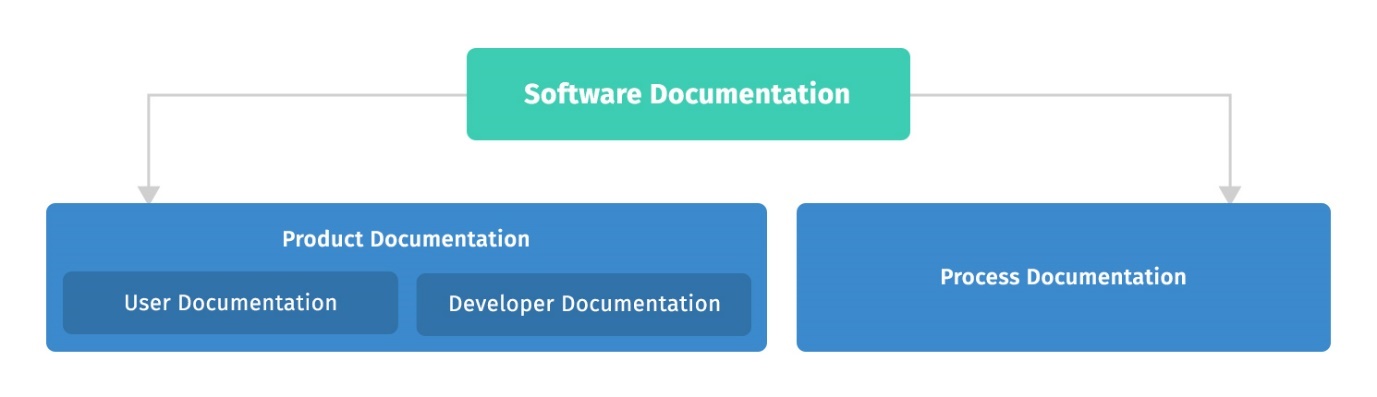
[Developer Documentation](https://www.archbee.com/blog/developer-documentation-guide)

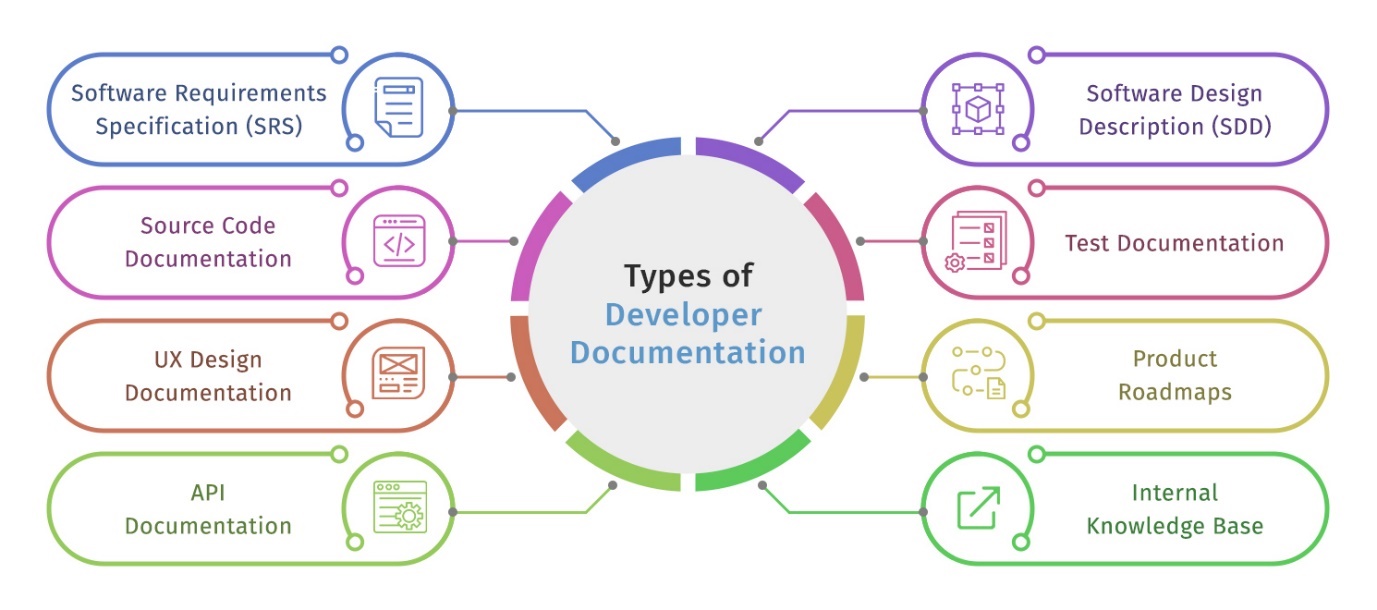
[Software Documentation](https://technicalwriterhq.com/documentation/software-documentation/" \l ":~:text=Developer%20documentation%3A%20used%20to%20document,during%20the%20software%20development%20process.)

Types of Software Documentation



1. Developer documentation: used to document software requirements, design, architecture, and source code. It is created by dedicated technical writers or software developers during the software development process. Developer documentation is used by software developers, programmers, project managers, and other stakeholders involved in the software engineering process. It serves as a reference for developers who may later work on updates to the software. Developer documentation is also known as system documentation.
2. User documentation: provides information about installing, configuring, or using software. Software is a product, and software documentation is part of the product. Comprehensive software documentation is one of the key factors that influence businesses’ buying decisions.

### **Types of Developer Documentation**



#### **Software Requirements Specification (SRS)**

The software requirement specification describes a software system to be developed. It lays out functional and non-functional requirements. The software requirement specification may include a set of use cases that describe user interactions that the software must provide.

The software requirement specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function.

#### **Software Design Description (SDD)**

The software requirements specification (SRS) serves as the basis for the software design description or SDD. The SDD contains the software design and overall architecture.

The SDD helps to ensure that the whole project team, including the software developers, are on the same page. The SDD also helps to ensure that all stakeholders vet the entire design and that all risks and assumptions are considered.

#### **Source Code Documentation**

Source code refers to the computer programs that programmers create. It is comprised of long sequences of programming language statements that make up a computer program.

Source code documentation contains all the computer programs related to a software product. It serves as a reference for developers who may work on later versions of the software, and for developers who may use components of the software for their own projects.

#### **Software Test Documentation**

Software development is an iterative process: software is developed and then tested, errors – known as bugs – are identified and removed, and then the software is tested again. Once a working software version is ready, work starts on the next software version.

Software test documentation contains detailed test plans and procedures for software testing. Components of the software test documentation include

* Master Test Plan (MTP): contains the overall test plan.
* Level Test Plan (LTP): contains the approach, resources, and schedule of the testing activities for each LTP.
* Level Test Design (LTD): contains details for the test cases and test pass criteria.
* Level Test Procedure (LTP): contains the detailed test procedure, including details for necessary pre-requisites.
* Level Test Report (LTR): contains a summary of the test for a specified test level.
* Master test report: contains a summary of the overall test report.

#### **UX Design Documentation**

UX is the acronym for “user experience.” When we say “user experience,” we refer to how people interact with a product. In the digital design world, UX refers to everything that affects a user’s interaction with a digital product. User experience is about what users both think and feel, and it also depends on the context in which the product is used.

UX design is the process of creating products that are practical and usable. UX requires a deep understanding of the user: their needs, wants, behaviors, and the context in which they will use a product. The ultimate goal of UX design is to make usable and useful products for users and businesses.

UX design is part of the product design, and that is why it begins at the requirements stage and proceeds through all the stages of software development, including the testing and post-release stages. UX documentation covers user personas, user scenarios, user story maps, and a UX style guide.

#### **Product Roadmap**

The product roadmap contains a plan of action for how a software product will evolve over time. It serves as a guide for both business and technical teams.

#### **API Documentation**

API is the acronym for Application Programming Interface. An API is a software intermediary that allows two applications to interface with each other.

API documentation contains instructions about effectively using and integrating with an API.

#### **SDK Documentation**

SDK is the acronym for Software Development Kit. An SDK is a set of software-building tools for a specific platform, including the building blocks, debuggers, and a group of code libraries such as a set of routines specific to an operating system (OS).

SDK documentation contains instructions about how to use an SDK effectively.

#### **UML Diagrams**

UML stands for Unified Modeling Language. It is intended to provide a standard way to visualize the design of a software system.

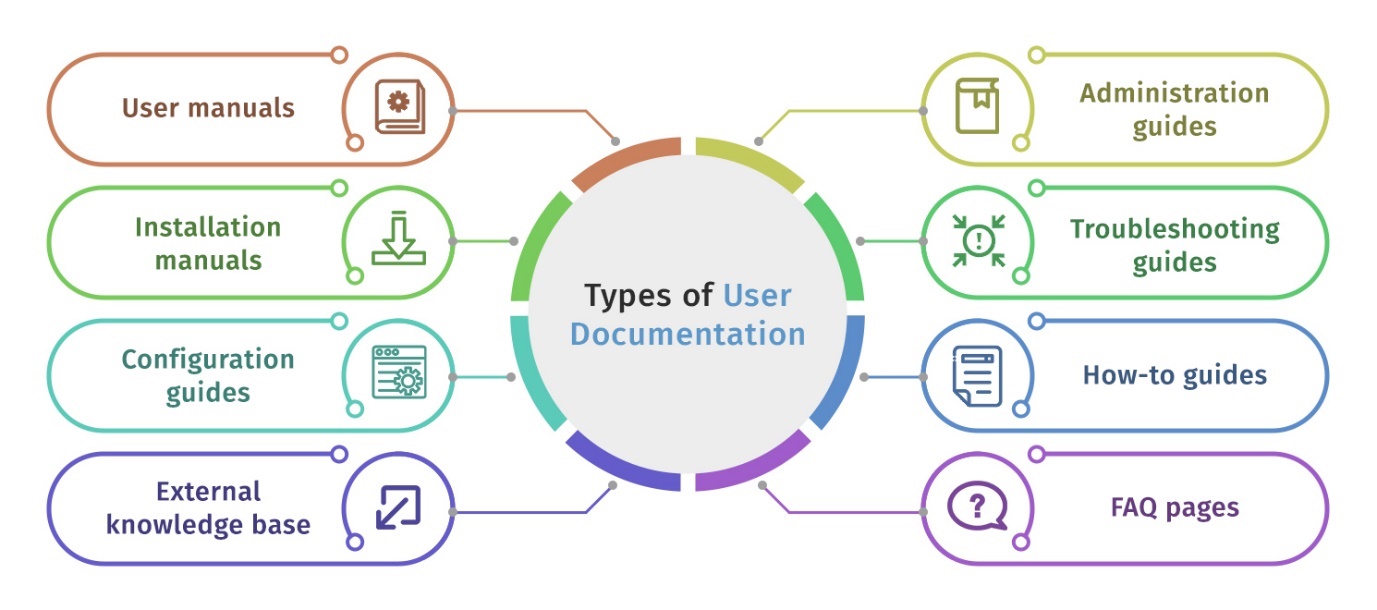
Creating a UML diagram before any code is written is an efficient way for programmers to keep track of all the components involved and how they relate to each other.

The current UML standard specifies 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing, and deployment. These 13 types of diagrams are organized into two groups: structural diagrams and behavioral or interaction diagrams.

#### **Internal Knowledge Base**

An internal knowledge base is typically utilized as a way to allow employees to collaborate and share all company knowledge and information internally. When creating an internal knowledge base, you can include anything that is meant for internal use.

### **Types of User Documentation**



Software users require several types of end-user documentation to install, configure and use software products.

#### **How-to Guide**

A how-to guide contains step-by-step instructions to help users perform specific tasks such as installing the software, upgrading the software, and activating the software.

How-to guides are especially relevant to software users with limited exposure to technology or users who are using software for the first time.

#### **Tutorials**

A tutorial is a learning aid designed to share knowledge and skills related to a certain topic. Examples include tutorials related to using a certain module of enterprise software or a tutorial on how to prevent software from unauthorized access.

Some tutorials include test questions to ensure comprehension of the material, while others may be simple walkthroughs of a software program. Tutorials are created for different levels of users such as basic, intermediate, and advanced.

#### **Configuration Guide**

Most enterprise-level software is complex, with hundreds or thousands of settings that require configuration. The setting for each parameter varies depending on the customer’s requirements. A configuration guide contains all the necessary details that allow system administrators to configure the software successfully.

#### **Administration Guide**

Usually, a team of IT professionals working under an administrator manages day-to-day operational issues such as adding new users, providing access rights, and taking data backups.

An administration guide contains all the relevant instructions administrators and their teams require for configuring and maintaining the software.

#### **Troubleshooting Guide**

A troubleshooting guide contains a list of common problems along with step-by-step solutions.

#### **External Knowledge Base**

A knowledge base is a library of information about your software. Its purpose is to make it easy for people to find solutions to their problems without having to ask for help. Knowledge bases use a combination of text, image, and video-based content.

An external knowledge base – also known as a customer-facing knowledge base – is where customers can go to learn anything they’d ever need to know about a company’s software-related products and services. It is usually public to everyone and can be easily found online. If you browse through any software’s Help and Documentation section, that’s their external knowledge base.

#### **FAQ Pages**

FAQs are answers to questions that have been either asked on a regular basis or that you expect your users to ask at some point. FAQs explain topics that don’t require too much depth or technical support. They cover topics that can be explained in one or two paragraphs.