Backyard Splashpad Specifications Document

Braydan J. Allen

James S. Humble

October 24, 2018

Acknowledgements

This template was developed and compiled from multiple sources. It owes its origins most firmly to a mil-spec template that I found on line years ago. That template contained several humorous lines about how boring specifications are and that cleanliness is next to Godliness. I had never laughed before when reading or writing a specification. I liked the author's approach.

The source of that original template is lost but the framework reappeared several years later as a Small Satellite requirements template but I am sure that it was a modification of the original. Next, this template also owes much of its structure to the brilliant book by Phillip Koopman 'Better Embedded Software' a book that describes a development process that is useful for a whole lot more than software.

Much of the current structure and helpful comments are the work of my colleague Jolynne Barrett. She has made the design process much less intimidating for Senior Design students. They tend to lose that 'deer-in-headlights' look after they realize that Jolynne will help them bear the heavy load of project documentation.

Preface

Requirements documents spring out of military/aerospace methods and are part of a formal process intended to minimize needed redesign. These systems design methods are heavily front-loaded requiring extensive pre-design effort. When properly exercised these methods can, and do, produce safe, effective systems that are on time and budget. Requirements documents help with the so-called concurrent engineering process in that all aspects of the project are specified at the beginning of the project down to and including the kind of box that the system is going to be put in. Concurrent engineering lessens the chances of unpleasant surprises late in the program. Hearing things like, "What do you mean it has to fit in a two inch cube?", six months into a program is frustrating and expensive. There are less formal methodologies that demand less up front work, but while it is easy to relax the formal it is often hard to go the other way.

Software developers often advocate design through methods like Extreme Programming or Agile. Some pieces of these methods are quite useful, e.g. users stories, but as a crusty old design engineer I do not like an approach that uses near constant redesign as a design methodology. When push comes to shove and lives are at stake, engineers will almost always choose a formal systems design methodology.

Many people say that this formal style of engineering development is passé and that newer, speedier, more flexible methods should be taught and used. And yet they are unknowingly depending on these formal techniques every time they step onto an airplane. To those who pine for the less formal methods I always ask the following question:

> *Would you fly on an airplane designed by the methods you advocate?*

The answer is seldom yes.

How to Use This Document

This document is divided into five major sections. While specification document formats vary from group to group the format presented here is a good representation of what you will encounter in industrial specifications. The sections of the document are:

1. *Scope*
2. *Applicable Documents*
3. *Stakeholder Requirements*
4. *Engineering Requirements*
5. *Verification of Requirements*

This document is not only intended to explain engineering specification documents and show the development of a specification through the use of a running example. It also aims to help you produce a specification for your project. In order to achieve this goal each section will be presented in three levels. I will:

1. *Attempt to explain the reasons that the section exists and what data goes into that section.*
2. *Complete that section for the running example and annotate the example to make it more understandable.*
3. *Strongly suggest that you complete the section for your project. When you see the*

symbol *it means you*

are going to get to produce something.

A skeleton template will be available in the same place that you found this document. Fill out that skeleton template as you follow through this document. You will quickly produce a specification in the least painful way I can think of. Note that I didn't say pain free. Sometimes thinking as hard as you need to think in engineering design just makes your head hurt.

Signature Page

Signature	*Signature*	*Signature*
Date and email	*Date and email*	*Date and email*

Revision History

Revision	Description	Author	Date	Approval
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

A simple hardware example will be used throughout this guide to illustrate the writing of a specification document. The USU ECE Controls Lab needs a new power amplifier. The requirements for its design follow the narrative in this document.

From experience I know that I will not get it all right the first time. If you find what you believe to be a mistake or omission, and can make a good case for why you feel that this is so, then I will revise the document and immortalize your name in the revision table.

Contents

Specifications

1 SCOPE

(a) **General: This document describes the design and verification requirements of the Backyard Splash Pad. The Backyard Splash Pad is used to provide backyard entertainment for adults and children alike.**

(b) **Acronyms:**

BSP: Backyard Splash Pad

2 APPLICABLE DOCUMENTS

This section is often poorly understood and poorly implemented. It is a list of documents that are a critical part of understanding the item or requirements imposed on the item. Every document listed must have a text reference in the body of the spec further describing and limiting how it is to be applied. Conversely, no document is to be referenced in the spec unless it is listed here. Don't put items here that are background information or of general interest. Always obtain and review all items listed here. This section often has the following statement:

> *The following documents shown shall form part of the specifications for this project. In the event of a conflict between requirements, priority shall first go to the contract, second to this document, and lastly to these reference documents.*

There are lots of MIL-STD(standards) and MIL-HDBK(handbooks) that cover an amazing range of subjects. Here is a website that has them plus NASA documents and others all available (every government specification and handbook you can imagine) for free. Other groups publishing standards include Institute of Electrical and Electronics Engineers (IEEE), American Society of Mechanical Engineers (ASME), Society of Automotive Engineers (SAE), American National Standards Institute (ANSI), American Society for Testing and Materials (ASTM) and Aeronautical Radio, Incorporated (ARINC) to name more than a few.

Another thing to note is that referencing documents will often save you time (and money). For example, a referenced document can contain a complete set of environmental tests. Stating that your system has to be tested to the requirements of MIL-STD xxx is a lot easier than making up the series of tests yourself. The process is akin to what you do when you use a library in C programming: someone has provided software that meets your needs. Why reinvent the wheel? Particularly in something as difficult to get right as environmental testing.

Another benefit to using standards documents is that it connects your project to what is typically done in industry. While this should be obvious it is sometimes forgotten. The designers that you hand the spec over to may already know the standard that you have specified and know how to meet these standards. The simple adherence to standard could save you a lot of money.

(a) **Government Documents** *This is where to put MIL-Specs, MIL-STDs, NASA specs and so forth. Be sure to include the revision level and date.*

(b) **Industry Documents** *This is where to put ANSI, ASTM, ASME, IEEE, Company specifications and so forth. Both this section and government documents can be divided up into logical subcategories.*

My project is pretty simple and I don't have any applicable documents, but your project might. Did you know that in order to use USB on a commercial device that you have to pay a licensing fee? I know, your development board has a USB port on it. You can use it because the board manufacturer paid the fee. The USB spec is about 1500 pages. Saying that your device must comply with this spec (or a part of it) is a lot easier than writing it all out.

Research and write down the external specifications that you want your system to meet. Think about standards (like USB) and MIL-Specs that cover things like environmental testing and electromagnetic compatibility.

3 STAKEHOLDER REQUIREMENTS

The stakeholders for the USU ECE Controls Lab Power Amplifier Module are:

1. Dr. Don Cripps
2. Jolynne Berrett
3. The Families of Braydan Allen and James Humble
4. The USU ECE Department

3.1 Stakeholders User Stories

The primary stakeholders needs are described below.

- **Dr. Don Cripps The device:**
    1. Must be complicated enough to challenge the designers.
- **Jolynne Berrett. The device:**
    1. Must have a useful, readable user manual.
- **The Families of Braydan Allen and James Humble will be the primary users of the device. The device:**
    1. Must not present safety/shock hazards to any user.
    2. Must have an easy user interface.
    3. Must be able to run in a typical backyard.
    4. Must have multi-colored lights.
- **The USU ECE Department is funding the project. The device:**
    1. Must be low in initial cost (design and prototype).
    2. Must meet the pedagogical requirements of the course (ECE 4820/4830/4840/4850) for which it is designed.

4 ENGINEERING REQUIREMENTS

4.1 Item Definition

The Backyard Splash Pad is comprised of four main components: a device controller, a user interface, a mechanical system, and an object-tracking system. The device controller receives input from the user interface and uses it to control the mechanical system. The user interface presents information about the system to the user and accepts input from the user. The mechanical system controls the flow of water. The object-tracking system detects objects near the Splash Pad and reports information to the controller. See Figure 1.

4.1.1 Functional Diagram

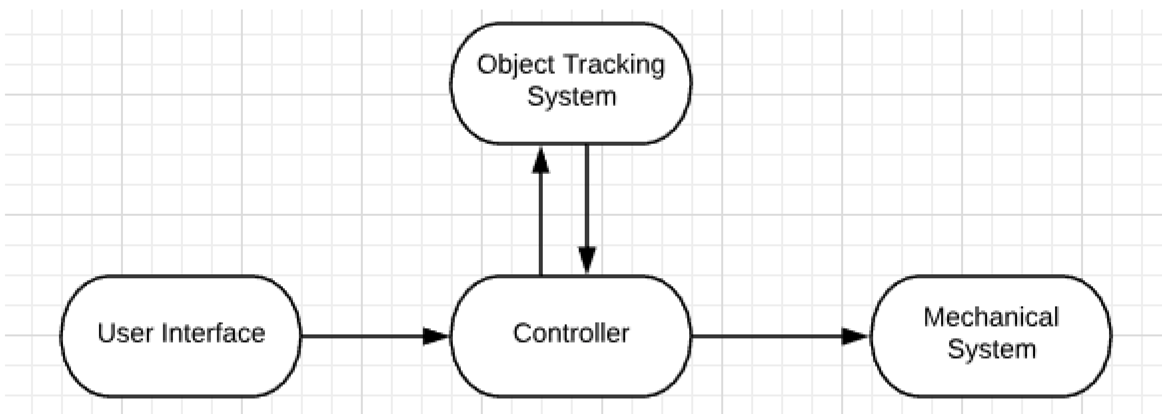

Figure 1: Functional Diagram.

4.2 Functional Requirements

4.2.1 Controller

4.2.1.1 The controller shall be able to turn on and off each nozzle individually.

4.2.1.2 The controller shall be able to turn on and off each light individually.

4.2.1.3 The controller shall be able to change the color of each light individually.

4.2.1.4 The controller shall be able to turn on a specified nozzle as directed by the Object Tracking System.

4.2.2 User Interface

4.2.2.1 The user interface shall allow the user to select what nozzle(s) are turned on.

4.2.2.2 The user interface shall allow the user to select what color the LEDs are.

4.2.3 Mechanical System

4.2.3.1 The mechanical system shall be able to create seven water streams that are 6.0 ft or less.

4.2.3.2 The mechanical system shall conform to the IP65 standard.

4.2.3.3 The mechanical system shall be able to operate continuously for 10 minutes.

4.2.3.4 The mechanical system shall not provide electrical shock to users.

4.2.4 Object Tracking System

4.2.4.1 The object tracking system shall be able to detect any object larger than an NBA regulation sized basketball that is within 2.0 ft of any nozzle.

4.2.4.2 The object tracking system shall be able to function when the sun is 20 degrees above the horizon.

4.3 Support Requirements

4.3.1 Device Controller

4.3.2 User Interface

4.3.2.1 The user interface shall require an android phone for more advanced features.

4.3.3 Mechanical System

4.3.3.1 The system shall operate on 1000.0 Watts or less.

4.3.3.2 The system shall be powered by a standard 120 V electrical outlet.

4.3.3.3 The system shall fit within a cube that is 10 ft on an edge.

4.3.4 Object Tracking System

4.3.4.1 The object tracking system shall fit within a cube that is 6 inches on a side.

4.3.4.2 The object tracking system shall have a case.

4.3.4.3 The object tracking system case shall have a mounting interface.

5 VERIFICATION OF REQUIREMENTS

We have to know that we designed what we were required to design by this document. This process is called 'verification' and it answers the fundamental question: "Did we build what we said we would build?". We verify requirements by testing to see if the requirements are met. Design teams must test every requirement in order to prove that the requirement is met. Testing each requirement means that each requirement must cast in some quantifiable way. In this section you specify how you will test each requirement to show that it has been met.

Don't worry! It is not as bad as it sounds. Sometimes (not always!) you can verify or test a requirement simply by looking at the completed system to make sure some required thing is present.

Testing often 'takes it on the chin' in terms of project schedule. Since integrated system testing typically occurs near the end of a project, the time for testing is compressed against the deadline. People start short-cutting tests to stay on schedule. Sometimes you may get away with it but it is never a good idea either technically or ethically. Epic failures have occurred because of truncated testing. One such failure occurred during the testing of the Hubble Space Telescope. The following is an excerpt from the the official report detailing the failure.

> *Reliance on a single test method was a process which was clearly vulnerable to simple error. Such errors had been seen in other telescope programs, yet no independent tests were planned, although some simple tests to protect against major error were considered and rejected. During the critical time period, there was great concern about cost and schedule, which further inhibited consideration of independent tests.*
>
> *The Hubble Space Telescope Optical Systems Failure Report-NASA November 1990*

If you are interested the whole report is available at (`https://www.ssl.berkeley.edu/~mlampton/AllenReportHST.pdf`).

The Hubble error wasn't caught until the telescope was deployed in space. Can you imagine the cost of fixing this problem? It is not simply a case of bundling you off with your instruments and putting you up in a fancy hotel for a week or two. Some estimates set the price at about $1 billion.

The Dilbert comic strip has a similar, and darkly amusing, view of testing truncation.

(`http://dilbert.com/strip/2010-08-21`)

(http://dilbert.com/strip/2009-07-01)

The key to completing this section is that every requirement has an associated test. The best practice in this section is to match the sub-paragraph numbers in the previous section to the sub-paragraph numbers in this section, e.g the requirement in 4.3.1.6 is covered by the test described in 5.3.1.6.

Possible verification methods include:

1. *Inspection:*
   Inspection is a method of verification consisting of investigation, without the use of special laboratory appliances or procedures, to determine compliance with requirements. Inspection is generally nondestructive and includes (but is not limited to) visual examination, manipulation, gauging, and measurement.

2. *Demonstration:*
   Demonstration is a method of verification that is limited to readily observable functional operation to determine compliance with requirements. This method shall not require the use of special equipment or sophisticated instrumentation.

3. *Analysis:*
   Analysis is a method of verification, taking the form of the processing of accumulated results and conclusions, intended to provide proof that verification of a requirement has been accomplished. The analytical results may be based on engineering study, compilation or interpretation of existing information, similarity to previously verified requirements, or derived from lower level examinations, tests, demonstrations, or analyses.

4. *Direct Test:*

   Test is a method of verification that employs technical means, including (but not limited to) the evaluation of functional characteristics by use of special equipment or instrumentation, simulation techniques, and the application of established principles and procedures to determine compliance with requirements.

5.1 Verify Coverage of Stakeholder Requirements

The tester verifies that everything that the stakeholders have asked for are covered by one or more requirements. It is a good idea for the requirements author(s) to perform a similar check at this point. The tester is likely to do his own analysis or disagree on points in yours, but the exercise itself is valuable. And if you do the analysis you might as well write it down here.

5.2 Interface

5.2.1 Functional Interface Constraints

5.2.2 Functional Interface Requirements

5.2.3 Support Interface Constraints

5.2.4 Support Interface Requirements

5.3 Functional Requirements

5.3.1 Functional Method Constraints

5.3.2 Functional Design Requirements

5.4 Support Requirements

5.4.1 Support Method Constraints

5.4.2 Support Requirements

A tabulation of all the requirements and the testing method with a blank space for results is useful for whomever is doing the testing.

Paragraph Number	Test Type	Tester's Name	Pass/Fail	Date

Now read over your completed specification and make additions and corrections. Find others who will be willing to read and comment on the specification (hopefully they will still like you when they are done). The more eyes the better. Ask yourself if you handed this spec to a competent classmate what would they build?

Congratulations! You have written an engineering specification and that is no mean feat.