Evaluation

This document contains a summary of the evaluation principles and criteria as described by the course staff in MyCourses and our coach himself.

Overview

Points allocation:

- 3x5p – Work practices (coach) – After each project review
- 3x5p – Progress (PO) – After each project review
- 30p – End result (coach + PO) – After the last project review
- 4x0.5p – EES participation – After each EES
- 62p – Total

5p from progress and 13-15p from final results are given by the PO only if they **exceed expectations**. 13-15p are given by the coach only if the project is both difficult and the results are **exemplary**.

The client evaluates our achievements against his expectations and what we promise, the Product Vision.

The coach evaluates our practices, behaviour and teamwork, and how we solve problems related to version control, testing, pair programming, code review, collaborative development, continuous integration etc. We need to do more than just that described in the course pages. We must try to innovate and find new practices and ways to work better and achieve more. The coach measures us against other teams!

Pitfalls:

- Hiding problems or failing to identify or react to them
- Failing to fulfill a requirement or make the fulfillment visible (most requirements listed below)

Requirements checklist

Each team member

- Allocate 225 hours to the project evenly throughout the course
- The effort spent per student must be available to coach weekly.
- Give feedback to team members of their success in their role.
- We must apply Scrum as defined in the Project Manual and Scrum Guide.
- Daily Scrum (once a week, max. 15 min)
- Learning Diaries (Before each project review)

Scrum master

- Ensure adherence to course requirements
- Lead and teach the application of scrum (scrum adjustment proposals can be sent to coach after trying)
- Lead the scrum events
- Manage team building (team spirit, communication practices)
- Initiate discussions on unaccounted problems
- Give tips on methods and tools (teamwork, user requirements, architecture, testing, . . .)
- Allocate 175 hours to the project and work as developer, as the budget allows

Artifacts

- Product Vision
- Product Backlog (user story description, order, effort estimate)
- Sprints (at least 6; Sprint Goal)
- Sprint Backlog (tasks to do for Sprint Goal)
- Process Overview (schedule of sprints and events, effort; recurring events, practices)
- Technical Overview (visualising software architecture)
- DoD (2 levels, unit test and functional system test coverage, coding standard, quality attributes)
- Test Session Charter (8 hours of peer team session-based testing).
- Progress/Final report (results: sprint goals, PBIsm, software demo script; status: quality evaluation, spent&remaining effort by student, results from retros; evaluation: product scope vs vision, simplifying/complicating factors, used work practices)
- Allocated and spent effort per person per Sprint?
- Learning Diaries
- 24h before each project review a link to a web page containing the above articles must be sent to the course e-mail, coach, and PO

EES

- Two topic proposals submitted in advance
- Topics can be 1) a problem description, 2) a good solution to a common problem, or 3) a short demo of a useful tool
- Topics are presented by proposers, and discussed
- 1-3 attendants per team

Topic candidates

- Leadership in self-organizing teams? Eliciting intrinsic motivation and self-direction and initiative in team members.
- Interviews for requirements gathering and/or researching feasibility of potential technical solutions?

Sources and links

- Evaluation Principles
- Project Manual
- The Scrum Guide
- The Scrum Primer
- Scrum Glossary