Information on the report

The deadline for the report is Wednesday 2006-10-25, 12.00

Important notes:

- Write your own argumentation individually and in your own words!
- Choose the perspective that is closest to your own opinion. Your choice has no impact on whether you pass or which mark you get.
- The credibility of your references will affect your mark. For example, peer reviewed scientific papers generally have a higher credibility than web pages. We recommend that you use scientific databases (e.g. INSPEC) or scientific search engines (e.g. www.scirus.com).
- You may include scientific papers that you do not have full access to in your reference list, if the abstracts support your argument, and if you append the abstracts to your report.
- Your mark will be affected by following assessments: language, argumentation, references, presentation and structure.
- Send the reports in PDF-format to Tim Heyer (tim.heyer@kau.se).

Process:

1. Print the assignment guidelines found on the course home page.
2. Choose the perspective (see further down in this document) that is closest to your own opinion on the subject.
3. Justify that perspective, and substantiate your justification by including five references. Your answer must be written in English and consist of 400 - 450 words, excluding references, quotations and appendixes.
4. Apply for a partner on the web. A link is provided on the course home page. You will receive an automatic email with contact information of your partner.
5. Give your text to your partner assigned to criticise your justification.
6. Criticize the other students justification, and substantiate your critique by including five references. Your answer must be written in English and consist of 400 - 450 words, excluding references, quotations and appendixes.
7. Hand in your reports: your own justification, your critique of your another students justification, and a copy of the report you criticised.

Perspectives:

A *Always.* Design documentation, like class and sequence diagrams, should always be created and kept complete and up to date.

B *Initially.* Design documentation, like class and sequence diagrams, should only be created for the initial development of the system but should not be kept up to date later.

C *Sometimes.* Design documentation, like class and sequence diagrams, may only be created when design ideas need to be communicated during development. The documentation should be discarded afterwards.

D *Never.* Design documentation, like class and sequence diagrams, should never be created. Instead code should be used to communicate design ideas during development.

Frequently asked questions

1. *I do not really understand how the reports are supposed to be. Where do I start?*

   The idea is that you first pick a statement you think is most appropriate with respect to software development.

   a) Then you think of why it seems to you the most appropriate statement with respect to software engineering. E.g., why do you think that important parts of programs should be developed in pairs? why do you think that important part of programs should not be developed individually? why do you think that less important parts of programs should be developed individually? why do you think that less important parts of programs should not be developed in pairs?

   b) Then you try to find sources that support your argumentation. E.g, you may think that not everything should be developed in pairs because working in pairs leads to interpersonal conflicts resulting in inefficiency. Can you find (e.g. in INSPEC) sources that support your argument? Such sources do not need to come from computer science but may come e.g. from behavioral science (i.e., general problems when working in groups or pairs). You continue until you found five good references supporting your argumentation.

   Of course, step b may (or rather should) result in new arguments and insights. Moreover there are surly other approaches to your task.

2. *Can I use the XP book by Beck as a reference?*

   The XP book by Beck can probably be used for arguing why (important parts of) programs should be developed in pairs. E.g. 'According to Beck (see [1]), pair programming improves ...'. Consider, however, that Becks bok is not exactly an independent study which shows the superiority of XP compared to other software development methods.

3. *The student who's justification I am supposed to criticize picked the same statement as me. What should I do? It is difficult to criticize the other student's justification since I have the same opinion.*

   You should criticize the other student's argumentation.

   Firstly, we picked statements such that there are enough articles, studies et cetera both to justify and to criticize each statement.

   Secondly, you are not supposed to criticize the statement in itself but the other student's argumentation. Chose arguments used by the other student and criticize them, e.g. 'X writes that working in pairs results in interpersonal conflicts. In [2], however, Beck shows that conflicts are prevented by ...'.

   Thirdly, it is a good exercise to check if and how your own position can be criticized.