Example Exam Questions

Agile Development Processes

Eric Knauss

Examination

- Written exam, individual, 3.0 credits
  - 60 points, 30 required to pass
- Project, 4.5 credits
  - Grades: Fail/Pass but gives 1-3 bonus on written exam (based on quality/ contribution/novelty)
- Grades
  - Chalmers:
    - [0-49%] → Fail,
    - [50-64%] → 3,
    - [65-79%] → 4,
    - [80-100%] → 5
  - GU
    - [0-49%] → Fail,
    - [50-79%] → G,
    - [80-100%] → VG

About the examples

- Two taken from example exam in git
- One from group exercise in class

- Points will be based on our educated guess on how much time it takes to answer the question
  - I.e. we will solve the exam ourselves with a stop watch before handing it out
  - We did not do that for the following examples
  - Points are therefore only a rough indication in these examples

Agile Principles (8 points)

The Agile Manifesto states the following:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

Explain and give examples on how these four statements are applied / visible (or not) in Scrum. That is what practices or lack of practices show that the process/methodology is indeed in (or not in) the spirit of the agile manifesto. It might be that one of the four statements is out of scope and hence not applied. If this is the case, say so and explain why and how. A maximum of eight points is given. In order to get the full pot two examples of practices are needed for each statement together with an explanation on how the practices support (or does not support) the statement.

Agile Practices (16 points)

For each of the following practices that are commonly used in agile development methodologies describe (1) what it is, (2) which problem in a more traditional/plan-driven development methodology it addresses, and (3) what its (3) advantages and (4) disadvantages are.

a) Sprint backlog (4p)
b) Test-driven development (4p)
c) Pair programming (4p)
d) Daily standup meeting (4p)

Benefits and Drawbacks

Describe the benefits and drawbacks for the use of the below stated agile practice and context. Hence, for each stated pair of practice and context write describe why the practice is suitable/not suitable and what benefit/drawback the developer/development company can expect.

- Test driven development with test first for a mobile phone game developed by a cross functional agile team where half of the people work with the graphics. (2p)
- Planning poker for the development of the software for a heart monitor developed by a team of 100 people. (2p)
- Common code ownership for a safety critical air traffic management system being developed by three teams situated in three different countries. (2p)
- Stand up meetings for the development of an embedded telecom system (hardware and software) developed by 10 teams with 8 people in each team. (2p)
- Scrum planning meetings for a web service development project with two sprints developed by one team with 8 people.

Testing (10 Points)

- Three test techniques have been described in the course, which are test first, unit testing and Visual GUI Testing.

  A. For each of the three techniques, describe its benefits and drawbacks and on what level of system abstraction the technique operates, i.e. what can the output of a test tell you about the cause of a defect? (8p)

  B. Because the three techniques operate on different levels of abstraction it is perceived to be beneficial to use all the techniques in unison, describe why? (2p)

Agile Documentation

Agile proposes advocate that you should have working software over comprehensive documentation. However, documentation is an important part of software development. Below are 5 agile practices. Describe how each one is performed and what benefits and drawbacks it has to traditional software documentation. You may assume that the practice is used in any context you want, e.g. for development of web-applications, mobile applications, safety-critical software, embedded systems, etc.

1. Test first with unit tests (2p)
2. User stories (2p)
3. Kanban boards (2p)
4. Stand up meetings (2p)
5. Retrospective meetings (2p)

Distributed agile

- Assume you are agile coaches for a Kanban team of 8 developers
  - BUT: 5 work here, 2 in Helsinki, 1 in New York
  - Give three examples of what is lost through not being co-located, and how this loss can be mitigated.

Agile Requirements Engineering

- Agile approaches advocate breadth first investigation of requirements. What is the consequence of this to requirements engineering activities in Scrum? What is done by whom, when, based on which practice? Discuss at least 2 roles and 3 practices.