

WWJMRD 2018; 4(2): 381-383
www.wwjmr.com
International Journal
Peer Reviewed Journal
Refereed Journal
Indexed Journal
UGC Approved Journal
Impact Factor MJIF: 4.25
E-ISSN: 2454-6615

Paramjeet Kaur
MCA, UGC -NET
Assistant Professor of
Computer Science
Guru Nanak College,
Ferozepur Cent. Punjab, India.

The Study of Software Re-Engineering

Paramjeet Kaur

Abstract

The nature of software re-engineering is to improve or transform existing software so it can be understood, controlled and reused as new software. To a large extent, it involves maintenance activities like predictive, corrective, perfective, and adaptive. Re-engineering may involve re-documenting the system organizing and restructuring the system, translating the system to a more modern programming language. The functionality of software is not changed and normally the architecture also remains same.

Keywords: Introduction, predictive, corrective, adaptive, reverse engineering, objective, conclusion

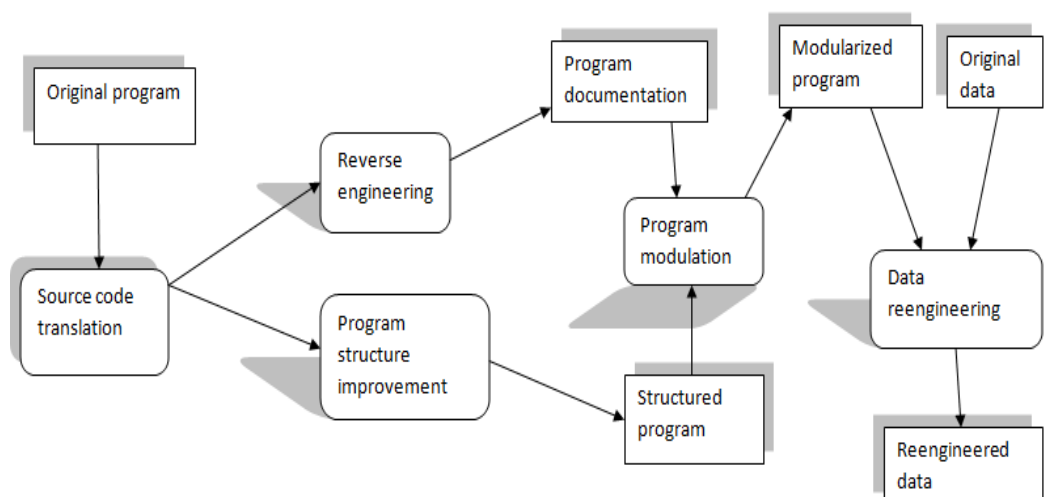
1. Introduction

Software Re-Engineering

Software re-engineering is concerned with re-implementing legacy system to make them more maintainable. The re-engineering re-use things already in the old software to avoid waste of material and spiritual reduce maintenance cost take to bring about the economic value, the most effective. The functionality of the software is not changed and normally the system architecture also remains same. It involves maintenance activities like understanding, repairing, improving and evolving

When deciding the software re-engineering?

- a) Software re-engineering applicable when some (but not all) subsystem of a large system requires frequent maintenance.
- b) When the software changes affect a subsystem and the subsystem that needs to be re-designed.
- c) When hardware and software becomes outdated. Software Re-engineering process is shown in diagram:



Correspondence:
Paramjeet Kaur
MCA, UGC -NET
Assistant Professor of
Computer Science
Guru Nanak College,
Ferozepur Cent. Punjab, India.

The goal of Re-engineering

The problem is that the system is in use today, the basic system to be a lack of well-designed structure and organization of code and changes the whole software system is difficult and expensive. Corporation do not want to destroy the system because it was built for many subsidiaries of the group which if destroyed will result in the application may have made will be lost. The initial cost for developing logic and component of the system software should not be wasted. Therefore, re-use through re-engineering is desired. There are four main objective of the re-engineering software

- Prepare for enhanced functionality.
 - Improve maintenance.
 - Access to the new platform.

- Improved reliability.

Although re-engineering should not be taken to enhance the functionality of existing systems, it is frequently used to prepare for advanced functionality.

General Model of Software Reengineering

Re-engineering starts with the source code of the basic system exist, and ends with the source code of the system will target. This process can be as simple as using the translation tool to translate source code from one language into another language (FORTRAN to C) or from one operating system to other operating systems (UNIX to DOS).

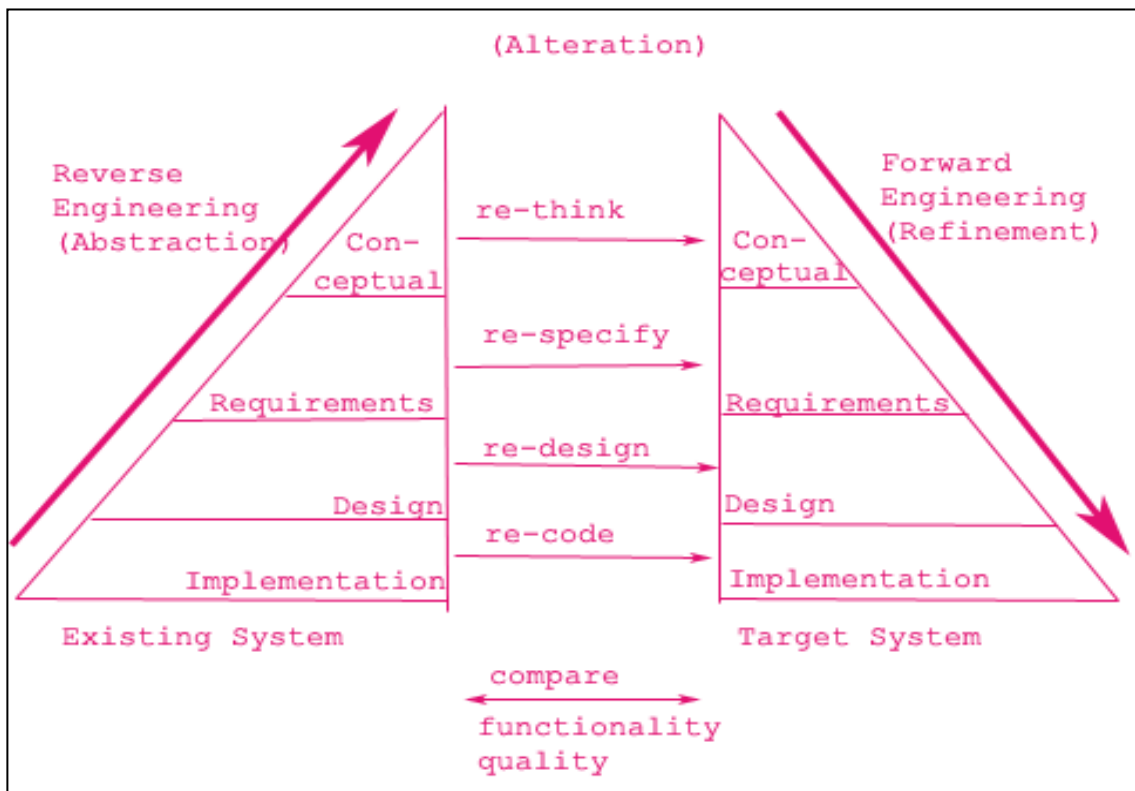


Fig: General model of software re-engineering.

The model in Figure applies three principles of re-engineering: abstraction, the amendment and refinement. This abstraction level is a gradual increase in the level of abstraction of the system. The present system was created by replacing a row of information is the information that is more abstract. Abstractions make the description emphasize the characteristics of the system. The upward movement is called reverse engineering and related accessories to the process, tools and techniques. The amendment is to create one or more to convert a representation of the system without changing the level of abstraction in which additional, delete and modify information. The refinement (Refinement) is the gradual reduction in the level of abstraction of the system caused by the continuous replacement information in existing systems With more detailed information. It is technically forward

(Forward engineering) as software developers with the code (code) but with some new screening process. To change a feature of the system, the work is done at the level of abstraction at which information about characteristics that are clearly presented. To translate the code (code) is languages aiming to reverse engineering are necessary, changes made at the level of implementation.

Reverse Engineering:

Reverse engineering is the process of analyzing software with the objective of recovering its design and specification. The program itself is unchanged by the reverse engineering process. The software source code is usually available as the input to the reverse engineering process. Sometimes, even this has been lost and the reverse engineering must start with the executable code.

Reverse Engineering is the process of analyzing a subject system with two main goals in mind:

- To identify the system's components and their interrelationships.

- To create representations of the system in another form or at higher level of abstraction.

The reverse engineering process:

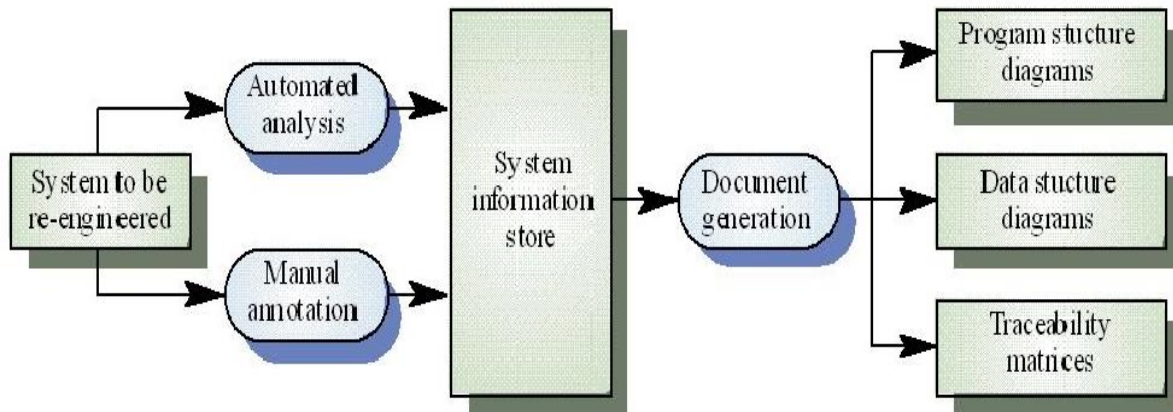


Fig: illustrates a possible re-engineering process. The input to the Process is a legacy program and the output is structured, modularized version of the same program

Data Re-Engineering

The process of analyzing and re-organizing the data structure and sometimes the data values in a system to make it more understandable is called data re-engineering. In principle, data re-engineering should not be necessary if the functionality of a system is unchanged. In practice, however, there are a number of reasons why you may have to modify the data as well as the programs in a legacy system

1. Data degradation over the time, the quality of data tends to decline.
2. Architectural evolution if a centralized system is migrated to a distributed architecture it is essential that the core of that architecture should be a data management system that can be accessed from remote clients. This may require a large data re-engineering effort to move data separate files into the server database management system.

Re-engineering a software system has two key advantages: Reduced risk there is a high risk in re-developing software that is necessary for an organization.

1. The cost of re-engineering is significantly less than the costs of developing new software.

The Risks and challenges of system Re-engineering:

- Integration with business engineering.
- Targeting non-traditional users.
- Integrating heterogeneous tool sets.

Conclusion

Many new software design methodologies and tools have been developed to improve reusability, maintainability and to decrease the cost of development and maintenance. Most companies have software systems that are out of date and costly to maintain. So re-engineering is the best solution to Replace the existing software systems. Data are also needed during the re-engineering, but it is not clear exactly

how the concept can be considered and expected that the actual data, evaluate what they are proposing to define quantify. Paper is a summary of strategy in re-engineering, to serve as a basis for future work.

References

1. Roger S. Pressman, Software Engineering, a Practitioner's approach; Fifth edition, McGraw Hill, 2001
2. Institute of electrical and electronics engineers (IEEE) www.ieee.org
3. <https://arxiv.org/ftp/arxiv/papers/1112/1112.4016.pdf>
4. <https://pdfs.semanticscholar.org/bb45/6475ef1ec0e2aaeafc47682dc3386c58f212.pdf>
5. M. Solvin, and S. Malik. "Re-engineering to reduce system maintenance: A case study", Software Engineering, pp.14-24, 2011.