**Software Evolution**

**Definition:**

Software Evolution is technique of developing software using principles and methods of Software Engineering

**The Software evolution process includes:**

* Initial software development
* Software maintenance
* Software Upgrade
* Desired software product that meets user needs

**Importance of Software evolution (Change)**

* The business environment changes
* New requirement emerge when software is used
* The performance or reliability may have to be improved
* Errors must be repaired
* New equipment must be accommodated

NB: A Key problem for the organization is implementing and managing change to their legacy systems

A legacy system is a degraded system which is still valuable to the stakeholders.

**Software change control**

Software change control is the process of managing the processes of software change.

**Reasons for urgent changes**

* When there are business changes that require a very rapid response
* When serious system fault has to be repaired to allow normal operation to continue;
* When changes to the system’s environment have unexpected effects;

**Software change strategies**

There are three main software evolution (change) strategies

* Software maintenance
* Architectural Transformation
* Software re-engineering

**Software maintenance**

These are changes made in response to changed requirements but the fundamental structure of software remain stable (unchanged).

**Categories or Types of Maintenance**

**Perfective Maintenance**

Is the modification of software product after delivery to improve on the performance, usability and maintainability.

**Corrective Maintenance:** Is the reactive modification of software product performed after delivery to correct discovered faults**.**

**Adaptive Maintenance:** Is the modification of a software after delivery to keep a computer program usable in a changed or changing environment.

**Preventive Maintenance:** Is a software modification performed for purposes of preventing problems before they occurs in the software

**Factors determining the cost of maintaining software**

**Team stability**

Maintenance costs are reduced if the same staff are involved with them for some time.

**Contractual responsibility**

The developers of a system may have nocontractual responsibility for maintenance so there is no incentive to design for future change.

**Staff skills**

Maintenance staff are often inexperienced and have limited domain knowledge.

**Program age and structure**

As programs age, their structure is degraded and they become harder to understand and change.

**Program Evolution Dynamics**

Program evolution (change) dynamics is the study of the processes of the software (system) change.

There are two major study of system (software) evolution change:

* Lehman
* Belady

**Lehman’s Law**

**Continuing of change**

A program that is used in a real-world environment must necessarily change, or else become progressively less useful in that environment.

**Conservation of familiarity**

Over the lifetime of a system, the incremental change in each release is approximately constant.

**Increasing complexity**

As an evolving program changes, its structure tends to become more complex. Extra resources must be devoted to preserving and simplifying the structure.

**Declining quality**

The quality of systems will decline unless they are modified to

reflect changes in their operational environment.

**Large program evolution**

Program evolution is a self-regulating process. System attributes such as size, time between releases, and the number of reported errors is approximately invariant for each system release

**Feedback system** Evolution processes incorporate multiagent, multiloop feedback systems and you have to treat them as feedback systems to achieve significant product improvement.

**Organizational stability**

Over a program’s lifetime, its rate of development is approximately constant and independent of the resources devoted to system development

**Continuing growth**

The functionality offered by systems has to continually increase to maintain user satisfaction.

**Evolution Processes Variation**

Software evolution processes vary depending on the following condition

* The type of the software being maintained
* The development processes used
* The skills of the people involves in the development

Software evolution process involves the following cyclic steps

* Change request
* Impact analysis
* Release planning
* Change implementation
* System release

Change request

* In this phase a customer request for the change

Impact Analysis:

During this step, the cost and impact of the changes on the system is assessed to find:

* How of the system will be affected by the change
* How much it might the cost to implement the change.

NB: When the proposed changes are accepted then the process moves to release planning step.

Release planning step