Object Oriented Software Engineering

Chapter 2: SRS Documentation

What is SRS?

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

Why SRS?

In order to fully understand one’s project, it is very important that they come up with an SRS listing out their requirements, how are they going to meet them and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

Characteristics of SRS

1.Correctness:

User review is used to ensure the correctness of requirements stated in the SRS. SRS is said to be correct if it covers all the requirements that are actually expected from the system.

2.Completeness:

Completeness of SRS indicates every sense of completion including the numbering of all the pages, resolving the to be determined parts to as much extent as possible as well as covering all the functional and non-functional requirements properly.

3.Consistency:

Requirements in SRS are said to be consistent if there are no conflicts between any set of requirements. Examples of conflict include differences in terminologies used at separate places, logical conflicts like time period of report generation, etc.

4.Unambiguousness:

A SRS is said to be unambiguous if all the requirements stated have only 1 interpretation. Some of the ways to prevent unambiguousness include the use of modelling techniques like ER diagrams, proper reviews and buddy checks, etc.

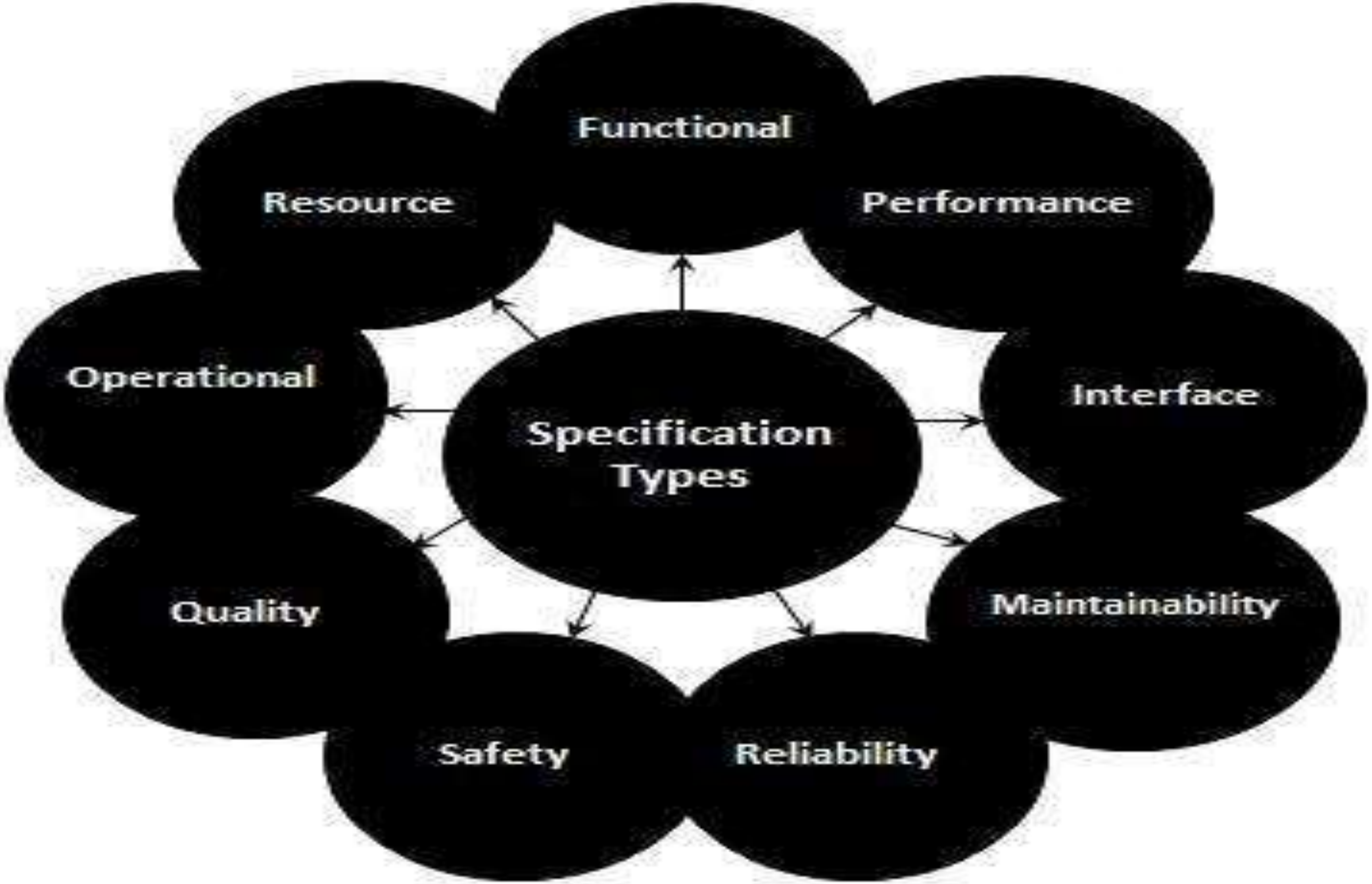
5.Traceable

- SRS is traceable when the source of each requirement is clear and facilitates the reference of each requirement in future
- For this, forward tracing and backward tracing are used
- Forward tracing implies that each requirement should be traceable to design and code elements
- Backward tracing implies defining each requirement explicitly referencing its source

What is Software Requirement Specification - [SRS]?

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

Types of Requirements:


Functional
Resource
Performance
Operational
Specification Types
Interface
Quality
Maintainability
Safety
Reliability

Types of Requirements:

1) Functional Requirement :

Functional requirement are statements or goals used to define system behavior .Functional requirements define what a software system must do or not do.

2) Non functional requirements : Non functional requirements relate to software usability .Non functional software requirements define how the system must operate .A system can meet its functional requirements and fail to scalability , security .

They basically deal with issues like :

Security

Maintainability

Reliability

Performance

Reusability

3)Domain requirements:

Domain requirements are the requirement which are characteristic of a particular category or domain of project . Domain requirements can be functional or non functional.

IEEE Standard Format of SRS :

The IEEE std 830-1998 was created to standardize the software requirements specification document.
The aim of an SRS document is to capture requirements in an unambiguous manner in order to facilitate communication between stakeholders.
IEEE std. provide a structure (template) for documenting the software requirements.

SRS Format :

SRS is the standard statement of what the system developers should implement.
SRS includes the user’s requirements for a system and detailed specification of the system requirement.
Structure of SRS document:
Section 1: Product Overview and summary
Section 2: Development , Operating and maintenance environments
Section 3: External interfaces and data flow
Section 4 : Functional requirements
Section 5 : Performance requirements
Section 6 : Exception handling
Section 7 : Early subsets and Implementation priorities
Section 8:Foreseeable modifications and enhancement
Section 9 : Acceptance criteria
Section 10 : Design hints and guidelines

Section 11 : Cross reference index
Section 12 : Glossary of terms

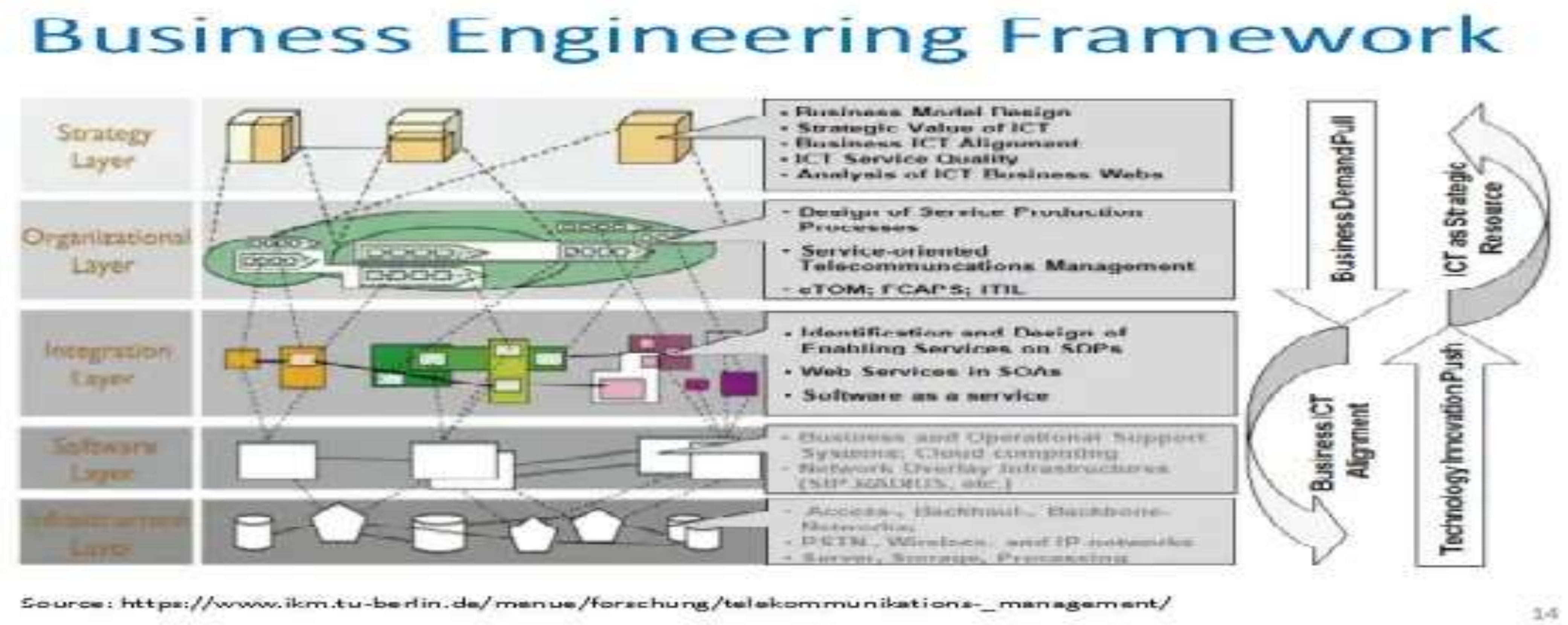
Requirement Elicitation :

The requirements elicitation process may appear simple: ask the customer, the users and others what the objectives for the system or product are, what is to be accomplished, how the system or product fits into the needs of business, and finally, how the system or product is to be used on a day-to-day basis.

Elicitation requirements is difficult because of

1.'**Problems of scope'**. The boundary of the system is ill-defined or the customers/users specify unnecessary technical details that may confuse, rather than clarify, overall system objectives.
2.Problems of understanding. The customers/users are not completely sure of what is needed, have a poor understanding of the capabilities and limitations of their computing environment, don't have a full understanding of the problem domain, have trouble communicating needs to the system engineer, omit information that is believed to be "**obvious**," specify requirements that conflict with the needs of other customers/users, or specify requirements that are ambiguous or untestable.
3.Problems of volatility. The requirements change over time.

Business Engineering :



Business Engineering Framework

- Business Model Design
- Strategic Value of ICT
- Business ICT Alignment
- ICT Service Quality
- Analysis of ICT Business Webs

- Design of Service Production Processes
- Service-oriented Telecommunications Management
- eTOM; FCAPS; ITIL

- Identification and Design of Enabling Services on SDPs
- Web Services in SOAs
- Software as a service

- Business and Operational Support Systems; Cloud computing
- Network Overlay Infrastructures (SIP, RADIUS, etc.)

- Access-, Backhaul-, Backbone-Networks;
- PSTN, Wireless, and IP networks
- Server, Storage, Processing

Source: https://www.ikm.tu-berlin.de/menue/forschung/telekommunikations-_management/

It is development and implementation of business solution from business model to business processes and organizational structure to information system and information technology
The business engineering focuses on developing innovative business solution that take a socio-Technical System approach
The goal of business engineering is to produce measurable result rather than just an arbitrary improvement

Business engineering framework distinguishes artefacts and models on a strategy, organizational, integration , software and IT infrastructure layer

The **Strategy** layer describe the positioning of an enterprise at a high level of abstraction .It is developed once the business strategy is defined.

The **organizational** layer specifies a firm’s organizational structure and it’s process organization

The **integration** layer describes how applications share or could share data and functions with other applications and databases.

The **software** layer determines the data elements and software application that support the business layer.

The **IT infrastructure** layer comprises the hardware platforms and communication infrastructure that support the applications.

Thank You