**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: Yifei Zhang  **Group Member** Name: Yulun Feng  **Group Member** Name: Chun Yang  **Group Member** Name: Yuhan 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 28 – Change happens  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*: Software change is an inevitable thing, since we cannot define all a product’s requirements up front. This chapter discuss why we should manage the changes and how we could manage the changes. It gives us the basic concepts of the change control process and change control tools. It introduces how to measure change activities and the importance of change impact analysis. And final it tells us how agile projects handles change management.** |

**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 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:**  **How do we manage scope creep?**  **ANSWER (as-is):**  **<ebook><Chapter 28 section “Managing scope creep”>**  **“The first step in managing scope creep is to document the business objectives, product vision, project scope, and limitations of the new system. Evaluate every proposed requirement or feature against the business requirements. Engaging customers in elicitation reduces the number of requirements that are overlooked. Prototyping helps to control scope creeps by helping developers and users share a clear understanding of user needs and prospective solutions. Using short development cycles to release a system incrementally provides frequent opportunities for adjustments.”**  **“The most effective technique for controlling scope creep is the ability to say “no”. The president of one software tool vendor is accustomed to hearing the development manager say “not now” when he suggests a new feature.”**  **YOUR COMMENT (also include where possible: an \*example\*, citation, justification, etc. -- to support your comment):**  **It is very handy in agile project development to control the scope creeps since we have sprint which is short development cycles to incrementally add changes. During my coop internship, we also use prototyping to help with scope creep control by sharing a software prototype demo with client in meetings such that all stakeholders would understand clearly about the customers' needs and avoid unnecessary features. For example, we have implemented a feature about show the breakdown for rubric component and the initial thought of concern before meeting with clients was if we need have a lot of fancy small help features in the components. After we build a prototype and showcase to clients, we found out there are only limited number of helper features were really needed for customers. So, we reduce the scope of this new feature and save the time.** |
|  |

**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. |
| --- |
|  |
| **Change Control:**   * **1. Purpose and scope** * **2. Roles and responsibilities** * **3. Change request status** * **4. Entry criteria** * **5. Tasks**   + **Evaluate change request**   + **Make change decision**   + **Implement the change**   + **Verify the change** * **6. Exit criteria** * **7. Change control status reporting** * **Attributes stored for each request:**   + **Change origin**   + **Change request ID**   + **Change type**   + **Date submitted**   + **Date updated**   + **Description**   + **Implementation priority**   + **Modifier**   + **Originator**   + **Originator priority**   + **Planned release**   + **Project**   + **Response**   + **Status**   + **Title**   + **Verifier** * **The change control board**   + **CCB composition**   + **CCB charter** |
|  |