CS587
Midterm Exam

**St. Name: Rangaswamy, Harshitha**

St. ID: A20400223

10 points for every question

Q1. Can the budgeted cost work scheduled, and budgeted cost work performed be used in predicting the completion date of the project? Explain

Solution: Yes, the budgeted cost work scheduled (BCWS) and the budgeted cost work performed (BCWP) can be used to predict the project completion date.
- BCWP is the budgeted cost of work that has been performed with in a scheduled task and scheduled period of time.
- In contract to BCWP, BCWS is the sum of the budget for all the work, planning and overhead, rather than the cost of the work actually performed.
- Using BCWS and BCWP we calculate schedule variance and cost variance which in turn helps in predicting the completion date of the project.
- Schedule variance (SV) is basically the difference between what work is done and was planned to be done.
- Cost variance (CV) is the difference between budgeted cost work performed and the actual cost work performed.
- The below graph provides a better understanding of the terms used and their inter relation.
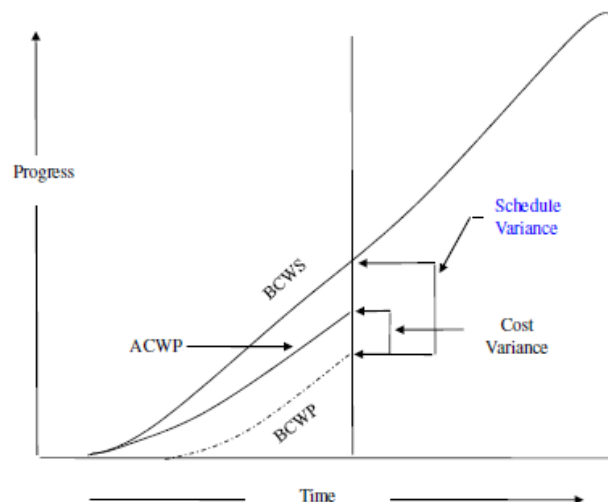
- As observed above a line parallel to y axis cuts BCWS curve at the height of BCWP. On cutting BCWS curve and pasting it on the end of the BCWP curve we can estimate the completion date of the project.

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

Solution: Software testing can begin once the application is available partially or completely. It is a process to detect any bug or defect in the application and check whether all the requirements are satisfied or not basically making the application to be robust for any kind of future failure.

It tests whether the requirements mentioned in the requirement document are all covered, functionality of a module is working as expected, how the software responds to different kinds of inputs, time taken to run a functionality, check whether it is getting installed properly in the intended environment.

Software testing is one crucial aspect to ensure the quality of the software produced because of the following reasons:
- Software testing can avoid to risk of software failure detected by the users.
- Software testing helps in checking whether the application has met all the requirement that guided its design and development process.
- Software testing can provide independent information about the quality of the software.
- Software testing is an iterative process one bug may lead to another one. On must test with all kinds of boundary values to make sure the software is robust.
- Software faults can be because of any reason like issues in infrastructure or issues in coding or due to configuration, testing will help us to know the reason for failure.
- Testing can also be used for non-functional requirements such as performance, security, usability etc.
- Even after the software is updated previous test cases can be used to check the performance of the software after the update.
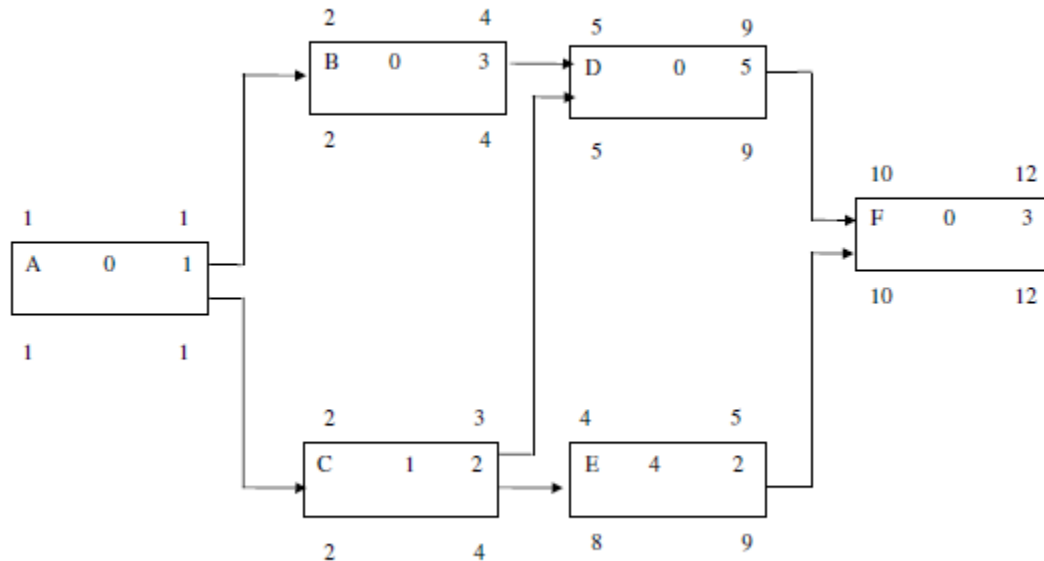
Once software testing phase is completed it makes sure the software is ready to go live and gain user's trust. As a result, software testing is the only one aspect to ensure the quality of the software produced.

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

Solution: A network diagram with few Zero-Slack activities is better than one with many Zero-Slack activities considering the factors which could cause delay in completion of activities and extending the completion date of the project.
- Slack is basically the difference between the late finish and the early finish (LF-EF).
- Slack in a project is used for various reasons. The sub-process in overall project plan can be de-optimized to run it smoothly.

- Slack checks in how many number of days an activity can be completed in the project without backing off.

	2		4		5		9		10		12
	B	0	3	→	D	0	5		F	0	3
	2		4		5		9		10		12

(Network diagram)

Node A: top-left 1, 1; center A 0 1; bottom 1, 1

Node B: top 2, 4; B 0 3; bottom 2, 4

Node D: top 5, 9; D 0 5; bottom 5, 9

Node F: top 10, 12; F 0 3; bottom 10, 12

Node C: top 2, 3; C 1 2; bottom 2, 4

Node E: top 4, 5; E 4 2; bottom 8, 9

- In the above example A has slack Zero, which means we cannot move back and forth. D has 2 predecessors B and C so EF for C is end of day 3 and EF for B is end of day 4. Hence ES for D can be beginning of day 5.
- The path which defines the finishing date of the project is called the critical path. The critical path has minimum slack. A critical path has zero buffer time. If a path has more than zero(minimum) slack time, then it is not considered to be a critical path.
- Critical path includes the set of activities which when combined gives the longest possible time for the completion of project. Such a path cannot have slack time which will extend the completion date of the project.
- Having slack in the critical path will affect the preceding tasks which will in turn delay the project finish date. As a result, in order to avoid risk task which requires more slack time should be moved from the critical path.
- Having more zero slack activities reduces the tolerance factor which is essential for a successful progression of the project. Having low tolerance would result in running the project on tight deadlines with zero flexibility.
- A zero slack time means we can have one critical path which in turn makes project to run easily with good quality and product management.

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

Solution: Requirements are the very important phase of any project. As further processes in a project are dependent on the requirement developed. Requirements must

be very clear and understandable for the developers. Requirements cannot be incomplete as they reflect the user needs. Equal importance must be given to both functional and non-functional requirements. The requirement documentation must be done in a standard language which will be understood by all the developers and analysts. If the requirements are not correct it will have an adverse effect on the rest of the project phase both in terms of price and cost. So basically, the requirement must be consistent, feasible and realistic.

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

Solution: System Engineers control the design review meeting.
The review meeting in various phases along with different metrics are listed below:
Requirement Specification Document Review
- Requirements meet customer needs
- Requirements provided by the clients must be implementable
- Check for omissions, inconsistencies, and ambiguities in the requirements

High Level Design Document Review
- High-level design implements the requirements.
- Make sure the design is practically possible to implement
- Check for omissions and other 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
- Make sure project management plan meets project management and control needs.
- Make sure the project management plan can 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.

Solution: No, the review and inspection tasks can not replace/eliminate 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. There are generally two types of reviews in software inspection process.
- Code review
- Peer review

The software project inspection and review process cannot replace/eliminate the testing tasks due to the following reasons.

- Inspection or reviews are done based on the project specifications whereas software testing is performed based on the test cases written by the test engineer.
- In inspection phase only the specifications are checked whereas in testing code is tested to find new errors.
- Inspection can only be manually done whereas testing can be done by test automation tools.
- Defects are identified in both testing and inspection phase but in testing the defects are tracked and reported immediately.
- Load test, robust test etc. are very important and can be achieved only using software testing. Similarly, it is difficult to replace unit testing and other testing with inspection because even for a small change in the code in inspection more time will have to be wasted later.

For all the reasons mentioned above it implies that both inspection/review and testing tasks must be included in order to reduce the time taken in rework which in turn reduces the cost.

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.

Solution: Three-point technique can be used for estimating activity duration in a CMM Level-3 software development organization.
- CMM Level -3 is defined, even through the organizations that fall under this level will have the process followed in projects which are previously executed and documented.
- 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 or duration estimation for the organization below CMM Level-4 which is CMM Level-3 in our case.

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.

Solution: If rework and defect fixing takes more than 60% of the effort then it implies that the coding principles and standards are not followed in the project plan. Now the code must be sent back to the development team. This will be an overhead to the manager as most of the code has to be rectified which definitely takes more time and money. In order to avoid such scenarios project manager may follow the below guidelines:
- The project manager should schedule a meeting with the design team in order to understand the reasons for the high percentage of rework.

- In order to make sure the mistakes are not repeated project manager might start the work from the beginning.
- The design document is revisited to identify the hidden flaws.
- Current resources may be replaced with highly skilled resources and previous resources may be reskilled in order to avoid such situation in the future.
- Once the design document is completed it is reviewed by other department resources.

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

Solution: In such scenario's managers should come with ideas which help in reducing rework reload. This scenario with 20% of the original effort to write design is still manageable. As the rework is not very large, it is not required to add additional resources or retrain resources. It can be resolved using the existing resources. Only the manager should make sure that the project is completed within the given time.

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

Solution: This is one of the inevitable situations in every project. Developer should take care of this minor bug detect these and correct them before passing them over to testers.

As most of the rework process generates at least 5% defects for the above scenario the project manager may follow the below steps to reduce the rework:
- The project manager might conduct review meeting with different reviewer to improve the previous design.
- Project manager can ask the document engineers to document the process and process and technique by design team for future reference.

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 weeks
Requirement Phase	3rd month	early 1 week
Analysis phase	4th month	early 2 weeks
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

Solution: The above data represents permanent schedule shift. This project probably completes early
Below is the milestone trend chart:
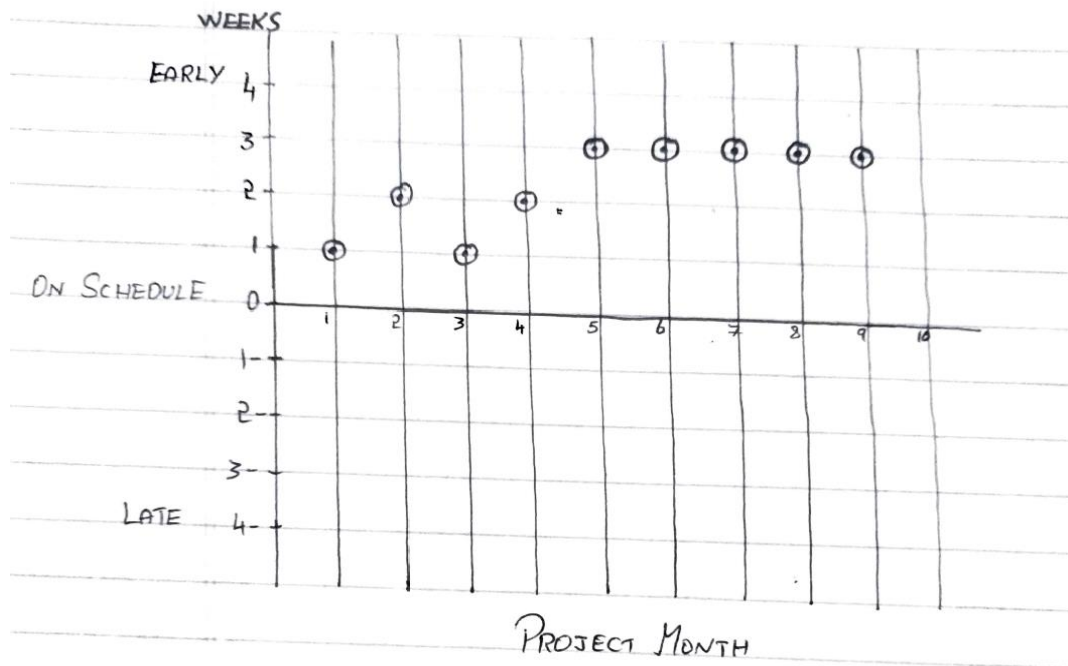
Q10. Consider the following data; calculate the effort and duration required for every task, considering the following constraints:

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				
Write Requirements Document	120 pages	1 page/Hour	15 days/1HC	15 days/1HC
Review Requirements Document				
Review Preparation for Req. Doc.		5 pages/Hour	3 days/1HC	3 days/4HC
Review Meeting		10 pages/Hour	1.5 days/1HC	1.5 days/5HC
Rework	13 defects	1 defect/Hour	1.625 days/1HC	1.625 days/1HC
Design				

Write Design Document	72 pages	1 page/Hour	9 days/1HC	9 days/1HC
Review Design Document				
Preparation for Design Document		5 pages/Hour	1.8 days/1HC	1.8 days/4HC
Review Meeting		10 pages/Hour	0.9 days/1HC	0.9 days/5HC
Rework	18 defects	1 defect/Hour	2.25 days/1HC	2.25 days/1HC
Testing				
Write Test Plan	50 pages	2 pages/Hour	3.125 days/1HC	3.125 days/1HC
Review Test Plan				
Preparation for Test Plan		5 pages/Hour	1.25 days/1HC	1.25 days/4HC
Review Meeting		10 pages/Hour	0.625 days/1HC	0.625 days/5HC
Rework	30 defects	5 defects/Hour	0.75 days/1HC	0.75 days/1HC