

Software Engineering Series

www.lastmomenttuitions.com

INDEX

Q1. Explain agile methodology. -----	03
Q2. Explain Formal Technical Review (FTR). -----	06
Q3. What is Software Engineering? -----	07
Q4. Difference between alpha testing and beta testing. -----	09
Q5. Explain Black Box Testing. -----	09
Q6. Explain Change Control In software engineering. -----	12
Q7. Explain iterative model in SDLC. -----	14
Q8. Explain process framework. -----	15
Q9. Explain waterfall model. -----	17
Q10. Write a short note on Capability Maturity Model (CMM). -----	18
Q11. Difference between verification and validation. -----	20
Q12. Write a short note on integration testing. -----	20
Q13. Explain Maintenance in Software Engineering. -----	22
Q14. Explain Object Oriented Testing Methods. -----	23
Q15. Write short note on Reverse Engineering. -----	24
Q16. Explain risk identification and risk projection. -----	26
Q17. Write a short note on extreme programming. -----	28
Q18. Explain scrum in software engineering. -----	29
Q19. Explain test driven development. -----	31
Q20. Explain Service Oriented Software Engineering. -----	33
Q21. Explain risk mitigation, monitoring and management. -----	34
Q22. Write a short note on Software Measurement. -----	36

Q23. Explain cohesion and coupling. -----	37
Q24. Explain White Box testing. -----	39
Q.25 Explain Schedule and Cost Slippage. -----	43
Q26 Explain COCOMO model. -----	44

1) Q1. Explain agile methodology.

A1. In terms of Software Development, basically 'agile' means the ability to respond to changes, i.e., changes from Requirements, Technology and people.

- It is an iterative and incremental process.
- It offers direct collaboration with the customers.
- Delivers multiple Software Increments.
- Here each iteration lasts from one to three weeks.
- Engineering actions are carried out by cross functional teams(a team expert in their respective work assigned)
- A team of software developers published the Agile Manifesto in 2001, highlighting the importance of the development team, accommodating changing requirements and customer involvement.
- The principles of agile methodology were defined by Manifesto (a public declaration of policy and aims).

The principles of agile process models are:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Provide the individuals the environment and support which they need, and trust them to get the job done.
5. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace.
6. Business men and developers must work together daily throughout the project.
7. To measure the progress done we run the software thus Working Software is the primary measure of progress.
8. The most efficient and effective method of conveying information to and within a development team is face to face.
9. Attention should be given to technical excellence and good design which enhances agilities.
10. Teams should know about the amount of work not completed.
11. The teams should be self-organized and independent.
12. At regular intervals, the team should reflect on how to become more effective and adjust its behavior accordingly.

Agile Process Models

1. Extreme Programming

Planning:-

- Start with creation of set of stories
- Describes required features and functionality
- Customer will give priority to each user story
- Xp team will access each story and cost
- Stories with highest priority will be developed first and then with risk stories

Design:-

- It should be as simple as possible
- CRC(class-responsibility-collaboration)- they identify and organize the object oriented classes relevant to current software increment
- If difficult design problem is encountered then immediately an operational prototype is designed called spike solution

Coding:-

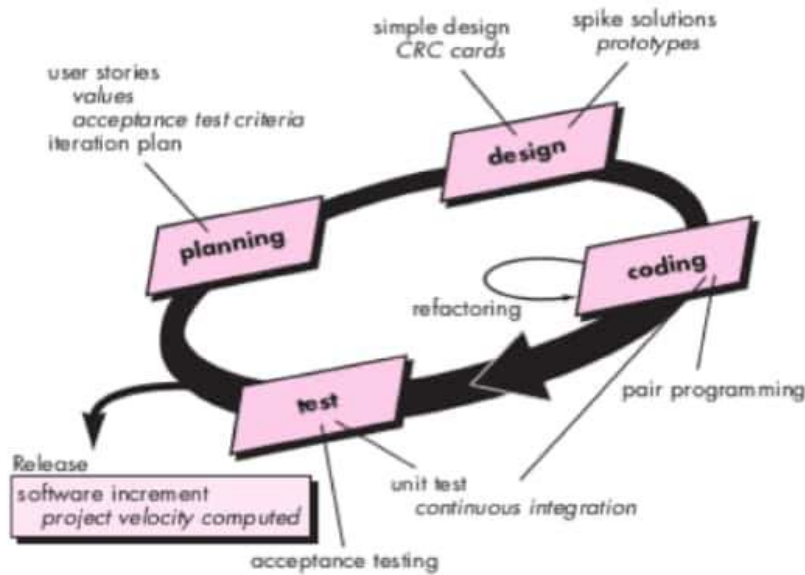
Pair Programming-

- It is a process of two people working together on one workstation to create code
- Many pair programmers integrate their code to get complete code of that unit

Testing:-

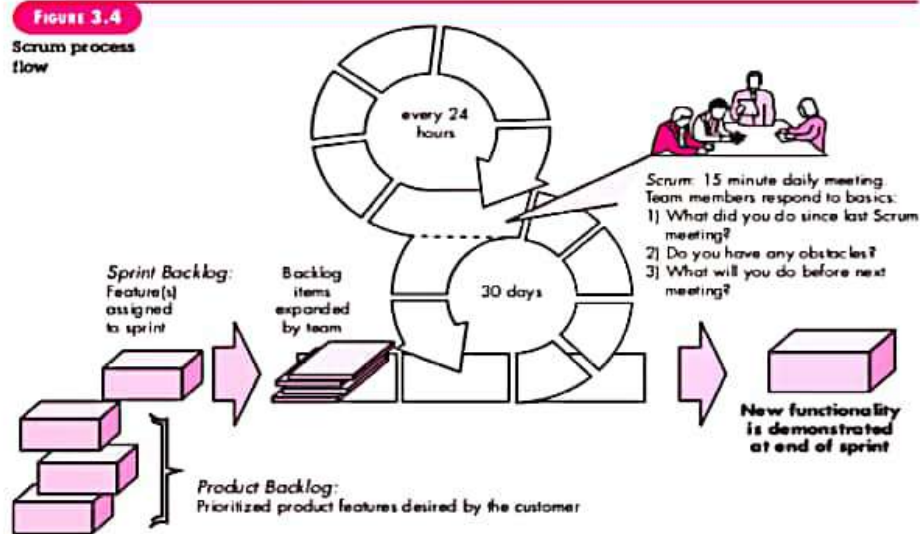
Unit testing-

Here acceptance testing also called as customer testing is used. It means comparing software with the expectations of users, then seeing if it satisfies the user.



2. Scrum

- It is iterative and incremental agile methodology principles
- Small teams to maximize communication and minimize overhead
- Process must be adaptable to technical and business changes to ensure best possible product is produced
- Process yields frequent software increments
- Constant testing and documentation is performed as product is built
- Backlog(list of prioritized requirements)
The product manager accesses the backlog and updates the priorities
- Sprints
-Work units required to achieve requirement are defined
- Scrum meetings
-Short meetings held daily by scrum team
-It includes the discussions about what did you do in the last meeting, what obstacles are you encountering, what do you plan to accomplish by next team meeting



2) Q2. Explain Formal Technical Review (FTR).

A2. Formal Technical Review (FTR) is a SQA Activity. SQA stands for Software Quality Assurance.

- FTR is a type of review conducted by technical staff-software engineers.
- It focuses on making system error free.

The main objectives of FTR are as follows:-

1. It is used to uncover the errors.
2. It is used to verify that software under review meets the requirements.
3. It maintains the standards which were pre-defined.
4. It makes the project manageable and effective.
5. It ensures that the software is developed in uniform manner.
6. Groom new resources.
7. Provide backup and continuity.

It includes –

Walk-through, inspections, other small group technical measurements of software.

It is conducted in a meeting that is successful only if it is properly planned, controlled, and attended.

Guidelines for walkthroughs

The review meeting is

- Between 3-5 people.
- Advanced preparation –Should not require more than 2 hours of work per person.
- Less than two hours.
- Focuses on specific part of the overall software.
- Focus is on work product.
- Producer- asks the project leader for review.
- Project leader informs the review leader.
- The review leader- evaluates the work process
 - generates copies of review material
 - prepares the agenda
- The review meeting is attended by the review leader, all reviewers, and the producer.
- One of the reviewers takes the roles of recorder.
- Producer walks through the product, explaining the material

Review Reporting and Record Keeping

- During the FTR the recorder notes all the issues
- These issues are summarized at the end and a review issue list is prepared
- A summary report is produced that includes: what is reviewed, who reviewed it, what were the conclusions
- This report makes error identification easy
- Serves as checklist for developers

The Reviewer raises issue based on their advanced preparation

At the end, all attendees of the FTR must decide

- Accept the product without further modification
 - Reject the product due to severe errors
 - Accept the product provisionally, i.e. with some modification
-