Chapter Five

Software Evolution and Maintenance

- Software development does not end when a system is delivered; instead, it continues throughout the system’s lifetime. Once deployed, a system must evolve to remain useful and relevant. This evolution is driven by **system change proposals**, which may arise from unimplemented requirements, new user needs, bug reports, or ideas from developers.

Chapter Five

Software Evolution and Maintenance

- The **software evolution process** includes several key activities:

1. **Change analysis** – studying the effect of proposed changes
2. **Release planning** – deciding which changes to implement in the next version
3. **Change implementation** – modifying specifications, design, and code
4. **System release** – delivering a new version to users

If changes are approved, a new release is planned and implemented. This process is iterative and repeats with each new set of change requests.

Chapter Five

Software Evolution and Maintenance

- **Software maintenance** refers to modifying a system after delivery. It is categorized into three main types:
- **Fault repair (Corrective maintenance):** fixing errors and bugs
- **Environmental adaptation (Adaptive maintenance):** adjusting the system to changes in hardware, operating systems, or platforms
- **Functionality addition (Perfective maintenance):** adding or improving features due to business or organizational changes

- Finally, **software reengineering** is introduced as a way to improve legacy systems. It focuses on enhancing system structure, readability, and maintainability without changing its core functionality.

Chapter Five

Software Evolution and Maintenance

1.Software maintenance is performed (**after)** system delivery.

2.Fixing bugs is known as (**corrective**) **maintenance**.

3.Adapting software to a new operating system is (**adaptive) maintenance**.

4.Software evolution is an (**iterative**) **process**.