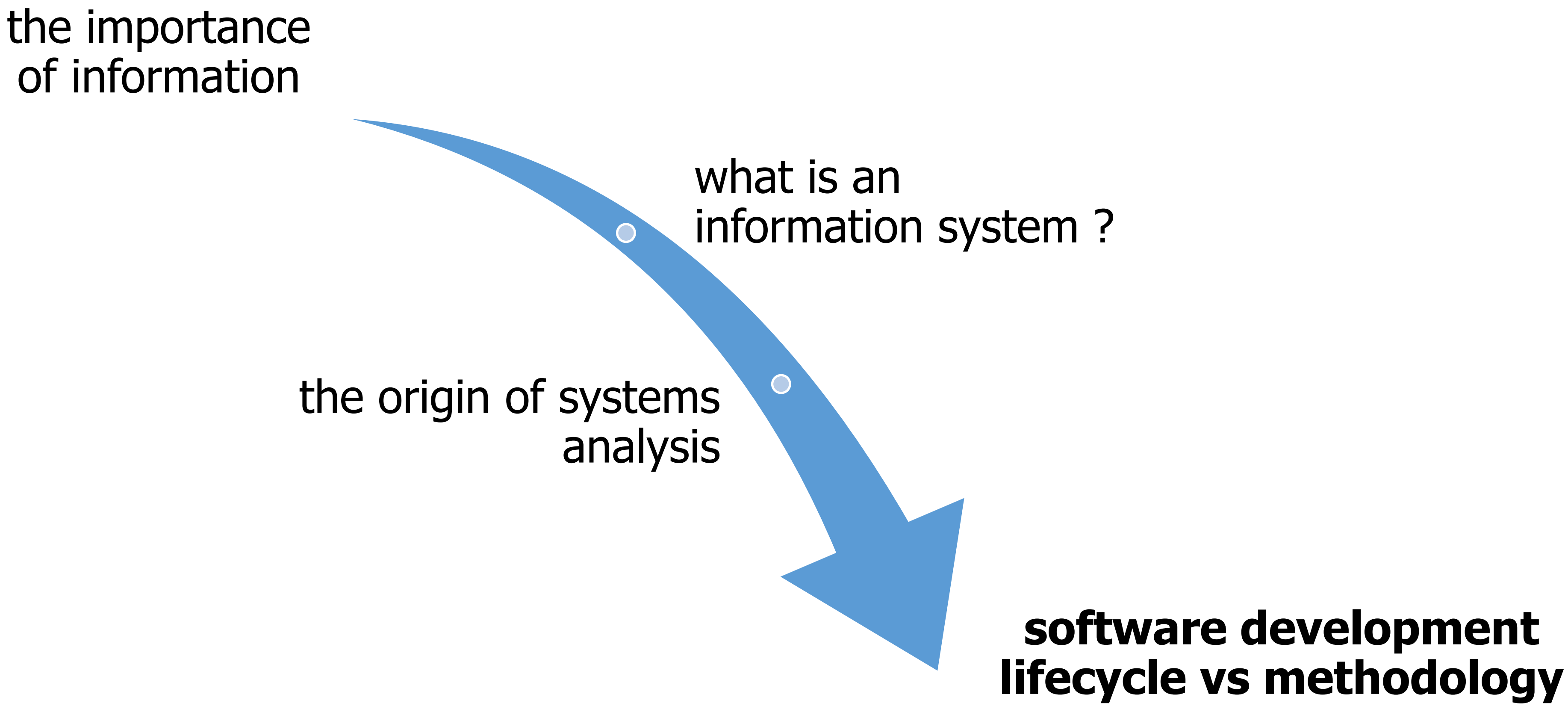
What is Systems analysis ?



the importance of information

what is an information system ?

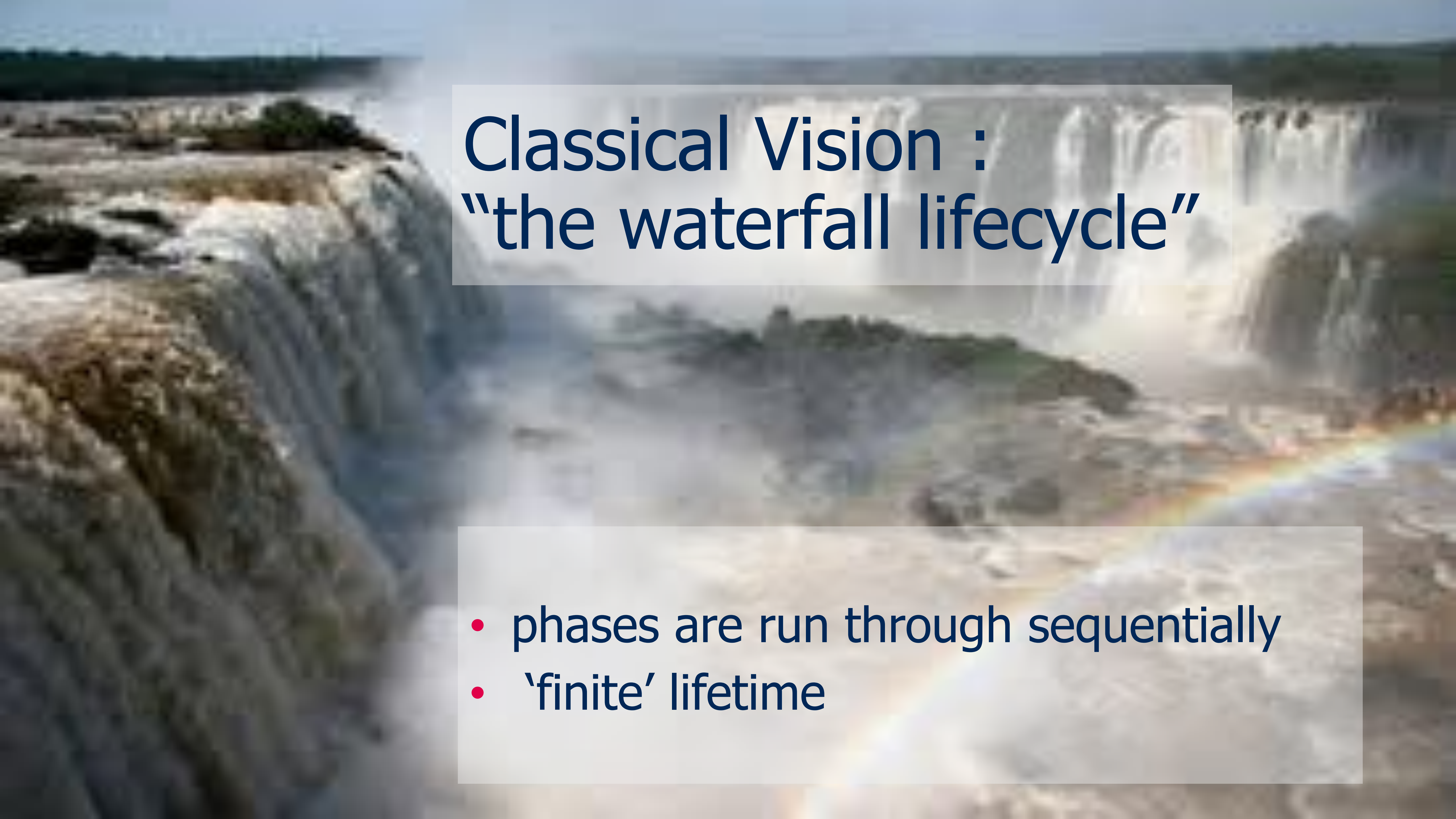
the origin of systems analysis

software development lifecycle vs methodology

Software Develoment Lifecycle

Software Development Lifecycle

phases



Classical Vision : “the waterfall lifecycle”

- phases are run through sequentially
- ‘finite’ lifetime

Classical Vision : “the waterfall lifecycle”

Request from business professional

Definition study

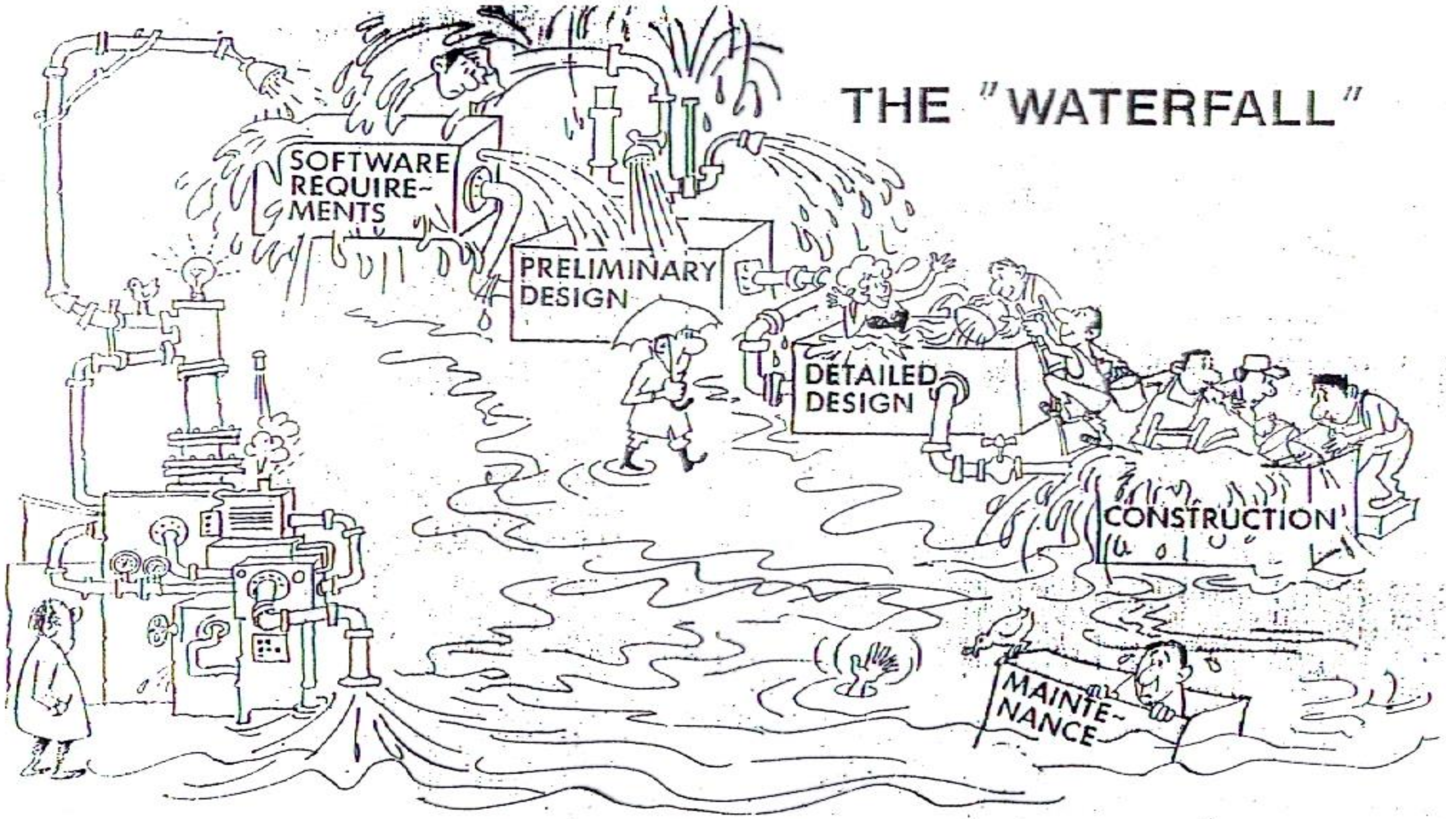
Logical design

Technical design

Realisation + test

Implementation

Waterfall Lifecycle

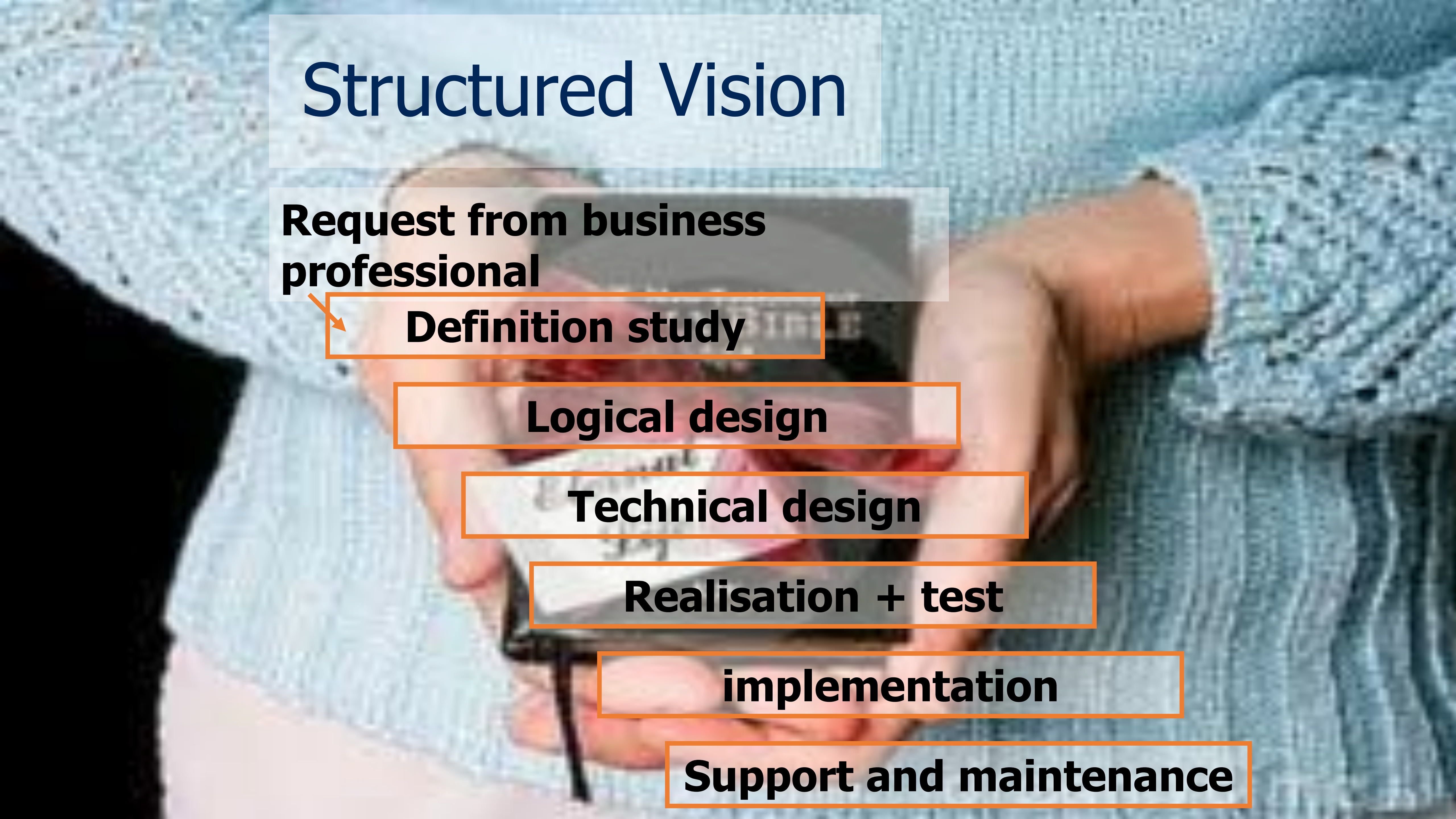


Christian Vision (structured vision)

‘immortality’

→ ‘eternal life’

Structured Vision



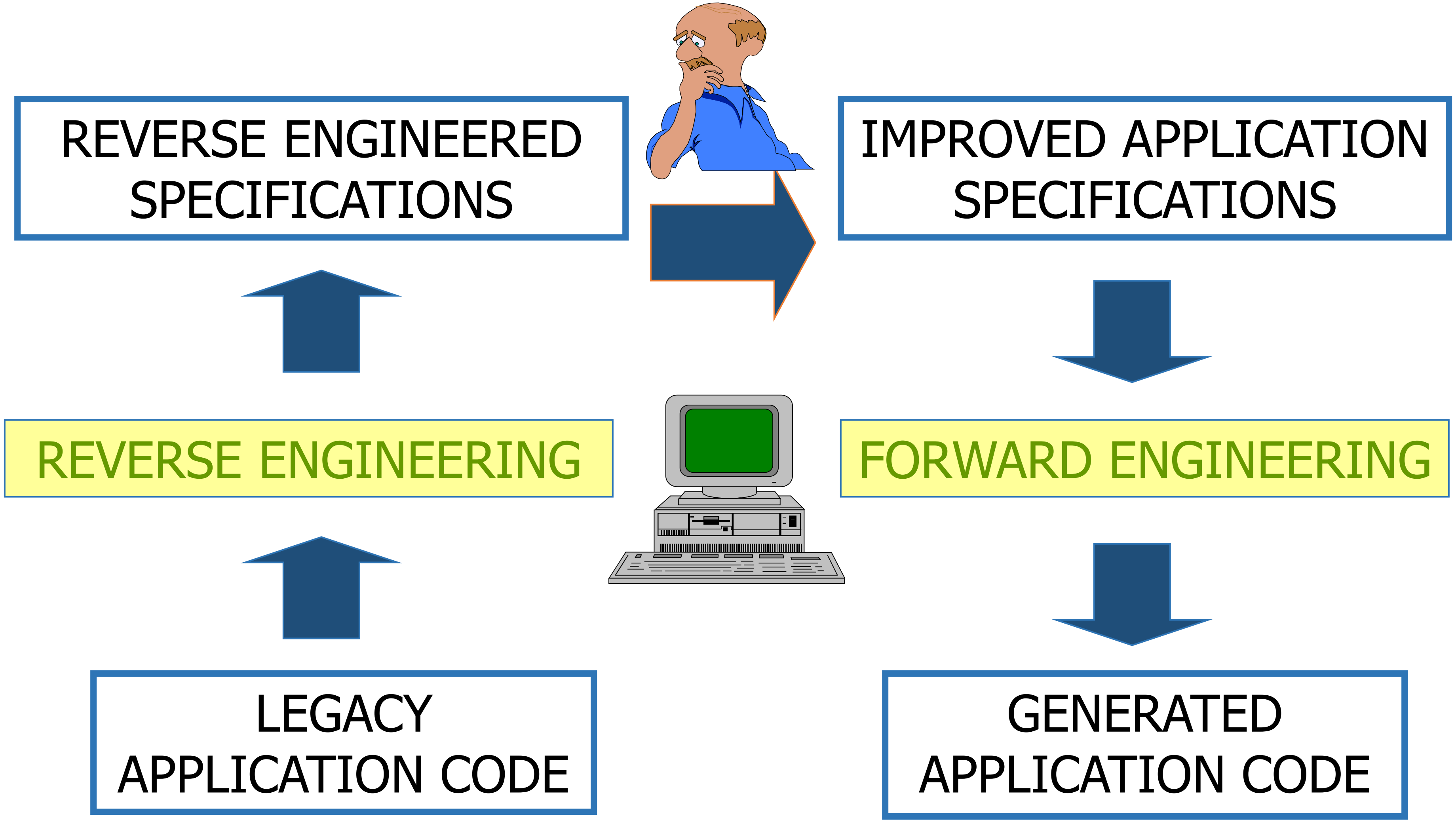
Request from business professional

- **Definition study**
- **Logical design**
- **Technical design**
- **Realisation + test**
- **implementation**
- **Support and maintenance**

Modernised Vision

- iterative process
- strategic information planning
- reverse engineering

Re-Engineering



REVERSE ENGINEERED SPECIFICATIONS

IMPROVED APPLICATION SPECIFICATIONS

REVERSE ENGINEERING

FORWARD ENGINEERING

LEGACY APPLICATION CODE

GENERATED APPLICATION CODE

Oriental Vision
(object-oriented vision)

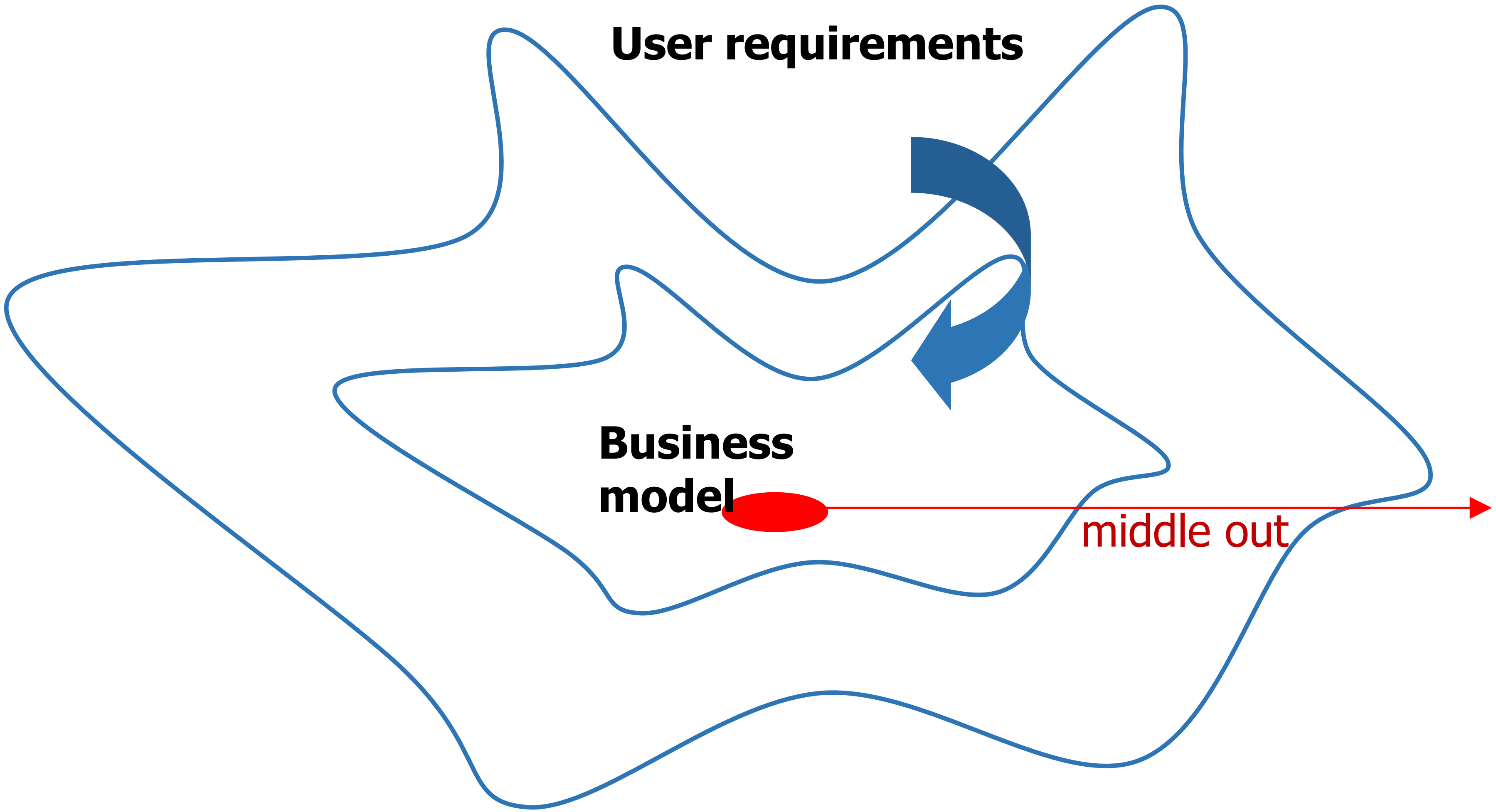
Business model: stable base

↔ User requirements

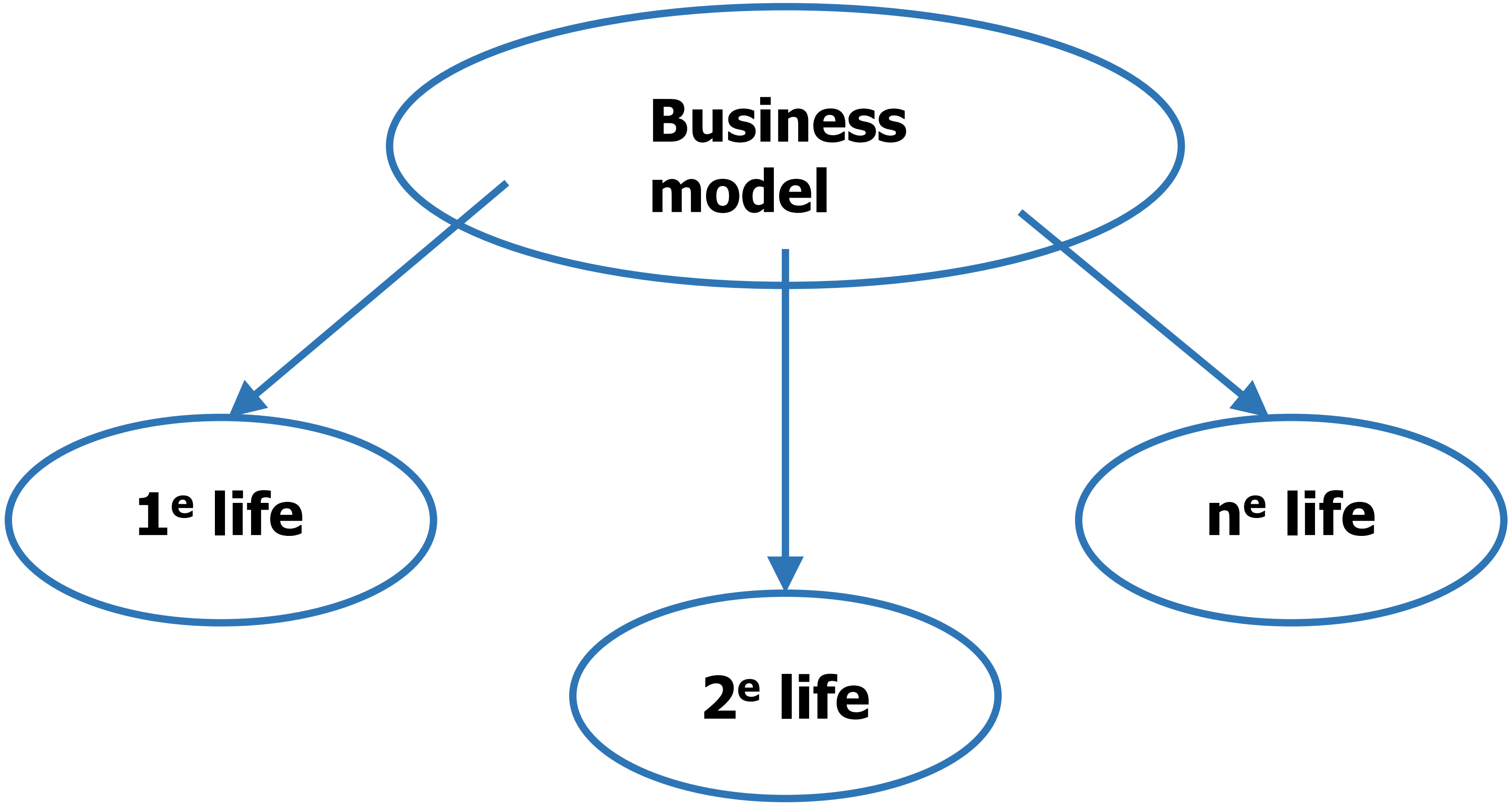
Middle-out approach

↔ Top down approach

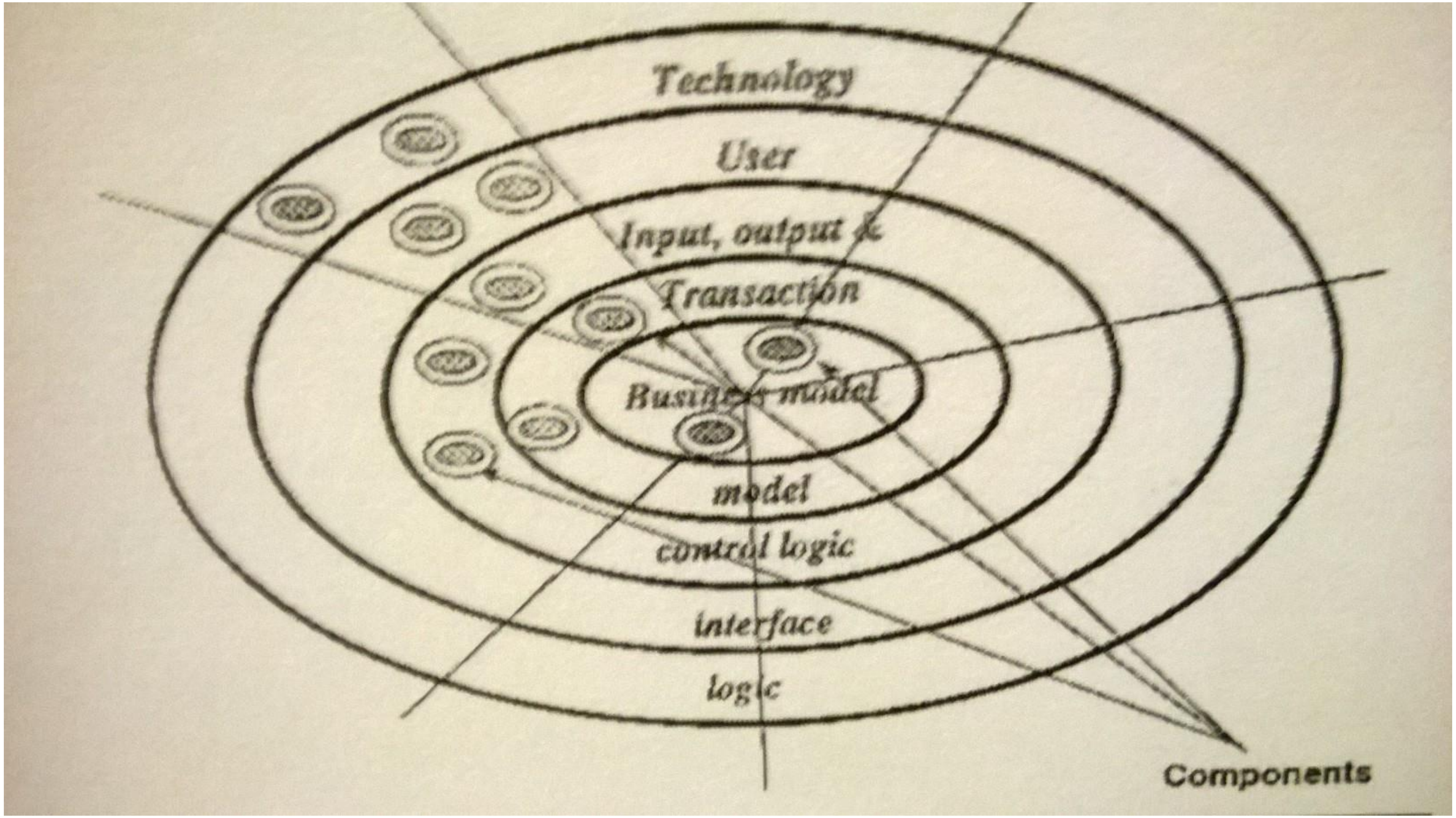
Object Oriented Vision



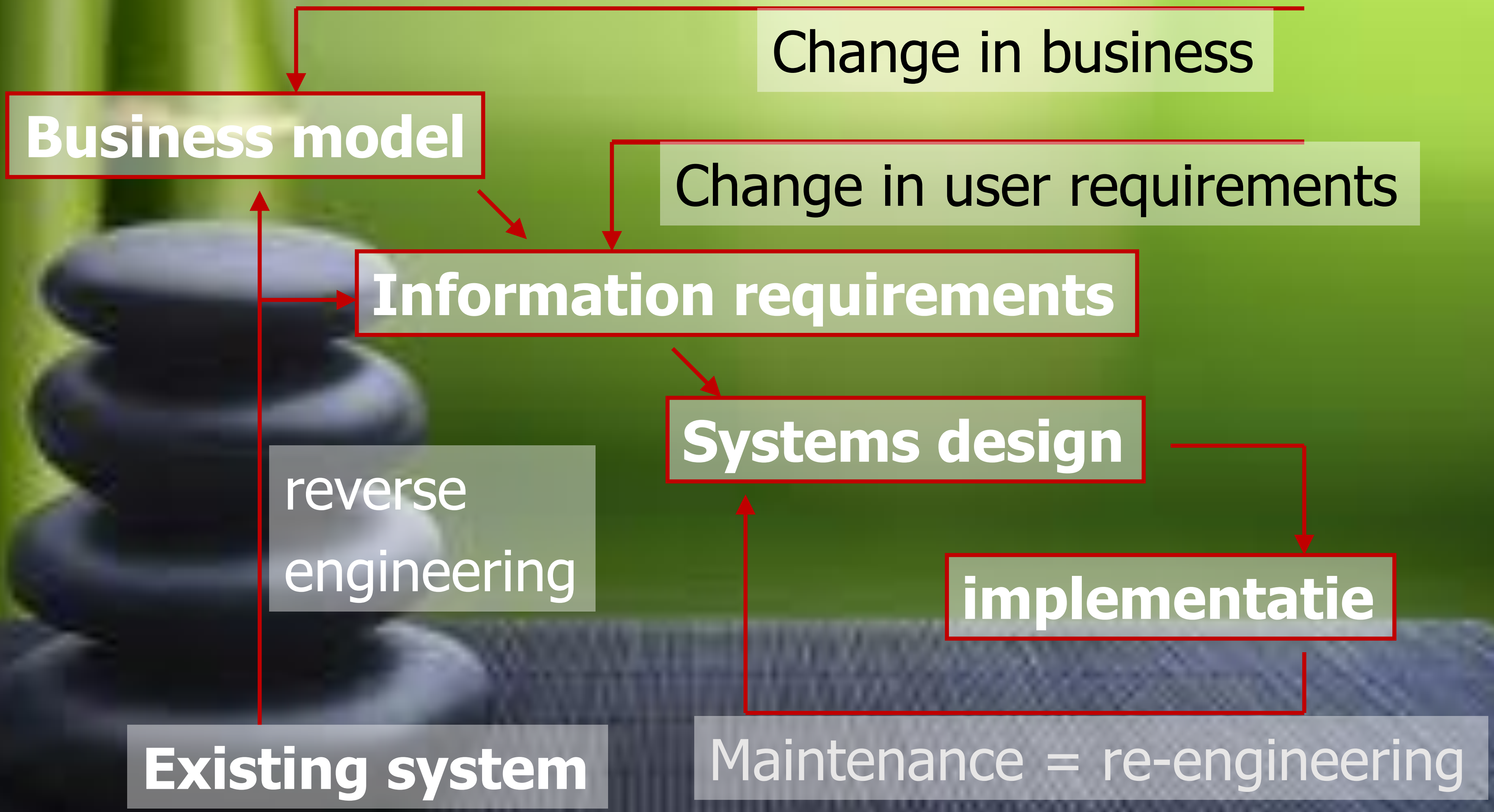
Object Oriented Vision



The Multiple-Layer Model



Model-Driven Lifecycle



Change in business

Business model

Change in user requirements

Information requirements

Systems design

reverse engineering

implementatie

Existing system

Maintenance = re-engineering

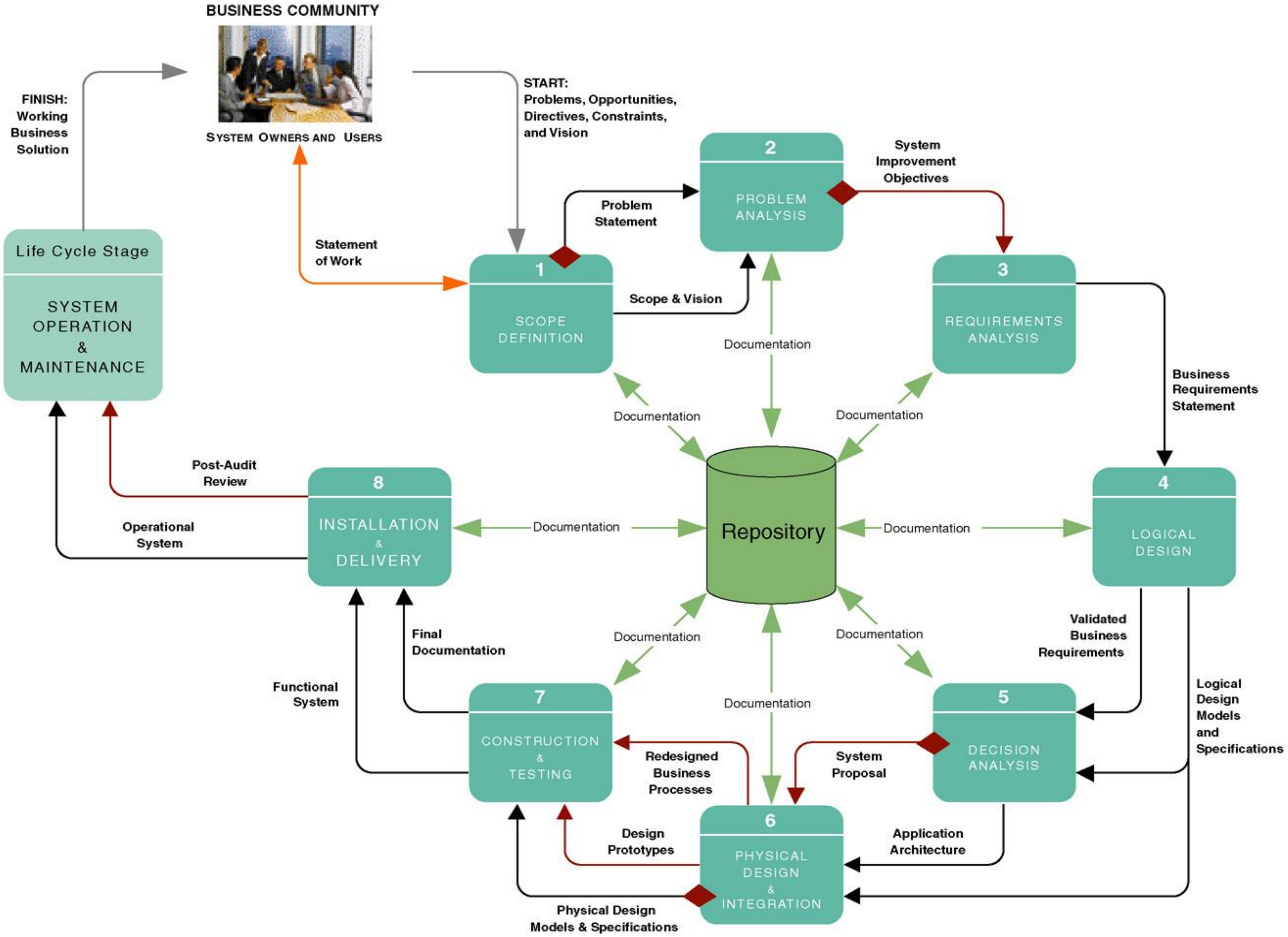
Cross Life-Cycle Activities

Cross lifecycle activity

any activity that overlaps many or all phases of the systems development process

- Fact-finding
- Documentation and presentation
  - Documentation
  - Presentation
  - Repository
- Feasibility analysis
- Process and project management

System Development Documentation, Repository, and Presentations



Model

abstraction

example: map

Methods en Techniques

method

what en when?

eg. Create a domain model

technique

how?

eg. Create an UML class diagram

Methodology = methods + techniques

Life Cycle vs Methodology

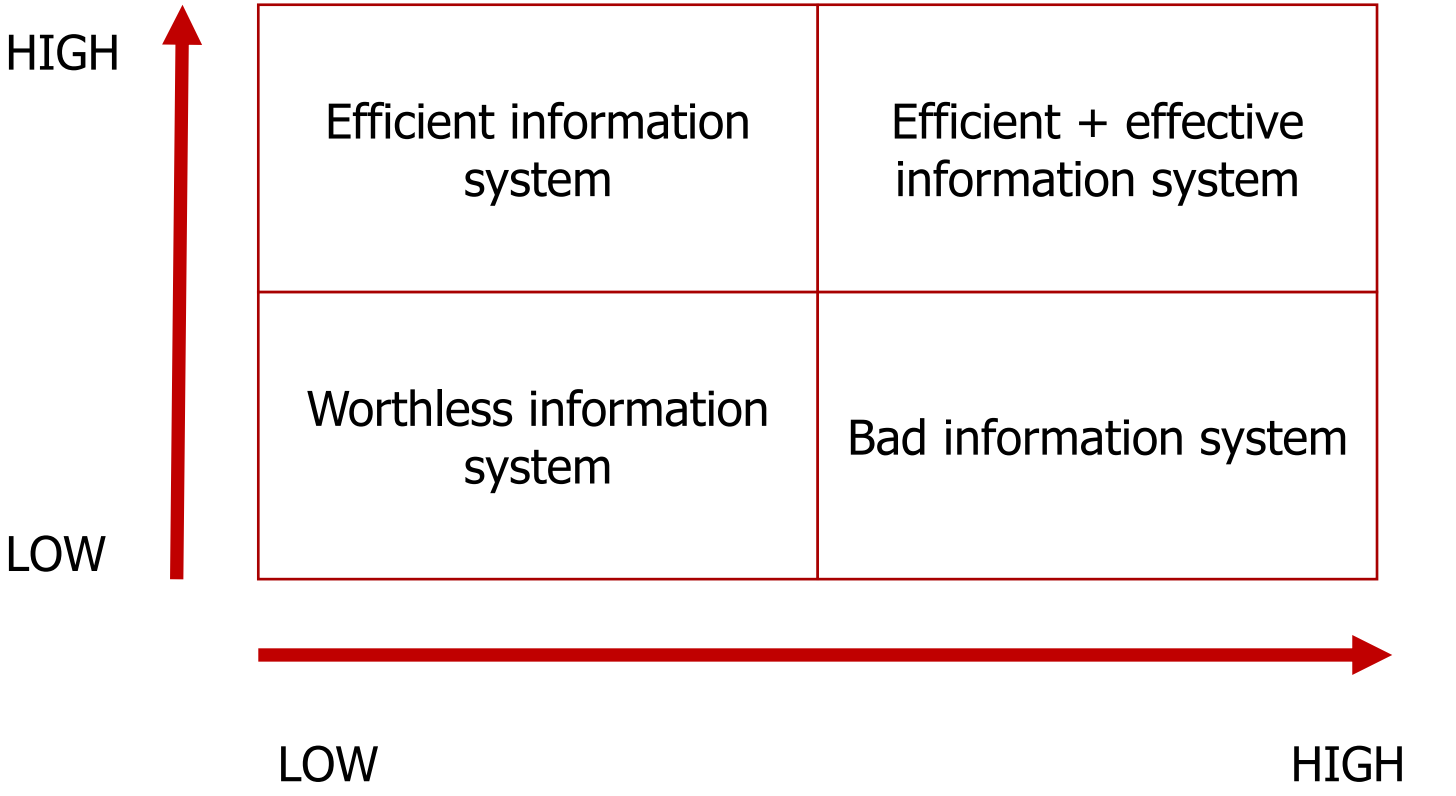
- **System life cycle** – the factoring of the lifetime of an information system into stages

- **System development methodology** – a standardized development process that defines a set of activities, methods, best practices, deliverables, and automated tools that system developers and project managers are to use to develop and continuously improve information systems and software.

Examples of methodologies

- Rapid Application Development (RAD)
- Rational Unified Process (RUP)
- Scrum / agile software development
- eXtreme programming (XP)
- …

Quality of the Developer



	LOW	HIGH
HIGH	Efficient information system	Efficient + effective information system
LOW	Worthless information system	Bad information system

Quality of the Methodology

Computer Aided Systems engineering (CASE)

the use of automated software tools that support the drawing and analysis of system models and associated specifications

Some CASE tools also provide prototyping and code generation capabilities.

Computer Aided Systems engineering (CASE)

CASE repository

- a system developers’ database where developers can store system models, detailed descriptions and specifications, and other products of system development

Synonyms include dictionary and encyclopedia.

Computer Aided Systems engineering (CASE)

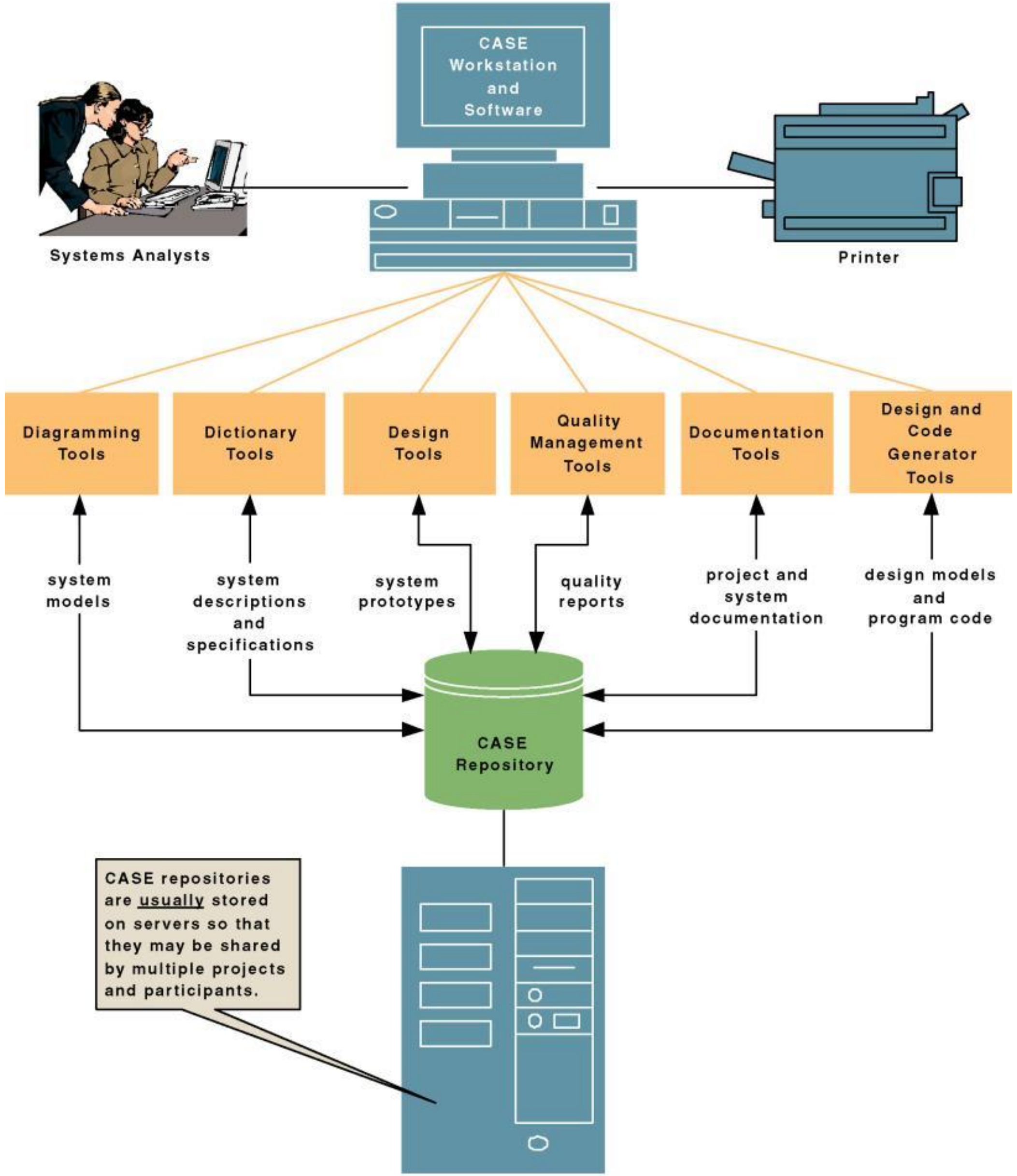
Forward engineering

- a CASE tool capability that can generate initial software or database code directly from system.

Reverse engineering

- a CASE tool capability that can generate initial system models from software or database code.

Architecture of CASE-tools



CASE repositories are <u>usually</u> stored on servers so that they may be shared by multiple projects and participants.

Application Development Environments (ADE)

an integrated software development tool that provides all the facilities necessary to develop new application software with maximum speed and quality

A common synonym is integrated development environment (IDE)

Application Development Environments (ADE)

ADE facilities may include:

- Programming languages or interpreters
- Interface construction tools
- Middleware
- Testing tools
- Version control tools
- Help authoring tools
- Repository links

Manage process and projects

Process management

an ongoing activity that documents, manages, oversees the use of, and improves an organization’s chosen methodology (the “process”) for system development. Process management is concerned with phases, activities, deliverables, and quality standards should be consistently applied to all projects.

Manage process and projects

Project management

the process of scoping, planning, staffing, organizing, directing, and controlling a project to develop an information system at a minimum cost, within a specified time frame, and with acceptable quality.

Where do system development projects come from?

- **Problem** – an undesirable situation that prevents the organization from fully achieving its purpose, goals, and/or objectives.
- **Opportunity** – a chance to improve the organization even in the absence of an identified problem.
- **Directive** - a new requirement that is imposed by management, government, or some external influence.

Decision analysis

Candidate solutions evaluated in terms of:

- Technical feasibility
- Operational feasibility
- Economic feasibility
- Schedule feasibility
- Risk feasibility

Physical design and Integration

Physical design

the translation of business user requirements into a system model that depicts a technical implementation of the users' business requirements. Common synonyms include technical design or implementation model.

Physical design and Integration

Two extreme philosophies of physical design

- Design by specification
- Design by prototyping

Rapid Application Development

Rapid Application Development (RAD)

a system development strategy that emphasizes speed of development through extensive user involvement in the rapid, iterative, and incremental construction of series of functioning prototypes of a system that eventually evolves into the final system.

Rapid Application Development (RAD)

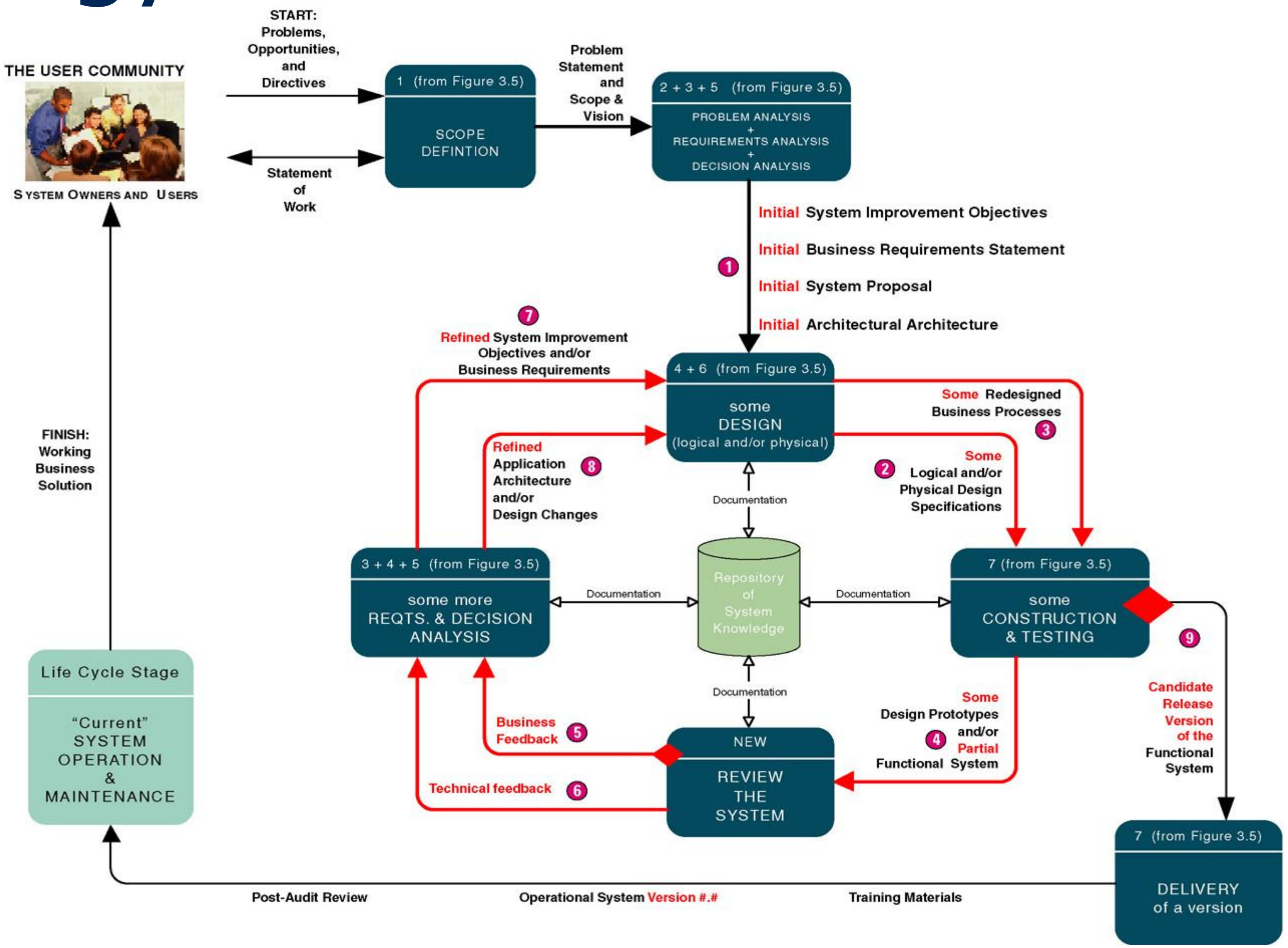
Prototype

a small-scale, representative, or working model of the users’ requirements or a proposed design for an information system.

Time box

the imposition of a nonextendable period of time, usually 60-90 days, by which the first (or next) version of a system must be delivered into operation.

Rapid Application Development Strategy



THE USER COMMUNITY

SYSTEM OWNERS AND USERS

START:
Problems,
Opportunities,
and
Directives

Statement
of
Work

1 (from Figure 3.5)
SCOPE
DEFINTION

Problem
Statement
and
Scope &
Vision

2 + 3 + 5 (from Figure 3.5)
PROBLEM ANALYSIS
+
REQUIREMENTS ANALYSIS
+
DECISION ANALYSIS

1
Initial System Improvement Objectives
Initial Business Requirements Statement
Initial System Proposal
Initial Architectural Architecture

7
Refined System Improvement
Objectives and/or
Business Requirements

4 + 6 (from Figure 3.5)
some
DESIGN
(logical and/or physical)

Some Redesigned
Business Processes
3

Refined
Application
Architecture
and/or
Design Changes
8

2 Some
Logical and/or
Physical Design
Specifications

Documentation

FINISH:
Working
Business
Solution

3 + 4 + 5 (from Figure 3.5)
some more
REQTS. & DECISION
ANALYSIS

Documentation

Repository
of
System
Knowledge

Documentation

7 (from Figure 3.5)
some
CONSTRUCTION
& TESTING

9

Documentation

Life Cycle Stage
"Current"
SYSTEM
OPERATION
&
MAINTENANCE

Business
Feedback
5

Technical feedback
6

NEW
REVIEW
THE
SYSTEM

Some
Design Prototypes
and/or
4 Partial
Functional System

Candidate
Release
Version
of the
Functional
System

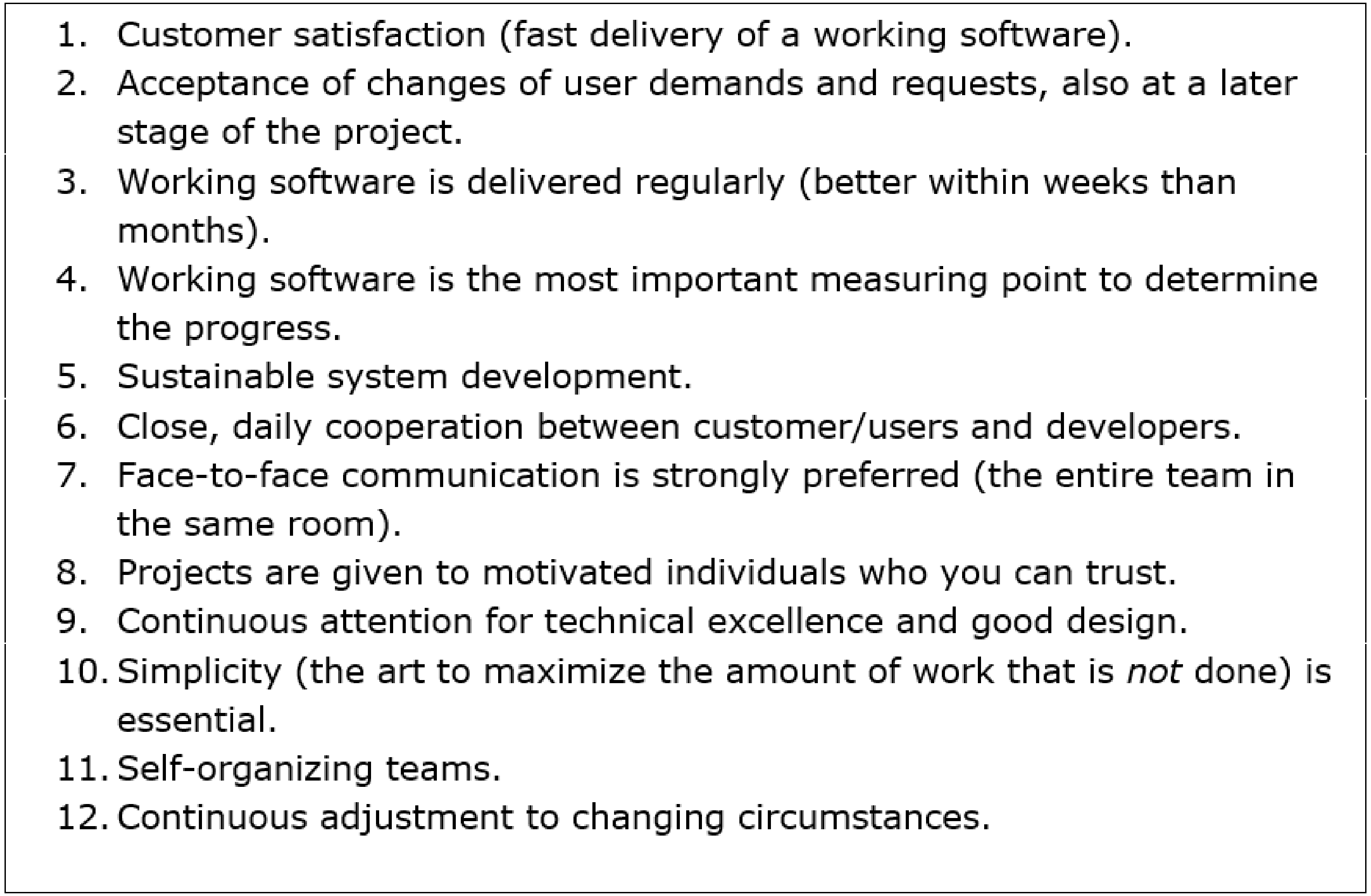
7 (from Figure 3.5)
DELIVERY
of a version

Post-Audit Review

Operational System Version #.#

Training Materials

Agile Manifesto



1. Customer satisfaction (fast delivery of a working software).
2. Acceptance of changes of user demands and requests, also at a later stage of the project.
3. Working software is delivered regularly (better within weeks than months).
4. Working software is the most important measuring point to determine the progress.
5. Sustainable system development.
6. Close, daily cooperation between customer/users and developers.
7. Face-to-face communication is strongly preferred (the entire team in the same room).
8. Projects are given to motivated individuals who you can trust.
9. Continuous attention for technical excellence and good design.
10. Simplicity (the art to maximize the amount of work that is *not* done) is essential.
11. Self-organizing teams.
12. Continuous adjustment to changing circumstances.