# Situation Assessment

The purpose of the PowerWizard software is to display diagnostic data relating the operation of Analytic Systems’ *Intelligent* series battery chargers. As well as, allow the user to control the charging parameters used by them. These chargers are designed for 12 volt and 24 volt batteries which are often used in methods of transportation. So we expect the business environment where PowerWizard is used to be in depots, garages, and aboard trains, camper vans, and motorboats.

This type of environment is expected to be loud, fast paced and hectic. The user will likely be operating the software through a laptop PC for the sake of portability. This is important since monitoring and controlling the battery charger would be only one aspect of this user’s whole job profile; also being responsible for managing several other tasks simultaneously. This business environment is typical for the **End User** user role.

The **Unit Tester** and **Software Engineer** roles will usually be using the software within a factory for the purposes of product development rather than product application. The Unit Tester’s end goal is using the software to test that the battery chargers function correctly before shipping them. The Software Engineer understanding the software well enough to write firmware which can interface with it.

This business environment is expected to be much slower paced than the other one; this work environment being stationary and having the user focused on a single task rather than having multiple tasks “on the go.” Additionally, this environment is expected to be friendlier, with alterative help resources (coworkers, engineers, other documentation) available should the user require them.

# Audience Assessment

## Unit Tester

Primary job tasks:

* Using PowerWizard to control battery charger operation
* Recording and documenting testing results
* Writing error reports for failed units

Expected level of technical proficiency:

* Novice user

This user role requires no technical knowledge to perform their job tasks. At its simplest this user just to follow a list of tasks in a precise order and record measurements. However, through proximity and repetition, the user will usually gain an understanding of how the battery charger and PowerWizard function to a shallow degree.

This user’s work environment is slow-placed and friendly as stated previously. This assumption coupled with the repetitive nature of the job and the easy access to help lends to a relatively low stress environment.

This user will not likely consult the documentation after initial training unless an error occurs and a reference for the purpose of writing a report is needed. The user may also seek out the documentation if a “refresher” after a long period of non-testing.

## Software Engineer

Primary job tasks:

* Coding firmware compatible to PowerWizard
* Testing and documenting firmware code

Expected level of technical proficiency:

* Advanced user

This user role easily has the largest technical knowledge base when it comes not only to operating the software (which they understand intimately) but also the principles that govern the operation of the charger and the science behind battery charging.

This user’s work environment is slow-paced like the unit tester but because of the abstract nature of coding and the high level of technicality, accessing help can be difficult. The level of stress in the environment is then dependent on how comfortable they are coding.

This user will be using the documentation religiously while coding the firmware, as knowledge of PowerWizard’s command outputs is mandatory when writing firmware to respond those outputs. The completeness of the documentation is vital in this regard, if a portion of the coding is outdated or missing, coding the firmware can range from infuriating to impossible.

## End User

Primary job tasks:

* Reading temperature, voltage and current data
* Recording diagnostic data
* Editing the battery charger’s charging parameters
* Creating new charging profiles
* Updating the charger’s firmware

Expected level of technical proficiency:

* Difficult to determine — Assumed Novice user

This user role is the most nebulous, its users can use the software for avariety of tasks greatly ranging in complexity determined by their individual needs. Speaking with an engineer on the topic, he has seen some customers use the software purely for diagnostics. These customers are strictly “plug and play”, they never touch any of PowerWizard’s parameter manipulation abilities which is arguably the largest portion of the program. Whereas, other customers are intimately familiar with these abilities being able to adjust charging parameters per *battery cell* compensated in response to temperature (A high level function of the program related to specificity and fine control)

This shows the difficulty in determining the amount of this user’s technical knowledge. To use the program a data readout, the only technical skills that the user would need are opening the program and then plugging a cord into a USB slot. Whereas, to use any features relating to charging profiles, the user should have a basic understanding of the function of the hardware and the science behind how batteries charge.

This user will likely be using this documentation *before* opening the program. Initially this user is expected to use the documentation extensively until all tasks they are memorized. After which, the documentation would only be revisited in the event of an error.

# Documentation Design

## Unit Tester

The unit tester has the most straightforward job tasks of the three user roles. They simply need to run an identical battery of tests for every battery charger in a given shipment. They input values in the software and measure current and voltage output in the hardware to confirm the charger is behaving properly.

Procedural documentation suits this task the best, a task based checklist would be a simple but effective way of fulfilling this user role’s needs. This user doesn’t need detailed accounts software’s functionality nor background context on what the readings mean. This user just needs clearly defined and laid out instructions, potentially cross-referenced to help topics in the event of error.

Since this sort of documentation would solely be for internal use, a less formal tone could be used for this design. Of the three user roles, this group would require the shortest documentation; again, a list of consist task-based checklists would work perfectly well.

An installation/set-up guide is unnecessary as the program will already be installed at their environment, a user guide or program orientation also be unnecessary as the Unit Tester should be very familiar with the program. An FAQ could be useful indicating common malfunctions and their associated symptoms for writing error reports.

## Software Engineer

The software engineer has the most technical requirements involved in their job tasks. Their tasks require a high-level understanding of the coding language and the principles behind the electronics that the software is controlling. Unlike the Unit Tester and End User, the engineer has no linear procedure to follow in writing firmware.

The type of documentation best suited to this user’s needs would be a categorical and complete list of all of PowerWizard’s coding outputs with one or two lines of plain text explaining the code. The information architecture of this document would be extremely important; the engineer valuing how complete, clear and searchable the documentation is.

The documentation would also be extremely “lean” in the sense that there would be almost no concept or procedural writing. These topics are assumed to already be a part of the user’s knowledge base if they are attempting to code firmware. The remaining reference topic writing is concise with no shortage of jargon. This would be again an internal document so a less formal tone could be used, though it may seem jarring considering the subject matter.

## End User

The end user has the most varied of technical requirements because they have the most varied range of job tasks.

The document tone would be more formal and authoritative than the other two user roles, since its purpose would be for guiding the end user along than reference document (at least initially). To this end, this user could make good use of an Installation guide and a Getting Started section, where the broad strokes of the GUI are explained in a visual glossary.

The length of this documentation would also be much longer than the other two due to the broadness of their job tasks and necessary conceptual “padding” for context and meaning. Based on the frenetic work environment expected for this user, the type of writing should seek not to overwhelm the end user. Chunking, white space, rule of 7, and simple language with minimal jargon will all be used to this end.

As for structuring, rather than going through each program aspect and explaining its function the writing would take place in task-based modules. The reason for using this structure rather than a line by line explanation of the GUI is because the same interface aspects are used across multiple different tasks. Which given their number compounds the difficulty for the End User to follow along. Information architecture and navigation are very important here to manage the bulk amount of task topics, ideally this will be accomplished before any writing.