



DIKTISAINTEK
BERDAMPAK

PRA-UTS · PENDEKATAN TERSTRUKTUR

SRS / SKPL: Software Requirements Specification

Information Systems Analysis and Design

What is an SRS (SKPL) and why does it matter?

DEFINITION

An SRS is a formal document that describes WHAT a system must do and the constraints it works under, not HOW it is built.

Known in Indonesia as SKPL (IEEE 830 / ISO 29148)

Shared agreement

1

Client, analyst, and developers work from the same reference.

Basis for design and testing

2

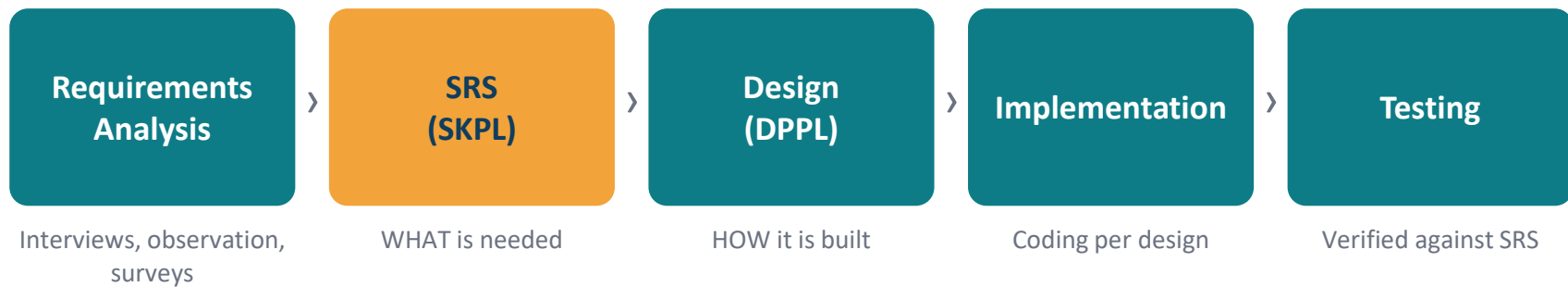
The design document and test cases are derived from it.

Controls cost of change

3

Wrong requirements cost the most when found late.

Where the SRS fits in the development cycle



The SRS bridges user needs and technical design. Testers also use it as the yardstick: the system is correct when it meets the SRS.

Structure of an SRS document

1

Introduction

Purpose, scope, definitions, references

2

Overall Description

Perspective, functions, users, constraints

3

External Interfaces

UI, hardware, software, communication

4

Functional Requirements

Requirement list, use cases or DFD

5

Non-Functional Requirements

Performance, security, availability

6

Data Requirements

ERD and data dictionary

7

Traceability Matrix

Requirement to use case to test case

Appendices: mockups, interview results, open issues

Functional vs non-functional requirements

Functional

What the system does

Examples:

- The system shall allow users to log in
- A student shall be able to borrow a book
- An admin shall be able to print reports

Non-Functional

How well the system works

Examples:

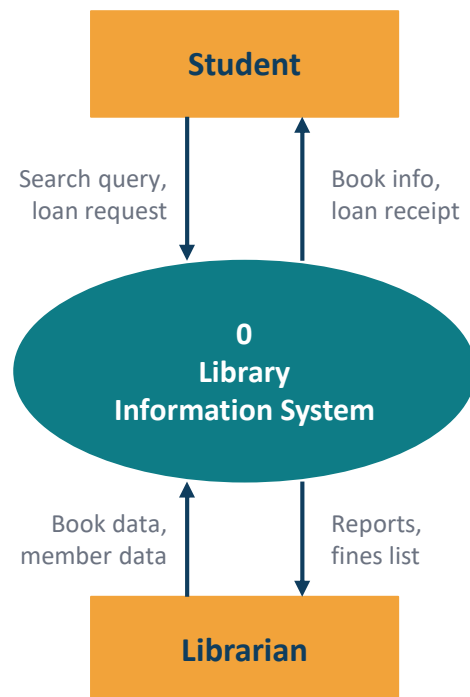
- Response time ≤ 3 seconds
- Passwords stored hashed
- Availability $\geq 99\%$ per month

Modeling approaches: Use Case vs DFD

SKPL Section	Use Case Approach	DFD Approach
2.1 Perspective	Context / architecture sketch	Context diagram (DFD Level 0)
4.2 Functional model	Use case diagram	DFD Level 1 (Level 2 if complex)
4.3 Functional detail	Use case description	Process specification (PSPEC)
4.4 Supporting models	Activity / sequence diagram	Flowchart, decision table, structured English
6 Data requirements	ERD + data dictionary	ERD + data dictionary (incl. flows and stores)

DFD rules: keep levels balanced, number processes hierarchically (1.0, 1.1), and trace each requirement to a process number.

Worked example: DFD of the library system



LEVEL 1 PROCESSES

- 1.0 Manage Members
- 2.0 Process Loans
- 3.0 Search Catalog
- 4.0 Generate Reports

DATA STORES

D1 Books
D2 Members
D3 Loans

Qualities of a good requirement

- Clear and unambiguous
- Measurable (numbers or criteria)
- Testable
- Consistent, no conflicts
- Unique ID (traceable)

BAD

"The system must be fast and easy to use."

GOOD

"SRS-NF-001: The system shall return book search results in ≤ 2 seconds for 500 concurrent users."

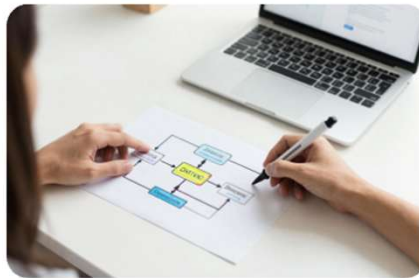
Workshop: Build Your SRS/SKPL

Develop an SRS Based on the Case Study

Use the analysis results from the previous meeting to develop a complete SRS draft according to the required format.



Requirements & Business Rules



DFD & Process Specs



Data Dictionary &
Traceability