Final Project: Part 1

# Software

I have chosen to do my final project on Analytic Systems’ PowerWizard software. PowerWizard is a proprietary piece of software that allows a user to change the default settings of a Intelligent series battery charger made by Analytic Systems.

The program has a variety of features, chief among them being the ability to adjust the voltage and current limits of their device in order to improve charging efficiency. The program allows the user to update their device’s firmware, and has built in graphing and data logging capability to monitor the voltage and current being supplied to the battery banks.

The battery chargers are designed primarily for automotive, military and naval purposes. While the charger is perfectly able to operate without the PowerWizard software, the battery charger set to a factory default assuming that the battery it is charging is a standard flooded lead-acid battery.

A real-life example of PowerWizard being used would be aboard a boat docked in port. Where a technician would connect the charger to a power source and charge the ship’s batteries, which will later power the vessel while at sea.

The software interface is very daunting to be honest. There are between 20-30 text fields and displays spread over three tabs and four sub-tabs. There is no built-in help system to ease a new user beyond grouping similar functionalities near each other. The interface is logically organized but requires understanding of the science and technology behind charging a battery to see that organization.

## Caveat

While PowerWizard is fully-functioning software, its unorthodox development cycle has led to an unpolished whole. Within the program, there are unfinished features, functionalities that were later removed in later versions, and the occasional bug. My point being that despite the software being functional it is still unfinished.

As I work on this Final Project, PowerWizard has currently been shelved so there are no expected changes. But this does raise the issue of maintaining documentation and version control. Using topic-based modules is a necessity to be able to delete sections related to removed features without impacting the information architecture or flow of the writing.

Drawing from what we’ve learned in class, task-based writing rather than interface-based is also very important. Both related to density of the interface as stated above but also because the interface is changing as the program is update. If the engineer should decide to completely the program interface, it could render documentation nearly unusable.

# User Roles

There are three user roles that I believe would be the typical users of PowerWizard. This knowledge is gleaned from first-hand observation of different groups of people I have seen using the software. But while I’m sure the user roles are exist, my assessments of their needs is based on assumptions.

## Unit Testers

Unit testers are the people who run the battery charger through a battery (Sorry! I had to) of tests in order to ensure they are functioning as expected. Unit testers use aspects of the software to test the charger functions and features. Unlike a field technician who may to use the software to different ends in response to changing work environment, a unit tester repeats a singular process usually in a lab or workbench.

I believe this group will be typically rote users, users accomplished at a specific repeated task. I predict they will have a high level knowledge base related to the charger units and fully capable at operating the software (the knowledge of one should help intuit the other). The type of documentation best suited to this role’s needs I imagine would be reference topics for troubleshooting malfunctions.

## Field Technicians

Field technicians are the people who are using the software in a non-factory setting. Simply put, this user role is the people who bought the software to use it to modify their battery charger.

I do not expect them to be a total novice user to using to the software as the charger and equipment are quite expensive and specialized so some training would be expected. However, this group will likely have the smallest knowledge base and would likely needing more conceptual and background information i.e.) Choosing which charging profile is best suited depending on needs, what an equalize cycle is and why it should be performed.

Unlike the two other user roles, this group likely has the most erratic work environment coupled with the least familiarity with the program. A field technician would have no use for a massive tome of references, their ideal documentation should be primarily procedural with some conceptual topics for context.

I predict that interviewing field technicians will give me the most insight on the structure and scope of the final document. I am particularly interested in the degree of detail and technicality that would best serve this user role’s needs, as the information I have above is an educated guess.

## Software/Hardware Engineers

Opposite to the Unit Tester user role, who are interested in PowerWizard in regard to using it change how a battery charger behaves, Software Engineers are focused on the software independent of it. For instance, if the engineer were designing a new battery charger for the company, they would likely want an in-depth understanding of PowerWizard’s code to ensure it ran properly with the firmware they were writing.

I believe this will be the user role with the largest knowledge base. They will be expert level users, valuing both high specificity and brevity. The type of documentation best suited to this role’s needs would be reference based for looking how certain lines of code translate to aspects of the software.

# Background and Interview Sources

The software’s extremely specific purpose and obscurity makes any form of information-gathering online a moot point. Simply not enough people have used it for any substantial help documentation to coalesce. Fortunately, I know the engineer who wrote the program so there should be no problem in gathering technical information regarding how to use the software. In addition, I have access to my coworkers who use the software as both Unit Tester and Software Engineer user roles for interviewing.

As for information related to how a battery charger functions and the electrical principles behind it, in addition to my co-workers I have found a great resource in *Batteries in a Portable World – A Handbook on Rechargeable Batteries for Non-Engineers” (4rd edition)*. The information in this book will be very useful when writing to the field technician user role. Even if knowledge of the principles isn’t necessary to use the software, I believe that giving the user a basic context will act as a way to ease the user into the process of using the software without overwhelming them. At the very least, the context should prove valuable if the documentation fails them and they need to resort to Googling a solution to an issue.