


The Re-engineering and Reuse of Software



SUBMITTED BY: DR. SANGEETA ARORA
(COMPUTER SCIENCE AND IT DEPARTMENT)




RE-ENGINEERING

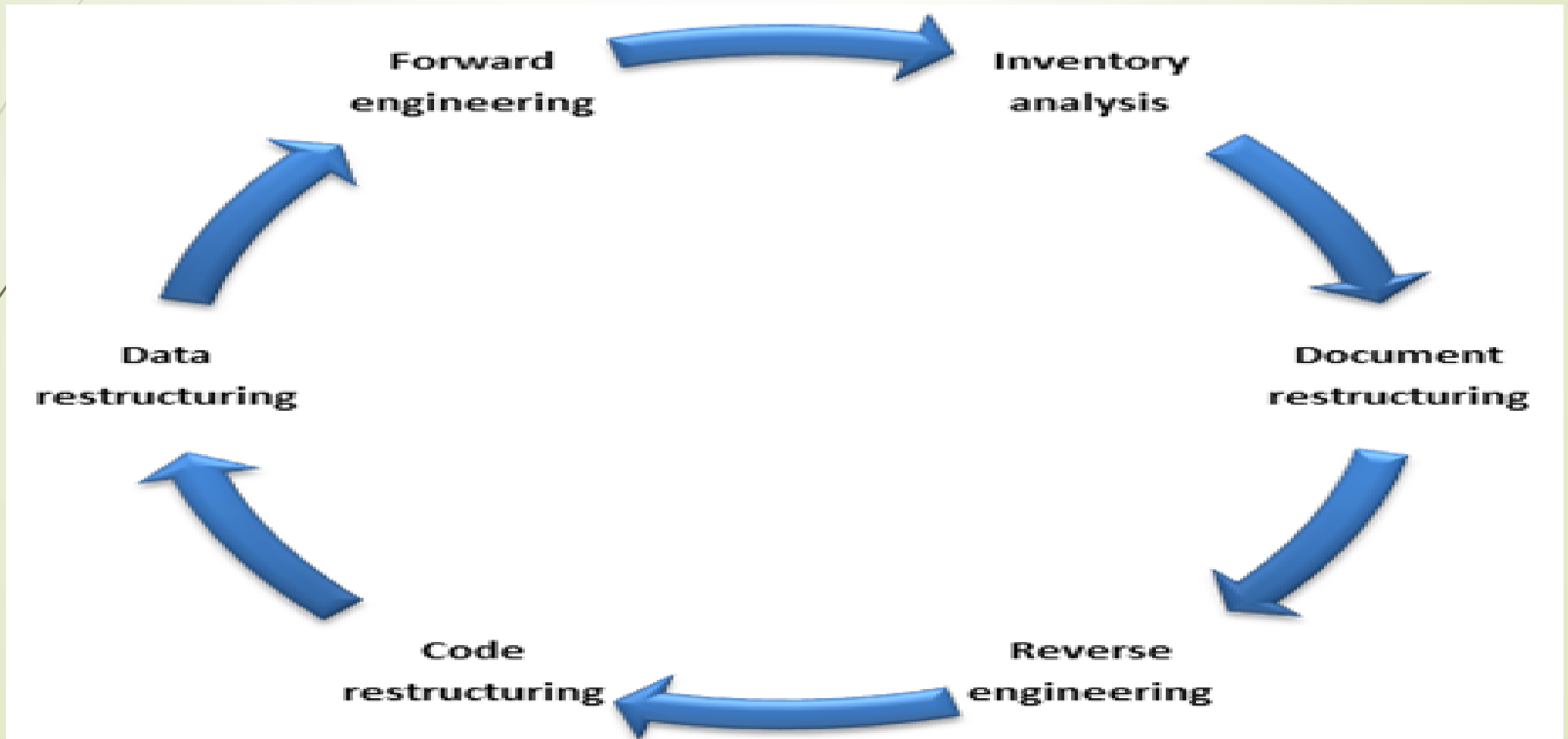
- Re-engineering is *the combination of reverse engineering and forward engineering*. The concept of “redoing” a software system.
- It is a process of software development which is done to improve the maintainability of a software system.
- Software reengineering is the examination and alteration of a system to reconstitute in a new form.



RE-ENGINEERING


- Software development which is done to improve the maintainability of a software system.
 - Reorganising and modifying existing software systems to make them more maintainable
- 

Software reengineering





Objectives of Software Reengineering

- To explain why software re-engineering is a cost-effective option for system evolution
 - To describe the activities involved in the software re-engineering process
 - To distinguish between software and data re-engineering and to explain the problems of data re-engineering
- 




History of reengineering

- Flow-charting tools have been around for awhile
 - Now evolved into UML and sequence diagrams
- Cross reference listings of variables
 - Helps understand large pieces of software
- Originally designed to help maintain COBOL legacy systems around 1980s




Need of software Reengineering

- Decreasing ratio of Successful projects.
 - Increasing number of competing Companies.
 - More demand for quality attributes.
 - Changing attitude of people.
 - Insistence for Software maintenance.
- 



Advantages of software Engineering

- Software Reengineering Cost
 - Improved Performance
 - Reduced Risk
 - Always Ready for Enhancements
- 



Disadvantages of Software Engineering

- Gap between problem /solution domain
- Gap between concrete and abstract ·
- Gap between coherency/disintegration ·
- Gap between hierarchical/associational

BENEFITS OF SOFTWARE RE-ENGINEERING

- Increased maintainability
- Improved performance
- Increased interoperability
- Decreased personal dependency
- Improved testability



Issues of Software Reengineering

- · Direct, visible benefits to the customer are vital for continued support. · Management commitment is crucial, learning curves require extra time, extra time causes slippage.
- Cost of planning, training, execution, testing and internal marketing are real engineering costs.
- Re-engineering requires a highly trained staff that has experience in the current and target system, the automated tools, and the specific programming languages.
- It is critical that the application system experts be involved throughout the reengineering process.
- They are essential for design recovery to identify hidden history.



Software Reuse

- ▶ Software reuse is the process of creating software systems from existing software rather than building software systems from scratch.
- ▶ This simple yet powerful vision was introduced in 1968. Software reuse has, however, failed to become a standard software engineering practice.
- ▶ In an attempt to understand why, researchers have renewed their interest in software reuse and in the obstacles to implementing it.



Advantages of Reuse

- Increase software productivity
- Shorten software development time
- Improve software system interoperability
- Develop software with fewer people
- Move personnel more easily from project to project
- Reduce software development and maintenance costs
- Produce more standardized software
- Produce better quality software and provide a powerful competitive advantage



Types of software reuse

- Horizontal reuse
 - Vertical reuse
- 



HORIZONTAL REUSE AND VERTICAL REUSE

- Horizontal reuse to software components used across a wide variety of applications.
- Vertical Reuse the reuse of system functional areas, or domains, that can be used by a family of systems with similar functionality .



Advantages of software reuse

- There is no need for writing code of software which already exists.
- It allows software to be inter-operable
- It allows software to be inter- operable
- It allows standardization of software
- Higher quality products
- Less development time
- Higher scheduling accuracy
- Reliability



Disadvantages of software reuse

- Application maintenance
- Long term reused code maintenance
- Tool support
- Not invented here syndrome
- Searching, evaluating, and adopting reused code



TOOLS USED IN REUSE

- Component management tools, such as repositories, for architectures, designs, documentation, and code must be developed and maintained
- Domain analysis tools and procedures for the development and maintenance of a domain architecture




CHALLENGES IN SOFTWARE REUSE

- As a developer with deadlines to meet and functionality to deliver it is challenging to keep reuse as a priority.
- Even when trying to design software for reuse, this often fails.
- Some efforts fail because they are overly ambitious where lot of big upfront design efforts are spent trying to design things future-perfect.
- Defining an organizational structure for maintaining the product line, including core assets and the customer specific products with special non-core functionality



CHALLENGES IN SOFTWARE REUSE

- Defining a process for producing a new member of the product line from the core assets with customer specific requirements
 - Defining a process for adding functionality to the core product line assets based on new customer requirements
 - Instituting a training program for reuse strategies in management, design, implementation, test-all phases of the development process.
- 



Risk with Software Reuse


- Needless complexity
- Inflexible design/will cost too much to modify
- Domain irrelevance
- Inadequate documentation/training/awareness
- Ability to meet deadlines/dates (schedule risk)
- Increased development, testing, and maintenance costs
- Pursuit of technical elegance/architectural purity

TAXONOMY OF REENGINEERING

- The re-engineering taxonomy defines re-engineering as the examination and alteration of software product to reconstitute in it a new form and subsequent implementation of new form. Traditionally proven methodology constituting 3 steps:
 - Reverse Engineering
 - Restructuring
 - Forward Engineering




REVERSE ENGINEERING

- Reverse engineering is the process of analysing a subject system to create representations of the system at a higher level of abstraction. It can also be seen as going backwards through the development cycle. Inventorying of the source codes, DLL s and other software components present
 - Debugging and recovery of source code
 - Making the software up and running in a demo environment to capture the business logic
 - Defining the existing architecture
 - Generating document for product description, installation and existing architecture
- 



RESTRUCTURING

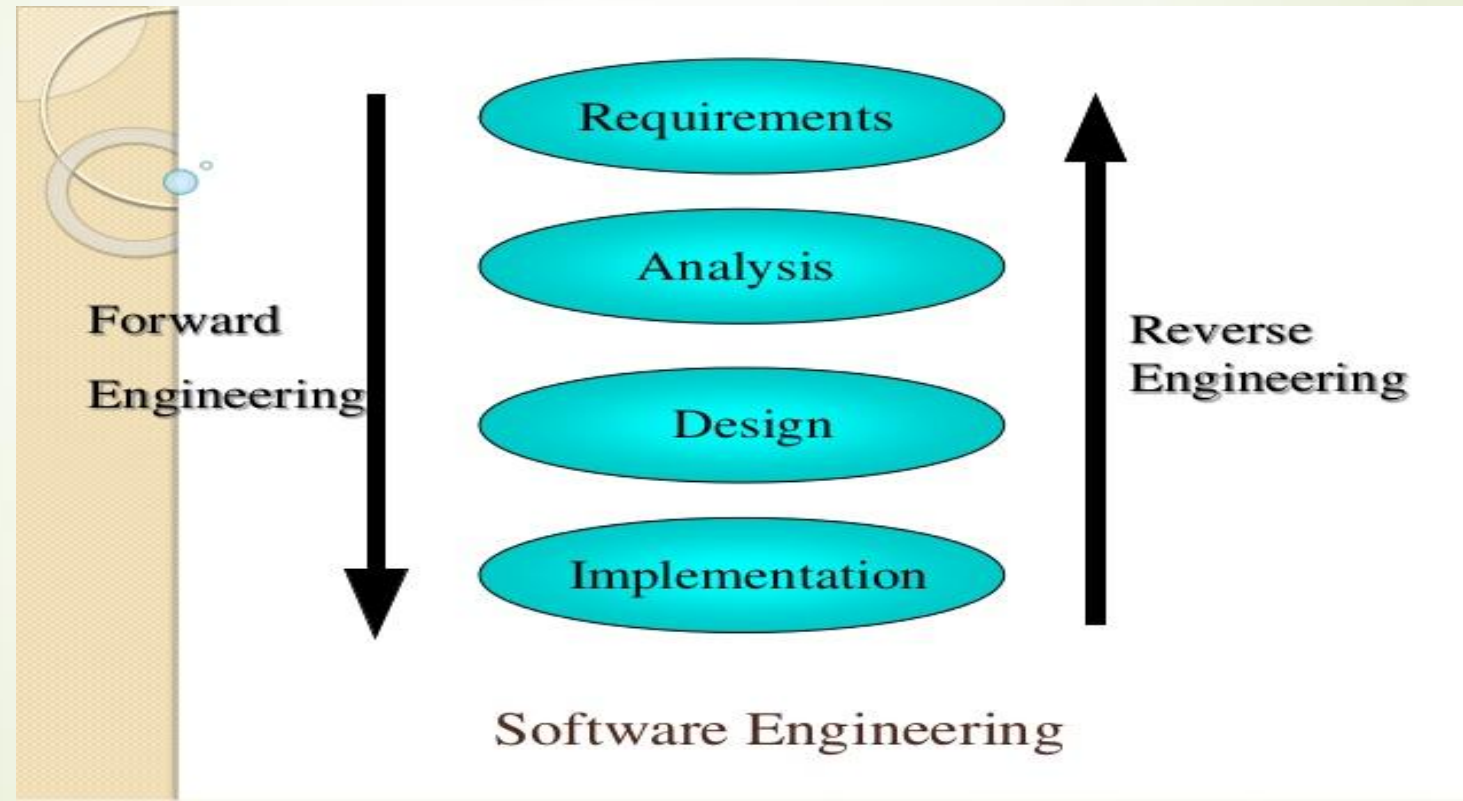
- Coming up with an architectural recommendation and best practices for re-engineering them in the client specified technology
High Level effort estimating of re-engineering for client required technology




Forward Engineering

- Conversion of the existing requirements and additional client specified requirements with the recommended technology and architecture into a software product with our Adaptive Product Development Lifecycle methodology.
- Forward engineering is the process of building from a high-level model or concept to build in complexities and lower-level details. This type of engineering has different principles in various software and database processes.

Forward and reverse reengineering






Need of Forward and Reverse Reengineering

- Assisting with maintenance
- Recovery of lost information
- Providing proper system information
- Facility of software reuse
- Synthesize higher abstraction




Advantages of Reverse Engineering

- ▶ you can work in most jobs in the technology industry, with good pay. You can build neat things. You get to work with very interesting science.
 - ▶ Reverse engineering in software is reversing a program's machine code back into its original source code.
 - ▶ The advantages to using this technique is that one can repair certain bugs, see how a program operates, and improve the operation of the program.
- 



Disadvantages of Reverse Engineering

- The disadvantage is that certain other bugs may be introduced during this process
 - At most schools, physics and engineering degrees are significantly more difficult to get than those in the humanities.
- 



Restructuring

- There are many Problems with re-structuring are:
 - Loss of comments
 - Loss of documentation
 - Heavy computational demands
- Restructuring doesn't help with poor modularisation where related components are dispersed throughout the code




Data reengineering

- ▶ Tools that perform all the reengineering functions associated with source code (reverse engineering, forward engineering, translation, documentation, restructuring/normalization, and retargeting) but act upon data files.
- ▶ Data reengineering tools help to translate flat files and hierarchical files to RDB's. Each reengineering activity for source code has a corresponding activity in data reengineering



Data Reengineering

- Reverse engineering of data captures the design information of data files.
 - Restructuring data could normalize that data. Data translation can move data files from a flat file configuration to a relational data base.
 - And retargeting data files assists the migration of data to new platforms.
- 



References

- ▶ <https://pdfs.semanticscholar.org/d6b2/89019f9fef924fd3300d1094f29c8bc079f9.pdf>
- ▶ <https://www.coursehero.com/file/p4lldiq/The-advantages-and-disadvantages-of-reverse-engineering-are-as-follows/>
- ▶ https://www.researchgate.net/publication/3501505_Data_reengineering_f_or_application_systems



Thank you

