Faculty of Computers and Artificial Intelligence
BSc. In Software Engineering Program

Mid Term Exam
April 2022

***Subject***: **Software Maintenance and Evolution**

***Time***: 1 hour

**Name:** ……………………………………………………………….

---

## Answer the following Questions

### Question 1: (12 marks)

1. Define **Software Evolution**. State how it is different from **Software Maintenance**.

   Software evolution means a continuous change from lesser, simpler, or worse state to a higher or better state.

Or

“software evolution is to describe the growth characteristics of software

Fundamental differences between software maintenance and software evolution are: (any 2 are valid):

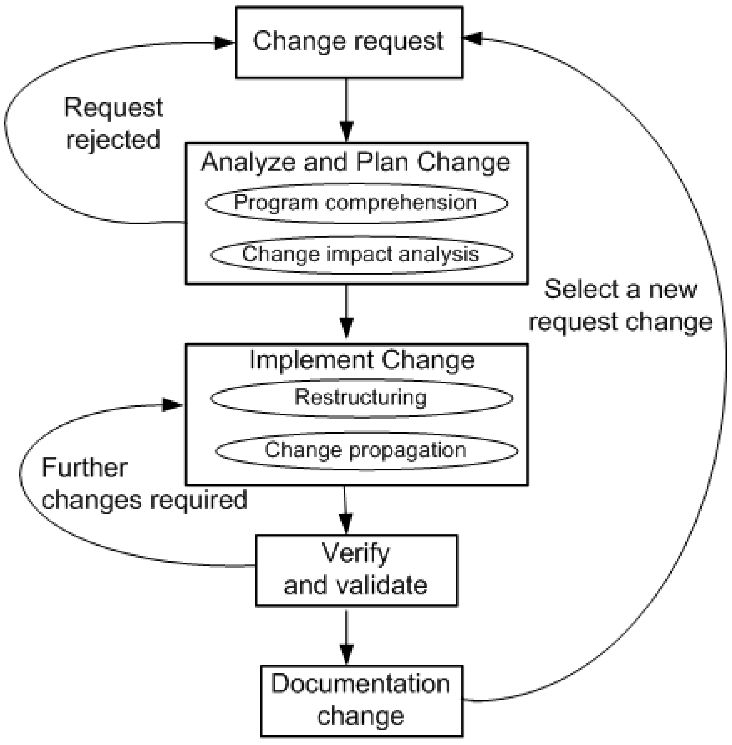
a. Software evolution means a continuous change from lesser, simpler, or worse state to a higher or better state.
b. Evolution of software systems means creating new but related designs from existing ones.
c. The objectives include supporting new functionalities, making the system perform better, and making the system run on a different operating system.
d. Software evolution is not a planned activity with fixed time rather it is the result of many maintenance activities over s period within the lifetime of the software.

2. Explain the **Change Mini Cycle** maintenance model. State when it is used.

   The change mini cycle is a software maintenance model that focus on addressing an incoming change request.

   1. Upon receiving a change request the request is analyzed in order to accept or reject it depending on it’s severity and cost of change. Via first implementing impact analysis in order to understand the effect of change on the system, then a plan for tackeling the MR is defined

2. The maintenance request is addressed based on the defined plan and code is restructured in order to fulfil the request. Next change is propagated to all affected parts of the code. Unit testing is implemented to asses change propagation.
3. Regression testing, Integration testing and system testing are implemented. Then Acceptance testing is implemented to validate the changes made.
4. The documentation is updated with the latest changes in the system. The CR is then marked as closed in the bug repository.



3. Define **Preventive Maintenance**, **Perfective Maintenance**. How do they relate to software evolution?
   - **Preventive Maintenance:** The purpose of this maintenance is to prevent so ftware from deterioration. Examples of this is to perform changes to enhance maintainability, and establish a base for making a future transition to an emerging technology.
   - **Perfective Maintenance** : The purpose of perfective maintenance is to make a variety of improvements, namely, user experience, processing efficiency, and maintainability.

- Examples of perfective maintenance are:
- the program outputs can be made more readable for better user experience; program can be modified to make it faster, thereby increasing the processing efficiency; and the program can be restructured to improve its readability, thereby increasing its maintainability.

  They are both important to reduce complexity and control software growth thus prolonging the life time of the software and keeping its value.

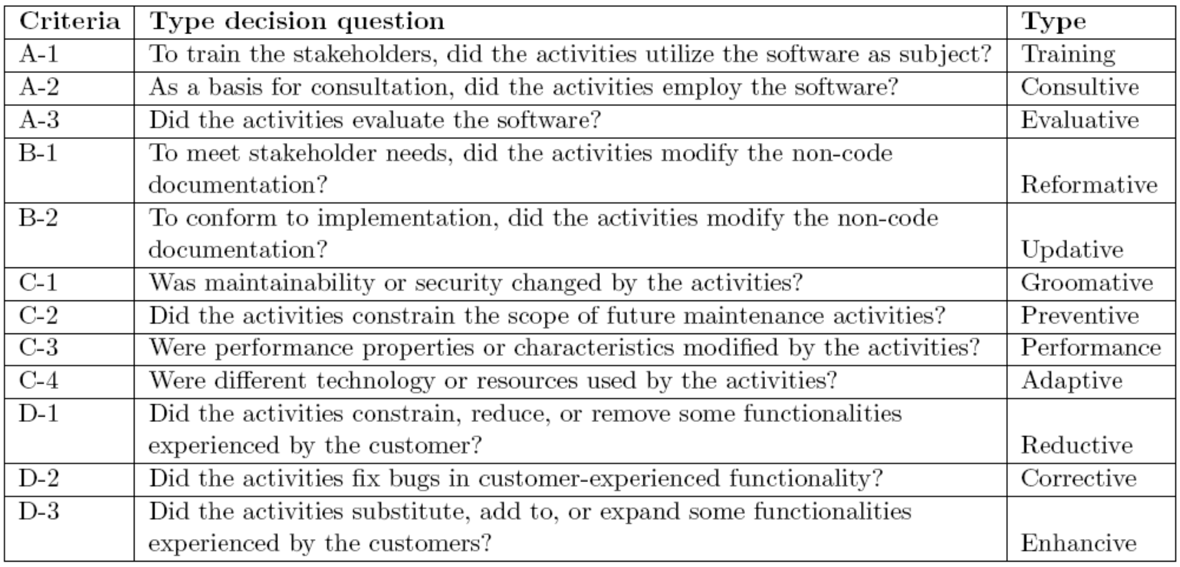
## Question 2 (6 marks)

A maintenance engineer, analyzed the documentation along with the program code, modified the program code for one component without modifying other documentation, built the rewritten component, and executed the test suite. Once finished he checked the new component into the version control, and embedded it into the production system. The end user did not observe any change.

*State the type of maintenance activity that was done according to:*

a) The **Evidence Based Maintenance** model

The type of maintenance is groomative maintenance. The system did not suffer from a bug nor did the user observe any change, rather the maintenance aimed enhancing the system qualities (performance) . Thus, modify the implementation without changing the requirements implemented by the system. This is a cleanup activity to enhance **maintainability.**



| Criteria | Type decision question | Type |
|---|---|---|
| A-1 | To train the stakeholders, did the activities utilize the software as subject? | Training |
| A-2 | As a basis for consultation, did the activities employ the software? | Consultive |
| A-3 | Did the activities evaluate the software? | Evaluative |
| B-1 | To meet stakeholder needs, did the activities modify the non-code documentation? | Reformative |
| B-2 | To conform to implementation, did the activities modify the non-code documentation? | Updative |
| C-1 | Was maintainability or security changed by the activities? | Groomative |
| C-2 | Did the activities constrain the scope of future maintenance activities? | Preventive |
| C-3 | Were performance properties or characteristics modified by the activities? | Performance |
| C-4 | Were different technology or resources used by the activities? | Adaptive |
| D-1 | Did the activities constrain, reduce, or remove some functionalities experienced by the customer? | Reductive |
| D-2 | Did the activities fix bugs in customer-experienced functionality? | Corrective |
| D-3 | Did the activities substitute, add to, or expand some functionalities experienced by the customers? | Enhancive |

Table 2.3: Summary of evidence-based types of software maintenance

b) The **Activity Based Maintenance** model

Enhancements Maintenance activities are designed to enhance the maintainability of the system.

## Question 3 [12 marks]

Select the **best** answer for the following questions:

1. When doing Code comprehension ………… is particularly helpful to track down problems in a large program
   a) Self-containment    b) Text Segments
   c) [✓ Cross Referencing]    d) Schema
2. “ The amount of new content in each successive release of a system tends to stay constant or decrease over time”. is known as
   a) Increasing complexity    [✓ b) Conservation of familiarity]
   c) Continuing growth    d) Declining quality
3. Which of the following hypotheses is a valid hypothesis formed during the early stages of “Top-Down” code reading?

   a. "This code fragment implements the *swap* idiom."

   b. "The variable *Sal* contains the *salary* of the *employee* selected by the user."

   [✓ c. "All error handling routines can be found in the file errhand.c."]

d. “The function *sort* sorts elements of a list”

4. The process of defining a mental model that represents existing code is known as?
   a) Reengineering                b) Impact analysis
   c) Cross referencing            [✓ d) Code comprehension]

5. . …………………. is a formal specification where the requester defines the change required.
   [✓ a) Change request]           b) Change assessment
   c) Change definition            d) Change impact

6. Lehman Law which states that “A system will become progressively more complex, unless work is done to explicitly reduce the complexity” is
   a) Law of self regulation       b) Law of conservation of organizational stability
   c) Law of continuing change     [✓ d) Law of increasing complexity]

7. “The creation of a new codeline from a version in an existing codeline” is known as?
   [✓ a) Branching]                b) Codeline
   c) Merging                      d) Mainline

8. The modification of the software to match changes in a continuously changing environment, falls under which category of software maintenance?
   a) Corrective                   [✓ b) Adaptive]
   c) Perfective                   d) Preventive

*Best of luck ………………. dr. Lamia Abo Zaid*