

Inf2C Software Engineering Coursework

Perdita Stevens

September 14, 2009

This document describes both of the two Software Engineering assignments for Inf2C. Due dates are 4pm on Wednesday of Week 4 (14th October 2009) and 4pm on Wednesday of Week 8 (11th November 2009), respectively.

Reminder: **Late coursework will not be accepted** unless you have a Good Reason for being late, in which case you need permission to submit late from the course organiser, Jacques Fleuriot (ask for this via the ITO's RT system). Do not ask me for extensions, I'm not allowed to grant them! Deadlines may be arbitrarily strictly applied: do not be even a few minutes late.

Introduction

The time allocated to Inf2C assignments in the course descriptor is 50 hours. (This assumes that you have attended lectures and tutorials and used "private study" time appropriately to keep up with all the course material before you start: time you spend understanding lecture notes, for example, does not count towards the assignment time.) Therefore, you can expect to spend 25 hours on the two Software Engineering assignments put together. The total credit available for Inf2C coursework is 25% of the Inf2C mark. Therefore the total credit available for Software Engineering coursework is 12.5% of the Inf2C mark. Each of the two assignments will be marked out of 10. Therefore, one mark on one assignment is worth one-twentieth of 12.5%, or 0.625% of your total Inf2C mark.

I point this out so as to emphasise that the main purpose of doing this assignment is not to get marks, but to take the opportunity to learn things you wouldn't learn by going to lectures, reading, and revising for exams. Remember that because of the "coursework hurdle" you need to get at least 25% of the available Inf2C coursework marks to pass Inf2C; but beyond that, your main aim should be to learn what you want to learn. This assignment gives you considerable freedom to go in the direction you want to.

The main Intended Learning Outcome from the course descriptor supported by these assignments is:

- Build, document and maintain large Java programs using a modern IDE, rapid development methods, and configuration management tools.

Depending on your choices, it may also support other listed learning outcomes, or others of your choice.

In later Informatics courses you will be expected, as a matter of course, to be able to cope with large software systems: building them, understanding them, modifying them. Many of you will end up in jobs doing this kind of thing. This coursework aims to let you introduce yourselves to working with real, large systems at your own pace and get credit for it.

Collaboration and plagiarism

What you may do You may discuss your work with others, indeed, I encourage you to do so. For example, you might find it useful to discuss issues that arise in open-source projects, each using your own chosen project as an example. If you need specific help with a problem on your own project, you may seek it (e.g., by posting to the newsgroup) and use it, **PROVIDED** that you explain in your own submission who gave you what help.

What you must not do You must not submit any of someone else's work, e.g. words, diagrams or code, as part of your submission without clearly acknowledging their source. If you made use of someone else's ideas, you must also say so.

Marking of work involving acknowledged collaboration In software engineering, you need to be able to solve your own problems, but you also need to be able to collaborate. So I *want* you to give and receive help, but I also want you to work on your own. Here's what I'll do. On each of the two assignments, I will use discretion to award up to 2 bonus marks out of 10 to any Inf2C students who are named by other Inf2C students as having been particularly helpful to others. I will not penalise students for using help from others: that is, it will be possible to get full marks even if you do get a lot of help from others. However, I will also expect to see evidence that each student has worked hard and can solve problems on their own: if *all* your problems were solved with help from others, you will not score as well as if you had successfully tackled some on your own.

I am aware that this is an unfamiliar way to run coursework. If you are in doubt as to what it means in a particular practical case, contact me.

PART 1

Preparation. 4.5 hours: no credit

Visit the website <http://sourceforge.net>, which is the leading support site for open source software, and browse. Aim to “learn your way around”: understand the structure of the site, what you expect to be able to find for a given project, etc.

Pick a package, which you will work on for BOTH parts of the SE coursework, which:

- you find interesting
- is reasonably active: check for evidence that several different people have been actively working on the project recently
- has source code you're prepared to read (this probably means it's in Java, but it's fine by me if you pick a project in another language, e.g. one which you want to learn, provided you're prepared to spend the extra time it will take)
- uses systematic bug tracking, and has at least a few open bugs.

You may find <http://sourceforge.net/top/> useful.

Once you have decided on your package, register it against your name using the link from the Inf2C page.

Every student must use a different package. (There are over 100,000 registered projects, so this should not be too restrictive: but you will probably want to investigate several packages in parallel, in case your first choice is taken by the time you come to sign up for it.)

Browse its SourceForge page. Download its *source* code and build the package. Check that it runs, and play with it. (You may work on any machine, any platform. Do remember, though, that if you work on a non-DICE machine, it is your responsibility to back up your work.)

Question 1: 4 marks, 2 hours

Explain

- why you picked this package
- briefly, what the package is for
- *exactly* what platform you built it on, whether you encountered any problems, and how you solved them if so.
- some kind of evidence that you actually ran the package, such as a screen shot with your name showing in a field of the GUI.

Question 2: 3 marks, 2 hours

Pick a bug report (link to it!). Drawing on the material we covered in the lecture called *The User's Role*, and the readings for it, comment on how useful or otherwise the report is. Can you reproduce the bug? Can you identify which source files are relevant to it? Explain, using any reasonable combination of text and/or UML diagrams, what the problem is. Discuss possible fixes. (If from the first bug report you pick you *cannot* reproduce the bug, so that it is hard

to do the rest of the question, explain why and pick a second bug report that seems more likely to be reproducible, and so on until you have spent 2 hours on this question.)

Question 3: 3 marks, 4 hours

Based on working with the package, reading the project page, bug list, source code, mailing lists, and anything else relevant: discuss what kinds of work contributions might be useful to the project, and why, if it is to continue successfully. (See below for ideas of the kinds of contributions you might consider, though you may well be able to think of others.) What would be *most* useful, and why? Explore what would be involved in each contribution, to the point where you can discuss how hard you think it would be for you to make each kind of contribution that you consider.

Submission Your answer to this assignment should be an HTML web page `index.html` (along with other files in the same directory if appropriate). Very basic HTML is fine, but use links into the SourceForge site as appropriate: I want to be able to mark this by starting by reading your `index.html` in a browser (Firefox, but try not to depend on that!) and looking at the links you give me from it. I don't want to have to go searching for the things you discuss: if you want me to see something, link to it. Put all the files you wish to submit, including `index.html`, in a directory called `inf2c-se1`. Make sure you are in the parent directory of `inf2c-se1`, and submit using the command:

```
submit inf2 inf2c se1 inf2c-se1
```

PART 2

Develop a contribution, of your choice, to the project.

Possible contributions (or parts of contributions: you don't have to choose just one of these) include:

- Fixing one or more bugs from the bug tracking system (remember testing!)
- Reporting bugs into the bug tracking system (be careful that they really are bugs, that you submit useful bug reports, and that you check that the bugs are not already known).
- Writing or improving user documentation for the project
- Developing documentation of the design of the project, e.g. a UML class diagram of the overall structure of the project (be sure it's correct UML, correctly corresponds to the project, and is helpfully laid out)
- Improving the design of the system by applying *relevant* refactorings
- Answering questions on the project's mailing lists, e.g., questions from new users of the project

- Adding tests, perhaps for a particularly critical part of the system, or a part known to be buggy.

You do not necessarily have to choose one of the contributions you identified as “most useful” in PART 1: you may take into account your own skills and indeed your own learning objectives. You should, however, be doing something which is genuinely useful to the project.

Notice that, while in PART 1 you might have built the package from the “latest stable version”, some contributions, notably bug fixes, will need you to download the latest sources (typically from the CVS archive).

Consider actually adding your contribution to the live project, and do so if appropriate: different projects will have different mechanisms for doing so. (However, only do this once you are confident that the quality of your contribution is high enough to be a genuine benefit to the project.)

Submission As for PART 1, write an HTML web page `index.html`, with links as appropriate, which explains what you did. If you successfully added your contribution to the project, so that you can point to it on SourceForge, do so. Otherwise, submit it to me (i.e. submit any new and changed files, together with your explanation). Put all the files you wish to submit, including `index.html`, in a directory called `inf2c-se2`. Make sure you are in the parent directory. Submit using the command:

```
submit inf2 inf2c se2 inf2c-se2
```

Timings The estimated time for this assignment is 12.5 hours. You should expect to spend 2 hours on writing up your contribution (though I suggest that you do this as you go along, rather than leaving it until the end). Within the 10.5 hours for making the contribution, the split will of course depend on what you choose to do. I suggest that you identify at least two subtasks. For example, perhaps start with a task you think relatively easy which you expect to take no more than 2 hours, followed by a related but harder task taking the remainder of the time available. I advise you to mention in your submission how long it took you to do each part of your contribution.

Marking guidelines A passing mark (4/10 or above) means that your submission showed that you have spent the recommended time on the assignment, and that in the process you demonstrated competence in the relevant skills from the course. (E.g., UML if your contribution involved UML, JUnit testing if you wrote tests, refactoring and understanding of good design if you refactored, etc.)

A first-class mark (7/10 or above) means that your submission also showed that you had made a substantial, genuinely useful contribution to the project.

Recall that up to 2 bonus marks out of 10 are available for helping others, at my discretion. (E.g., you could obtain 8/10 even if your own submission wasn't quite first-class in the sense above, if in my judgement you had been exceptionally helpful to others.)

Bear in mind:

- The correctness of your contribution is more important than its size. If there's one thing worse than a bugfix that introduces another bug, it's documentation that misleads the reader... Consider carefully what evidence you can provide that your contribution is correct.
- I can only give you credit for something if I understand what you did, so be careful with explanations.
- Usefulness is in the eye of the user, so if you have (verifiable!) evidence that other people appreciate your contribution (e.g., positive mail on the project's mailing list), reference it in your submission.
- You will not be penalised for seeking help from others, provided that you also solve some problems yourself, so if you get stuck and think others might be able to help you get unstuck, ask (e.g., in the lab or on the newsgroup – I will read the newsgroup, so you need only mail me if you want to ask something in private). Remember to mention who helped you in your submission.
- If you find it useful to spend more than the recommended time on the assignment, that's fine. However, if you have kept up with the course material and are competent to pass overall, you should be able to obtain at least a pass mark within the recommended time (and I expect to be able to give full marks to some exceptional students who have spent the recommended time).

If any of this is unclear, ask on the newsgroup

Have fun!