Software Requirements Specification Document

greenwich.edu.vn

UNIVERSITY of GREENWICH

Alliance with FPT Education

Table of Contents

- Requirement Process
- SRS Introduction
- General Description
- Functional Requirements
- Non Functional Requirements
- System Architecture
- System Models
- Appendices

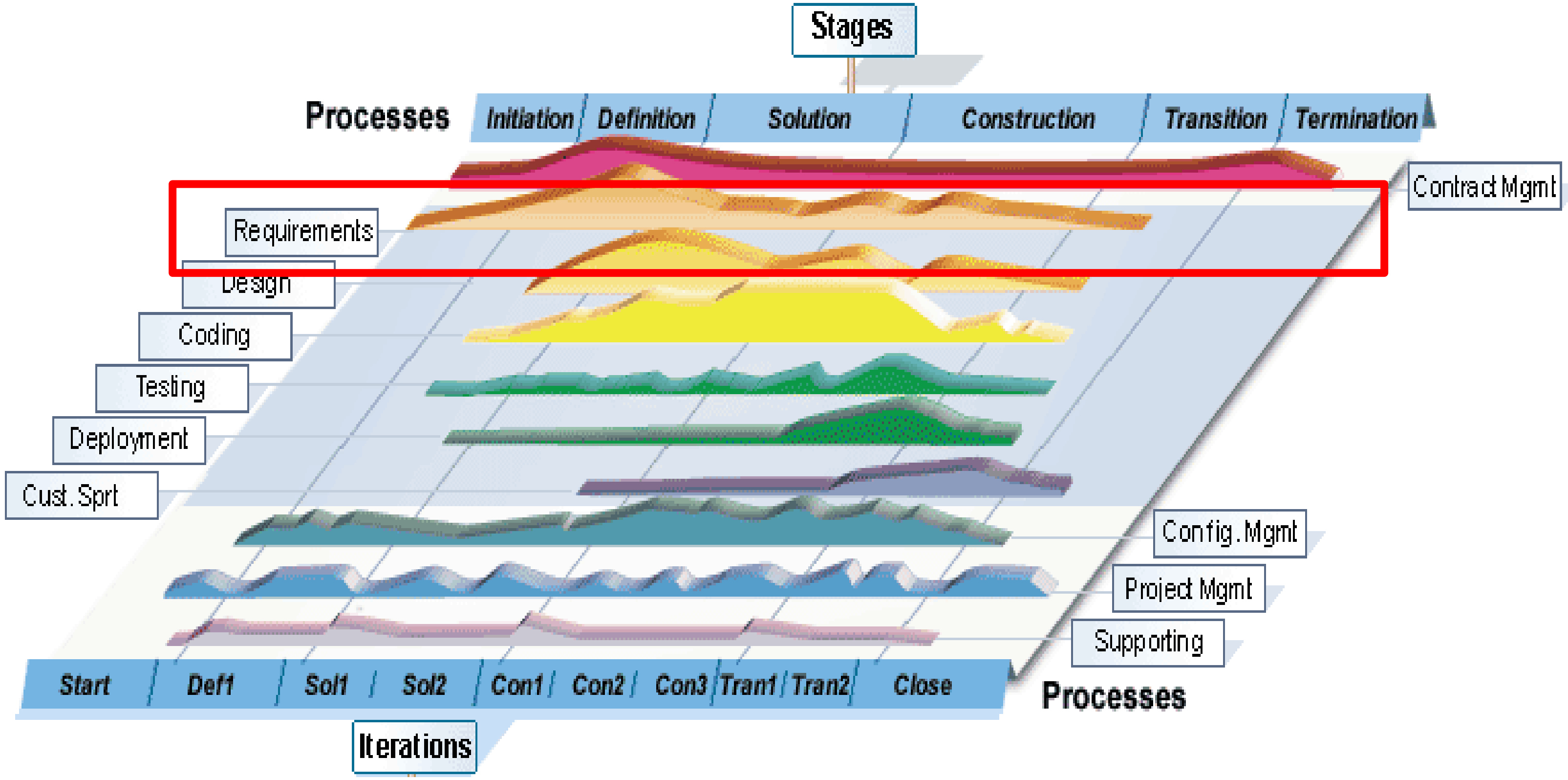
UNIVERSITY of GREENWICH

Alliance with FPT Education

REQUIREMENT PROCESS

Requirement Process

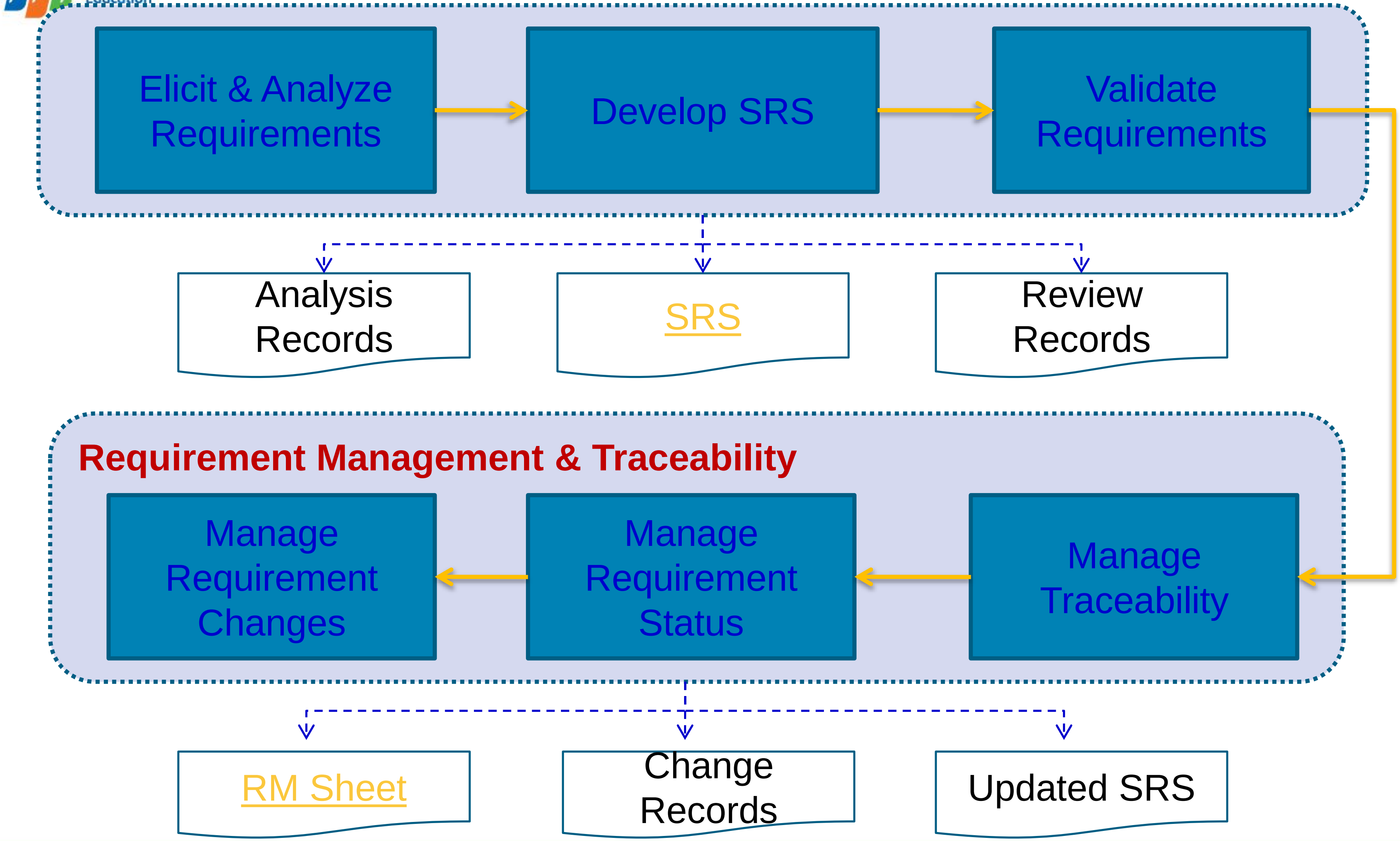
- First phase of Software engineering



Requirement Process Objectives

- To ensure that requirements for the software product are defined and understood.
  - – Get to know what customer’s requirement is
  - – Understand the customers’ needs & expectation
- To create SRS - Establish and maintain requirements agreement with the requestor and affected groups
- To ensure that the requirements are met.
- Requirements are documented and controlled to establish a basis for software development and project management use.

Requirement Process Workflow



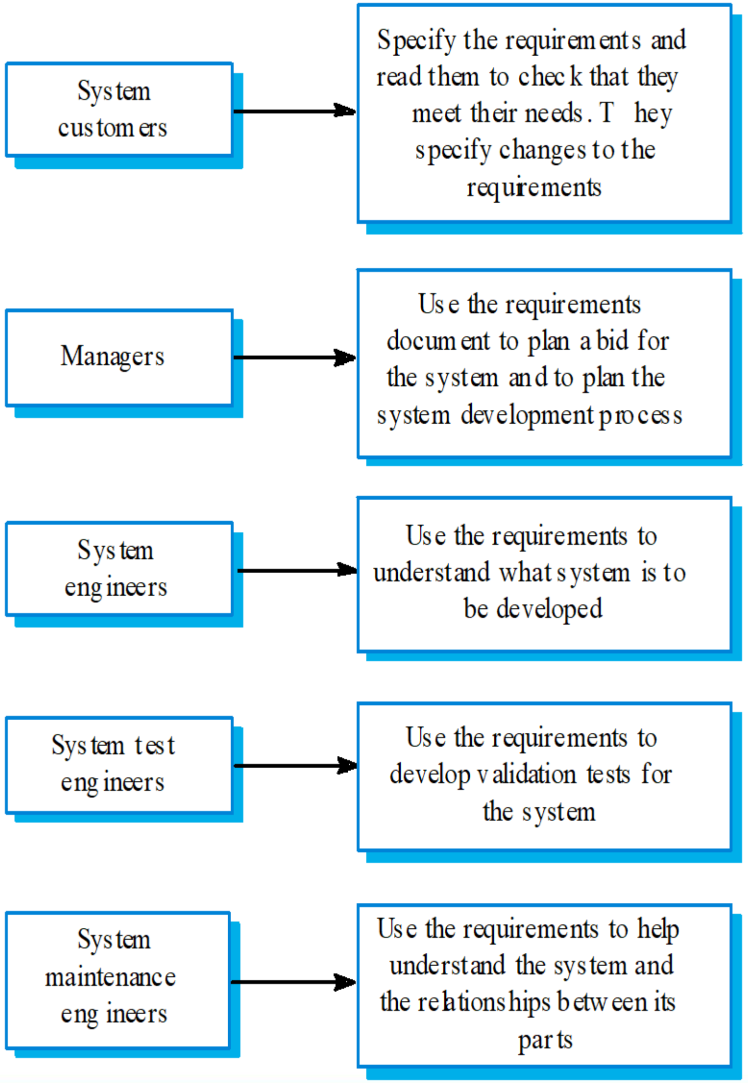
Elicit & Analyze Requirements
Develop SRS
Validate Requirements
Analysis Records
SRS
Review Records
Requirement Management & Traceability
Manage Requirement Changes
Manage Requirement Status
Manage Traceability
RM Sheet
Change Records
Updated SRS

Requirement Process
Elicit & Analyze Requirements

- Sometimes called **requirements discovery**
- Requirements are often not given to you, you have to **elicit** them; must work with **customers** and relevant **stakeholders** to elicit:
  - – the **services** that the system should provide
  - – the **constraints** that the system should satisfy
- Requirement analysis is done to:
  - – Detect and resolve conflicts between requirements
  - – Discover the bounds of the software and how it must interact with its environment
  - – Elaborate system requirements to derive software requirements

Requirement Process
Develop SRS – Requirement documents 1/3

- Requirements document is the official document of what is required for the system
- Often include only system requirements but sometimes may also include user requirements
- It is NOT a design document. Describe WHAT the system should do rather than HOW it should do



System customers → Specify the requirements and read them to check that they meet their needs. They specify changes to the requirements

Managers → Use the requirements document to plan a bid for the system and to plan the system development process

System engineers → Use the requirements to understand what system is to be developed

System test engineers → Use the requirements to develop validation tests for the system

System maintenance engineers → Use the requirements to help understand the system and the relationships between its parts

Requirement Process
Develop SRS – Requirement documents 2/3

- URD – User requirement definition
  - Address what users need to do their jobs
  - Composed all business requirements formulated by customer, business rules and other constrains
- SRS – Software requirement specification
  - A set of software requirements as complete, consistent, and correct as possible, from the developer's point of view
  - Document which after base lining, common reference point of the software requirements for customer, developer, tester and PM .

Requirement Process
Develop SRS – Requirement documents 3/3

- Benefit of good document
  - Basis for agreement between the customers and the team on what the software product is to do.
  - Reduce the development effort.
  - Provide a basis for estimating costs, schedules.
  - Provide a baseline for validation and verification.
  - Facilitate transfer.
  - Serve as a basis for enhancement

Requirement Process
Develop SRS – Steps & Activities

- Study URD:
  - Analyze user requirement
  - Prepare Q&A list to clarify unclear items with customers
  - Call/interview customers if needed
- SRS:
  - Develop use cases, system requirement
  - Develop functional specification
- Review and approve SRS:
  - Call up meeting for review
  - Keep meeting minutes records

Requirement Process
Develop SRS – Techniques

- Specify requirements using **structured natural language** (forms, tables, etc.)
- **Functional requirements** can be specified using modeling - a combination of graphical notations and structured natural language
  - Use cases, Use case diagrams, Use case specification
  - Activity Diagram, State Diagram
  - DFD, Concept ERD
  - Prototype: Screen Flow, Screen spec scpecification
  - ...
- **Non-functional requirements** can’t be modeled => specified using structured natural language only

Requirement Process
Develop SRS – Characteristics of good SRS 1/2

- Correct: requirement ~ what the software shall meet.
- Unambiguous:
  - Has only one interpretation (to both creator & user)
  - Use natural language & avoid the words like: maybe, generally, etc.
- Complete
  - Include all significant requirements.
  - Define all the software responses & include all the refs/labels.
  - Use of TBD: should avoid OR mention why, what to do, who, when.

Requirement Process
Develop SRS - Characteristics of good SRS 2/2

- Consistent: no conflict between individual requirements.
- Verifiable: reviewable & testable in finite cost-effective process.
- Traceable: clear origin & good reference for future develop/enhance documents.

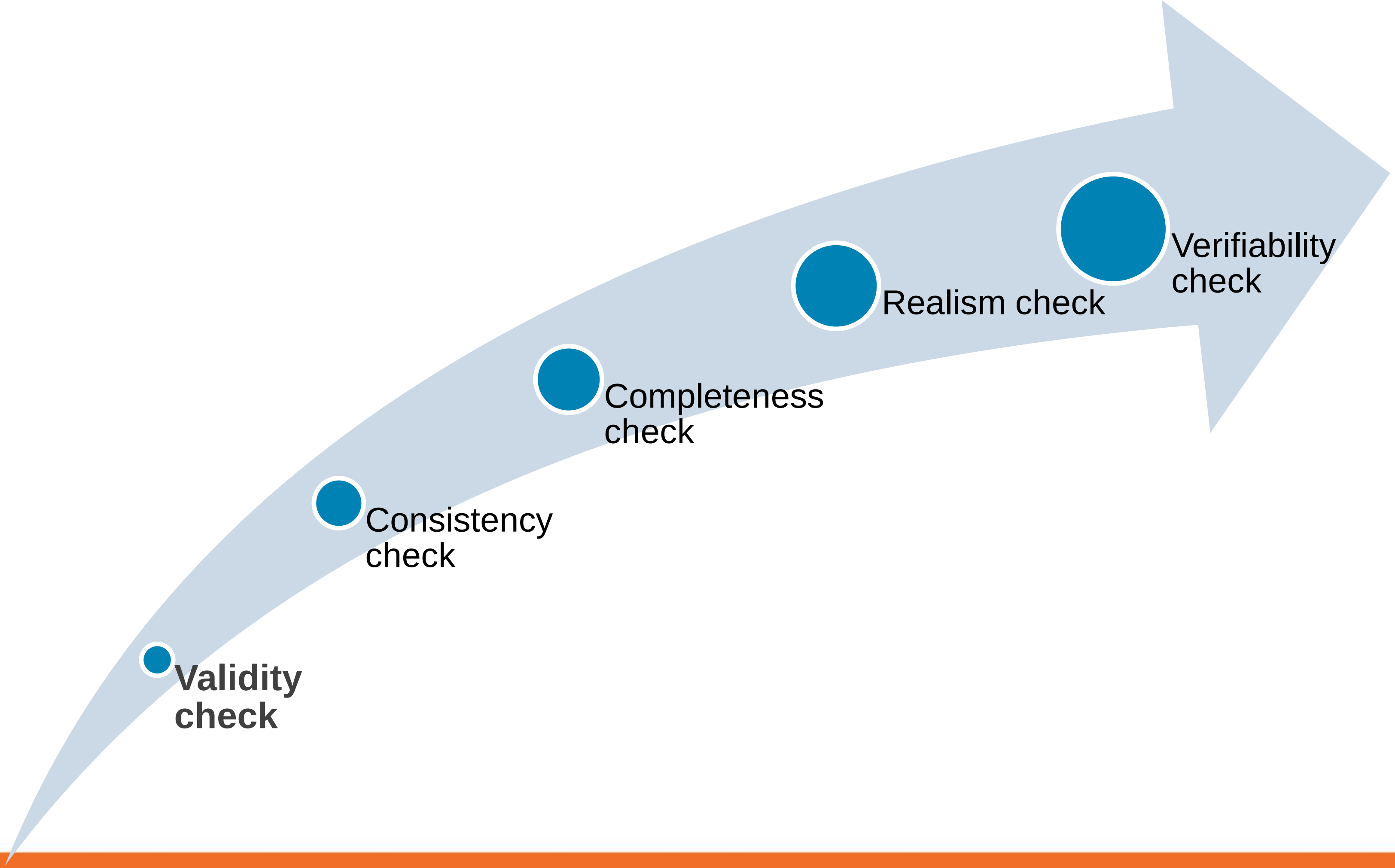
Requirement Process
Develop SRS – SRS Review Checklist

- SRS Review Checklist
  - To review the requirements by yourself
  - Make sure you understood completely the requirements:
    - Organization and Completeness: adequate, no missing, etc.
    - Correctness: no conflict, verifiable, in scope, message, etc.
    - Non-functional requirements, quality attributes, etc.

Requirement Process
Validate Requirements – Purpose

- Make sure that the requirements define the system that the customer really wants
- Requirements error costs are high so validation is very important
  - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error

Requirement Process
Validate Requirements – Process



Validity check

Consistency check

Completeness check

Realism check

Verifiability check

Requirement Process
Validate Requirements – Techniques

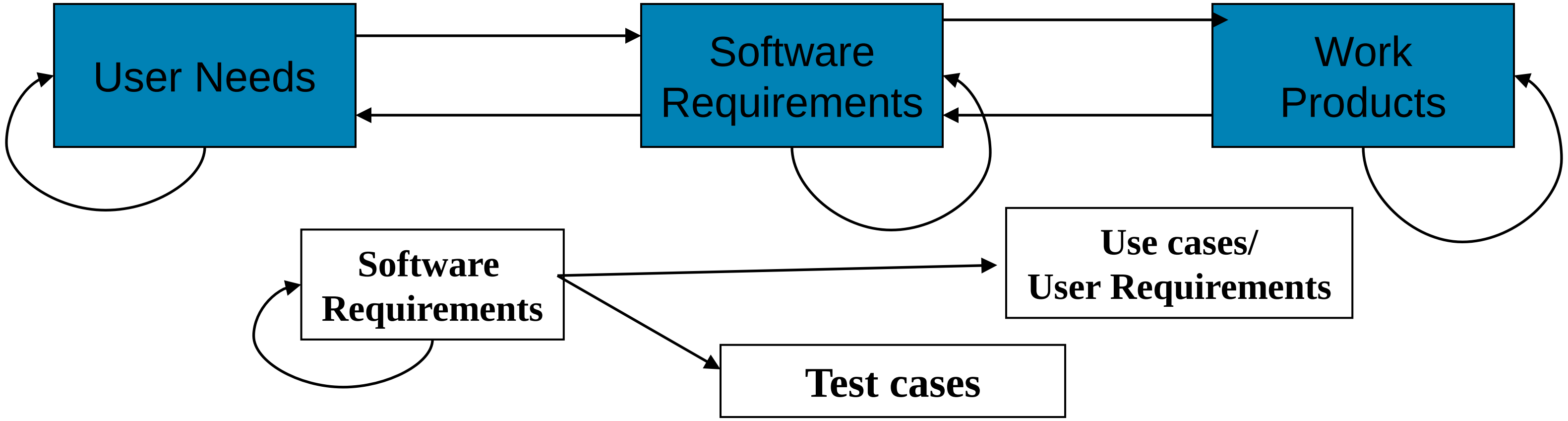
- Requirements Review
  - Systematic manual analysis of the requirements
  - Involving development staff, customers and relevant stakeholders
- Prototyping
  - Using an executable model of the system to check requirements
- Model Validation
  - Validate the quality of the models developed during analysis
- Test-case generation
  - Developing tests for requirements to check testability

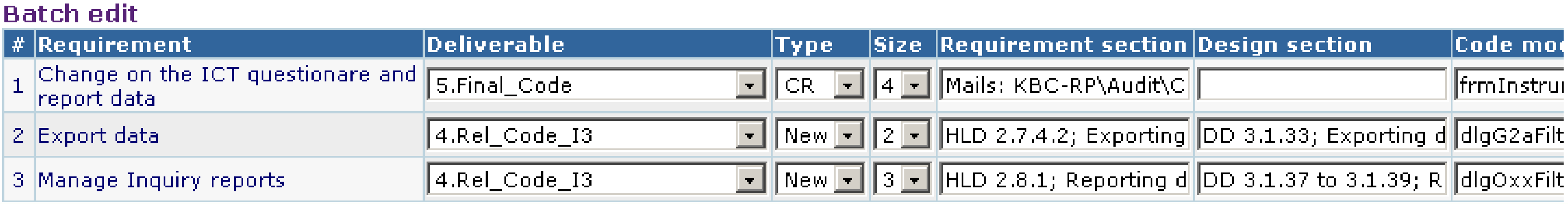
Requirement Process
Requirements management

- Manage requirement
  - Requirement Management Sheet, Excel sheet, used to track the status, relationship and change of requirements during the whole project.
  - A mandatory document (dynamic version of SRS)
    - Classify requirement to functional/non-functional requirement
    - To maintain the common reference for all related parties (traceability of requirement and software product)
    - To track the project progress (status of requirement)
    - To track the change (including change request)
    - To collect requirement related metrics for reporting
  - The sheet is created the first time client requirement come

Requirement Process
Manage Traceability & Requirement Status

- Why is traceability necessary?
  - The requirements can change at any stage during the product’s life.
  - If the requirements are traceable, then when changes happen, it is far easier to find the impacted parts of the product

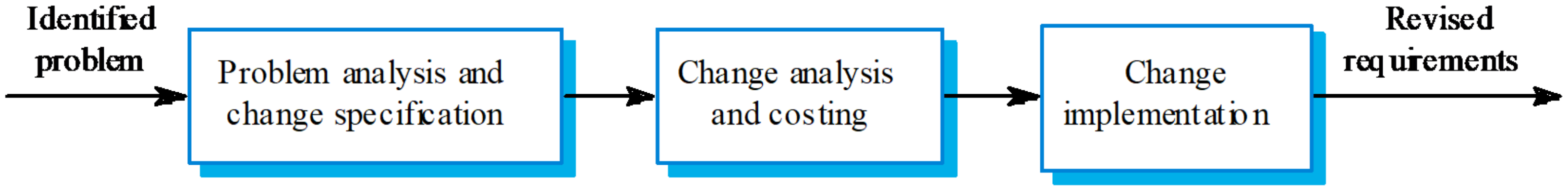



Batch edit

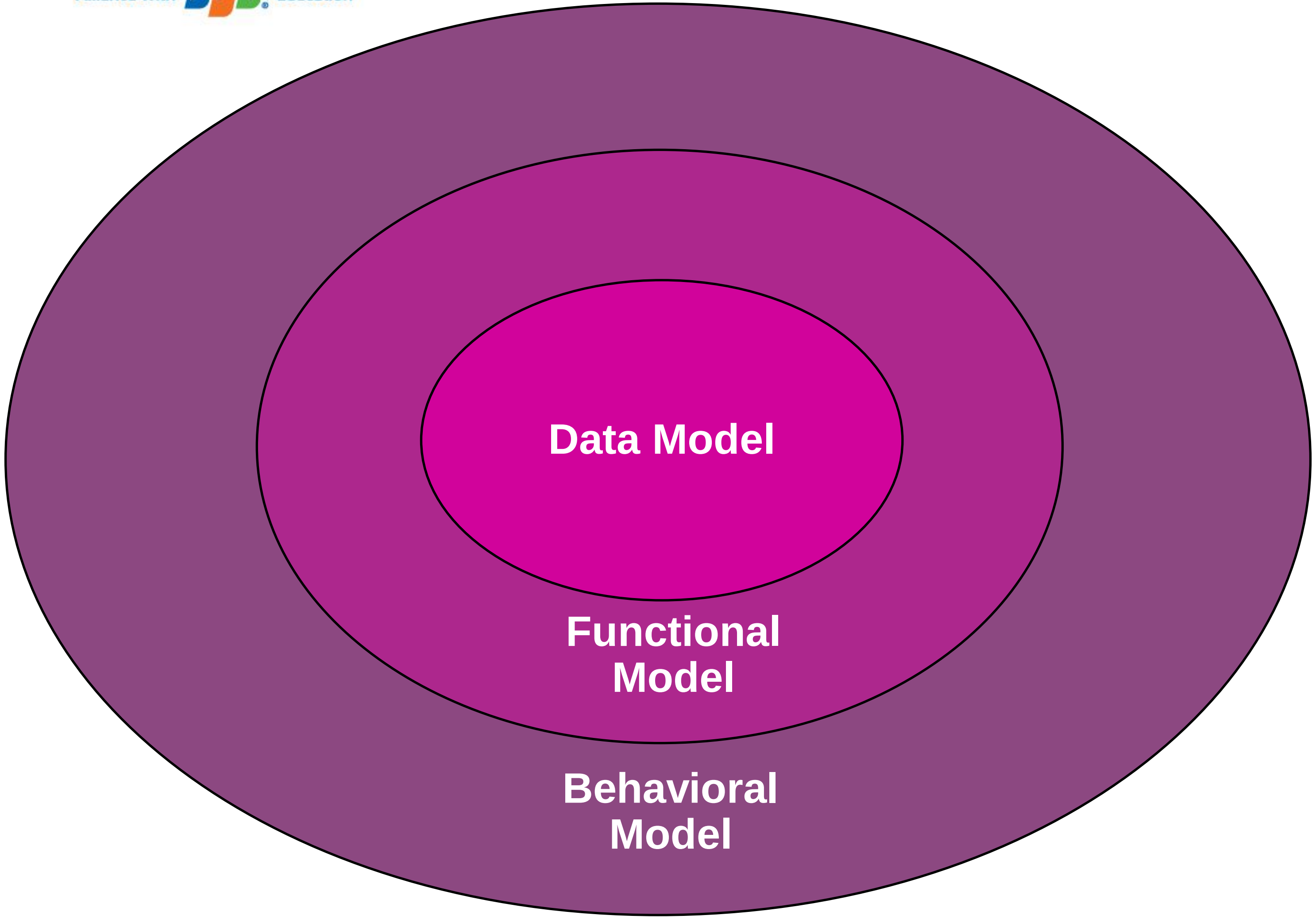
#	Requirement	Deliverable	Type	Size	Requirement section	Design section	Code mo
1	Change on the ICT questionare and report data	5.Final_Code	CR	4	Mails: KBC-RP\Audit\C		frmInstru
2	Export data	4.Rel_Code_I3	New	2	HLD 2.7.4.2; Exporting	DD 3.1.33; Exporting d	dlgG2aFilt
3	Manage Inquiry reports	4.Rel_Code_I3	New	3	HLD 2.8.1; Reporting d	DD 3.1.37 to 3.1.39; R	dlgOxxFilt

Requirement Process
Manage Requirements Changes & Status

- Requirements change (CR – Change request)
  - The priority of requirements from different viewpoints changes during the development process
  - Customers may specify requirements from a business perspective that conflict with end-user requirements
  - The business and technical environment of the system changes during its development
- Requirements change process


Identified problem
Problem analysis and change specification
Change analysis and costing
Change implementation
Revised requirements

Systems Requirements Specification



The SRS is composed of the outer layer of the behavioral model, the functional model, then the data model.

Systems Requirements Specification

I. Introduction

A Purpose

B Scope

C Definition, Acronyms, or Abbreviations

D References

E Overview

I. Introduction

A. Purpose

- The purpose of this Software Requirements Specification document
- Intended audience of this document

I. Introduction

A Purpose

The purpose of the Software Requirements Specification document is to clearly define the system under development, namely the Video Rental System (VRS). The intended audience of this document includes the owner of the video store, the clerks of the video store, and the end users of the VRS. Other intended audience includes the development team such as the requirements team, requirements analyst, design team, and other members of the developing organization.

I. Introduction

B. Scope

– Origin of need
– High-level description of the system functionality
– Goals of proposed system

I. Introduction

B. Scope

- Origin of need
  - who and what triggered the request for this software development activity
  - gives developers an understanding of the goals for the proposed system

I. Introduction

B. Scope

- High-level functionality
  - – defined for the system
  - – usually in list separated by commas

B. Scope

- Goals are general purposes of a system. They are fuzzy and non measurable.
- A typical goal would be things like
  - Increase customer satisfaction
  - Make xyz easier for the customer
  - Improve customer relationships

I. Introduction

B. Scope

The owner of a local video store wanted to create a new business plan where everything about renting a video (except the picking up and returning of videos) was done online. Therefore, the new VRS will allow the following functionality online: to search for videos, to become members, to rent videos, to modify membership information, and to pay overdue fees. The store personnel may use the VRS to process the rented or returned videos, to add or remove videos to/from his store’s video inventory and to update video information. The VRS is intended to increase the owner’s profit margin by increasing video sales with this unique business approach and by allowing him to reduce the staffing needed in his stores.

I. Introduction

C. Definitions, Acronyms

– As you begin to define a system, you will encounter words which need definition and general usage acronyms. These should be documented for new personnel and for clarity of all concerned parties.

I. Introduction

C. Definitions, Acronyms

MVC – Model View Controller

CS - Computer Science

MSES - Masters in Software Engineering Science

DOE - Department of Education

….

I. Introduction

D. References

- Many references may be used to define existing systems, procedures (both new and old), documents and their requirements, or previous system endeavors. These references are listed here for others.

- If any of these references are provided in the appendices, it should be noted here.

I. Introduction

D. References

Clerk - Personnel staff who is working in a video store

Customer - Anyone who interacts with the VRS with becoming a member

Functional requirement - A service provided by the software system

Member - Anyone who registers with the VRS to acquire membership in the video store

I. Introduction

E. Overview

- – This section defines the organization of the entire document. It will lay the framework for reading the document.

I. Introduction

E. Overview

Section 2 of the SRS describes the product in more detail. Section 3 provides a complete list of the functional requirements of the intended system. Section 4 provides the non-functional requirements. Section 5 shows the class diagram, and Section 6 the use case diagram. The appendices appear next.

I. Introduction

I.A Purpose	Paragraph form
I.B Scope of the System Specified	Paragraph form
I.C Definitions, Acronyms, and Abbreviations	Table form or bulleted list
I.D References to Supporting Documents	Bulleted list
I.E Overview of rest of SRS	Paragraph form

Systems Requirements Specification

II. General Description

A Product Perspective

B Product Functions

C User Characteristics

D General Constraints

E Assumptions

II. General Description

A. Product Perspective

- This defines the relationship this product has in the entire spectrum of products.
- It defines who will be responsible for the product and what business purpose it serves.
- It also defines what interfaces it may have to other systems.

II. General Description

A. Product Perspective

The VRS is a web-based system. The system interfaces with two other systems, the owner's email system, the video distributor's video system, and the browsers used by VRS customers. The system provides a secure environment for all financial transactions and for the storing and retrieving of confidential member information.

II. General Description

B. Product Functions

- This section lists the major functions of the system.
- It provides a summary of all the functions of the software. The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.
- This section should be consistent with the functional requirements defined in Section III.

II. General Description

B. Product Functions

The VRS allows customers to search the video inventory provided by this video store. To rent videos through the VRS, one must register as a member using the VRS. Upon becoming a member and logging into the VRS, the VRS provides the functionality for renting videos, modifying membership information, and paying overdue fines.

The clerks of the video store use VRS to process the return of rented videos. The owner of the video store uses VRS to add new videos into the system, remove videos from the system, and modify video information.

The VRS sends emails to members concerning video rentals. One day before a rented video is due to be returned, VRS emails the member a reminder of the due date for the video(s). For any overdue videos, VRS emails the member every 3rd day with overdue notices. At the 60-day limit for outstanding videos, VRS debits the member's credit card with the appropriate charge and notifies the member of this charge.

II. General Description

C. User Characteristics

- List the users involved with the proposed system including the general characteristics of eventual users (for example, educational background, amount of product training).
- List the responsibility of each type of user involved, if needed.

II. General Description

C. User Characteristics

The three main groups of VRS users are customers, members, and store personnel. A customer is anyone who is not a member. The customer can only search through the video inventory. The amount of product training needed for a customer is none since the level of technical expertise and educational background is unknown. The only skill needed by a customer is the ability to browse a website.

Member is someone who has registered with VRS. A member can rent videos and pay fees online. As with a customer, these activities require no product training since the level of technical expertise and educational background of a member is unknown. The only skill needed by a member is the ability to browse a website.

The store personnel are divided into two groups: the clerk-level personnel and owner-level personnel. Their educational level is unknown and both group needs little to no training.

II. General Description

D. General Constraints

- In this section, the constraints of the system are listed. They include hardware, network, system software, and software constraints. It also includes user constraints, processing constraints, timing constraints, and control limits.

II. General Description

D. General Constraints

This system provides web access for all customer and member functions. The user interface will be intuitive enough so that no training is required by customers, members, or store personnel. All online financial transactions and the storage of confidential member information will be done in a secure environment. Persistent storage for membership, rental, and video inventory information will be maintained.

II. General Description

E. Assumptions and Dependencies

- This includes assumptions made at the beginning of the development effort as well as those made during the development.
- List and describe each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but any changes to them can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change.

II. General Description

II.A Product Perspective	Paragraph form
II.B Product Functions	Paragraph form
II.C User Characteristics	Paragraph form
II.D General Constraints	Paragraph form
II.E Assumptions and Dependencies	Paragraph form

III. Functional Requirements

- Functional requirements are those business functions which are included in this software under development. It describes the features of the product and the needed behavior.
- The functional requirements are going to be written in narrative form identified with numbers. Each requirement is something that the system SHALL do. Thus, it has a common name of a shall list. You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.

IV. Functional Requirements

- Non functional requirements are properties that the system must have such as performance, reusability, usability, user friendliness, etc.

- The same format as the functional requirements is to be used for the non-functional requirements. You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.

VI. System Models

This section presents the use case diagram for the system under development. The use case diagram should be a complete version containing all the use cases needed to describe the functionality to be developed.

VII. Appendixes

Appendix A. Data dictionary

Appendix B. Raw use case point analysis

Appendix C. Screens and reports with navigation matrix.

Appendix D. Scenario analysis tables

Appendix E. Screens/reports list

Appendix F and following. Other items needed