

UNIT-9

Software Evolution

⊗ Software Evolution Process:

Software development process does not end when system is delivered but continue through the lifetime of system. After a system has been deployed it needs to be changed time to time due to factors like: requirement changes, Environment changes, Errors or security breaches, improvements to system etc. This time to time process of developing, maintaining, and updating software for various reasons is referred to as Software evolution.

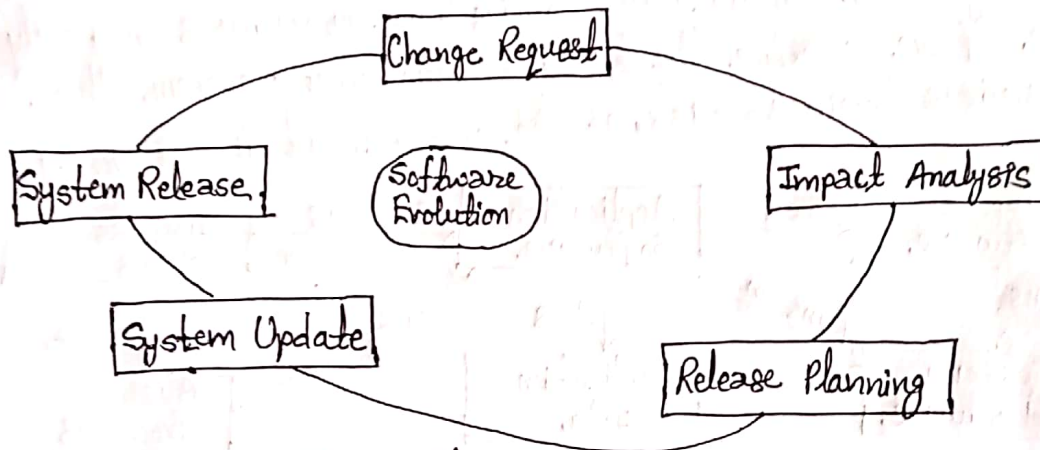


Fig: Software Evolution Process

Software evolution processes vary depending on type of software being maintained, the development process used in an organization, and the skills of people involved. However evolution process includes the fundamental activities of change request, impact analysis, release planning, system update, and system release.

Evolution process starts from the change request process. After which the impact of these changes are analyzed to see how much of the system is affected by the change and much it might cost to implement the change. If the proposed changes are accepted a new release of the system is planned. Then the changes are implemented to the next version of system and the system is released to customers.

Legacy Systems:

A legacy system is a computer system, programming language, software application, process or other technology that is outdated or that can no longer receive support and maintenance but is essential for organizations or companies and cannot be replaced or updated for different reasons.

Legacy systems are socio technical computer based systems that have been developed in past using older technology. They include application software together with business process, support software, system hardware etc.

For example: In a bank, banking management system was one of their earliest systems. Organization policies and procedures may rely on this system. If we replace the banking management system there would be a serious business risk, if the replacement did not work properly.

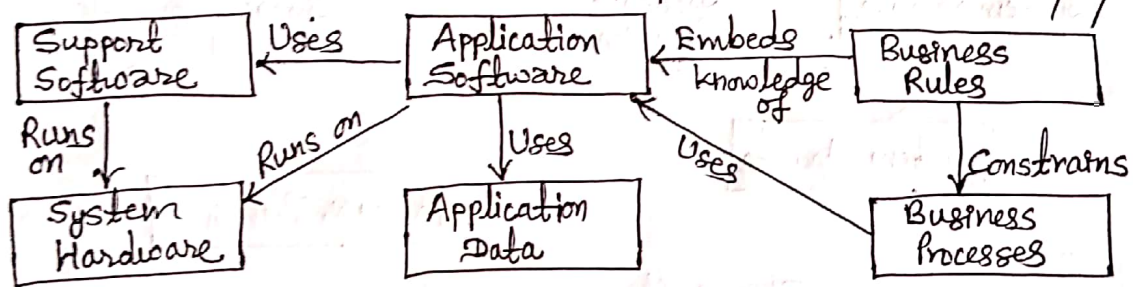


Fig: Legacy System Components

Strategies for evolving legacy systems:

- i) Scrap the system completely: When the system is not making an effective contribution to business processes then the system will be scrapped.
- ii) Reengineer the system to improve its maintainability: This method is valid when the system quality has been degraded by change and where a new change to the system is still being proposed.
- iii) Leave the system unchanged and continues with regular maintenance: This option is chosen when the system is still required but is fairly stable and the system users make relatively few change requests.

iv) Replace all or part of the system with a new system:

This option should be chosen when factors, such as new hardware, mean that the old system cannot continue in operation or where off-the shelf systems would allow the new system to be developed at a reasonable cost.

⊗ Software Maintenance:

Software maintenance is an activity which includes optimization, error correction, and deletion of discarded features and enhancement of existing features. Maintenance does not normally involve major changes to the system's architecture. Changes are implemented by modifying existing components and adding new components to the system. There are four major activities that occur during maintenance:

i) Obtaining maintenance requests: In this step a formal process is established where users can submit system change requests.

ii) Transforming Request into Changes: Once a request is received, analysis must be performed to identify scope of the request. It must be determined how the request will affect the current system and how long such a project will take.

iii) Designing changes: A change request can be transformed into a formal design change, which can then be fed into maintenance phase.

iv) Implementing changes: Once the change design is approved, proposed changes are implemented in respective components of the system.

Types of Maintenance:

i) Corrective maintenance: Corrective maintenance deals with the repair of faults or defects found in day-day system functions. It refers to changes made to repair defects in the design, coding, or implementation of the system.

ii) Adaptive maintenance: Adaptive maintenance is the implementation of changes in a part of system, which has been affected by a change that occurred in some other part of the system.

iii) Perfective maintenance: Perfective maintenance involves making functional enhancements to the system to increase systems performance.

iv) Preventive maintenance: Preventive maintenance involves performing activities to prevent the occurrence of errors.

⊗. The Cost of Maintenance:

The cost of maintenance represent a large proportion of the budget of most organization that use the software system. For some organizations, as much as 60 to 80 percent of their information systems budget is allocated to maintenance activities. These huge maintenance costs are due to the fact that many organizations have accumulated more and more older legacy systems that require more and more maintenance.

Factors influencing Maintenance Cost:

i) Latent defects: This is the number of unknown errors existing in the system after it is installed.

ii) No. of customers: Greater the number of customers in the system, greater the maintenance costs.

iii) Quality of system documentation: Higher the quality of system documentation, lower the maintenance costs.

iv) Tools: Tools that can automatically produce system documentation can also lower maintenance costs.

v) Well-structured programs: Well-designed system is easier to understand and fix. So, more well-structured programs in system leads to lower maintenance costs.