Software Re-Engineering

Lecture: 07

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

Sequence [Todays Agenda]

Content of Lecture

- Software Re-Engineering Approaches

Re-Engineering Approaches

- There are five basic approaches to reengineering software systems.
- Each approach advocates a different path to perform reengineering.
- Several considerations are made while selecting a particular reengineering approach.
  - Objective of the project;
  - Availability of resources;
  - The present state of the system being reengineered;
  - The risks in the reengineering projects.

Re-Engineering Approaches

- The five approaches are different in two aspects.
  - The extent of reengineering performed
  - The rate of substitution of the operational system with the new one.

The Extent of Re-Engineering Performed

- The extent of reengineering refers to the degree or scope of changes made to an existing software system during its reengineering process.
- Reengineering involves improving or transforming a legacy system to meet current requirements, extend its lifespan, or make it more maintainable and efficient.
- The extent of reengineering can vary significantly depending on factors such as the system's age, technology stack, business needs, and the desired outcomes.

The Extent of Re-Engineering Performed

- It generally falls into different levels of change, which can be described as:
- Minor Changes (Corrective Reengineering): Involves fixing bugs, updating documentation, or addressing small issues without changing the core functionality.
- Moderate Changes (Adaptive Reengineering): Entails adapting the system to work with new environments, technologies, or platforms.
- Major Changes (Perfective Reengineering): This type of reengineering makes significant changes to the software to improve functionality, usability, or performance.
- Comprehensive Overhaul (Transformational Reengineering): This is the most extensive form, where the system may undergo a complete redesign, with significant changes to both its architecture and functionality. The goal is often to modernize the software, possibly re-implementing it with new technologies or frameworks to meet current and future business requirements.

The Rate of Substitution of Operational System with New One

- The rate of substitution of the operational system with the new one refers to the pace at which legacy system is replaced or integrated with a new, re-engineered system during the re-engineering process.
- This concept involves how quickly the new system takes over the role of the old system in daily operations, with minimal disruption to the business or users.

The Rate of Substitution of Operational System with New One

- Factors influencing the rate of substitution include:
  - System Complexity: Complex legacy systems often require a slower, more phased approach to ensure all parts of system are properly integrated.
  - Risk Management: If operational system is critical to business operations, the substitution process might be slower to avoid potential disruptions or downtime.
  - Implementation Strategy: The approach used to introduce the new system can affect the rate of substitution:
    - Big Bang Approach
    - Phased Approach
  - User Training and Acceptance: How quickly users can adapt to the new system impacts the substitution rate. More training and support for users might slow the rate of substitution to ensure proper adoption and minimize resistance.

Re-Engineering Approaches

- The five approaches have their own risks and benefits.
- In the following, the five basic approaches of software reengineering are introduced one by one.
  - Big Bang Approach.
  - Incremental Approach.
  - Partial Approach
  - Iterative Approach
  - Evolutionary Approach.

Big Bang Approach

- The Big Bang Approach replaces the whole system at once rather than being done gradually or in phases.
- This approach is typically characterized by a single, large-scale transition or "cutover" from the old system to the new one.
- All the users and processes switch to the new system simultaneously.
- Once a reengineering effort is initiated, it is continued until all the objectives are achieved and the target system is constructed.
- This approach is generally used if reengineering cannot be done in parts.

Big Bang Approach

- Example:
  - If there is a need to move to a different system architecture, then all components affected by such a move must be changed at once.

Big Bang Approach

- Advantages:
  - The consequent advantage is that the system is brought into its new environment all at once

- Disadvantages:
  - On the other hand, the disadvantage of Big Bang is that the reengineering project becomes a monolithic task, which may not be desirable in all situations.
  - In addition, the Big Bang consumes too much resources at once for large system and takes a long stretch of time before the new system is visible.

Incremental Approach

- A system is reengineered gradually and integrated into the existing system over time, rather than replacing the old system all at once (as with the Big Bang approach), one step closer to the target system at a time.
- Thus, for a large system, several new interim versions are produced and released.
- Successive
- Interim versions satisfy increasingly more project goals than their preceding versions.
- The desired system is said to be generated/reengineered after all the project goals are achieved.

Incremental Approach

- Advantages:
  - Locating errors becomes easier (newly added components).
  - It becomes easy for the customer to notice progress, because interim versions are released.
  - Lower risk due to code can be identified and monitored.
- Disadvantages:
  - Needs long time and careful version controls.
  - Even if there is a need, the entire architecture of the system cannot be changed.

Partial Approach

- Only a part of the system is reengineered and then it is integrated with the non-engineered portion of system.
- The following three steps are followed in the partial approach.
  - Existing system is partitioned into two parts: reengineered and the not reengineered.
  - In the second step, reengineering work is performed using selected approach.
  - The two parts are integrated to make up the new system.

Partial Approach

- Advantages.
  - Scope of reengineering is decided by considering the level of need, resources, cost, and urgency, etc.
- Disadvantages.
  - Special care must be taken while integrating the engineered part with the non-engineered part, otherwise, integration issues may happen.

Iterative Approach

- The reengineering process is applied on the source code of a few procedures at a time, with each reengineering operation lasting for a short time.
- This process is repeatedly executed on different components in different stages.
- Consider the following four areas during re-engineering process.
  - Old components not re-engineered.
  - Components currently being re-engineered.
  - Components already re-engineered.
  - New components added to the system
- Their coexistence is necessary for the operational continuity of the system.

Iterative Approach

- Advantages.
  - Continued operation of the system during reengineering process.
  - The maintainers' and the users' familiarities with the system are preserved.
- Disadvantages.
  - Need to keep track of the four types of components during the reengineering process.
  - In addition, both the old and the newly reengineered components need to be maintained.

Evolutionary Approach

- Similar to the incremental approach.
- Components of the original system are substituted with reengineered components but the existing components are grouped by functions and reengineered into new components.
- Focus is on identifying functional objects irrespective of the location of those components within the current system.
- As a result, the new system is built with functionally cohesive components as needed.

Evolutionary Approach

- Advantages.
  - The resulting design is more cohesive.
  - The scope of individual components is reduced.
- Disadvantages.
  - All the functions with much similarities must be first identified throughout the operational system; next, those functions are refined as one unit in the new system.


Thank You!