Product documentation

Product documentation describes the software product to be developed and is aimed at different audiences including business analysts, developers, testers, users, support and maintenance teams. The main product documentation that should be produced in a project are detailed below.

- Software requirements document [Mandatory]
- Design documentation [Mandatory]
- API documentation [Mandatory]
- Test documentation [Mandatory]
- Data migration documentation [Mandatory]
- User documentation [Mandatory]
- Software configuration and maintenance documentation [Mandatory]

Software requirements document [Mandatory]

A software requirements document provides information about the required system functionality based on the needs of the institutions. Requirements are statements of what a system should do. They include functional and nonfunctional requirements and detail the product’s purpose, expected features and behavior. Functional requirements define the features the product needs to have to support the needs of users. In the Agile methodology user stories are used to describe requirements from the perspective of the end user. Non functional requirements include requirements for usability, performance and security.

Design documentation [Mandatory]

Software architecture design documents, sometimes also called technical specifications, include the main architectural decisions made by the solution architect. Unlike software requirement documents that describe what needs to be built, the architecture design documentation is about how to build it. It describes in what way each product component will contribute to and meet the requirements, including solutions, strategies, and methods to achieve that. It includes:

- Architecture & design principles - It defines the guiding architecture and design principles to be used to engineer the product. For instance, it may define that the solution will be built using microservices architecture
- Solution design details. Describes the contemplated solution by listing planned services, modules, components and their importance. The design details include:
  - ○ System architecture
  - ○ User Interface (UI) Design
  - ○ Integration Design
  - ○ Database Design
- Diagrammatic representation of the solution. Includes diagrams and/or other graphic materials to help understand and communicate the structure and design principles.

> In the agile methodology, the design, development and testing phases are iterative and therefore the design documentation can be updated incrementally.

API documentation [Mandatory]

API documentation contains instructions about how to effectively use and integrate with an API. It’s a concise reference manual containing all the information required to work with the API, with details about the functions, classes, return types, arguments and more, supported by tutorials and examples. A suitable tool can be used to automate the documentation of APIs

Test documentation [Mandatory]

Test documentation describes the process, objectives, and results of software testing. It can also include information on the environment, setup, and configuration required to perform testing. Test documentation is used to communicate the details of a test plan or strategy to stakeholders, developers, and testers. Test documentation includes:

- A **test strategy** is a document that describes the software testing approach to achieve testing objectives. This document includes information about team structure and resource needs along with what should be prioritized during testing. It should include details of how the various types of testing should be performed including functional testing, performance and security testing. A test strategy is usually static as the strategy is defined for the entire development scope.
- A **test plan** is a brief document that describes what should be tested at a given moment. In the agile methodology, a test plan can be defined for each sprint. This document should contain:
  - the list of features to be tested
  - testing methods
  - timeframes
  - roles and responsibilities
- **Test case** documentation is a set of detailed actions/ scenarios and expected results to verify each feature or functionality of a product. Test case specifications are based on the approach outlined in the test strategy and test plan.
- **Test reports** - test reports list of tests that have been run at a particular time. It represents what tests are completed and how many have passed or failed. It also includes a list of issues which have been identified as part of the testing process.

Data migration documentation [Mandatory]

Data migration documentation is required when a software project involves moving data from a legacy software system to the new software system. It can also include moving data from manual files or spreadsheets to the new system. Types of data migration documentation includes:

- **Data migration strategy** - a data migration strategy defines the scope, approach to be used in data migration as well as roles and responsibilities. It defines scope of data migration which include the data sets to be migrated including historical period to be included e.g last 5 years. It also defines the approach to be used to migrate the data for each data set including work required to cleanse or convert the data
- **Data migration reports** - Data migration reports are used to record the results of a data migration process which includes validating that the data has been migrated successfully. The process of data migration may include several test cycles to test effectiveness of data migration tools and quality of data. For each cycle a data migration report is produced capturing the data migrated and any issues faced in the migration process. A final migration report is produced when data is migrated to the production environment.

User documentation [Mandatory]

This documentation is created for end-users and should explain in the simplest way possible how users can effectively use the software. User documentation provides an overview of the product’s functionality and gives basic guidelines on how to use it. The documentation can be provided in various forms including printed form, online or offline on a device. Institutions can be creative on how to develop the user documentation based on the complexity and need. For example online end-user documentation can come in the form of knowledge bases and include FAQs, video tutorials, embedded assistance and support portals

Software configuration and maintenance documentation [Mandatory]

Software maintenance and configuration document is a document that provides key information required to effectively maintain the software. It includes the configurations required for the software including security settings, installation instructions, version control management and change management controls. It captures the following:

- **System Overview:** A high-level description of the software system, including its purpose, functionality, and architecture.
- Configuration Items: A list of all the software components or items that need to be configured, such as modules, libraries, databases, and external dependencies.
- **Configuration Management Plan:** Procedures and guidelines for managing changes to the software configuration throughout the development lifecycle, including version control, release management, and change control processes.
- **Configuration Settings:** Detailed specifications for each configuration item, including parameters, options, default values, and any dependencies or constraints.
- **Installation Instructions:** Step-by-step instructions for installing and configuring the software system on different platforms or environments.
- **Integration and Interoperability:** Information on how the software system interacts with other systems, including APIs, protocols, data formats, and integration points.
- **Performance and Scalability:** Guidelines and recommendations for optimizing the performance and scalability of the software system through configuration settings and tuning parameters.
- **Security Considerations:** Requirements and best practices for securing the software system, including authentication, authorization, encryption, and data protection measures.
- **Testing and Validation:** Procedures and criteria for testing and validating the configuration settings to ensure that the software system meets its functional and non-functional requirements.
- **Documentation and Support:** References to additional documentation, support resources, and contacts for assistance with configuring and using the software system.