**CS4473B/CS9551B**

**Requirements Engineering**

**GROUP TEMPLATE**

**Reading Summary and Questions and Answers**

**Rules – please note these carefully:**

● Submission filename MUST be: **“Group”<id>\_”Chapter” (or reading) <id> (e.g., Group 3\_Chapter 2)**

● This template is similar in style to the Individual template.

o However, there is a new section (Part 3) on capturing concepts, entities, relationships, etc., which would be handy for creating a domain model.

● Group deliberates over the Individual Templates from the group members and creates a Group Template that is the shared view of the group members. Source material can be from one or more Individual Templates, adapted, or entirely newly created by the group.

● Pay particular attention to the “Comment” section as this records the group’s thinking.

● **Submission to be done on OWL as announced.**

● Group Template will be assessed.

**Part 1: Summary**

| **Group No:2** |
| --- |
| **List here the Group Members actually present in the class (absentees will be penalised):**  **Group Member** Name: Sihui He  **Group Member** Name: Chun Yang  **Group Member** Name: Yulun Feng  **Group Member** Name: Yuhan Zhang  **Group Member** Name: |
| Please write the **full reference** of the reading in the WHITE box below.  o Chapter #, Chapter title (or article title if appropriate).  o Book title  o Author(s)  o Publisher  o Book edition, Year of publication  (Example shown below; overwrite on that space.) |
| Chapter 10 – Documenting the requirements  Software Requirements  Wiegers and Beatty  Microsoft  3rd Ed., 2013 |
| Please write in the WHITE box below an abstract of the reading in **50-75 words**. |
| ***Abstract*:** **The chapter emphasizes agile approaches to requirements specification, highlighting user stories recorded on index cards or in tools. It explores the iterative refinement of details through team conversations, often represented by user acceptance tests. Nonfunctional requirements may be expressed as constraints or associated with user stories. The flexibility in choosing documentation forms and levels is underscored, with a focus on effective communication and the overarching goal of shared understanding. The abstract encourages reviewing project requirements, adopting suitable templates, and ensuring labeling conventions for successful outcomes.** |

**Part 2: Questions, Answers and Comments**

| Please create **ONE** important **Question-Answer-Comment set**  as agreed by the **group** from the given reading.  · Source can be from Individual Templates or completely new.  · The key is in discussing the individual templates and agreeing upon a shared view by the group. Prioritise what your group considers as a key issue to put forward. |
| --- |
|  |
| For staff use only:  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **QUESTION (state your question such that the answer is what you captured from the source):**  **What are the advantages and disadvantages of including user interface (UI) designs in the Software Requirements Specification (SRS)?**  **ANSWER (as-is):**  **<ebook><Ch 10, User interfaces and the SRS, #189>**  **Incorporating user interface designs in the SRS has both benefits and drawbacks. On the plus side, exploring possible user interfaces with paper prototypes, working mock-ups, wireframes, or simulation tools makes the requirements tangible to both users and developers. If the product’s users have expectations of how portions of the product might look and feel—and hence could be disappointed if their expectations weren’t fulfilled—those expectations belong in the realm of requirements.**  **On the negative side, screen images and user interface architectures describe solutions and might not truly be requirements. Including them in the SRS makes the document larger, and big requirements documents frighten some people. Delaying baselining of the SRS until the UI design is complete can slow down development and try the patience of people who are already concerned about spending too much time on requirements. Including UI design in the requirements can result in the visual design driving the requirements, which often leads to functional gaps. The people who write the requirements aren’t necessarily well qualified for designing user interfaces. Additionally, after stakeholders see a user interface in an SRS (or anywhere else), they will not “unsee” it. Early visualization can clarify requirements, but it can also lead to resistance to improving the UI over time.**  **YOUR COMMENT (also include where possible: an \*example\*, citation, justification, etc. -- to support your comment):**  **During my internship, I have personally witnessed the issues that arise from depending too much or too early on UI design in terms of requirement analysis. A typical problem is that when UI design precedes the detailed definition of core functionality and business logic, projects can fall into the trap of form driving function. In this case, teams may find themselves trying to fit an established interface design while ignoring the features or efficiencies that users need. In addition, once the UI design is introduced into the SRS and seen by stakeholders, it is often difficult to accept late changes, even if these changes are to improve the user experience or meet unforeseen needs.** |
|  |

**Part 3: Elements of the Architectural Domain Model**

| Please list below, in bullet point form, ideas that capture noteworthy information regarding artefacts, operations, conditions, relationships (e.g., produced-by, used-in, acts-on, etc.) from the assigned reading (and possible other sources – identify these). This could then be a source of information for creating your domain model. |
| --- |
| |  | | --- | |
| * The software requirements specification   + Labeling requirements   + Sequence number   + Hierarchical numbering   + Hierarchical textual tags * Dealing with incompleteness * User interface and the SRS * Software requirements specification template   + Introduction     - Purpose     - Document conventions     - Project scope     - References   + Overall description     - Product perspective     - User classes and characteristics     - Operating environment     - Design and implementation constraints     - Assumptions and dependencies   + System features     - System feature X       * Description       * Functional requirements   + Data requirements     - Logical data model     - Data dictionary     - Reports     - Data acquisition, integrity, retention, and disposal   + External interface requirements     - User interfaces     - Software interfaces     - Hardware interfaces     - Communication interfaces   + Quality attributes     - Usability     - Performance     - Security     - Safety   + Internationalization and localization requirements   + Other requirements * Requirements specification on agile project |
|  |