

SASE – Module 2 Previous Year Questions

Sem 4 BCA MGU



For Notes download BCA Resources App

1. Differentiate Program & a software (2019) [2 Marks]

Program

- A computer program that is generally used by the author of the program
- Contains little documentation or other aids to help other users to use the program
- Presence of bugs is not a major concern
- Not designed with issues as portability, reliability & usability in mind

Software

- Used by largely by people other than the developer of the system.
- Sufficient documentation to help users from different background.
- Tested before release.
- Portability is a key issue.



2. Objective of Software Engineering (2020) [2 Marks]

Develop methods and procedures for software development that can scale up for large systems & that can be used to consistently produce high-quality software at low cost & with a small cycle time & scalability.



3. If we have less domain knowledge which software lifecycle will be chosen?(2020) [2 Marks]

RAD can be suitable choices when domain knowledge is limited because they allow for incremental development, which means you can start with a basic understanding and gradually refine the software as you gather more information.



4. Differentiate product & process?(2021) [2 Marks]

- Product: What is delivered to the customer, is called a product. It may include source code, documentations etc.
- Process: The way which we produce software. Collection of activities that leads to a product.



5. What are the Life Cycle Phases?(2021) [2 Marks]

- **Requirements Gathering**: In this phase, the project team works with stakeholders, users, and clients to identify and document the requirements and expectations for the software. It involves understanding the problem, determining the scope, and defining what the software should do.
- **Analysis**: During the analysis phase, the gathered requirements are analyzed in detail to ensure that they are clear, complete, and feasible. Any inconsistencies or ambiguities are resolved, and the team creates a detailed specification or design document.
- **Design**: In the design phase, the high-level architecture and low-level design of the software are created. This phase involves defining the system's structure, components, interfaces, and algorithms to meet the specified requirements.
- **Implementation**: This phase is where the actual coding and programming take place. Developers write the code based on the design documents, following coding standards and best practices.
- **Testing**: Once the software is implemented, it goes through various testing activities to identify and fix defects. Testing ensures that the software functions correctly, meets the requirements, and is free of errors or bugs.
- **Deployment**: After the software has been thoroughly tested and is considered stable, it is deployed to the production environment or made available to end-users.
- **Maintenance**: The maintenance phase involves ongoing support, bug fixes, updates, and enhancements to the software to ensure it remains functional and up-to-date over time.



6. List the Drawbacks of waterfall model?(2021) [2 Marks]

- System development is likely to take time.
- Requirements of the system may change before the system is completed.
- Working version of the system is not seen until late in the project's life.
- Not suitable for accommodating any change



7. Write a note on project & it's associated risk in selection of lifecycle model?(2021) [5 Marks]

The selection of an appropriate software development lifecycle model is a critical decision that significantly impacts the success of a project. Each lifecycle model has its strengths and weaknesses, and choosing the right one depends on various factors, such as project size, complexity, scope, budget, timeline, team expertise, and customer requirements. However, the decision-making process is not without its risks.

- **Requirements Volatility:** Some lifecycle models, like the Waterfall model, assume stable and well-defined requirements at the project's outset. However, in reality, requirements often change and evolve throughout the project's life. Choosing a rigid lifecycle model in the face of volatile requirements may lead to delays, increased costs, or dissatisfaction among stakeholders.
- **Development Complexity:** Different projects have varying degrees of complexity. Projects involving cutting-edge technologies, innovative solutions, or a large number of stakeholders may not fit well with simpler lifecycle models like the Waterfall. Complexity can introduce risks related to uncertainties, integration challenges, and resource management.
- **Time Constraints:** Projects that have strict time constraints may find traditional sequential lifecycle models, such as Waterfall, to be inflexible. In such cases, iterative models like RAD can be more suitable as they allow for incremental development and early delivery of working software.



7. Write a note on project & it's associated risk in selection of lifecycle model?(2021) [5 Marks]

- **Resource Availability:** The availability of skilled resources and expertise in a particular lifecycle model can be a risk factor. Adopting a complex lifecycle model that requires specialized knowledge when the team lacks the necessary skills can lead to inefficiencies, errors, and delays.
- **Customer Collaboration:** Some lifecycle models, it requires frequent customer collaboration and feedback.
- **Cost and Budget Constraints:** Projects with strict budget limitations must carefully evaluate the cost implications of the chosen lifecycle model. Some models require more upfront investment or continuous iterations, which may not align with the budget available.



8. If user have no previous experience in similar projects, which all lifecycle models will be choosen. Explain it?(2021) [15 Marks]

If the user has no previous experience in similar projects, it is essential to choose a software development lifecycle model that is relatively simple, easy to understand, and provides a structured approach to software development.

- Waterfall Model

The Waterfall model is one of the most straightforward and traditional software development lifecycle models. It follows a linear and sequential approach, where each phase (requirements, design, implementation, testing, deployment, and maintenance) must be completed before moving on to the next. This model is easy to understand and provides a clear path for development. However, it lacks flexibility and does not accommodate changes well. It is suitable for projects with well-defined requirements and a stable scope.

- Iterative Model

The Iterative model is an enhancement of the Waterfall model, where development is divided into smaller cycles or iterations. Each iteration goes through the same phases as the Waterfall model, but these cycles are repeated, allowing for continuous improvement and feedback from users and stakeholders. The iterative model is suitable when the project requirements are not fully clear at the beginning or when the user needs to see progress in smaller increments.



8. If user have no previous experience in similar projects, which all lifecycle models will be choosen. Explain it?(2021) [15 Marks]

- RAD (Rapid Application Development)

RAD is a model that prioritizes rapid prototyping and user feedback. It aims to quickly develop a working prototype of the software and involves close collaboration with users and stakeholders. RAD is suitable for projects where getting early user feedback and validation is crucial.



9. Write a note on increment process model?(2021) [15 Marks]

The Incremental Process Model is a software development lifecycle model that breaks down the development process into smaller, manageable increments or parts. Each increment represents a part of the final software product and goes through the entire development lifecycle, including requirements gathering, design, implementation, testing, deployment, and maintenance. Subsequent increments build upon previous ones, adding new features and functionality to the software.

- **Iterative Development:** The Incremental Process Model is iterative in nature. It divides the development process into multiple iterations, where each iteration delivers a specific set of features or functionalities. These iterations are usually time-boxed and follow a regular release cycle.
- **Early and Frequent Delivery:** One of the main advantages of the Incremental Process Model is the early and frequent delivery of working software. Users and stakeholders get to see a functional product early in the development process, which enables them to provide feedback and validate the direction of the project.
- **Continuous Improvement:** Each increment builds upon the previous one, allowing for continuous improvement and refinement of the software. This iterative approach facilitates adapting to changing requirements and incorporating user feedback throughout the development lifecycle.



9. Write a note on increment process model?(2021) [15 Marks]

- **Reduced Time-to-Market:** By delivering working software in increments, the Incremental Process Model reduces the time it takes to bring valuable features to users. This can be particularly beneficial in projects with time constraints or rapidly changing market demands.
- **Risk Management:** The incremental nature of the model helps in risk management. Early increments serve as prototypes, allowing the team to identify and address potential risks and issues early in the development process, before they become more challenging and costly to resolve.
- **Modularity and Flexibility:** The Incremental Process Model promotes modularity in software development. Different increments can focus on specific modules or features, making it easier to manage and maintain the codebase. It also allows flexibility in the development process, enabling adjustments based on user feedback and changing priorities.
- **Customer Involvement:** The Incremental Process Model encourages active customer involvement throughout the development lifecycle. By delivering increments and soliciting feedback at regular intervals, the model ensures that the final product meets customer expectations and needs.



9. Write a note on increment process model?(2021) [15 Marks]

Some potential challenges associated with the Incremental Process Model:

- **Integration Complexity:** As each increment is developed independently, integrating different increments can be challenging, especially when there are dependencies between them.
- **Planning and Coordination:** Proper planning and coordination are crucial to ensure that increments are delivered in a logical order and that dependencies are managed effectively.
- **Documentation:** With multiple iterations, it's important to maintain clear and up-to-date documentation to avoid confusion and ensure consistency across increments.

The Incremental Process Model is well-suited for projects where requirements are subject to change, and where early and frequent delivery of working software is desired. It promotes collaboration and adaptability, making it a popular choice in many software development scenarios.



10. Compare Iterative model with RAD model.?(2022) [15 Marks]

The Iterative Model and the Rapid Application Development (RAD) Model are both software development lifecycle models that emphasize incremental and iterative development. However, they differ in certain aspects, including their approach to development, scope, and application.

Development Approach:

- Iterative Model: The Iterative Model divides the software development process into smaller iterations, with each iteration going through the entire development lifecycle (requirements, design, implementation, testing, etc.). Each iteration produces a working version of the software, and subsequent iterations build upon the previous ones, refining the product based on feedback and changing requirements.
- RAD Model: The RAD Model focuses on rapid prototyping and quick development. It involves gathering requirements, building prototypes, and obtaining user feedback to refine the product rapidly. RAD places a strong emphasis on user involvement and frequent interactions with stakeholders throughout the development process.



10. Compare Iterative model with RAD model.?(2022) [15 Marks]

Timeframe

- Iterative Model: The Iterative Model follows a regular and predictable release cycle. Each iteration typically has a fixed duration, and the development team delivers a new working version of the software at the end of each iteration.
- RAD Model: The RAD Model aims to deliver a functional product in a significantly shorter timeframe compared to traditional approaches. Rapid prototyping allows users to see a working version of the software early in the development process.

Scope

- Iterative Model: The Iterative Model is well-suited for projects with a large scope and evolving requirements. It allows for incremental development, making it easier to manage complex projects and accommodate changes in requirements over time.
- RAD Model: The RAD Model is suitable for projects with well-defined requirements and a focus on speed and early delivery. It works best when users and stakeholders have a clear understanding of what they need from the software.



10. Compare Iterative model with RAD model.?(2022) [15 Marks]

User Involvement

- Iterative Model: The Iterative Model encourages regular feedback and collaboration with users throughout the development process. Users have opportunities to see the software's progress at each iteration and provide feedback to guide further development.
- RAD Model: The RAD Model heavily emphasizes user involvement and feedback. Users are actively engaged in the prototyping phase, and their feedback drives the refinements in subsequent prototypes.

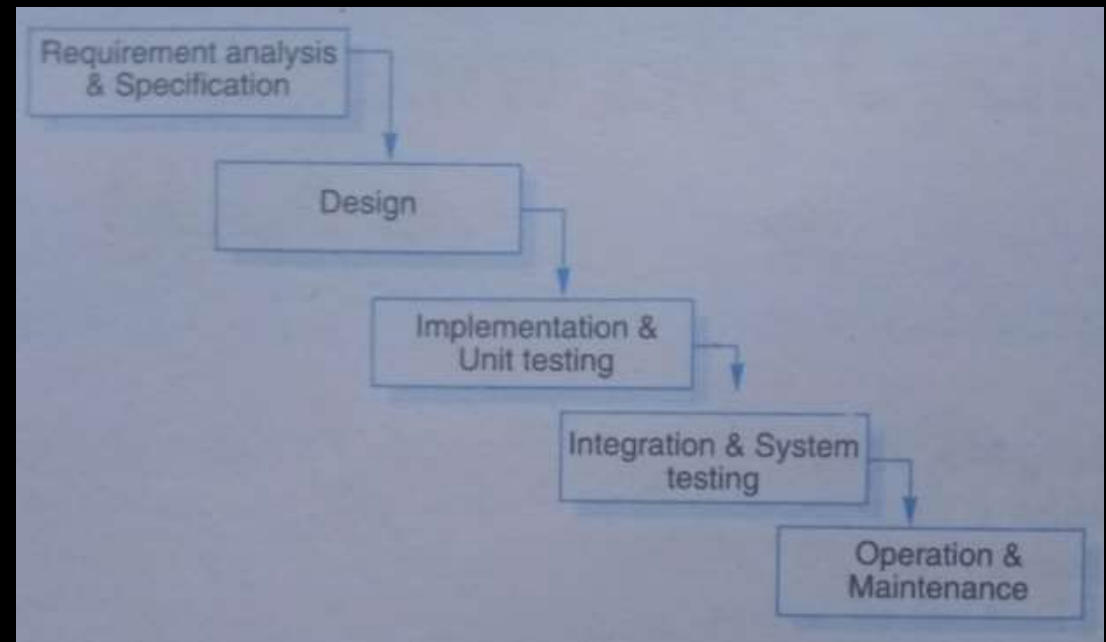
Complexity

- Iterative Model: The Iterative Model is suitable for projects of varying complexity. It is capable of handling complex projects, especially when requirements are uncertain or subject to change.
- RAD Model: The RAD Model is more suitable for projects with moderate complexity and well-understood requirements. It may not be the best fit for highly complex projects where the development team requires extensive time for analysis and design.



11. With neat diagram explain the steps in a waterfall model ?(2022) [5 Marks]

- Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- System Design – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.



11. With neat diagram explain the steps in a waterfall model ?(2022) [5 Marks]

- Integration and Testing – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

