Software Development Project – Requirements Analysis

Overview

In this activity, students will analyze the requirements for the online learning management system, in order to understand them, possibly improve them, and extend them with additional quality requirements. Each team will produce a utility tree for quality scenarios and will model the security requirements using abuse cases.

Each team member will fairly contribute to the software requirements analysis activity. The requirements analysis lead, together with the project manager, are responsible for the coordination and timely completion and submission of the required deliverables for this activity. The effort spent in this activity needs to be recorded, and project progress needs to be tracked against the plan. Risk management has to be performed throughout this activity.

Description

Part 1

Students will review the provided ***Software Requirements Specification (SRS)*** document(s), and perform analysis, in order to understand the requirements for the system to be developed.

These requirements are not perfect, as it happens in real life. When you review them, keep in mind qualities of good requirements listed below:

- Clarity
- Consistency
- Completeness
- Verifiability (testability)
- Feasibility

For reviewing use cases, you can employ, as review criteria, the Formation Rules and the Heuristics (see Chapter 6 of the class textbook by Fox):

Use Cases Formation Rules
Every use case diagram must have:

- At least one use case
- At least one actor
- At least one actor associated with each use case
- At least one use case associated with each actor
- No association line between actors
- No association line between use cases

- Name every actor and use case
- Not label any association line

Use Cases Heuristics:
- The product (software that you are developing) shall not be an actor.
- Actors should be named with noun phrases.
- Use cases should be named with verb phrases.
- An actor's goal should be achieved in one use case.

Additional use case review considerations:
- Review the stakeholders-goals list to make sure no actors are missing.
- Review the needs list to make sure no uses cases are missing.
- Review constraints and limitations to make sure they are no violated.
- Check that the collection of use cases covers all externally visible behavior.
- Check the diagram against the use case heuristics above.

You are already familiar with a reference/gold standard online LMS, that is, the ELMS/Canvas used by UMD. Use this system to supplement your understanding of the features and requirements for the software you will need to develop. Keep in mind, though, that your software will not be an exact replica of the UMD ELMS; your software will have smaller scope, different attributes, and most probably a different design and implementation.

As you review the SRS, identify and document issues and assumptions you might uncover.

Part2

Identify the quality attributes and respective scenarios for the software to be developed. Assign a priority (**L**ow, **M**edium, or **H**igh) to each scenario. Construct a utility tree for the identified attributes and scenarios. Include the utility tree in the SRS, under the section *Non-functional Requirements*.

If the SRS already contains quality attributes and scenarios, you could use them as a starting point, if/as you find appropriate. However, your team will need to develop your own utility tree. Remember that you will use this as a driver for your design.

Part3

Identify the security requirements and represent them in the form of abuse cases model. This model will consist of a) an abuse case diagram (using a UML Use Case diagram) and b) description of each abuse case that appears in the diagram. This description can be textual/natural language (using the template provided in class) or graphical (e.g., using UML activity diagrams). You can combine the two notations, describing some of the use cases textually, and others graphically.

Use the above rules, heuristics, and considerations for developing the abuse cases.

Remember that you will use this as a driver for your design.

Deliverables and Schedule

The deliverable for this phase is a MS Word or pfd file containing:

- The quality utility tree and
- The abuse cases model

This file must be submitted through ELMS (Canvas), **by the deadline specified in ELMS**. When you work with large files, please zip them before uploading in ELMS.

Peer review will be performed by the respectively assigned *Reviewer Team* on the utility tree and the security requirements represented as abuse cases. The Reviewer team must record issues and questions in ELMS, and discuss them with the *Author Team*. The review criteria for use cases, listed above, should be used for the Team to Team (T2T) review.