Tools für dein SEP
(Unit-)Testing

Unit Testing Basics

- Unit Tests sind …
  - die Grundlage einer gut getesteten Anwendung
  - relativ schnell in der Ausführung
  - auf kleine, isolierte Teile der Anwendung fokussiert
- Problem: Interaktion zwischen verschieden Klassen
- Lösung: Mocking/Stubbing/…
  - Abhängigkeiten zwischen Klassen werden durch Mock o. Stub Objekte ersetzt
  - Diese bilden simple Funktionalität der benötigten Klassen nach

Mutation Testing

- Problem: Wie stelle ich fest, ob ich auch sinnvoll teste?
- hohe Coverage ist gut, sagt aber nichts über die Qualität der Tests aus
- daher: Mutation Testing
- Tests werden auf leicht veränderter Version des zu testenden Codes ausgeführt

Testen von Webanwendungen

- Wie teste ich eine Webanwendung?
- manuell durchklicken ist sehr lästig
- Lösung: Automatisiertes Testen mit Selenium

Tools

JUnit 5	https://junit.org/junit5/
AssertJ	https://joel-costigliola.github.io/assertj/
Hamcrest	http://hamcrest.org/JavaHamcrest/
Mockito	https://site.mockito.org/
PowerMock	http://powermock.github.io/
PIT	https://pitest.org/
Selenium	https://www.seleniumhq.org/download/

UML-Diagramme

Klassendiagramm:

- IBM Rational Software Architect
- yatta UML Lab
- IntelliJ IDEA Ultimate Edition

ER-Diagramm, Sequenzdiagramm, GANTT-Diagramm, Pert-Chart, etc.:

- draw.io
- yED