A Friendly Introduction to Software Documentation

Overall Documentation and READMEs
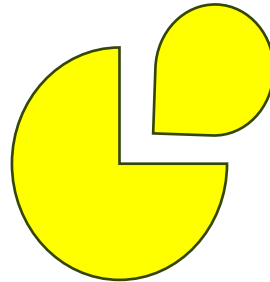
By Arist Bravo

ACCEU and Overall Documentation

- So far, we looked at applying ACCEU to one file's documentation

- What about documenting multiple files or a project overall?

- Dimensions still hold

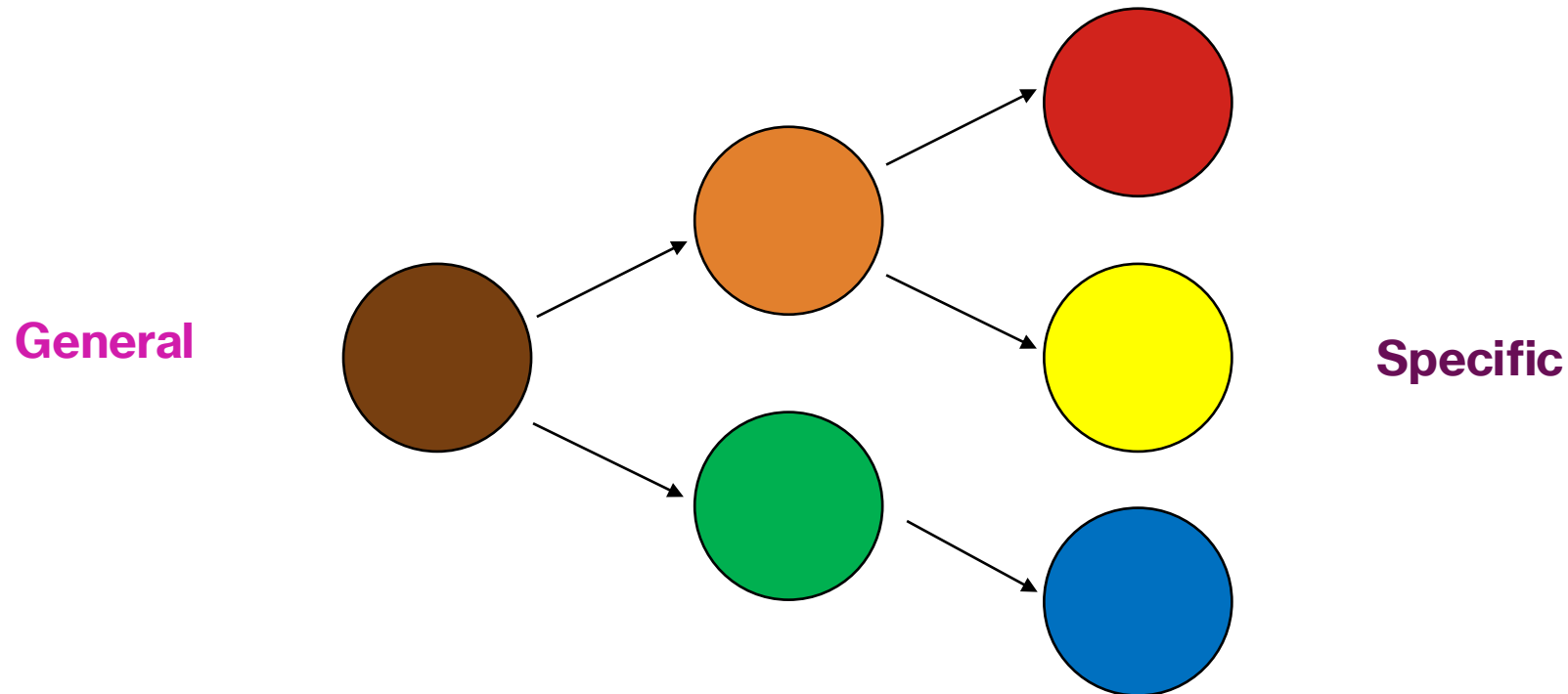**Accuracy**   **Clarity**   **Completeness**   **Ease-of-use**   **Up-to-dateness**

Reference(s): Forward and Lethbridge, Garousi et al., Parnas, Treude et al., Zhi et al.
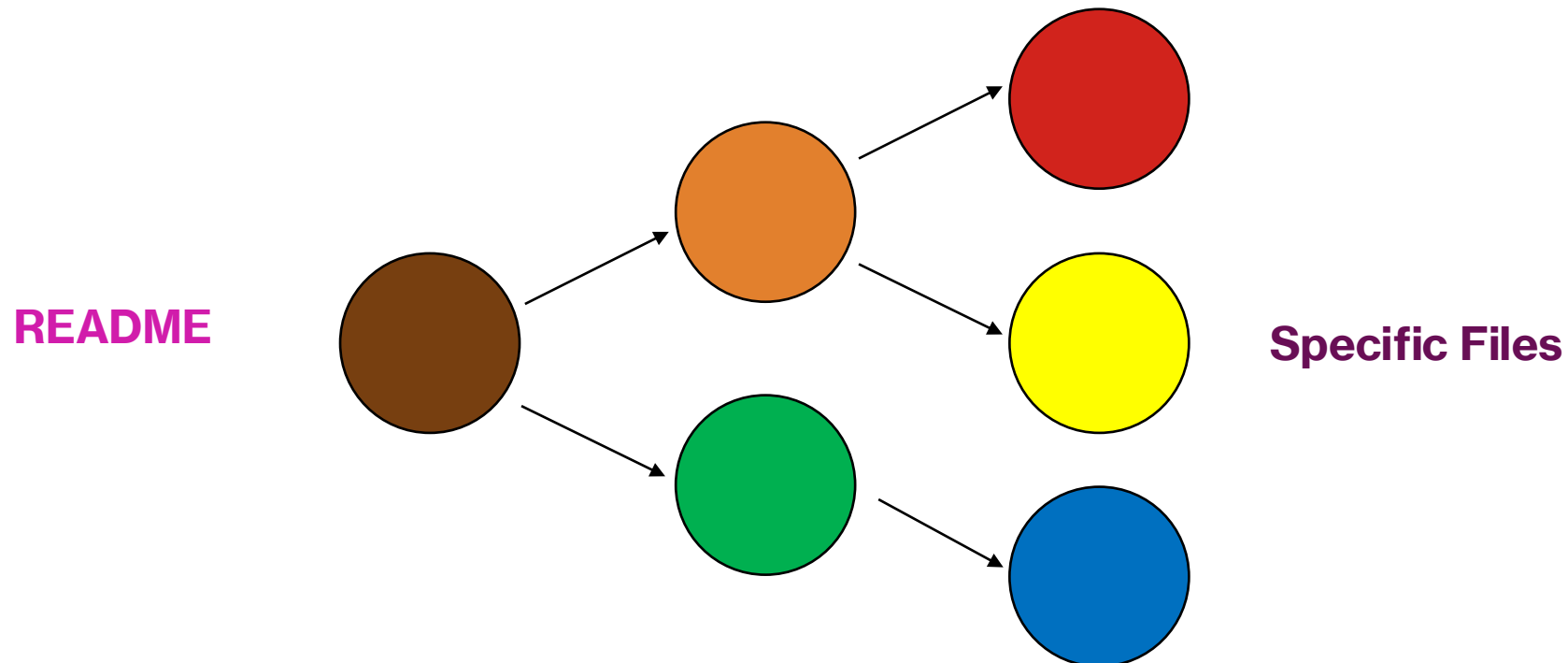
ACCEU and Higher-Level Views

- Key: high-level documentation links to/references other files (i.e. "graph of files")

- High-level docs must be complete (covering all related items), easy to use (in an obvious place), and consistent (with other files' documentation)

General

Specific

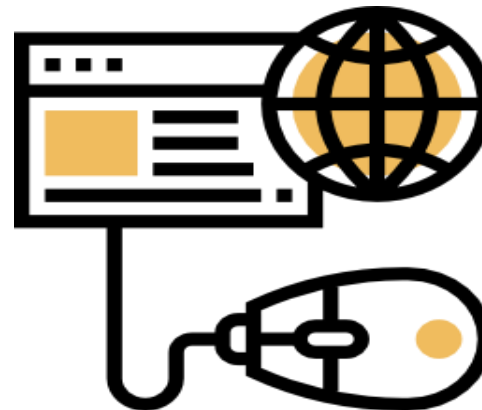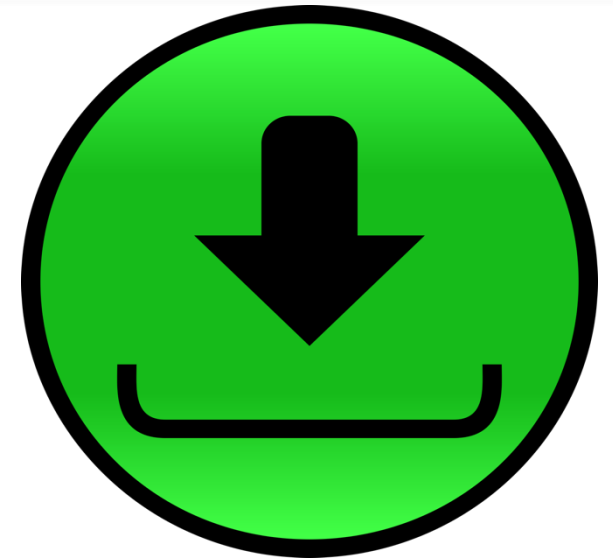*Reference(s): Treude et al., Wingkvist et al.*

Role of READMEs

- READMEs: possibly the highest-level technical doc on a repo (especially on GitHub)

- Purpose: introduce the repo/project to an unfamiliar programmer

- Usually written in Markdown (.md format) on GitHub

README

Specific Files

Possible Sections of a GitHub README

- Title, authors, and short description of the repo or project's purpose

- Table of contents

- Features (what the code can do)

- Installation instructions

- Usage (how to use the project, with examples)

- License (how others can use your code)

- Feedback and Contributions (how others can give feedback or help)

Reference(s): Drupal, GitHub, Guo, Nyakundi, Otieno, Patel, Sullivan
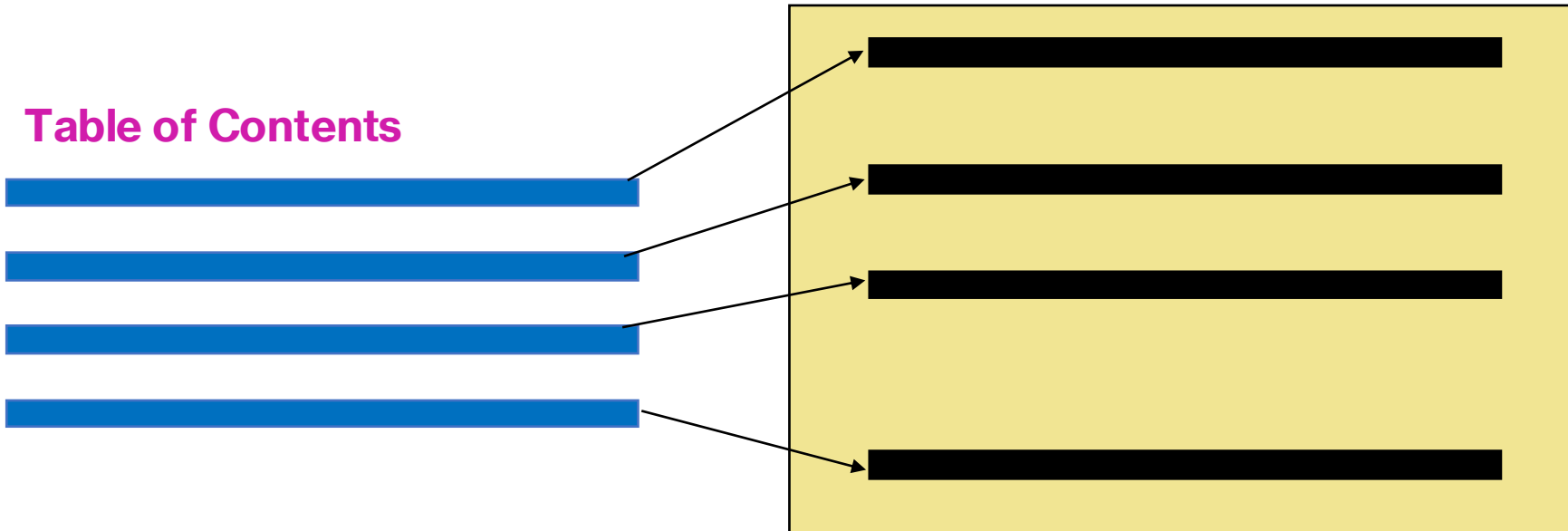
Title, Authors, Short Description

- Project name and the creators (you and your team members – give credit!)

- Briefly answer:
  - What does this project do?
  - Why was this project made?

Reference(s): Sullivan

Table of Contents

Headers

- Increases ease-of-use by allowing for simple navigation

- Navigates to large headings (easily noticeable for *ease-of-use*)

- Think about the major sections (that we discussed) as possible headings
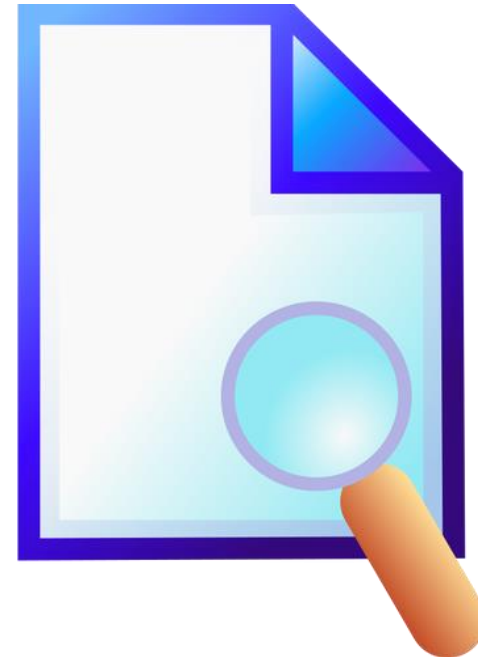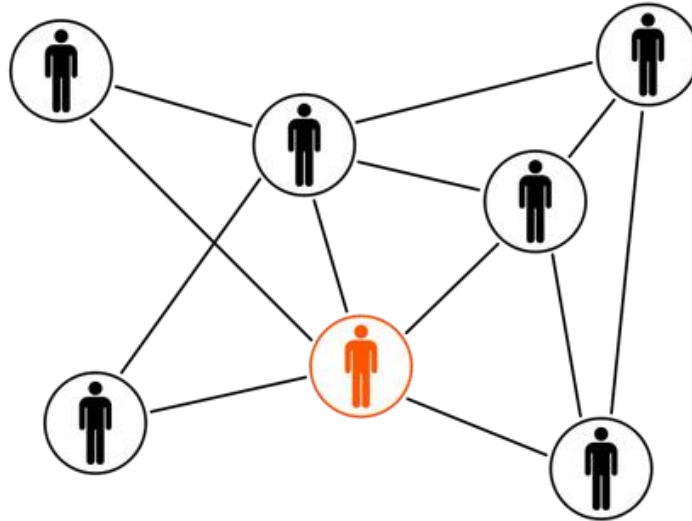
- GitHub may auto-generate this, but check to be sure

```
# H1
## H2
### H3
#### H4
##### H5
###### H6
```

Table of Contents

Reference(s): GitHub, Pritchard

Features

- Highlight major features of your software

- By saying what the software specifically does, you help people figure out whether this software is for them (*ease-of-use*)!

Reference(s): Sullivan

Installation Instructions

- Very important: *clearly* and *accurately* describe how your software is to be downloaded

- Be *complete*: include all info needed to download *every* part, and mention any common issues (and how to overcome them in downloading!)
  - What extra software or packages must be downloaded? What version?
  - Note if the software's only being tested for a specific OS or hardware (EX: Mac, Windows)

- For *ease-of-use,* link to any needed software

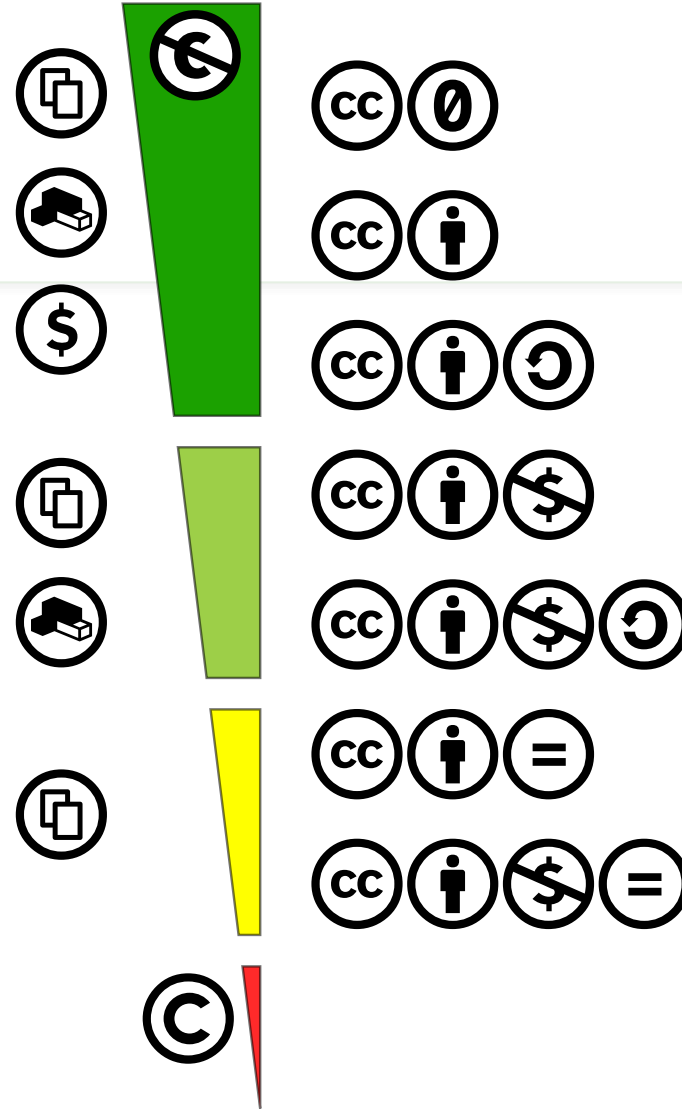- Gives examples or tutorials if needed

Usage

- Give a *clear, accurate* guide on how to use the software, once downloaded

- Use code examples (EX: how to start up or use the code), screenshots, .gif files, or videos to make the README file *clear* and *easy-to-use*
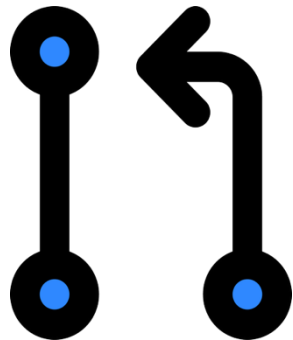

Reference(s): Sullivan

License

- How can other people use your code? It's being shared on GitHub, after all...

- What if you put this project on your resume/portfolio? You *definitely* want a license

- Read about different ways to license: MIT license, public domain, etc.

Feedback and Contributions

- How can people give you feedback on the software?
    - Google Forms
    - Discussion boards
    - Etc.

- How can people contribute?
    - Instructions for making a fork
    - Guidelines for what a good merge request looks like
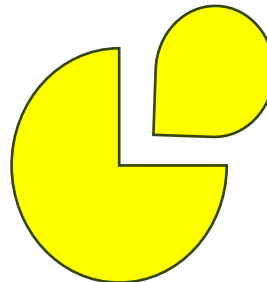   - Protocols you follow for reviewing contributions

Reference(s): Sullivan

[Feature 6 Issue 2] Fixing User History `bug` `enhancement`
#51 by llukeham was closed on Dec 8, 2022

[Issue 4a-03] Adapting to others' CLEAN `documentation` `enhancement`
#47 by GlassInterface was closed on Dec 8, 2022

Fix Independency problem in LikeBlockController & UI `bug` `enhancement`
#45 by Adelineleonardi was closed on Dec 2, 2022

[Issue 6a] Fix User History `documentation` `enhancement`
#40 by llukeham was closed on Dec 2, 2022

[Issue 4a-01] Refinement and Code Cleaning `documentation` `enhancement`
#39 by GlassInterface was closed on Dec 8, 2022

[Issue main-02] Add Design Patterns `documentation` `enhancement` `help wanted`
#36 by GlassInterface was closed on Dec 8, 2022

[Issue main-01] Compilation Errors `bug`
#35 by GlassInterface was closed on Dec 8, 2022

[Issue 4c-01] Improving control flow of 4c `help wanted` `question`
#34 by GlassInterface was closed on Dec 8, 2022

ACCEU as Rules-of-Thumb

- ACCEU helps you write READMEs more generally

- Is all the info *accurate*?

- Are your sentences *unambiguous/clear*?

- Is the README *complete*, referencing every part of the project or links to docs which explain the different parts?

- Is the README *easy to use* and navigate, with clear headings and images?

- Is the README kept *up-to-date* each time the repo is updated?

ACCEU and Higher-Level Views

- Key: high-level documentation links to/references other files (i.e. "graph of files")

- High-level docs must be complete (covering all related items), easy to use (in an obvious place), and consistent (with other files' documentation)

General

Specific

Reference(s): Treude et al., Wingkvist et al.