techitive press

Understanding Source Documents

Tenneti C S

Copyrights

Author: *Chakravarthy Srinivas Tenneti*

Book: *Understanding Source Documents*

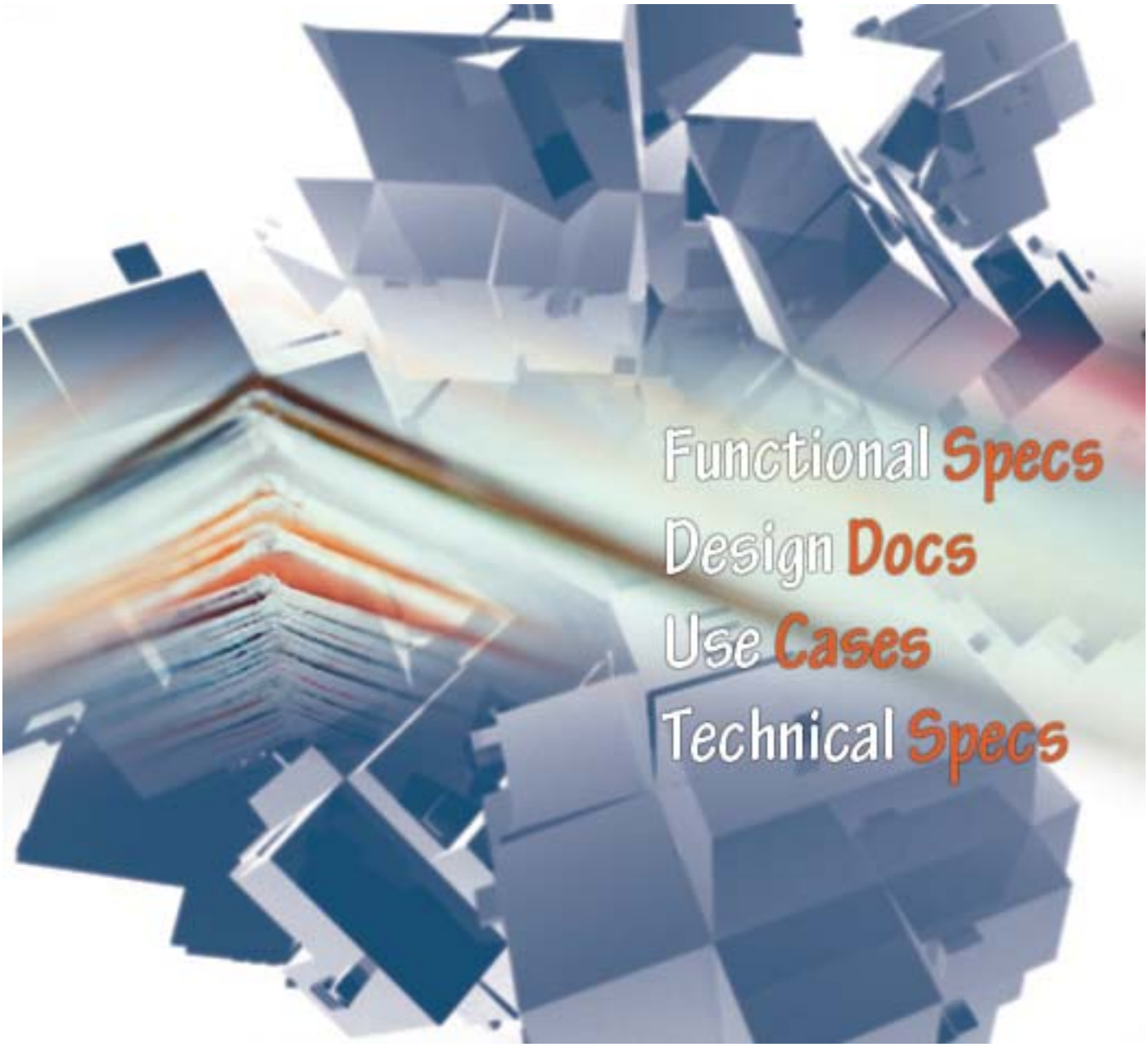
You can find further information and articles about technical writing at http://techitive.com.

Reading Source Documents

Source documents include functional specifications, system requirements specifications, technical design specifications, architecture documents, troubleshooting documents, and other design documents. These documents act as the basic plan for the developer to build the application. Hence, these are a good starting point from the documentation point of view as well.

You may not always get all the required information from the source documents. They may contain information irrelevant to the end user. You need to read such information and analyze the impact on the end user. Hence, the basic tasks you must perform include:

- ***Studying and discerning the available information***
- ***Identifying gaps and problem areas***
- ***Making notes and establishing relationships***
- ***Listing down questions for the subject matter experts***

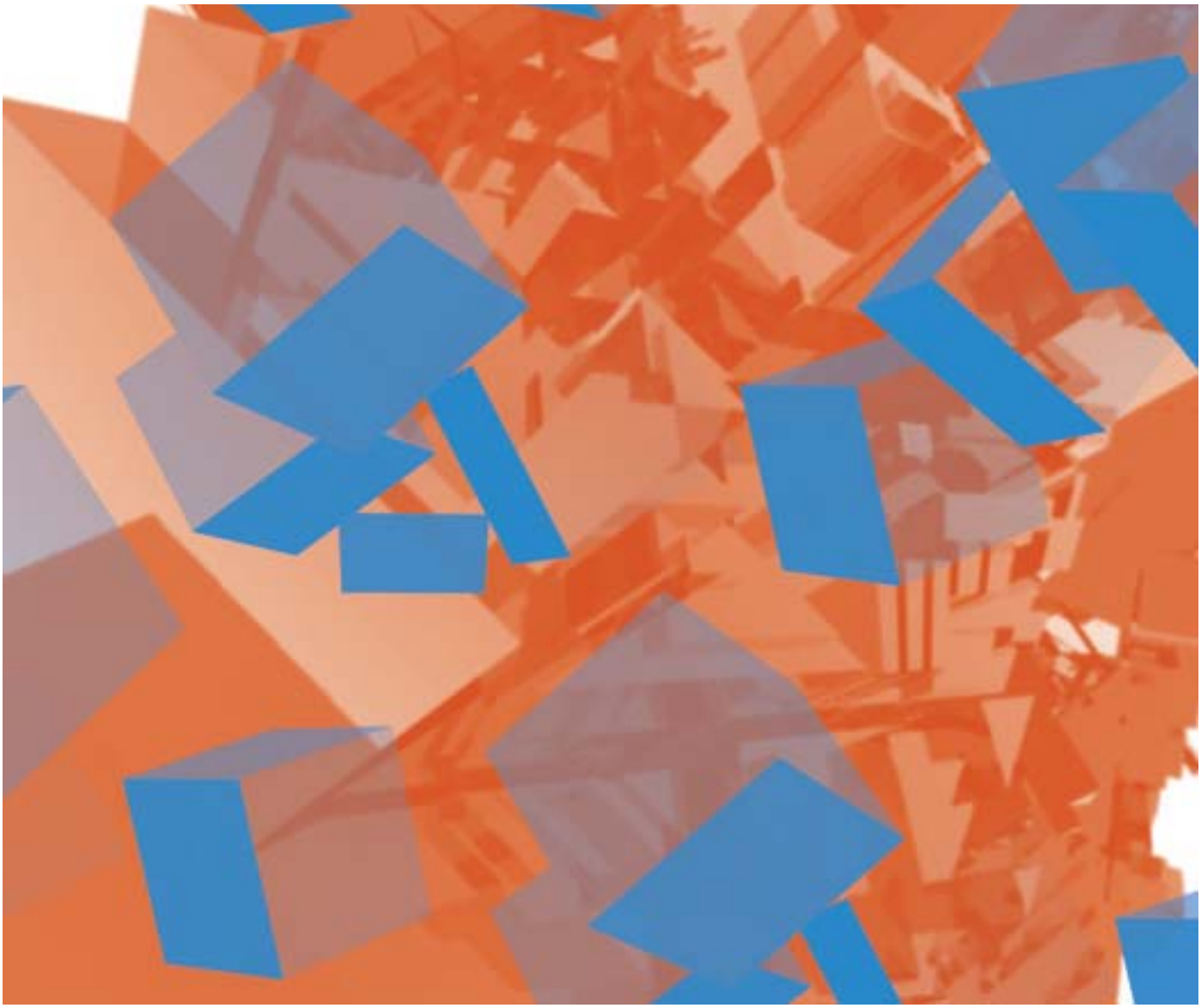


Types of Source Documents

There are different types of source documents that are available. The product development team, the quality assurance team, and the technical publications team depend upon the source document to plan, schedule, and execute their work. Hence, these source documents act as a reference point for all the parties involved in the product development during its various stages.

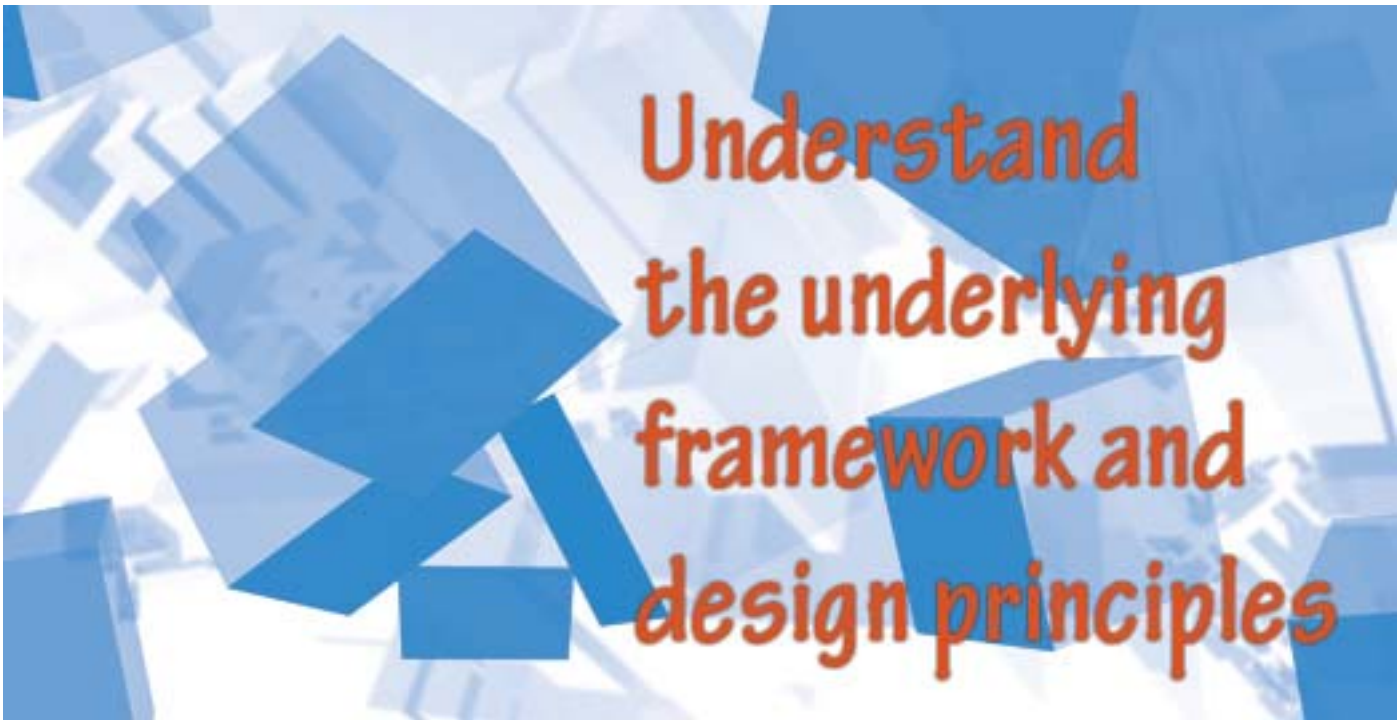
The common types of source documents are as follows:
1. Design Documents
2. Requirements Specifications
3. Architecture Documents
4. Use Cases

As stated earlier, these documents act as a reference for different teams. Hence, you need to be adept in the information provided in these documents.



Design Documents

The system architect or the software designer prepares the design document that describes the product in detail. It acts as a beginning point for the product development team. These documents help in providing a broad vision to all the teams involved in product development. Design documents are classified either as high level and low level design documents.



Contents

The typical sections in a design document are as follows:

1. The basic architecture of the system – lists out the components and the logical and informational flow between the various components. It is normally depicted in the form of an architecture diagram or a table.
2. The interface elements of the system – details the various interfaces including the human interface to the system.
3. The procedural design of the system– explains the programming constructs such as the procedures, methods, and references that are necessary to build the system.

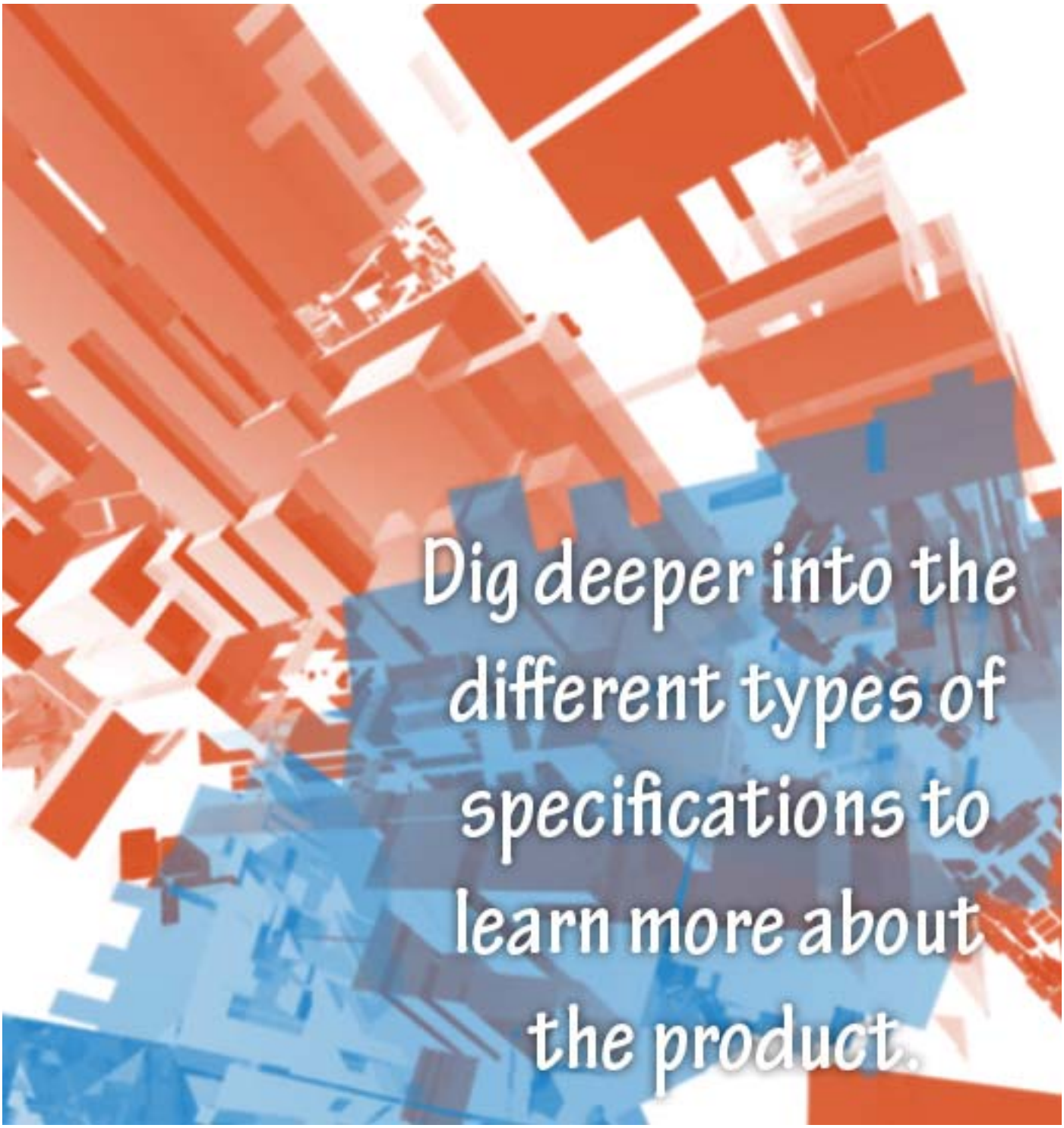
Derivatives

Information from the design documents can be used for various end user documents such as the following:

1. **User's guide** – The detailed technical specifications, the list of features and their explanation can be used to prepare and update the User's guide.
2. **Online Help** – The features and the interrelationship between the various components can be used to prepare or update the Online Help.
3. **Architecture planning guides** – The detailed technical specifications, the components of the system, and the details to begin using the system can be used to prepare physical and virtual planning guides that help the users plan and implement their system and environment.
4. **API documentation** – The high-level technical information such as the procedural elements and programming constructs can be used to prepare and update the API deliverables.

Requirements Specifications

After the analysis of the product to be developed is done, the architect or the system engineer prepares the requirements specifications document. This document explains the detailed requirements of the system to be developed along with the behavior details of the system. It lists the client's requirements from the system. It is normally drafted and sent to the client party to get a consensus on the product to be developed. While the design document lists the basic design elements of the system, the specifications documents explain in detail the system requirements so that the development team can adhere to the mutually agreed upon requirements.



Functional Specifications

This document lists the features of the product and the various functionalities that must be available in the product. You can use this document to update the User's guide and Online Help and other end user documents where the functionality of the system needs to be clearly explained.

Contents

Typical sections of the functional requirements document include the following:

1. **Purpose and scope** - This section describes the basic purpose of the system, along with identification details of the system. It also includes the scope of the system and the performance of the system within this scope.
2. **Assumptions and constraints** - This section provides a list of all technical, legal, and task level assumptions and constraints that have an impact of the system. Assumptions include the details of situations that are beyond human control and have an impact on the success of the system. Constraints list the limitations within which the system must be designed. Legal requirements, business conditions, market situation, and other technical and socio-economic conditions fall under the category of constraints.
3. **User requirements** - Along with providing a list of users and the different classes/categories of users, this section provides detailed description about all the users. All the functional requirements are related to the needs of the user. The user requirements can either be in the form of text or process flow diagrams.
4. **Data Flow Diagrams** - A DFD is a graphical representation of the flow of the data through the various components in an information system. The data that flows within the components through the sub-components will also be depicted using DFDs.
5. **Functional requirements** - This section includes the list of functions or features that are expected to be developed as a part of the system. You can use this information to document concepts guides and end user guides. However, you need to decipher information that is relevant to the end user in the context of your audience analysis.
6. **Performance requirements** - This section includes the performance expectations from the system with regard to all the functional aspects of the system.

Technical Specifications

This document lists all the technical requirements of the system (that are not functional) and the technical expectations of the system. This document also includes non-functional requirements such as quality standards, performance requirements, technical constraints, and design constraints of the system.

Contents

Typical sections of a technical specifications document include the following:

1. **Technical overview** - This section provides and overview of the system in technical terms. It also includes architecture diagrams that explain the technical elements of the system.
2. **System specifications** - This section provides the tangible technical details. In case of a technical specifications document for a hardware system, the detailed technical information such as the measurements of the system, environment details of the system, and other physical and technically-relevant information are included. Most of these technical details will be helpful when you are drafting documents for technical people such as architecture documents and troubleshooting guides.
3. **Interface design** - This section details the various technical interface elements. It includes the list of system objects used for development and their interrelationship. Such information can be used to develop API documentation.
4. **Functions, methods, and their relationships** - These are documented in the case of a software product to let the developers understand the intricacies of the system. A writer can use such information to understand the logical and technical relationships before translating it into audience-friendly topics or chapters.
5. **Other sections** – Details about the user's technical requirements and performance requirements that are akin to the functional specifications document are included in all the other sections as a part of this document.

Use Cases

A use case describes the system's relationship with external elements and its behavior with all the external elements. Use cases, typically, contain life like examples of the usage of the system. These examples are depicted in the form of scenarios with the help of information from functional requirements.

Contents

You need a good number of use cases to describe the process to achieve various goals or tasks using all the features of the system. Each feature can be explained using different use cases and need not be restricted to one feature.

The concept of Use Cases originated from Unified Modeling Language (UML). In the context of UML, a use case diagram is a behavioral diagram that provides the functionality of the system in terms of actors, and their goals, and any dependencies between the various use cases.

Software blueprints can be written and constructed using Unified Modeling Language. UML is a visual language for modeling and communicating about systems through the use of diagrams and supporting text.

UML is a combination of graphical interpretation with well-defined semantics to ensure that different people can interpret the model unambiguously.

UML models can be directly used for construction by connecting to a variety of programming languages. While graphical expression is done using UML, textual expression is done using the various programming languages.

A typical use case consists of the process of simulating a real time environment, defining the various interactions and processes, identifying the roles played, and establishing patterns.

About the Author:

Chakravarthy Srinivas Tenneti is a technical writer with more than eight years of experience. Tenneti also likes composing music and often indulges in web design and graphic design.