FURPS

- **Functionality:** feature sets, capabilities
- **Usability:** aesthetics, UI consistency, docs/help
- **Reliability:** frequency/severity of failures, MTBF
- **Performance:** speed, throughput, response time
- **Supportability:** extensibility, maintainability, compatibility, localizability, configurable…

Functional Requirements

1. The system shall allow the user to select, open, and run a UVSim file with appropriate inputs and outputs.
2. The system shall provide a user-friendly GUI for interacting with UVSim that has buttons and menus for executing operations.
3. The system shall allow users to input data into UVSim through the GUI.
4. The system shall display the output from UVSim to the GUI.
5. The system shall read a word from the keyboard into a specific location in memory if the first two digits of the BasicML equals 10.
6. The system shall write a word from a specific location in memory to the screen the last two digits of the BasicML equals 11.
7. The system shall load a word from a specific location in memory into the accumulator the last two digits of the BasicML equals 20.

8. The system shall store a word from the accumulator in a specific location in memory the last two digits of the BasicML equals 21.
9. The system shall add a word from a specific location in memory to the word in the accumulator, leaving the result in the accumulator the last two digits of the BasicML equals 30.
10. The system shall subtract a word from a specific location in memory from the word in the accumulator, leaving the result in the accumulator the last two digits of the BasicML equals 31.
11. The system shall divide the word in the accumulator by a word from a specific location in memory, leaving the result in the accumulator the last two digits of the BasicML equals 32.
12. The system shall multiply a word from a specific location in memory to the word in the accumulator, leaving the result in the accumulator the last two digits of the BasicML equals 33.
13. The system shall branch to a specific location in memory if the last two digits of the BasicML equals 40.
14. The system shall branch to a specific location in memory if the accumulator is negative and the last two digits of the BasicML equals 41.
15. The system shall branch to a specific location in memory if the accumulator is zero and the last two digits of the BasicML equals 42.
16. The system shall stop the program if the last two digits of the BasicML equals 43.

Non-Functional Requirements

1. The system shall maintain performance to ensure smooth execution of BasicML programs even on low-end hardware.

2. The system shall have a GUI interface that follows industry-standard design principles that will have properly labeled controls and an intuitive workflow.
3. The system shall be secure to prevent unauthorized access to sensitive data.