Quiz 1 Answer Key

Note: The question numbers are ordered as per set A

Question 1-10 (Pavitra)

1. [A] The evolutionary model breaks product development into a series of releases.
2. [B] Continuous collaboration with customers and responding to change
3. [C] It requires a fixed set of phases that must be completed in order.
4. [B] When there is uncertainty in user requirements and frequent feedback is essential.
5. [C] It can lead to incomplete documentation if not properly managed.
6. [D] User requirements are continuously gathered, evaluated, and incorporated based on feedback received after each iteration.
7. [B] It emphasizes that the accuracy of project estimates improves as more information becomes available over time
8. [C] To guide the team through the estimation process while managing group dynamics and ensuring all voices are heard.
9. [-] No correct answer. Marks for everyone.
10. [B] Strong focus on requirements elicitation and understanding of customer needs

Question 11 (Karan)

The triple constraints in project management are: (1/2 mark each)
Time/ schedule
scope/ requirements
cost/ resources
Scope defines the work to be done, time sets the deadlines, and cost determines the

budget. Any change in one constraint directly impacts the others, requiring careful trade-offs to maintain balance. Throughout the project, managers must optimize resources, mitigate risks, and adjust strategies to meet objectives without exceeding limitations (1 mark)

1/2 mark for an example or the triangle.
example:
if a software development project is initially planned to be completed in 6 months with a $50,000 budget and a defined feature set (scope), any change in one constraint affects the others. If the client requests additional features (scope increase), the project may require more time or additional budget to accommodate the changes. If the deadline is fixed, the team may need to increase resources (raising costs) or prioritize essential features to meet the schedule.

Question 12 (Pranav)

Key differences between Waterfall and Scrum development methodologies (2 Marks):

Out of the points mentioned below, students are expected to write a brief explanation for 2 points for full marks. Each point is worth 1 Mark.

1. **Explanation**: In waterfall model, all the requirements are laid out well in advance, in the form of a sequential and a linear chain, and each of the later stages follow only after the completion of the earlier stages. Scrum is an Agile methodology that focuses on incorporating iterative and incremental development, where projects are divided into sprints where the requirements/scope is modified based upon regular feedback.
2. **Incorporation of changes**: Classical Waterfall Model is idealistic, changes are very difficult to be incorporated with the progress of the project, however, scrum is highly adaptive on account of regular user feedback for the requirements.
3. **Documentation**: In waterfall model, there is heavy emphasis on documentation for chunking out all the thoughts well in advance, while scrum methodology keeps documentation minimal by focusing on the team's immediate needs.
4. **Client Feedback**: Client provides feedback in the beginning of the project for charting out the requirements of the project in waterfall model, however, in scrum methodology, regular sprint reviews adapts to the changing requirements of the project.
5. **Risk Management and Defects**: Risks get detected much later in waterfall model, while in scrum model, risks are minimised by matching client's expectations/requirements of the project via daily sprint meetings.

When we would choose Waterfall/Scrum development strategy (1 Mark):

Waterfall works best for projects where requirements are well understood in advance and unlikely to change, with the scope and timeline of the project charted out well in advance. Scrum is ideal for projects that benefit from flexibility, continuous feedback, and evolving requirements.

0.5 Marks each for writing the Use-Cases of Waterfall and Scrum development strategy.

Question 13 (Aniket)

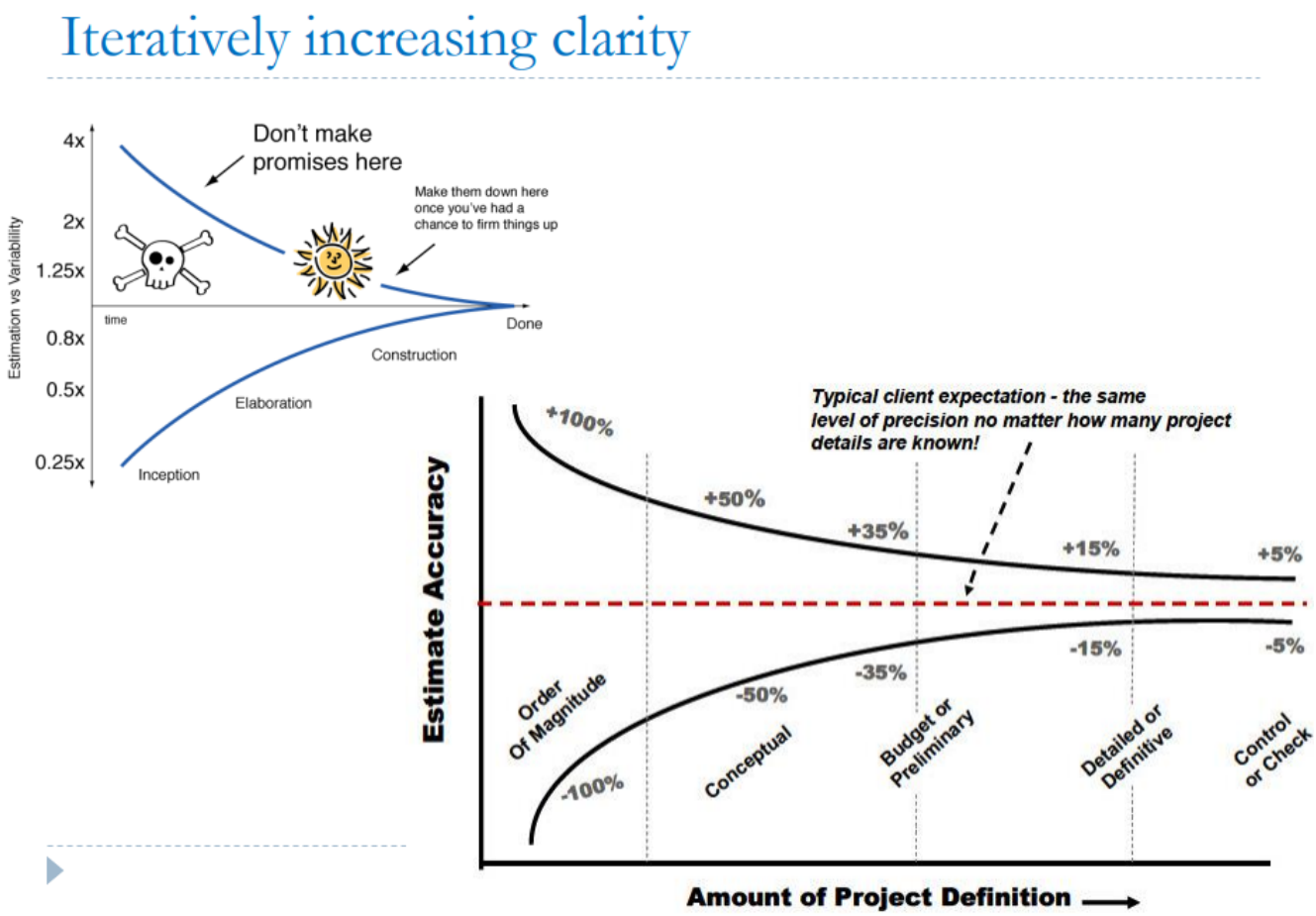


Diagram - [1 mark]

The cone of uncertainity represents the variation in the project clarity or estimations over time, showing how uncertainity decreases as more information becomes available. Initially the estimates can vary significantly (by upto 4x or 0.25x of the actual value). As the project progresses and requirements become clearer, the range of uncertainity narrows, improving estimation accuracy.

Defination - [1 mark]

How it evolves throughout a project's lifecycle:

Early Stages: During the early stages of the project when there is not much information about the requirements and other details the uncertainity is significantly high.

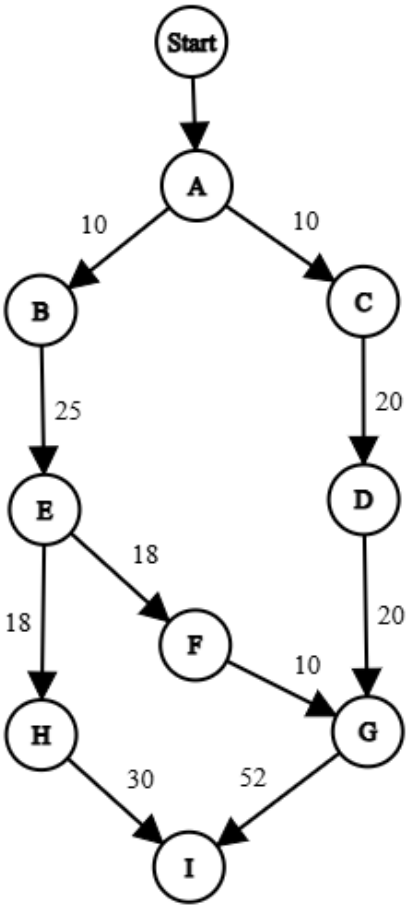
Planning and Development: As the project progresses decisions are made and the risks are identified, uncertainity decreases.

Execution & COmpletion: Uncertainity is minimized as the project reaches completion, and the estimates become accurate.

How it evolves over time - [1 mark]

Question 14 (Divij)

Part A (2 marks)



Part B (2 marks)

Maximum Time = 115 Hours

Part C (1 mark)

Task D must finish before Task G starts as G is a critical task and D is a predecessor of Task G. Task G starts at after the chain (A-B-E-F-G) is complete which takes 10 + 25 + 18 + 10 = 63 Hours.
Completing Task A and C takes time = 10 + 20 = 30 hours.
Earliest Finish Time for Task D = 30 + 20 = 50 Hours
Latest Finish time for Task D = Start time for Task G = 63 Hours

Hence, maximum slack time for Task D = 63 - 50 = 13 Hours

Part D (1 mark)

Task E is on the critical path so it has 0 slack time. The earliest and latest finish time would be the same to ensure no delay at the end.

Question 15 (Harpreet)

Correct Answer - A (3 points)

Wrong answers - any other combination (B, C, D, AB, AC, AD..)

Explain the advantages and it's disadvantages: 2 points

Advantages 1 point (any 2)

Rapid feedback: customer feedback taken into account after each iteration.
Core Functionalities: Quick delivery for functionalities that has never been offered.
Quick Fix: allow developers to fix unanticipated problems of a critical system quickly.

Disadvantages 1 point(any 2)

Bad Documentation: Due to quick changes in the system, documentation can't keep up.
Integration challenges: Combining all functionalities developed previously into one cohesive system can be challenging.
Coordination: keeping the overall system architecture consistent and coherent during testing.

NOTE: No marks for the component that you've missed writing about. 0 only for writing A, 3 only for without proper explaination wrt the context in the problem statement.