Evaluation and Grading of TDT4240 Software Architecture Project

The grade on the final delivery in the project is calculated accordingly:

$$\text{Grade} = (\text{Req}_{\text{Grade}} + 2^*\text{Arch}_{\text{Grade}} + \text{Impl}_{\text{Grade}} + \text{Game}_{\text{Grade}})/5 + \text{Adjust}_{\text{Complexity}}$$

The grade of the project be the average grade of the requirements, architecture, implementation documents as well as the implementation, where the architecture will have double weights (2X). In addition, you can get from -2 to +2 adjustment for complexity of your implementation.

Here is a description of all the components and description of what the evaluators look for:

$\text{Req}_{\text{Grade}}$ 0-60 points

Evaluation of the content and quality of the requirements document with extra emphasis on functional & quality requirements and COTS.

-Functional Reqs: Complete, prioritized, and defined prioritization

-Quality Reqs: Relevance, good coverage, correctly defined and measurable

-COTS: Complete and specific description of how COTS affect or put restrictions on the architecture

$\text{Arch}_{\text{Grade}}$ 0-60 points - (2x) double weight. In total 120 points

Evaluation of the content and quality of the architecture document with extra emphasis on tactics, patterns, views, and rationale. The architectural views are especially important.

-Arch drivers: Relevant and good coverage

-Stakeholders & Concerns: Good coverage and relevant concerns

-Selection of view: Include all views, all notations used and who and what addressed

-Tactics: List only relevant tactics and tactics you plan to implement. Describe why tactics are chosen. No need to describe tactics in detail if they are from slides or the textbook.

-Patterns: List the design and architectural patterns you will use and why they were chosen. No need to describe patterns in detail if they are from slides or the textbook.

-Views: Provide at least logic, process, development, and physical view with sufficient detail to be useful. Provide both diagrams and textual description. Include how patterns and tactics are used in diagram, text or both. If diagrams get too complex, provide several diagrams with decomposition. Diagrams must be readable.

-Consistency: State if views are consistent or not (e.g., use of names or parts left out).

-Rationale: Provide a motivation for your architecture (structures, choice of tactics and patterns).

$\text{Impl}_{\text{Grade}}$ 0-60 points

Evaluate the design, implementation, and test document with extra emphasis on Design & Implementation, test report and relationship to architecture.

-Design & Implementation: Provide design details not described in the architectural documentation and specifically, how the prototype was implemented.

-User manual: Explain how to install, run, and play the game in an illustrated and understandable format.

-Test report: Report the results from testing all functional and quality requirements according to the provided template. If some requirements are not tested, just put "Not tested" in the table for that requirement. For a test to be a success, it but meet the requirements. If not, provide a comment why. It is not a problem that several requirements are not met.

-Relationship with architecture: Is the implementation according to the documented architecture. If not, describe the differences.

$\text{Game}_{\text{Grade}}$ 0-60 points

Evaluation of the prototype and code with focus on functionality of the app/game, code structures and use of patterns in code vs. architecture, and how easy it is to install/run the app/game.

$\text{Adjust}_{\text{Complexity}}$ Adjustment for level of complexity: From -2 to 2 points