



National University of Computer & Emerging Sciences, Karachi  
Quiz 1 – SREE – BSE – 8A (Spring-2024)



20<sup>th</sup> February 2024 (Tuesday), 12:00 Noon – 12:55 PM

Course Code: SE4001	Course Name: Software Re-Engineering (SREE)
Instructor Name: Dr. Syed Muazzam Ali Shah	
Student Roll No:	Section No:

Time: 40 minutes.

Max Marks: 10 Points

**Q1: Explain why do we need to re-engineer an existing (legacy) software system?**

**Answer:**

- Software system is an integral part of the organization.
- System must be regularly maintained, and maintenance is becoming a large cost factor.
- Less risk and cost than redevelopment.

**Q2: Differentiate between re-engineering and refactoring?**

**Answer:**

**Re-engineering** takes place after a system has been maintained for some time and maintenance costs are increasing. You use automated tools to process and re-engineer a legacy system to create a new system that is more maintainable.

**Refactoring** is a continuous process of improvement throughout the development and evolution process. It is intended to avoid the structure and code degradation that increases the costs and difficulties of maintaining a system.

**Q3: Discuss some of the re-engineering challenges and difficulties.**

**Answer:**

Reasons why legacy systems have not been designed to accommodate change:

- Short life expectancy – not anticipated to last decades when first developed.
- Failure of process models and software engineering culture to treat evolution as a first class activity – future requirements ignored.
- Satisfying constraints that existed at the time of development – hardware (memory and processing power).

**Q4: Explain the Gradual Migration strategy of re-engineering?**

**Answer:**

Gradual Migration means rearchitect and transition gradually

When can the gradual migration option be used?:

- Integration is not economical over a long period of time or those legacy systems are to be phased out.
- There is an immediate need to update some of the more important components.
- Time and resources are not available to do the reengineering all at once.
- It is unclear how the reengineering effort will proceed – need for prototyping of different reengineering methods.

**Q5: Differentiate between evolution and servicing.**

**Answer:**

When software is first used successfully, many changes to the requirements by stakeholders are proposed and implemented. This is However, as the software is modified, its structure tends to degrade, and system changes become more and more expensive. This often happens after a few years of use when other environmental changes, such as hardware and operating systems, are also required. At some stage in the life cycle, the software reaches a transition point where significant changes and the implementation of new requirements become less and less cost-effective. At this stage, the software moves from evolution to servicing. During the servicing phase, the software is still useful, but only small tactical changes are made to it. During this stage, the company is usually considering how the software can be replaced.

**Q6: In which circumstances do we need to adopt spiral model of development and evaluation.**

**Answer:**

Spiral model of development and evaluation is appropriate for the software systems where there is a need to release a series of version after the system has been delivered because of the continuous change requests.

**Q7: Discuss two advantages of re-engineering over re-implementation.**

**Answer:**

- Reduced risk
  - There is a high risk in new software development. There may be development problems, staffing problems and specification problems.
- Reduced cost
  - The cost of re-engineering is often significantly less than the costs of developing new software.

**Q8: Discuss two bad smells that can be improved using refactoring.**

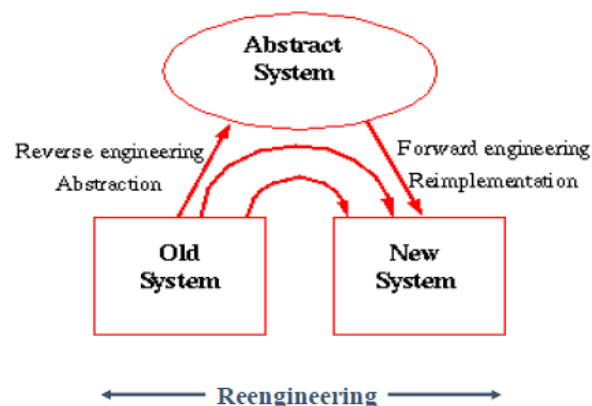
**Answer:**

- **Duplicate code**
  - The same or very similar code may be included at different places in a program. This can be removed and implemented as a single method or function that is called as required.
- **Long methods**
  - If a method is too long, it should be redesigned as a number of shorter methods
- **Switch (case) statements**
  - These often involve duplication, where the switch depends on the type of a value. The switch statements may be scattered around a program. In object-oriented languages, you can often use polymorphism to achieve the same thing.

**Q9: You can re-engineering legacy systems using two strategies/techniques/methods to make the legacy system easier to maintain. What are those techniques/strategies/methods?**

**Answer:**

**Reverse Engineering and Forward Engineering**



**Q10: Why it is expensive to add new functionality or made modifications to existing functionality while maintaining an existing (legacy) software system?**

**Answer:**

Incorporating new functionality or new requirements to the existing system may require extensive analysis and validation during requirements specification, design and implementation rework. Therefore, it is often expensive to add new functionality to the existing system.