



# Projekt-Challenge



S. Berninger



DHBW Heidenheim



# Bsp.: Epic für das WG-Aufräumen

„Unser Vermieter möchte heute um 16:30 Uhr sein an uns vermietetes WG-Appartement besichtigen, bevor er den aktuell auslaufenden Mietvertrag verlängert.“

Project challenge in folgenden Projektgruppen oder denen des Semesterprojekts:

A1	A2	B1	B2
Flaig, Moritz	Gerold, Jonas	Schlegel, Joel	Back, Marvin
Hirschmiller, Tim	Hein, Dominik	Wilzinger, Uli	Behr, Tobias
Honold, Noah	Hong, Ziyi	Rais, Patrick	Ismail, Sam
Müller, Luca	Tulainov, Danilo	Huynh, Diana	Oberländer, Jan-David
Reiber, Aaron	Neßler, Ruben	Küfer, Mara	Schönig, Marc
Sarikaya, Melih	Stephan, Renè	Penschke, Cosmas	Cikrik, Aleyna
Di Caprio, Mario	Schröppel, Paul	Egger, Simon	Schüle, Mario
Seidl, Anian			

# Weitere Epics für die Challenge (bitte eins aussuchen oder eigene Idee...)

- WG-Appartement renovieren
- Gartenparty planen
- die Marathonteilnahme des Teams vorbereiten
- ein Tool für die Raumplanung der DHBW
- einen neuen Kollegen suchen
- eine Mitfahr-App für DHBW-Studierende
- gemeinsam ein WG-Appartement suchen
- eine Speiseplan-App der DHBW-Kantine
- Abschlussfahrt planen...

Das ist nicht das  
Anwendungsprojekt!

# Start: Projektinhalte definieren (30 (?) min.)

Projekt festlegen und Projektziele und –inhalte diskutieren

User stories festlegen (s. nächste Folie)

# User stories – Project backlog

- Ca. 15 für den User im Ergebnis wahrnehmbare (möglichst unterschiedlich große) User stories definieren, formulieren und in Taiga eintragen
- die kleinste User story bildet die Basis für die Schätzeinheiten der anderen User stories (z.B. „Klamotten in den Wäschekorb“ = 1 Story point) – oder eine spezielle kleine definieren
- User stories in Reihenfolge der Wichtigkeit bringen
- noch keine Abschätzungen vornehmen!

**Formulierung Userstory:** **"Als** <Rolle> **möchte ich** <Ziel/Wunsch>, **damit** <Nutzen>"

Beispiel-Stories:

„Als WG-Bewohner möchten wir, dass keine Klamotten mehr herumliegen, damit der Vermieter bei seinem angekündigten Besuch nicht darüber stolpert und unseren Mietvertrag verlängert.“ ( evtl. 1 Story point)

„Als WG-Bewohner möchten wir, dass unsere Küche sauber und ordentlich ist, damit der Vermieter bei seinem angekündigten Besuch einen angebotenen Kaffee annimmt.“ ( evtl. 8 Story points)

Messbare Akzeptanzkriterien (mind. 1) pro User story definieren und eintragen (Bsp.: User story „Wohnzimmer aufräumen“):

- keine sichtbare Unordnung
- innerhalb von 5 min. kein Schmutz auffindbar
- keine sichtbaren Beschädigungen im Raum
- Wohnzimmer riecht für mindestens 2h gut
- Kissen sind exakt ausgerichtet

# Im Forum: Projektinhalte präsentieren

- Die User stories und Akzeptanzkriterien sich gegenseitig vorstellen (je Gruppe max. 10 min.)

# Sprint 0

# Planning poker für User stories (30 min.)

- Jede Gruppe pokert mit Karten um die Story points für ihre 5 User stories (max. 30 min.)
- die kleinste User story (oder eine extra definierte) bildet die Basis für die Schätzeinheiten der anderen User stories (z.B. „Klamotten in den Wäschekorb“ – 1 Story point)



# Nächster Schritt: Präsentation

- Estimates präsentieren, je Gruppe max. 5 min.



## Sprint 0: Parallelisierbare Tasks erstellen, im Team „abarbeiten“ und Retrospektive (30 min.)

- Der Sprint soll möglichst ein sinnvolles (Zwischen-) Ergebnis (Inkrement) erzeugen
- Unser Sprint hat: 20 Story points
- User stories für insgesamt max. 20 Story points auswählen (der Reihenfolge nach)
- Pro ausgewählter User story mindestens 1 Task erstellen und Sprint 0 zuweisen: möglichst ohne Abhängigkeiten!
- Im **Round-robin** darf sich jeder im Team eine Task nehmen, die in der nächsten Runde als „abgearbeitet“ gilt (mehrere Runden, bis zur letzten Task)
- Reflexion im Team (Retrospektive):
  - hätte die Taskbearbeitung so gut funktioniert?
  - was generell im Sprint hat gut funktioniert, was nicht so gut?
  - welche Prozessverbesserungen sind für den nächsten Sprint nötig?
  - welche Unterstützung wird gebraucht?
  - welche Fragen sind offen?

# Im Forum: Erfahrungen darstellen

- **Diskussion der Ergebnisse der Retrospektiven**
- Product owner prüft Sprintergebnis
- Generell dann weiter mit nächsten Sprint... u.s.w. ...

# Ablauf

