CS587
Midterm Exam

St. Name : Namachivayan, Sivasenthil

St. ID : A20391478

10 points for every question

Q1. A code inspection task was executed by a team of reviewers who found the code to be in compliance with the process and according to the project plan, however, the number of defects detected as a result of 9this inspection was significantly higher than the upper control limit per the process and project plan, explain what are the possible actions to be considered by the project manager and moderator.

During the review, if the number of defects found is within the range mentioned in the baseline then the review is considered effective and the exit criteria is satisfied else the moderator or the project leader needs to determine the cause and take the preventive measures and corrective actions to rectify the issue. Possible reasons and actions to consider if the defect is found are more than norms,

Work Product is of low quality

Examine the training needed for the author
Need to redo the work product
Consider assigning easier tasks to the author in the future

Work Product is very complex

Ensure good review or testing downstream
Increase the estimates for the system testing
Break the work product into smaller components to show the complexity in an easier way

There are too many minor defects and few major defects.

Identify the causes of minor defects. Correct the defects in the future by making proper checklists and follow them. Make the authors to be aware of all the common causes.

Reviewer may have insufficient understanding of the work product, if so then hold a discussion or an overview meeting with different reviewers in-order to give them the clarity of the work product

Reference document may also be a factor where it may not be precise or clear.

Make sure the reference document is reviewed and approved

Reviewed modules are considered primary in the projects

Analyze the defects, update the review checklist and inform the developers and schedule training sessions for the proper update.

Q2. From the perspective of software project management, software testing is only one aspect to ensure the quality of the software produced, Explain.

In order to ensure the quality of the product one must follow the quality assurance standards strictly during all the phases of the project. Several audit groups verify these process to ensure the process comply with ISO standards.

In SQA, process can be improved by periodically gathering statistics about Requirements review, Design review/analysis, code inspections, test plans phases.

Requirement review:

Check whether project can be done/developed
Testability of the project Correctness of the project

During Analysis/Design:

Check whether the design is meeting the requirements accurately
Check if the design can be implemented successfully
Checking the correctness and the standards of the design as per standards
Ensure that the design is based on the formal model

During Implementation Phase:

Conduct code inspection meetings to make sure that the implementation is matching the design document.
Check whether the implementation is driven using CASE tools.
Check if the code is meeting the global coding standards
Ensure implementation is driven by CASE tools
Identify logical errors in relation to the requirements/design
Make sure all the requirements are met.

During testing phase:

Formulate test plans to conduct testing of the developed code
Test the product with the requirements given
Test the results in the repository.

Software testing is only one aspect to ensure the quality of the software produced. We need to maintain all the process mentioned above for a quality product to be developed.

Q3. Which one is better a network diagram with few Zero-Slack activities or many Zero-Slack activities?

Critical path is the sequence of activities with which the project completion date is determined. It is the longest path duration in the network diagram. This critical path can be calculated using two techniques

Find all the possible paths through the network diagram and add the duration of all the activities for each path. The path that has the farthest duration is called the critical path. Summarize all the duration of the activities that lie along those paths. This method is not useful for the large size project since we cannot calculate for more number of nodes.

Find the critical path through the slack time. Slack time is the difference between the latest finish(LF) and the earliest finish(EF) (LF-EF). The critical path is the slack time where all the activity nodes are zero.
If the Slack time is greater than zero (Slack time > 0), the activity has a range of time in which it can start and finish without delaying the project completion date. The critical path has the minimum slack.

If there are few zero slack activities then the critical path would contain less number of activity nodes in the path. Hence the path would be smaller. If the path is small then the project completion date would also be small.

Q4. Explain how the requirements may become a risk factor for the software project plan?.

Risk factors for the feasibility study activity:

Inadequate estimation of project time, cost, scope and other resources

Unrealistic schedule will be implied on the developers and other engineers which will mostly affect the progress of the project

Unrealistic Budget for the project plan which actually depends on the time, effort and resources available.

Unclear Project scope in managing the project(i.e Size goals and requirements)

Insufficient resources(People, tools and technologies) will affect the progress and completion of the project

Risk Factors for the requirements elicitation activity:

Unclear Requirements if the requirements are not understandable by the analysts and developers

Incomplete Requirements if the requirements are missing some of the user needs, constraints and other requirements

Inaccurate Requirements are considered if the requirements does not reflect the real user needs

Ignoring the Non-Functional requirements which is also equally important like functional requirements

Conflicting user requirements

Unclear Description of the real environment

Gold Plating – adding extra functionality to the system which is not in the actual requirement will cause threat to the project

Risk Factors for the requirements analysis activity:

Non Verifiable requirements- The requirements are non verifiable if there is not a finite cost effective process(testing, inspection, demonstration or analysis) with which we can check the software meets the requirements

Infeasible Requirements- If there is no sufficient resources available for the implementation then it is considered as infeasible

Inconsistent Requirements- If the requirements contradict with the other requirements in the projects

Non- Traceable Requirements- Track all the origin source of all the requirements for referencing in future.

Unrealistic Requirements- If the requirements are not clear, verifiable, accurate, consistent, complete and feasible then they are unrealistic for implementation.

Risk Factors for requirements validation activity:

Misunderstanding domain specific terminology- Application specialists and developers use domain specific terminologies that are different and not understandable by most end users which cause confusion.

Mis-expressing user requirements in natural language- Natual language is not the best way to express the requirements since many use different natural language and conventions. There should be a formal way of expressing and documenting which will be easily understandable for all.

Risk Factors for requirements documentation activity:

Inconsistent requirements data and RD- The inconsistency arise between the actual requirements data and the documented data which is due to the difference in gathering and documenting techniques.

Non-Modifiable requirement document- While documenting the requirements, structuring the document with maintainability is not considered which will affect in modifying the data in requirement document. This might cause the rewriting of the entire document.

Q5. Who controls the design review meeting? What are the different metrics collected in the review meeting?

System Engineers controls the design review meeting.

The Various phases along with different metrics which involves the review meeting are listed below.

Requirement Specification Document Review

Requirements must meet the customer/client needs

Make sure that the requirements provided by the customer/client is practically possible to implement with the current development standards

Check for the redundancy, omissions, inconsistencies and ambiguities in the requirements

High Level Design Document Review

Make Sure that the design is in parallel with the requirements provided in the requirement phase

Make sure that the design is practically possible to implement with the current development standards

Check for the omissions and defects in the design

Code Review

Check whether the code implements the design developed

Verify the completeness and correctness of the developed code

Check for the defects in the code

System Test Case Review

Make sure all the test cases developed verify the requirements

Check if the system test cases are correctness

Make sure the test cases are executable and get the desired results

Project Management Plan Review

Check whether the project management plan is meeting with the project management and the control needs

Make sure that the completeness of the project management plan

Make sure the project management plan is able to be implemented practically

Check for the omissions and ambiguities in the project management plan

Q6. Can reviews and inspections tasks replace/eliminate the testing tasks? Explain.

No. The reviews and inspection tasks cannot be replaced or eliminated in the testing tasks.

Inspection is the most known way of review practices followed in software projects.

The main aim of inspection process is to identify defects and check the quality of the product. Inspection is the most formal form of reviews. The software inspection process generally has two types of reviews

1.code review

2. peer reviews.

The software project inspection and reviews process cannot be replaced with the testing tasks due to the following reasons:

Software testing is performed based on the test cases written by the testing engineer whereas inspection or reviews are done based on the project specifications.

While in testing phase, code is tested to find new errors Whereas in inspection only the specifications are checked.

While testing can be done by test automation tools, Inspection can only be done manually

Defects are identified in both testing and inspection phases but in testing the defects are tracked and reported immediately unlike inspection

In inspection or reviews 4 engineers with at least one engineer from every department is required where as it is not the case in software testing

Software testing is used to test the performance, but using inspection or reviews we cannot measure the performance

Load test, Regression tests etc. are very important and can be achieved only using software testing. Similarly it is difficult to replace unit testing and other testings with inspection because in inspection even for a small change in the code more time will be wasted later.

Q7. For a software development organization that is CMM level-3, which method can be used for estimating activity duration: historical data or three-point technique? Explain.

Three-point technique can be used for estimating activity duration in a CMM Level-3 software development organization.

CMM Level-3 means Capability Maturity Level even through the organizations that fall under this level will have the process followed in previously executed documented projects, it will not have a proper quality in the records.

In-order to use the historical data method for the activity duration/effort estimation, the organization must have proper quality in the records of the estimated and actual activity duration of the past projects that is saved in a common document repository which is a part of project development process.

Hence, historical data method is not a suitable method for effort/duration estimation for the organization below CMM Level-4 which is in our case it is CMM Level-3

Q8. What are the possible actions that the project manager and review moderator might consider to take for the following outcomes of design document review?

1. Rework and defect fixes turned out to require more than 60% of the original effort to write the design.

If something of that sort is to happen, this means the coding principles and standards aren't followed properly in the project. The same code should be sent back to developers and this should be included in sprint back log. This is an overhead to the manager because most of the code needs to be corrected which will take time. The whole cycle would be disturbed and a strict message needed to be conveyed to the developers regarding the issue. For the above scenario, the project manager may take any of following steps based on the situation,

The Project Manager schedules a meeting with design team and identify the cause for the enormous number of defects.

To make sure the flaws are not repeated and he might discard the existing document and ask to develop the design document again for the rework.

The design document is revisited to identify the hidden flaws.

More skillful resources are allocated to make sure the there are no defects in the documents and the previous resources may be put into training to improve their skills.

The design work can be split to different phases and integrated later after completion of the design.

After the modern design document is created, it is reviewed by the other technical resources.

2. Rework and defect fixes turned out to require 20% of the original effort to write the design.

Manager should offer insights, advice and comes up with a list of preventive actions that helps in reducing work reload. Detect fixes are something that no one can avoid while developing. These changes can be easily accommodated in the sprint cycle and managers usually reinsert defective WIP upstream. The causes for these should be left to the developers to be discussed during sprint retrospective meetings between developers.

It is still manageable using 20% of the original effort to write the design. As the rework is not very large, it is not required to introduce additional resources as the rework can be done using the existing resources. The manager makes sure that the task is completed within the given time.

3. Rework and defect fixes turned out to require 5% of the original effort to write the design.

   This is very usual and unavoidable phase which happens in every project and sprint cycle. Managers should make sure testing is done properly and in an effective manner to avoid these in later phases. Testers should take care of these minor bugs or developers while doing unit testing can detect these and correct them before passing them over to testers. Sprint cycle would never be disturbed because of these defects and errors.

   As most of the rework process generates at least 5% defects, for above scenario, the project manager may take any of the following steps to reduce the defects:

   He might initiate review meeting with different reviewer/moderator to improve the previous design.

   If the amount of rework remains same, then he asks to go ahead with the documents with identified defects to be fixed.

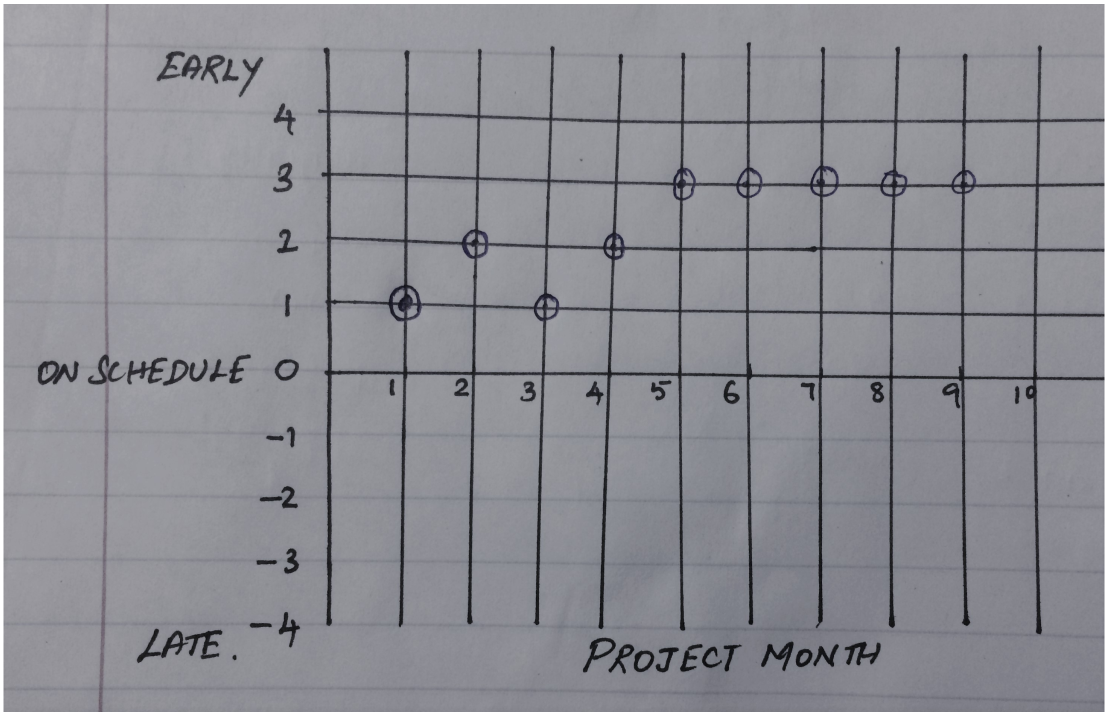
   He might ask documentation team to document the process and methods, techniques followed by design team for future use

Q9. Consider the following milestone table, what is the milestone trend chart that the following project follows? Name and draw the milestone trend chart.

Milestone	**Expected Delivery**	**Actual delivery**
Project Planning	1st month	early 1 week
Lab/Environment Installation	2nd month	early 2 week
Requirement Phase	3rd month	early 1 week
Analysis phase	4th month	early 2 week
Design phase	5th month	early 3 weeks
Coding	6th month	early 3 weeks
Testing	7th month	early 3 weeks
Documentation	8th month	early 3 weeks
Installation/Training	9th month	early 3 weeks

The above data represents permanent schedule shift. Barring any radical changes and availability of resources, this project will probably complete early.

Milestone trend chart follow successive runs.



Q10. Consider the following data; calculate the effort and duration required for every task, considering the following constraints:

1. An artifact is produced by only one author
2. Every review "meeting" task shall be carried by 5 engineers including the author
3. Every review "preparation" task shall be carried by 4 engineers excluding the author
4. Any "Rework" task can be executed by the author of the original task

Tasks	**Amount of Work**	**Productivity**	**Effort ***	**Duration ***
Requirements				17.5
Write Requirements Document	100 pages	1 page/Hour	12.5	12.5
Review Requirements Document				
Review Preparation for Req. Doc.		5 pages/Hour	10	2.5
Review Meeting		10 pages/Hour	6.25	1.25
Rework	10 defects	1 defect/Hour	10	1.25
Design				15.5
Write Design Document	80 pages	1 page/Hour	10	10
Review Design Document				
Preparation for Design Document		5 pages/Hour	8	2
Review Meeting		10 pages/Hour	5	1
Rework	20 defects	1 defect/Hour	2.5	2.5
Testing				5.75
Write Test Plan	50 pages	2 pages/Hour	3.125	3.125
Review Test Plan				
Preparation for Test Plan		5 pages/Hour	5	1.25
Review Meeting		10 pages/Hour	3.125	0.625
Rework	30 defects	5 defects/Hour	0.75	0.75

**** All the numbers mentioned in the columns "Effort" and "Duration" of the above table is in "Days"***