Topic: System Implementation

Overview of System Implementation

A. Software Development

B. Software Installation

C. Software Support & Maintenance

(A**) Software Development**

• Acquiring software & hardware
– Alternatives: Suitability & Cost

• Software testing
– Types of testing

Acquiring software & hardware

• In house development / Outsourcing / Off-the-shelf purchase

– The objective is to translate designs into code modules hence a system that function properly
– Usually, involve team of developers, database and servers in house or by a third party

• Custom designed (End user programs, Open sources)
– EUP: Less formal,solving immediate needs, by power users

Topic: System Implementation

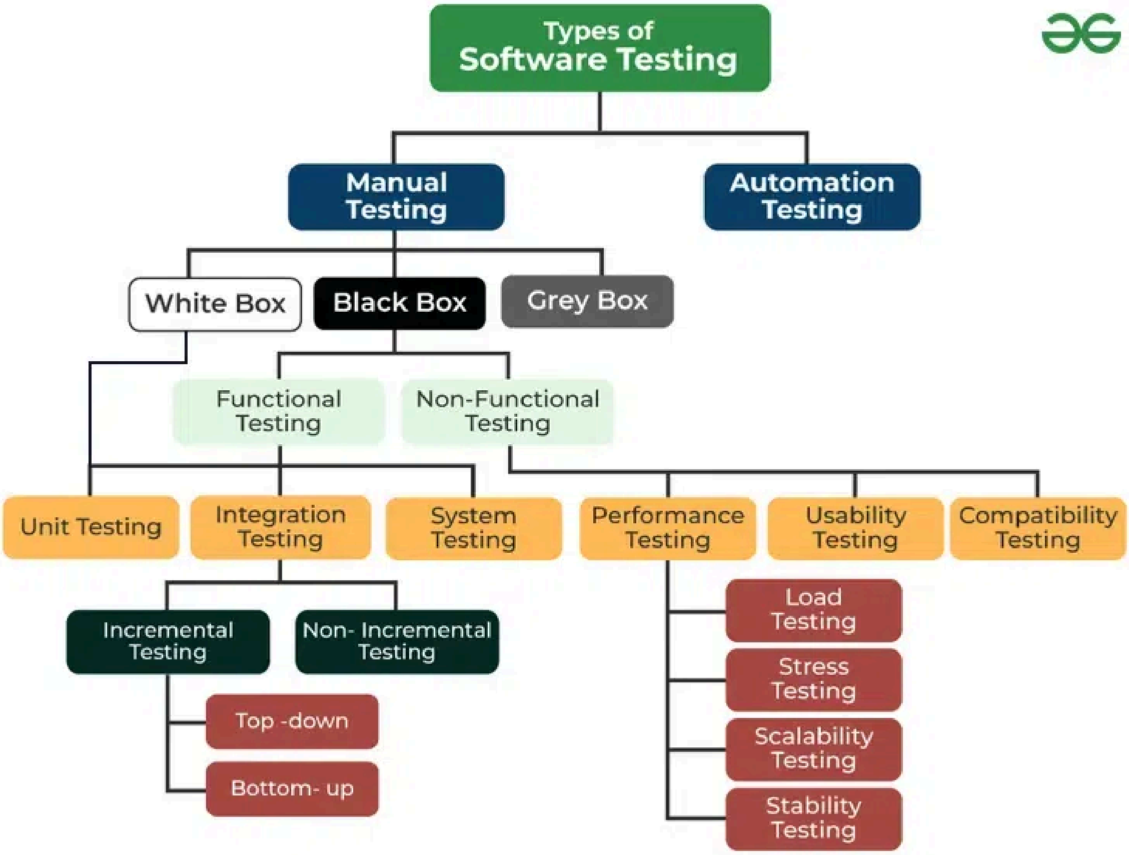
Software testing

Software Testing can be broadly classified into 3 types:

Functional testing : It is a type of software testing that validates the software systems against the functional requirements. It is performed to check whether the application is working as per the software's functional requirements or not. Various types of functional testing are Unit testing, Integration testing, System testing, Smoke testing, and so on.

Non-functional testing : It is a type of software testing that checks the application for non-functional requirements like performance, scalability, portability, stress, etc. Various types of non-functional testing are Performance testing, Stress testing, Usability Testing, and so on.

Maintenance testing : It is the process of changing, modifying, and updating the software to keep up with the customer's needs. It involves regression testing that verifies that recent changes to the code have not adversely affected other previously working parts of the software.



Topic: System Implementation

Manual Testing

Manual testing is a technique to test the software that is carried out using the functions and features of an application. In manual software testing, a tester carries out tests on the software by following a set of predefined test cases

Automation Testing

Automated Testing is a technique where the Tester writes scripts on their own and uses suitable Software or Automation Tool to test the software. It is an Automation Process of a Manual Process. It allows for executing repetitive tasks without the intervention of a Manual Tester

Different Levels of Software Testing

Software level testing can be majorly classified into 4 levels:

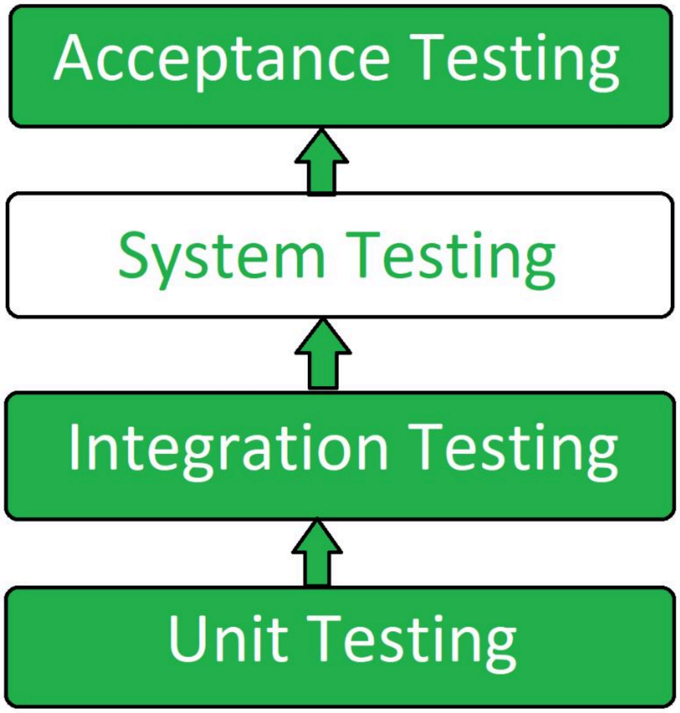
Unit testing : It a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

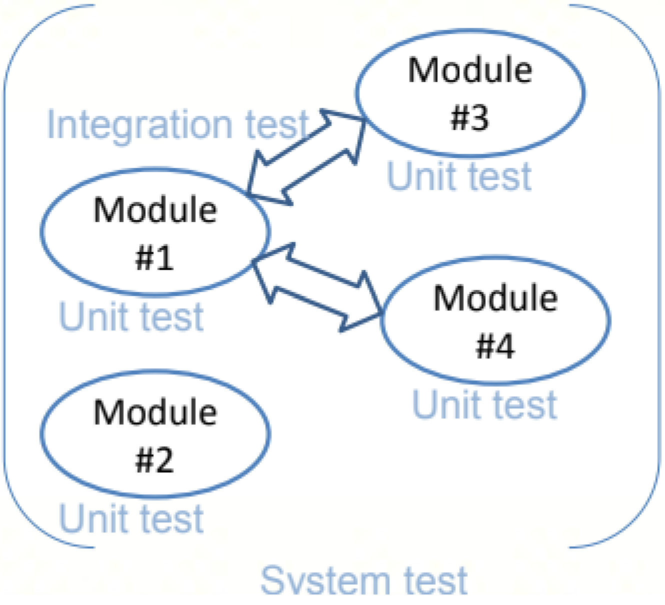
Integration testing : It is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

System testing : It is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

Acceptance testing : It is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Topic: System Implementation





.

Types of System Testing

- □ **Performance Testing**: Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.

- □ **Load Testing**: Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.
- □ **Stress Testing**: Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.
- □ **Scalability Testing:** Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load

Tools used for System Testing

- JMeter
- Gallen Framework
- HP Quality Center/ALM
- IBM Rational Quality Manager
- Microsoft Test Manager
- Selenium
- Appium
- LoadRunner
- Gatling
- JMeter
- Apache JServ
- SoapUI

(B) Software Installation

• Types of conversion
– Activities in data and system conversion

Data Conversion

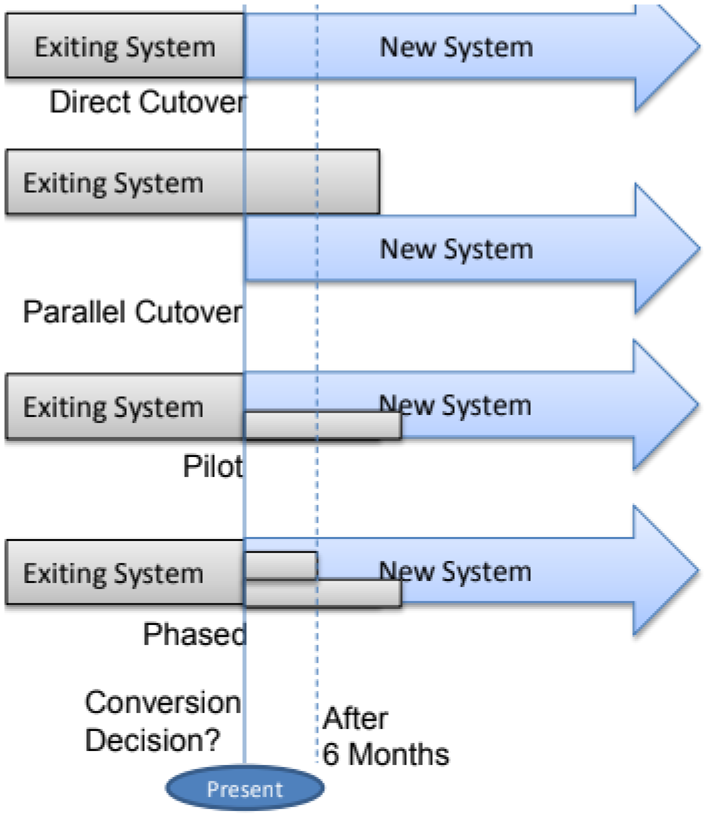
• Acquire required data from the existing
systems
– Use data export tools
– Develop a data extraction program
• Consolidate & clean data from multiple
sources
• Transform data as required by the new system
• Feed the data into the new system

System Conversion

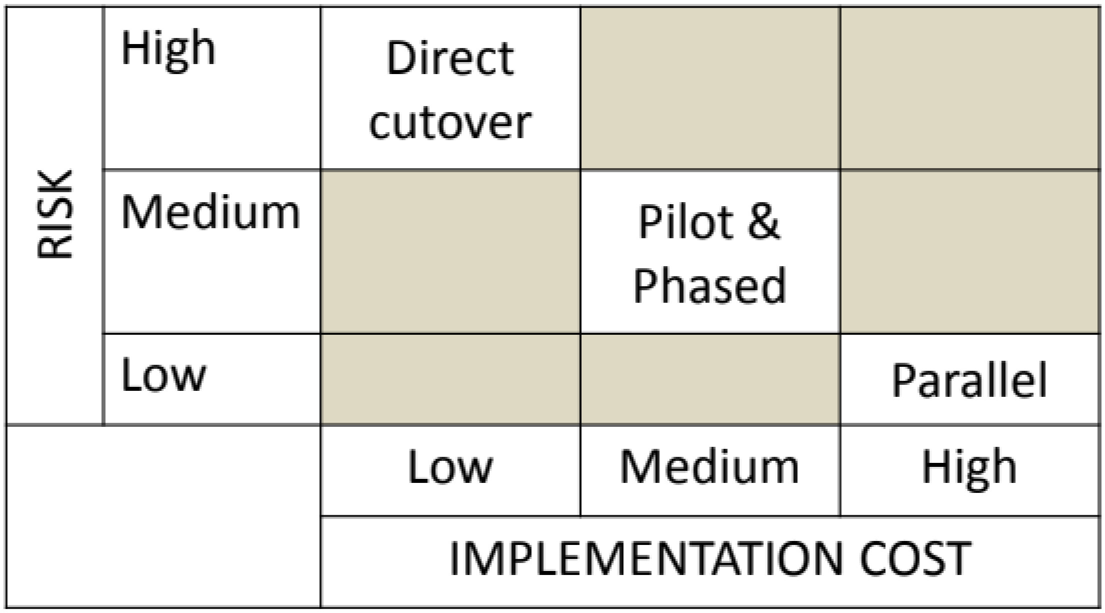
Choose a type of conversion
• Direct cutover

Topic: System Implementation

- Parallel
- Pilot
- Phased



System Conversion



RISK	High	Direct cutover		
	Medium		Pilot & Phased	
	Low			Parallel
		Low	Medium	High
		IMPLEMENTATION COST		

Topic: System Implementation

(C) Software Support & Maintenance

After sign-off by the top management and celebrated over success :

- Training

- System audits & documentation

Training

- New system is installed and people are trained to use it
– End users, managers, in house IT personnel
– Training parties: In-house, vendor, 3rd Party

System audits & documentation

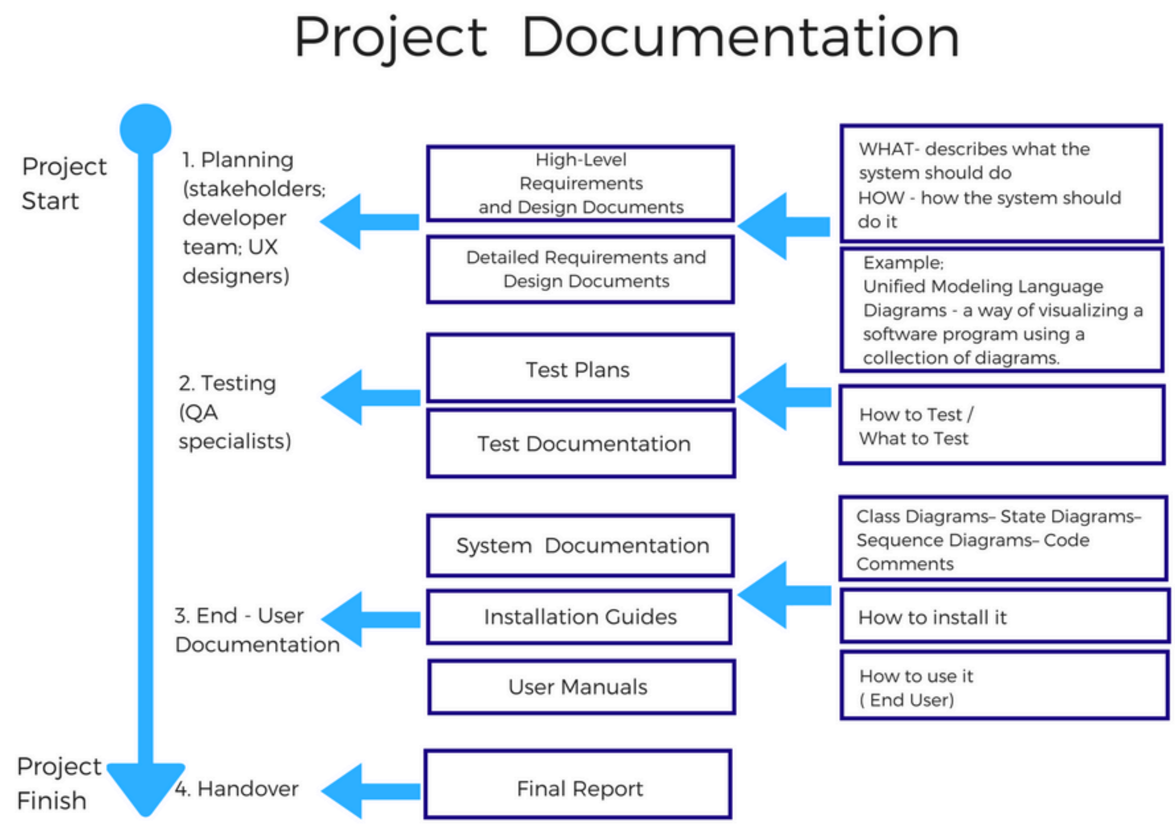
- **Systems audit**
– The system's performance is compared to the original design specifications to ensure the new procedures improve productivity

- **Periodic evaluation**
– The system is evaluated from time to time to ensure they still meet the goals and providing the service as specified
– Users' needs change over time

Documentation

What is technical documentation in software development?
Technical documentation in software engineering is the umbrella term that encompasses all written documents and materials related to software product development. All software solutions, whether created by a small team or a large corporation, require various types of documents throughout the whole software development lifecycle (SDLC).

Topic: System Implementation


Project Documentation
Project Start
1. Planning (stakeholders; developer team; UX designers)
High-Level Requirements and Design Documents
Detailed Requirements and Design Documents
WHAT- describes what the system should do
HOW - how the system should do it
Example: Unified Modeling Language Diagrams - a way of visualizing a software program using a collection of diagrams.
2. Testing (QA specialists)
Test Plans
Test Documentation
How to Test / What to Test
3. End - User Documentation
System Documentation
Installation Guides
User Manuals
Class Diagrams- State Diagrams- Sequence Diagrams- Code Comments
How to install it
How to use it (End User)
Project Finish
4. Handover
Final Report

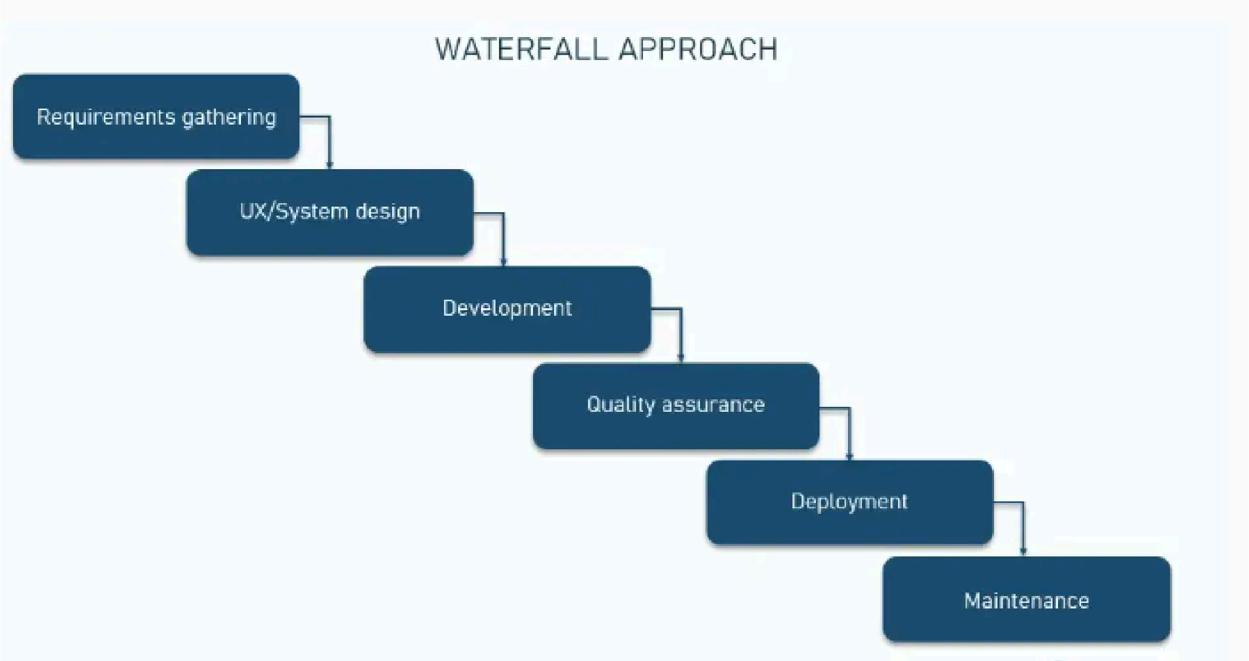
Agile and Waterfall approaches to software documentation

The documentation types that the team produces and its scope depend on the software development approach that was chosen. There are two main ones: Agile and Waterfall. Each is unique in terms of accompanying technical documentation.

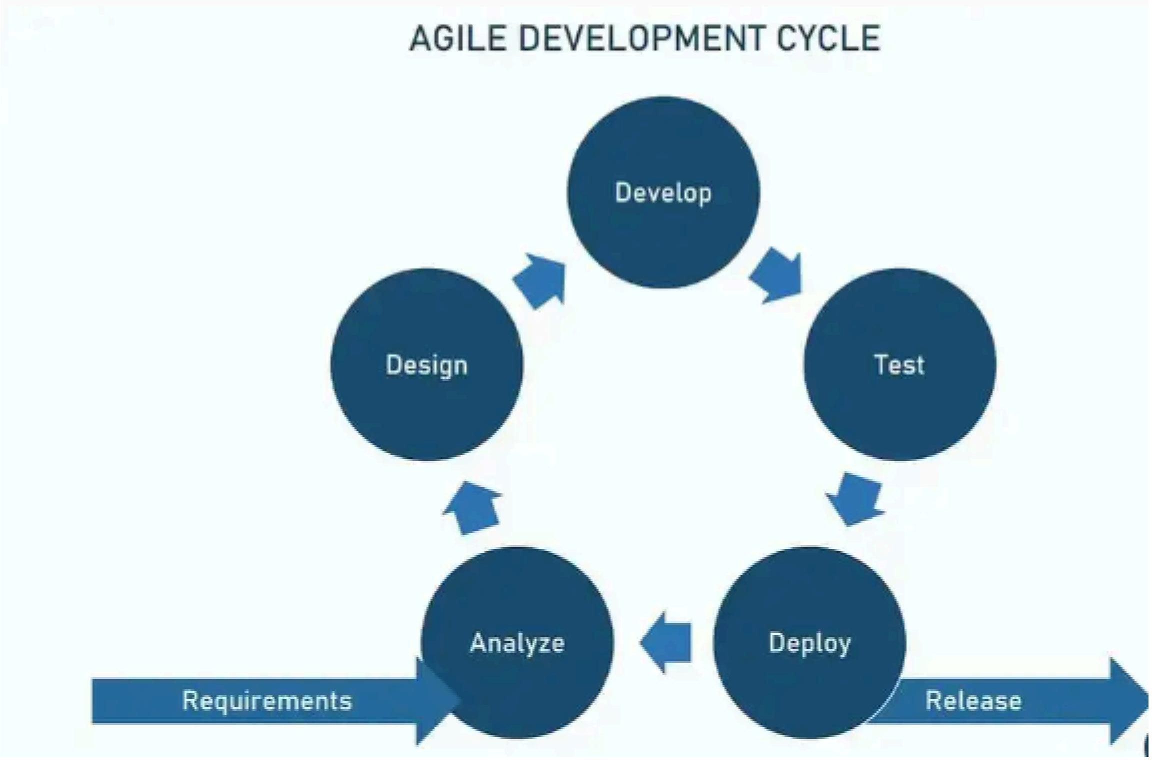
Waterfall approach

The Waterfall approach is a linear method in which you can only proceed to the next stage when you finish the previous one. Teams that use waterfall spend a reasonable amount of time on product planning in the early stages of the project. They create an extensive overview of the main goals and objectives and plan in detail what the working process will look like.

Topic: System Implementation



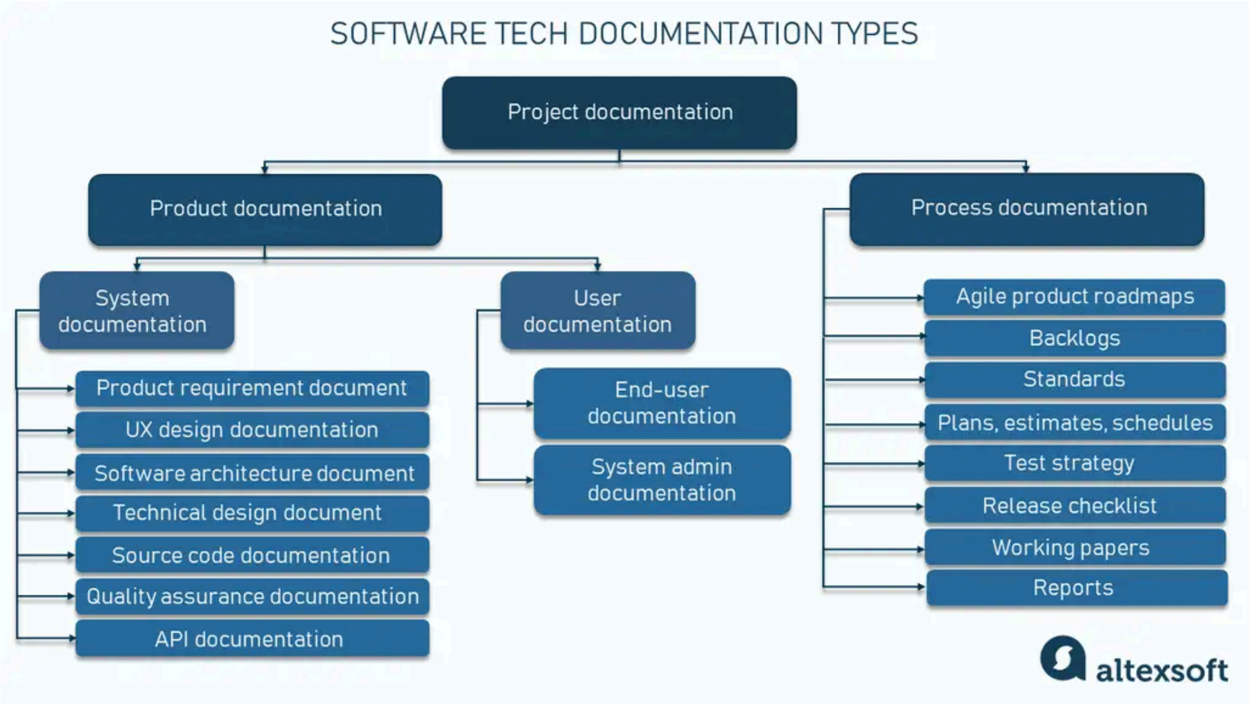
Agile approach
The Agile approach is based on teamwork, close collaboration between developers, business stakeholders, and end customers, flexibility, and the ability to quickly respond to changes. The basic building blocks of Agile development are iterations: Each of them includes such stages as planning, design, development, etc.



Topic: System Implementation

Types of technical documentation

The main goal of effective documentation is to ensure that developers and other stakeholders are working together to accomplish the project's objectives. Different software documentation types exist to achieve this goal.



All software documentation can be divided into two main categories:

Product documentation
Process documentation

Product documentation describes the solution that is being developed and provides instructions on how to interact with it. In general, product documentation includes requirements, tech specifications, business logic, and manuals. There are two main types of product documentation:

System documentation describes the system itself and its parts. It includes requirements documents, design decisions, architecture descriptions, program source code, etc.
User documentation covers manuals mainly prepared for product end-users and system administrators. It includes tutorials, user guides, troubleshooting manuals, installation instructions, and so on.
Process documentation describes how teams work when developing software. It's like a detailed guidebook that lists the procedures team members follow during the SDLC, the tools they use, and the rules they must adhere to. This documentation helps everyone understand how things are done, ensures that the process is consistent, and makes it easier for new team members to get up to speed.

Topic: System Implementation

Common examples of process-related documents are backlogs, roadmaps, coding and testing standards, reports, meeting notes, release checklists, and even business correspondence.

Product: System documentation

System documentation provides an overview of the solution and helps engineers and stakeholders understand the underlying technology. It usually covers

product requirements,
UX design,
architecture,
source code,
QA activities, etc

Product requirements document (PRD)

A product requirements document or PRD provides information about system functionality and behavior. Generally, requirements are statements of what a system should do and how it should work.

User experience design documentation

User experience design (UX design) begins at the planning stage and proceeds through all the stages of development, including the testing and post-release stages. The process of UX design includes research, prototyping, usability testing, and the actual designing phase, during which lots of documentation and deliverables are produced.

Software architecture document

A software architecture document (SAD) includes the main architectural decisions made by the solution architect.
So the software architecture document gives an overview of the product structure and determines the full scope of work, looping in all the stakeholders and providing the overall guidance to the development team.

Technical design document (TDD)

A technical design document (TDD), also often referred to as technical specification, provides detailed, low-level information on how a software system's requirements are to be implemented. It bridges the gap between system architecture and the actual codebase, detailing the specific configurations, interfaces, and coding standards that developers will follow

Topic: System Implementation

Source code documentation
Source code documentation is a type of technical documentation embedded directly within the solution's source code. It explains what the code does, how it works, and why certain decisions were made. This can include descriptions of algorithms, configurations, and complex logic.

Source code documentation comes in the form of comments that can range from single-line notes explaining a particular operation to bigger, block explanations describing more complex logic or data structures.

Quality assurance documentation
QA activities are an indispensable part of any development project. The most common documents related to quality assurance are

Quality management plan
Test plan
Test case specifications
Test checklists

A quality management plan is an analog of a requirement document dedicated to testing. This document sets the required standard for product quality and describes the methods to achieve it. The plan helps to schedule QA tasks and manage testing activity for product managers, but it is mainly used for large-scale projects.

A test plan is a detailed document that outlines the objectives, resources, scope, and schedule of intended testing activities in a project. It defines the roles and responsibilities of a QA team, specifies what is going to be tested and what is not, describes the types of testing to be performed (like unit, integration, and system testing), and the methodologies and tools to be used.

A test case specification is a set of detailed actions to verify each feature or functionality of a product. Usually, a QA team writes a separate specification document for each product unit. Test case specifications are based on the approach outlined in the test plan. A good practice is to simplify specification descriptions and avoid test case repetitions.

A test checklist is a list of tests that should be run at a particular time. It shows what tests are completed and how many have failed. All points in the test checklists should be defined correctly. Try to group test points in the checklists.

API documentation

Most products include APIs (Application Programming Interfaces) to enable data exchange with other systems. API documentation contains a list of all product APIs and their specs. It describes requests, responses, error messages, and other essential details, and informs developers how to effectively interact with the system APIs.

Product: User documentation

As the name suggests, user documentation is created for product users. However, there are different types of them, i.e., end users and system administrators. So you should structure user documentation according to the different user tasks and their different levels of experience.

End-user documentation

The documentation created for end users should explain in the simplest way possible how the software can help solve their problems. Such user instructions can be provided in the printed form, online, or offline on a device.

System administrator documentation: help and maintenance guides

System administrators' documents are specifically designed for the personnel responsible for installing, configuring, maintaining, and troubleshooting computer systems and networks. They provide detailed guidelines and instructions that ensure the proper functioning and security of IT infrastructure.

Some common system administrator documents are

- □ installation guides,
- □ configuration manuals
- □ maintenance procedures (including backup and restoration processes),
- □ security protocols (guidelines on security measures, such as firewalls, antivirus software, and access controls),
- □ troubleshooting guides, and
- □ disaster recovery plans.

Process Documentation

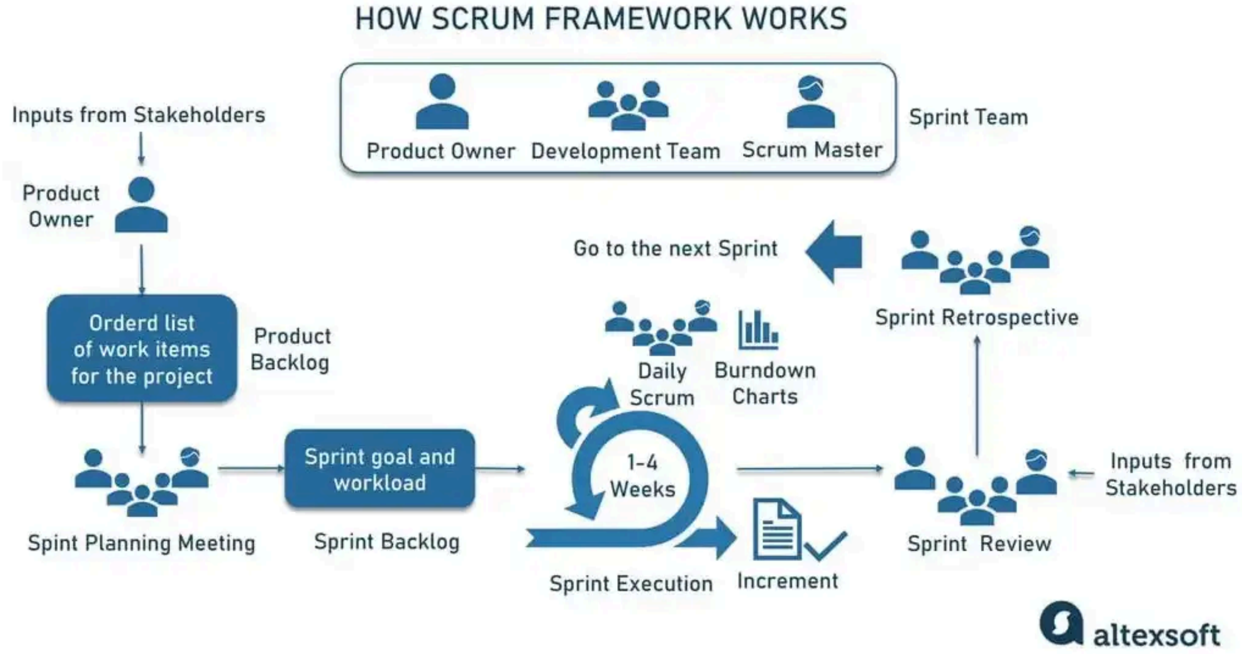
Process documentation covers all activities surrounding product development. The value of keeping process documentation is to make development more organized and well-planned.

Agile product roadmaps

Product roadmaps are used in Agile software development to document the vision, strategy, and overall goals of the project. They help keep the course of development in sync with initial goals. Depending on the type of product roadmap, it can express high-level objectives, task prioritization, the sprint timeline, or low-level details.

Topic: System Implementation

Backlogs
Backlogs are a fundamental component of Agile project management methodologies, particularly in frameworks like Scrum. They provide a clear, actionable plan for the team as they work on the product.



Other process documents: plans, standards, strategies, etc.
There are many other types of process documentation that are worth creating to either describe the procedures in a company or reflect the specific processes during a certain project.

Standards include all coding, testing, and design standards that the team adheres to throughout the project.

Plans, estimates, and schedules are usually created before the project starts and can be altered as the product evolves.

A test strategy is a high-level document that describes the organization’s overall software testing approach. It includes information about team structure and resource needs, as well as what should be prioritized during testing. A test strategy is usually static, as it is defined for the entire development scope.

A release checklist is a list of tasks and checks to be completed before software is released, ensuring that nothing important is overlooked.

Working papers record engineers’ ideas and thoughts during project implementation. Working papers usually contain some information about an engineer’s code, sketches, and ideas on how to solve technical issues. While they shouldn’t be the major source of information, keeping track of them allows for retrieving highly specific project details if needed.

Reports reflect how time and human resources were used during development. They can be generated on a daily, weekly, or monthly basis.

Who writes tech documentation?

Creating technical documentation for a project isn’t an easy task. Many different stakeholders are involved in the process, each contributing specific expertise and perspectives to ensure the documentation is accurate, comprehensive, and useful.

Here’s a list of the key roles involved and their responsibilities.

Business analysts work closely with the customer to gather and define the business requirements that the product must meet. They also act as a bridge between nontechnical stakeholders and the technical team. They ensure that the technical documents are aligned with business goals and are understandable to nontech stakeholders. In addition, BAs often contribute to the creation of user documentation.

Market analysts collect information from end users to create user personas and user scenarios. They also research the market and contribute to the requirement documents by validating market fit and business objectives.

Project managers oversee the documentation process to ensure it aligns with project timelines and goals. They are responsible for the majority of process documents and plans related to the project.

Solution architects create comprehensive architectural design documentation. These documents outline the proposed architecture for the project, including high-level structures, software design, and integration of various components and external systems. Architects also contribute to the creation of technical roadmaps and set the technical standards and guidelines for the project.

Developers create source code documentation that describes the software elements. They also explain how features are implemented, provide code samples, and clarify complex technical processes that need to be documented.

UX designers are involved in creating UX/UI design documentation that includes interface descriptions, prototypes, site maps, etc.

QA engineers contribute by documenting testing protocols, results, and configurations. They ensure that the documentation reflects the necessary steps for effectively testing the software and reporting bugs.

Topic: System Implementation

Technical writers are primarily responsible for drafting and editing the documentation. They collaborate with technical experts to gather accurate details and present them in a user-friendly format.