**COMPUTER ENGINEERING DEPARTMENT**

**SUBJECT: SOFTWARE ENGINEERING**

**COURSE: T.E.** **YEAR: 2020-2021** **SEMESTER: VI**
**DEPT: COMPUTER ENGINEERING**
**SUBJECT CODE: CSC601** **EXAMINATION DATE: 02/06/2021**

==========

# SOFTWARE ENGINEERING
# ANSWER SHEET

**NAME** : AMEY MAHENDRA THAKUR

**SEAT NO.** : 61021145

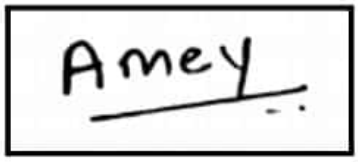
**EXAM** : SEMESTER VI

**SUBJECT** : SOFTWARE ENGINEERING

**DATE** : 02-06-2021

**DAY** : WEDNESDAY

**STUDENT SIGNATURE:**



Q 3.

A]

Risk can be quantified.

- Risk Quantification is a process to evaluate identified risks to produce data that can be used in deciding a response to corresponding risks. It is second step of project risk management. according to standards so risk can be quantified.

Risk Management

- It is the process of identifying, accessing and controlling threats to an organizations capital and earning..
- These threats or risks could stem from a wide variety of sources including financial uncertainity, legal liabilities, natural disasters etc

Steps in Risk Management:

① Risk identification:

- It can be done by identifying the known and predictable risk

② Risk Analysis

- Risk can be analyzed by accessing consequences of problem associated with risk

③ Risk Planning:

- Making plan to address the risk either by avoiding or minimizing the effect.

02-06-2021 STUDENT SIGNATURE: Amey

Q3

B] COCOMO - II model

COCOMO - II Model

- COCOMO - II is the revised version of the original COCOMO (Constructive Cost Model) and is developed at University of Southern California.
- It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.
- It consists of three sub-models.

| | | |
|---|---|---|
| | Application generators and composition aids | |
| End User Programming | Application Composition | Infrastructure |
| | System Integration | |

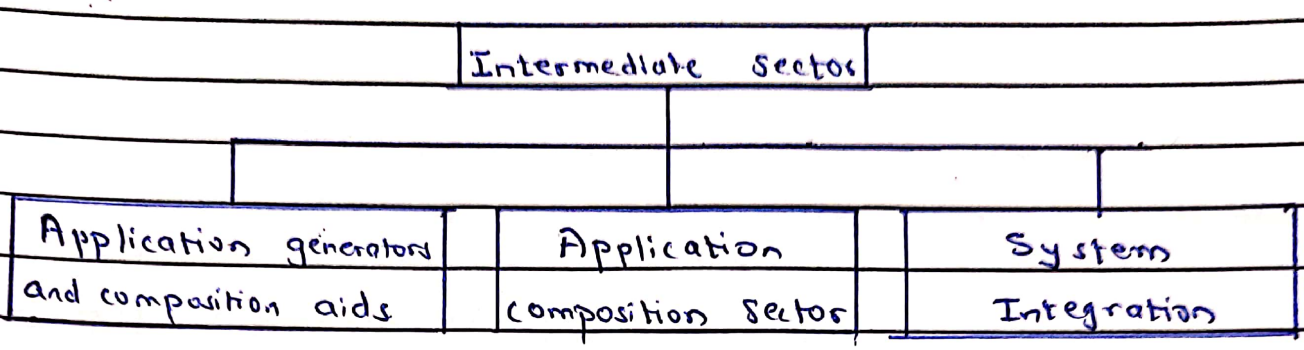
① End User Programming:

- Application generators are used in this sub-model.
- End user write the code by using these application generators.
- Example: Spreadsheets, report generator, etc.

02-06-2021

STUDENT SIGNATURE: Amey

② Intermediate Sector:




① Application generators and composition aids:
- This category will create largely prepackaged capabilities for user programming.
- Their product will have many reusable components.
- Typical firms operating in this sector are Microsoft, Lotus, Oracle, IBM, Borland, Novell.

② Application Composition Sector:
- This category is too diversified and to be handled by prepackaged solutions.
- It includes GUI, Databases, domain specific components such as financial, medical or industrial process control packages.

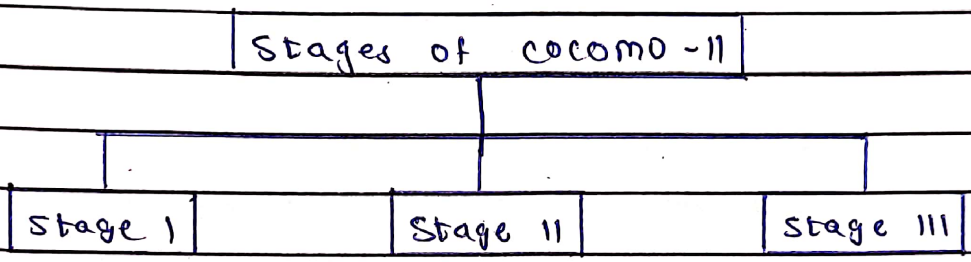
③ System Integration:
- This category deals with large scale and highly embedded systems.

02-08-2021

STUDENT SIGNATURE: Amey

③ Infrastructure Sector :
- This category provides infrastructure for the software development like Operating system, Database management system, User Interface management system, Networking system, etc.

Stages of COCOMO - II :




① Stage - I :
- It supports estimation of prototyping.
- For this it uses Application Composition Estimation model.
- It is used for prototyping stage of application generator and system integration.

② Stage - II :
- It supports estimation in the early design stage of the project, when we less know about it.
- For this it uses Early stage Estimation model
- It is used in early design stage of application generators, infrastructure, system integration

③ Stage - III :
- It supports estimation in the post architecture stage of a project.
- For this it uses post Architecture Estimation model.
- It is used after completion of the detailed architecture of application generator, infrastructure.

STUDENT SIGNATURE: Amey
02-06-2024

Q3

B] Project size of 200 KLOC is to be developed

Soln.

The Semidetached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence,

$E = 3.0 (200)^{1.12} = 1133.12$ PM

$D = 2.5 (1133.12)^{0.35} = 29.3$ PM

Average staff size (SS) $= \frac{E}{D}$ Persons

$= \frac{1133.12}{29.3} = 38.67$ Persons

Productivity $= \frac{KLOC}{E} = \frac{200}{1133.12}$

$= 0.1765$ KLOC/PM.

$P = 176$ LOC/PM.

∴ Average staff size = 38.67 persons.

∴ Effort = 1133.12 PM

∴ Development time = 29.3 PM

∴ Productivity = 176 LOC/PM.

02-06-2021

STUDENT SIGNATURE: Amey

Q3.

C]

Verification is Software Testing

- It is a process of checking documents, design, code and program in order to check if the software has been built according to the requirements or not.
- The main goal of verification process is to ensure quality of software application, design, architecture, etc.
- Verification process involves activities like reviews, walkthrough, etc.

Validation in Software Testing

- Validation in software testing is a dynamic mechanism of testing and validating if the process product actually meets exact needs of customer or not.
- The process helps to ensure that the software fulfills the desired use in an appropriate environment.
- The validation process involves activities like unit testing, integration testing, system testing and user acceptance testing.

- Verification comes first because it checks the document before executing it / Validating it.

02-06-2021

STUDENT SIGNATURE: Amey

## Example of Verification and Validation.

A Clickable button with name Submet
- Verification would check the design document and correcting the spelling mistake
- Otherwise, the development team will create a button like

[Submet]

- So new specification is [Submit]

A Clickable button with name Submit

- Once the code is ready, validation is done
- A Validation test found.

[Submit]

- Owing to Validation testing, the development team will make the submit button clickable.

02-06-2021 STUDENT SIGNATURE: Amey

## Verification and Validation explained :

| Verification | Validation |
|---|---|
| ① It includes checking documents, design, codes and programs. | ① It includes testing and validating the actual product. |
| ② Verification is the static testing | ② Validation is the dynamic testing |
| ③ It does not include the execution of the code. | ③ It includes the execution of the code. |
| ④ Methods used in verification are reviews, walkthroughs, inspections and desk-checking. | ④ Methods used in validation are Black Box Testing, White Box Testing and Non-Functional Testing. |
| ⑤ It checks whether the software conforms to specifications or not. | ⑤ It checks whether the software meets the requirements and expectations of a customer or not |
| ⑥ It can find the bugs in the early stage of the development. | ⑥ It can only find the bugs that could not be found by the verification process. |
| ⑦ The goal of verification is application and software architecture and specification. | ⑦ The goal of validation is an actual product. |

06-02-2021

STUDENT SIGNATURE: Amey

| | |
|---|---|
| ⑧ Quality Assurance team does verification. | ⑧ Validation is executed on software code with the help of testing team. |
| ⑨ It comes before validation. | ⑨ It comes after verification. |
| ⑩ It consists of checking of documents / files and is performed by human. | ⑩ It consists of execution of program and is performed by computer. |

Verification comes first because

→ It checks the documents / files whereas validation executes the program.

→ In order to execute a program it needs to be validated first.

→ Another reason is verification finds the bugs early in the development cycle whereas validation finds the bugs that verification cannot find.

06-02-2021

STUDENT SIGNATURE: Amey