Software Evolution

Lehman’s Law, Maintenance

Deployment

Deployment

- Finally, a time has come where you will deliver products to the client.

- This could be considered the last phase of SDLC in many of the cases, off course if you are not responsible for the maintenance.

- If you are the part of maintenance phase then there is one more phase for you called maintenance.

Deployment

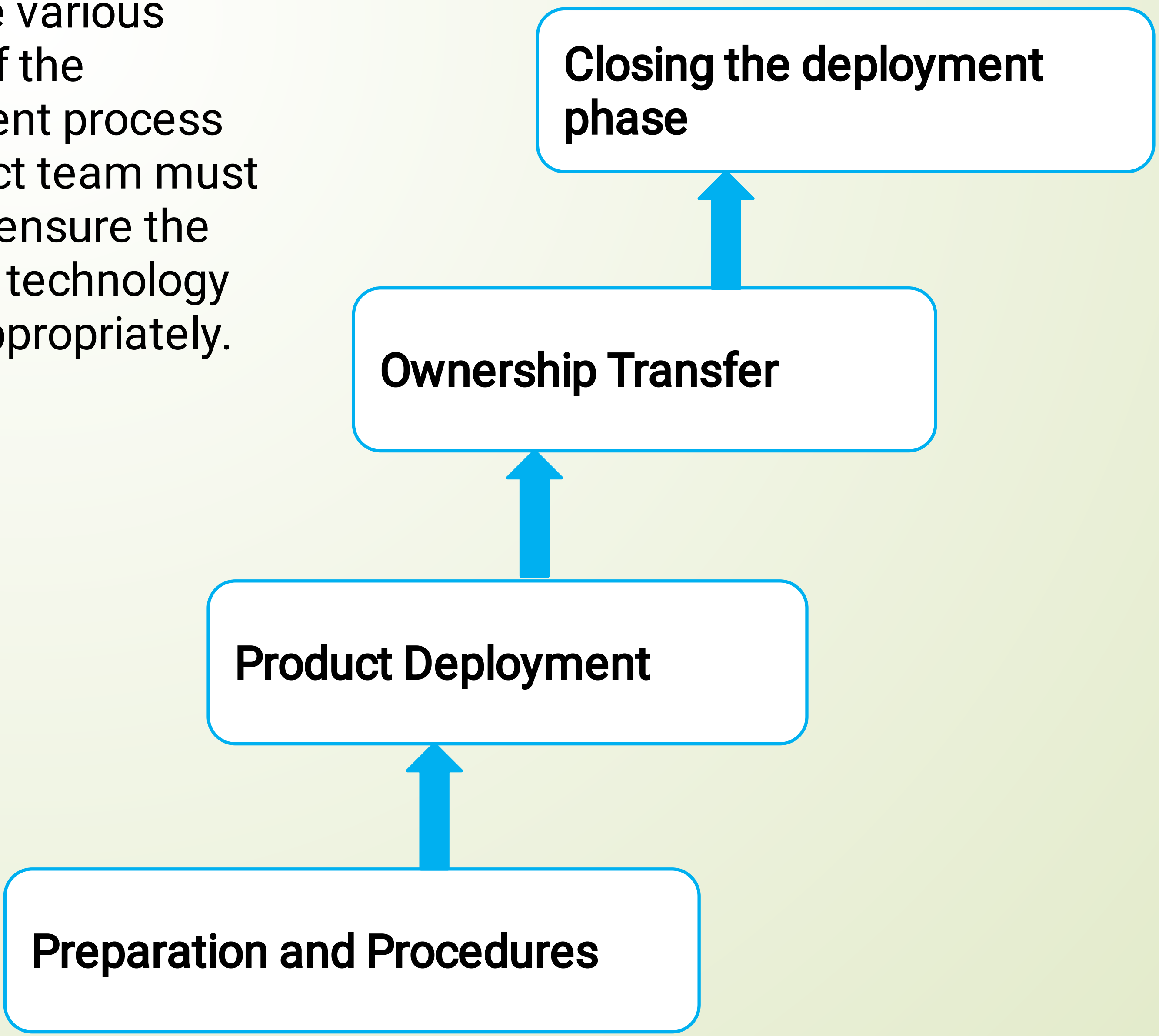
- The objective of deployment phase is to make the developed software operational in a live environment.

- A deployment in the operational environment comes only after the product is fully tested and accepted by the business in the acceptance stage of the testing phase.

- We may need to provide training to the real time users and post-deployment review is done in this phase.

Phases of Deployment

There are various phases of the deployment process the project team must follow to ensure the code and technology deploy appropriately.



Closing the deployment phase

Ownership Transfer

Product Deployment

Preparation and Procedures

Preparation and Procedures

- In the **preparation and procedures phase**, the project team installs the software and conducts another test to ensure successful installation.
- Once the installation is complete, the project team creates operating procedures, which include instructions for how the software should work in the information technology environment.
- If there are issues with system functionality, the operating instructions also provide a mitigation plan to help the end user repair the issue.

Product Deployment

- Under the **product deployment phase**, the project team implements the programming and coding to each system location.

- For example, say a company has two regional worksites in Karachi and Islamabad and over 5,000 computer systems. The deployment phase includes pushing the program and coding to each regional site and each computer system.

Ownership Transfer

- Once the program and code deploy to each system, the project team's work is complete.

- They can **transfer ownership** to the maintenance team who is now responsible for maintaining the system functionality.

- The project team transfers ownership of the program and code, along with all operating procedures, to the maintenance team.

Closing the deployment phase

- After successful deployment of software and transfer ownership of software to maintenance team, the phase of deployment is closed.

- After Deployment next phase is software evolution and maintenance.

Software Evolution and Maintenance

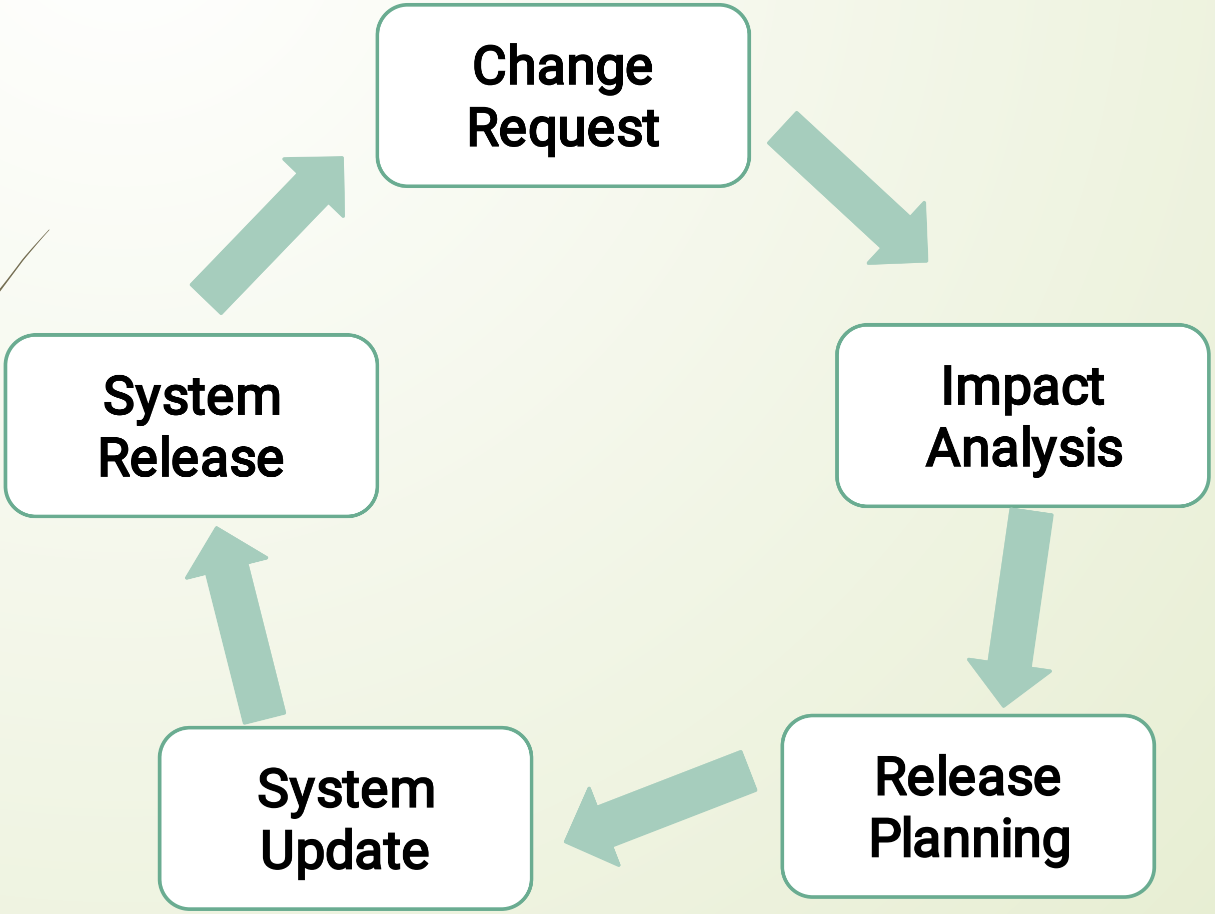
Software Evolution

- **Software Evolution** is a term which refers to the process of developing **software** initially, then timely updating it for various reasons, i.e., to add new features or to remove obsolete functionalities etc.

- The evolution process includes fundamental activities of change analysis, release planning, system implementation and releasing a system to customers.

- Why we need software evolution or update?

Software Evolution



Lehman’s Law of Software Evolution

- **Continuing Change** : This law states that any software system that represents some real-world reality undergoes continuous change or become progressively less useful in that environment.

- **Law of increase Complexity**: As an evolving program changes, its structure becomes more complex unless effective efforts are made to avoid this phenomenon.

- **Fundamental law of software evolution:** Software evolution processes are self-regulating with the distribution of product and process measures close to normal.

Lehman’s Law of Software Evolution

- **Law of conservation of organization stability:**
  Over the lifetime of a program, the rate of development of that program is approximately constant and independent of the resource devoted to system development.

- **Law of conservation of familiarity:**
  This law states that during the active lifetime of the program, changes made in the successive release are almost constant.

Software Maintenance

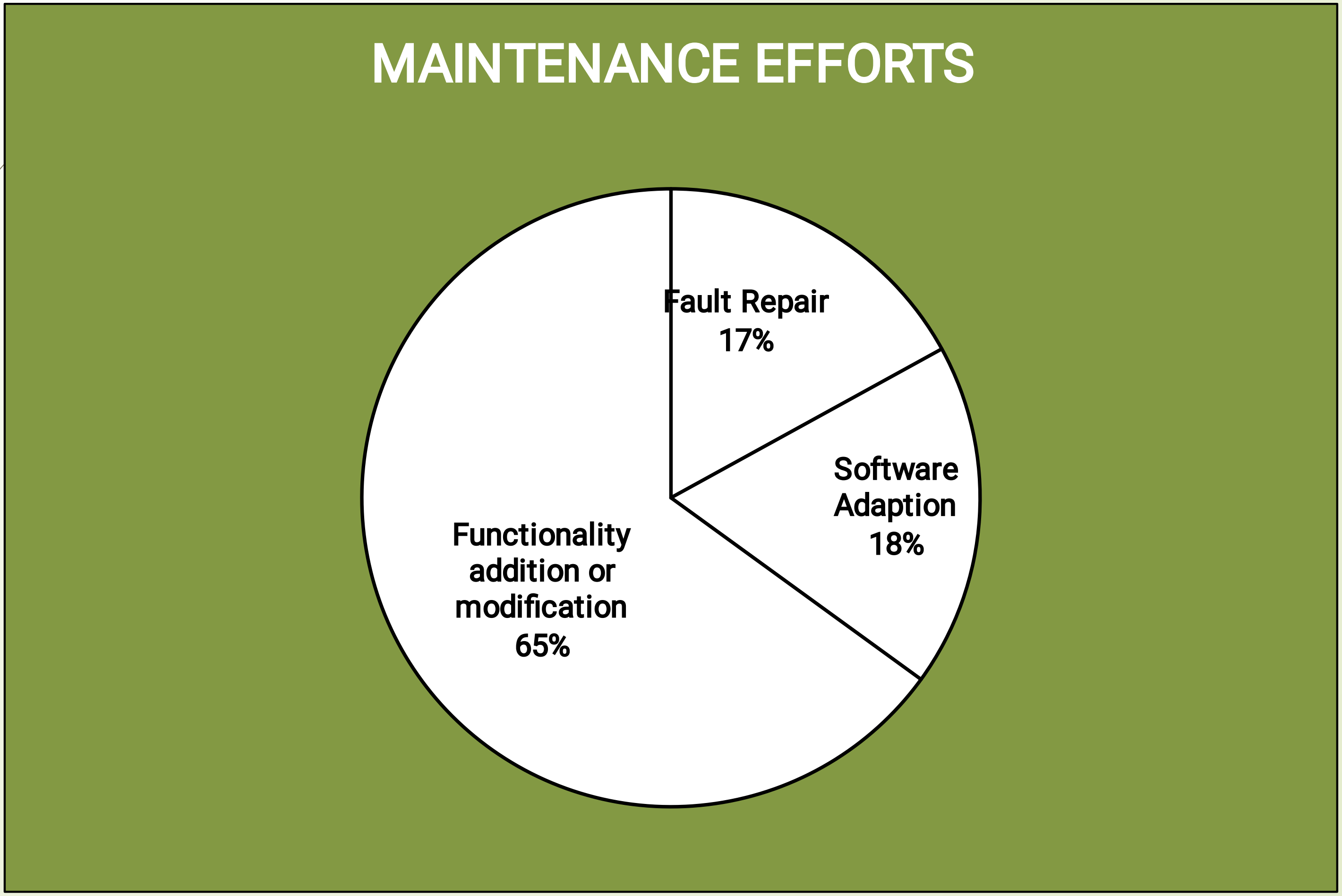
Software Maintenance

- Modifying a program after it has been put into use.
- Changes are implemented by modifying existing components and adding new components to the system.
- Maintenance can’t be avoidable.

Causes of Maintenance

- Maintenance to repair software faults.
- Maintenance to adapt software to a different operating environment such new version of OS, hardware upgrade.
- Maintenance to add or modify system’s functionality.

Distribution of Maintenance Efforts



Type of Maintenance

- **Corrective Maintenance**: Corrective maintenance deals with the repair of faults or defects found in day-today system functions. A defect can result due to errors in software design, logic and coding.

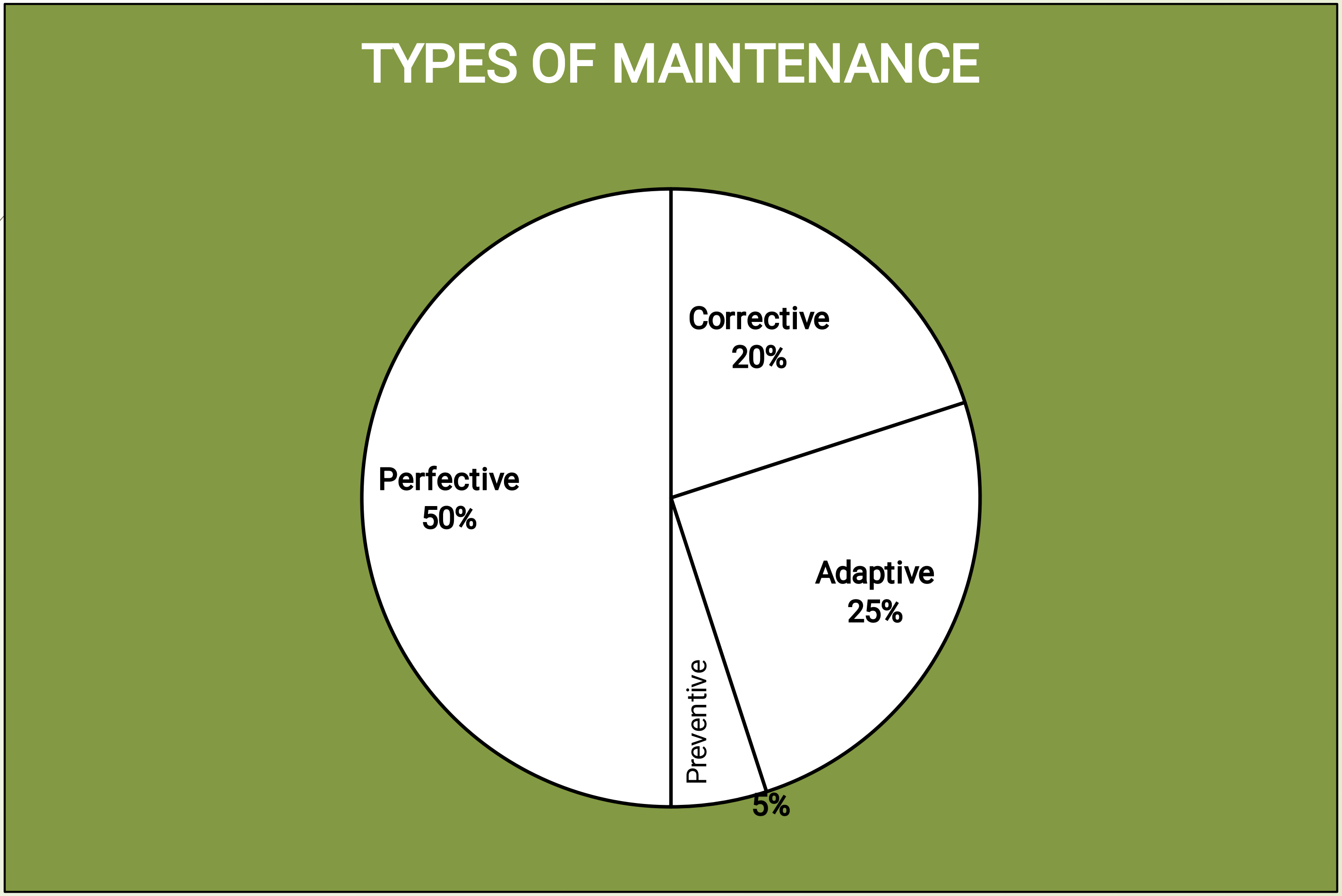
- **Adaptive Maintenance**: Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system.

Type of Maintenance

- **Perfective Maintenance**: Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults.

- **Preventive Maintenance**: Preventive maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understandability and increasing software maintainability. It comprises documentation updating, code optimization, and code restructuring.

Distribution of Type of Maintenanc



Maintenance Cost

- Usually maintenance cost is 2 to 100 times more than the development cost.
- Affected by both technical and non-technical factors.
- Ageing software can have high support costs.

Maintenance Cost Factors

- Team Stability
- Contractual responsibility.
- Staff skills
- Program age and structure

Problem Facing by Maintainers

Top Five problems

- Poor quality of documentation.
- User demands for enhancement.
- Difficulty in meeting schedule commitments.
- Turnover in user organization.
- Limited understanding, 47% of software maintenance efforts are devoted to understanding the software.