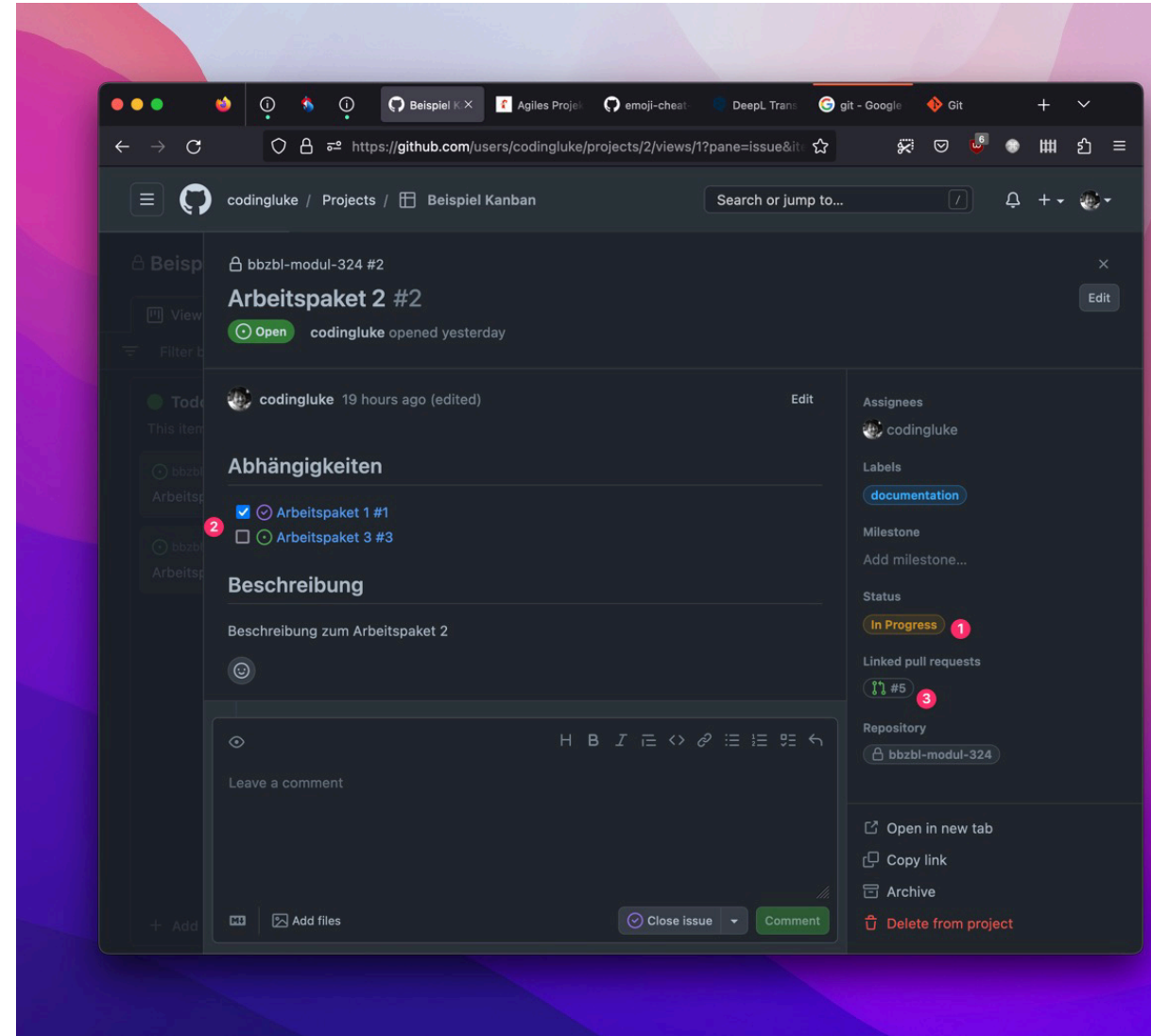
# GitHub Issue in Projekt

Issues im Projekt sehen ein wenig anders aus als im Repo 🧑

1. Status im Projekt
2. Link zu einem Pull-Request
3. Link zu anderen Issues

Ein "Item" im Projekt, das kein Issue repräsentiert ist mit "draft" markiert.

- Ein "Draft" kann zu einem Issue umgewandelt werden.



# GitHub Projects und Issues



zum Beispiel Projekt

# Teambildung

- 🎯 Jeder kennt sein Team

## Auftrag

Zusammen werden die Gruppen zusammengestellt:

- Eine Person die gerne koordiniert
- Eine Person die gerne überprüft
- Eine Person die leidenschaftlich gerne programmiert

- 🧬 Gruppenarbeit
- ⌚ 10 Min
- ✨ **Nur zusammen ist man Stark!**

# Rollen verteilen

- 🎯 Jeder kennt seine Rolle im Team

## Auftrag

- **1 Teamleader:** Erstellt das GitHub repo und ladet andere ein, übernimmt Admin Aufgaben
- **1 Stakeholder:** Schaut sich die Arbeitspakete besonders gut an und prüft die Kriterien
- **1-2 Mitarbeiter:** Ist besonders fleissig am Programmieren

- 🧬 Gruppenarbeit
- ⌚ 5 Min
- ✨ **Gute Rollenverteilung ist die halbe Miete!**

# GitHub Repository erstellen

- 🎯 Arbeitsgrundlage schaffen

## Auftrag


- Der Teamleader **erstellt** ein GitHub repository
  - Ihr dürft meine Vorlage nehmen
  - Der Namen ist änderbar (z.B `projekt - m324-gruppe-x` )
- Der Teamleader gibt allen Mitgliedern die Berechtigungen, inkl. der Lehrperson.

- 🧬 Gruppenarbeit  
(*einer macht, die anderen schauen zu*)
- ⌚ 10 Min
- ✨ **Ihr arbeitet alle im gleichen Repository!**

# GitHub Projekt erstellen

- 🎯 Arbeitsgrundlage schaffen
- 🎯 GitHub kennenlernen

## Auftrag

- Im erstellten Repository ein  **GitHub Projekt** erstellen.
- Erste **test-Issues** erstellen und damit spielen
  - Verknüpfen, Taggen
  - Erstellt Tasks in Markdown

```
- [ ] Task 1
```

- 🧬 Gruppenarbeit (oder auch Einzeln)
- 🕒 20 Min
- ✨ **Schliesst die test-issues, nicht löschen! Sie dienen als Doku.**

# Projekt finden

- 🎯 Arbeitsgrundlage schaffen
- 🎯 Kompromisse eingehen

## Auftrag

- Projekt als **GitHub Issue** beschreiben
- Es muss ein **Web-Projekt** sein
- Es **muss nicht fertig werden!**
- Es **darf ein bestehendes Projekt** sein
- 🧬 Gruppenarbeit
- 🕒 20 Min
- ✨ **Dieses Issue muss nicht nur eine Aufgabe besitzen**

💡 *Statische HTML-Seite, ToDo App oder was grosses? Seit mutig und kreativ!*



# Erste Epics und Issues definieren

- 🎯 Projekt grob erfasst haben

## Auftrag

Hier ein Beispiel von Themen:

- Technologiewahl treffen
  - Die Programmiersprache ist **frei wählbar**
- Infrastruktur aufbauen
- Hello World mit gewählter Technologie
- Erste Features beschreiben
- Wireframe erstellen

- 🧬 Gruppenarbeit
- ⌚ 45 Min
- ✨ **Für alle ein Ticket für die Reflektion zum Sprint 1 erstellen**