Software maintenance
Chapter_2

أ.م.د.عباس عبد العزيز عبد الحميد

كليـة العلـوم / قسم الحاسـوب
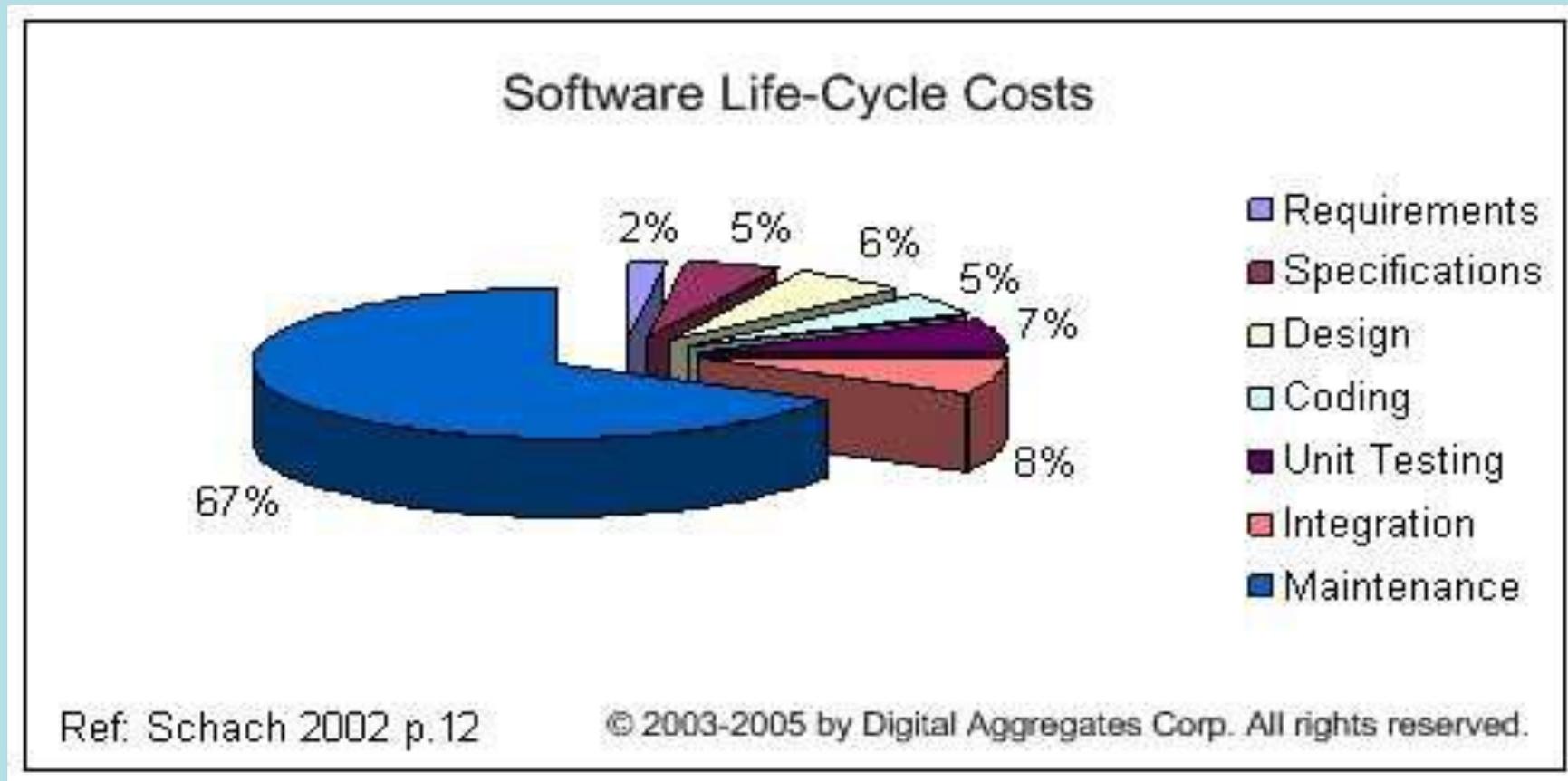
abbasabdulazeez@uomustansiriyah.edu.iq

Introduction

- Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes.

- A common perception of maintenance is that it merely involves fixing defects.

- Software maintenance is the general process of changing a system after it has been delivered.

Reasons for maintaining software

- Over a period of time software's original requirement may change to reflect the customer's needs.

- Errors undetected during software development may be found during use and require correction.

- With time new technologies are introduced such as new hardware, operating system etc.

Cost distribution of SDLC

Software Life-Cycle Costs

- Requirements — 2%
- Specifications — 5%
- Design — 6%
- Coding — 5%
- Unit Testing — 7%
- Integration — 8%
- Maintenance — 67%

Ref: Schach 2002 p.12

Software Maintenance Problems

- Most computer programs are difficult and expensive to maintain.

- Software changes are poorly designed and implemented.

- The repair and enhancement of software often injects new bugs that must later be repaired.

- Changes are often not documented.

- Changes often cause new faults in the system.

Types of software maintenance
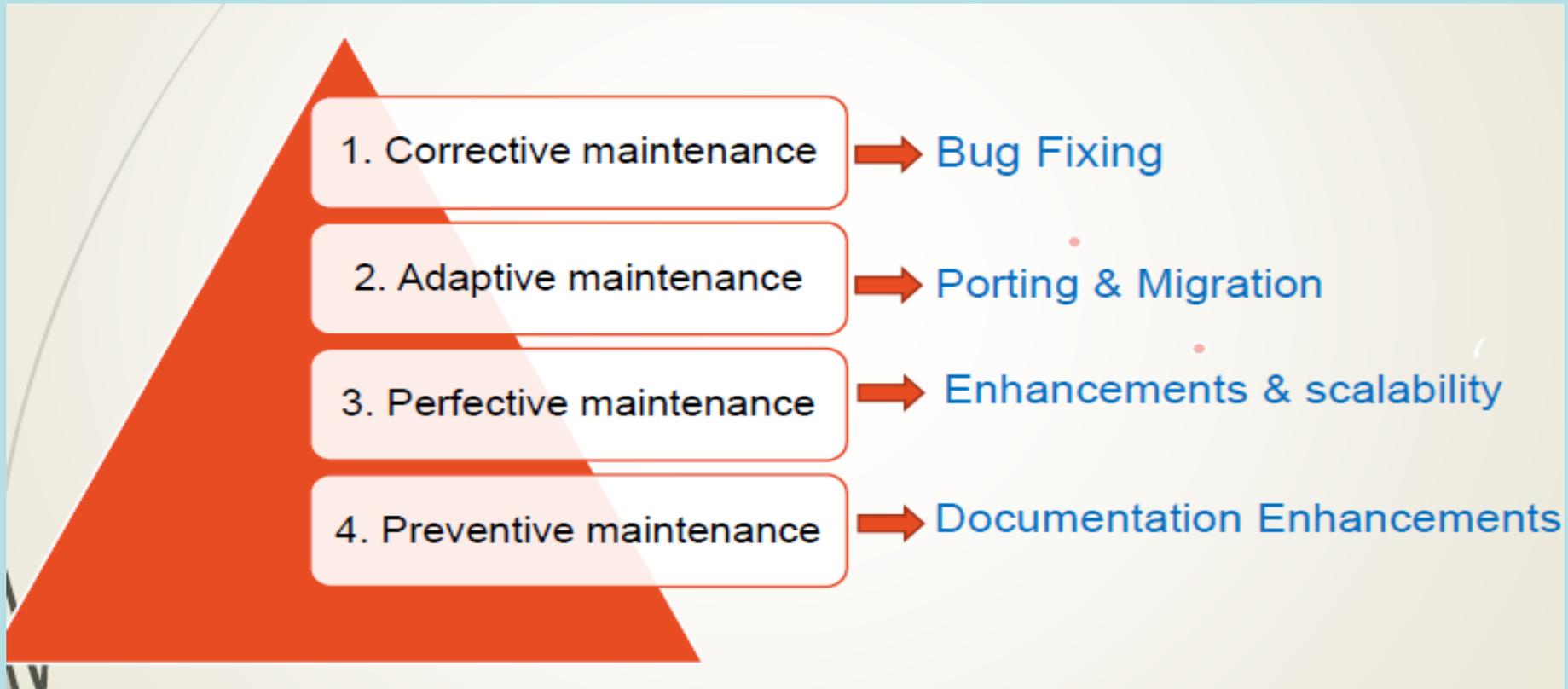
1. Corrective Maintenance

- Taking existing code and correcting a fault that causes the code to behave in some way that deviates from its documented requirements.

- Focuses on bug fixing and reporting errors fixing.

- i.e. defects generally need to be corrected either immediately or in the near future.

- Fixing a fault has 20 to 50 % chances of introducing another fault.

2. Adaptive Maintenance

- Taking existing code and adapting it to provide new features and functionality. These are typically part of a new release of the code and part of a larger development effort.

- Making changes in existing software to accommodate a changing environment

- includes all work related to how the software functions.

- i.e. relates to enhancing software functionality.

3. Perfective Maintenance

- Implementing new or changed user requirements which concern functional enhancements to the software.
- These are typically made to improve the maintainability of the code such as restructuring it to make it more easily understood or to remove ambiguities.
- includes all efforts to improve the quality of the software.
- Includes improving reliability or efficiency.

4. Preventive Maintenance

- Includes anticipating future problems and to improve the maintainability using techniques like documenting, commenting or even re-implementing some part of documenting.

- More commonly known as Software Re-engineering.

- Old system starts as a specification for new system.

- increasing the system's maintainability.

Cost of software maintenance
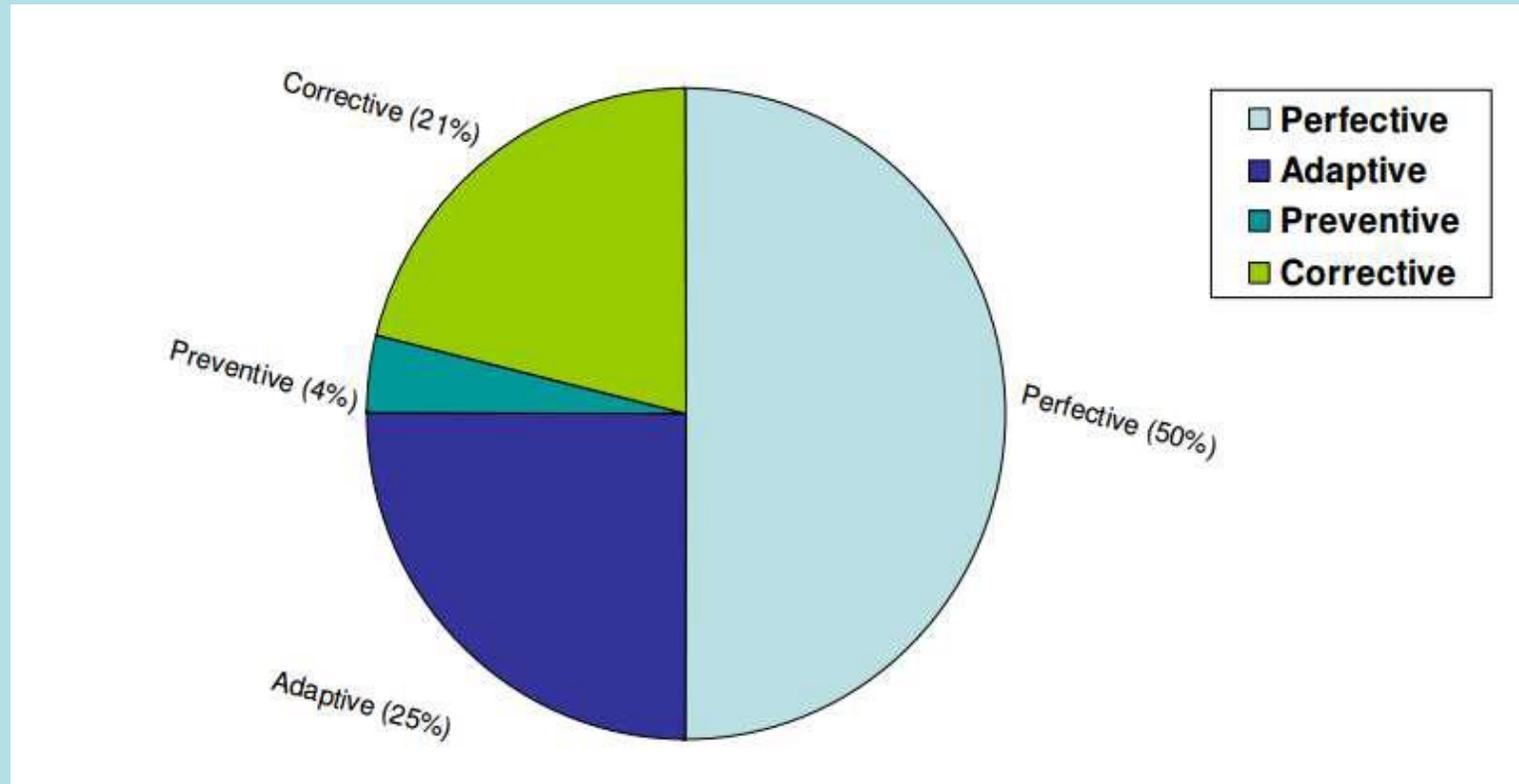
Software Maintenance models
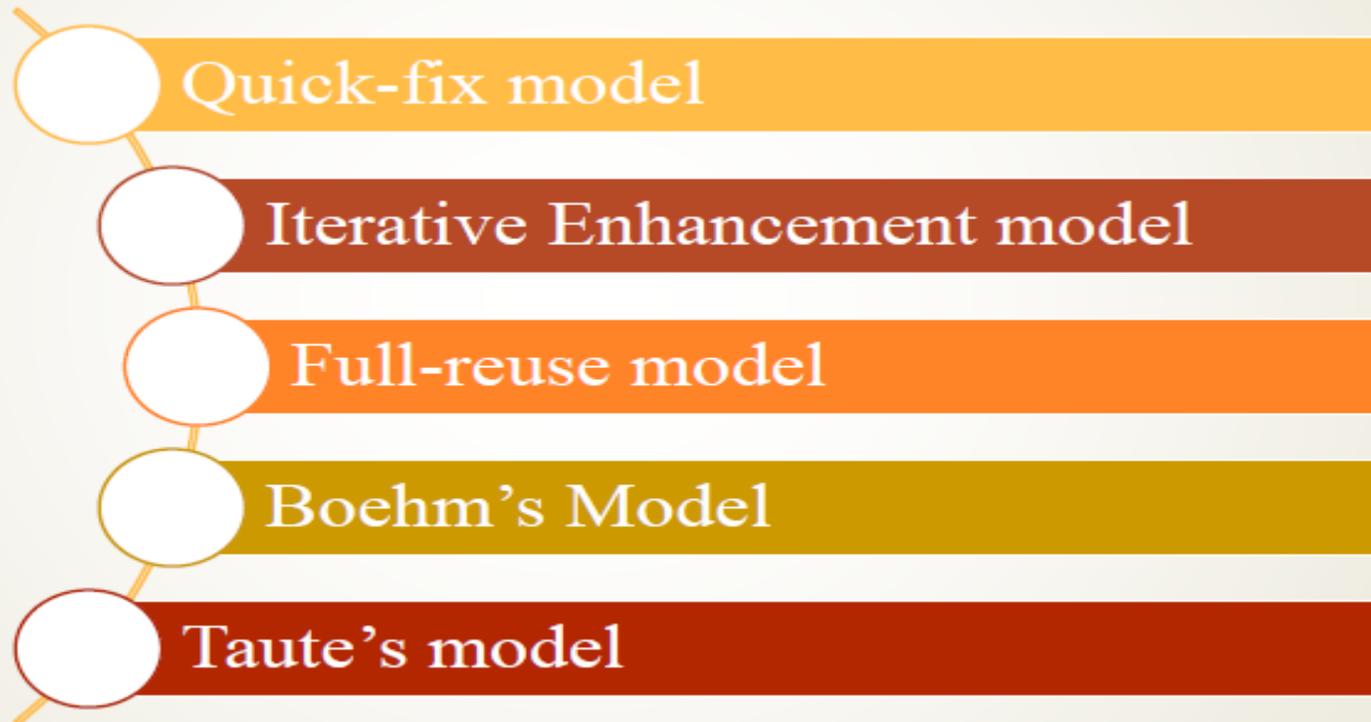
1. Quick-fix Model

- This is basically an adhoc approach to maintaining software. It is a fire fighting approach, waiting for the problem to occur and then trying to fix it as quickly as possible.

- Changes are made at code level as early as possible without anticipating future problems.

- As a result, the structure of software degrade rapidly.

- Not suitable for large software systems.
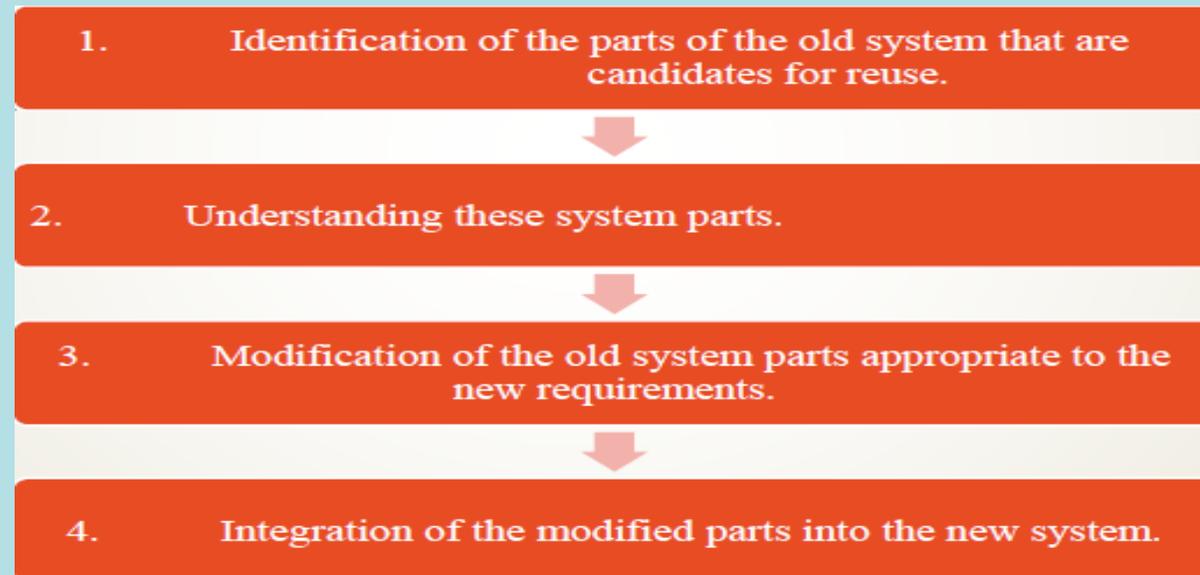
2. Iterative Enhancement Model

- Incorporates changes in the software based on the analysis of the existing system.

- Assumes complete documentation of software is available in beginning.

- Attemps to control complexity and tries to maintain good design.

- The document of each software life cycle phase i.e SRS, design document, testing document etc. are also modified.

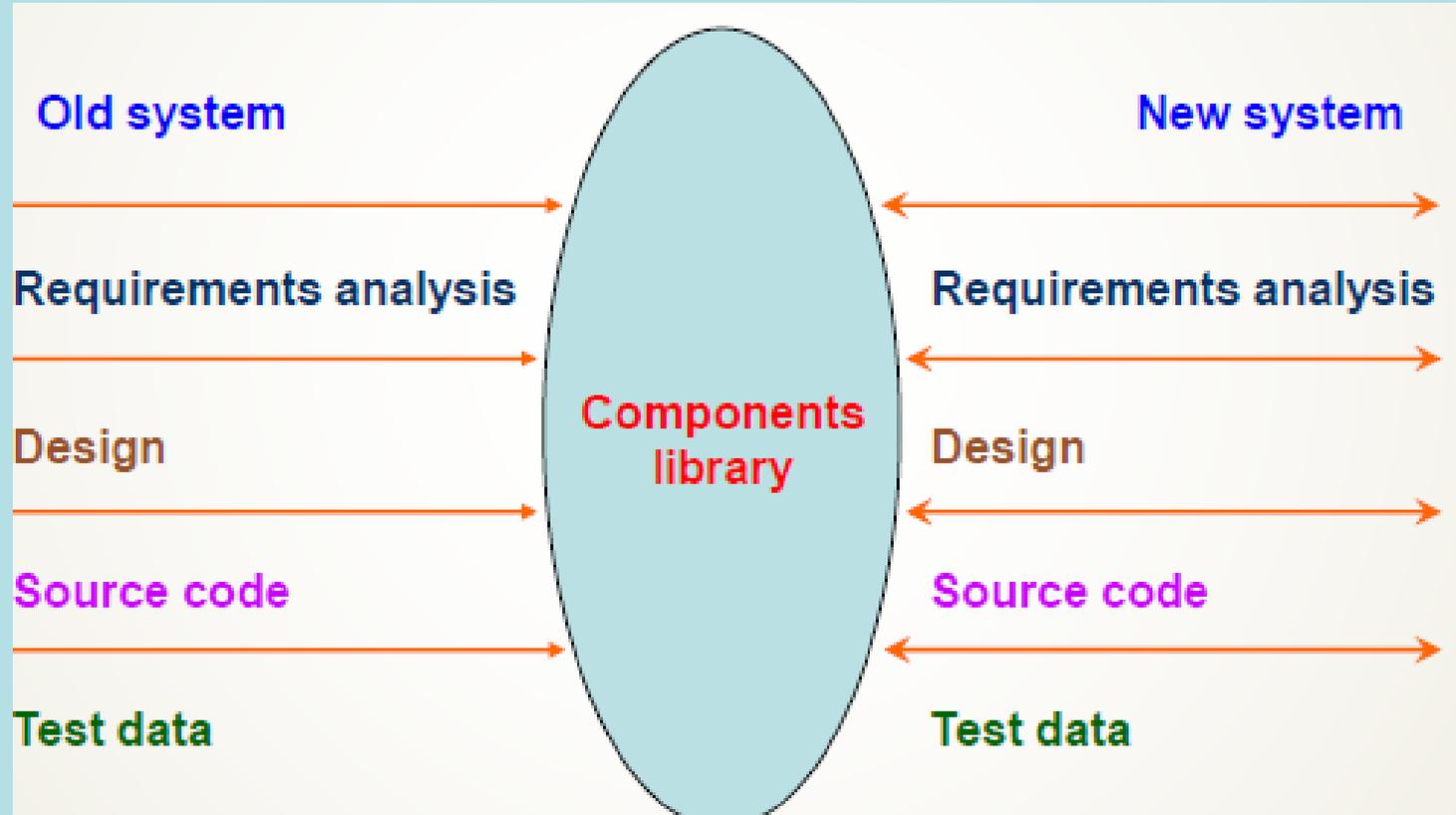# 2. Iterative Enhancement Model(cont.)

Redesign current and proposed

Characterize version

Implementation modifications

3. Full-reuse Model

- Based on reuse of existing software components.
- The reuse model has four main steps:

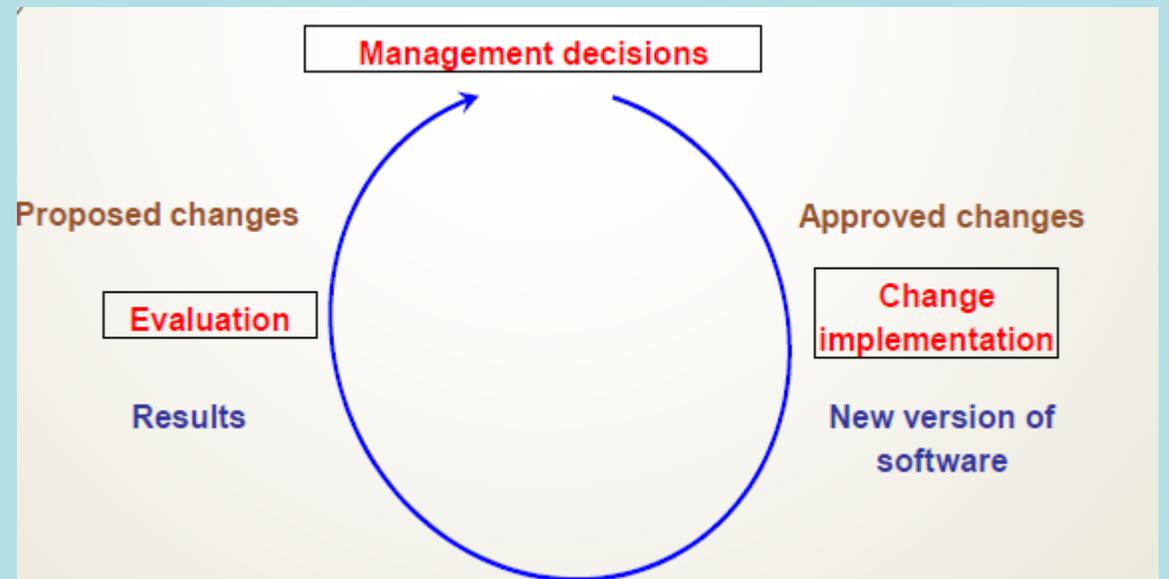|	
1.	Identification of the parts of the old system that are candidates for reuse.
2.	Understanding these system parts.
3.	Modification of the old system parts appropriate to the new requirements.
4.	Integration of the modified parts into the new system.

- Need mature reuse culture.
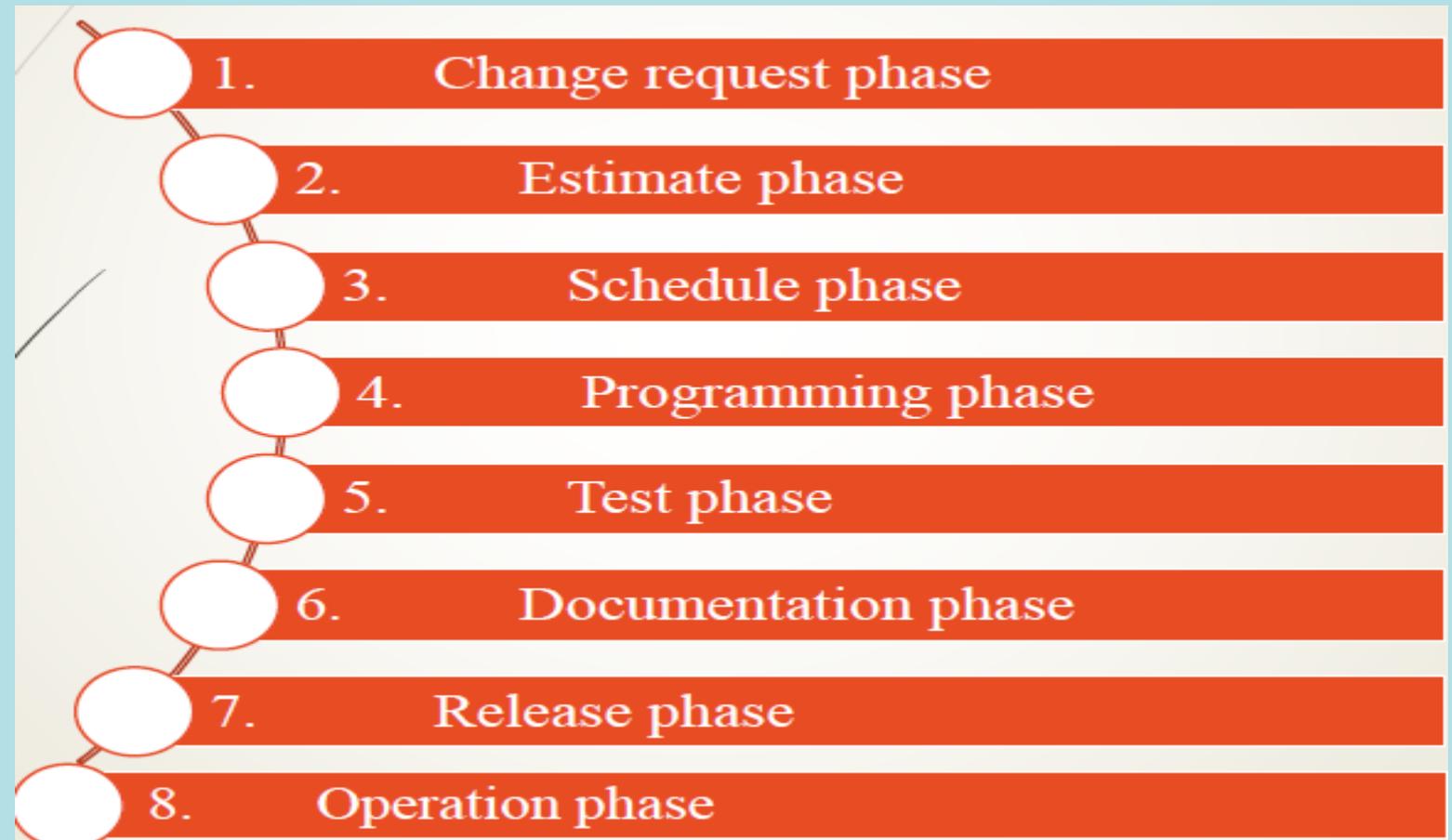
3. Full-reuse Model(cont.)

4. Boehm's Model

- Boehm proposed a model for the maintenance process based upon the economic models and principles.

- Boehm represent the maintenance process as a closed loop cycle.

- Changes are proposed first.

- Then changes are made.

Management decisions

Proposed changes

Approved changes

Evaluation

Change implementation

Results

New version of software

5. Taute Maintenance Model

- It is a typical maintenance model and has eight phases in cycle fashion.

1.	Change request phase
2.	Estimate phase
3.	Schedule phase
4.	Programming phase
5.	Test phase
6.	Documentation phase
7.	Release phase
8.	Operation phase

5. Taute Maintenance Model(cont.)