EECE 491-493 Requirements

Introduction

Requirements are the statements we use to define a project. They are part of the systems engineering process that provide a detail definition of the project and a basis to show that the project is successful through verification. In this course, requirements are documented in the *Requirements and Verification Document*.

Requirements Writing Checklist

Most of these are derived from the "How to Write a Good Requirement – Checklist" in the NASA Systems Engineering Handbook.

1. **Determine an unambiguous name for the device or, if required, each piece of the overall system.**

   You may not use the term "*The device*" as it is not clear what "*The device*" refers to. So, you need to name the device. For example, "*The motor controller*". Or, "*The ASV*" as long as '*ASV*' has been defined.

2. **The term '*shall*' shall be used to define a requirement.**

   In a requirements document '*shall*' is the required term to use. It is similar to '*must*' but, by convention, '*must*' is not acceptable.

   Note that '*should*' does not define a requirement. The device '*should*' do something means it is not required to do something. Therefore, there shall be no *should* in a requirements document.

3. **Avoid ambiguity due to syntax.**

   The English language can introduce ambiguity if you are not careful. To avoid this all requirements are of the form:

   > "The *device* shall …"

   For example:

   > "**The ASV prototype shall** be completed by June 3, 2019"

4. **We will define and use two types of requirements in this course – *User Requirements* and *Technical Requirements*.**

   A User Requirement is determined by the user (or customer, or client, or stakeholder) while the Technical Requirement is determined by the engineer. Here are some basic rules:

   a. The set of User Requirements shall define the complete product. That is, if an engineer designed the product by blindly following the User Requirements and no other input or assumptions, there would not be anything missing.

   b. All Technical Requirements are based on a User Requirement.

   c. User Requirements and validation make sure you are building the right product. Technical Requirements and verifications make sure you are building the product right.

   d. All Technical Requirements shall be testable and will be verified. In this course, it is not required to validate the User Requirements.

5. **We will only define product requirements in this course.**

   More complex systems, like NASA's, have Personnel Requirements, etc. We will not and we will assume that we can not require the user to do anything. That is, there shall be no requirement that states "*The User shall…*"

6. **Technical Requirements must be measurable; therefore, they must include tolerances for values or minimum/maximum values, etc.**

   When writing Technical Requirements ask yourself, "will I be able to measure this?" For example:

   The device shall operate with a 12V power supply

   How do you measure this? A real instrument can not determine if the supply is exactly 12V. What if the supply is 11.9999V? Should the device work? In this case, we would likely have a tolerance like 12 ± 5%.

   Make sure the Technical Requirements are free of unverifiable terms (e.g., flexible, easy, sufficient, safe, adequate, accommodate, user-friendly, usable, when required, if required, appropriate, fast, portable, light-weight, small, large, maximize, minimize, sufficient, robust, quickly, easily, clearly, other "ly" words, other "ize" words)?

7. **Technical Requirements shall be free of implementation unless the implementation is required.**

   Requirements should state WHAT is needed, NOT HOW to provide it; i.e., state the problem not the solution. Ask, "Why do you need the requirement?" The answer may point to the real requirement.

   There are exceptions when, what appears to be, a solution is part of the requirement. Typical examples are communications protocols.

   A User Requirement may be:

   The device shall have a serial digital interface with four or less connections and can transfer 10MB/s.

   This looks like a good requirement as it does not include implementation. However, this allows the designer to create a proprietary communication system that is not compatible with anything. It may be the case that compatibility is a requirement. Therefore, the more correct User Requirement may be:

   The device shall follow USB full speed or faster and connect as a mass storage device.

   This looks like it has implementation (USB) but the USB protocol is actually part of the requirement.

8. **Requirements must be stated positively.**

   There shall be no requirements that say "The device shall not…" If you think about it, these are normally not testable.

9. **Rationale and assumptions shall be separate from the requirement.**

   Never include rationale or assumptions within a requirement.

   To save lives, the device shall meet UL requirements

   The rationale must be removed from the requirement. The requirement forms used in this course include a place to state the rationale and assumptions.

10. **A requirement must define single requirement, not be redundant or conflict with another requirement.**

    This means your requirement must have a single subject and single predicate. For example:

    The device shall measure ambient temperature with a resolution of 1 °C and a range of -10°C to 40°C

    This is actually two requirements, one for the resolution and one for the range. Also, note the accuracy is missing. Do not confuse resolution with accuracy. Resolution is normally a requirement of the display while the accuracy is normally a requirement for the data acquisition.

11. Do not over require.

Remember that requirements will need to be met and verified so if you over require something you may be causing undue complications or cost. For example, it may cause a conflict with the cost, time, or size requirement that cannot be resolved. A classic example of this is when someone uses the IC temperature range as their product requirement:

The device shall operate over an ambient temperature range of -40 °C to 105°C

This is the operating range of a popular MCU. So, the implementation, that MCU, is driving the requirement. Not only will this cause significant cost increases, but it likely makes no sense.

In the past, I have seen a requirement like this for a camping device, a boat, and a television remote. A person will not be camping or watching TV at 105°*C*! Let alone boating at -40°*C.*

The technical requirement for the operating temperature range shall first be derived from a user requirement, THEN the MCU range is compared to the technical requirement to see if it meets the requirement. For example:

User Requirement: *The product shall operate over the ambient temperature range of common temperature-controlled buildings.*

Technical Requirement: *The product shall operate over an ambient temperature range of 5℃ to 40℃.*

12. Make sure to include realistic constraint requirements.

Requirements for realistic constraints that are <u>required</u> for this course are cost, size, weight, operating temperature range, and completion date. Most projects will have more of these. For example, you may require a water resistance standard, or maximum noise requirement. There are many others depending on your project.

Examples of realistic constraint requirements:

User Requirement: *The product prototype shall be completed by demonstration day*.

Technical Requirement: *The product prototype shall be completed by June 7, 2022*.

User Requirement: *The device must fit in most clothing pockets*

Technical Requirement: *The PCB maximum dimensions shall be within 2.5” x 4” x 0.5”.*

User Requirement: *The device MSRP shall be less than $75*

Technical Requirement: *The BOM shall be less than $25.*

Requirements Documentation

1. Forms

The requirements and verification template, *ReqsVerfForms.xlsx*, is required to document your user and technical requirements and the verification of the technical requirements. The requirements are written first, during Fall quarter, and the verification is normally done towards the end of Spring quarter when the project implementation is complete. In this course, user requirements are not validated so the form is simplified to not include verification.

The template includes a sheet for the summary of all requirements and two sheet templates, one for user requirements and one for technical requirements. For each requirement, and each requirement revision, duplicate the appropriate form (user or technical) to create a new sheet in the order described below. All requirements are to be in this single Excel spreadsheet – this is your Requirements and Verification Document.

2. Numbering

Numbering of the requirements must be consistent, so they are bidirectionally traceable to-from higher-level requirements such as user requirements. Numbering is of the form: U.T.T…V

Where U indicates a user requirement, T are the technical requirement within U. This form allows you to have more than one level though, for this course, one level is the norm. Finally, V is a letter that indicates the version for that user or technical requirement.

During the start of Spring quarter there will be an opportunity to make revisions to your requirements. A revision is a change to a requirement that has been approved. So, if a requirement is not approved and you make a change, it is not a revision. Each revision has a version letter, starting at *a*. So, your first approved requirement will be version *a*. Each approved revision will increment the version letter.

Here is an example of the numbering and order for the requirements within the requirements document:

1.b	second version of user requirement 1
1.a	first version of user requirement 1
1.1.a	first and only version of technical requirement 1.1
1.2.c	third version of technical requirement 1.2
1.2.b	second version of technical requirement 1.2
1.2.a	first version of technical requirement 1.2
2.a	first and only version of user requirement 2

Etc…

You must include a revision form if you remove a requirement. In this case, the revision form simply indicates that the requirement has been removed with an explanation in the rationale. Once a requirement has been removed, the number(s) for that requirement can no longer be used. i.e. do not reuse the number.

3. Process

When you have new completed requirements added to the requirements document, place – or update – the document in the project PAL under the requirements folder. Send an email to the instructor indicating what has been added or changed. The instructor will review the document, provide feedback or approve the requirement(s). This is an iterative process until the end of Fall quarter and again at the start and end of Spring quarter.