18CSC206J SOFTWARE ENGINEERING AND PROJECT MANAGEMENT
UNIT-V
SOFTWARE REUSE

1 MARKS

S.NO	QUESTIONS	LEVEL	CLO	PG. NO
1.	_____ will provide a step-by-step guide for using the product under scenarios. a. **User Training** b. User Requirements c. User Maintenance d. User Validation	1	5	203
2.	If the software has some defects, then it will take a _____ to rectify it. a. **Corrective Maintenance** b. Adaptive Maintenance c. Preventive Maintenance d. Perfective maintenance	1	5	205
3.	_____ is necessary to do so that the software product becomes reusable. a. Corrective Maintenance b. **Adaptive Maintenance** c. Preventive Maintenance d. Perfective maintenance	1	5	205
4.	_____ is needed when there is a change in the business environment, and thereby users need additional/modified functionality in the software product to do their tasks. a. Corrective Maintenance b. Adaptive Maintenance c. Preventive Maintenance d. **Perfective maintenance**	1	5	205
5.	_____ on the software product can make sure that the product will be useful even after environmental changes occur. a. Corrective Maintenance b. Adaptive Maintenance c. **Preventive Maintenance** d. Perfective maintenance	1	5	205
6.	A _____analysis can be done, to see if it is more profitable to conduct a Maintenance program on the software or keep using it as it is. a. **Profit/Loss** b. Test c. Maintenance d. Corrective	1	5	206
7.	In which model there is no planning involved in the whole process and is mostly an ad hoc Approach?	1	5	207

a. **Quick Fix Model**
b. Boehm's Model
c. Osborne's Model
d. Iterative Enhancement Model

8. _____ Model is based on economic models and often involves calculating ROI, for any planned maintenance. 1 5 207
a. Quick Fix Model
b. **Boehm's Model**
c. Osborne's Model
d. Iterative Enhancement Model

9. A quality assurance plan should accompany the maintenance plan in which model? 1 5 206
a. Quick Fix Model
b. Boehm's Model
c. **Osborne's Model**
d. Iterative Enhancement Model

10. _____ model is based on the similar concept of iterative software development. 1 5 207
a. Quick Fix Model
b. Boehm's Model
c. Osborne's Model
d. **Iterative Enhancement Model**

11. _____ type of process is adopted for component-based software products. 1 5 207
a. Quick Fix Model
b. Boehm's Model
c. Osborne's Model
d. **Reuse Oriented Model**

12. _____ is also known as reuse engineering. 1 5 208
a. Reverse Engineering
b. **Reengineering**
c. Forward Engineering
d. Both a & b

13. _____ technique is most useful when nonexistent or sketchy documentationis available for the software product 1 5 208
a. **Reverse Engineering**
b. Reengineering
c. Forward Engineering
d. Both a & b

14. _____ the opposite of reverse engineering. 1 5 209
a. Reverse Engineering
b. Reengineering
c. **Forward Engineering**
d. Both a & b

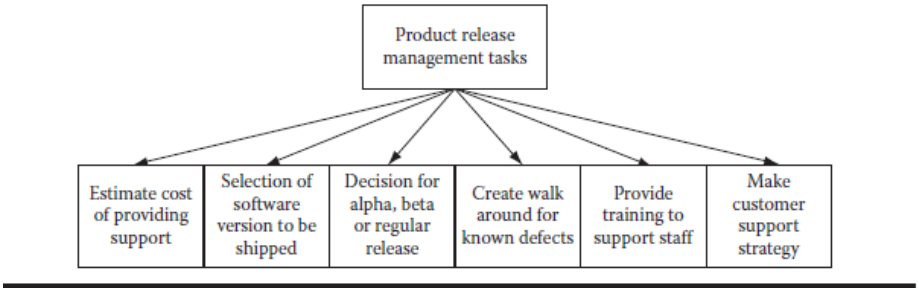
15. _____ is also known as Renovation and Reclamation 1 5 209

a. Reverse Engineering
b. Reengineering
c. **Forward Engineering**
d. Both a & b

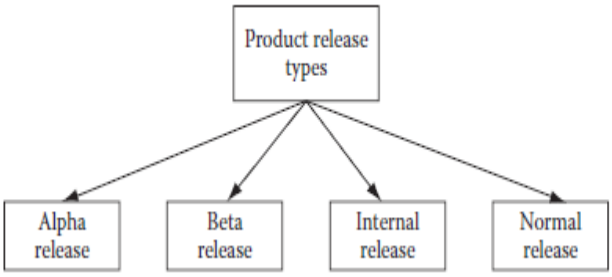
4 MARKS:

1. Explain about Product Release Management. L2 CLO5

Project teams working for software product vendors struggle to keep pace with release of the software product. There is pressure from the market to launch new versions by certain dates. New features are to be added, porting the product to new platforms, old features are to be enhanced, existing bugs are to be removed, and yet it has to meet the deadline. It is a constant struggle that calls for good product release strategies. Depending on the situation, the project manager may need to convince the management to cut short some of the product features to meet the deadline as well as meet quality standards.



2. Explain the idea list for software product release. L2 CLO5



A product release is the process of launching a new product for a specific market or user base. In software development, a product release is sometimes done with a beta version so that core developers/users can assist with debugging and feedback prior to the release of the actual software.

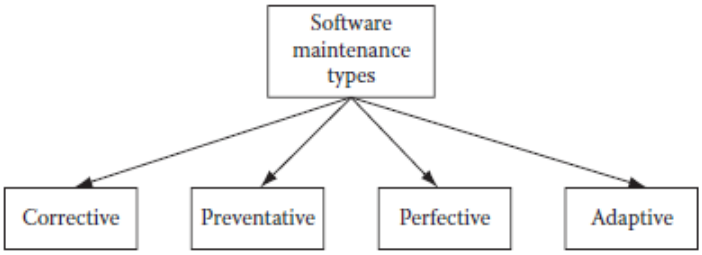
3. Analyze the reasons for maintenance in software products. L2 CLO5

1.Technology obsolescence : The software platform (operating system, medium of user, interface) or the hardware platform on which the software product runs gets obsolete.

2. Software defects: There are major software defects in the product and it is difficult to operate. For this reason, a software patch may be needed to be applied so that these defects are removed.

3. Change in user requirements: The business organization that was using the software product has seen a change in business transactions or business workflows that are not supported by the software product.

4. List out the software maintenance types. L2 CLO5



Corrective: after thorough reviews and testing, the software product contains many defects when it goes into production. These defects are uncovered as users start using the application. They are logged with the support staff and after a sizable number of errors are detected, the software vendor instructs his maintenance team to create a patch to rectify them.

Adaptive: The operating environment in which a software product runs in operation includes the hardware and software platform as well as the interfaces for human and other machine interactions. If any of these change over time, it becomes difficult to run the software product. In such cases, it becomes necessary to do adaptive maintenance so that the software product becomes reusable.

Perfective: This kind of maintenance is needed when there is a change in the business environment, and thereby users need additional/modified functionality in the software product to do their tasks. A business workflow may have changed, a business transaction may have changed, or an altogether new business transaction was represented in the software product.

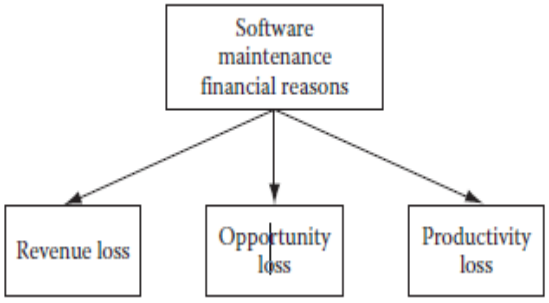
Preventive: Generally after a lapse of time, there are likely changes in business or operative environment, or there may be changes in hardware/software environment. These changes are bound to occur and they affect the way the software product operates. Many of these changes can be perceived in advance.

5. Classify the financial reasons for software maintenance. L2 CLO5

1. Loss in business revenue: It may happen that business transactions are faulty and thus the business may lose revenue.

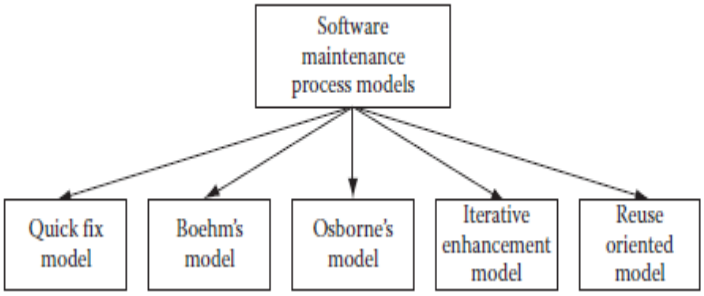
2. Opportunity loss: Sometimes there could be some business opportunity in the marketplace, but due to some software problems it could not be availed.

3. Productivity loss: If the software product becomes difficult to operate due to many walk arounds or lengthy processing then productivity will become lower for business personnel.



6. Sketch the software maintenance model. L1 CLO5

Software maintenance process models have been defined and Some of the popular ones include the quick fix model, Boehm's model, Osborne's model, iterative enhancement model, and reuse oriented model



7. Demonstrate the Quick Fix Model. L3 CLO5

This is the simplest of maintenance models; whenever any defects with the software products are found they are immediately fixed. There is no planning involved in the whole process and it is mostly an ad hoc approach.

8. Examine Reuse Oriented model. L2 CLO5

This type of process is adopted for component-based software products. For fixing lany defects, existing components are analyzed and then the appropriate changes are made.

9. Define Iterative Enhancement Model. L1 CLO5

This model is based on the similar concept of iterative software development. All software defects and change requests are logged and then a small set from this list is taken for making fixes. This set is prepared based on the priority of changes required. High priority fixes are done before low priority fixes.

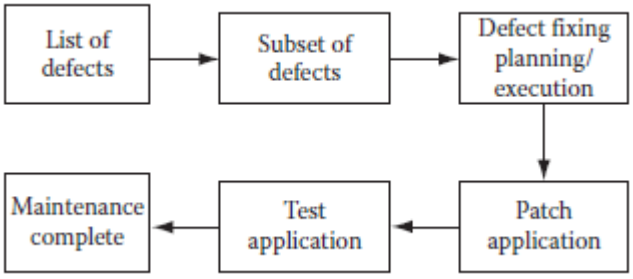
10. Compare and contrast Boehm's Model Osborne's Model L2 CLO5

Boehm's model: Boehm's model is based on economic models and often involves calculating ROI, for any planned maintenance. If ROI turns out to be good, then it is carried out or else it is dropped.

Osborne's model: Osborne realized that difficulties in carrying out maintenance work are due to gaps in communication. He proposed four steps to prevent this situation. He stated that a maintenance plan should include all change requests in the form of maintenance requirements. A quality assurance plan should accompany the maintenance plan. Metrics should be developed to measure and assess quality of work carried out during maintenance.

11. Design Software Maintenance Life Cycle. L2 CLO5

Software maintenance life cycle, testing is a crucial phase. This phase also consumes a lot of time and effort. But the value addition in all this effort and time spent helps in reducing defects, which in the long run is a much cheaper alternative compared to no testing/cursory testing and later spending money in providing support.



12. Write about software maintenance engineering techniques. L2 CLO5

Maintenance of software products sometimes becomes a tough proposition. There is no proper documentation that can be used for understanding how the product is designed and constructed. Sometimes there is no documentation at all. Even if documentation is there, it is not up to date. This out-of-date documentation is not of much use for any maintenance work. Sometimes even if the documentation is up to

date, the maintenance work is difficult due to dirty design or construction work. All these situations call for some specific techniques for maintenance work depending on the situation. Some of the common maintenance techniques include reengineering, reverse engineering, and forward engineering.

13. Differentiate Reverse Engineering and Forward Engineering. L2 CLO5

Reverse engineering technique is most useful when nonexistent or sketchy documentation is available for the software product. Due to unavailability of documentation, there is no information as to what the design is and how the product is constructed. In such a situation, it is almost impossible to do any modification in the source code for any maintenance work.

In such cases, the reverse engineering technique is adopted. Using this technique, similar components or product parts are constructed as compared to existing product components/parts. This way the software product functionality is changed as the new constructed parts will have the desired functionality.

Forward engineering is just the opposite of reverse engineering. In this situation, we have ample documentation about the existing product. Due to new customer needs, the existing product needs to be extended so that the new needs can be fulfilled. All new extended development is based on the existing design and construction methods and will be made for the same hardware/software platform.

12 Marks:

1.	Explain in detail about Project Release Management.	L2	CLO5
2.	List out the software maintenance types and describe in detail.	L2	CLO5
3.	Distinguish Briefly about Software maintenance process model.	L2	CLO5
4.	Illustrate about Software maintenance techniques in detail	L2	CLO5
5.	Discuss in detail about Software maintenance Life cycle.	L2	CLO5
6.	Write down the steps for Sofware product implementation.	L2	CLO5