Software Maintenance
Chapter_4

أ.م.د.عباس عبد العزيز عبد الحميد

كليـة العلـوم / قسم الحاسـوب

abbasabdulazeez@uomustansiriyah.edu.iq

WHAT IS SOFTWARE MAINTENANCE

- Any work done to change the software after it is in operation this include:
  - Error corrections
  - Enhancements of capabilities
  - Deletion of obsolete capabilities
  - Optimization
- The purpose is to preserve the value of software over time

CATEGORIES of MAINTENANCE

- **Corrective Maintenance**: refers to modifications initiated by defects in the software
- **Adaptive Maintenance**: includes any work initiated as consequence of moving the software to a difference hardware or software platform-compiler, operating system or new processor.
- **Perfective Maintenance**: refers to enhancements such as making the product better, faster, smaller, better documented, cleaner structured.
- **Preventive Maintenance**: the purpose is making program easier to understand and hence facilitating future maintenance work.

PROBLEMS DURING MAINTENANCE

- Difficult to have the persons, who were constructed the program
- The program is changed by person, who did not understand it clearly
- Program listings are not structured to suppose reading for comprehension.
- The systems are maintained by the persons who are not original authors.
- There is the information gap between user and developer
- The systems are not designed for change.

POTENTIAL SOLUTIONS to MAINTENANCE

- Budget and Effort Reallocation: It is suggested that more time and resources should be invested in development specification and design of more maintainable systems rather than allocating resources to develop unmaintainable or difficult to maintain system.
- Complete Replacement of an exist System if the cost for maintaining as much as developing a new one.
- Maintenance of existing system.

MENTENANCE PROCESS

- Program Understanding: Analyzing the program in order to understand it.
- Generating Particular Maintenance Proposal to accomplish the implementation of the maintenance objective.
- Accounting for all ripple effects as a consequence of program modifications.
- Modified Program Testing to ensure the modified program has at least the same reliability level as before.

MENTENANCE MODELS

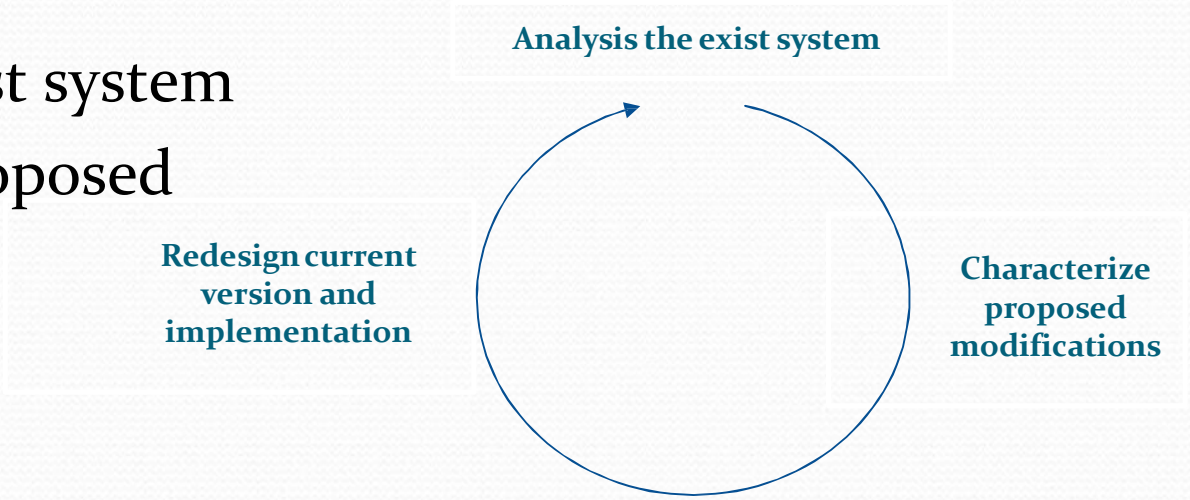
Quick-Fix Model

- The fire fighting approach
- Waiting for the problem to occur and then trying to fix it as quickly as possible.
- Fixes would be done without detailed analysis of the effects.
- In an appropriate environment it can work perfectly well e.g. The system is developed by a single person.

MENTENANCE MODELS

Iteractive Enhancement Model

- The model based on the existing of documentation.
- Three states:
  - Analysis the exist system
  - Characterize proposed modifications
  - Redesign and implementation



MENTENANCE MODELS

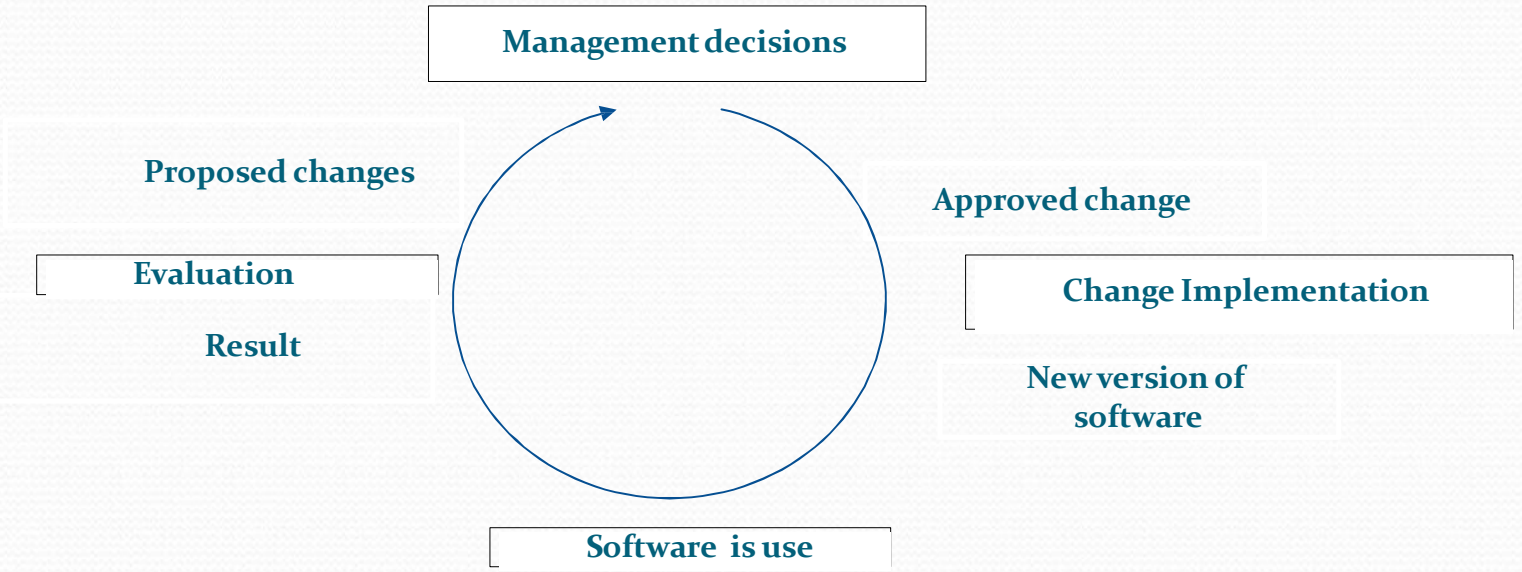
Reuse Oriented Model

- The maintenance could be viewed as an activity involving the reuse of existing program components.
  - Identification of the parts of the old system that are candidates for reuse
  - Understanding these system parts
  - Modification of the old system part appropriate to the new requirements
  - Integration of the modified parts into the new system.

MENTENANCE MODELS

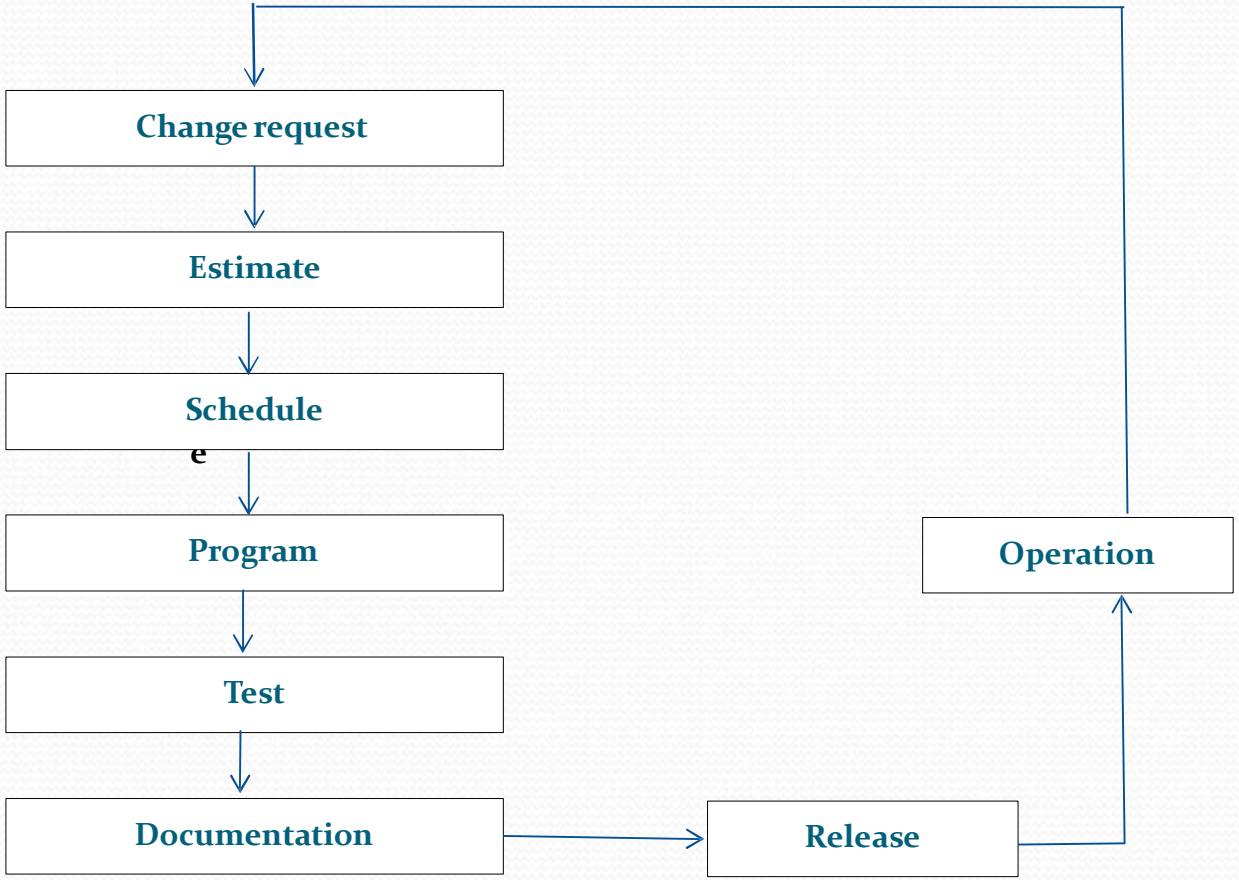
Boehm's Model

- Based on the economic models and principle



MENTENANCE MODELS

Taute's Model



ESTIMATION OF MAINTENANCE COST

- **Belady and Lahman Model:** The effort and cost can increased exponentially if the poor software development approach is use and the person or group that used the approach is no longer available to perform maintainance.
  - $M = P + Ke^{(c-d)}$
    - M: Total effort expended
    - P: productive effort that involves analysis , design, coding, testing and evaluation
    - K: an empirically determine constant
    - c: Complexity measure due to lack of good design and documentation
    - d: degree to which maintenance team is familiar with the software.

REGRESSION TESTING

- Regression testing is the process of retesting the modified parts of the software to ensure that no new errors have been introduced into previously tested code.
- Development Testing versus Regression Testing

S.no.	Development Testing	Regression Testing
1.	We create test suite and test plan	We can make use of existing test suite and test plan
2.	We test all S/W components	We test modified components or components affected by modification.
3.	Budget gives time for testing	No time (performed in crisis situation)
4.	Once on S/W product	Many times

REVERSE ENGINEERING

- Reverse engineering is the process followed in order to find difficult, unknown and hidden information about a software system. It is become important since several software products lack of documentation and are highly unstructured.
- **Scope and Stacks:** the main purpose is to recover information from existing code or any other intermediate documents, any activity that requires program understanding at any level.

REVERSE ENGINEERING (2)

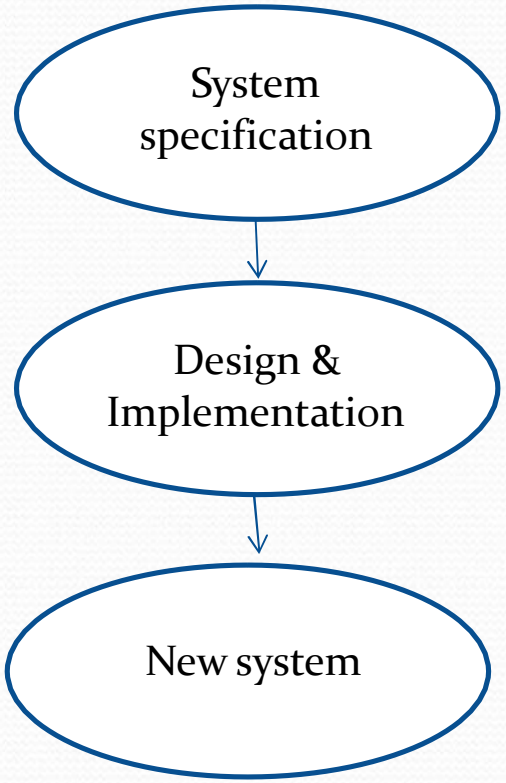
- **Application of Reverse Engineering:**
  i. Program comprehension.
  ii. Redocumentation and/or generate documentation.
  iii. Recovery of design approach and design detail at any level of abstraction.
  iv. Identify reusable components.
  v. Identify components that need restructuring.
  vi. Recovering business rules.
  vii. Understanding high level system description.

SOFTWARE RE-ENGINEERING

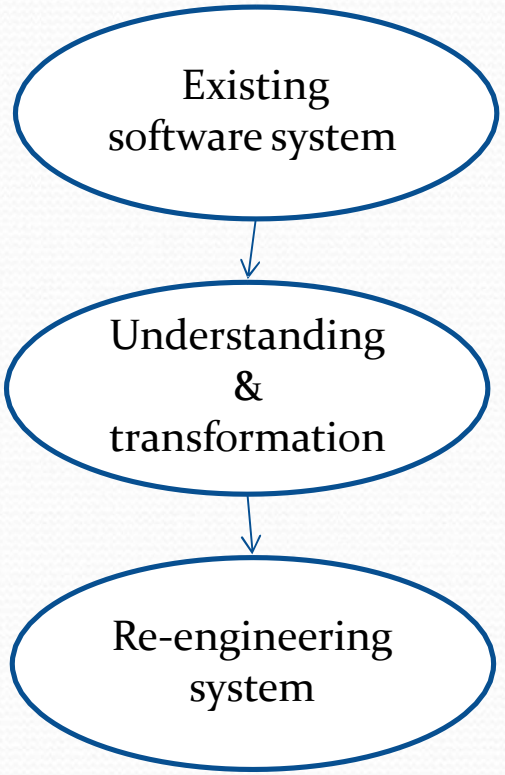
- The old systems that must still be maintained are sometimes called legacy systems.
- Software re-engineering is concerned with taking existing legacy system and re-implementing them to make them more maintainable. This maybe include re-documented and re-structured.
- Software re-engineering allow us to translate source code to new language, restructure our old code, migrate to a new platform (such as client/server), capture and then graphically display design information, and re-document poorly documented system.

SOFTWARE RE-ENGINEERING (2)

Different between re-engineering and new software development:



New software system



Re-engineering

DOCUMENTATION

- **User Document**: refers to documents which containing description of the functions of a system without reference to how these functions are implemented.
  - System overview
  - Installation guide
  - Beginner's guide
  - Reference guide
  - Enhancement guide
  - Quick reference card
  - System administration

DOCUMENTATION

- **System Document:** refers to documents which containing all facets of system, include analysis, specification, design, implementation, testing, security, error diagnosis, and recovery
  - System Rationale
  - SRS
  - Specification /design
  - Implementation
  - System test plan
  - Acceptance test plan
  - Data dictionaries