Agile Software Construction
AGILE/2I1281/2I4181
Exam for Spring Term 2007

Beatrice Åkerblom

17th February 2008

Below, you find instructions on how to answer the exam questions and on how to hand in the exam after you are finished. Read them carefully and don't hesitate to contact Beatrice (`beatrice@dsv.su.se`) if you find any errors or instructions that you find vague.

How does this "hemtenta"-thing work?

A *hemtenta* is an exam written by you on your own accompanied by the course literature, other literature and notes from the course. Writing a *hemtenta*, you have plenty of time to structure your answers and to check that your argumentation is clear and easy to follow. This means that the demands on structure and language are considerably higher than when writing an ordinary exam.Since you have all course (and other) literature available, you are expected to use it when writing your answers and you are supposed to show that you use it by referring to it correctly.

You are supposed write detailed answers – if you get stuck you can take a break or continue the next day. Before you hand in your answers you should have read them through and corrected errors and made additions more than once. A *hemtenta* takes longer time to write than an ordinary exam, but the time is more spread out over several days. One could say that a hemtenta takes as long as it takes to write an ordinary exam, plus the time one normally spends on preparations for the exam (though it's not forbidden to use less time). Write *enough* to answer the questions. Don't fill the file with any interesting facts irrelevant to the question you're supposed to answer.

The work should of course be done by you on your own and no co-operation is allowed. Identical answers (that is for example exactly the same answer or the same answer with sentences moved around or slightly changed) are not allowed and will be graded **U**. You can't ask the teachers for directions or help, but help to understand the questions will of curse be given.

Handing in your *hemtenta*

The answers to the *hemtenta* must be handed in both on paper and as a pdf-file.

The file should be named after you, that is if I was to write the hemtenta my answer would be handed in in the file beatriceÅkerblom.pdf. The text should contain your name and email address on the front page.

The electronic version of the *hemtenta* should be handed in attached to an e-mail to Beatrice (beatrice@dsv.su.se). Deadline for handing it in is at 13.55 on Thursday, February 21, 2008. Deadline for handing in the paper version of the exam is at 14.00 on Thursday, February 21, 2008 at the beginning of the lecture in room 510.

A *hemtenta* is not the same thing as an ordinary assignment and handing it in after deadline means that you fail automatically. The results will be announced by a reply to your e-mail.

Grading

Except for the correctness of the answers they have to

- be connected to the course (and/or other) literature by referring to it
- be independently formulated and not copied from literature
- have clear and distinct argumentation
- be formulated in relation to the question, stressing the relevant parts from the irrelevant
- motivated – that is you have to explain why they are correct

Each one of the answered questions will be marked with grades **A-F**. The final grade for the entire exam will be the average grade of all questions. If necessary the final grade will be translated to another grading scale.

The grade Fx will be given individual students who has, with small exceptions, reached the grade E, without serious errors regarding facts, serious language errors, typos or other errors due to sloppiness.

The grade E will be given individual students who has shown that all relevant course goals have been reached and has motivated his or her answers in a correct way while referring correctly to the sources used to support the argumentation.

The grade D will be given individual students who has reached the grade E and has shown in the answers that he or she can see relevant connections between the different course goals showing the ability to make non-trivial comparisons (if applicable).

The grade C will be given individual students who has reached the grade D and also goes beyont the course literature to deepen his or her answers in a relevant way.

The grade B will be given individual students who has reached the grade C and also independently, from his or her own opinions (motivated and well underbuilt) can critisise and value things in a relevant manner.

The grade A will be given individual students who has reached the grade B and also makes nuanced analyses in his or her answers, e.g. draws parallells and can make strong connect between answers and theory.

In addition to the above will the grade C and higher never be given to a student showing big errors due to sloppiness, e.g. serious errors in facts, serious language errors (like broken sentences or spelling errors). The grade A will never be given to a student showing obvious errors in facts or obviously bad language.

Boa sorte! Bonne chance! Good luck! Viel Glück! och Lycka till!

/Beatrice

Questions

1. Sommerville and Cockburn have different opinions about what Software Engineering really is. How do their opinions differ and what consequences does this have on how they think software development should be performed?

   What is Software Engineering in your opinion and is Software Engineering an important concept to you? Which ones do you think are the two most important properties of Software Engineering?

   Which properties makes it important to use Software Engineering methods in software development? Do you think that this will change in a close future (5-10 years)?

   Are there any conflicts between using classical Software Engineering principles and techniques and the increased use of agile methods? Discuss and motivate how and why you think the Software Engineering discipline will change, or why you don't think it will change. Has Software Engineering changed in any significant way during the last 20 years? Discuss how and why.

2. What is really the difference between Scrum and XP? Could one of them be argued to be "more agile" than the other? Why? Which one would you prefer to use in a thought project? Explain why and also describe the project well enough to make it possible to argue for your choice of agile model based on the properties of the project.

3. Test-first development is considered somewhat of a cornerstone in agile software development. Explain what test-first development is and why it is important, especially in the agile setting. Reason about advandages and disadvantages using test-driven development. What is it in test-first development that makes it an agile practice? Could test-first development be used in a traditional plan driven project? Motivate your answer. If you think it could be used in a plan driven setting, does this mean that test-first development *should* be used in more traditionally managed projects? Pros and cons?

4. When planning for a software project and the way the work will be done one can use one or several of the existing process models as a starting point. Describe three different process models from classical Software Engineering, how they are supposed to work and their strengths and weaknesses. Argue both for and against the need to use an existing process model. Discuss the possibilities to combine these models with an agile approach to software development. Would it be possible? How? Where would the major difficulties be found?