

SE

### CAM ON? YES

**External - Rupali Sawant**

**Chatbox mai she will ask some questions**

Keep pressman open side mai

[http://ce.sharif.edu/courses/98-99/2/ce474-](http://ce.sharif.edu/courses/98-99/2/ce474-2/resources/root/Roger%20S.%20Pressman%20Bruce%20R.%20Maxin%20-%20Software%20Engineering%20A%20Practitioner%E2%80%99s%20Approach-McGraw-Hill%20Education%20(2014).pdf)

[2/resources/root/Roger%20S.%20Pressman %20Bruce%20R.%20Maxin%20-%20Software%20Engineering %20A%20Practitioner%E2%80%99s%20Approach-McGraw-Hill%20Education%20\(2014\).pdf](http://ce.sharif.edu/courses/98-99/2/ce474-2/resources/root/Roger%20S.%20Pressman%20Bruce%20R.%20Maxin%20-%20Software%20Engineering%20A%20Practitioner%E2%80%99s%20Approach-McGraw-Hill%20Education%20(2014).pdf)

### Roll No 3 and 4

Asked Prathamesh what exp he knows

He said RMMM

What is fullform of rmmm

A risk management technique is usually seen in the software Project plan. This can be divided into **Risk Mitigation, Monitoring, and Management Plan (RMMM)**. In this plan, all works are done as part of risk analysis.

Gave mcq in **chatbox**

Which one of the following is not a characteristic usually not expected of an SRS (Software Requirements Specification) document?

- a. Concise
- b. Consistent
- c. Complete
- d. Formal

We answered formal jo galat lag raha hai

Who does requirement gathering in SE project – **analysts**

Who sets the deadline - **project manager(not sure)**

Who writes the code - **developers/coders**

What happens if there is change in code - explain the process “**arey yeh hai na Pressman mai**”

Who initiates the changes - **the dev who worked on that module/feature**

, then what happens - we couldn't answer this

I think i said something about sqa so she steered in that direction

Software quality assurance

What is sqa, what happens in sqa - SQA Activities (Slide 9-12,33-34)

<https://www.guru99.com/software-quality-assurance-test-audit-review-makes-your-life-easy.html>



What is ftr

**Formal Technical Review (FTR)** is a **software** quality control activity performed by **software engineers**. Objectives of **formal technical review (FTR)**: ... To ensure that **software** is represented according to predefined standards. It helps to review the uniformity in **software** that is development in a uniform manner.

What is their difference

**She expects answers from Pressman**

### Roll No. 5 and 6

Process Models - factors to select them

Scenario based - 1. experience of staff factor, 2. requirements which process model to select?

Questions on Spiral Model and Agile

**piral Model** is a risk-driven software development process model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process.

Each phase of spiral model in software engineering begins with a design goal and ends with the client reviewing the progress. The spiral model in software engineering was first mentioned by Barry Boehm in his 1986 paper.

The development process in Spiral model in SDLC, starts with a small set of requirement and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase. The below figure very well explain Spiral Model:

## Spiral Model Phases

Spiral Model Phases	Activities performed during phase
---------------------	-----------------------------------

<b>Planning</b>	<ul style="list-style-type: none"><li>It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer</li></ul>
-----------------	--

<b>Risk Analysis</b>	<ul style="list-style-type: none"><li>Identification of potential risk is done while risk mitigation strategy is planned and finalized</li></ul>
----------------------	--

<b>Engineering</b>	<ul style="list-style-type: none"><li>It includes testing, coding and deploying software at the customer site</li></ul>
--------------------	---

<b>Evaluation</b>	<ul style="list-style-type: none"><li>Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun</li></ul>
-------------------	--

When to use Agile?

## When to use Spiral Model?

- A Spiral model in software engineering is used when project is large
- When releases are required to be frequent, spiral methodology is used
- When creation of a prototype is applicable
- When risk and costs evaluation is important
- Spiral methodology is useful for medium to high-risk projects
- When requirements are unclear and complex, Spiral model in SDLC is useful
- When changes may require at any time
- When long term project commitment is not feasible due to changes in economic priorities

Unit testing and Integration testing when are they used?

Unit Testing	Integration Testing
Unit testing is a type of testing to check if the small piece of code is doing what it is suppose to do.	Integration testing is a type of testing to check if different pieces of the modules are working together.

Unit testing checks a single component of an application.	The behavior of integration modules is considered in the Integration testing.
The scope of Unit testing is narrow, it covers the Unit or small piece of code under test. Therefore while writing a unit test shorter codes are used that target just a single class.	The scope of Integration testing is wide, it covers the whole application under test and it requires much more effort to put together.
Unit tests should have no dependencies on code outside the unit tested.	Integration testing is dependent on other outside systems like databases, hardware allocated for them etc.
This is first type of testing is to be carried out in Software testing life cycle and generally executed by developer.	This type of testing is carried out after Unit testing and before System testing and executed by the testing team.
Unit testing is not further sub divided into different types.	Integration testing is further divided into different types as follows: Top-down Integration, Bottom-Up Integration and so on.
Unit testing is starts with the module specification.	Integration testing is starts with the interface specification.
The detailed visibility of the code is comes under Unit testing.	The visibility of the integration structure is comes under Integration testing.
Unit testing mainly focus on the testing the functionality of individual units only and does not uncover the issues arises when different modules are interacting with each other.	Integration testing is to be carried out to discover the the issues arise when different modules are interacting with each other to build overall system.
The goal of Unit testing is to test the each unit separately and ensure that each unit is working as expected.	The goal of Integration testing is to test the combined modules together and ensure that every combined modules are working as expected.
Unit testing comes under White box testing type.	Integration testing is comes under both Black box and White box type of testing.

What is regression testing and when is it used?

**REGRESSION TESTING** is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.

The **Need of Regression Testing** mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not. Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

How to calculate size of project - **By Estimation techniques**

Give examples of Estimation techniques for calculating size of project - **LOC, FP**

Disadvantages of LOC

- Different programming languages contains different number of lines.
- No proper industry standard exist for this technique.
- It is difficult to estimate the size using this technique in early stages of project.

Advantages of LOC

- Universally accepted and is used in many models like COCOMO.
- Estimation is closer to developer's perspective.
- Simple to use.

Questions in chat:

At least how many test cases are required to achieve *multiple condition coverage* of the following code segment:

If((a>5) and (b<100) and (c>50)) x=x+1;

Ans: **8 is correct.**

10. At least how many test cases are required to achieve multiple condition coverage of the following code segment:

**If((a>5) and (b<100) and (c>50)) x=x+1;**

- a) 2
- b) 4
- c) 6
- d) 8**

**Explanation:** The number of test cases required can be easily found by the formula  $2^n$ . Here  $n = 3$ , so the number of test cases required is 8.

Which one of the following characteristics of a system makes it necessary to have a complex system made of both hardware and software, as opposed to having an entirely hardware system?

- a. High reliability requirement of the system
- b. Low development cost of the system
- c. Easy to change the system to meet changing customer requirements**
- d. Low operating cost

Roll no 07 & 08:

What are the responsibilities of a change auditor ?

Considering auditor is changed without completion of his term. Following step should be taken.

1. Pass resolution at board meeting.
2. Obtain the approval of Regional Director.
3. After obtaining the approval from Regional Director call a General meeting and pass a special resolution.

Before initiating the proceedings a notice must be given to the auditor.

What is boundary value analysis?

Boundary value analysis is one of the widely used case design technique for black box testing. It is used to test boundary values because the input values near the boundary have higher chances of error.

Whenever we do the testing by boundary value analysis, the tester focuses on, while entering boundary value whether the software is producing correct output or not.

Boundary values are those that contain the upper and lower limit of a variable. Assume that, age is a variable of any function, and its minimum value is 18 and the maximum value is 30, both 18 and 30 will be considered as boundary values.

The basic assumption of boundary value analysis is, the test cases that are created using boundary values are most likely to cause an error.

There is 18 and 30 are the boundary values that's why tester pays more attention to these values, but this doesn't mean that the middle values like 19, 20, 21, 27, 29 are ignored. Test cases are developed for each and every value of the range.

<https://www.javatpoint.com/boundary-value-analysis-in-black-box-testing>

Which black box testing technique should you use?

Which testing technique (white box to black box) to use for a coercive model?

What is re engineering ?

**Software Re-engineering** is a process of software development which is done to improve the maintainability of a software system. Re-engineering is the examination and alteration of a system to reconstitute it in a new form. This process encompasses a combination of sub-processes like reverse engineering, forward engineering, reconstructing etc.

**Objectives of Re-engineering:**

- To describe a cost-effective option for system evolution.
- To describe the activities involved in the software maintenance process.
- To distinguish between software and data re-engineering and to explain the problems of data re-engineering.

**Steps involved in Re-engineering:**

1. Inventory Analysis
2. Document Reconstruction
3. Reverse Engineering
4. Code Reconstruction
5. Data Reconstruction
6. Forward

Engineering

Definition of milestone, product, deliverable

**What is a project milestone?**

- Can be conceptual or tangible
- Signals the reaching of a key stage in the project
- An important sign for the team
- A point for project management to check project goal alignment

**What is a project deliverable?**

- Must be tangible (i.e. a product or service)
- Signals the completion of a project phase
- An important sign for the client
- A point for the client to sign-off on project status

**Milestone vs. deliverable: What's the same and what's different?**

As you can see from the above characteristics, there IS considerable overlap between project deliverables and project milestones. For example, a particular deliverable may coincide with a certain milestone or vice-versa. But ultimately, deliverables and milestones represent two distinct – if often parallel – tracks along which projects are measured and carried out.

Let's dive a little deeper into the distinctions.

Roll No 9:

What is your preferred experiment.. I didn't say any specific one

What are the different types of analysis?

What is WBS?

The Work Breakdown Structure (WBS) is developed to establish a common understanding of project scope. It is a hierarchical description of the work that must be done to complete the deliverables of a project. Each descending level in the WBS represents an increasingly detailed description of the project deliverables.

The first two levels of the WBS (the root node and Level 2) define a set of planned outcomes that collectively and exclusively represent 100% of the project scope. At each subsequent level, the children of a parent node collectively and exclusively represent 100% of the scope of their parent node.

Here is a Work Breakdown Structure example:

Why is it needed?

What is the Critical path?

**in project management, the critical path is the longest sequence of tasks that must be completed to successfully conclude a project, from start to finish.** The tasks on the critical path are known as critical activities because if they're delayed, the whole project will be delayed. By identifying the critical path, you can determine the total duration of a project.

Calculating the critical path is key during the planning phase because the critical path identifies important deadlines and the activities which must be completed on time. Once a critical path is determined, you'll have a clear picture of the project's actual schedule.

What is buffer time?

**Buffer time**, in project management, is the extra **time** added into a **time** estimate to keep a project on track. ... It allows project managers to be able to account for unforeseen situations without having to change the coordination of a project in a major way.

**Roll No 20, 21**

Cost slippage



n project planning, a **slippage** is the act of missing a deadline. It can be an arbitrary milestone put in place to help track progress. To avoid **slippage**, one must plan his or her projects (especially research) carefully to avoid delays in schedule.

### Cost estimation

For any new software project, it is necessary to know how much it will cost to develop and how much development time will it take. These estimates are needed before development is initiated, but how is this done? Several estimation procedures have been developed and are having the following attributes in common.

1. Project scope must be established in advanced.
2. Software metrics are used as a support from which evaluation is made.
3. The project is broken into small PCs which are estimated individually.  
To achieve true cost & schedule estimate, several option arise.
4. Delay estimation
5. Used symbol decomposition techniques to generate project cost and schedule estimates.
6. Acquire one or more automated estimation tools.

## Uses of Cost Estimation

1. During the planning stage, one needs to choose how many engineers are required for the project and to develop a schedule.
2. In monitoring the project's progress, one needs to access whether the project is progressing according to the procedure and takes corrective action, if necessary.

### Analytical estimation techniques

In this **technique**, firstly the task is divided or broken down into its basic component operations or elements for analyzing. Second, if the standard time is available from some other source, then these sources are applied to each element or component of work.

### EVA

Earned Value Analysis (**EVA**) is an industry standard method of measuring a project's progress at any given point in time, forecasting its completion date and final cost, and analyzing variances in the schedule and budget as the project proceeds.

### Extreme Programming (XP)

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the

development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

Phir agile pe aaye mam

Sprint scrum

A **sprint** is a short, time-boxed period when a scrum team works to complete a set amount of work. Sprints are at the very heart of scrum and agile methodologies, and getting sprints right will help your agile team ship better **software** with fewer headaches.

**Scrum** Overview for Agile **Software** Development. **Scrum** is an agile process most commonly used for product development, especially **software** development. **Scrum** is a project management framework that is applicable to any project with aggressive deadlines, complex requirements and a degree of uniqueness.

Which one of the following is usually not a distraction for adoption of the agile model in software development projects?

- a. Sketchy definitions of the model, make it possible to have Inconsistent and diverse definitions
- b. The project may never converge into a meaningful solution
- c. Short iterations inhibit long-term perspective
- d. Higher risks due to feature creep.

Which one of the following is not a basic principle of the agile model of software development?

- a. Frequent delivery of versions, at least once every few weeks.
- b. Completion of the SRS (Software Requirements Specification) document before the design phase.
- c. Close cooperation between customers and developers.
- d. Face-to-face communication among team members.

### Roll No 24

Why and how to find cyclomatic complexity?

Cyclomatic complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.

Lower the Program's cyclomatic complexity, lower the risk to modify and easier to understand. It can be represented using the below formula:

```
Cyclomatic complexity = E - N + 2 * P
where,
  E = number of edges in the flow graph.
  N = number of nodes in the flow graph.
  P = number of nodes that have exit points
```

If two code segments have Cyclomatic complexities of N1 and N2 respectively, what will be the

Cyclomatic complexity of the juxtaposition of the two code segments?

- a. N1+N2
- b. N1+N2+1

c.  $N1+N2-1$

d.  $N1*N2$

Justify the answer

**Roll Number 26,27**

Project Metrics, Process Metrics

### 1. Process Metrics

These are metrics that pertain to Process Quality. They are used to measure the efficiency and effectiveness of various processes.

### 2. Project Metrics

These are metrics that relate to [Project Quality](#). They are used to quantify defects, cost, schedule, productivity and estimation of various project resources and deliverables.

### 3. Product Metrics

These are metrics that pertain to Product Quality. They are used to measure cost, quality, and the product's time-to-market.

What is fan in fan out

**Fan-in** refers to the maximum number of input signals that feed the input equations of a logic cell. **Fan-out** refers to the maximum number of output signals that are fed by the output equations of a logic cell.

Which is more beneficial fan in, fan out

What is coupling and cohesion

Cohesion	Coupling
<b>Cohesion</b> is the indication of the relationship within <b>module</b> .	<b>Coupling</b> is the indication of the relationships between modules.
Cohesion shows the module's relative <b>functional</b> strength.	Coupling shows the relative <b>independence</b> among the modules.
Cohesion is a degree (quality) to which a component / module focuses on the <b>single</b> thing	Coupling is a degree to which a component / module is connected to the <b>other</b> modules.

## Coupling

## Vs

## Cohesion

Cohesion is the concept of intra module.	Coupling is the concept of inter module.
Cohesion represents the relationship within module.	Coupling represents the relationships between modules.
Increasing in cohesion is good for software.	Increasing in coupling is avoided for software.
Cohesion represents the functional strength of modules.	Coupling represents the independence among modules.
Highly cohesive gives the best software.	Where as loosely coupling gives the best software.
In cohesion, module focuses on the single thing.	In coupling, modules are connected to the other modules.

Which testing method should be used for coupling

As a debugger what is better for testing coupling or cohesion

Integration testing is important more in coupling or cohesion

Upto which level you can go in WBS

The **level can go upto** 99 but the **WBS** length is restricted to 24 characters.

### Roll Number 28

Started with one mcq question in the chat box.

What is software configuration management?

In Software Engineering, **Software Configuration Management(SCM)** is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle. The primary goal is to increase productivity with minimal mistakes. SCM is part of cross-disciplinary field of configuration management and it can accurately determine who made which revision.

What is version control?

**Version control**, also known as **source control**, is the practice of tracking and managing changes to **software** code. **Version control** systems are **software** tools that help **software** teams manage changes to **source** code over time.

If we have 4 functions A B C D and we implement only A and B then should we count it as version or not?

What is enhancement of a project?

An **enhancement project** is one in which new capabilities are added to an existing system. **Enhancement projects** might also involve correcting defects, adding new reports, and modifying functionality to comply with revised business rules or needs.

When should we go for an enhancement project and start a project from scratch?

Which would cost more?

### Roll Number 29

MCQ :

Which one of the following is not a factor explaining why software development using a high-level

programming language takes less time and effort as compared to development of the same software using an assembly language?

- a. Using a high-level language, it is easier to write structured programs as compared to writing the same software using assembly language.
- b. When software is developed using high-level language, reuse of code is easier
- c. When software is developed using high-level language testing the full software is easier
- d. When software is developed using high-level language, when any test case fails during testing, debugging the software is easier

Give methods to find cost/budget of project

Full form of COCOMO

The Constructive Cost Model (**COCOMO**) is a procedural software cost estimation model developed by Barry W. Boehm. The model parameters are derived from fitting a regression formula using data from historical projects (63 projects for **COCOMO** I and 163 projects for **COCOMO** II).

What is COCOMO

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981. COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

**The necessary steps in this model are:**

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort  $E_i$  in person-months the equation used is of the type is shown below

What is FP