Agile Principles / Miniature (DIT191 / EDA397)

Eric Knauss
<eric.knauss@cse.gu.se>

Organizational

- Thursday
  - Customer meeting
- Exam date
  - Jun-2nd, 2pm

Agile workshop – Acceptance Test No 1.

Apr-14 13:15-17:00

- Welcome to the first Acceptance Test for the Android App project.
- Each group will have **30 minutes** to show **current status and features of their project** and also to discuss the **use of an Agile development approach**.
- Please note the time restriction of 30 minutes, meaning **"Be on time!"**
- The group will meet their group supervisor (Terese or Magnus) in a booked room.
- Each group should bring a laptop where the App should be demonstrated and the development environment (i.e. Trello, Github) should also be accessible.

Acceptance Test - Schedule Group 1-5

Room: Aquarium, 4th Floor in house Jupiter (Close to evaluator)
Supervisor: **Terese Besker**

Time (30 min/each)	Group	Spokesperson
13.15 – 13.45	1	Pedram Shirinbolaghi
13.45 – 14.15	2	Dominik Muth
14.15– 14.45	3	Sebastian Blomberg
14.45 – 15.15	4	Sofia Edström
15.15 – 15.45	5	Wissam Alfreijat

Acceptance Test - Schedule Group 6-9

Room: Room 322, 3th Floor in house Jupiter
Supervisor: Magnus Ågren

Time (30 min/each)	Group	Spokesperson
13.15 – 13.45	6	Fredrik Holmdahl
13.45 – 14.15	7	Johan Eriksson
14.15– 14.45	8	Mustafa Hussein
14.45 – 15.15	9	Linus Hagvall

Agenda today

- Course overview
- Miniature
- Agile Principles revisited

Sprint 1: Getting started

My idea of this course…

- 2 Streams
  - Lectures – *Learn Agile*
  - Project work – *Experience Agile*

- 3 Sprints
  - First sprint – *Getting started*
  - Second sprint – *Focus on Project work*
  - Third sprint – *Advanced Concepts*

Course Objectives

Knowledge and understanding	Skills and ability	Judgement and approach
Compare agile and traditional softw. dev,	Forming a team organically	Explain: people/commun. centric dev.
Relate lean and agile development	Collaborate in small software dev. teams	Apply fact: people drive project success
Contrast different agile methodologies	Interact and show progress continuously	Describe: No single methodology fits all
Use the agile manifest and its accompanying principles	Develop SW using small and frequent iterations	Discuss: methodology needs to adopt to culture
Discuss what is different when leading an agile team	Use test-driven dev. and automated tests	
	Refactor a program/design	
	Be member of agile team	
	Incremental planning using user stories	

Sprint 1

Sprint 2

Sprint 3

What is agility in Software Development?

Miniatures

- Good to get started in the project
  - Shared ideas / concepts
  - http://c2.com/xp/ExtremeHour.html
  - http://www.massey.ac.nz/~dpparson/agilehour.htm

- Idea: Simulate an agile project within a limited time
  - Agile / Extreme Hour do not scale
  - Lego-Scrum does not scale

  - Thus, falling back to a simulation first presented by Chris Rupp, Sophist

Round 1

- Create teams of 4 to 6 persons
- Assign roles in each team: same number of customers and developers
- Customers and developers sit as far apart as possible
- Customers write instructions for developers
- One of the customers
  - brings written instructions to developers
  - can answer (written) questions with (written) answers
- Talking and drawings between customers and developers are not permitted

- Time for this round: 5 minutes

Retrospective of Applied Strategy

What did work well?

- …

What did not work well?

- Not enough time/suddenly over
- Customers wrote too long → no time for developers
- Communication not fast enough
- Descriptions confusing, full of contradictions

What should we change?

- Time management
- Task management
- Incremental work
- Iterative work
- Define/control language
- Use coordinate-system
- Specify from abstract descriptions to specifics
- Communicate “big picture”
- Use metaphors

Round 2

- Same rules as in Round 1, except …
- Shorter Iterations:
  - Developers can send Shape/Picture back
  - Customers can write change request for a Shape or continue with next Shape
- Time for this round: 5 minutes

Retrospective of Applied Strategy

What did work well?

- Task management (increments)
- Iterations
- Metaphors
- Common language
- Time management

What did not work well?

- Integration of Increments and Iterations to whole picture
- Ambiguity of metaphors

What should we change?

- Introduce Integration Management
- More and faster feedback

Round 3

- Only one customer per team! All others are developers
- Customer is allowed to see drawing and memorize it
- Customers explains the drawing using words only
  - No hands!

- Time for this round: 5 minutes

Retrospective of Applied Strategy

What did work well?

- Task management
- Direct feedback of customer
- Verbal communication

What did not work well?

- Integration is challenging
- Customer cannot keep all developers busy
- Strategy not applicable
- Common language not applicable

What should we change?

Conclusion

- What did we learn?
  - Spatial distance hinders communication
  - Multimodal communication helps
  - Communication has limitations
  - Feedback is important: On Product and on Process level
  - Process Improvement is crucial
  - Feedback minimizes Ambiguities

- Interrupting and Reflecting on the process helps to improve it!

Agile Manifesto

http://agilemanifesto.org

- Began as a provocation: Plan-driven development did not safe the Software world…
- Now a very serious movement, well adapted in industry.
- There are a couple of established agile methods: How to integrate these values in everyday software development

Agile Principles

1. Early and continuous delivery of valuable software
2. Welcome changing requirements, even late
3. Deliver working software frequently
4. Business people and developers must work together
5. Build projects around motivated individuals
6. Face-to-face communication is most effective and efficient
7. Working software is the primary measure of progress
8. Sustainable development
9. Continuous attention to technical excellence and good design
10. Simplicity is essential
11. Self-organizing teams
12. Regular reflection