KASNEB REVISION KIT (Q&A)

ICT REVISION KIT

www.masomomsingi.com
0728 776 317

2026

MASOMO MSINGI PUBLISHERS

CPA

CS

INFORMATION COMMUNICATION TECHNOLOGY

REVISION KIT

TOPICALLY ARRANGED

Updated With December 2025 Past Paper with Answers

PART A

PAST EXAMINATION QUESTIONS

TOPIC 10

SYSTEMS ANALYSIS AND DESIGN

QUESTION 1

December 2025 Question One D

Explain **SEVEN** key experts involved in systems analysis and design. (7 marks)

QUESTION 2

December 2025 Question Two B

Highlight **FOUR** key objectives of system requirements elicitation. (4 marks)

QUESTION 3

December 2025 Question Four D

Analyse **THREE** reasons why models are used in software or systems development. (6 marks)

QUESTION 4

August 2025 Question Seven B and D

(b) Highlight FIVE advantages of using a prototype during system development. (5 marks)

(d) Describe FIVE events that might spark an iterative process in software development. (5 marks)

QUESTION 5

April 2025 Question Two B

Discuss **FOUR** techniques used to measure the effectiveness of system maintenance. (8 marks)

QUESTION 6

December 2024 Question One B

Describe **FOUR** activities that are carried out during information system implementation phase. (4 marks)

QUESTION 7

December 2024 Question Two D

Discuss **FOUR** steps carried out during system requirements elicitation process.(8 marks)

QUESTION 8

December 2024 Question Four C

PART B

SUGGESTED

ANSWERS AND SOLUTIONS

TOPIC 10

SYSTEMS ANALYSIS AND DESIGN

QUESTION 1

December 2025 Question One D

Key experts involved in systems analysis and design.

- **Systems analyst**: This individual has the primary responsibility for the analysis and design of information systems, serving as a "change agent" to identify needed organizational improvements, design systems to implement those changes, and train others to use the systems.
- **Business analyst**: Focuses on identifying how a system can improve business processes and ensure that the technical solution fulfills the business problems and needs.
- **Project manager**: Coordinates the various jobs, manages resources, scope, budget, and ensures the entire project is completed on time and within specifications.
- **Software architect/senior engineer**: Primarily involved during the design phase, focusing on technical trade-offs, data flows, system architecture, performance, and resilience.
- **User/business stakeholder**: These are the individuals who will ultimately use the system or have a vested interest in its outcomes (e.g., managers, employees, customers). They provide critical input on requirements and user experience.
- **Programmer/developer**: Converts the system specifications and design artifacts (like API contracts and database schemas) into actual computer programs.
- **Infrastructure analyst**: Focuses on the non-functional requirements and ensures that the new system is compatible with the existing technical infrastructure and standards.
- **Quality assurance (QA) specialist**: Involved in testing the system throughout the development life cycle to ensure it meets the specified requirements and quality standards.

QUESTION 2

December 2025 Question Two B

Key objectives of system requirements elicitation.

- **Clarity and understanding:** Accurate elicitation helps in clearly understanding what is expected from the system or project, reducing ambiguities.
- **Scope definition:** It helps in defining the scope of the project, ensuring that all necessary requirements are captured and irrelevant ones are excluded.
- **Stakeholder alignment:** Engaging stakeholders early ensures that their needs and expectations are met, fostering a sense of ownership and collaboration.

- **Risk reduction:** Identifying potential issues and requirements early helps in mitigating risks associated with scope creep and project delays.

QUESTION 3

December 2025 Question Four D

Reasons why models are used in software or systems development.

- **Visualize and understand complexity**: Models create abstract representations (diagrams, blueprints) that simplify complex systems, allowing developers and non-technical stakeholders to see structure, behavior, and data flow, making intricate concepts concrete and easier to grasp.
- **Enhance communication & collaboration**: Visual models serve as a shared language, fostering clear discussions, aligning expectations, and ensuring everyone from developers to clients understands the goals, progress, and design decisions, reducing misunderstandings.
- **Guide development & planning**: They provide a structured framework (like Agile or Waterfall) that defines the steps, order, and roles in the development lifecycle, helping to manage timelines, allocate resources, and build a systematic approach to creating the software.
- **Support analysis, design & quality**: Models help in analyzing requirements, specifying system architecture, verifying functionality, and validating against user needs, leading to better-designed, higher-quality, more reliable, and maintainable software by focusing on key attributes like scalability and security.

QUESTION 4

August 2025 Question Seven B and D

(b) **Highlight FIVE advantages of using a prototype during system development.**

- **Helps gain buy-in and validate ideas:** If you're still in the process of trying to obtain budget for a project, a great way to help your stakeholders visualize the product and show value early on is to start with a low-investment, low-risk prototype phase. You can put your prototype in front of real users to validate your ideas and designs to ensure the product fits a need and is feasible to build before jumping into development.
- **Aligns teams around the objective:** A successful development project requires that everyone involved has a clear picture of what needs to be accomplished. Once you get the green light on a project, your prototype will be a visual aid that ensures everyone is talking the same language and understands what is proposed and planned.
- **Encourages questions up front:** Once your developers see the prototype, the cogs in their brain will start turning as they plan how to accomplish the project. As a result, they'll formulate questions that are better asked upfront then halfway

through the development cycle. The questions could be as simple as how certain text is populated or more detailed to delve into some business knowledge you may have overlooked. Developers' questions could help identify problems or gaps with the system before it's too late and the code is already live.

- **Creates a comprehensive project plan:** Prototyping forces you to plan the details of your project in great depth. You get into the nitty-gritty, such as which button goes where and what each button does. A better understanding of the product before development gives stakeholders and developers a clear path forward so they can create a detailed and successful plan of attack.
- **Minimize surprises and code rework:** With a comprehensive project plan, developers are less likely to have surprise requirements come up, forcing code revisions. Applications can quickly become complex, involving several layers like databases, services, and a front-end interface. When a developer must refactor code, it could entail rewriting one or more of these layers and ensuring they all still work correctly together, resulting in a very time-consuming process.
- **Saves time and money:** We all know development time is costly. Starting with a prototyping phase allows you to receive accurate feedback from stakeholders and end-users early on, minimizing confusion when it's time to begin coding. Getting these answers and catching mistakes upfront is critical – they'll only cost you more in the long run.

(d) **Events that might spark an iterative process in software development.**

- **Changing or unclear requirements:** When project requirements are expected to evolve during development or are not fully defined at the outset, an iterative approach allows teams to start with high-level needs and refine details in later cycles.
- **Early user or stakeholder feedback:** Receiving feedback after developing an initial, basic version (prototype or first iteration module) helps identify issues and gather valuable insights, which are then incorporated into subsequent iterations for continuous refinement.
- **Need for early delivery of a working product:** If there is a business need to deliver a functional, albeit basic, version of the software to the market quickly, an iterative process allows for faster initial delivery and subsequent enhancements.
- **High project risks or technical complexity:** For projects involving new technologies or high-risk features, an iterative process helps identify and mitigate risks early in the development lifecycle by testing assumptions and learning as the project evolves.
- **Detection of bugs or design flaws during testing:** When testing uncovers bugs, errors, or usability issues, the feedback loop inherent in iterative development

allows for immediate rework and correction within the same or next iteration, rather than waiting until the end of the project.

- **Desire for continuous improvement and quality optimization:** The goal of achieving a more polished and reliable final product through repeated cycles of testing, evaluation, and refinement encourages the adoption of an iterative methodology.

QUESTION 5

April 2025 Question Two B

Techniques used to measure the effectiveness of system maintenance.

- **Mean Time Between Failures (MTBF):** MTBF measures the average time a system or component operates without failure. It's a key indicator of reliability. A higher MTBF indicates a more reliable system and more effective preventive maintenance.
- **Mean Time To Repair (MTTR):** MTTR measures the average time it takes to repair a failed system or component and restore it to full operational status. This includes time spent on diagnosis, actual repair, and testing. It's an indicator of maintenance efficiency.
- **System Availability (Uptime Percentage):** This metric indicates the percentage of time a system is operational and accessible to users. It's a direct measure of the system's readiness and reliability from a user's perspective.
- **Planned Maintenance Percentage (PMP) / Preventive Maintenance Compliance:** Measures the proportion of total maintenance hours spent on planned (proactive) maintenance activities versus unplanned (reactive) emergency repairs.
- **Cost of Maintenance (Total and Per Unit/Asset):** This involves tracking the total expenditure on maintenance activities, including labor, parts, software licenses for maintenance tools, and external vendor costs. It can also be broken down per system, per user, or per unit of output.
- **Backlog of Maintenance Work Orders:** This is the total volume of pending or overdue maintenance tasks. It indicates the amount of work that needs to be done.
- **User Satisfaction Surveys/Feedback:** Collecting direct feedback from end-users on their experience with system performance, reliability, and the responsiveness of IT support regarding maintenance issues.

QUESTION 6

December 2024 Question One B

Activities that are carried out during information system implementation phase.

- **System Configuration and Customization**: Adjustments are made to the system's settings and interfaces to tailor it to the specific needs of the organization. This may involve customizing modules, workflows, and user interfaces.