Guidelines for the Final Report, 2009

October 2, 2009

**Deadline:** The deadline is *midnight Friday 23rd October.*

**Note:** this is the last day of semester and no extension is possible.

1 Report Size

The final report should be a summation of your design activities on this project. As such it implicitly includes everything that was in your previous reports, but there may be changes.

This report is potentially very large. It must be recognised that a large report simply cannot be easily read, and hence cannot be effectively assessed. We need to define realistic parameters for the report and guidelines for how the report can direct the reader to the important components of the design and assessment of those components.

<table>
<thead>
<tr>
<th>Size limits:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The maximum size of the main body of the report is 75 pages. <em>Note change.</em></td>
</tr>
<tr>
<td>The maximum size of the total report including appendices is 200 pages.</td>
</tr>
</tbody>
</table>

*But it should be recognised that a report of this size cannot be read in its entirety. The report must be organised to guide the reader to the important parts.*

*Information in the appendices will only be read if referenced.*

**Bigger is not better!**

2 What is this report about?

Your final report should be concerned with explaining your overall design. It should consist of:

- Executive summary giving overall summary of the report.
- A review of Requirements.
• Specification.
• Design.
• A review and assessment of your prototype implementation
• Discussion of possible physical deployment
• Requirements tracing
• Assessment of your development process.
• Project management
• Special features.
• Reflections and Introspection

some parts may be organised as appendices. The organisation is up to you.

It is important to understand that the above items should not necessarily be interpreted as discrete sections of the report. They represent parts of the report, but they should be integrated.

Write to be Read

Your report should be written in a style that assists the reader to navigate your design and discussion.

This report can be very large and quite indigestible. The report must be written to be read. Don’t assemble page after page of technical detail with no linking narrative. If you do, there is a large probability that many of those pages won’t be read carefully. If you want something to be read, then direct the reader to what should be read and explain why it is important. A number of things may be placed in appendices, but if you want them to be read they must be referenced.

You should try to write the report in a way that enables the reader to see the important things that you want them to see. Don’t expect the reader to “fall over” those aspects of your design “by accident”. You should be telling a story.

Executive summary  Probably doesn’t need explaining, but this should be a summary, not an advertising blurb. It can and should guide the reader to the important parts of your report, and draw attention to important properties of your system.

Requirements  the requirements are important and should be in the report somewhere.

Specification  Again this is critically important and should be in the report. This should be principally the EventB model.

Design  This is what this report is all about. Try to explain your design. Probably your design is best explained through software architecture and class diagrams, probably using UML. Many teams claimed a strong linkage between the prototype implementation and the EventB model. If claimed this should be clear in this report.
Prototype implementation  Also an important part of your design. You should describe how you implemented the prototype. This raises the issue of how much and what implementation you put in the report. That’s your decision, as to how you do that. Remember that the prototype is part of design: what did you learn from the prototype?

Physical deployment  The implementation of your design, as represented by your prototype will probably be a monolithic lump of code. This may not be how your final system should be implemented, so you are being asked here to consider the physical distribution of components of the system; the last step if you like before the next step, which would be the final implementation. UML offers deployment diagrams to assist your depiction of deployment.

The eBay system obviously offers much opportunity for distributed deployment, but your prototype may not.

Requirements tracing  Can you trace the requirements through your design? Explain how you did this.

Project management  How did you manage the project?

Special features  What are the special features of your design? Explain why they are special. Remember that features that are part of requirements are not special features, but you may be providing that feature in some special way.

Reflections and Introspection  This workshop and project is not simply a technical exercise in design; it should be a learning exercise. Discuss what you have learnt;

- What did you attempt but did not achieve?
- What would you do differently if you were to do it again?
- What did learn about design?
- What errors/faults/bugs did you experience? Why did they happen? Did you learn how to prevent them in the future?
- What did you learn as a team, and about teamwork.

This is not an invitation to waffle; it’s an invitation to critically evaluate your team’s experience over the semester. Nor is it an invitation to view your experience through rose-tinted glasses: you can comment on things you could have done better, and on things that we could have done better. You are not going to be penalised for revealing faults.

Note on narrative structure  The above headings indicate content, not necessarily the structure of the document. You may wish to mix or weave different aspects of the story. In the past the best reports have managed, very effectively, to mix and link all the parts together. It’s your story.