Sprint 4 - Retrospective

Process Measures

Macro Statistics

Committed Stories: 7
Implemented Points: 31
Planned hours: 70

Implemented Stories: 6
Implemented Points: 23
Spent hours: 74

Detailed Statistics

Hours per task planned: 1.6 h

Hours per task actual: 1.7 h

id of story	number of tasks	points	total hours of estimation	total hours spent
14	6	8	12 h	15.5 h
15	7	5	8.5 h	10 h
16	4	3	7 h	10.5 h
17	2	2	6 h	6 h
18	9	3	7.5 h	8.5 h
19	4	2	7 h	7.5 h
20	4	8	9 h	1 h
0	8		13 h	15 h

Total task estimation error ratio: 0.95

Quality measures

- **Unit Testing:**
  - Total hours estimated - 13 h
  - Total hours spent - 14 h
  - Nr of automated unit tests - 392
  - Coverage of BK (model-related classes)
    - statement coverage - 92.35%
    - branch coverage - 89.76%

- - - function coverage - 94.94%
- **E2E testing:**
  - Total hours estimated - 9 h
  - Total hours spent - 8 h
- **Code review:**
  - Total hours estimated - 9 h
  - Total hours spent - 7 h
- **Technical Debt management** (i.e., analysis with SonarQube, evaluation of issues, remediation):
  - Total hours estimated - 3 h
  - Total hours spent - 2 h
- **Overall Technical Debt** at the demo:
  - BackEnd:
    - number of days - 40 m
    - debt ratio - 0%
    - rating - A
  - FrontEnd:
    - number of days - 1 h
    - debt ratio - 0%
    - rating - A

Assessment

- **What caused your errors in estimation (if any)?**
  - Underestimation of learning process
  - Change of focus: we decided to focus on improving the existing platform and not to add a new functionality as planned
- **What lessons did you learn (both positive and negative) in this sprint?**
  - Learning from sonarqube analysis and issues fixing
- **Which improvement goals set in the previous retrospective were you able to achieve?**
  - Planning GUI design together
  - Team coordination even during holidays
- **Which ones you were not able to achieve? Why?** None
- **Propose 1 or 2 improvement goals for the next sprint and specify how to achieve them (technical tasks, team coordination, etc.)**
  - Better unit test organization by using some of the more advanced features of the test library
- **One thing you are proud of!!**
  - GUI is still a strong point of our platform
  - Test coverage

- Low amount of technical debt
- Input data validation