Agile Software Development

Project Introduction

Ezequiel Scott
ezequiel.scott@ut.ee
Recap

• ASD Practices
  • Requirements and project management
  • Engineering practices
• Web application development
  • Elixir/Phoenix
  • BDD/TDD
• Two homeworks
  • HW1 (Elixir/TDD)
  • HW2 (Phoenix/TDD/BDD)
Project: Find Me A Parking Space!

As many other cities around the world, Tartu has not been designed with a layout to hold large parking supplies. The city government has engaged your team for implementing a **system for managing the parking spaces** in Tartu.
General information

• You will implement a parking management system
  • Using the technology stack
  • Applying agile practices

• Teams of 4 students

• Project duration: 4 weeks

• There will be 4 intermediate assessments (checkpoints) ~ weekly meetings
About the technical solution...

- Achieve a **Minimum Marketable Product**
  - There is pre-defined set of functional features that add significant value
- A backend component that manages the logic of the application (e.g. parking space availability, account management, billing)
- A client (simple view) interface
- Re-use!
The parking management system

The MMP:

• User Management
• Parking Space Search
• Parking Payment
• Billing/Invoicing
About the process...

- **TDD** → Unit and integration tests that verify the logic
- **BDD** → Acceptance tests
- **CI/CD** → A branch strategy and a CI/CD mechanism
- **Scrum** → Practices and evidence
  - Plan the practices by using a [checklist](#)
  - Practices should create artefacts (evidence)
    - Examples: User Stories in JIRA, Retrospectices in a wiki page
  - Scrum variations are allowed but must be justified
The Scrum framework

- **Closing the Sprint**
  - Feedback
  - Self-reflection
  - Celebration
  - Improvements

- **Organizing and Preparing User Stories**
  - Building of Product Backlog
  - Configuration of development environment
  - Distribution of workstations

- **Planning the Sprint Backlog**
  - Sprint Backlog
  - Planning Poker

- **Controlling and Monitoring Sprint Work**
  - Product Increment
  - Product Integration

- **Daily Scrum 24 HS**

- **Product Owner**
  - As an User I want - - -
  - As an User I want - - -
  - As an User I want - - -
  - As an User I want - - -

- **Scrum Master & Scrum Team**

- **Agile Coach**

---

MTAT.03.295 | Agile Software Development | © Ezequiel Scott 2020
Timeline

- **Checkpoint 01**: Process definition
  - Submit the Scrum checklist indicating the practices you will implement, justify the omission of practices, explain it during the checkpoint
- **Checkpoint 02**: At least 2 user stories fully implemented and tested
- **Checkpoint 03**: At least 2 user stories fully implemented and tested
- **Checkpoint 04**: At least 2 user stories fully implemented and tested

- **Sprint 1:**
  - Process definition → submit the Scrum checklist indicating the practices you will implement, justify the omission of practices, explain it during the checkpoint
  - Environment a tool → JIRA or Trello or other. VSCode or other.
  - Backlogs → User stories in proper format covering all the requirements described by the narrative. Acceptance criteria.

- **Sprint 2-4:**
  - At least 2 user stories fully implemented and tested
Checkpoints (intermediate assessment)

👍 Show the team progress (DONE)
👍 Stay within the timebox (~15 min)

👎 No time for discussions
👎 No over-explanations
👎 No excuses

Schedule*:

* Use this Doodle to register your team [https://doodle.com/poll/7riarmevzdyesggz](https://doodle.com/poll/7riarmevzdyesggz) and please contact your TA if you have specific time-constraints
Grading

- Max 32 points

- Overall, the grade is determined based on:
  - The compliance with agile practices
  - The scope of the solution (how much was implemented?)
  - The functional correctness of the solution
  - Functional validation

- Each sprint gives 8 points (max) if you succeed in achieving the goal

- A completeness factor will adjust the final grade (scope and correctness)

  Grade = Sprint1 + (Sprint2 + Sprint3 + Sprint4) * completeness