**Chapter 9 : Project Management**

**9.1**

****

**About concerning system change, Lehman and Belady introduce set of laws. In this “Continuing change” is one.**

**Law: A system that is used in a real-world environment necessarily must change or become progressively less useful in that environment.**

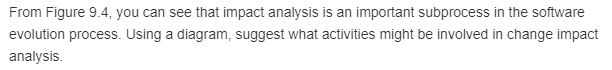
**- According to this, it states that system maintenance is an inevitable process. As the system’s environment changes, new requirements emerge and the system must be modified. When the modified system is re-introduced into the environment, this promotes more environmental changes so the evolution process recycles.**

**- If the number of users may increases, then the burden of the system requiring is expanding to its hardware capability for handle several connections.**

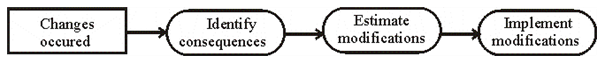
**- The business model of the company may change so the system become obsolete and need for a change to cope its requirements.**

**- The law in the particular country may impose a particular standard to conform legal usable software.**

**9.2**

****

**Following diagram shows the activities that might be involved in the change in the impact analysis.**

****

**Software evolution processes is vary and depending on the type of software being maintained. And, the change impact analysis includes the activities like identifying the consequences of the changes occurred, estimating the modifications need to be done to accomplish the changes and implementing the modifications**

**9.3**

****

**Legacy system**

**The old software systems which are still used for business purposes and introduced in 1960s are known as legacy systems. The reasons for using legacy systems as sociotechnical systems are as given below:**

**• These systems are developed for specific hardware and maintenance of hardware is very expensive. Another thing is that these hardware are no longer used and don’t satisfies the purchase policies of IT organizations.**

**• The legacy software is obsolete and original vendors are not ready to provide support.**

**• A huge amount of data is generated by legacy software in its lifetime. It is possible that it contains duplicate data, inconsistent data or may be the data is stored in different databases.**

**• The legacy software is not fitting in the IT policies of the business organization.**

**9.6**

****

**Legacy systems are responsible for the quality software, their application, system’s support and hardware. They need to ensure the quality of the support software. Problems with the support software may mean that the functioning or performance of the legacy systems is not as expected. Organizations which depend on many legacy systems and which have a limited budget for maintaining and upgrading these systems have to decide how to get the best return on their investment. This means that they should make a realistic assessment of their legacy systems. So, when problems with support software are found, the organization thinks of replacing their legacy systems so as to improve their quality.**

**9.4**

****

**“High quality and high business value” is the one of the legacy system categories. And it continues in operation using normal system maintenance.**

**In an organization, scrap is decided on local circumstances.**

**Where software might be scrapped and rewritten are:**

**1. When the cost of maintenance is high and the organization has decided to invest in new hardware. This will involve significant conversion costs anyway so the opportunity might be taken to rewrite the software.**

**2. When a business process is changed and new software is required to support the process.**

**3. When support for the tools and language used to develop the software is unavailable. This is a particular problem with early 4GLs where, in many cases, the vendors are no longer in business.**

**Example 1:**

**From a business perspective and a technical perspective in a legacy system, business perspective is decided whether or not the business really needs the system. And from a technical perspective, it assesses the quality of the application software and the system’s support software and hardware.**

**If it uses a combination of the business values and the system quality, then “High quality-high business value” has to be kept in this operation. Here, high quality means that it doesn’t have to invest in transformation or system replacement. Normally system maintenance should be continued.**

**To assess the business values of a system, you have to identify system stakeholders, such as end-users of the system and their managers, and ask a series of questions about the system i.e.:**

* **The use of the system**
* **The business processes that are supported**
* **The system dependability**
* **The system outputs**

**9.5 **

**Legacy system is an operational system that has been designed, implemented and installed in a radically different environment. In critical business systems, it can use many legacy systems.**

**The most appropriate strategic options for evolving these systems are:**

**1. Scrap the system completely. This option is performed when the system is ineffective to the business process or the business process has changed, leaving the system to be obsolete.**

**2. Leave the system unchanged and continue regular maintenance. This option is chosen when the system is still required but is fairly stable and change requests is rarely happen.**

**3. Reengineer system to improve maintainability. This is performed when the system maintenance cost exceeds the cost of reengineering the system.**

**4. Replace all or part of the system with a new system. This is to be chosen when old system can’t continue operation or where off-the-shelf systems would allow the new system to be developed at a reasonable cost.**

**It would normally chosen by the replacement option for all or part of a system rather than continue maintenance of the system.**

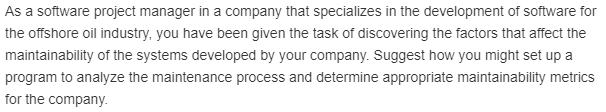
**In this situation,**

**1. The hardware platform for the system is being replaced.**

**2. The system wishes to standardize on some approach to development that is not consistent with the current system.**

**3. Some major sub-system is being replaced (e.g. a database system) or where the technical quality of the existing system is low and there are no current tools for re-engineering.**

**9.7**

****

**The software project manager has to go for at least one offshore industry to which the organization is going to develop the software. Various issues like the location, environment, people in the industry, users of the software etc are to be noticed personally by going there. By knowing all these issues, the organization can identify various metrics like time, cost of maintenance, reachability, understandability and usability of users etc., can be identified. Also some ideas and suggestions from the users are to be known to assess their way usage of the software. By assessing the cost and time to reach to the off-shore industry, the maintenance costs can be assessed.**

**9.8**

****

**Types of software maintenance:**

**1. Fault repairs: Coding errors are usually relatively cheap to correct; design errors are more expensive as they may involve rewriting several program components. Requirements errors are the most expensive to repair because of the expensive system redesign which be necessary.**

**2. Environmental adaptation: This type of maintenance is required when some aspect of the system’s environment such as the hardware, the platform operating system, or other support software changes the application system must be modified to adapt it to cope with these environmental changes.**

**3. Functionality addition: This type of maintenance is necessary when the system requirements change in response to organizational or business change .the scale of the changes required to the software is often much greater than for the other types of maintenance.**

**In practice, there is not a clear-cut distinction between these types of maintenance, when the system adapt to new environment, then add functionality to take advantage of new environmental features. Software faults are often exposed because users use the system in unanticipated ways. These types of maintenance are recognized but a different person sometimes gives them different names.**

**‘Corrective maintenance’ is universally used to refer to maintenance for fault repair’**

**“Adaptive maintenance’ sometimes means adapting to new environment and sometimes means adapting the software to new requirements.**

**‘Perfective maintenance’ sometimes means perfecting the software by implementing new requirements; in other cases it means maintaining the functionality of the system but improving its structure and performance.**

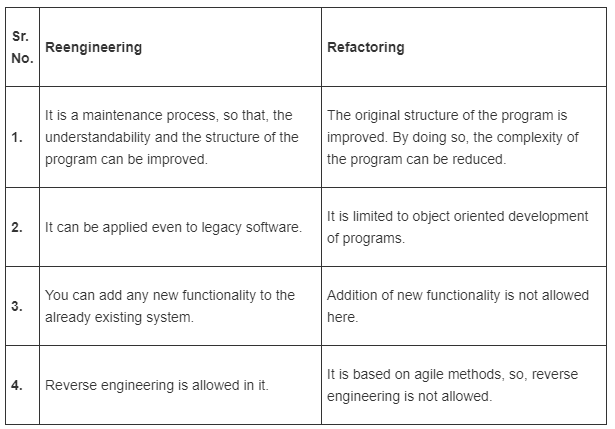
**9.9**

****

**Software refactoring and reengineering**

**The processes of software engineering which involves refactoring architecture of software system, re-documentation, translating programming language, and modifying the data structure and values is known as software reengineering.**

**On the other hand, the improvement process of any program or software, so that, the degradation can be slow down is known as refactoring.**

**The table to show the differences between Software refactoring and reengineering is as given below**

**9.10**

****

**Yes, software engineers have a professional responsibility to produce a code that can be maintained and changed. Even if it is not requested by their employer, it is one of their responsibilities to produce such a code. It is one of their professional responsibilities. A software engineer’s responsibility is to develop a code that is easy to understand, maintained, changed and flexible. These are the minimum requirements and considerations that a software engineer has to keep in mind while developing the code. A code that can be maintained and changed will always be a benefit in future for making enhancements.**