CSC444 **Software Engineering I** Fall 2002

Assignment 5: change request response and project wrapup

Part 1 – change request

This is a report on how you responded to the change request. Note: you do not need to have completed the implementation and testing of the change request to complete this assignment, but you will need to have completed a fairly detailed design of your set of changes.

You should submit the following:

1) A short report (up to 1500 words) on the process of implementing the change request. Your report should be as specific as possible in addressing the following points, using examples where appropriate. Your report should cover the following:

- Which of the three features described in the change request did you chose to add to your system and why?
- What simplifying assumptions did you make (if any) in implementing the change request?
- How did the architecture of the software affect your ability to add the new feature?
- Would a different architecture have helped?
- Which parts of the system are affected by the change? (Be specific)
- What proportion of the overall system is affected? (how did you measure this?)
- What impact will your changes make to the quality and complexity of the software? (Be specific about how you measure complexity)
- Which software assets were you able to re-use, and which have to be modified? (you software assets include documentation, test cases, test drivers and stubs, configuration files, manuals, packaging, etc)?
- How far have you progressed in making the change, and how much more effort would it take to complete the change?

Part 2 – project wrapup

This is an exercise in packaging your phase 2 integrated system ready for delivery to *end users*. Note that end users are not the same as your customers (i.e. other teams) for phase 3. End users typically don't want the source code, and don't want to have to compile anything (or at least not knowingly!). They want installation to be as simple as possible, so that they can get on and use the software!

You should submit the following:

1. A user manual for your game. (Note: this manual should also be included in the package that is delivered to the customer).
2. Packaging instructions for your game. These should include a description of how the software is to be shipped to customers (users), the an inventory of the shipping (file names), an indication of where to find installation instructions within the package, and a description of how to get more help and how to report errors. If your system is available in multiple versions (e.g. for different platforms), make sure you describe and distinguish between the versions.
3. An installation guide for your game. Make sure your instructions give clear description of any system requirements for the program to run.
4. The finished package. Include your finished software package so that we can try out the above installation instructions and use the editor. What you actually submit will vary according to your shipping mechanism. If you plan to distribute the software electronically, include a URL or FTP location. If you are distributing it physically, include a disk, CD-ROM (or whatever) with the assignment.

Note: In grading this exercise, we will attempt to install your software and run it, using your instructions. If we can't follow the instructions, or the installation fails, this will be reflected in your grade for the assignment.

Note 2: Item 4 must be included when you submit this assignment. If it is a pointer to a website or FTP location, the files at that location must be available at the time you submit this assignment. Late penalties will apply otherwise.

Background Information

There is no background information for this assignment!

Marking scheme

Listed below are some of the things your TA's will look for when marking this assignment.
Use this list to check your work before you hand it in!

Part 1 – change request

Is the user manual clearly written so that a new user could figure out how to use the program quickly?

- Is the document written in simple English, so that a wide range of users can understand it?
- Is there an introduction that gives a quick overview of what the software does?
- Is there a brief "getting started" guide for users who don't want to read lots of detailed description?
- Are the file formats explained?
- Does the manual have a clear table of contents and index to help readers find what they need?

Does the user manual give a good impression of what the user will see as they use the software?

- Does it include screen snapshots?
- Does it include lists of commands/menus?
- Does it include some sample input/output files?

Is the user manual consistent with the packaged product?

- Is the user manual clear about which version(s) of the software it applies to?
- Does the user manual only describe commands/functions that are actually implemented?
- Does the user manual make it clear which functions (if any) do *not* work?

Were packaging instructions provided that give a clear idea of how the software would be shipped to the customer? (Note: The packaging instructions could just be a README text file).

- Do the packaging instructions make it clear what platforms the software works on?
- Is it clear from these how the customer would get hold of manuals, installation instructions, etc, as well as the program itself?
- Is it clear what the customer can do in the event of any problems downloading, installing or running the software?
- Do the instructions include information about how to report errors?

Are the packaging instructions complete?

- Do they clearly state which version of the software they refer to?
- Do they give a complete inventory of files included in the release?
- Do they give version and date information for each file?
- Do they include manuals and other documentation, as well as the software itself?

Is the installation guide clearly written and easy to follow?

- Did it allow you to (download and) install the software without any additional knowledge?
- Did it make it clear to you what support software you needed, and tell you how to get it?
- Were the installation steps simple?

Could you successfully download and install the software by following the instructions?

- Award this mark only if you ended up with a running system by following only the provided instructions.

Was the installation complete?

- Is it the right version?
- Did it include the user manual?
- Did it include the packaging and installation instructions as described above? (These are all part of the software and should *always* be bundled with it, even if they are also provided in hardcopy)
- Is the version information on the submitted assignment consistent with the version information on the installed software?

Was the finished package made available to you in a timely way?

- only award this mark if all parts of the assignment were ready, including any downloadable files

The last mark is for creativity in packaging the software. Note: no extra credit will be given for use of colour, as not all students have free access to colour printing. Also, size (of the package) is irrelevant for this mark: if all the software and manuals are distributed electronically, then something as small as a postcard would work as the deliverable, with sufficient instructions to find and download the software.

- Was the packaging attractive and helpful to the customer?
- Is the distribution mechanism sensible?

- Is the packaging likely to help to sell the software?
- Does it enhance the brand name image of the company that produced it?
- Can it easily be shipped to the customer without being damaged or falling apart?

Part 2 – project wrapup

Did they describe which of the three features in the change request they selected?

- Did they actually select one?
- Did they discuss *why* they selected this one in preference to the other two?
- Did they consider the cost of making each change when they made their selection?
- Did they consider how much of the system might be affected by the change?
- Did they mention any other factors in making the decision?

Did they describe the assumptions and/or limitations they used for making the change?

- Did they document any (sensible) assumptions at all? (there ought to be some, as the change request is underspecified)
- Did they say anything at all about their assumptions for why the users want these changes?
- Did they consider the relative importance of the various functions needed for the feature they chose?
- Did they describe any effort to balance the need to satisfy users with the need to make the change easy?
- Did they identify limits to the requested changes that could make the job easier?

Did they assess how well the architecture of the program permitted the change?

- Do they say what the architecture is? (e.g. the original 4 modules with some control scheme)
- Do they assess whether the architecture itself will have to be changed?
- Did they assess whether the architecture allows them to "contain" the changes?
- Did they assess how much the change alters the interconnections between modules/units
- Did they consider whether the architecture supports (re-)verification of the system once the change was made?

Did they compare how other choices of architecture may have made the change easier or harder?

- Do they describe any different architectural styles? (e.g. pipe-and-filter, implicit invocation, etc).
- Give credit for any sensible discussion of how one of these styles might have made the change easier or harder.

Did they say which modules/unit are affected?

- To get credit they need to be specific, i.e. name the modules/units, and say why those modules units will be affected.

Did they give some sensible measure of the proportion of the system affected by the change?

- Is it a concrete metric? e.g. numbers of procedures new or altered vs. total numbers, or % of lines of code, etc.
- Do the estimates look sensible?

Did they assess the impact on quality and complexity?

- Did they select sensible specific measures for quality and complexity?
- Did they give sensible estimates for how the change would alter those measures?
- Did they consider whether there would be any change to coupling and/or cohesion?
- Do they combine the metrics together to give some high level summary of the impact?

Did they say which assets they were able to re-use and which they would modify?

- Does the discussion cover all of: documentation, test cases, test drivers and stubs, configuration files, manuals, packaging?

Did they give a summary of progress on making the change?

- Did they estimate total effort (including an estimate of remaining effort if not completed yet)
- Does their assessment consider the amount of re-testing of the software needed?
- Did they assess any problems encountered or anticipated (i.e. risks) that affected or could affect progress?

The last mark is for a good attempt at completing the change.

- On a scale of 1 to 10, how far did they get?