

①

Ans to Q. 1(a)

Software Maintenance: Software Maintenance refers to ^{modification} ~~to~~ of a software right after delivery to correct faults, to improve performance or other attributes or to adapt the product to a modified environment.

CMM Models: ~~The~~ The CMM model has a total of 5 levels. They are:

① Initial: in this stage few processes are defined & Success ~~of~~ depends on an individual or team

② Repeatable: in this stage basic processes are established & Success can be repeated on projects with similar application

③ Defined: in this level processes are well defined & standardized. All the projects use a refined and approved version of the standard process.

②

① Managed :- in this level, detailed measures are collected and a better understanding of the process is achieved.

⑤ Optimizing :- in this stage, the quantitative feedback gathered in the last stage is evaluated & used from the processes.

Ans to Q. 1(b)

Evolution of Software through maintenance activities:

The software will evolve through preventive, adaptive, ~~correcting~~ corrective & perfective maintenance. The corrective maintenance will play a significant role here as the software ~~addresses~~ addresses a good amount of bug fixes.

Evolution of Software through Software Maintenance Lehman's

Laws :- The following laws will be applicable for the software's evolution:

③

① Continuing change: The software needs to be continuously adapted in order to keep up with the developing world.

2. low conservation of organisational stability: average work rate ^{XYZ} ~~of~~ ^{an} organi for "abc" software will ~~be~~ remain constant more or less.

3 - Continuing growth: The system's functional capability should increase in a steady fashion over a certain period of time.

4. declining quality: if abc software isn't rigorously adapted to meet the standards of the industry, its quality and value will gradually decline.

⑤ feedback system: the processes in "abc" software are multi-loop multi feedback system.

④

Ans to Q. 10

The process that was previously used for 'abc' software was Boehm's Model. Because the company is working on 'abc' software based on a fixed set of changes approved by the team.

Osborne's Model: if we use the Osborne's Model instead of Boehm's the following changes can be visible:

- ① In Boehm's model, the company only focused on the change requests. Osborne's Model will allow them to focus on the maintenance requirements as well.
- ② Boehm's model did not have any specific focus on the quality of the software. Osborne's Model makes up for that.
- ③ Osborne's model verifies the maintenance requirements.

⑤

are met accordingly. It's more efficient than Boehm's model.

④ Osborne's Model also ensures performance reviews

Ans to Q. 1 (D)

The maintenance framework will conduct the maintenance process in the following ways:

① User requirements: a the ~~the~~ user of "abc" ~~is~~ software may request for a refinement of an existing ~~for~~ function (progressive) or better documentation of the process (regressive)

② Environment: in the Operating Environment,

XYZ company can suggest hardware ~~is~~ & software innovations in the organizational environment

⑥

there can be a change in policies, analysis of market requirement, ~~change request~~ a variation in coding etc.

③ Maintenance Process : in the maintenance process there can be capturing change requests & variation in coding.

④ Software product : software product discusses quality of documentation, malleability of program etc.

⑤ Maintenance Personnel : XYZ also needs to think about staff turnover, domain expertise etc in terms of maintenance personnel.

④ ⑦

Ans to Q. 2(a)

They can extract information on ~~the~~ the legacy system using ~~the~~ techniques of reverse engineering.

The level of abstraction they need to adjust to are:

① Redocumentation: ~~Red~~ Redocumentation can help in

- ① creating ~~old~~ alternate perspectives
- ② improve current documentation
- ③ accommodate for modification etc.

② Design Recovery: design recovery ~~can~~ can help in identifying the overall hierarchy, design patterns etc of the code. This makes things easier for a new programmer to work on code.

③ Specification Recovery: if there are drastic changes in between the previous & the current version of the code, design recovery won't help. In cases like these we need to go to root & recover the SRS for the software.

~~first~~

Ans to Q. 2 (b)

~~X A X~~ "XYZ" company can use the following supporting techniques:

Forward Engineering: forward engineering refers to the traditional approach where we have the requirement analysis & use that to implement the system.

9

170092052

Restructuring: as the software is introduced to more features, the code keeps growing in size & complexity. Restructuring helps in maintaining that. The following techniques are used for restructuring.

① Control flow driven: in this approach, we try to understand the control flow of the source code very clearly & restructure accordingly.

② efficiency driven: this involves a function or an algorithm to reduce its time & space complexity.

③ Adaption-driven: adaption-driven ~~not~~ restructuring refers to changes in coding convention.

⑩

Re-engineering: it's the process of examining & altering a target system to ~~acomate~~ accomodate desired specifications modifications.

Am to Q.3 ①

To use the existing components of the software, & "XYZ" company will ~~ex~~ have to use different techniques of code reuse.

The targets of Reuse: They need to focus on the following targets or resources:

① Process: process refers to the application of a given methodology for a different problem. ~~They refer~~ i.e. formal methods

11

17/01/2022

② Personnel : reuse of personnel refers to the involved people using their knowledge from a ~~pre~~ previous project that was similar to the current ones. i.e. Lesson learned.

③ Product : products reuse refers to the usage of ~~pre~~ previous documents, code etc to make things easier in the current project. i.e. Data, Design, Program.

Ans to Q. 3(b)

The following components can be used

① Reusable Components : Reusable components refers to components of a project that was previously in a different but can now be used in the current project with little or no modification.

12

17/09/2022

The characteristics are:

- ① Generality
- ② Cohesion vs Coupling
- ③ Interoperability
- ④ Interaction
- ⑤ Uniformity
- ⑥

① Generality: solutions that are used in a more generic approach can be modified & used for specific cases

② Cohesion vs coupling: reuse to reusable components introduce high cohesion & low coupling.

③ Interoperability: allows components to make use of remote services.

(13)

Interaction: more use of functions rather than source code lines.

Uniformity: uniformity gives better control over all the components of the software.

Problems with Reusable component:

① Granularity: sometimes reusable components can lead to ~~long~~ lots of files with ~~a~~ very few codes.

② Classification: it becomes complex at times to ~~see~~ maintain the folder structure of the project.

③ Less flexibility: ready made components often don't allow much room for ~~the~~ necessary changes.

Reverse Engineering: reverse engineering allows us to extract components from existing software. They can also be used in program reuse.

Ans to Q. 3(c)

Generic Process Model: the generic process model is a part of the reuse process models. The process can be implemented using the following steps.

① understanding problem: in this step, we understand the given problem & come up with a feasible solution to it.

② Reconfiguration: The solution structure is then reconfigured to maximize reuse in the current stage.

③ prepare reusable components : at this step the reusable

components are prepared & integration testing begins.

④ Integration : at this stage, we have to integrate

all the components into the main code.

⑤ Evaluation : in this phase, the experience

from the previous steps is used to evaluate the

reusability of the ~~per~~ ~~cor~~ ~~re~~ components ~~integrated~~

integrated so far.

Why the implementation will perform worse in this case :

The company 'XYZ' is a CMM level 2 company, with limited resources & maturity. The accommodating process model may not work well in their case.

Since they are a company with lower budget & resources, accommodating necessary tools on other ~~things~~ for the process model will be difficult for them. Also, they may not have a proper mechanism ~~is~~ in place to implement the ~~the~~ steps properly. They may also not have enough ~~to~~ skilled personnel ~~to~~ who can conduct the maintenance process properly. They might end up falling into the run-maintenance vicious cycle. In short, the company does not have the maturity level to conduct the process model properly.