Documenting Software

George Lesica - Wheeler Lab

Software is meant to be used by people to do things.

You keep thinking I’m kidding, but I’m not!

Documentation is what allows people to use software.

Goals

Goals of Documentation

Or, what do we need to communicate?

1. What the software does
2. How to use the software
3. How the software works
4. How the software is designed

These are all very different from one another, I promise!

What the Software Does

What is it used for?

What problem does it solve?

What are the inputs and outputs?

Who might want to use the software?

How to Use the Software

How is it installed?

What platforms are supported?

What should the input look like?

What will the output look like?

What options are available?

How are options specified?

How the Software Works

What are the interesting algorithms and data structures?

What is novel about the software compared to its competitors?

How can they reach you about your Nobel Prize or Turing Award?

How the Software is Designed

What programming language is it written in?

How can the software be extended?

Where in the codebase is the primary business logic?

Audiences

Primary Audiences

Or, the N kinds of people who will read your documentation.

Most of them can be described thusly, where the blanks are a row and column (respectively) from the table…

Someone who _____ _____ your software.

	Use	Modify
Could		
Wants to		
Needs to		

WANT vs NEED

Why are those different categories?

- People who WANT to use your software are willing to jump through hoops and dive deep. They might even read your paper!
- People who NEED to use your software want to spend as little time as possible learning how, it's just* a brick in whatever edifice they're constructing.

* It might be a really important brick, but it's still a brick.

Other Audiences

These are generally of mixed importance for us (as an academic lab), but they ought to be considered…

- Someone who wrote, is writing, or might write competing software
- A journalist or other interested non-user
- An industrial developer or practitioner
- A student learning about a field your software touches
- The “boss” of any of the people previously described

Media

Documentation Media

We have so many choices, but they aren’t all created equal (for all audiences)!

1. README
2. Peer-reviewed paper / poster
3. API documentation
4. Tutorial
5. Project website

Note: These are not entirely disjoint!

README

- Absolutely required
- What the software does… always
- How to use the software… when you don’t have a website
- How the software works… when you don’t have a paper
- How the software is designed… always

Note: You should link to resources that are elided for whatever reason.

Peer-reviewed Paper / Poster

- Optional
- What the software does… always
- How to use the software… ask Travis
- How the software works… always
- How the software is designed… ask Travis

API Documentation

- Optional - meaningful only for libraries
- What the software does… no
- How to use the software… required (how to use the library)
- How the software works… limited (things developers must know)
- How the software is designed… yes (by definition)

Tutorial

- Required
- What the software does… no
- How to use the software… yes (by definition)
- How the software works… no
- How the software is designed… limited (only for libraries)

Project Website

- Optional
- What the software does… always
- How to use the software… always
- How the software works… if not included elsewhere
- How the software is designed… if not included elsewhere

Wrapping Up

If your software is important, then your documentation is important.

Cheat Sheet

1. Have you accomplished all of the goals?
2. Have you targeted all the primary audiences?
3. Have you provided an appropriate combination of media?