Software Re-Engineering

Lecture: 09

Dr. Imran Ali
School of Computing- Software Engineering
FAST-National University of Computer & Emerging Sciences, Karachi

Sequence [Todays Agenda]

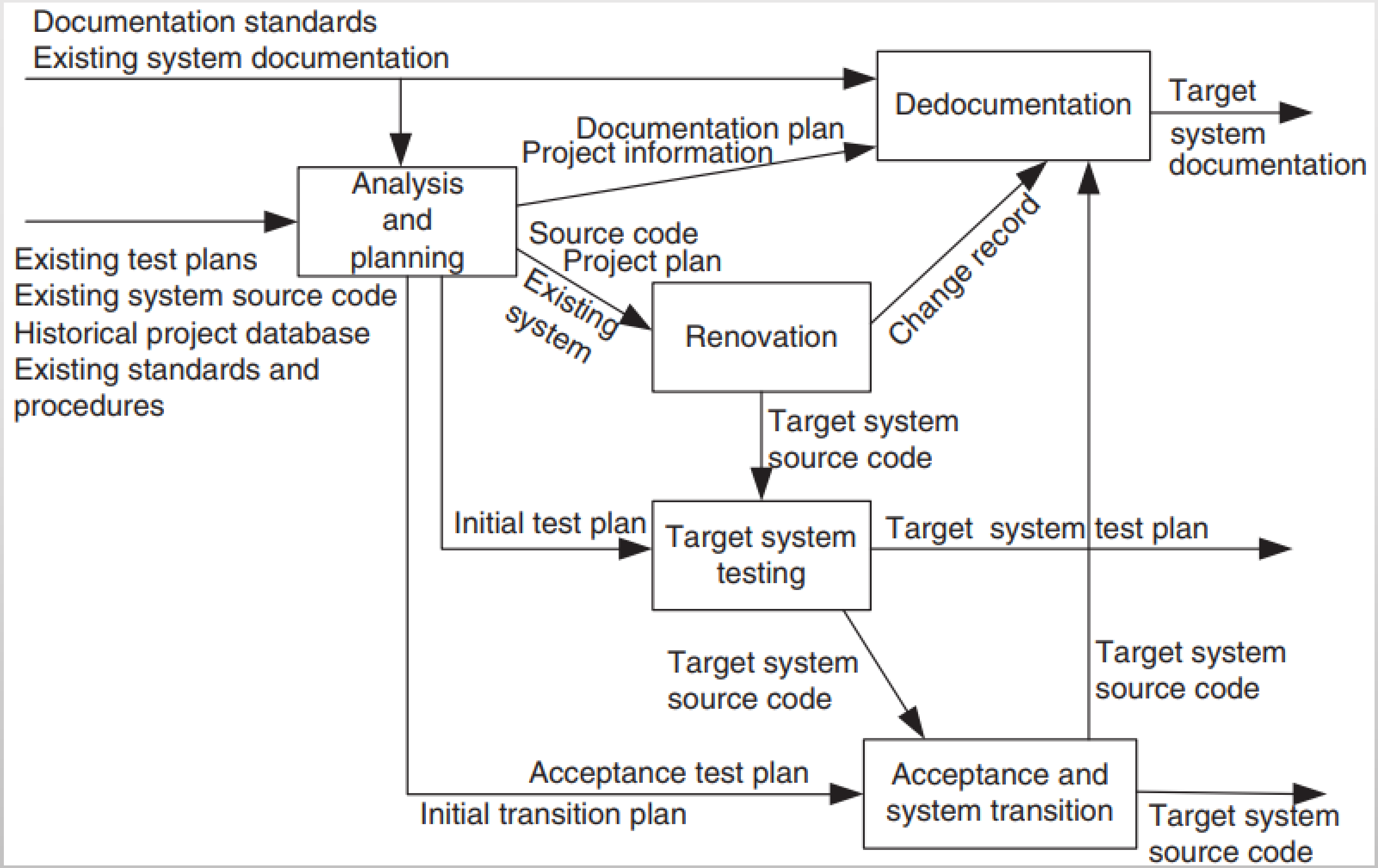
Content of Lecture

- Phase Re-Engineering Model

Phase Reengineering Model

- The phase model of software reengineering was originally proposed by Byrne and Gustafson.
- The model comprises five phases:
  - Analysis and planning,
  - Renovation,
  - Target system testing,
  - Re-documentation, and
  - Acceptance testing and system transition.
- The labels on the arcs denote the possible information that flows from the tail entities of the arcs to the head entities.

- A major process activity is represented by each phase.
- Tasks represent a phase’s activities.
- Tasks can be further decomposed to reveal the detailed methodologies.


Documentation standards
Existing system documentation
Dedocumentation
Target system documentation
Documentation plan
Project information
Analysis and planning
Existing test plans
Existing system source code
Historical project database
Existing standards and procedures
Source code
Project plan
Existing system
Renovation
Change record
Target system source code
Initial test plan
Target system testing
Target system test plan
Target system source code
Target system source code
Acceptance test plan
Initial transition plan
Acceptance and system transition
Target system source code

Analysis and Planning Phase

- Analysis addresses three technical and one economic issue.
- Technical issues:
- The first technical issue concerns the present state of the system to be reengineered and understanding its properties.
- The second technical issue concerns the identification of the need for the system to be reengineered.
- The third technical issue concerns the specification of the characteristics of the new system to be produced.
- Specifically, one identifies:
  - (i) The characteristics of the system that are needed to modified and
  - (ii) The functionalities of the system that are needed to be modified.

Analysis and Planning Phase

- Economic issue:
- The economic issue concerns a cost and benefit analysis of the reengineering project.
- The economics of reengineering must compare with the:
  - Costs,
  - Benefits,
  - Risks of developing a new system
  - The costs and risks of maintaining an old system

Analysis and Planning Phase

- Planning includes:

(i) Understanding the scope of the work;
(ii) Identifying the required resources;
(iii) Identifying the tasks and milestones;
(iv) Estimating the required effort; and
(v) Preparing a schedule.

Analysis and Planning Phase

- The tasks to be performed in this phase are listed in this Table.

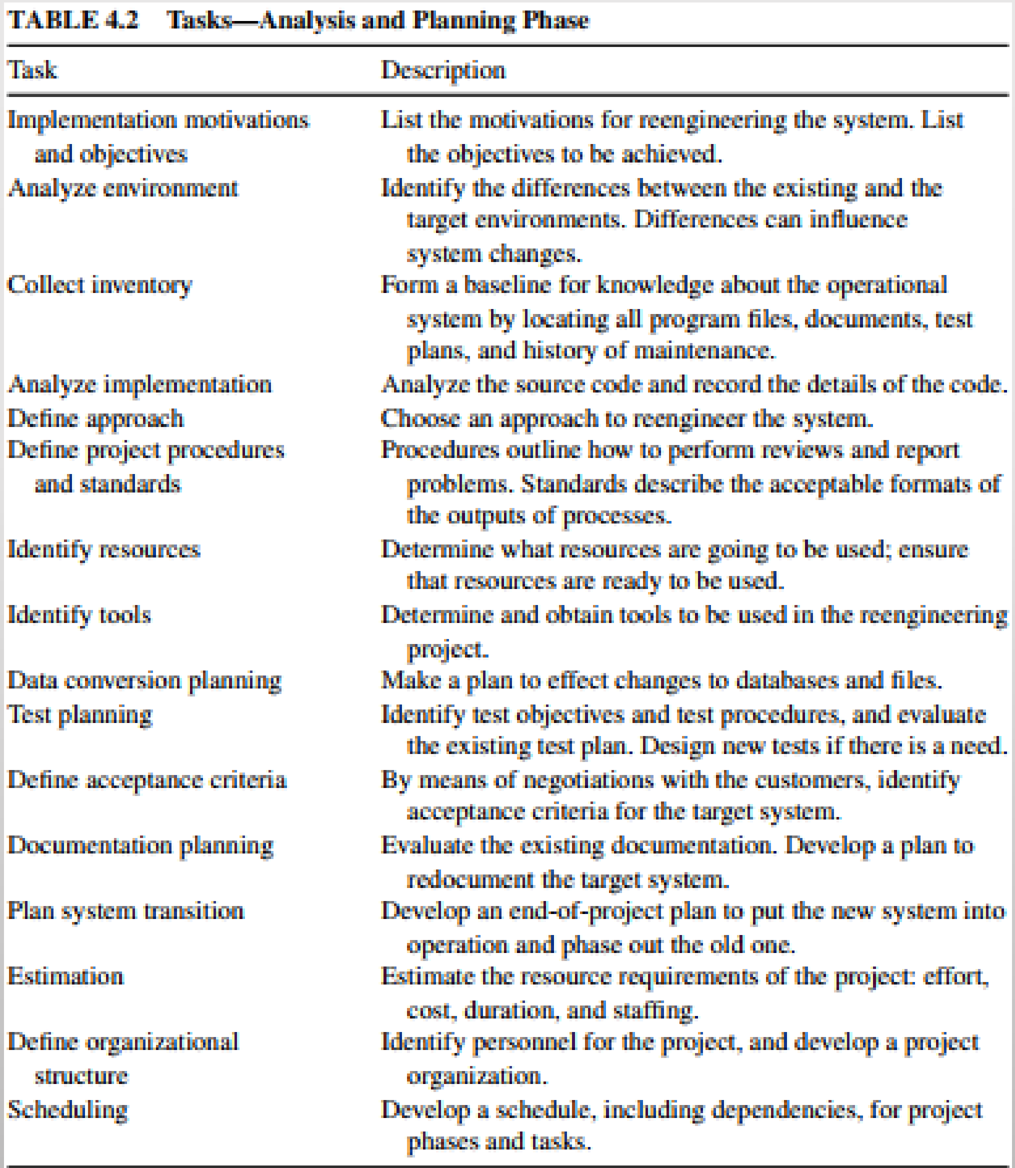


TABLE 4.2 Tasks—Analysis and Planning Phase

Task	Description
Implementation motivations and objectives	List the motivations for reengineering the system. List the objectives to be achieved.
Analyze environment	Identify the differences between the existing and the target environments. Differences can influence system changes.
Collect inventory	Form a baseline for knowledge about the operational system by locating all program files, documents, test plans, and history of maintenance.
Analyze implementation	Analyze the source code and record the details of the code.
Define approach	Choose an approach to reengineer the system.
Define project procedures and standards	Procedures outline how to perform reviews and report problems. Standards describe the acceptable formats of the outputs of processes.
Identify resources	Determine what resources are going to be used; ensure that resources are ready to be used.
Identify tools	Determine and obtain tools to be used in the reengineering project.
Data conversion planning	Make a plan to effect changes to databases and files.
Test planning	Identify test objectives and test procedures, and evaluate the existing test plan. Design new tests if there is a need.
Define acceptance criteria	By means of negotiations with the customers, identify acceptance criteria for the target system.
Documentation planning	Evaluate the existing documentation. Develop a plan to redocument the target system.
Plan system transition	Develop an end-of-project plan to put the new system into operation and phase out the old one.
Estimation	Estimate the resource requirements of the project: effort, cost, duration, and staffing.
Define organizational structure	Identify personnel for the project, and develop a project organization.
Scheduling	Develop a schedule, including dependencies, for project phases and tasks.

Renovation

- An operational system is modified into the target system in the renovation phase.
- There are two main aspects of a system are considered in this phase:
- <u>Representation of the system:</u>
- It refers to source code, but it may include the design model and the requirement specification of the existing system.
- <u>Representation of the external data:</u>
- It refers to the database and/or data files used by the system.
- Often the external data are reengineered, and it is known as data reengineering.

Renovation

- An operational system can be renovated in many ways, depending upon:
  - The objectives of the project,
  - The approach followed, and
  - The starting representation of the system.
- It may be noted that the starting representation can be source code, design, or requirements.

Renovation

- Example:
- A project in which the objective is to re-code the system from Fortran to C.
- Figure 2 shows the three possible replacement strategies.
- First, to perform source-to-source translation, program migration is used.
- Second, a high-level design is constructed from the operational source code, say, in Fortran, and the resulting design is re-implemented in the target language, C in this case.
- Finally, a mix of compilation and decompilation is used to obtain the system implementation in C.

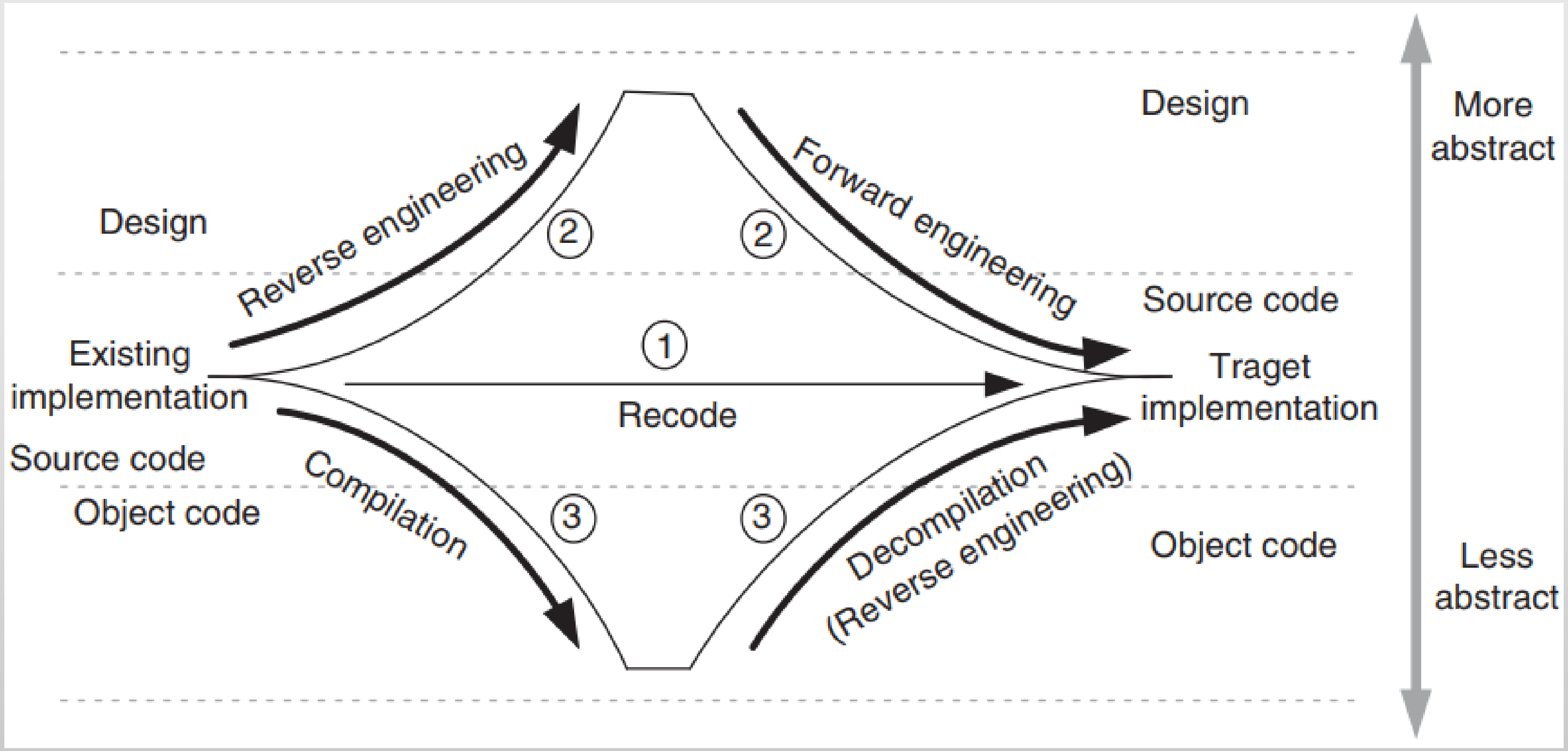


Figure 2: Replacement strategies for recoding

Target System Testing

- In this phase for system testing, faults are detected in the target system.
- Those faults might have been introduced during reengineering.
- Fault detection is performed by applying the target system test plan on the target system.
- The same testing strategies, techniques, methods, and tools that are used in software development are used during reengineering.
- Tools may include:
  - Static Analysis Tools
  - Dynamic Analysis Tools
  - Testing Frameworks

Target System Testing

- Testing frameworks provide a structured approach to software testing, automating the process and making it more efficient.
- They offer guidelines, tools, and best practices for creating, executing, and analyzing test cases.
- Popular types include:
- Linear scripting: This is the simplest type, recording user actions with web applications and playing them back as automated tests (e.g., Selenium). It's suitable for small applications and rapid prototyping.
- Modular: The application is broken down into smaller modules, and tests are created for each module independently. This approach promotes reusability of test code and makes maintenance easier.
- Data-driven: Test data is stored externally (e.g., in CSV, Excel) and loaded into test scripts. This allows for testing with multiple data sets, making tests more versatile and efficient.

Static Analysis Tools

- Static analysis tools in software testing are used to examine source code without actually executing the program, helping developers and testers identify potential issues like bugs, vulnerabilities, and coding style violations early in the development process.
- These tools automate the code review process, allowing for quicker and more efficient detection of defects.
- Example:
  - PVS-Studio: Detects bugs and potential vulnerabilities in C, C#, C++, and Java code.
  - Embold: A code analysis tool that identifies critical issues in the initial testing phase.

Dynamic Analysis Tools

- Dynamic analysis tools in software testing examine code execution during runtime to identify issues like errors, memory leaks, and performance bottlenecks.
- These tools simulate user inputs, monitor code behavior, and can be used for various purposes, including functional testing, security testing, and performance testing.
- Example:
  - Jmeter: A popular tool for load and performance testing, simulating multiple users accessing a web application simultaneously to identify bottlenecks.
  - Selenium: An automation framework used for web browser automation, enabling testers to simulate user interactions and test functionalities.

Testing Frameworks

- Testing frameworks provide a structured approach to organizing, executing, and managing software tests, enabling more efficient and effective software testing processes.
- They offer a combination of guidelines, tools, libraries, and best practices to streamline the creation, execution, and maintenance of test scripts.
- Key aspects of testing frameworks:
  - Structure and organization:
  - Automation
  - Reusability
  - Scalability

Redocumentation

- In the redocumentation phase, documentations are rewritten, updated, and/or replaced to reflect the target system.
- Documents are revised according to the redocumentation plan.
- The two major tasks within this phase are:
  - Analyze new source code, and
  - Create documentation.

Redocumentation

- Documents requiring revision are:
- Requirement specification
- Design documentation
- A report justifying the design decisions, assumptions made in the implementation
- Configuration
- User and reference manuals
- On-line help
- Document describing the differences between the existing and the target system

Acceptance and System Transition

- In this final phase, the reengineered system is evaluated by performing acceptance testing.
- Acceptance criteria should already have been established in the beginning of the project.
- Should the reengineered system pass those tests, preparation begins to transition to the new system.

Acceptance and System Transition

- On the other hand, if the reengineered system fails some tests, the faults must be fixed; in some cases, those faults are fixed after the target system is deployed.
- Upon completion of the acceptance tests, the reengineered system is made operational, and the old system is put out of service.
- System transition is guided by the prior developed transition plan.

Thank You!