# SQA - Software Requirement Evaluation: IN-CLASS Team Activity#1

Gio Monci, Joseph L, Derek M, David R

Department of Engineering, Florida Gulf Coast University

CEN 4930 - CRN: 4930 - Software Quality Assurance

Dr. Buckley

1/28/24

**SQA - Software Requirement Evaluation: IN-CLASS Team Activity#1**

1. **What metrics will you be using to evaluate the requirements for quality?**

- Completeness: gauge if all necessary requirements are included, this will make sure that the system's functionality is fully conveyed.
- Clarity: check to see if all requirements are unambiguous, easy enough for a kid to understand, and does have any conflicting interpretations.
- Feasibility: determine whether or not the requirements can actually be realistically implemented given our constraints (ex: time, technology, budget)
- Testability: make sure that every requirement can be verified through some sort of testing or inspection.
- Consistency: Make sure that the requirements don't contradict each other or any part of the document.

2. **What the standard you will use to:**

    1. **Assess the requirements.**

        - IEEE/ISO/IEC 29148-2018: I was first looking at IEEE 830-1998, But it got superseded by ISO/IEC/IEEE 29148:2011, which then got superseded by this new one. If I can't pay for it or find it free, then it will revert back to 29148:2011.

        - Requirements check against CUPRIMDSO framework (Correctness, Unambiguousness, Prioritization, Realism, Importance, Modifiability, Dependency, Safety, and Operationality)

        - SMART criteria (Specific, Measurable, Achievable, Relevant, Time-bound)

        - GQM analysis: Goal/ Question/ Metric from requirements engineering

    2. **Assess software quality.**

        - ISO/IEC 25010: Using this standard to assess software quality because, this software quality model takes into account the best characteristics to evaluate when looking at software quality. **linked in references

    3. **Assess the domain of your project.**

        - Just like in Requirements Engineering with the pharmacy project, we will be referencing:

        - HIPPA, HL7, FDA SaMD

        - ISO 13485: quality standards for medical devices and software

- ISO/IEC 27001: security standards for handling patient data

**3. How will you determine which requirements are missing?**

- Use Case testing: create real world scenarios and go through each and figure out if we missed any functionality.
- Stake Holder Interview: get feedback from our client, users, etc. See if they think we are missing some functionality or if it's over the scope.
- Gap Analysis: Compare our requirements with industry standards and or from SRS / any documentation we can find of similar medical software.
- Risk Based Analysis: Research and brainstorm with team about potential risks associated with the software. Do our requirements cover these risks? If not, then implement.

**4. How will you determine if the existing requirements are compliant with industry standards?**

- Perform requirements audit: by mapping each requirement to a certain or relevant standard.
- NASA ARM Tool (Automated Requirements Measurement): analyzes requirements for various text-based properties such as depth, imperatives and directives.
- NASA FRET (Formal Requirements Elicitation Tool): help elicit and formalize requirements by inputting semi existing requirements into a hierarchal system form in a Json file and it should reformat / formalize the requirements if applicable.
- Find and talk to a domain expert: ask if they can review the requirements and provide any feedback and or give us information that we may not know.

# References

Metrics to evaluate requirements: https://www.ppi-int.com/wp-content/uploads/2019/05/Requirements-Quality-Metrics-Paper-with-Addendum-PPA-005330-9-140710.pdf

ISO/IEC 25010: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

NASA ARM: https://arm.laplante.io/

NASA FRET: https://software.nasa.gov/software/ARC-18066-1