Description of the Requirements Engineering Process

The Requirements Engineering (RE) process that should be followed during this project is based on the TORE approach that has also been introduced in one of the GSE lectures (see the slides in "GSE_4_Software Application Engineering_IS_WS10.pdf" provided as supporting material for more details).

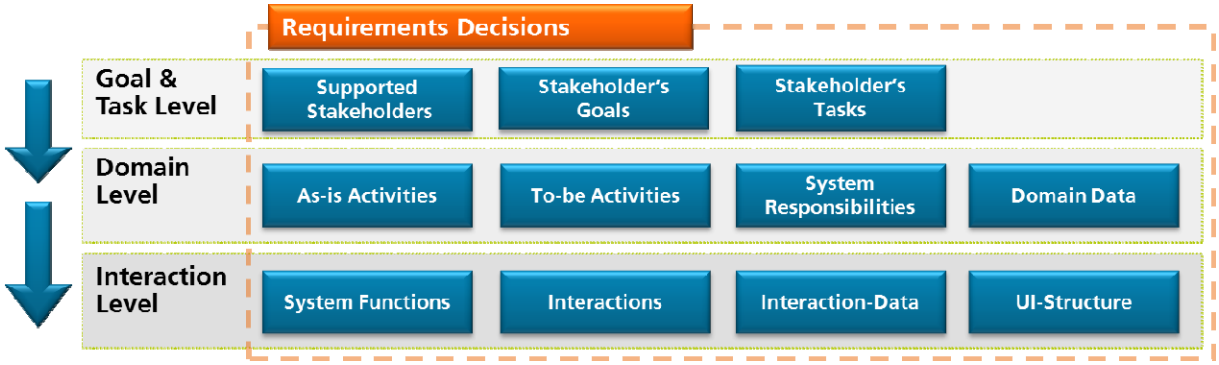

Figure 1: TORE Decisions

In order to systematically derive and specify the requirements based on the decision framework illustrated in Figure 1, you will basically run through the following phases: *Stakeholder Analysis*, *Goal & Task Analysis*, *Workflow & Data Analysis*, and *Interaction & System Function Analysis*. Furthermore, you should elicit and specify non-functional requirements in the corresponding NFR Phase.
Each of the phases will be described in more detail in the following sections.

1 Stakeholder Analysis

In general, supported stakeholders can be defined as (groups of) persons or (units of) organizations that have defined goals on a system to be developed or who are affected by a system to be developed.
The "Stakeholder Analysis" phase that you should perform within this project refers to the decision point "Supported Stakeholders" in the TORE decision framework (Figure 1), and covers decisions about the different roles that are to be supported by the system.
That is, *in the interview with the customer you should elicit information about all supported stakeholders, in particular, (groups of) persons and/or (units of) organizations that have defined goals on the user account management system including their particular roles and responsibilities.*
The output of this activity should be a description of the identified stakeholders by means of suitable descriptions, such as role descriptions and/or personas (see the slides in "GSE_4_Software Application Engineering_IS_WS10.pdf" for more details) in the requirements specification document.
Figure 2 illustrates a role description on the example of a department head.

ID	Role_1
Name	Department Head
Responsibility	management of a department and its employees
Success Criteria	incoming orders, department revenues, employee satisfaction
Typical Tasks	acquisition, personnel management and development, staffing, project planning, strategic planning, budget administration
Communication Partner	division head, staff, secretary

Figure 2: Role Description "Division Head" (Example)

2 Goal & Task Analysis

Goals usually describe high level objectives that different stakeholders would like to achieve with a system. Thus, the identified goals guide the subsequent activities, especially influencing decisions that have to be taken during the requirements refinement. Goals thus can be seen as rationale for the overall system, the different system requirements, as well as the different system functionalities. A *Task* can be considered as a sequential set of activities to achieve a specific goal under specific preconditions.
The "Goal and Task Analysis" phase that is described in the following refers to the decision points "Stakeholder's Goals" and "Stakeholder's Tasks" within the TORE decision framework illustrated in Figure 1 and covers decisions related to the goals of the stakeholders identified during the "Stakeholder Analysis" phase and the tasks these stakeholders perform as part of their typical work[1].

That is, *for each identified stakeholder (i.e., role) you should elicit in the interview with the customer the particular goals they want to achieve with the user account management system as well as tasks that should be supported by the user account management in order to achieve these goals.*
The output of this phase is twofold:

- A description of the identified tasks (see Figure 3 for an example)
- A goal graph visualizing the refinement of the goals and mapping to the tasks (see Figure 4 for an example)

Tips related to modelling of goal graphs:
Goal graphs allow hierarchical decompositions of goals. In the goal graphs you should create within this project, there is a distinction between three elements: Quality Goals, Functional Goals and Tasks. In most cases you will start with one or more Quality Goals. Refine them into more detailed Quality Goals or Functional Goals. Leafs of the graph should always be Tasks. Quality Goals describe something that should be easier, better, faster and so on. Within one or more Functional Goals each Quality Goal is solved. Finally, the Tasks are the operations which must be performed to reach the Functional Goals. Quality Goals have to be measurable in order to check easily whether they are achieved.
You may use MS Visio to model the goal graphs.

[1] In the context of this project you will only identify tasks that are related with the user account management system rather than analyzing all tasks related to their work as it is typically done with TORE in order to understand the whole working context of a particular stakeholder.

ID	Task_01
Name	Perform Business Trip
Goal	Attend business appointment in different location
Reasons	Meeting with business partner, presentation, rendering of a service
Priority	High
Execution Profile	Frequent, interruptions not common
Precondition	Travel date known
Input	Travel Date
Output	Reimbursement of travel expenses
Resources	Business trip management system, department head, secretary, employees, financial settlement, etc.

Figure 3: Task Description "Perform Business Trip" (Example)

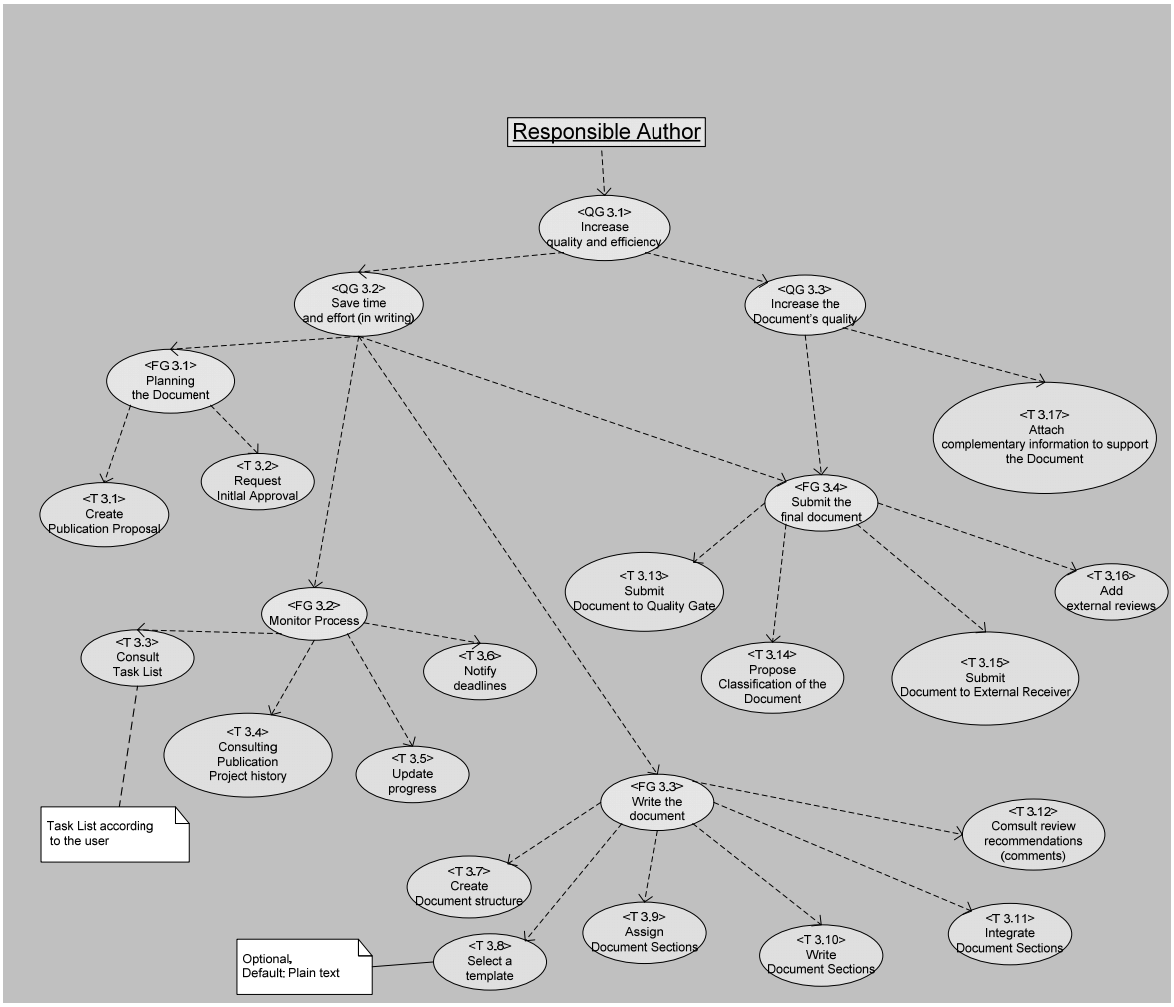

Figure 4: Goal Graph "Responsible author" (Example)

3 Workflow & Data Analysis

As already mentioned in the previous phase, tasks can be considered as sequential sets of activities (i.e., workflows) that are executed to achieve a specific goal under specific preconditions. In the "Workflow Analysis" phase you are now asked to analyse each task identified in the previous phase regarding activities that **currently** performed to execute the task (see Step 1: Identify As-Is Workflows) and how the execution of the tasks should look like in the future with the user account management system that you will develop within the scope of this project. (See Step2: Identify To-Be Workflows). Furthermore, you are asked to identify data that is relevant for the to-be workflow (see Step 3: Develop Data Model) and to identify system responsibilities (see Step 4: Identify System Responsibilities).
The TORE decisions (see Figure 1) related to this phase correspond to the decisions on the domain level, i.e., "As-Is Activities" and "To-Be Activities", "System Responsibilities", and "Domain Data".
The information required to accomplish this phase should be elicited in the interview with the customer and afterwards be specified in the requirements specification document.
That is, this phase basically comprises the following four steps:

***Step 1*: Identify As-Is Workflows**
Goal of this activity is to understand current workflows including problems that currently exist from the stakeholders' perspectives when executing the activities related to the tasks identified in the previous activities. This step is important for you to understand the actual workflow and to identify how the user account management system could improve the current workflows.
Output of this step should be a description of the As-Is workflows either in form of UML activity diagrams as illustrated in Figure 5 or Event-driven process chains (EPCs) modelled by using MS Visio. You will find a short tutorial of each of the notations in the supplementing material.

***Step 2*: Identify To-Be Workflows**
Similar to step 1, you are now asked to identify for each task the corresponding to-be workflows. That is, the sequential activities that are required to accomplish the tasks with the user account management system that you will develop. As in case with the As-Is-Workflows, To-Be Workflows should be described using UML Activity Diagrams or Event-driven Process Chains (EPCs).
In case that there are different variants of a workflow create one separate UML Activity Diagram / EPC for each variation. Model the diagrams as detailed as needed to understand the process and to create the use cases later on, but keep the diagrams as simple as possible.

***Step 3:* Develop Data Model**

In parallel to the identification of To-Be-Workflows, a data model has to be developed that captures all relevant data that is needed to execute the activities within the to-be workflow. The information related to the data model needs to be elicited in the interview with the customer. You may use a UML class diagram or Entity Relationship Diagram to model the data.

Tips related to data modelling:

- Collect terms (e.g., from the to-be workflows)
- Differentiate between classes and attributes
  - Classes have their own identity, several attributes, e.g., travel application
  - Attributes reflect a (possible changeable) value, e.g., date, location
- Assign attributes to the classes
- Define relationships
  - Reflect dependencies (often verbs), e.g., submits (employee – travel application)
- Determine multiplicities

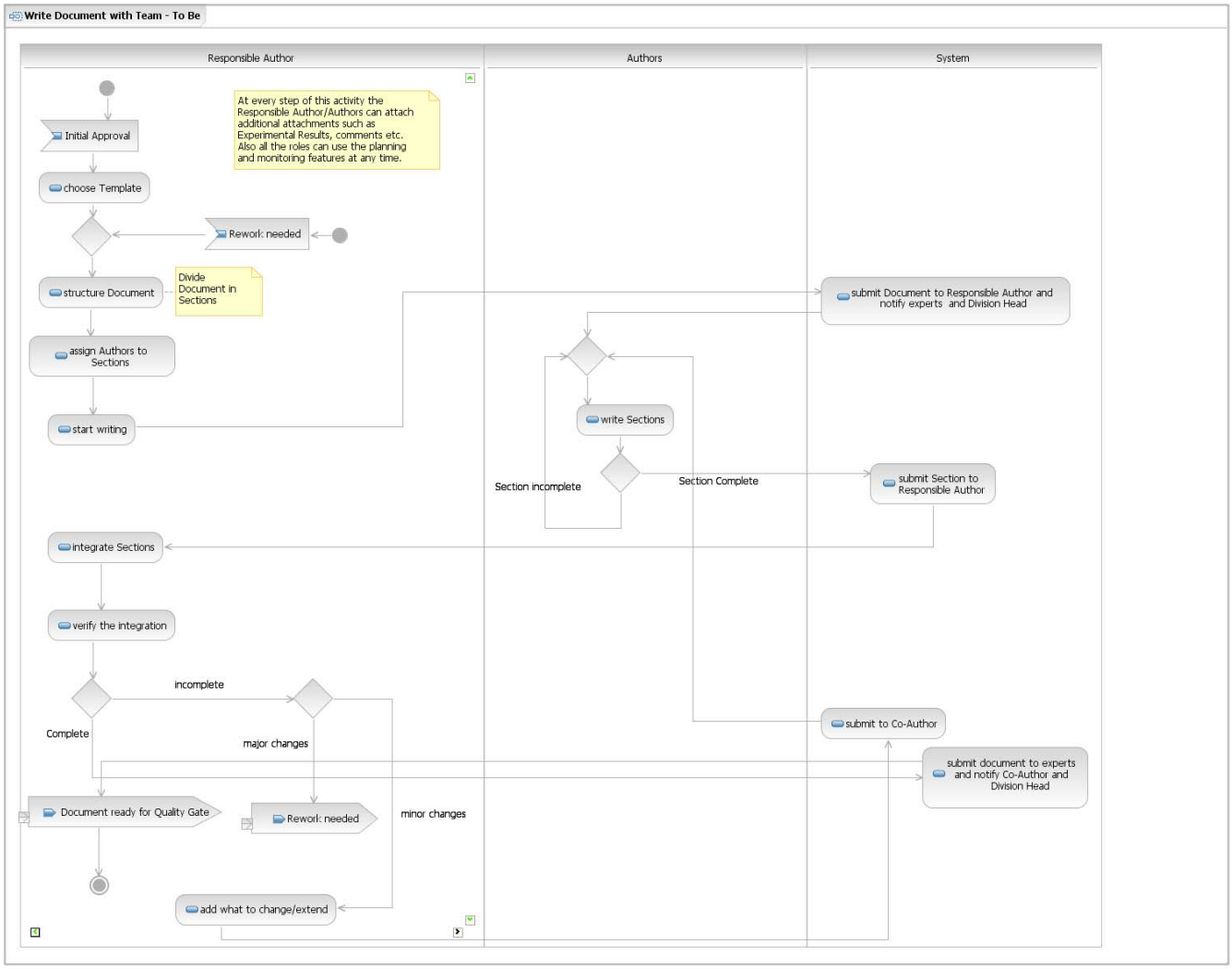

Figure 5: As-Is Workflow Diagram (Example)

Step 4: Identify System Responsibilities

Finally, after all To-Be Workflows are identified you are now able to identify system responsibilities, i.e. key contributions of the system. System responsibilities can be identified by classifying activities in the to-be workflows according to the following classification-scheme that depends on the type of underlying interaction:

- *Human-Activities*: these activities are only executed by a human actor without any system involvement. They are not further specified.
- *Human-System-Activities*: these activities require interaction between a human actor and the system. They are further specified in form of Use Case Descriptions in Step 1 of 4 Interaction & System Function Analysis.
- *System-Activities:* these activities do not require any human interaction but are only executed by the system itself. They are further specified in form of System Function Descriptions in Step 4 of 4 Interaction & System Function Analysis.
- *System-System-Activities:* these activities require interaction with external systems that are outside the scope of the system to be developed. They are further specified in form of Use Case Descriptions in Step 2 of 4 Interaction & System Function Analysis.

In order to visualize this classification in the requirements specification document, you may use a Use Case Diagram modelled with MS Visio or the scheme illustrated in Table 1 below. In order to identify all Activities, give them unique numbers (do not number the Human-Activities).

	Human	*Human-System*	*System*	*System-System*
A01	**X**			
A02		**X**		
A03				**X**
A04				
…				

Table 1 : Example table for the classification of the Activities

4 Interaction & System Function Analysis

In this phase the system responsibilities identified in *(Example)*

3 Workflow & Data Analysis that require interaction (i.e., Human-System Activities, System-System Activities) are further specified in form of detailed use case descriptions. Goal of this phase is to identify and describe all relevant system functions that will finally be implemented. Furthermore, interaction data has to be identified and modelled. The TORE decisions (see Figure 1) related to this phase correspond to the decisions on the interaction level, i.e., "System Functions" and "Interaction", and "Interaction Data". This phase can be subdivided into the following steps:

***Step 1*: Specify Human-System Activities**

In this activity it is specified how the user interacts with the system. This is achieved by using Use Case Descriptions as illustrated in Figure 6. All steps that are executed by the system correspond to system functions that need to be described in step 5 and are finally being implemented.

***Step 2*: Specify System-System Activities**

As in case with Human-System Activities, System-System Activities have to be specified in form of Use Case Descriptions with the external system as actor.

***Step 3*: Model Interaction Data**

Additionally, data has to be specified that is exchanged between user and system as well as external system and system during their respective interactions using the UML class diagram notation. For this purpose you might use the domain data model and update / supplement the model by new data that you identify during the interaction specification in step 1 and step 2.

***Step 4*: Describe System Functions**

In this step you should describe **all** system functions that you identified in more detail. That is:

- *System Activities* (already identified in Step 2 of (Example)
-
- 3 Workflow & Data Analysis)
- *System Functions* identified in Use Case Descriptions of Human-System Interactions in 4 Interaction
- *System Functions* identified in Use Case Description of System-System Interactions 4 Interaction.

Note: every action from the system in the use cases must be explained here. The System Functions are the functions that must be implemented later. That is why the System Functions must be explained as detailed as possible.

***Step 5*: Generate a Traceability Matrix**

A Traceability Matrix is a table that correlates the Goals and Tasks with the Use Cases and System Functions. This is important to guarantee the completeness of the Use Cases and System Functions. All Goals must be reached with the new system. The Traceability Matrix must be adapted and updated during the rest of the RE Process.

In order to describe the functions use the template as illustrated in Table 2.

Identifier	*A Number or unique name*
Name	*short description of the function*

Input Data	*What data is processed by the function*
Output Data	*What data is produced or changed by the function*
Description	*How is the result normally calculated*
Exceptions	*What kind of exceptions exists? What happens in case of an exception?*
Rules	*Complex causal or functional relation during calculation (if necessary)*
Quality Attributes	*What quality attributes need to be considered.*
Preconditions	*state of system and environment form the perspective of the actor before execution of the function*
Postconditions	*state of system and environment form the perspective of the actor after execution of the function*

Table 2 : Template for System Function Description

Identifier	1.0.3.1
UseCase	Receive Emergency Notification
Actors	Emerge System, Dispatcher
Intent	Dispatcher receives emergency notification/alert and according information
Preconditions	EMS gets called after emergency happened
Flow of events	1. The system notifies the dispatcher about a new emergency 2. The dispatcher acknowledges the alert. 3. The system shows the dispatcher a. The name of the person that has an emergency b. Type of emergency c. Severity [not available] d. Address of person e. Room where 4. The dispatcher … 5. …
Exceptions	3: [not available] Information on severity is not available: 3.1The dispatcher activates the camera in the room where the emergency happened. 3.2 The system shows the current situation in that room.
Rules	• It must be guaranteed that there is a dispatcher available all the time 24/7.
NFRs	Reliability: 1: all alerts that are triggered must arrive at the dispatcher! Availability: This use case must be available 99,999% of the time. Response / usage time: 1.The time between detection of an emergency and the notification of the dispatcher must be less than 5 Seconds. Usability: 3: The information for the dispatcher must be displayed on one screen without navigation effort.
Postconditions	The alert is acknowledged and the dispatcher has the necessary information for dispatching a rescue team.
Data	The name of the person that has an emergency Type of emergency Severity [not available] Address of person Room where
System Functions	Alert notification, Alert Information display, Room View

Figure 6: Use Case Description (Example)

5 NFR Analysis

Finally, you should elicit and specify non-functional requirements. The NFR Analysis phase distinguishes between quality attributes (QA) and non-functional requirements. A QA is a non-functional characteristic of a *system*, *user task*, *system task*, or *organizational* aspect. Table 3 provides a short definition of the various QA types and indicates relationships to system responsibilities you already identified in Step 4 of the Interaction & Data Analysis phase:

Quality Attribute Type	Definition	Corresponding system responsibilities
System QA	QAs related to the system as a whole and its subsystems (e.g., related to the database, secondary storage or network). Example: Capacity	(none)
User Task QA	QAs related to identified human system activities (use cases as a whole) in the to-be workflow. Example: Usage Time	Human-System Activities
System Task QA	QAs related to system tasks, i.e., tasks that are carried out by the system, not including the user any more. This could be referred to steps in a use case that are executed by the system Example: Response Time	System Activities
Organization QA	QAs related to organizational aspects. This also includes development process related aspects, such as required documentations, reviews, etc. Example: Standard conformity.	(none)

Table 3: Quality Attributes Types

A non-functional requirement (NFR) on the other hand describes a certain value of a QA that should be achieved in a specific project. The NFR constrains a QA by determining a value for a metric associated with the QA. For example, the NFR “The database of the new system shall handle 1000 queries per second” constrains the QA “workload of database”.

In order to elicit and specify the non-functional requirements you should execute the following two steps:

Step 1: Prioritize Quality Attributes

The first step towards the elicitation of non-functional requirements is the prioritization of quality attributes. For this purpose, you should go through the list of quality attributes (see file “ListOfQAs_Questionnaire.xls”) in the interview with the customer and let the customer rank each of the statements with priority “1” (= important for the system) or “0” (= unimportant for the system).

Step 2: Elicit and specify non-functional requirements

In the second step you should elicit concrete and measurable non-functional requirements. You might either do this already in the first interview with the customer together with step 1 (i.e., let the customer directly concretize the QA assigned with priority 1 by means of non-functional requirements), or discuss the NFRs separately with the customer as soon as you have finished the workflow analysis phase.

Finally, the elicited NFRs must be included into the requirements specification. Depending on the respective QA type, please annotate the elicited NFRs either in the To-Be workflows, the Use Case descriptions, or in a separate chapter in the requirements specification document according to the following scheme:

- NFRs related to System QA / Organizational QA: Specify in **separate chapter**
- NFRs related to User Task QA: Specify in **To-Be Activity Diagram** (annotate them as notes to activities)
- NFRs System Task QA: Specify in **use case description** (in the corresponding row in the description template)