“Working software over comprehensive documentation” does not mean “don’t write documentation”.

The code doesn’t tell the whole story. Supplementary documentation can be used to describe what you can’t get from the code. Make the scope of the documentation a single software system. Here is a starting point, and there are others including arc42.

Think about supplementary documentation as being a guidebook containing maps, points of interest, sights, itineraries, history, culture, practical information, etc. It should be lightweight, readable in 1-2 hours and give software developers enough information to get started, accelerating the process of exploring an unfamiliar codebase.

Documentation isn’t a one-time task. Instead, create living documentation that evolves continuously. Keep it up to date automatically with tooling, or by adding an item to your “definition of done”.

There are many tooling options; from Microsoft Word and Atlassian Confluence to Markdown and AsciiDoc files versioned alongside the source code. Reduce duplication and increase consistency by generating diagrams and documentation from a single source where possible.