

Agile Software Construction
AGILE/2I1281/2I4181
&
Programvaruteknik
IV1300
Re-Exam for Spring Term 2010

Beatrice Åkerblom

12th August 2010

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 course be given.

Handing in your *hemtenta*

The answers to the *hemtenta* must be handed in 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 document should *contain your name and email address on the front pages*.

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 18.00 on Wednesday, August 18, 2010.

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 a student 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 a student who has shown that all relevant course goals have been reached.

The grade D will be given a student who has reached the grade E and has motivated his or her answers in a correct way while referring correctly to the sources used to support the argumentation.

The grade C will be given a student who has reached the grade D 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 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 criticise 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 parallels and can make strong connections between answers and theory and also goes beyond the course literature to deepen his or her answers in a relevant way.

Please remember to check that you have remembered to answer all parts of the questions, it's sadly a too common error made.

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

/Beatrice

Questions

1. Read the post “The Decline and Fall of Agile” that can be found at:
<http://jamesshore.com/Blog/The-Decline-and-Fall-of-Agile.html>
Summarize his opinions and discuss them based on the course literature and your own experiences from using agile methods.
2. What is Software Engineering, really? How would you describe the field in which software engineers do their work? Can it really be argued to be an engineering field? Why, or why not? How would Sommerville and Cockburn answer this question?
3. When working in a software project there are always a number of risks threatening the possibilities to finish the work on time and within the budget and risks that might threaten the quality of the resulting software system. Describe different ways of discovering, evaluating and monitoring these risks. How would you plan for discovering, evaluating and monitoring risks in a project like the project on the course? Motivate your choices.
4. 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 advantages 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?