

		Ordinality of Documentation By Tier (M - Mandatory, R - Recommended, N - Not Recommended)					
File	Sections	Tier 0: Private Repo	Tier 1: One-time Release	Tier 2: Close Collaboration	Tier 3: Working in Public	Tier 4: Community Governance	Notes
LICENSE.md	License	M	M	M	M	M	
SECURITY.md	Security and Responsible Disclosure Policy	N	M	M	M	M	This outlines the Security & Responsibility Disclosure policies including vulnerability submission, etc.
README.md	Project Description	M	M	M	M	M	This should be 1-3 sentence short description of the project that can be used as a 'one-liner' to describe the repo. A best practice is using this same language as the official 'description' on a GitHub repo landing page.
	About the Project	M	M	M	M	M	This should be a longer-form description of the project. It can include history, background, details, problem statements, links to design documents or other supporting materials, or any other information/context that a user or contributor might be interested in.
	Core Team	R	M	M	M	M	This information helps with succession planning and provenance for security compliance and remediation. It helps future users and contributors understand where the code originated.
	Documentation Index	R	R	R	M	M	This is a like a 'table of contents' for your documentation. Tier 0/1 projects with simple README.md files without many sections may or may not need this, but it is still extremely helpful to provide "bookmark" or "anchor" links to specific sections of your file to be referenced in tickets, docs, or other communication channels.
	Repository Structure	R	R	R	M	M	Using the "tree" command can be a helpful way to generate this information, but, be sure to update it as the project evolves and changes over time.
	Development and Software Delivery Lifecycle	N	R	R	M	M	Tier 1: Even if the lifecycle is "one-time release" being explicit is better than implicit
	Local Development	R	R	M	M	M	Use step by step instructions to get from 'zero' to 'running code.' Should include any system libraries or packages that are a 'pre-requisite' to installation of your project. When possible, including install instructions for multiple Operation Systems (or being explicit about which operating system the project was developed on) is a recommended practice.
	Coding Style and Linters	R	R	M	M	M	
	Branching Model	R	R	R	M	M	Even for Tier 0/1 projects with one or a few contributors, branching models (such as git flow) are still recommended as a best practice for keeping feature development history clear, and to help reinforce development best practices.
	Contributing	N	R	M	M	M	For projects that accept contributions, point towards the CONTRIBUTING.md file. For those that do not (tier0/1) not recommended to include this section, instead, mention one-time release, or private repo status instead.
	Codeowners	N	R	M	M	M	Though all tiers have an 'implied' code-owner, since there is at least one author of the repo, explicit is better than implicit. In the case that a project may outlive the employment or contract of the original author, a shared inbox or alias is recommended for longer-lived projects (e.g. opensource@cms.hhs.gov)
	Community	R	R	M	M	M	For projects that are higher tier than a one-time release, pointing your contributors towards wherever your community exists (e.g. email lists, online discussion boards or channels, project backlogs and documentation, etc...)
	Community Guidelines	R	R	M	M	M	Project tiers above one-time release should point towards a CODE_OF_CONDUCT.md file or website providing information around acceptable conduct and reporting mechanisms and escalation strategies. It is better to have these processes defined before they are needed, so you can focus on support if/when there is an incident. (e.g. Contributor-covenant.org)
	Governance	N	N	R	R	M	Make a short statement about how the project is governed (formally, or informally) and link to the GOVERNANCE.md file.
	Feedback	N	R	R	M	M	Direct users towards the channel where they're encouraged to provide feedback, typically a github.com/\$REPO/issue/new URL
	Glossary	R	R	R	R	M	Even for early-tier projects, this documentation can be extremely valuable. Good candidate content includes any project-specific acronyms (esp applicable for Government projects) and any critical Subject Matter Expertise related vocabulary for people who are new to the domain your project is within.
	Policies	M	M	M	M	M	This section is to explicitly link to Federal policies and guidelines that are required or recommended for Federal projects to comply with, such as Accessibility (508) Interoperability, Anti-deficiency, Security, Licensing, and other policies that can vary between agencies and domains.
	Public Domain	M	M	M	M	M	A best practice is to list the LICENSE under which a project is released at the bottom of the README. In most cases for Federal repos, we default to Creative Commons Zero 1.0 International (world-wide public domain.)
CONTRIBUTING.md	How to Contribute	R	R	M	M	M	Basic instructions about where to send patches, check out source code, and get development support.
	Getting Started	R	M	M	M	M	Often includes installation steps, pre-requisites for installation, and instructions for working with the source code.
	Team Specific Guidelines	N	N	R	M	M	This section helps contributors understand any team structure in the project (formal or informal.) Encouraged to point towards the MAINTAINERS.md file for further details.
	Building dependencies	R	M	M	M	M	This step is often skipped, so don't forget to include the steps needed to install on your platform. If you project can be multi-platform, this is an excellent place for first time contributors to send patches!
	Building the Project	R	M	M	M	M	Be sure to include build scripts and instructions, not just the source code itself!
	Workflow and Branching	R	R	M	M	M	If your project has a preferred workflow or branching structure, mention it here. We recommend 'git flow' as a good default.
	Testing Conventions	R	R	R	M	M	Discuss where tests can be found, how they are run, and what kind of tests/coverage strategy and goals the project has.
	Coding Style and Linters	R	R	M	M	M	HIGHLY ENCOURAGED. Specific tools will vary between different languages/frameworks (e.g. Black for python, eslint for JavaScript, etc...)
	Writing Issues	R	M	M	M	M	Make a brief statement about where to file issues, and conventions for doing so. Link to ISSUE_TEMPLATE.md file.
	Writing Pull Requests	N	N	R	M	M	Make a brief statement about where to file pull/merge requests, and conventions for doing so. Link to PULL_REQUEST_TEMPLATE.md file.
	Reviewing Pull Requests	N	N	R	M	M	Make a brief statement about how pull-requests are reviewed, and who is doing the reviewing. Linking to MAINTAINERS.md can help.
	Shipping Releases	N	N	R	R	M	What cadence does your project ship new releases? (e.g. one-time, ad-hoc, periodically, upon merge of new patches) Who does so?
	Documentation Updates	R	R	R	M	M	Where is the documentation hosted? How is it updated? Who updates it?
	Policies	R	M	M	M	M	This section is here to explicitly link to Federal policies and guidelines that are required or recommended for Federal projects to comply with, such as Accessibility (508) Interoperability, Anti-deficiency, Security, Licensing, and other policies that can vary between agencies and domains.
	Public Domain	R	M	M	M	M	This section is to explicitly link to Federal policies and guidelines that are required or recommended for Federal projects to comply with, such as Accessibility (508) Interoperability, Anti-deficiency, Security, Licensing, and other policies that can vary between agencies and domains.
	Maintainers	N	N	R	M	M	Who are the project maintainers? List out @USERNAMES where possible so they can be tagged in issues/PRs directly.

		Ordinality of Documentation By Tier (M - Mandatory, R - Recommended, N - Not Recommended)					
File	Sections	Tier 0: Private Repo	Tier 1: One-time Release	Tier 2: Close Collaboration	Tier 3: Working in Public	Tier 4: Community Governance	Notes
MAINTAINERS.md	Maintainers Table	N	N	R	M	M	What groups/domains are maintainers a part of? Does your project have domains/areas that are maintained by specific people? List @USERNAMES directly, or any @ALIASES for groups/teams.
GOVERNANCE.md	Governance	N	N	N	R	M	Starting at Tier 3 GOVERNANCE.md has basic language about early community governance, how the project make decisions, and how contributors are elevated through the leadership process if any (e.g. joining teams, getting maintainer status, etc...)
CODEOWNERS.md	list of Users	N	N	R	M	M	Who are the points of contact in your project who are responsible/accountable for the project? This can often be an engineering or design manager or leader, who may or may not be the primary maintainers of the project.
	list of repo domains by owner	N	R	R	M	M	e.g. Frontend, Backend, Documentation