Software Engineering

Prof Ravi Prakash Gorthi

Jan-April, 2011

Topics

1. Introduction
2. Software Engineering Models
3. BPM, BRMS, SOA
4. Software Requirements Engineering
5. Software Analysis Models
6. Software Project Management
7. Software Design Concepts, Principles and Models
8. Software Coding Practices
9. Software Testing Techniques
10. Software Quality Assurance
11. Emerging Trends in Software Engineering

Text & Reference Books

Text Books:

- Software Engineering: A Practitioner's Approach by Roger Pressman, McGraw-Hill, 6th edition
- Software Engineering by Ian Sommerville, Pearson Education LPE, 8th or 9th edition
- An Integrated Approach to Software Engineering, Narosa Publishing, 2nd edition

Acknowledgement: Contents of some of the slides are taken from Ian Sommerville's course ppt

Reference Books:

4. Software Engineering by S. L. Pfleeger, MacMillan Publishing
5. Research papers in the area of SE – to be distributed by Prof Gorthi

Topics

- ✓ Introduction
- ✓ Software Engineering Models
- ➢ BPM, BRMS, SOA

4. Software Requirements Engineering
5. Software Analysis Models
6. Software Project Management
7. Software Design Concepts, Principles and Models
8. Software Coding Practices
9. Software Testing Techniques
10. Software Quality Assurance
11. Emerging Trends in Software Engineering

BPM, BRMS and SOA

BPM (Business Process Modeling): Any business is run as a set of business processes; it is very important to identify and model the business process activities and routes within these business processes;

Examples:

Consider a retail bank; some of the main business processes of a bank are:

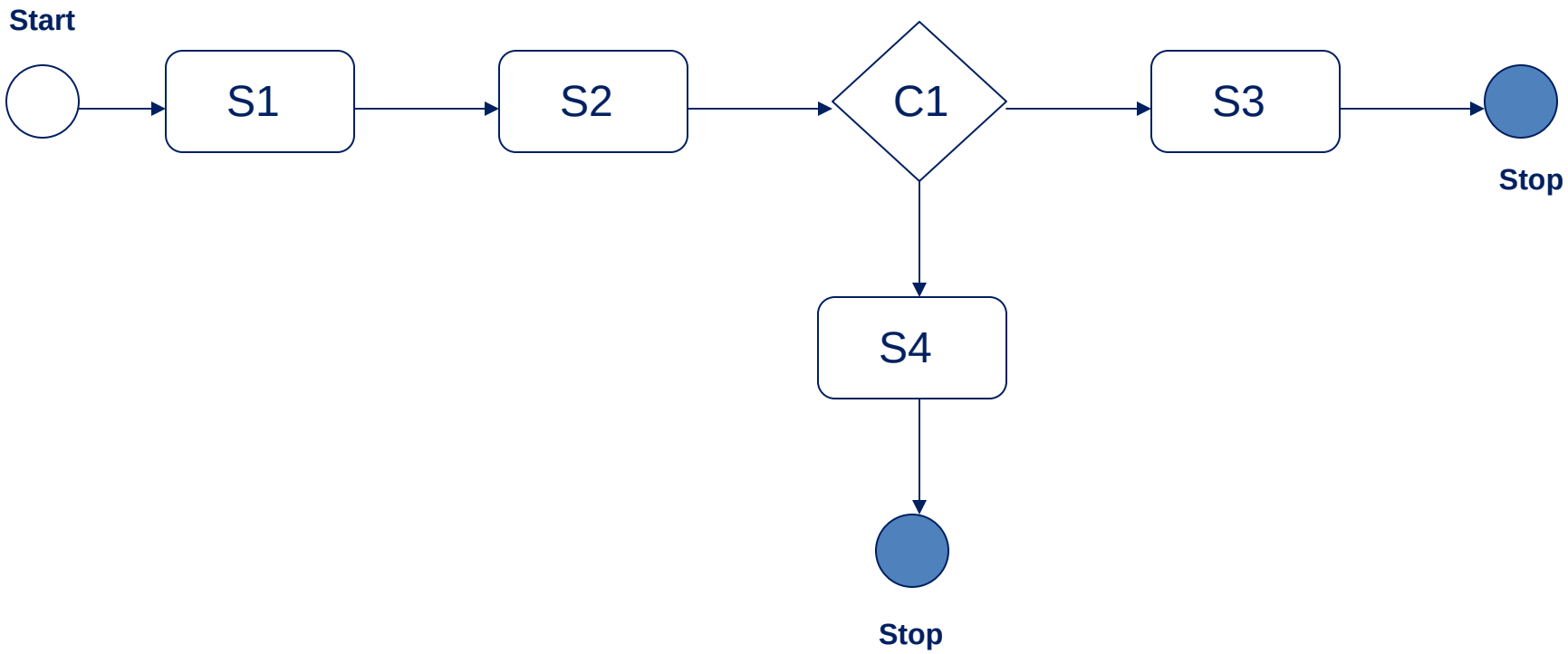
1. Operating savings accounts
2. Operating current accounts
3. Offering and managing loans
4. Offering and managing wealth management schemes

Examine the business process of Loan Management:

BPM, BRMS and SOA

BPM (Business Process Modeling): Examples:

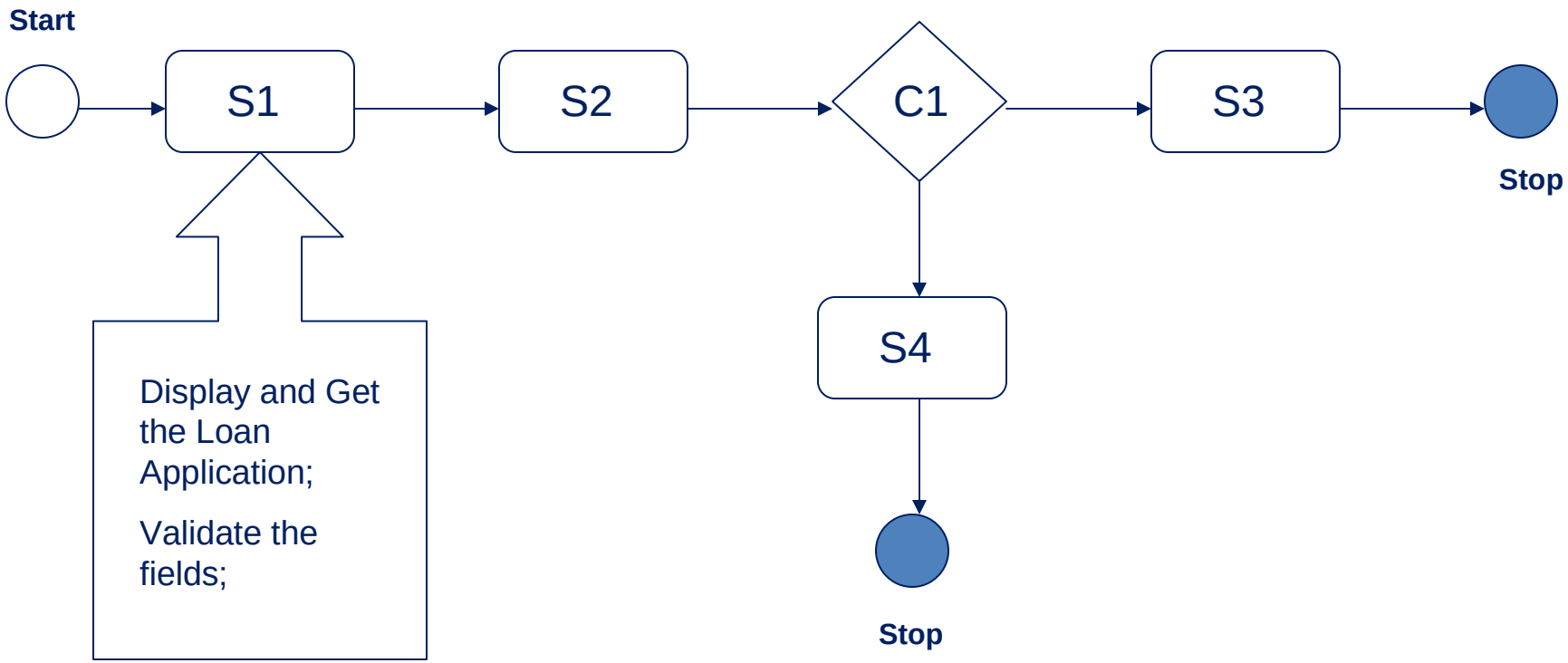
Consider a retail bank; examine the business process activities of Loan Management:



BPM, BRMS and SOA

BPM (Business Process Modeling): Examples:

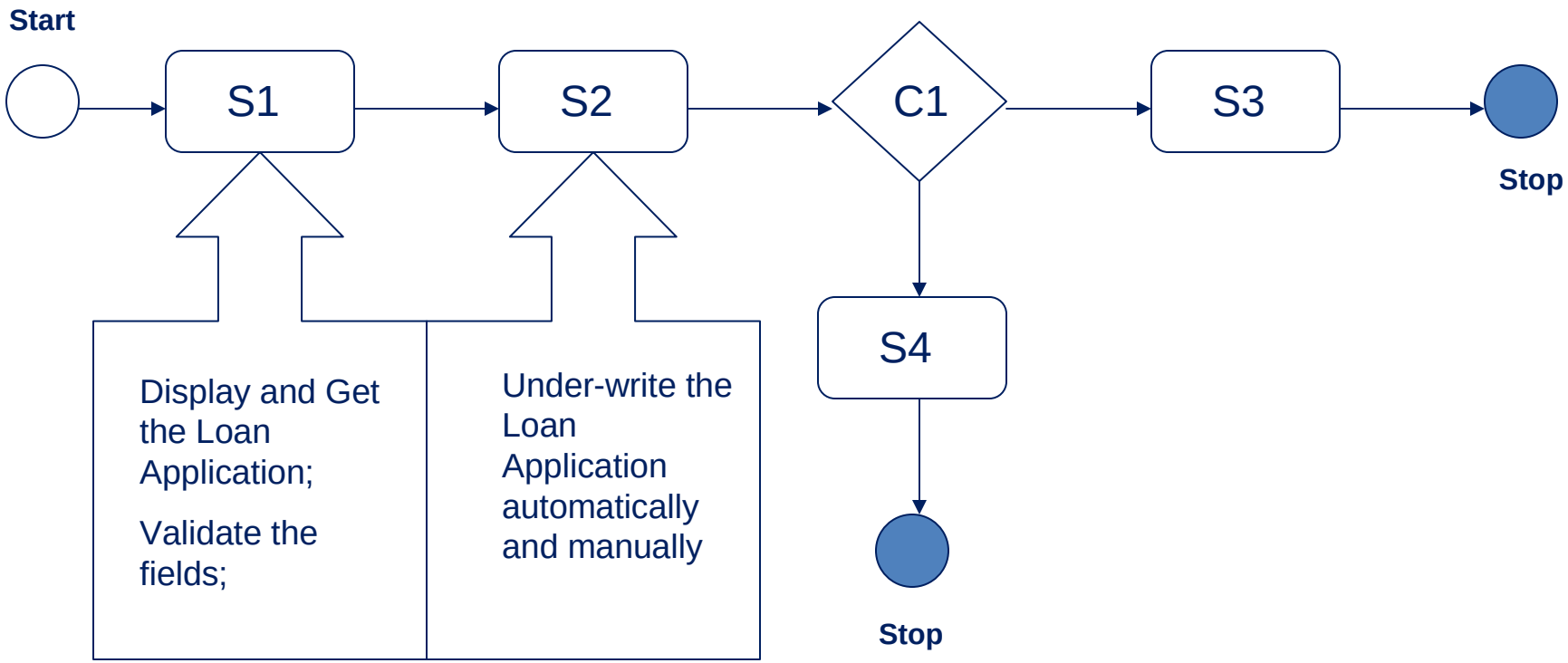
Consider a retail bank; examine the business process of Loan Management:



BPM, BRMS and SOA

BPM (Business Process Modeling): Examples:

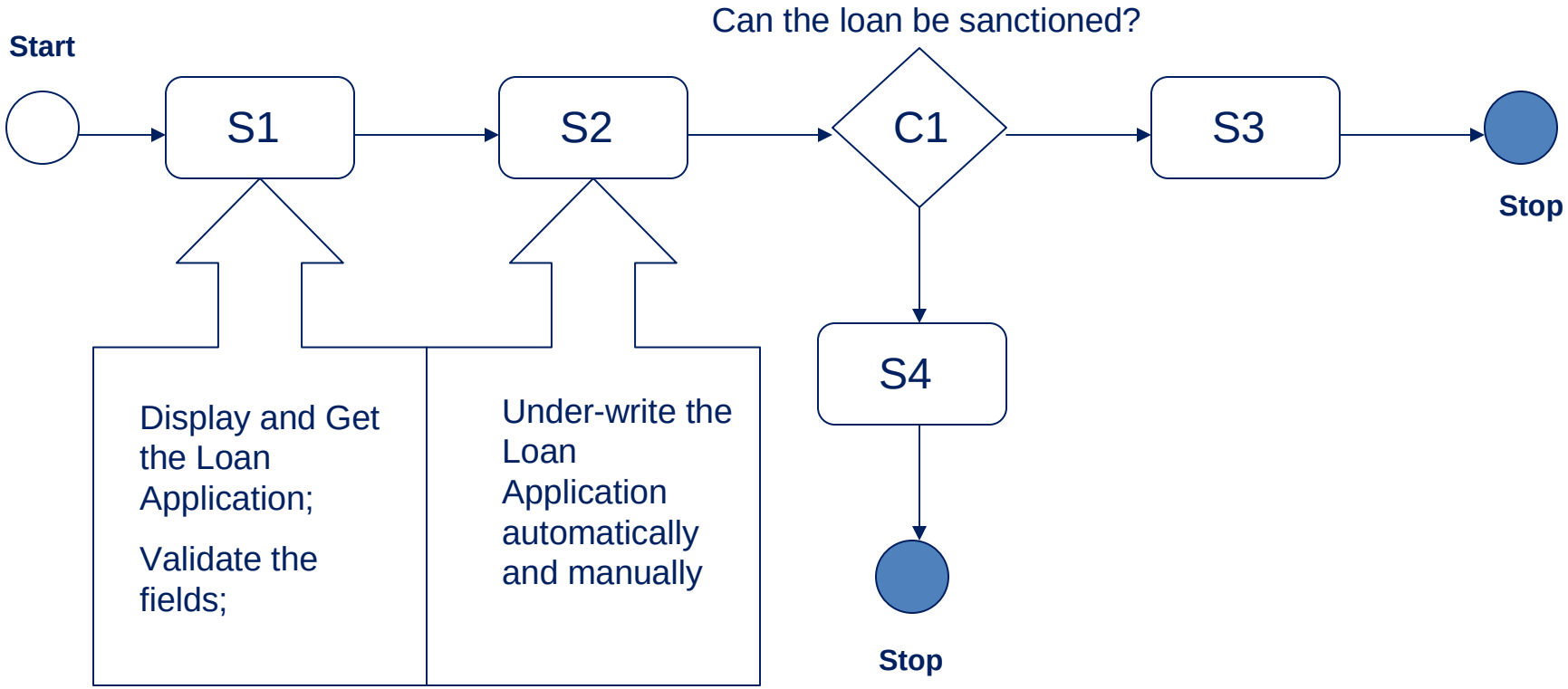
Consider a retail bank; examine the business process of Loan Management:



BPM, BRMS and SOA

BPM (Business Process Modeling): Examples:

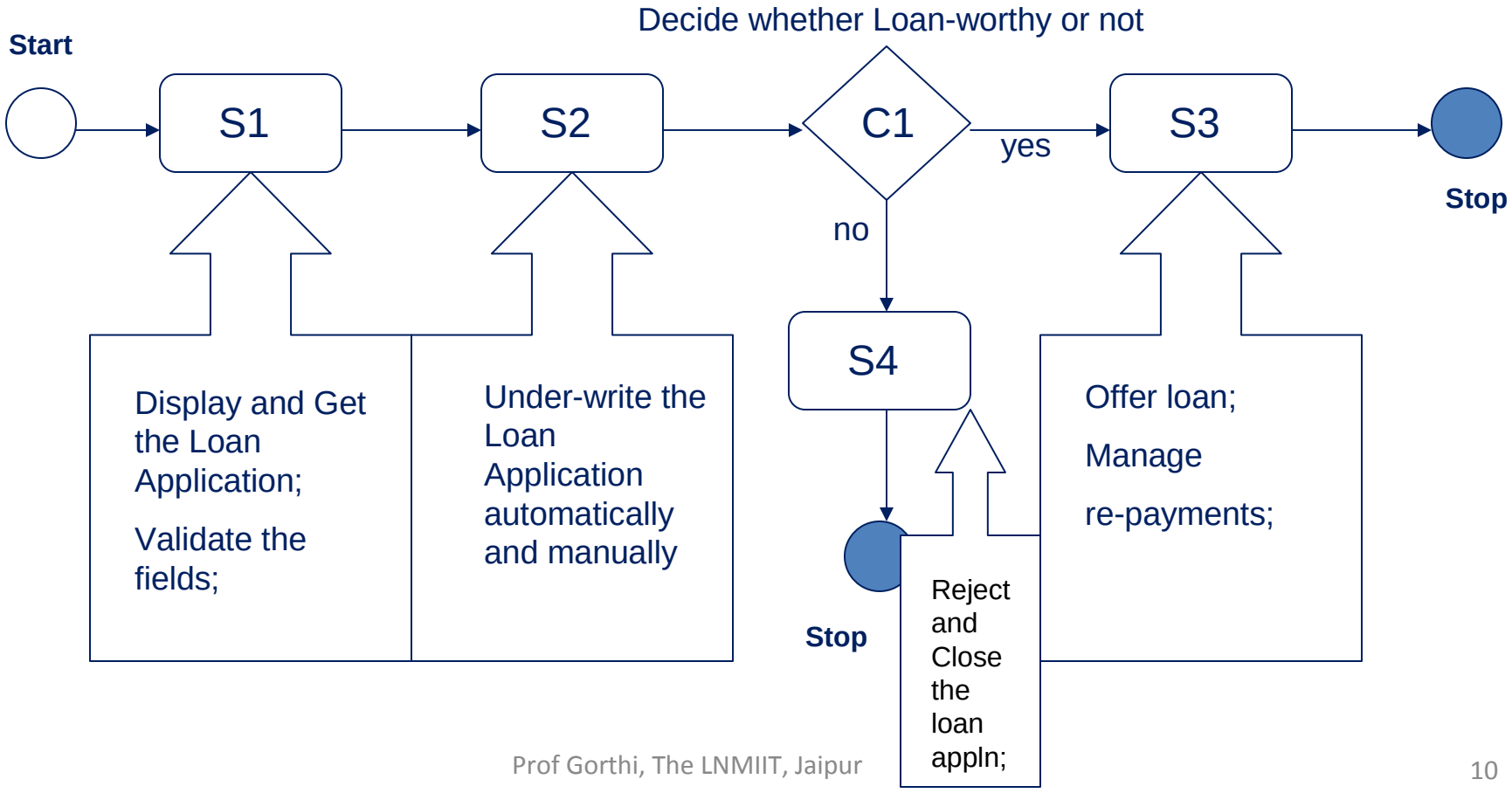
Consider a retail bank; examine the business process of Loan Management:



BPM, BRMS and SOA

BPM (Business Process Modeling): Examples:

Consider a retail bank; examine the business process of Loan Management:



BPM, BRMS and SOA

BPM

1. Business Process Model Design (Notation)
2. Business Process Model Execution Languages and Engine
   a) BPEL4WS (Business Process Execution Language for Web Services)
3. IBM, Microsoft, Oracle, etc offer BPM tools

Benefits of BPM approach: by explicitly modeling, analyzing, coding and executing business processes

1. One can easily adapt to market changes and bring-in process changes
2. One can identify bottle-necks and eliminate / reduce inefficiencies

BPM, BRMS and SOA

BRMS (Business Rules Modeling Systems): Any business process has a set of business rules; these business rules make the operation of business consistent and transparent;

Consider a retail bank; examine the business process of Loan Management; there are:

1. Loan eligibility rules
2. Loan under-writing rules
3. Loan approval rules
4. Loan rate rules
5. Loan re-payment default rules
6. Premature Loan closing rules

BPM, BRMS and SOA

BRMS (Business Rules Modeling Systems):

Examples of Business Rules

Consider a retail bank; examine the business process of Loan Management; there are:

1. Loan approval rules
   a) If (loan-type is home-loan) AND (applicant-age >= 60 years) then (loan-application status = rejected)
2. Loan rate rules
   a) If (loan-duration is <= 2 years) then (loan-rate = 5%)

Propositional / Predicate Logic based Rule Languages (OPS5) and Rete based Rule Engines are popular;

ILOG, Fair-Isaac, Yasu Technologies, Pega, etc are some of the leaders in the BRMS space;

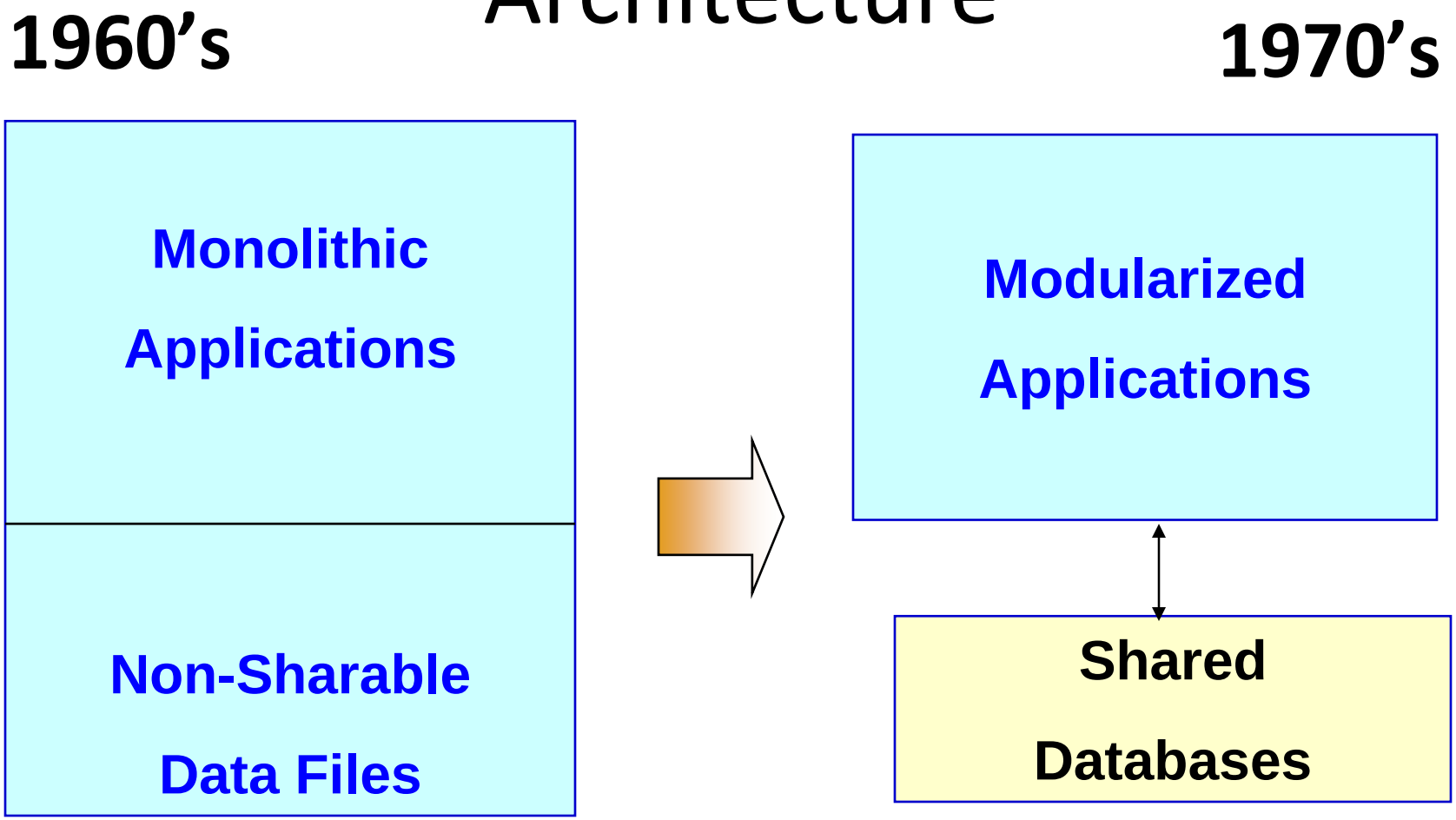
BPM, BRMS and SOA

SOA (Service Oriented Architecture): Any business process consists of / offers / utilizes a set of business services;

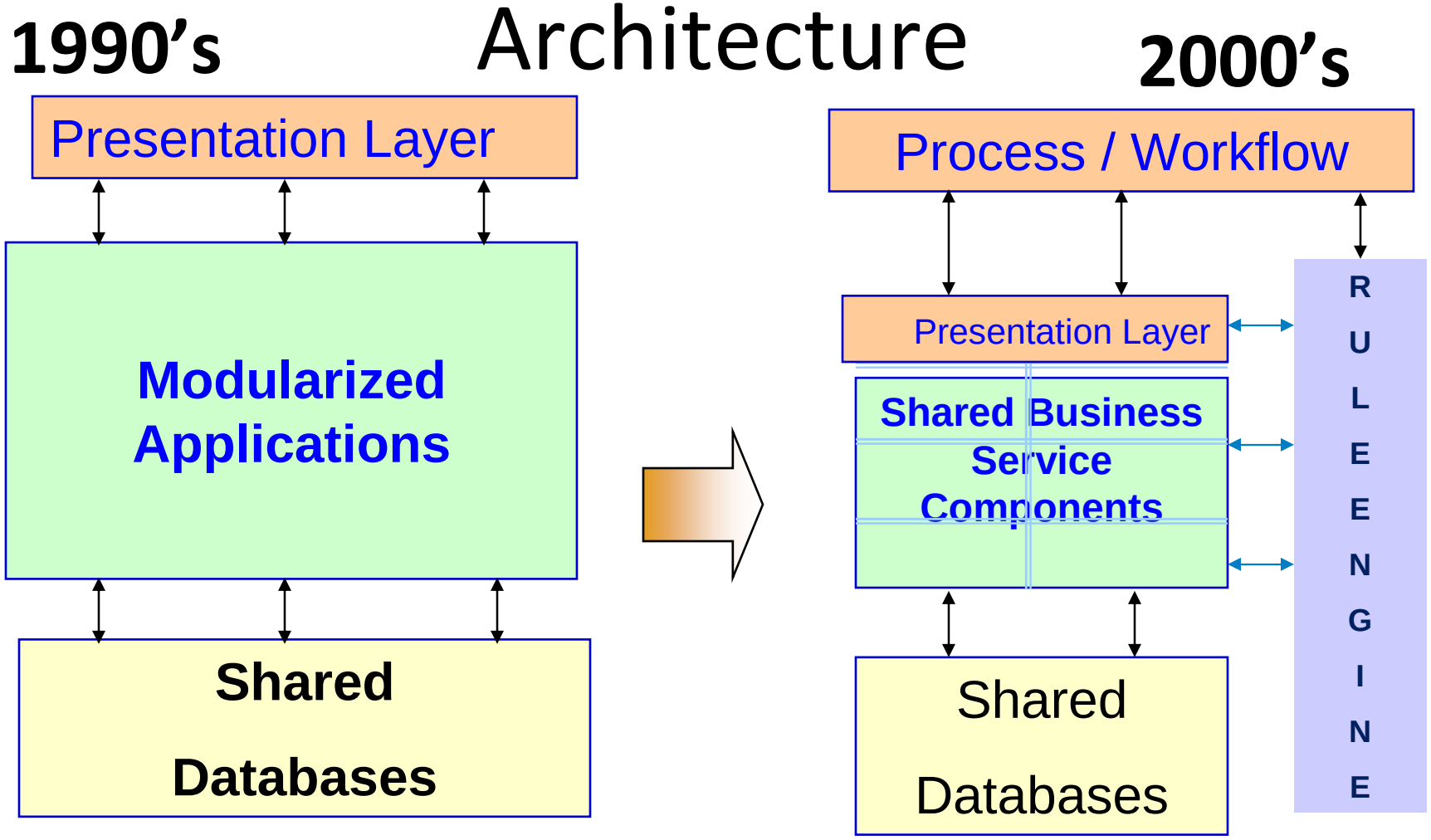
Consider a retail bank; examine the business process of Loan Management; there are business services such as:

1. Loan application validation services
   a) Display Loan Application
   b) Display Loan Eligibility Rules
   c) Validate the Loan Application (and highlight errors, if any)
2. Loan under-writing service
   a) Apply loan under-writing rules automatically and Generate *FICO* Score
   b) Facilitate manual modifications to FICO score with reasons by experts
3. Loan closure service
4. Loan sanctioning service
5. Loan re-payment management service

Evolution of Enterprise Architecture



Evolution of Enterprise Architecture



BP & BR were always present in Business Automation!

BPM, BRMS and SOA

Summary:

Identify and model,

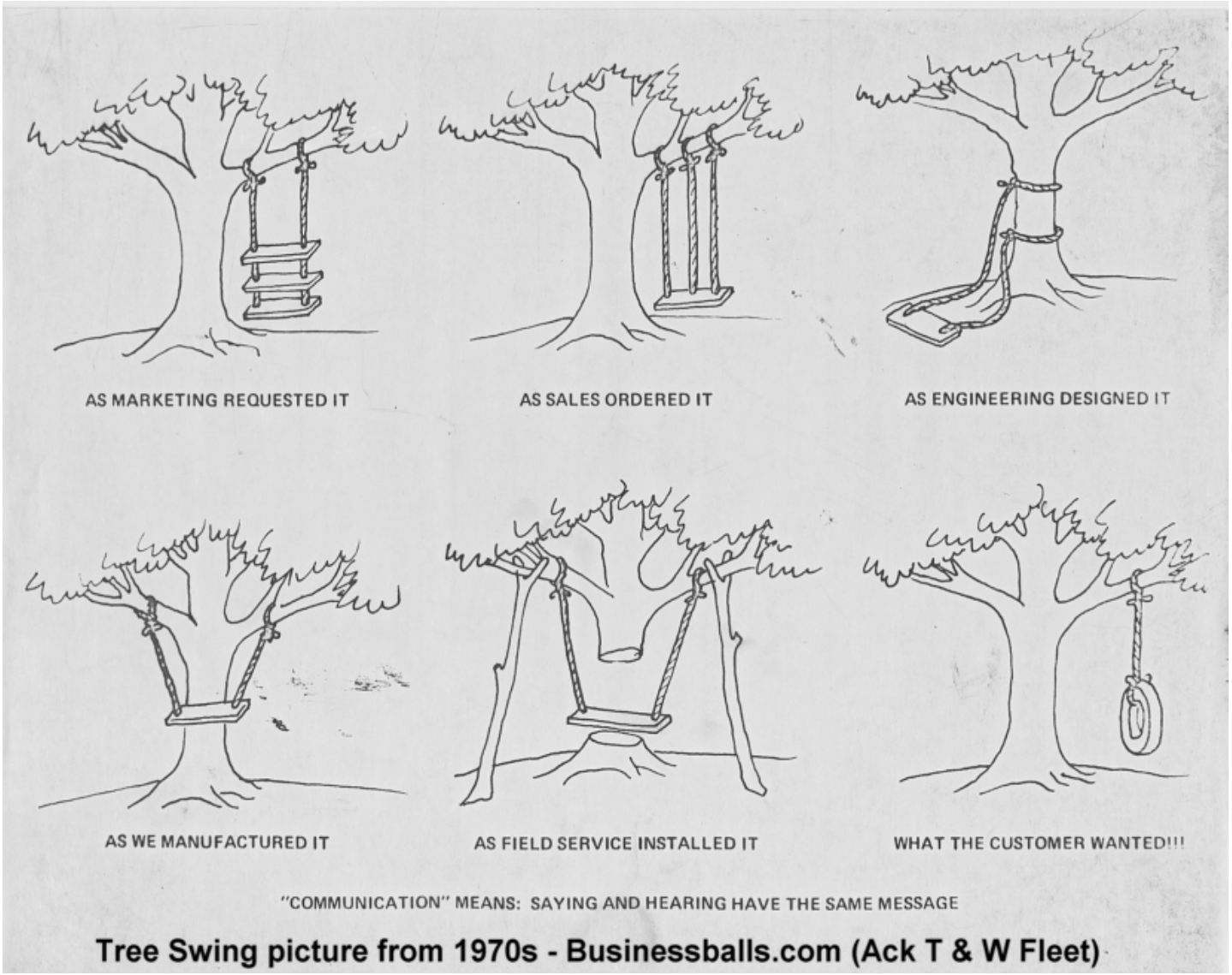
- The Business Processes and the Activities and Routings among the Processes;
- The sets of Business Rules present in each Business Operation;
- The Service Components of each Business Operation

Topics

- ✓ Introduction
- ✓ Software Engineering Models
- ✓ BPM, BRMS, SOA
- ➢ Software Requirements Engineering

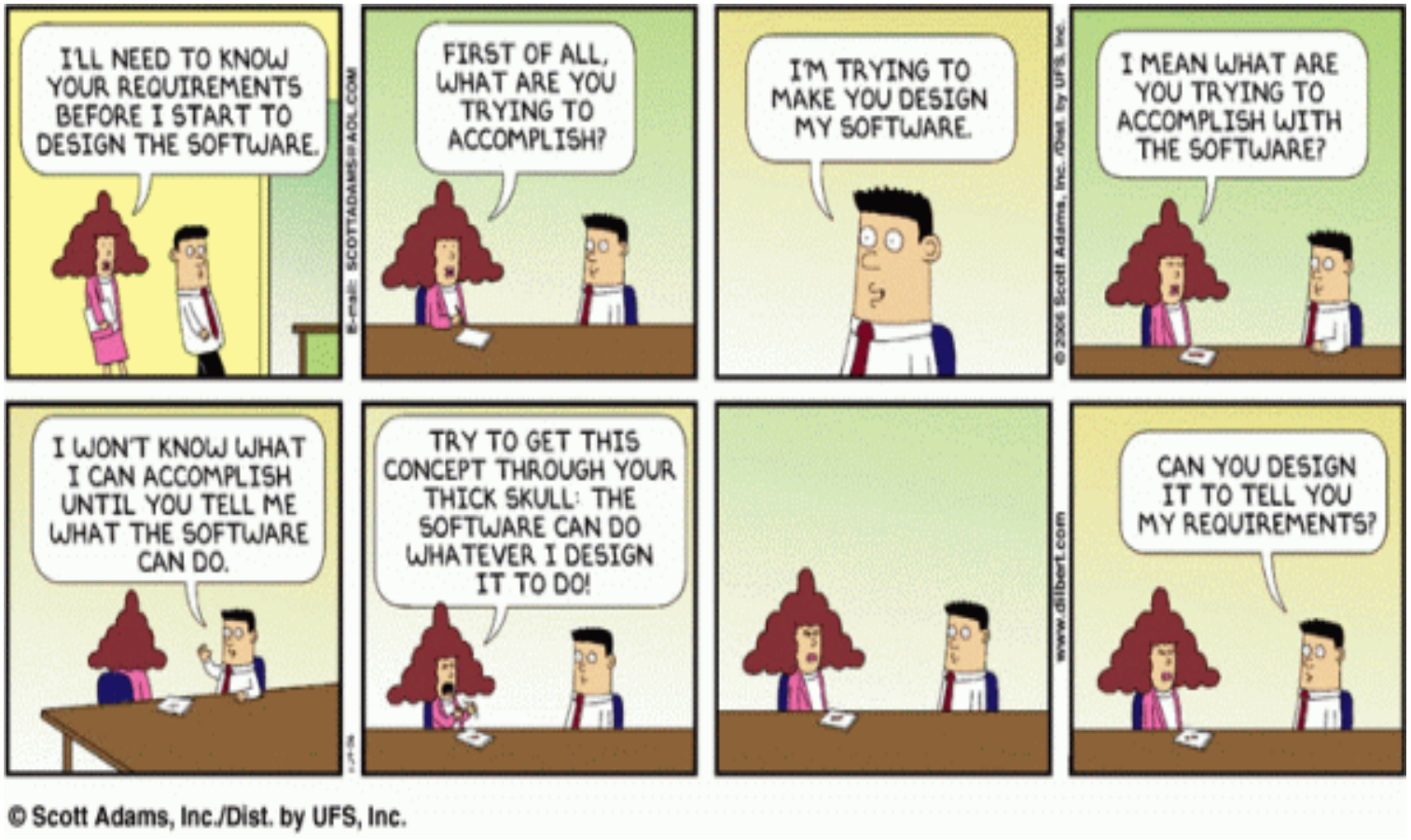
5. Software Analysis Models
6. Software Project Management
7. Software Design Concepts, Principles and Models
8. Software Coding Practices
9. Software Testing Techniques
10. Software Quality Assurance
11. Emerging Trends in Software Engineering

Software Requirements Engineering



Tree Swing picture from 1970s - Businessballs.com (Ack T & W Fleet)

Software Requirements Engineering



Software Requirements Engineering

- Functional and non-functional requirements
- The software requirements document
- Requirements specification
- Requirements validation

Functional and non-functional requirements

- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
  - May state what the system should not do.

- Non-functional requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
  - Often apply to the system as a whole rather than individual features or services.

Types of requirement

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.
- System requirements
  - A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.
- User & System requirements should specify *WHAT* the 'to-be-developed' system is expected to do but *not HOW* to do (how to do is part of the design phase)

Functional specifications

Business Use Case Approach:

- Identify the different users of the business operation (example: Loan Management)
  - Loan Management Department Staff
  - Bank's Customers
- Identify the functions performed by each user
  - By Loan Management Staff:
    - Define Loan Types, Loan Process for each Loan Type
  - By Bank's Customers:
    - Apply for Loans and Find the Status of a Loan Application

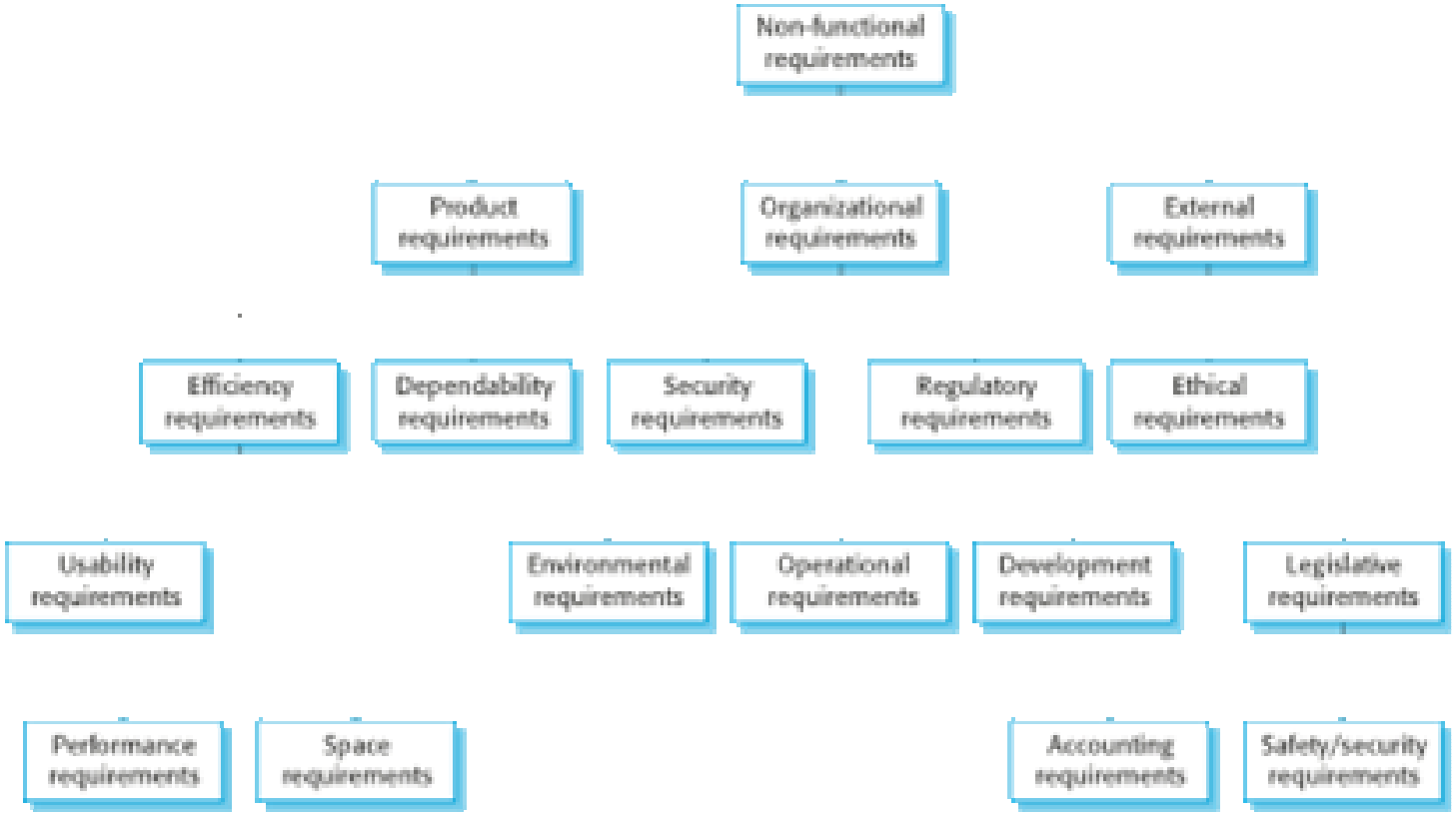
Functional specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Details about the information needed for the computation and other entities used.
- Description of the action to be taken.
- Pre and post conditions (if appropriate).
- The side effects (if any) of the function.

Requirements document

- The process of writing down the user and system requirements in a requirements document.
- User requirements have to be understandable by end-users and customers who do not have a technical background.
- System requirements are more detailed requirements and may include more technical information.
- The requirements may be part of a contract for the system development
  - It is therefore important that these are as complete as possible.

Types of nonfunctional requirement



Characteristics of a Good SRS

1. Unambiguous
2. Correct & Complete
3. Consistent
4. Verifiable
5. Ranked for importance
6. Traceable

Characteristics of a Good SRS

1. Unambiguous
   - Ambiguous requirement statements are those that are prone to be interpreted differently by the business users and the software developers
- Example: From the Loan Management SRS:
  - "Creation of a new Loan Type OR Modification or deletion to an existing Loan Type needs to be approved by the Loan Management Head"
  - In the ABC Bank, there is a Loan Management Head at each branch office, regional office, country level office and at global head-quarters; who is authorized to approve the above requirement?

Characteristics of a Good SRS

2. Correct & Complete

- Incorrect requirement statements are those that do not state either the inputs required or the processing steps or the output produced by an activity correctly
- Incomplete requirement statements are those that have not stated the processing steps or the output produced for some combinations of inputs

- Example: From the Loan Management SRS:
  - The Loan Management Staff modifies an existing Loan Type details; the Loan Management Head approves the changes;
  - The Loan Management Staff deletes an existing Loan Type; the Loan Management Head approves the deletion;

Characteristics of a Good SRS

3. Consistent

- Two requirement statements are said to be inconsistent if they contradict each other.

- Example: From the Loan Management SRS:
  - A requirement in one section reads as “The Loan Management Staff deletes an existing Loan Type; the Loan Management Head approves the deletion;”
  - A requirement in another section reads as “An existing Loan Type can never be deleted; it can only be de-activated with effect from a date and time;”

Characteristics of a Good SRS

4. Verifiable

- A requirement statement is such that there is no known technical mechanism to check whether it is offered by the software

- Example: From the Loan Management SRS:
  - Customer of a Loan Application should provide his / her residential address proof by uploading a recent electricity bill; the software should automatically verify whether the electricity bill is genuine or not and if genuine proceed to the next step else inform the user that his / her residential address proof is not genuine;

Characteristics of a Good SRS

5. Ranked for Importance

- Every requirement should be tagged with its priority number; the priority number is a two-digit number and a value of ’99’ indicates the highest and ’00’ the lowest priority;

- Example: From the Loan Management SRS:

- “Once the Loan Management Staff submits the form corresponding to creating a new Loan Type, the software should validate all the fields; if there are any errors, the software should display the error messages along with the new Loan Type details to the Loan Management Staff; if there are no errors, the submitted form is sent to the Loan Management Head of the branch office for approval; (priority 99 . . . If this requirement is not implemented, other requirements cannot be tested)

Characteristics of a Good SRS

6. Traceable

- For each requirement, there is a evidence as to the source of that requirement; sources of requirements can be either (a) an interview with the business users or (b) a reference document provided by the business users;

- Example: From the Loan Management SRS:
  - The Loan Management Staff deletes an existing Loan Type; the Loan Management Head approves the deletion; (source: the Team Lead of the Software Development team)

Guidelines for writing requirements

- Use IEEE Standard SRS Template for all requirements.
- Use language in a consistent way. Use shall for mandatory requirements, should for desirable requirements.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon.
- Include an explanation (rationale) of why a requirement is necessary.

Requirements validation techniques

- ✧ Requirements reviews
  - Systematic manual analysis of the requirements.
- ✧ Prototyping
  - Using an executable model of the system to check requirements. Covered in Chapter 2.
- ✧ Test-case generation
  - Developing tests for requirements to check testability.

Requirements reviews

- Regular reviews should be held while the requirements definition is being formulated.
- Both client and contractor staff should be involved in reviews.
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.