Software RE-Engineering
Assignment # 2
K20-1052 BSE-8B

Summary of Research paper: **THE STUDY AND APPROACH OF SOFTWARE RE-ENGINEERING**

The research paper discusses various approaches to software reengineering, which involves improving or transforming existing software systems. The main reengineering approaches mentioned are:

Big Bang Approach:

- Replaces the entire system at once.
- Useful for projects that need immediate resolution, like moving to a different architecture.
- Advantages: The entire system moves to the new environment simultaneously, no need to develop interfaces between old and new components.
- Disadvantages: High risk, may not produce the desired result, consumes many resources, requires a long time before the new system is ready.

Incremental Approach:

- The system is re-engineered in incremental phases or portions.
- Advantages: Faster delivery of system components, easier error control, customers can see progress, easier to manage changes to existing components.
- Disadvantages: Longer overall completion time, requires careful planning and control.

Evolutionary Approach:

- Parts of the system are replaced based on functionality rather than the existing system structure.
- Advantages: Modular design, reduced scope for single components, suitable for object-oriented conversion.
- Disadvantages: Same functions must be first defined in the current system and then refined.

Hybrid Re-engineering:

- Combines different reengineering levels, abstraction techniques, and methodologies based on project needs.
- Involves three tracks: Translation (converting code to new language), COTS (using commercial off-the-shelf components), and Custom (developing new custom code).
- Aims to reduce development time and cost by leveraging COTS, code translation, while maintaining required functionality.
- Risks include interface/integration issues between the different components.

Furthermore, the paper provides a structured overview of software re-engineering, its challenges, risks, approaches and the need for diligent planning and processes when undertaking such initiatives.

Reverse Engineering:

- The paper explains reverse engineering as the process of analyzing the existing system to identify its components, extract design information, requirements etc. at higher abstraction levels.
- This is a key first step before actually re-engineering the system.
- Reverse engineering aims to recover lost information, understand system functionality, detect side effects and prepare for efficient reuse.
- Techniques like program structure analysis, data structure extraction, document generation are used.

Forward Engineering:

- This step involves using the outputs from reverse engineering to actually develop the new/re-engineered system.
- It follows the standard software development life cycle of design, coding and implementation using the extracted requirements and architectural details.
- New technologies, languages, design paradigms can be adopted in this phase like object-orientation.
- Testing and quality assurance are vital to ensure functional equivalence with the legacy system.

Feasibility Analysis:

- Before initiating re-engineering, the paper suggests analyzing the costs, benefits, risks and business motivations carefully.
- This includes assessing the value of the application, desired quality improvements, maintenance efficiencies etc.
- Overall return on investment and costs of re-engineering versus redevelopment need evaluation.

Planning and Process:

- The paper describes a comprehensive 5-stage process for re-engineering projects:

1. Establish re-engineering team and objectives
2. Feasibility analysis
3. Requirements/design analysis and planning
4. Reverse and forward engineering implementation
5. Conversion, testing and maintenance transfer