**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: Yifei Zhang |
| 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 14 – Beyond Functionality  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*: This chapter signifies the non-functional requirements in software development by emphasizing quality attributes beyond mere functionality. It outlines the necessity of considering external (user-facing) and internal (development-focused) attributes like usability, reliability, and efficiency. The chapter guides on eliciting, prioritizing, and specifying these attributes, illustrating their role in balancing software success and user satisfaction through practical examples and detailed discussions.** |

**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 Should business analysts consult software developers for nonfunctional requirements?**    **ANSWER (as-is):**  **<ebook><Chapter 14 Section “Implementing quality attribute requirements”> Yes If BA who lack development experience might not realize the technical implications of quality requirements. Therefore, the BA should engage developers who have knowledge of these implications and learn from those collaborations. Consider scalability, which can be profoundly affected by architecture and design choices. Scalability expectations can affect the hardware and operating environment decisions that developers make. This is why it’s important to elicit and document scalability requirements early on so developers can ensure that the product can grow as expected and still exhibit acceptable performance.**    **YOUR COMMENT (also include where possible: an \*example\*, citation, justification, etc. -- to support your comment):**  **Another reason for getting technical people involved in the quality requirement part of elicitation is related to the cost and expenses on the project in the future. For example, I worked in a startup company last year, they are using the MERN stack, and the PM in the founding team didn’t really think about the scalability issue and didn’t include developers for quality requirement discussion. In result, founding developers include all backend logics directly in the backend controllers. When the code base grows, it becomes hard to maintain, slow down the performance, and slow down the speed for development of new features. When I joined the team, we spent over 2 months on planning how to refactor the existing backend code to micro-services architecture to improve the scalability which will not affect the performance while keeping a moderate speed on developing new features. It also took over 3 months to complete the tickets related to the code refactoring. This does not cost company money for refactoring the code due to the scalability issue but also costs the extra effort from developers for which should be used for developing new features.** |
|  |

**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. |
| --- |
|  |
| * Quality attributes   + Functional requirement     - Installability     - Integrity     - Interoperability     - Reliability     - Robustness     - Safety     - Security     - Usability     - Verifiability   + System architecture     - Availability     - Efficiency,     - Modifiability     - Performance     - Reliability     - Scalability   + Design constraint     - Interoperability     - Security     - Usability   + Design guideline     - Efficiency     - Modifiability     - Portability     - Reliability     - Reusability     - Scalability     - Verifiability     - Usability   + Implementation constraint     - Portability |
|  |