CS587
Midterm Exam

St. Name: Kumari, Ritika

St. ID: A20414073

Q1. Explain the difference between desk review and group review when performing peer-review for the different software artifacts produced as a result of following the project plan and the development process. Give examples for when we should use desk reviews and when we should use group reviews.

Answer: A software review is a process or meeting during which a software product is examined by a project personnel, managers, users, customers, user representatives, or other interested parties for comment or approval. Review are the most effective method to improve quality by identifying defects in the origination phase of the artifacts. Reviews can be used to track the progress and prevent defects discovered by Customer. Review can come in different forms, it can be Formal Group Review (also called Inspection) and Desk Review.

Desk Review is an informal review where the author distributes a work product to peers for reviews and comments. Desk Review is used for inspection small task like inspecting 100 lines of code or not more than 2-3 pages.

Group Review is a formal review where a work product is selected for review and a team is gathered for an inspection meeting to review the work product. A moderator is chosen to moderate the meeting. Each inspector prepares for the meeting by reading the work product and noting each defect. In an inspection, a defect is any part of the work product that will keep an inspector from approving it. For example, if the team is inspecting a software requirements specification, each defect will be text in the document which an inspector disagrees with.

For inspecting more tasks like 2000 lines of code or around 20 -30 pages then we choose formal group review. Example: If I have a developer who wrote 5000 Line of Code. I expect 10 defects to be caught for a good inspection per 100 line of code. I will choose group review since the number of lines of code need to be inspected are more. If there were only 500 lines of code and I expect 10 defects to be caught for a good inspection per 100 line of code I would have chosen desk review.

Q2. Quantitative process management is one of the KPA in CMM level 4, explain.

Answer: The purpose of Quantitative Process Management in CMM level 4 is to control the process performance of the software project quantitatively.

In quantitative process management CMM level 4, following things are performed:

i) It collects and records the data for selected process. In CMM Level 4, artifacts and quality records are documented.

ii) It reduces process variation by analyzing software process being followed by a software project and bring under quantitative control according to a documented procedure.

iii) It controls process variation by collecting measurement data and then stores these data in organization's measurement repository to support fact-based decision in future.

iv) It prepares and distributes the reports having the results of any software project's quantitative process management activities.

v) It establishes and maintains the process capability baseline for the organization standard software process.

vii) Once the quantitative objectives for software quality and software process performance is established, then it is used as criteria in managing processes.

Quantitative objectives are mainly focused on the requirements of the customer, end users, organization, and process implementers.

This KPA in CMM level 4 can be summarized as predictable because the process is measured and operates within measurable limits. This level of process capability allows an organization to predict trends in process and product quality within the quantitative bounds of these limits. When these limits are exceeded, action is taken to correct the situation.

Q3. What is the purpose of the work package? Who will create it?

Answer: The work to be done within activity is called a work package. A work package is a group of related tasks within a project. Tasks are typically grouped into work packages based on geographical area, engineering discipline, technology, or the time needed to accomplish them.
Work package will be created by the Activity Manager.

Below are the purposes of the work package:
i) It describes each tasks in detail. The work package manager may include the start and end date for each task in the package.
ii) For each task, start and end date is defined in the work package. Task must be finished within defined start and end date in order to complete the work for an activity.
iii) The work package can also be used to track the status of work completed. The percent of tasks completed is used as the percent of activity completion.
iv) The work package manager may allow others to micromanage your work items or artifacts produced in each task of an activity.
v) Work packages allow for simultaneous work on different components.
vi) Costs of activities are collected at the work package level so they can be measured, monitored and controlled.
vii) Work package includes work product, checklists, specifications and standards.

Q4. For a software development organization that has the quality records for projects it executed for the past 10 plus years, what is/are method(s) to estimate activity durations for new projects? Explain.

Answer: There are six methods to estimate activity durations for new project:

1. **Historical Data:** According to question, an organization has the quality records for project it executed for the past 10 plus years. These records will be saved in a common document repository of the estimated and actual activity duration. If new project is having activities identical to the activities records in the common repository then we can refer the recorded data for that similar activity from the common repository. So, we can estimate activity duration referring past records of similar activities if any exists in the common repository.

2. **Similarity to Other activity:** This technique does not depend on any stored documents for estimation of durations. It depends on your peer's experience or personal experience if they are including yourself had worked in past 10 plus years projects. For this method, official documentation or quality records are not required for the referenced experience i.e. peer or yourself. If the company has carried out a similar activity, it may be possible to adapt the duration to the current case. Project managers must study the similarities of the two activities and adjust for any features that may result in differences in duration.

3. **Expert Advice:** Expert judgment means using specialists who have a reputation for knowledge of the field and experience in estimating activity duration within it. If new project has technology that is being used for the first time in the organization and no engineer is skilled in that technology, then it depends on outside sources for advice and consultation in order to estimate activity durations.

4. **Delphi Technique:** If past 10 plus years records doesn't help to estimate new project's activity and if no expert advice is available then this technique can be used. This technique is used when there is lack of expert advice. Firstly, the team members are briefed on the nature of the new project. Then everyone is asked to guess the activity durations. The results are discussed in the team meeting. This is done three times. The average of the third guess is considered as the final estimate.

5. **Three-Point Technique:** This technique is based on three types of estimates – optimistic, pessimistic and most likely. Optimistic time is the shortest duration a person might take to complete the activity given everything works as expected. Pessimistic time is the longest duration a person might take when everything

goes wrong. Most likely time is the time that is taken usually by a person. This time is decided after consulting professionals who have worked on similar activities in the past but not documented.
Then the estimate is calculated as:

$$E = \frac{O + 4M + p}{6}$$

Where E is the estimate, O is the optimistic time, P is the pessimistic time and M is the most likely time.

6. **Wide-Band Delphi Technique:** It is a combination of Delphi technique and Three-Point Technique. Group or panel give optimistic, pessimistic and most likely estimates for the duration for selected activity at each iteration. The results for each iteration are compiled and any extreme estimates are removed. After these averages are computed for each of the 3 estimates and final average value are used as the optimistic (O), pessimistic (p) and most likely (M) in order to calculate activity duration.
Estimate can be calculated by,

$$E = \frac{O + 4M + p}{6}$$

Where E is the estimate, O is the optimistic time, P is the pessimistic time and M is the most likely time.

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

Answer: A design review is a milestone within a product development process whereby a design is evaluated against its requirements in order to verify the outcomes of previous activities and identify issues before committing to - and if need to be re-prioritize - further work.
Moderator controls the design review meeting.
The attendance at a design review depends on the level of the review. A high-level design review can include developers, the technical management of the project, users or customers, and participants from related development teams that will make use of the final product. Lower level design reviews should include the team that will be implementing the design added with others affected by aspects of the design, such as users of a networking interface or a database design.

The different metrics collected in the review meeting are:
• Do-ability: It will check feasibility aspects of data.
• Testability: It is particular method for testing. It is the degree to which a software artifact supports testing in a given test context.
• Correctness: It will verify the correctness of the data.
• Review: Review meeting forms generate the review metrics which will identify each of the reviewed problems in the form of “errors” or “defects”.
• Change Request: It will automatically be collected and entered into the project’s change management database where changes are triaged and considered for implementation.
• Additional Metrics: It gives the details about WBS activities.
• Publishing and archiving of metrics: All of the above aggregated metrics will be published and archived as part of project reduction.

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

Answer: No, reviews and inspections tasks cannot replace/eliminate the testing tasks.
Inspection is one of the most known way of review practices found in software projects. The goal of inspection process is to identify defects and check quality. It is the most formal form of reviews. The software inspection process generally has code review and peer reviews.
The inspection process and reviews cannot be replaced with the testing tasks based on the following reasons:
i) Inspections and reviews are static test technique means software work product is tested manually or with number of tools, but they are not executed. By this way one can catch syntax errors or sometimes logical errors or whether code is following the defined standards or not. Also, code is inspected or reviewed not considering the entire system.

ii) However, in testing task, author's code is tested and executed against entire system by combining various or all modulus or components of the software product.
iii) In testing task, testing of all conditions mentioned in the requirement specification document where as in code inspection or review, code is verified against the design document so there is a chance of missing few of the requirements which can be caught in the testing phase.
iv) Sometimes rework or fixing the found defects after code inspection may affect the other modulus or components of the software product. Such kind of defects can be caught during the testing phase so delivered end software product to the customer will be having minimum defects.
v) In testing phase, we also test for whether hardware and software work together to give the user the intended functionality which is also called acceptance testing. Such thing is not done in the review or inspection.
vi) There are various types of testing like performance testing, load testing, smoke testing, sanity testing, regression testing, security testing, compatibility testing etc. With various types of testing we can find maximum number of defects which are not caught during the inspection and reviews.

Q7. Which one is better, a network diagram with few Zero-Slack activities or many Zero-Slack activities? Explain

Answer: A network diagram with few zero slack activities is better when compared to a network diagram with many zero slack activities, because we can use slack time in activities to adjust, if any changes in the project plan occurs. And it can also be used to handle unexpected circumstances such as resource unavailability.

• Slack time is the difference between late finish and early finish time. It is the time that could be tolerated in the starting time or completion time of an activity without causing a delay in the completion of the project. Hence, slack time would be very helpful if there are any delays in some activities of the project.
• Progress of software development process is not entirely predictable. Hence, there will always be a possibility of occurrence of errors and delays due to internal or external factors. Thus, slack time plays a major role as the schedule can be shifted without changing the final completion date.
.
• If there are more number of zero-slack activities, then the burden to complete the project on time would be increased.
• Therefore, it is always better to have few numbers of zero-slack activities.

Q8. What are the possible actions that the project manager and review moderator might consider to take for the following outcomes of design reviews?
Rework and defect fixes turned out to require more than 80% of the original effort to write the design.

Answer: Rework and defect fixes have upper bound of 50%. Having more than 80% of rework and defect fixes mean that there are lot of defects and inconsistencies in code. This is a bad sign for a software development process.
The project manager and review moderator would immediately schedule a design team meeting and they would take necessary action against those in the team who are majorly responsible for this horrible performance. If needed, they would also allocate new resources.

Rework and defect fixes turned out to require 25% of the original effort to write the design.
Answer: Rework and defects fixes have an upper bound of 50% and lower bound of 10% which means that 25% is not bad. The design was written decently and would require some changes which is acceptable. Rework is being done efficiently. The project manager and review moderator would excuse that.

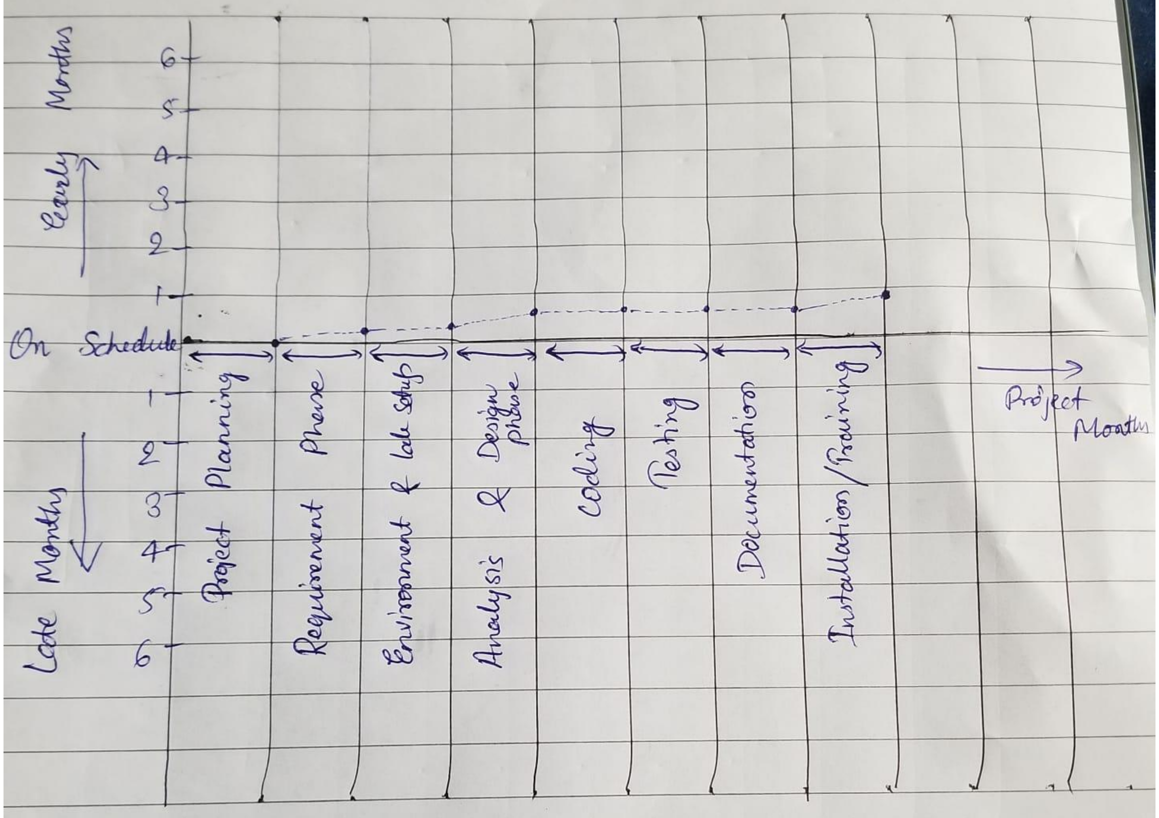
Rework and defect fixes turned out to require 5% of the original effort to write the design.
.
Answer: The lower bound of rework and defects fixes is 10%. If only 5% of the rework is required, then project manager should definitely do this rework because there will not be any change in the budget or in the

completion time of the project due to this effort. The project manager and review moderator would go through the rework. The whole rework process would be under scanner. The performance of resources who have done rework would be checked. The fixes would be reviewed closely. They would also check if all defects have been identified or not. Additionally, the efficiency of design would also be checked.

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	On time
Requirement Phase	2nd month	early 1 week
Environment and Lab setup	3rd month	early 1 week
Analysis & Design phase	4th month	early 2 week
Coding	5th month	early 2 weeks
Testing	6th month	early 2 weeks
Documentation	7th month	early 2 weeks
Installation/Training	8th month	early 3 weeks

Question: 9


Early Months
Late Months
On Schedule
6
5
4
3
2
1
1
2
3
4
5
6
Project Planning
Requirement Phase
Environment & lab setup
Analysis & Design phase
Coding
Testing
Documentation
Installation/Training
Project Months

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

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

Tasks	Amount of Work	Productivity	Effort	Duration
High Level Design (HLD)				
Write HLD Document	120 pages	4 page/Hour	30 hours/HC	3.75 days/HC
Review HLD Document				
Preparation for HLD Document		4 pages/Hour	30 hours/HC =120 hours/4 HC	3.75 days/4HC
Review Meeting		5 pages/Hour	24 hours/HC =120 hours/5HC	3 days/5HC
Rework	110 defects	4 defect/Hour	27.5 hours/HC	3.4375 days/HC
Low Level Design (LLD)				
Write LLD Document	72 pages	1 page/Hour	72 hours/HC	9 days/HC
Review LLD Document				
Preparation for LLD Document		5 pages/Hour	14.4 hours/HC =57.6 hours/4HC	1.8 days/4HC
Review Meeting		8 pages/Hour	9 hours/HC =45 hours/5HC	1.125 days/5HC
Rework	44 defects	1 defect/Hour	44 hours/HC	5.5 days/HC
Testing				
Write Test Plan	69 pages	5 pages/Hour	13.8 hours/HC	1.725 days/HC
Review Test Plan				
Preparation for Test Plan		5 pages/Hour	13.8 hours/HC = 55.2 hours/4HC	1.725 days/4HC
Review Meeting		8 pages/Hour	8.625 hours/HC = 43.125 hours/5HC	1.078 days/5HC
Rework	73 defects	5 defects/Hour	14.6 hours/HC	1.825 days/HC