10 March 2014

**INB201 Project – Required Documentation - 2014**

Overall we will try to minimise the amount of documentation – a guiding principle for Agile software development. We will document only the important things.

*Think of this documentation as material that is going to be read/used by real-world clients, not just the staff at QUT.*

This document provides more information as to what is required as part of your team’s final submission at the end of Week 14, i.e. in the **What to Submit** section (near the end of the **Project Description**), the item listed as **Product Documentation, in hardcopy, as well as a softcopy in MS Word or PDF format**.

As such, it *excludes* these two documents which you will use *during* the semester:

* Product Backlog (which may be empty or nearly empty, by the end of the semester);
* Fortnightly self reflection and peer reviews.

As you know, the end of semester is a very busy time for most people. Many units have assignments due in Week 13. *To avoid a major panic at that time, your team should be developing this documentation during the semester, rather than at the end.*

Your team can please itself whether it: (a) submits the following items as parts of one large, single document; or (b) submits separate documents for at least some of the parts. While you may find it easier to have one or more team members working on separate documents, at any point in time, your team must make sure that it has some way of keeping track of which version is the latest.

You should number the sections of your documentation in accord with the numbers given below, i.e. 1, 2, 3, etc. Where you wish, you may use numbered subsections of your own choosing, e.g. 2.1, 2.2, 2.3, etc.

1. Project overview - A summary of critical information such as the vision for the system, and the benefits and risks of the system.
2. A Statement of Contribution – who did what in the entire project, including the documentation. *In the hardcopy, this must be signed by each member of your team*, but this is not needed in the softcopy.
3. Functional Requirements - This defines what the system does. This includes user roles, typical and significant user stories, and may include prototype layouts of essential user interfaces (on paper, or PPT, etc.).
4. System Documentation / Design Specification - This should provide an overview of the system and help people from outside of the team to understand the overall structure of the system. This normally includes an overview of the architecture of the system – i.e. the main components of the system and the relationships between them.

As part of this, describe the technologies and tools used to build the system. If you have included any open source code components or special code libraries in your system, give a brief description of each such item and the web address from where you obtained that item. This includes any content management frameworks (CMS), or web application frameworks (WAF) that you may have used.

You do not need to include very detailed design information in this section.

1. Database Design – This should include a diagram of your database structure, showing the table names and relationships between those tables, i.e. the foreign-key to primary-key relationships. You do not have to show the names of all the columns in each table.

For each table, include a one or two sentence description of what that table contains. I.e. why it needs to be part of your database?

If you used a proper database design approach, such as E-R or ORM diagrams, include those, as well.

1. Code Quality Assurance – This includes coding standard(s) used for the project and whether any kinds of code reviews were done.
2. Acceptance Test Plans – How did you test your system? Did you use GUI testing, non-GUI testing, or both? List the specific tests you used. What errors did you find during testing? What known errors remain in the final system, if any?
3. Operations documentation - Information on the installation and configuration of the system; backup procedures; and any troubleshooting guidelines.

The information provided here will clearly vary a lot from team to team, depending on what software platform you have chosen to use. While you are not expected to write an installation program (or “installer”), you might include one or more installation scripts on your CD/DVD ROM or USB device, and provide associated instructions in this section, e.g. a script for creating all the tables in your database.

1. User Guides - These describe how to use the system from an end-user’s perspective, including screenshots.

To support different user roles, you may decide to have different sections within the same User Guide, or you may decide to have different User Guides for (at least some of the) different roles.

Keep in mind that your User Guide(s) must be easy to use by someone who has never seen your system before, e.g. for on-the-job self-training. Instructions should be ***task-oriented*** *–* that is, written for specific tasks that users must perform – rather than simply listing what each screen, menu item, button, etc., does. The latter approach is a common mistake by naïve developers. They see their system only from a developer’s point-of-view, rather than an end-user’s. And then they wonder why the users don’t like their User Guide(s) and their system.

Your User Guide(s) need not be limited to hardcopy. You can use videos for some or all parts of it. But again, make sure that your videos are task-oriented – preferably one task per video – rather than being long and boring descriptions of what each button does, etc.

1. Sprint backlogs, burn down charts – This is an optional section. As these items have been covered in the Sprint Meetings during the semester, you do not have to include them in your final documentation. But some teams may like to include them for their own future benefit, i.e. as a way of keeping a record of what your team has done, for reference when working on future projects.