CHAPTER 1

Introduction to Software Development

www.ilmkidunya.com

Q. What is software development?

Software Development

Software development is the process of creating computer programs to perform specific tasks or solve particular problems. Software development plays an important role in our daily life by making the tasks easier, faster and more efficient.

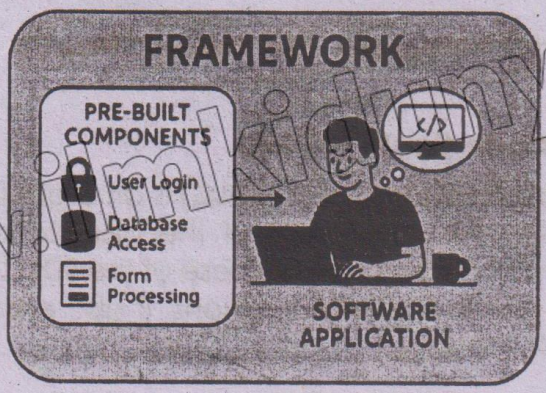
Software development involves the following main steps:

1. **Writing Code** — This step involves writing instructions for computer using a programming language to build the software.
2. **Testing the Program** — This step checks whether the software works as expected and performs all the required tasks correctly.
3. **Fixing Problems** — This step involves finding and correcting errors or bugs in the program. It is also known as **debugging**.

Q. What is meant by framework in software development? Describe the benefits of using frameworks.

Framework in Software Development

A **framework** in software development is a pre-built structure that helps the developers to build software more easily and quickly. It includes built-in tools, components and rules that provide a foundation for developing software applications. Developers can then focus on writing the specific code for their application instead of creating everything from scratch. This makes the development process faster, more organized and efficient.



Benefits of using Frameworks

Some important benefits of using framework are as follows:

1. Efficiency	Frameworks provide built-in tools and features that speed up software development. This reduces the time and effort needed to create applications.
2. Consistency	A framework offers consistency by following standard structures and coding styles. This makes the code easier to read, use and maintain.
3. Code reusability	Developers can reuse the same code in different parts of the application without rewriting it. This saves time and helps to minimize coding errors.

Example

Django is a popular framework used for web development in Python. It provides many built-in features to create websites such as user login, form handling, page templates and database management. The developers do not need to build these features from scratch. This makes website development faster and easier.

Q. What is a stakeholder? List some important stakeholders in software development.

Stakeholder

A **stakeholder** refers to any person or group who is affected by the system directly or indirectly. The stakeholders can be within the organization such as employee or outside of the organization such as customers.

Important Stakeholders

Some important stakeholders in software development are as follows:

- **End Users:** These are the people who will actually use the software.
- **Clients or Customers:** These are the individuals or organizations that request or pay for the software.
- **Project Managers:** They are responsible for planning, organizing and overseeing the project to ensure successful delivery.
- **Developers:** They design, build, test and maintain the software.
- **Business Analysts:** They understand business needs and convert them into software requirements.

Q. What is software development life cycle? Briefly discuss its different phases.

Software Development Life Cycle (SDLC)

The **Software Development Life Cycle (SDLC)** is an organized method to develop software. It begins with the initial concept of the application and continues to its deployment and maintenance. The main purpose of the SDLC is to create high-quality software that meets user needs and expectations. It also ensures that the software is delivered on time and within budget.

www.ilmkidunya.com

Phases of SDLC

The SDLC includes several phases or stages. Each phase has its own specific objectives and is used to perform relevant activities.

Different phases of SDLC are as follows:

1. Requirement Gathering

The **requirement gathering** phase is the first phase in the software development life cycle. Its main goal is to understand and collect all necessary information about the software to be developed. It involves communicating with the end users and other stakeholders to find out their needs and expectations.

2. Design

The **design** phase of SDLC is used to plan the appearance and functionality of software. It is based on requirements gathered during requirement gathering phase. The developers create diagrams, models and architecture that define the structure of the software. A good design ensures that the software is user-friendly and meets all the requirements.

3. Coding / Development Phase

In **development phase**, the actual creation of the software begins. Programmers write the code based on the specifications created during the design phase. The code is written in a programming language such as Python, Java, or C++. The instructions in the code tell the computer how to perform different tasks. Each function or feature of the software is created by writing lines of code. It translates the software design into a working software.

4. Testing

The **testing** phase is used to check the software for any bugs, errors or issues. It ensures that the software functions correctly and meets the specified requirements. Testing improves the overall quality, performance and reliability of the software.

5. Deployment

The **deployment** phase is used to make the software available for users to access and use. It takes place after the software has been fully developed and tested. This phase includes activities such as installation, configuration and testing in the real-world environment. The users can start using the software after successful deployment.

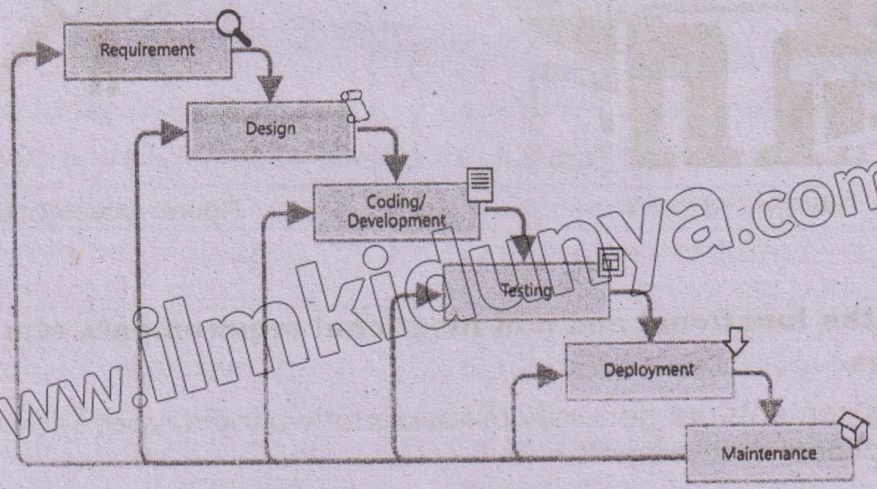


Figure: Phases of SDLC

www.ilmkidunya.com

6. Maintenance

The **maintenance** phase is used to fix bugs, make updates and add new features as user needs or technology change over time. This is an ongoing process that ensures the software continues to work correctly.

Q. Describe the requirement gathering phase of SDCL. Which key activities are performed in this phase?

Requirement Gathering

The **requirement gathering** is the first phase in **Software Development Life Cycle (SDLC)**. Its main goal is to understand and collect all necessary information about the software to be developed. It involves communicating with the end users and other stakeholders to find out their needs and expectations. This phase is important to collect accurate information so the developers can build the software according to the user needs.

Example

Suppose a new app is being developed for a school. It is important to meet with stakeholders such as principal, teachers and administrative staff. It helps to understand the required features such as attendance, grading and messaging etc.

Key Activities

The key activities performed in this phase are as follows:

- **Interviews and Surveys:** It involves asking questions and collecting feedback from the users to understand their needs and preferences.
- **Observations:** It involves watching how the users use current systems to identify problems or opportunities for improvement.
- **Document Review:** It involves studying existing reports, manuals or documents to gather additional information about the requirements.

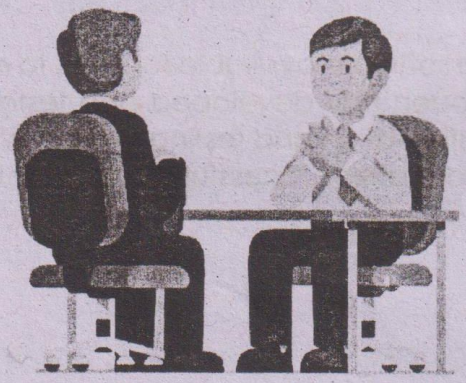


Figure: Interviews

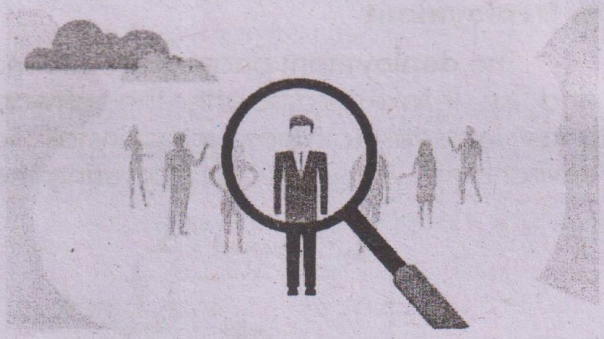


Figure: Observations

Q. Explain the functional and non-functional requirements of a system with examples.

The requirements are generally divided into two main types known as **functional** and **non-functional** requirements.

www.ilmkidunya.com

1. Functional Requirements

Functional requirements describe the specific behaviors or functions of a system. These requirements specify what a system should do. It includes the tasks, services and functionalities that should be performed by the system. These requirements define the interactions between the system and its users or other systems.

Example

Some functional requirements for a library management system are as follows:

- **User Registration:** The system should allow users such as students and faculty to register and create an account.
- **Book Borrowing:** The system should allow users to search and borrow the books.
- **Inventory Management:** Librarians should be able to add, update and remove books from the inventory.

2. Non-Functional Requirements

Non-functional requirements specify how the system performs a function rather than what it should do. They define the quality attributes and performance standards that the system must meet such as:

- **Performance:** It refers to how fast and efficiently the system operates.
- **Reliability:** It ensures that the system operates consistently and is available when needed.
- **Security:** It ensures that the system protects user data from any unauthorized access.
- **Usability:** It means that the system is easy to use and understand.
- **Constraints:** It defines limits or conditions under which the system must operate.

Example

Some non-functional requirements for library management system are as follows:

- **Performance:** The system should support up to 1,000 users at the same time without performance issues.
- **Reliability:** The system should be available 99.9% of the time to ensure high availability and minimum downtime.
- **Security:** The system should protect user data and allow only the authorized users to log in.
- **Usability:** The system should have simple and user-friendly interface.
- **Constraints:** The system runs on Windows operating system and requires at least 4 GB RAM.

Q. Differentiate between functional and non-functional requirements.

The difference between functional and non-functional requirements is as follows:

Functional Requirements	Non-Functional Requirements
1. Functional requirements describe what the system should do.	1. Non-functional requirements describe how the system should perform.
2. They define specific functions, features or tasks to be performed by the system.	2. They define the quality attributes such as speed, security and reliability.
3. They are directly related to the user interactions and system operations.	3. These are related to system performance, reliability and user experience.

www.ilmkidunya.com

Q. What is design phase of SDCL? Describe different activities performed in this phase.

Design Phase

The design phase of SDLC is used to plan the appearance and functionality of software. It is based on requirements gathered during requirement gathering phase.

Activities in Design Phase

Design phase includes different activities to ensure that the final software is well-organized, user-friendly and meets the user needs. Different activities are as follows:

1. Create Diagrams

The diagrams are created to show how different parts of the software will connect and work together. It helps to visualize the flow of data and processes within the system. For example, a flowchart can be created to show the steps of the program to complete a task.

2. Develop Models

The models are developed to represent the structure and features of the software. It may include creating mockups of the user interface. A **mockup** is a visual draft of the user interface that shows what the software will look and how the users will interact with it. For example, a model can show buttons, menus or input fields.

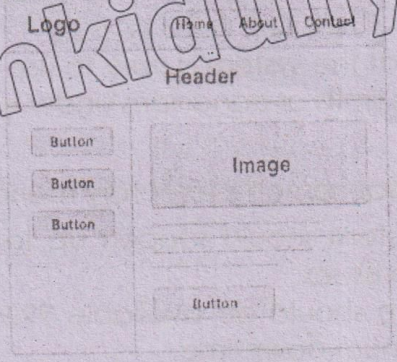


Figure: Mockup

3. Plan the Architecture

It defines the overall structure of a software system. It determines how different components of the software will work together. It ensures that the software is well-organized and functions smoothly.

4. Specify Requirements

This involves clearly defining the expected functionality and behavior of each part of software. It ensures that every feature is properly planned and documented. It reduces the chances of missing key details during development.

Q. What is the purpose of testing phase of SDCL? Explain common types of testing performed in this phase.

Testing

The testing phase is used to check the software for any bugs, errors or issues. It ensures that the software functions correctly and meets the specified requirements. Testing improves the overall quality, performance and reliability of the software.

www.ilmkidunya.com

The common types of testing performed in this phase are as follows:

1. Functionality Testing

Functionality testing checks whether a software works according to its defined requirements and features. It ensures that each function of the application behaves as expected when given the correct input. For example, a web application should allow the user to login successfully using correct username and password. The functionality testing ensures that this process works properly.

2. Performance Testing

Performance testing checks how well a software performs under various conditions such as high user traffic or heavy data load. For example, a website can be tested if it loads quickly even when 10,000 users access it at the same time.

3. Compatibility Testing

Compatibility testing checks whether a software application works correctly across different devices, browsers and operating systems. For example, a mobile app should be tested to verify if it runs smoothly on both Android and iPhone.

Q. What is the purpose of deployment phase of SDCL? Briefly discuss different steps involved in this phase.

Deployment

The **deployment** phase is used to make the software available for users to access and use. It takes place after the software has been fully developed and tested. This phase includes activities such as installation, configuration and testing in the real-world environment. The users can start using the software after successful deployment.

Steps in Deployment

Deployment often includes the following steps:

1. Installation

The software is installed on the user's system or server. This may involve running an installation program that copies and sets up the necessary files.

2. Configuration

The software is customized according to the specific needs of the user or organization. It can include setting up user preferences, network settings and database connections.

3. Testing in the Real World

The software is tested in its real-world environment after installation. It ensures that it works correctly with other systems and meets user needs.

Q. What are software development methodologies and software process models? Discuss the importance of software process models.

Software Development Methodologies

Software development methodologies are structured approaches used to plan, develop and manage software projects. These methodologies ensure that the development process is systematic, efficient and produces high-quality software.

www.ilmkidunya.com

Software Process Models

Software process models represent the structured processes involved in the software development lifecycle. They provide a framework for planning, organizing and controlling the development of software. These models help the development teams to follow a consistent approach to ensure quality and efficiency.

Importance of Software Process Models

The software process models are important as they provide the following:

- **Predictability:** Software development teams can predict the outcomes and manage risks more effectively by following a defined process.
- **Efficiency:** The structured methodologies make the development process smooth. It helps to minimize the wasted time and effort.
- **Quality:** The process models ensure that quality assurance practices are followed throughout the development lifecycle. It leads to better and more reliable software.

Q. What is waterfall model in software development? Discuss its phases, benefits and limitations.

Waterfall Model

Waterfall Model is one of the earliest and simplest software development process models. It follows a linear and sequential approach. It means that each phase must be completed before the next one begins without moving back to the previous phase. This model is easy to understand and manage because of its clear and structured flow. The Waterfall Model is best suited for small projects with fixed and well-defined requirements.

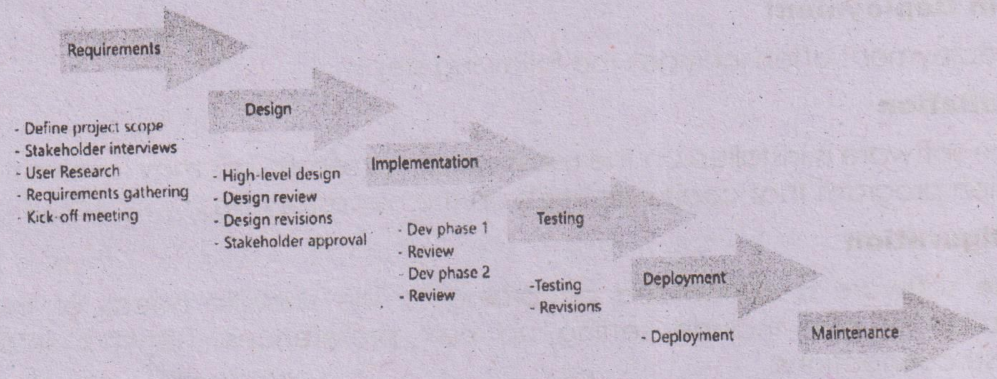


Figure: Waterfall model

Phases of Waterfall Model

The main phases of the Waterfall Model are as follows:

- **Requirements:** It is used is to collect and document the details about what the software needs to do.
- **Design:** It is used to plan the appearance and functionality of software.
- **Implementation:** It is used to write the actual code to create the software.
- **Testing:** It is used to find and fix any problems or bugs in the software.
- **Deployment:** It is used to make software available for users to access and use.
- **Maintenance:** It is used to make updates and fix any issues that occur after the software is in use.

www.ilmkidunya.com

Benefits

The benefits of the Waterfall Model are as follows:

1. **Simple and Easy to Understand:** The Waterfall Model is easy to follow because it has clear and distinct phases.
2. **Sequential Process:** Each phase is completed one at a time that makes it easier to manage and track progress.
3. **Suitable for Small Projects:** It works well for small projects where requirements are fixed and unlikely to change.

Limitations

The limitations of the Waterfall Model are as follows:

1. **Inflexibility:** The Waterfall model is not very flexible. It is difficult and costly to go back and make changes once a phase is completed.
2. **Not Ideal for Complex Projects:** This model is not ideal for complex projects with changing or evolving requirements. It can be challenging to use it effectively for large projects.
3. **Risk and Uncertainty:** This model assumes that all requirements are known from the beginning. It can be risky if new needs or unexpected issues arise later in the process.

Q. What is agile methodology in software development? Discuss its practices, benefits and limitations.

Agile Methodology

Agile methodology is a flexible and adaptive approach to software development. It focuses on delivering small and functional parts of the software quickly instead of building the entire software at once. This approach allows teams to adapt to changes as the project progresses. Agile is useful for projects with changing or evolving requirements.

Agile divides the development process into short cycles called **iterations** or **sprints**. A small part of the software is developed and tested in each sprint. The team then gathers feedback from users and makes improvements in the next sprint. This cycle repeats until the complete software is developed.

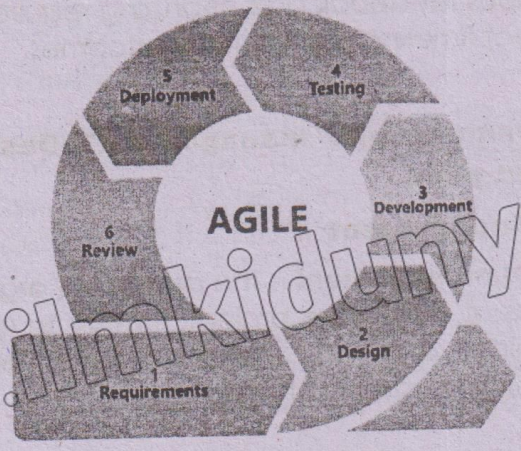


Figure: Agile methodology

www.ilmkidunya.com

Practices in Agile Methodology

Agile include the following practices:

1. **Continuous Integration:** It is a key practice in Agile where developers frequently integrate the code changes into a shared repository. This repository is a shared location where entire project's code is stored and managed. Each integration of code is automatically tested using different tools. This process allows multiple developers to collaborate effectively and helps detect errors early through automated testing.
2. **Test-Driven Development:** It involves writing the tests before writing the code to ensure the software works as expected. It ensures that the software works as expected and meets all specified requirements.
3. **Pair Programming:** It is a practice where two developers work together on the same task using a single computer. One developer writes the code while the other reviews it in real-time. This method improves code quality and helps developers to learn from each other.

Benefits

The benefits of Agile methodology are as follows:

1. **High Flexibility:** Agile is highly flexible and allows teams to make changes at any stage of the development based on feedback or new requirements.
2. **Improved Customer Satisfaction:** Customers get regular updates and working versions of the software. It allows them to see progress and give feedback to ensure that the final product meets their needs and expectations.
3. **Improved Product Quality:** The testing is done continuously during the development. It helps to find and fix the problems early that results in improved product quality.

Limitations

The limitations of Agile methodology are as follows:

1. **Scaling Challenges:** It is difficult to manage large projects with multiple teams using Agile as it requires effective coordination and communication.
2. **Stakeholder Involvement:** Agile requires active participation from all stakeholders. The process can slow down if some of the stakeholders are unavailable or not fully involved.
3. **Less Predictable:** Agile is considered less predictable because projects evolve based on continuous feedback and changing requirements. It can be harder to predict the exact timeline, cost or final outcomes.

Q. What is project planning and management? Describe five phases of project management plan.

Project Planning and Management

Project planning and management is the process of organizing, scheduling and controlling all the activities of a software project. It helps to ensure that the project is completed on time, within budget and meets its goals. The project planning and management process typically consists of five phases.

www.ilmkidunya.com

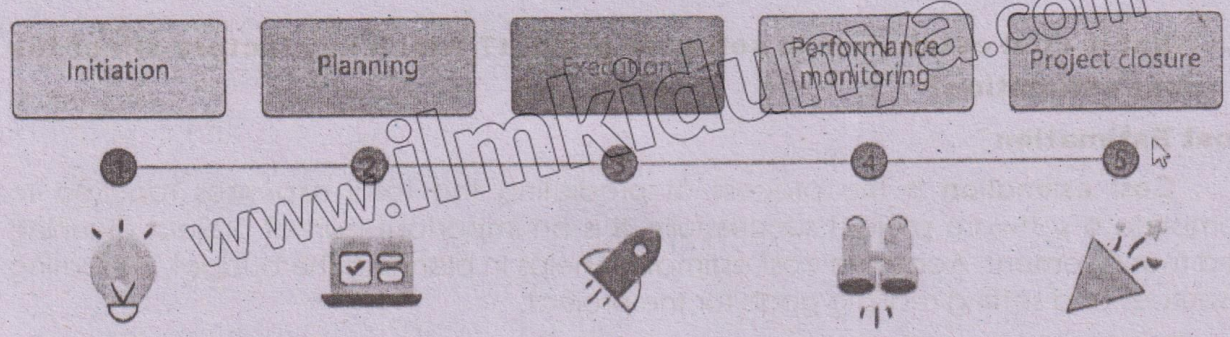


Figure: Phases of Project management plan

1. Initiation

The **initiation** phase is the starting point of a project. It is used to clearly define the goal or purpose of the project. The project team checks if the project is useful and possible to complete.

2. Planning

The **planning** phase is used to create a detailed plan that outlines how the project will proceed. This includes estimating the required resources, setting timeline and calculating budget. A good plan helps the team to work in an organized way.

3. Execution

The **execution** phase is where the actual development work takes place. The team writes code, creates designs and builds the software according to project plan.

4. Performance Monitoring

The **performance monitoring** phase is used to check project's progress regularly. The team compares actual work with the planned work to see if the project is on schedule and within budget. If any issues are found, they are fixed early to keep the project on track.

5. Project Closure

The **project closure** phase is used to complete all tasks and deliver the final product to the client. This is the final phase of the project management.

Q. Briefly discuss comprehensive project planning and setting project timelines in software project.

Comprehensive Project Planning

Comprehensive project planning refers to carefully thinking about every part of the project before starting the work. This includes understanding what tasks need to be done, who will do each task and how the work will be completed. A proper planning increases the chances of completing the project successfully.

Setting Project Timelines

Setting project timelines involves estimating the time taken by each part of the project. It helps the team stay organized and ensures the project is completed on time. Timelines are essential for project planning and management.

www.ilmkidunya.com

Q. What is cost estimation in software project? Which key factors affect the cost estimation?

Cost Estimation

Cost estimation is the process of predicting the total expenses required to complete a software project successfully. It is an important part of project planning and management. Accurate cost estimation helps in planning the budget, allocating resources and setting realistic goals for the project.

Key Factors in Cost Estimation

The key factors in cost estimation are as follows:

1. Development Team

The cost of development team depends on the number of developers, their expertise and hourly rates. The experienced developers usually charge higher rates than beginners. The selection of right team helps to keep the project within budget while maintaining quality.

2. Technology Stack

The technology stack affects project cost based on the tools, programming languages and platforms used. Some tools are free while others need to be purchased. Some latest or complex technologies may require experts that increases overall cost.

3. Project Duration

The longer projects usually cost more because the team and equipment are needed for a longer time. It increases the risk of delays that may increase the cost.

4. Risk Management

Risk management is the process of identifying and evaluating potential problems that may affect the success of a software project. These risks may include delays, cost overruns or technical issues. Dealing with such problems often requires extra money and careful planning. A **contingency budget** is often included in the project estimate to handle unexpected issues without stopping the work.

5. Quality Assurance

Quality assurance refers to the process of testing the software, fixing bugs and making improvements to ensure that it meets required quality standards. These activities require skilled personnel, time and additional resources that can increase the overall cost of the project.

Q. What is risk assessment and management in software projects? Discuss the steps used in this process.

Risk Assessment and Management

Risk assessment and management are important aspects of a software project. They involve the following:

- Identifying potential risks that may impact the project's success
- Analyzing the possibility and impact of these risks
- Developing the strategies to manage these risks

www.ilmkidunya.com

Steps in Risk Assessment and Management

Different steps in risk assessment and management are as follows:

1. Identify Risks

The project team lists all possible risks that may affect the project. These risks may include the following:

- Technical risks such as rapid changes in technology
- Operational risks such as staff shortages or equipment failure
- External risks such as market changes or natural disasters

Example

Suppose the team plans to use a new development technology. It is a technical risk because the team has never used the technology before.

2. Analyze Risks

Evaluate the possibility of each risk occurring and its potential impact on the project.

Example

The new development technology can be difficult to learn. The possibility of the risk is high and the impact is medium as it can delay in project delivery.

3. Develop Mitigation Strategies

A mitigation strategy is a plan that reduces the possibility of a significant risk or minimizes its impact. It ensures that potential problems are managed effectively to avoid disruption to the project. It may include the following:

- Adding extra time to the project schedule to handle unexpected issues
- Arranging backup staff or resources
- Conducting additional testing to identify and fix issues early

Example

The impact of using a new development technology can be reduced by providing training sessions and hiring an expert.

4. Monitor and Review

Risk management is an ongoing activity throughout the project. The team should regularly monitor the project for new risks and review existing ones to adjust strategies as necessary. It helps to keep the project on track.

Q. What is meant by quality assurance in software projects?

Quality Assurance

Quality assurance ensures that a software project meets the defined standards and works correctly. It involves the following methods:

1. Testing

It involves evaluating a software to ensure it works correctly, meets the user requirements and is free from bugs.

2. Reviewing Code

It involves checking the code written by developers to ensure that it is correct and follows the coding standards.

www.ilmkidunya.com

3. Getting Feedback

It is the process of receiving opinions or evaluations from the stakeholders. It helps to improve and maintain the product quality.

4. Regularly Checking Project's Progress

It involves monitoring project's progress compared to the planned timeline. It identifies delays or risks and helps to make better decisions.

Q. What is meant by graphical representation of software systems?

Graphical Representation of Software Systems

Graphical representation of software systems involves using visual diagrams to show different components of a system and how they interact with each other. This approach helps in simplifying the complex systems. It makes it easier for developers and stakeholders to understand, communicate and manage the system. The diagrams like **UML (Unified Modelling Language)** are commonly used for this purpose.

Q. What is UML? How can UML be used to represent software systems?

UML

UML stands for **Unified Modeling Language**. It is a standard way to visualize the design of a software system. UML helps developers to understand how the system works and how its components communicate with each other.

Using UML to Represent Software Systems

UML can be used in various stages of software development to improve the understanding and communication among team members.

Some practical applications of UML are as follows:

1. Planning

UML can be used to visualize and analyze the system before writing the code. It helps to create a clear plan and reduces misunderstandings.

2. Development

The developers use UML diagrams during the development phase to understand the structure of the system and the relationships between its components.

3. Communication

UML diagrams help both technical and non-technical stakeholders to visualize how the system works. This makes it easier to explain system behavior and functionality to everyone.

Q. What is a Use Case diagram in UML? Discuss its elements and give an example.

Use Case Diagram

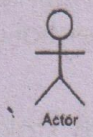
A **Use Case diagram** is a type of UML diagram that shows the functionality of a system from the user's point of view. It helps to identify what the system should do and how users interact with it.

www.ilmkidunya.com

Different elements in a Use Case diagram are as follows:

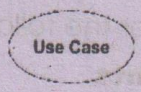
1. Actor

An **actor** represents a user or another system that interacts with the system. It can be a human such as customer or an external system such as payment gateway (PayPal). An actor is represented by a stick figure in Use Case diagrams.



2. Use case

A **Use Case** describes how an actor interacts with a system to achieve a specific goal. Each use case represents a complete task or action to be performed by a user. A use case is shown as an oval shape in a Use Case diagram. The name of the function or task is written inside the oval.



3. Association

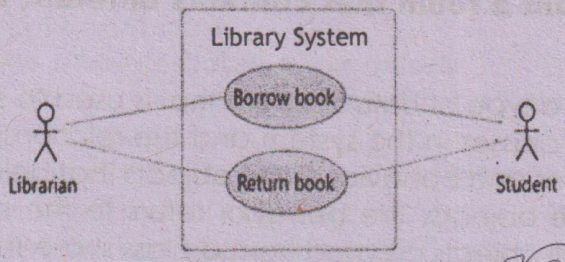
An **association** is a line that connects an actor to a use case. It shows the interaction between the user and the system. An actor can be linked to one or more use cases.

4. System Boundary

A **system boundary** defines the scope of the system. It is represented by a rectangle. All use cases are placed inside while actors are placed outside the system boundary.

Example

The following Use Case diagram shows a simple Library System. It shows how two different users interact with the system. The rectangle represents the system boundary of the library system. The **Librarian** and **Student** are actors while **Borrow book** and **Return book** are use cases. The **lines** connecting actors to use cases show interactions between the users and the system.



Q. What are the steps involved to identify Use Cases in Use Case diagram?

Identifying Use Cases

Identifying use cases is an important step and it involves the following steps:

1. Identify Actors

Determine different types of users who will interact with the system. It can be a human such as customer or an external system such as payment gateway.

www.ilmkidunya.com

2. Define Goals

Identify what each actor wants to achieve using the system. These goals will guide the creation of use cases.

3. Outline Interactions

Describe how each actor will interact with the system to achieve the goal. Each meaningful interaction is considered a potential use case.

4. Validate Use Cases

Review the identified use cases with stakeholders to ensure they accurately represent system's required features and user interactions.

Q. Briefly describe some uses of Use Case diagrams.

Uses of Use Case Diagrams

The use case diagrams are used for the following purposes:

1. Capturing Functional Requirements

Use case diagrams help to identify and document what functions the system is expected to perform. They focus on the main functions that must be performed by the system to meet the user requirements.

2. Understanding User Interactions

Use case diagrams identify who will use the system and how different users will interact with it. This helps the developers to clearly define user roles and their respective system activities.

3. Planning and Testing

Use case diagrams help in planning the development process by outlining the expected system behavior. They also help in designing test cases to ensure that all functions work as expected.

Q. Describe class diagram and its sections. Give an example of a class diagram to represent a room that contains different types of boxes.

Class Diagram

A **Class Diagram** is a type of UML diagram that is used to show the structure of a system. It represents the classes in the system and the relationships between them. A **class** defines the attributes and behavior of the objects that belong to it. The attributes are the properties of the objects. The behavior refers to the methods or operations performed by objects. All objects created from a class share the same attributes and methods. A class in UML is represented using a rectangle divided into three sections:

1. **Top Section – Class Name:** This section shows the name of class such as Room.
2. **Middle Section – Attributes:** This section shows the attributes or characteristics that describe the class. For example, the **Room** class may have attributes like **name** and **size**.
3. **Bottom Section – Methods:** This section shows the functions or operations that objects of the class can perform. For example, the **Room** class may include the methods **open()** and **close()**.

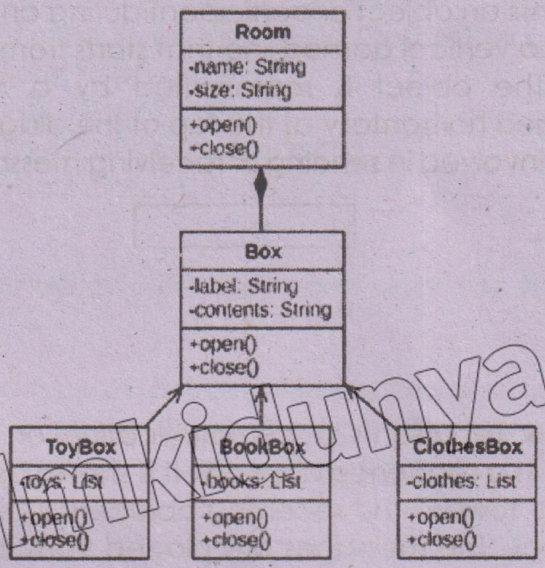
www.ilmkidunya.com

Example

A room represents the overall space with all other elements. A box represents a container within the room that contains specific items. A room can have different types of boxes to store different items such as toy box, book box and clothes box etc.

- The room can be represented as main class **Room**.
- The box can be represented as another class **Box** that is part of **Room** class.
- The toy box, book box and cloth box can be represented as three classes **ToyBox**, **BookBox** and **ClothesBox**. These classes are inherited from **Box** class.

This example can be represented in UML as follows:



Explanation

The above example has various classes as follows:

- **Room:**
  - It has two attributes **name** and **size**.
  - It can perform two operations **open()** and **close()**.
- **Box:** The **Box** class has the composition relationship with **Room** class. That means Box class is part of **Room** class.
  - It has two attributes **label** and **contents**.
  - It can perform two operations **open()** and **close()**.
- **ToyBox:** It is a child class of **Box** class. This box is used to store toys.
  - It inherits two attributes **label** and **contents** from Box class.
  - It has another attribute **toys** that indicates a list of toys in the box.
  - It can perform two operations **open()** and **close()**.
- **BookBox:** It is a child class of **Box** class. This box is used to store books.
  - It inherits two attributes **label** and **contents** from Box class.
  - It has another attribute **books** that indicates a list of books in the box.
  - It can perform two operations **open()** and **close()**.
- **ClothesBox:** It is a child class of **Box** class. This box is used to store clothes.
  - It inherits two attributes **label** and **contents** from Box class.
  - It has another attribute **clothes** that indicates a list of clothes in the box.
  - It can perform two operations **open()** and **close()**.

www.ilmkidunya.com

Q. Describe sequence diagram and its elements with the help of example.

Sequence Diagram

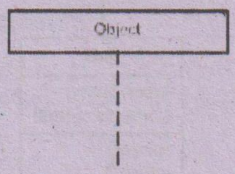
Sequence diagram is a type of UML diagram that show how objects in a system interact with each other over time in a specific sequence. They help in understanding the flow of messages between objects.

Elements in Sequence Diagram

Some important elements used in sequence diagrams are as follows:

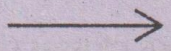
1. Lifeline

A **lifeline** represents an object or participant during an interaction in a sequence diagram. It is shown as a vertical dashed line that starts from the object at the top and extends downward. The object is represented by a rectangle. All objects or participants are arranged horizontally at the top of the diagram. A lifeline shows when and how participant is involved in sending or receiving messages during an interaction.



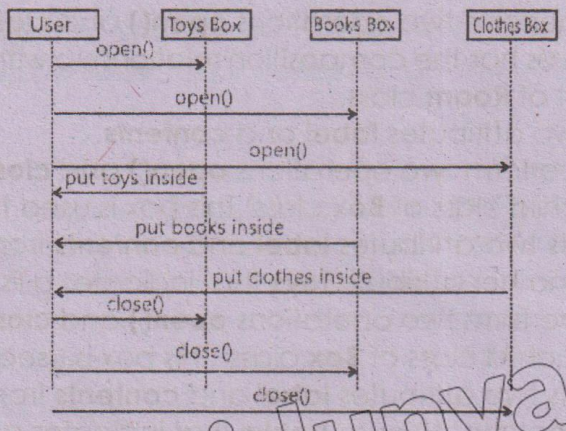
2. Message Arrow

A **message arrow** represents a communication or interaction between two objects or participants in a sequence diagram. It is shown as a solid horizontal line with an arrowhead pointing toward the receiving object. It indicates the direction of the message or function call. The messages are placed from top to bottom to show the order in which they occur.



Example

An example of sequence diagram is as follows:



The above example contains the following:

Objects

The four lifelines indicate four objects as follows:

- **User:** It represents the person who performs all the actions.
- **ToysBox:** It represents the object where toys are stored.
- **BooksBox:** It represents the object where books are stored.

www.ilmkidunya.com

- **ClothesBox:** It represents the object where clothes are stored.

Interactions

Different interactions in the system are as follows:

1. The **User** performs three **open()** operations:
   - The first **open()** method opens **ToysBox**.
   - The second **open()** method opens **BooksBox**.
   - The third **open()** method opens **ClothesBox**.
2. The **User** puts items after all boxes are open:
   - The **putToys()** method puts the toys in **ToysBox**.
   - The **putBooks()** method puts the books in **BooksBox**.
   - The **putClothes()** method puts the clothes in **ClothesBox**.
3. The **User** performs three **close()** operations:
   - The first **close()** method closes **ToysBox**.
   - The second **close ()** method closes **BooksBox**.
   - The third **close ()** method closes **ClothesBox**.

Q. Describe activity diagram and its elements with the help of example.

Activity Diagram

An **Activity Diagram** is a type of UML diagram that represents the flow of activities or actions in a system or process. It provides a step-by-step visualization of how tasks are carried out and in what order they occur. They are useful to represent the logic of complex operations.

Elements in Activity Diagram

Some important elements used in activity diagrams are as follows:

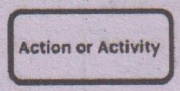
1. Start

The **Start** node indicates the beginning of the process. It is represented by a solid filled black circle (●). Every activity diagram must contain exactly one start node.



2. Activity (Action)

An **Activity** or **Action** represents a specific task or step within the process. It is shown as a rounded rectangle with the action name written inside it. Different activities are connected to each other using control flow arrows.



3. Control Flow

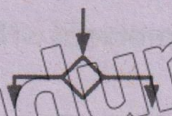
A **Control Flow** shows the direction in which the process flows from one action to the next. It is represented by a solid arrow (→) that connects different actions.



4. Decision

A **Decision** node is used to represent a point where the flow divides based on a condition. It is represented by a **diamond shape (◊).** It has one incoming flow and multiple outgoing flows. These branches are usually labeled with conditions such as "Yes" or "No".

www.ilmkidunya.com



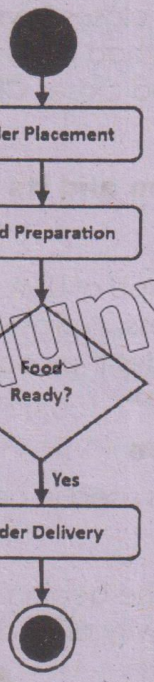
4. Final

The **Final** node indicates the end of the activity or process. It is represented by a circle with a black dot inside (◉).



Example

The following example shows an activity diagram of a restaurant management system. It represents the process from Order Placement to Food Preparation and finally to Order Delivery.



Q. Differentiate between sequence diagram and activity diagram.

The difference between sequence diagram and activity diagram is as follows:

Sequence Diagram	Activity Diagram
1. Sequence diagram shows how the objects interact with each other over the time.	1. Activity diagram shows the flow of activities or steps involved in a process.
2. It focuses on object interactions and message flow.	2. It focuses on the control flow of operations or actions.
3. It represents sequence of messages exchanged between objects.	3. It represents the sequence of actions or decisions in a process.
4. Its key elements include objects, lifelines and messages.	4. Its key elements include Start, Action, Decision, Control flows and Final.
5. It is suitable to show how different parts of a system communicate.	5. It is suitable to show the step-by-step logic of an operation.

www.ilmkidunya.com

Q. What is flowchart and what symbols are used to draw flowcharts? Give an example.

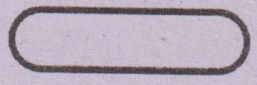
Flowchart

A flowchart is a diagram that represents the sequence of steps or actions in a process or system. It uses standard symbols to show the flow of control from one activity to another.

Flowchart Symbols

Flowcharts use different symbols to represent different types of actions or steps. The text is also used in the symbols to explain the process clearly. The commonly used symbols for drawing flowcharts are as follows:

1. Terminal

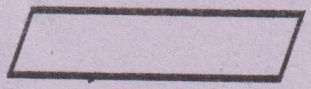
The **oval** is used to represent the start or end of the flowchart. 

Example

The following example displays the start and end steps in the flowchart:

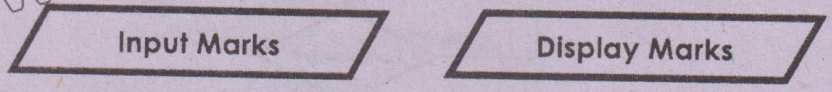


2. Input / Output

The **parallelogram** is used to represent an input or output step in the flowchart. 

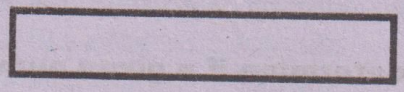
Example

The following example displays the input and output steps in the flowchart:



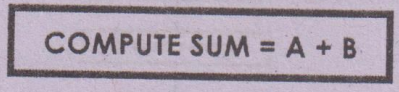
3. Process

The **rectangle** is used to represent a specific tasks, operation or action in the flowchart.



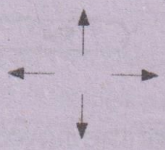
Example

The following example displays the processing step in the flowchart:



4. Flow Lines

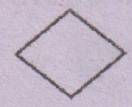
The **flow lines** are the arrows that represent the direction of flow in a flowchart. There are four flow lines to represent four directions. Different symbols in a flowchart are connected using flow lines that show the flow of different steps.



www.ilmkidunya.com

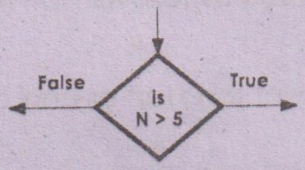
5. Decision

The **diamond** is used to represent decision step in the flowchart. A condition is given in the diamond. The flow of control from diamond may go in two possible directions. It goes to one direction if the condition is **True** or **Yes** and to second direction if the condition is **False** or **No**.



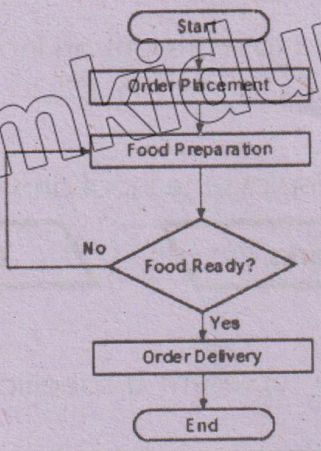
Example

The following example displays the decision step in the flowchart:

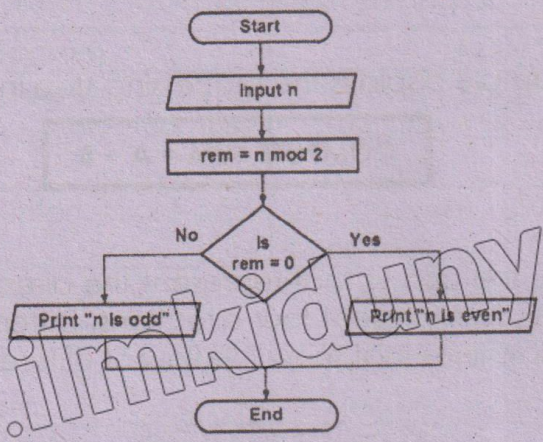


Example

The following example shows a flowchart of a restaurant management system. It represents the process from Order Placement to Food Preparation and finally to Order Delivery.



Q. Draw a flowchart to determine if a given number is odd or even.



Explanation

Step 1: It indicates the start of the flowchart.

Step 2: It shows that a number is input in **n**.

www.ilmkidunya.com

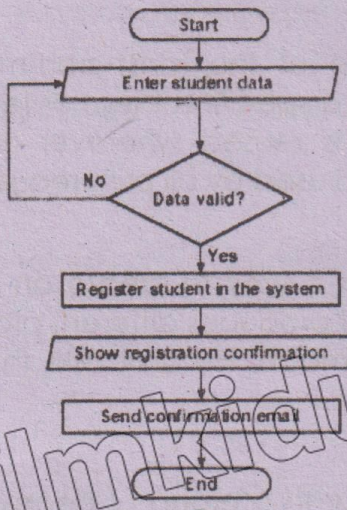
Step 3: It shows that the number **n** is divided by 2 and the remainder is stored in **rem**. The operator **mod** is used to get the remainder.

Step 4: The condition checks if the value of **rem** is 0. The result of the condition will be **true** if the value of **rem** is 0. It means that the value of **n** is even and the message "n is even" will appear. The result of the condition will be **false** if the value of **n** is odd and the message "n is odd" will appear.

Step 5: It indicates the end of the flowchart.

Q. Design a flowchart for user registration process in software application. Outline its key steps.

Student registration



Key Steps in Student Registration

The key steps in the student registration are as follows:

1. **Start**: The registration process begins.
2. **Enter student data**: The user is asked to input student information such as name, ID, email, etc.
3. **Validate data**: The system checks if the entered data is complete and valid.
   - If the data is not valid, the system asks the user to re-enter data.
   - If the data is valid, the process continues.
4. **Register student in the system**: The system saves the valid student data.
5. **Show registration confirmation**: A message is displayed confirming successful registration.
6. **Send confirmation email**: The system sends a confirmation email to the student's email address.
7. **End**: The process is complete

Q. Explain the concept of design patterns and their applications in software design.

Design Patterns

Design patterns are common solutions to problem in software development. They work like templates to make the process of coding easier, faster and more consistent. Many popular software frameworks and libraries are built using design patterns. For example, the **Model-View-Controller (MVC)** pattern is used in web development frameworks like **Ruby on Rails** and **Angular**.

www.ilmkidunya.com

Applications of Design Patterns in Software Design

Different applications of design patterns in software design are as follows:

- They reduce code complexity by providing a clear structure.
- Developers can apply proven and tested design patterns across different parts of a system or reuse them in future projects to save time and effort.
- They provide a common vocabulary among developers to make it easier to explain and discuss system design and architecture.

Q. What are the commonly used design patterns in software development?

Some of the most commonly used design patterns are as follows:

1. Singleton Pattern

Singleton Pattern is one of the most common design patterns in software development. It ensures that a specific object or resource is created only once in a program. The same object is reused wherever required. For example, a printer manager is created once and used by all print requests.

2. Factory Pattern

Factory Pattern provides a way to create an object without showing how it is created. It is like a factory that produces different products. The user does not need to know how the products are made. He only asks the factory what is needed and it produces the finished product.

3. Observer Pattern

Observer Pattern is similar to a group of people who want to get updates from one particular source. The source automatically notifies all the interested people whenever something important happens. The interested people are called **observers**. For example, a YouTube channel sends notifications to all subscribers when a new video is uploaded.

4. Strategy Pattern

Strategy Pattern is like having a toolbox full of different tools where each tool is designed for a specific task. The user picks the right tool from box depending on the problem. Similarly, the system selects the most suitable strategy based on the user requirement. Suppose a payment system has different options like credit card, PayPal or bank transfer. The system selects the appropriate option based on the user's choice.

Q. What is debugging in software development? What tools and practices can help to identify and fix bugs effectively?

Debugging

Debugging is the process of finding and fixing bugs or errors in a software. **Bugs** are errors or mistakes in the code that cause the software to behave unexpectedly or produce incorrect results. The bugs can be identified by observing software's behavior and finding the source of the problem. The code is corrected to remove the bugs.

Tools and Best Practices

There are several useful tools and best practices that can be used to identify and fix bugs effectively.

www.ilmkidunya.com

1. Debuggers

Debuggers are the tools that help developers to find and fix bugs in the code. They allow developers to go through the code step by step, inspect variable values and monitor program execution. This process helps to identify the exact location of a bug and understand its cause.

2. Print Statements

This involves adding print statements in the code to display the values of variables or messages during program execution. This technique is useful to understand what the program is doing and where it may be wrong.

3. Code Reviews

Code review involves other developers carefully read and review the code. This process helps to identify possible errors and improves the overall quality of the code.

Q. What is testing and its objectives? Describe different levels of software testing.

Testing

Testing is the process of evaluating software to ensure it meets the defined requirements and functions correctly. It checks whether the software produces the desired results and performs well in real-world conditions.

Objectives of Testing

Some important objectives of testing are as follows:

- Testing helps to identify bugs before the software is released.
- It improves the overall quality and performance of the software.
- It ensures that the software is reliable, secure and user-friendly.

Levels of Software Testing

The hierarchy of testing starts with testing individual components and gradually moves toward testing the complete system. The main levels of testing are as follows:

1. Unit Testing

It is the first level of testing where individual components or modules of the software are tested. Each unit is a small part of the software such as a function or method. The primary goal of unit testing is to verify that each component works correctly according to its design and performs as expected.

2. Integration Testing

Integration Testing is conducted after unit testing to evaluate how different components or modules interact with each other. It checks whether all the units work together correctly when combined. The main purpose is to ensure that data is properly passed between modules and their integration does not cause any errors.

3. System Testing

System Testing is a higher level of testing where the entire software system is tested as a whole. The goal is to evaluate the software's overall functionality, performance and security to ensure that it meets specified requirements. For example, the system testing of an e-commerce application involves checking the complete shopping process.

www.ilmkidunya.com

The tester verifies that the user can:

- Search for products
- View product details
- Add selected items to the shopping cart
- Proceed to checkout
- Make a secure payment
- Receive an order confirmation message or email.

4. Acceptance Testing

Acceptance Testing is performed to determine whether the software is ready for release. It is often performed by the end-users or clients to ensure that the software meets their expectations and requirements. For example, a school admin tests a newly developed student management system to check if it provides student registration, fee tracking and attendance etc.

Q. What are software development tools? Give some examples.

Software Development Tools

Software development tools are programs or applications that assist in various stages of software creation. These tools are important because they help developers to write, edit, test, debug and manage code efficiently. They make the development process faster, more organized and reduce the chances of errors. These tools ensure that the final software works correctly and meets user requirements.

Examples

Some examples of software development tools are:

- **Language Editors** such as Notepad++ and Visual Studio Code
- **Debuggers** such as GDB and Visual Studio Debugger
- **Translators** such as compilers and interpreters
- **Integrated Development Environments (IDEs)** such as Visual Studio and PyCharm

Q. Explain the role of language editors and translators in software development. Provide examples of each.

Software development involves writing code in a programming language and then converting that code into a form that can be understood by the computer. Two important tools that assist in this process are **language editors** and **translators**.

1. Language Editors

A **language editor** is also known as **code editor**. The language editors are used by developers to write and edit code in various programming languages. Most language editors provide various features like syntax highlighting, auto-completion, line numbering and error detection to make coding easier and faster.

Some examples of language editors are as follows:

- **Notepad++**: It is a simple, lightweight and powerful code editor. It is commonly used by beginners to write and edit code easily.
- **VS Code**: It is a popular code editor developed by **Microsoft**. It supports many programming languages and offers many features for coding.

www.ilmkidunya.com

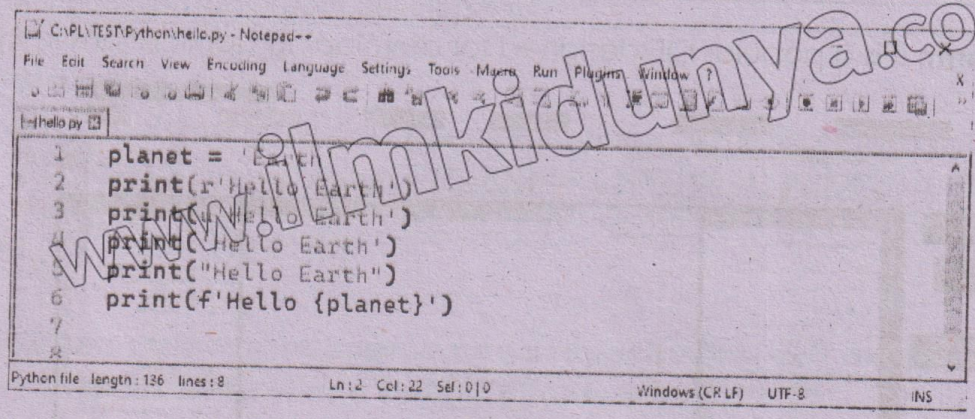


Figure: Notepad++ language editor

2. Translators

Translators are the tools that convert code written in high-level programming languages into machine language. The computer can understand and execute the machine code. A program written in high level language cannot be executed directly on the computer. It must be converted into machine language to execute.

There are two main types of translators:

- **Interpreter:** An interpreter translates and executes code line by line. The execution stops immediately if an error is found. Python is a programming language that uses an interpreter.
- **Compiler**: A compiler translates the entire program into machine code at once before execution begins. It checks the code for errors and then generates the executable file. For example, the **GCC (GNU Compiler Collection)** is commonly used to translate C or C++ programs.

Debuggers

Debuggers are the tools that help developers to find and fix bugs or errors in the code. They allow developers to go through the code step by step, inspect variable values and monitor program execution. This process helps to identify the exact location of an error and understand its cause.

Some examples of debuggers are as follows:

- **GDB (GNU Debugger):** It is commonly used for C and C++ programs.
- **Visual Studio Debugger:** It is a built-in debugger in Visual Studio IDE.

Q. What is an Integrated Development Environment (IDE)? Give examples of some common IDEs.

Integrated Development Environments (IDEs)

An **Integrated Development Environment (IDE)** is a software suite that provides all the essential tools for software development in one place. Different tools available in an IDE include code editors, compilers and debuggers to help developers write, test and debug code more efficiently.

Common IDEs

Two common IDEs are as follows:

- **Visual Studio:** It is a widely used IDE for developing applications in **.NET, C++,** and **C#**. It is available as part of the Visual Studio IDE.

www.ilmkidunya.com

- **PyCharm:** It is a popular IDE designed for developing applications in **Python**.

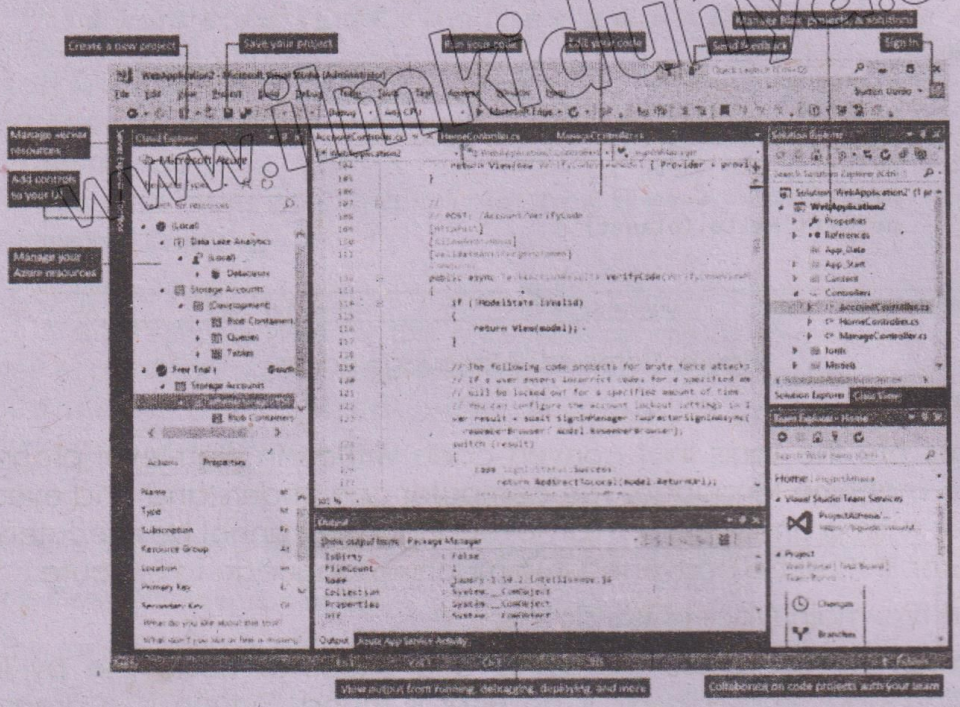


Figure: Visual Studio IDE

Q. What are online and offline computing platforms and how do they differ in terms of development environment and usage?

Online and Offline Computing Platforms

Computing platforms provide the environments where developers can write, run and test code. These platforms can be either online or offline.

Online Platforms

The online platforms are cloud-based development environments that are accessible through the Internet using a web browser. The developers can write, run and test code directly in a web browser without installing any software on computer. Some examples of online platforms are **Repl.it** and **Gitpod**.

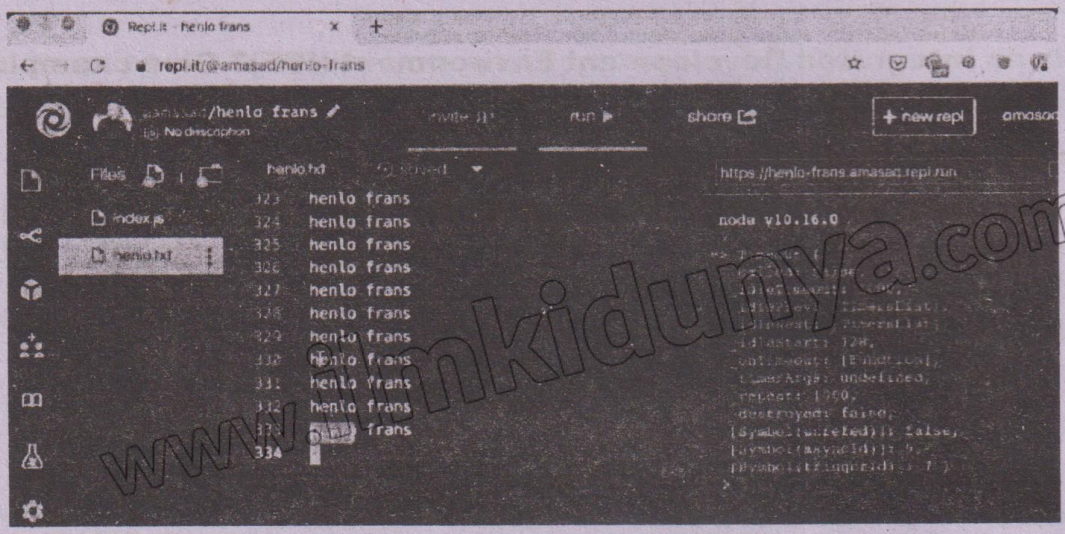


Figure: Online platform Rep.it

www.ilmkidunya.com

Offline Platforms

The offline platforms are local development environments installed directly on a computer. The developers can write, run and test code using the tools available in IDEs without connecting to the internet. Some examples of offline platforms are **Visual Studio Code**, **PyCharm** and **Eclipse**.

Q. Describe the purpose of source code repositories and explain how they support version control and team collaboration.

Source Code Repositories

A **source code repository** is a centralized platform or storage location where developers can save, manage and organize the code for a software project. Source code repositories work with version control systems to keep a record of every change made to the code over time. Repositories make it easier for teams to collaborate on the same project. Multiple developers can work on different parts of the code at the same time without interfering with each other's work. This ensures smooth teamwork and helps to avoid code conflicts during development.

Examples

Some examples of source code repositories are as follows:

- **GitHub:** It is a popular platform for open-source projects.
- **Bitbucket:** It is mainly used by development teams and organizations. It supports both private and public repositories. A **private repository** is accessible to specific users and hidden from the public. A **public repository** is open to everyone and allows anyone to view, download or contribute to the code.

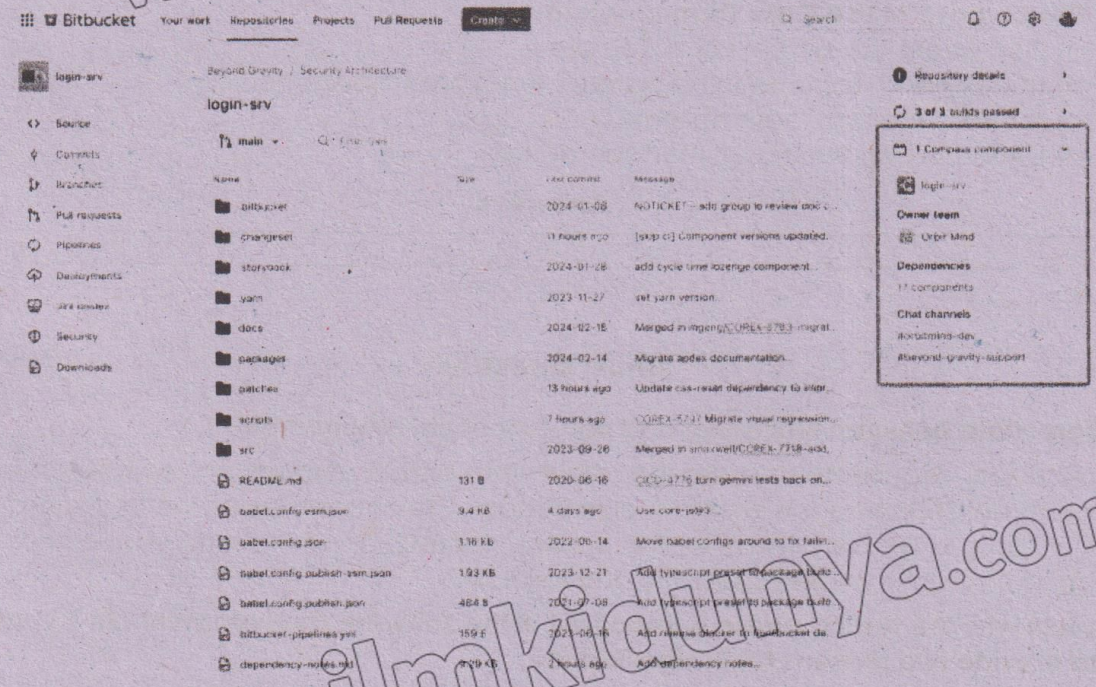


Figure: BitBucket interface

www.ilmkidunya.com

Exercise Solution

Multiple-Choice Questions (MCQs)

1. **Primary purpose of the Software Development Life Cycle (SDLC) is to:**
   a) Design websites
   b) Deliver high-quality software within time and cost estimates
   c) Manage database systems
   d) Create hardware components
2. **A type of requirement specifying system performance is:**
   a) Functional Requirements
   b) Non-Functional Requirements
   c) Technical Requirements
   d) Operational Requirements
3. **Role of a framework in the context of SDLC is to:**
   a) Write code from scratch
   b) Provide a structured foundation with predefined components and architectures
   c) Manage hardware
   d) Perform manual testing
4. **Software development model involving short cycles or sprints:**
   a) Waterfall Model
   b) Agile Methodology
   c) Lean Software Development
   d) Scrum
5. **Crucial aspect of comprehensive project planning:**
   a) Understanding the project scope and tasks
   b) Deciding the project's colour scheme
   c) Hiring a large development team
   d) Ignoring potential risks
6. **Factor that does not influence cost estimation of a software project:**
   a) Scope of the project
   b) Technology stack
   c) Number of meetings held
   d) Operational costs
7. **The purpose of Use Case Diagrams is to:**
   a) Document the system's architecture
   b) Identify and document the system's functional requirements
   c) Illustrate the database schema
   d) Define the system's user interface design

Answers

1. b	2. b	3. b	4. b	5. a	6. c
7. b					

Short Questions

Q.1. Differentiate between functional and non-functional requirements.

Functional requirements describe what the system should do while non-functional requirements describe how the system should perform. Functional requirements define functions, features or tasks to be performed by the system whereas non-functional requirements define quality attributes such as speed, security and reliability.

Q.2. Explain why the testing phase is important in the Software Development Life Cycle (SDLC), and provide two reasons for its significance.

The testing phase is important in SDLC because it ensures that software works correctly and meets the specified requirements. The first reasons for its signficance is that it helps to identify and fix bugs before the software is released to reduce the risk of failure. The second reasons is that it ensures that the software meets the expected quality standards including functionality, performance, security, and user experience.

www.ilmkidunya.com

Q.3. Illustrate the concept of continuous integration in Agile Methodology and discuss its importance in software development.

Continuous integration is a key practice in Agile where developers frequently merge code changes into a shared repository. This shared location stores entire project's code and allows for regular updates from all team members. Each integration is automatically tested using different tools that helps to detect errors early in the development process. This improves code quality and team coordination.

Q.4. Evaluate the main steps involved in risk assessment and management, and assess their importance in a software project.

The main steps in risk assessment and management include identifying potential risks, analyzing their impact, developing mitigation strategies and monitoring risks throughout the project. These steps are important because they help project teams to anticipate problems, minimize disruptions and allocate resources properly. Risk assessment and management allows teams to avoid delays, control costs and improve the overall success of the software project.

Q.5. Explain the purpose of a Use Case Diagram in software development.

The purpose of a Use Case Diagram is to visually represent the functionality of a software system from user's point of view. It helps to identify what the system should do and how users interact with it. This makes it easier to understand the system's behavior and define the required features clearly.

Q.6. Compare and contrast a sequence diagram with an activity diagram, highlighting the key differences.

Sequence diagram shows how objects interact over time by exchanging messages. It focuses on object interactions and message flow. Activity diagram shows the flow of activities or steps in a process. It focuses on control flow and decisions. Sequence diagrams are best for showing communication between different parts of the system whereas activity diagrams are ideal for showing step-by-step logic.

Q.7. Describe the Factory pattern and explain how it differs from directly creating objects, with an example.

Factory pattern provides a way to create an object without showing how it is created. It works like a factory that produces different products. The user only asks for the product and gets the result without knowing the creation process. For example, the user asks Vehicle factory to give him a vehicle for family travel. The factory gives a car based on the user's request. The user does not need to build the car or find the parts or assemble anything.

Long Questions

Q.1. Design a flowchart for a user registration process in a software application. Outline its key steps. (See the chapter for answer)

Q.2. Imagine you are managing a project to develop a simple mobile application. Describe how you would use the Agile Methodology to handle this project.

1. Requirements

In Agile, requirements are gathered continuously by working closely with stakeholders and users. I will define key features like user login, notifications and activity tracking as user stories and add them to product backlog. Each user story describes a desired feature from the user's point of view such as "I want to log in to the app so I can access my account". These stories are reviewed, prioritized and updated regularly based on feedback and feasibility.

www.ilmkidunya.com

2. Design

Agile uses an iterative design approach, which means the design improves with each sprint. I would work with a designer to create mockups or simple screen layouts for the features being developed in that sprint. This ensures that the app is user-friendly and allows for continuous improvements based on feedback.

3. Development

During each sprint, the development team selects user stories from the backlog and starts building the features. Agile focuses on delivering small, working parts of the app regularly. I will use the tools that allow the app to run on both Android and iOS platforms.

4. Testing

Testing is not a separate phase but an integral part of each sprint. Each feature is tested using manual and automated testing. For mobile apps, this includes checking whether the features work correctly, the design is responsive and the app performs well on different devices and screen sizes.

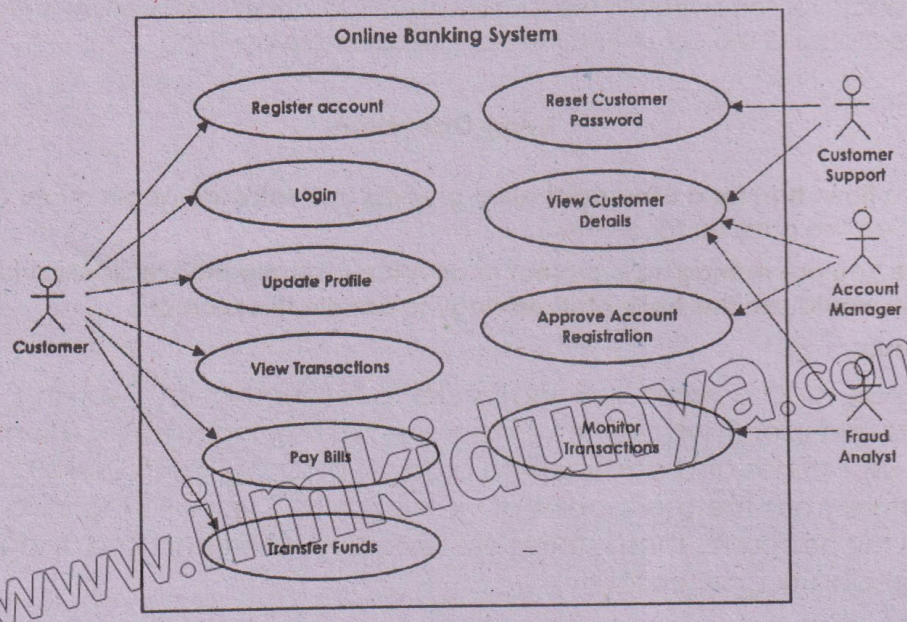
5. Deployment

Agile supports frequent and incremental deployments. After successful testing, I will prepare the completed features for release. This involves internal releases to get feedback from stakeholder or direct deployment of updates to platforms like Google Play Store and Apple App Store.

6. Review

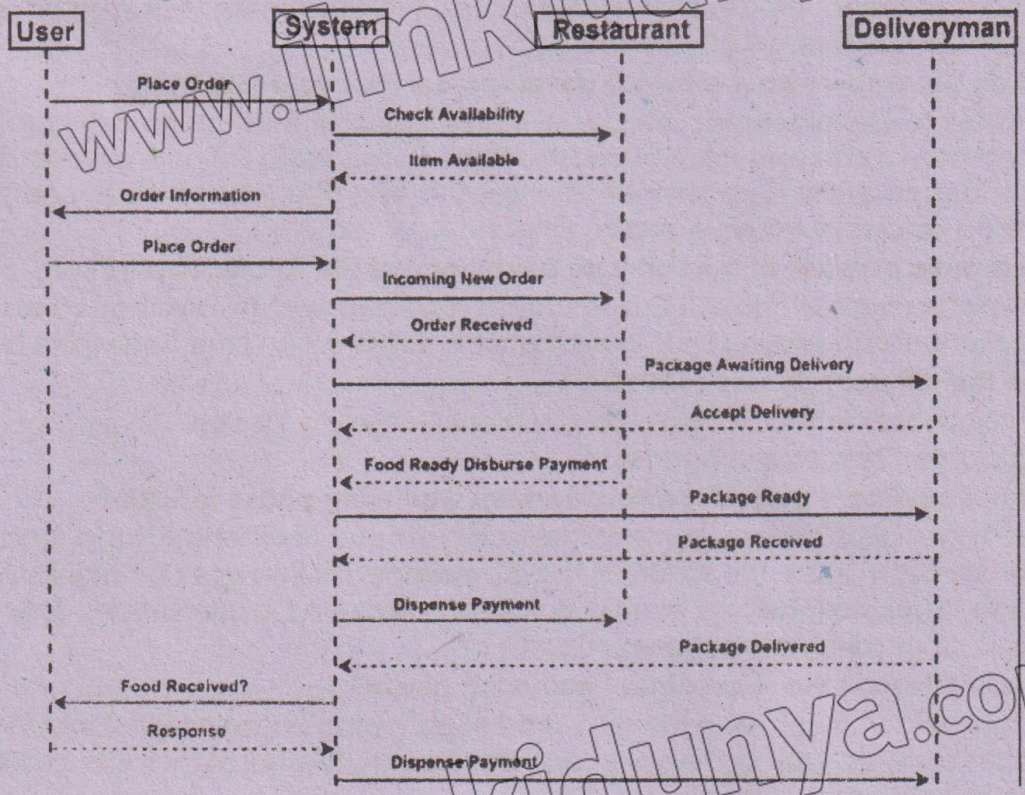
At the end of each sprint, I will hold a Sprint Review to present the new features to stakeholders and gather their feedback. This will ensure that we are on the right track. I will also conduct a **Sprint Retrospective** to discuss what went well, what went wrong and how we can improve in the next sprint.

Q.3. Consider an online banking system. Create a Use Case Diagram to show the interactions between customers, bank staff, and the system.



www.ilmkidunya.com

Q.4. You are developing a food delivery application. Create a Sequence Diagram to show the process of placing an order, from customer selecting items to the delivery of the order.



Q.5. Discuss the importance of software development tools in the software development process. (See the chapter for answer)

a) Explain the role of language editors, translators, and debuggers in creating and maintaining software.

b) Provide examples of each tool and describe how they contribute to the efficiency and accuracy of software development.

SHORT QUESTIONS

Q.1. What is software development?

Software development is the process of creating computer programs to perform specific tasks or solve particular problems. It involves writing code, testing it and fixing any issues.

Q.2. What is a framework in software development?

A framework is a pre-built structure that provides tools, components and rules to help developers build software applications. Developers can then focus on writing the specific code for their application instead of creating everything from scratch. This speeds up and organizes the development process.

Q.3. How do frameworks help developers during software development?

Frameworks allow developers to focus on writing code specific to their application. They eliminate the need to build common features from scratch. This saves time and simplifies the development process.

Q.4. What is code reusability and how do frameworks support it?

Code reusability means using the same code in different parts of the application without rewriting it. Frameworks support this by providing reusable components and modules. This saves time and helps to minimize coding errors.

www.ilmkidunya.com

Q.5. Give an example of a software framework and its use.

Django is a popular framework used for web development in Python. It provides many built-in features to create websites such as user login, form handling, page templates and database management. The developers do not need to build these features from scratch. This makes website development faster and easier.

Q.6. What do you know about software development life cycle (SDLC)?

The Software Development Life Cycle is an organized method to develop high-quality software. It consists of different steps or phases. Each phase in SDLC has a specific purpose and is used to perform relevant activities. These phases ensure that the software meets user needs and is delivered efficiently, reliably and on time.

Q.7. What is main purpose of the Software Development Life Cycle (SDLC)?

The main purpose of the SDLC is to create high-quality software that meets user needs and expectations. It also ensures that the software is delivered on time and within budget.

Q.8. Name the different phases of SDLC.

Different phases of SDLC include Requirement Gathering, Design, Development (Coding), Testing, Deployment and maintenance.

Q.9. What is the primary goal of the Requirement Gathering phase in SDLC?

The primary goal of requirement gathering phase is to understand and collect all necessary information about the software to be developed. It involves communicating with the end users and other stakeholders to find out their needs and expectations. This ensures the software is built according to what users require.

Q.10. Who is involved in the requirement gathering phase?

The stakeholders involved in the requirement gathering phase of SDLC include end-users, clients, business analysts, project manager and developers. They all participate in gathering and clarifying the software requirements.

Q.11. What are the key activities involved in Requirement Gathering phase?

The key activities involved in requirement gathering phase include interviews, surveys, observation, and document review. Each activity helps to gather detailed and reliable information about user needs.

Q.12. How do interviews differ from observations in requirement gathering?

Interviews involve direct conversations with the users to understand their needs and preferences. Observations involve watching users interacting with current system to identify problems and opportunities for improvement. Observations are useful to know what users actually do while interviews are useful to know the users needs.

Q.13. How does document review support requirement gathering?

Document review involves studying existing manuals, reports or documents. It provides background information and helps to clarify how the current system works. This helps in gathering accurate requirements for the new system.

Q.14. How is requirement gathering similar to event planning?

Requirement gathering is similar to event planning. For example, the event planner of a wedding needs to know the preferences of the bride and groom. Similarly, the developers need to know the users requirements to create a useful software.

Q.15. What are functional requirements in software development?

Functional requirements describe the specific behaviors or functions of a system. These requirements specify what a system should do. It includes the tasks, services and functionalities that should be performed by the system. Examples include user registration, book borrowing and inventory management.

Q.16. Give two examples of functional requirements for a Library Management System.

Two examples of functional requirements for a library management system include User Registration and Book Borrowing. User Registration means that the system should allow users to register and create an account. Book Borrowing means the system should allow the registered users to borrow available books by entering their user ID and book details.

www.ilmkidunya.com

Q.17. What are non-functional requirements in software development?

Non-functional requirements specify how the system performs a function rather than what it should do. They define quality attributes and performance standards such as speed, security, reliability, usability and constraints.

Q.18. Give two examples of a non-functional requirement for a Library Management System.

Two examples of non-functional requirements for a library management system include Performance and Reliability. Performance means that the system should support up to 1,000 users at the same time without performance issues. Reliability means that the system should be available 99.9% of the time to ensure high availability and minimum downtime.

Q.19. What are the key activities involved in the Design phase?

The key activities in the Design phase include creating diagrams, developing models, planning the system architecture and specifying Requirements. These activities help visualize how different parts of the software will work together. They ensure that the final software is well-organized, user-friendly, and meets the user needs.

Q.20. Give an example of a diagram used in the Design phase.

An example of a diagram used in the Design phase is a flowchart. A flowchart visually represents the sequence of steps or actions in a process using symbols. It helps developers understand how data and logic flow through the system.

Q.21. What are mockups and why are they useful?

A **mockup** is a visual draft of the user interface. It shows what the software will look like and how the users will interact with it. Mockups help the designers and stakeholders to visualize the final product before development begins.

Q.22. Why is planning the architecture important in the Design phase?

Planning the architecture involves defining the overall structure of the software system. It determines how different components will interact and function together. This helps ensure the system is well-organized and works smoothly.

Q.23. Why is it important to specify requirements during the design phase?

Specifying requirements during the Design phase means clearly defining the expected functionality and behavior of each part of the software. It ensures that every feature is properly planned and documented. It reduces the chances of missing key details during development.

Q.24. How is the design phase similar to designing a house?

The design phase is similar to designing a house. A house needs blueprints to plan rooms, layout and structure before construction. Similarly, software needs a detailed design to plan features and structure before coding begins.

Q.25. What happens in the development phase of the SDLC?

In development phase, the actual creation of the software begins. Programmers write the code based on the specifications created during the design phase. The code is written in a programming language such as Python, Java, or C++. This phase translates the software design into a working software.

Q.26. What is the purpose of the testing phase in SDLC?

The testing phase is used to check the software for any bugs, errors or issues. It ensures that the software functions correctly and meets the specified requirements. Testing improves the overall quality, performance and reliability of the software.

Q.27. What is functionality testing in software development?

Functionality testing checks whether a software works according to its defined requirements and features. It ensures that each function behaves as expected when given the correct input. For example, a web application should allow the user to login successfully using correct username and password. Functionality testing ensures that this process works properly.

Q.28. What is performance testing in software development?

Performance testing checks how well a software performs under various conditions such as high user traffic or heavy data load. For example, a website can be tested if it loads quickly even when 10,000 users access it at the same time.

www.ilmkidunya.com

Q.29. What is compatibility testing in software development?

Compatibility testing checks whether a software application works correctly across different devices, browsers and operating systems. For example, a mobile app should be tested to verify if it runs smoothly on both Android and iPhone.

Q.30. What could happen if testing is skipped in SDLC?

The software may contain bugs, errors or performance issues if testing is skipped. This can lead to poor user experience, system failures and costly fixes after the software is released.

Q.31. Why is the deployment phase of SDLC important?

The deployment phase is important because it makes the software available for users to access and use. It ensures software is properly installed, configured and functions correctly in the real-world environment.

Q.32. What are the key steps involved in deployment?

The key steps involved in deployment include installation, configuration and testing in the real-world environment. These steps ensure software is properly set up and performs as expected for the end users.

Q.33. Why is configuration important during deployment?

Configuration is important because it customizes the software to meet the specific needs of the user or organization. It includes the tasks such as setting user preferences, network settings and database connections. It ensures the software runs smoothly and functions correctly in its real-world environment.

Q.34. What is the purpose of the maintenance phase in SDLC?

The maintenance phase ensures the software continues to function correctly after deployment. It involves fixing bugs, making updates and adding new features as user needs or technology change over time.

Q.35. What are software development methodologies?

Software development methodologies are structured approaches used to plan, develop and manage software projects. These methodologies ensure that the development process is systematic, efficient and produces high-quality software.

Q.36. What are software process models?

Software process models represent the structured processes involved in the software development lifecycle. They provide a framework for planning, organizing and controlling the software development process. Two examples are the Waterfall Model and the Agile Model.

Q.37. What is the importance of software process models?

Software process models are important because they provide predictability, efficiency and quality in software development. They help manage risks, reduce wasted effort and ensure reliable software.

Q.38. How do software process models improve predictability?

Software process models improve predictability by providing a well-defined and structured approach to software development. This enables development teams to predict the outcomes and manage risks more effectively throughout the project.

Q.39. How do software process models contribute to software quality?

The process models ensure that quality assurance practices are followed throughout the development lifecycle. It leads to better and more reliable software.

Q.40. What is Waterfall model?

The Waterfall Model is one of the earliest and simplest software development process models. It follows a linear and sequential approach where each phase must be completed before moving to the next. It is easy to understand and manage because of its clear and structured flow. It is best suited for small projects with fixed and well-defined requirements.

Q.41. Why is the Waterfall Model called a sequential model?

Waterfall Model is called a sequential model because each phase of development is completed one after the other in a specific order. You cannot move to the next phase until the current one is fully completed. This step-by-step process flows from top to bottom like a waterfall.

www.ilmkidunya.com

Q.42. When is Waterfall model a good choice for software development?

The Waterfall Model is a good choice for software development when the project is small with clearly defined and fixed requirements. It is also suitable when the scope is well understood and unlikely to change during development process.

Q.43. List the main phases of the Waterfall model.

The main phases of Waterfall model are Requirements, Design, Implementation, Testing, Deployment and Maintenance.

Q.44. Write any three benefits of the Waterfall Model.

It is simple and easy to understand due to its clear and structured approach. Its sequential process makes it easier to manage and track progress as each phase is completed one at a time. It is well-suited for small projects where the requirements are fixed and not expected to change.

Q.45. Write any two limitations of the Waterfall Model.

This model is not flexible as it is difficult and costly to go back and make changes once a phase is completed. It is not ideal for complex projects with changing or evolving requirements.

Q.46. Why is the Waterfall Model not suitable for large or complex projects?

The Waterfall Model is not suitable for large or complex projects because it is difficult to adapt to changing or evolving requirements. In large projects, new requirements often arise during development. The Waterfall Model lacks flexibility to handle such changes efficiently.

Q.47. What is Agile methodology in software development?

Agile methodology is a flexible and adaptive approach to software development. It focuses on delivering small, working parts of the software quickly. This approach allows development teams to adapt to changes during the development process.

Q.48. Why is Agile useful for modern software projects?

Agile is useful because it allows teams to adapt to changes in requirements as the project progresses. It is ideal for projects where user needs may evolve over time.

Q.49. What are iterations or sprints in Agile?

Iterations or sprints are short development cycles in Agile. Each sprint focuses on building and testing a small, functional part of the software. Feedback is gathered after each sprint to make improvements in the next one. This allows continuous development and flexibility.

Q.50. How does Continuous Integration benefit developers?

Continuous integration allows multiple developers to work together by integrating their code frequently into a shared repository. The errors are detected early through automated testing. This improves code quality and team coordination.

Q.51. What is Pair Programming in Agile?

Pair programming is a practice in Agile where two developers work together on the same task using a single computer. One developer writes the code while the other reviews it in real-time. This method improves code quality and helps developers to learn from each other.

Q.52. What is Test-Driven Development (TDD)?

Test-Driven Development is a practice where tests are written before the actual code. The code is then written to pass those tests. It ensures that the software works as expected and meets all requirements.

Q.53. How does Agile model differ from Waterfall model?

The Agile model is flexible and iterative that allows changes and feedback at any stage of development. In constract, the Waterfall model is linear and sequential approach where each phase must be completed before moving to the next. Agile is better suited for projects with evolving requirements while Waterfall works best for projects with fixed and well-defined requirements.

Q.54. Write any two benefits of Agile methodology.

Agile allows changes at any stage of development based on user feedback or new requirements. Customers can see progress and give feedback during development to ensure the final product meets their expectations.

www.ilmkidunya.com

Q.55. Why Agile is considered flexible?

Agile is flexible because it allows changes at any stage of development. Teams can update features based on user feedback or evolving requirements. It ensures the final product meets user needs.

Q.56. Write any two limitations of Agile methodology.

Agile requires effective coordination and communication so it is difficult to manage large projects with multiple teams. It is harder to predict the exact timeline, cost or final outcome of the project due to ongoing changes and feedback.

Q.57. Why is stakeholder involvement critical in Agile?

Stakeholder involvement is critical in Agile because the methodology relies on regular feedback and close collaboration throughout development process. The process can slow down if some of the stakeholders are unavailable or not fully involved. Their active participation ensure the product meets real user needs and expectations.

Q.58. What is meant by scaling challenges in Agile methodology?

Scaling challenges refer to the difficulty of managing large projects with multiple teams in Agile. Agile requires effective coordination and communication among all teams to maintain smooth progress. It can lead to confusion, delays and decreased efficiency without proper management.

Q.59. Why is Agile considered less predictable than other models?

Agile is considered less predictable because projects evolve based on continuous feedback and changing requirements. It can be hard to predict the exact timeline, cost or final outcomes.

Q.60. What is meant by comprehensive project planning?

Comprehensive project planning refers to carefully thinking about every part of the project before starting the work. This includes understanding what tasks need to be done, who will do each task and how the work will be completed. Proper planning increases the chances of completing the project successfully.

Q.61. What is cost estimation in software projects?

Cost estimation is the process of predicting the total expenses required to complete a software project successfully. Accurate cost estimation helps in planning the budget, allocating resources and setting realistic goals for the project.

Q.62. What factors affect cost estimation in software projects?

Key factors affecting cost estimation include development team costs, technology stack, project duration, risk management and quality assurance. These factors can significantly impact the overall cost of a project.

Q.63. How does the development team affect project cost?

The cost of development team depends on the number of developers, their expertise and hourly rates. The experienced developers usually charge higher rates than beginners. The selection of right team helps to keep the project within budget while maintaining quality.

Q.64. How does the technology stack affect project cost?

The technology stack affects project cost based on the tools, programming languages and platforms used. Some tools are free while others need to be purchased. Some latest or complex technologies may require experts that the increases the overall cost.

Q.65. What is risk management in software projects?

Risk management is the process of identifying and evaluating potential problems that could affect the success of a software project. These risks may include delays, cost overruns or technical issues. Effective risk management helps teams prepare in advance to handle such challenges.

Q.66. Why is a contingency budget important in project planning?

A contingency budget is important because it provides extra funds to handle unexpected issues without affecting the project. It helps to deal with risks like delays or unplanned expenses.

www.ilmkidunya.com

Q.67. How does quality assurance affect the total project cost?

Quality assurance involves testing the software, fixing bugs and making improvements to meet quality standards. These activities require time, skilled personnel and resources that can increase the overall cost of the project.

Q.68. What is risk assessment in a software project?

Risk assessment is the process of identifying and analyzing potential problems that could impact the success of a software project. It helps the team to evaluate the possibility and impact of each risk.

Q.69. Name three types of risks in a software project.

Three types of risks in a software project are technical risks, operational risks and external risks. All these risks can affect the project's timeline, cost or overall success.

Q.70. What is a mitigation strategy?

A mitigation strategy is a plan that reduces the possibility of a significant risk or minimizes its impact on project. It ensures that potential problems are managed effectively to avoid disruption to the project.

Q.71. What is the difference between risk identification and risk mitigation?

Risk identification is the process of finding and listing potential problems that may affect the project. Risk mitigation involves planning and implementing strategies to reduce the possibility or impact of those problems.

Q.72. What happens if risks are not managed in a project?

The project may face unexpected issues, delays, increased costs and reduced quality if risks are not managed. It can even result in a complete project failure.

Q.73. What is the purpose of Quality Assurance (QA) in a software project?

The purpose of quality assurance is to ensure that the software meets the defined standards, works correctly and satisfies user requirements. It helps to identify and fix issues before the software is delivered.

Q.74. Name some common methods used in Quality Assurance (QA)?

Some common quality assurance methods include software testing, code reviews, stakeholder feedback and routine progress checks. These methods ensure the software is reliable and meets all expected standards.

Q.75. What is meant by the graphical representation of software systems?

Graphical representation of software systems involves using visual diagrams to show different components of a system and how they interact with each other. It makes it easier for developers and stakeholders to understand, communicate and manage the system.

Q.76. What does UML stand for and what is its main purpose?

UML stands for Unified Modeling Language. Its main purpose is to provide a standard way to visualize the design of a software system. It helps developers to understand how the system works and how its components communicate with each other.

Q.77. List four types of UML diagrams.

Four types of UML diagrams include Use Case Diagram, Class Diagram, Activity Diagram and Sequence Diagram. Each diagram type serves a specific purpose in modeling different aspects of a system.

Q.78. What is the purpose of a Use Case Diagram in UML?

A Use Case Diagram shows the system's functionality from the user's point of view. It helps to identify what the system should do and how users interact with it.

Q.79. What does an actor represent in a Use Case Diagram?

An actor represents a user or another system that interacts with the system. It can be a human such as customer or an external system such as payment gateway.

Q.80. How is an actor represented in a Use Case Diagram?

An actor is shown as a stick figure in a Use Case Diagram. This symbol visually indicates who or what is interacting with the system.

www.ilmkidunya.com

Q.81. Can a single actor have multiple use cases?

Yes, a single actor can have multiple use cases. Each use case represents a specific interaction or goal and actors can initiate several different interactions with the system.

Q.82. What is a use case in software modeling?

A **Use Case** describes how an actor interacts with a system to achieve a specific goal. Each use case represents a complete task or action to be performed by a user.

Q.83. How is a use case represented in a Use Case Diagram?

A use case is shown as an oval shape with the name of the function or task written inside. It illustrates the functionality that the system provides to the actor.

Q.84. Give an example of an actor and a related Use Case.

An example of an actor is Customer and a related use case is Place Order. This Use Case describes how the customer interacts with the system to browse products, select items, and complete a purchase.

Q.85. How do you identify Use Cases?

First, identify actors by determining different types of users who will interact with the system. Second, define goals by identifying what each actor wants to achieve using the system. Third, outline interactions by describe how each actor will interact with the system to achieve the goal. Fourth, validate these Use Cases.

Q.86. What is the role of stakeholders in validating Use Cases?

Stakeholders review and validate Use Cases to ensure they accurately represent system's required features and user interactions

Q.87. What is a class diagram?

A Class diagram is a type of UML diagram that shows the structure of a system. It represents the classes in the system and the relationships between them. It helps in modeling the system's design.

Q.88. What do attributes in a class represent?

Attributes represent the properties or characteristics of a class. For example, a Room class may have attributes like name and size.

Q.89. What are methods in a class?

Methods are the operations or functions that objects of a class can perform. For example, the Room class can have methods like open() and close().

Q.90. How is a class represented in a UML Class Diagram?

A class is represented using a rectangle divided into three sections. The top section is for the class name, the middle is for attributes and the bottom is for methods.

Q.91. What is a Sequence Diagram in UML?

Sequence diagram is a type of UML diagram that shows how objects in a system interact with each other over time in a specific sequence. They help in understanding the flow of messages between objects.

Q.92. What is a lifeline in Sequence diagram?

A lifeline represents an object or participant involved in an interaction. It shows when and how the object is active using a vertical dashed line below the object's name.

Q.93. How a lifeline is visually represented in a Sequence diagram?

A lifeline is shown as a vertical dashed line extending downward from a rectangle that represents the object or participant at the top of the diagram.

Q.94. What does a message arrow indicate in a Sequence diagram?

A message arrow represents a communication or interaction between two objects or participants. It indicates the direction of the message or function call being sent.

Q.95. What is Activity diagram in UML?

An Activity diagram is a type of UML diagram that represents the flow of activities or actions in a system or process. It provides a step-by-step visualization of how tasks are carried out and in what order they occur. They are useful to represent the logic of complex operations.

www.ilmkidunya.com

Q.96. What is a flowchart?

A flowchart is a diagram that represents the sequence of steps or actions in a process or system. It uses standard symbols to show the flow of control from one activity to another

Q.97. What symbols are used in flowcharts?

In flowcharts, the ovals represent Start/End, rectangles represent processes and diamonds represent decisions. The arrows represent the direction of the flow.

Q.98. What are design patterns in software development?

Design patterns are common solutions to recurring problems in software development. They work like templates to make the process of coding easier, faster and more consistent.

Q.99. How do design patterns help developers?

Design patterns reduce code complexity, promote code reuse and improve communication by providing a common vocabulary among developers.

Q.100. Give an example of a design pattern used in web development?

An example of a design pattern used in web development is the Model-View-Controller (MVC) pattern. It is commonly used in frameworks like Ruby on Rails and Angular.

Q.101. Define the term bug and debug

Bug is an error or mistake in the code that causes the software to behave unexpectedly or produce incorrect results. Debugging is the process of finding and fixing bugs in a software.

Q.102. What are some tools and best practices for debugging?

Some common tools and best practices for debugging include using debuggers to step through code, print statements to display variable values and code reviews to get feedback from other developers. These methods help to identify bugs more effectively and improve code quality and reliability.

Q.103. What is the role of a debugger in software development?

Debuggers are tools that help developers to find and fix bugs in the code. They allow developers to go through the code step by step, inspect variable values and monitor program execution. This process helps to identify the exact location and cause of bugs in the program.

Q.104. How are print statements used in debugging?

Print statements are added to the code to display the values of variables or messages during program execution. This technique is useful to understand what the program is doing and where it may be wrong. It is a simple but effective technique to trace and fix bugs.

Q.105. How do code reviews help in debugging?

Code reviews involve other developers carefully reading and reviewing the code. This process helps to identify possible errors and improves the overall quality of the code.

Q.106. What is the hierarchy of testing?

The hierarchy of testing typically begins with smaller components and progresses to the entire system. The main levels of testing are unit testing, integration testing, system testing, and acceptance testing.

Q.107. What type of testing is performed first in the testing hierarchy?

Unit Testing is performed first in the testing hierarchy. It tests individual components or modules to ensure they work correctly according to their design and perform as expected.

Q.108. What is the main goal of unit testing?

The main goal of unit testing is to verify that each component works correctly according to its design and performs as expected. This helps catch bugs early in the development process.

Q.109. How does integration testing differ from unit testing?

Unit testing checks individual components or modules to ensure they work correctly as designed. Integration testing checks how these modules interact when combined. It ensures data is passed correctly between modules and their integration does not cause errors.

Q.110. What is system testing?

System testing involves testing the entire software system as a whole. The goal is to evaluate the software's overall functionality, performance and security to ensure it meets the specified requirements.

www.ilmkidunya.com

Q.111. Who typically performs Acceptance Testing?

Acceptance testing is often performed by the end-users. The users evaluate whether the software meets their needs and requirements.

Q.112. Why are software development tools important?

Software development tools are important because they help developers to write, edit, test, debug and manage code efficiently. They make the development process faster, more organized and reduces the chances of errors. These tools ensure that the final software works correctly and meets user requirements.

Q.113. What are language editors in software development?

The language editors are tools used by developers to write and edit code in various programming languages. Most language editors provide various features like syntax highlighting, auto-completion, line numbering and error detection to make coding easier and faster. Some examples include Notepad++ and Visual Studio Code (VS Code).

Q.114. Why is a translator needed in software development?

A translator is needed to convert code written in high-level programming languages into machine language. A program written in high level language cannot be executed directly on the computer. It must be converted into machine language to execute.

Q.115. Describe two main types of translators.

Two types of translators are interpreters and compilers. Interpreters translate and execute code line-by-line while compilers translate the entire code into machine code before execution.

Q.116. Give two examples of debuggers.

Two examples of debuggers are GDB (GNU Debugger) and Visual Studio Debugger.

Q.117. What is an Integrated Development Environment (IDE)?

An IDE is a software suite that combines all the essential tools for software development in one place. Different tools available in an IDE include code editors, compilers and debuggers to help developers write, test and debug code more efficiently.

Q.118. Write two examples of Integrated Development Environments (IDEs) and what are they used for?

Two examples of Integrated Development Environments are Visual Studio and PyCharm. Visual Studio is widely used for developing applications in .NET, C++, and C#. PyCharm is a popular IDE designed for developing applications in Python.

Q.119. What are computing platforms in software development?

Computing platforms are environments where developers can write, run and test code. They provide the necessary tools and services for developing software applications. Platforms can be online (cloud-based) or offline (locally installed).

Q.120. What are online computing platforms?

The online platforms are cloud-based development environments that are accessible through the Internet using a web browser. The developers can write, run and test code directly in a web browser without installing any software on computer. Some examples of online platforms are Repl.it and Gitpod.

Q.121. What are offline computing platforms?

Offline platforms are local development environments installed directly on a computer. The developers can write, run and test code using the tools available in IDEs without connecting to the internet. Some examples of offline platforms are Visual Studio Code, PyCharm and Eclipse.

Q.122. What is a source code repository?

A source code repository is a centralized platform where developers can save, manage and organize the code for a software project. It allows multiple developers to work on different parts of the code simultaneously without causing conflicts.

Q.123. What are GitHub and Bitbucket?

GitHub is a source code repository platform especially popular for open-source projects. Bitbucket is another source code repository used mainly by development teams and organizations. It supports both private and public repositories.

www.ilmkidunya.com

Multiple Choice Questions

1. **The process of creating computer programs to perform specific tasks is called:**
   a. Software installation b. Software development
   c. System configuration d. Software debugging
2. **SDLC stands for:**
   a. Software Design and Logic Control b. Software Development Learning Course
   c. System Design Language Code d. Software Development Life Cycle
3. **The primary purpose of the Software Development Life Cycle (SDLC) is to:**
   a. Deliver quality software b. Design hardware
   c. Install operating systems d. Manage networks
4. **A reusable set of concepts, practices and tools that provides a structured foundation for software development is called:**
   a. IDE b. Framework
   c. Utility library d. Debugging tool
5. **Which of the following is an example of a web development framework?**
   a. UML b. C++ c. Django d. Python
6. **Django is a framework that helps developers create:**
   a. Mobile games b. Desktop software
   c. Websites d. Hardware drivers
7. **Which of the following is NOT a feature of a framework?**
   a. Efficiency b. Code reusability
   c. Unstructured development d. Consistency
8. **Which of the following is the first phase of the Software Development Life Cycle (SDLC)?**
   a. Testing b. Coding
   c. Requirement Gathering d. Deployment
9. **Which phase of the SDLC focuses on understanding and gathering the software requirements from stakeholders?**
   a. Testing b. Coding
   c. Requirement Gathering d. Deployment
10. **Which of the following is NOT an activity in Requirement Gathering phase of the SDLC?**
    a. Interviews b. Observations
    c. Writing code d. Reviewing documents
11. **Functional and non-functional requirements are collected during which phase of the SDLC?**
    a. Deployment b. Requirement Gathering
    c. Testing d. Maintenance
12. **Which of these defines a system's specific behaviors and operations?**
    a. Non-functional requirements b. Performance requirements
    c. Functional requirements d. Design constraints
13. **"Librarians can add or remove books from inventory" is an example of:**
    a. Non-functional requirements b. Performance constraint
    c. Functional requirements d. Security constraint
14. **Which type of requirements define the quality attributes, performance criteria and constraints of a system?**
    a. Non-functional requirements b. Business requirements
    c. Functional requirements d. Hardware requirements
15. **A requirement that requires the system to support 1000 users at once is a:**
    a. Non-functional requirement b. Business requirement
    c. Functional requirement d. Design requirement
16. **Which of the following is an example of a non-functional requirement?**
    a. Book search feature b. User registration
    c. System availability of 99.9% d. Book borrowing

www.ilmkidunya.com

17. **Which non-functional requirement ensures minimal downtime?**
    a. Performance b. Availability
    c. Usability d. Scalability
18. **"User data encryption" aligns with which non-functional requirement?**
    a. Performance b. Compatibility
    c. Maintainability d. Security
19. **Which of the following describes how quickly the system responds to user actions?**
    a. Reliability b. Performance
    c. Portability d. Functionality
20. **Which phase of SDLC focuses on how the software will look and function?**
    a. Requirements b. Design
    c. Deployment d. Testing
21. **Which of the following activities is not part of the Design phase in the SDLC?**
    a. Creating diagrams b. Creating mockups
    c. System testing d. Planning architecture
22. **Which of the following is used to visually represent the steps of a program?**
    a. Flowchart b. Algorithm
    c. Gantt chart d. ER Diagram
23. **Which of the following is the main activity in the Development phase of the SDLC?**
    a. Planning b. Coding
    c. Testing d. Documenting
24. **In the _____ phase, programmers begin writing code to create the software.**
    a. Development b. Requirement Gathering
    c. Testing d. Design
25. **Who is primarily responsible for writing the code during software development?**
    a. Software testers b. End users
    c. Programmers d. Business Analyst
26. **Which phase comes immediately after coding in the SDLC?**
    a. Requirement Gathering b. Designing
    c. Coding d. Testing
27. **_____ is the process of checking the software to identify any bugs, errors or issues.**
    a. Designing b. Coding c. Testing d. Installing
28. **Which SDLC phase verifies that software works correctly and meets quality standards?**
    a. Designing b. Coding
    c. Testing d. Requirement Gathering
29. **Which type of testing evaluates system behavior under high user traffic or large data volumes?**
    a. Unit testing b. Performance testing
    c. Visual testing d. Manual testing
30. **The term for an error in software that leads to incorrect or unexpected results is:**
    a. Patch b. Interface c. Bug d. Module
31. **Which type of testing validates that software behaves according to the requirements and design specification?**
    a. Compatibility testing b. Performance testing
    c. Functionality testing d. Unit testing
32. **Which type of testing checks if software works properly on different browsers or mobile devices?**
    a. Compatibility testing b. Load testing
    c. Security testing d. Usability testing
33. **Which of the following is NOT a type of software testing?**
    a. Performance testing b. Functional testing
    c. Compatibility testing d. Presentation testing
34. **_____ is the phase where the software is installed, configured, and made available to end-users**
    a. Deployment b. Testing c. Design d. Coding

www.ilmkidunya.com

35. **What is typically the first step in the software deployment process?**
a. Configuration
b. User Training
c. Installation
d. Real-World Testing

36. **Which of the following is NOT part of the Deployment phase in SDLC?**
a. Configuration
b. Requirement analysis
c. Installation
d. Real-World Testing

37. **Which deployment step involves setting up user preferences and connecting to databases or networks?**
a. Compilation
b. Installation
c. Configuration
d. Documentation

38. **Which SDLC phase involves releasing the software to users for real-world testing and use?**
a. Requirement Gathering
b. Design
c. Deployment
d. Development

39. **Which type of users typically interact with the software after deployment?**
a. Designers
b. Testers
c. End users
d. Programmers

40. **Which of the following is the final phase in the SDLC?**
a. Design
b. Deployment
c. Maintenance
d. Testing

41. **Which SDLC phase involves ongoing updates and support after deployment?**
a. Design
b. Development
c. Maintenance
d. Testing

42. **Which of the following is NOT a benefit of software process models?**
a. Predictability
b. Efficiency
c. High memory usage
d. Quality

43. **What key aspect does a software process model help maintain during software development?**
a. Passwords
b. Graphic design
c. Quality assurance
d. Hardware compatibility

44. **_____ are abstract representations of the processes involved in the software development lifecycle.**
a. Development environments
b. Software tools
c. Software process models
d. Software design principles

45. **Which of these elements is NOT typically included in a software process model?**
a. Planning
b. User testing
c. Structuring tasks
d. Marketing campaigns

46. **A team following a process model can better manage:**
a. Database indexing
b. Debugging tools
c. Risks and outcomes
d. Presentation styles

47. **Which software process model is linear and sequential?**
a. Agile Model
b. Spiral Model
c. Waterfall Model
d. Prototyping Model

48. **Which software process model is well-suited for projects with well-defined requirements that are unlikely to change?**
a. Agile
b. Waterfall
c. Prototyping
d. Spiral

49. **Which Software process model is a good choice for projects with constantly evolving requirements?**
a. Waterfall Model
b. Agile Model
c. Prototyping Model
d. V-Model

50. **The short development cycles in Agile are called:**
a. Loops
b. Phases
c. Sprints
d. Modules

51. **Which software development practice involves writing tests before writing the actual code?**
a. Feature-Driven Development
b. Manual Testing
c. Test-Driven Development
d. Regression Testing

www.ilmkidunya.com

52. **Which Agile practice has two developers collaborating at one workstation, with one coding and the other reviewing simultaneously?**
a. Test-Driven Development b. Continuous Integration
c. Pair Programming d. Code Deployment

53. **Which of the following is NOT a practice in Agile software development methodology?**
a. Pair Programming b. Continuous Integration
c. Test-Driven Development d. Big Bang Integration

54. **Which of the following is the first phase in a project management plan?**
a. Execution b. Initiation c. Planning d. Closure

55. **Which technology company reached a $2 trillion market valuation in 2023?**
a. Facebook b. Amazon c. Microsoft d. Google

56. **Which financial reserve is designated for unexpected project expenses?**
a. Marketing budget b. Resource allocation
c. Contingency fund d. Gantt chart

57. **Which of the following is not a key factor in cost estimation?**
a. Quality assurance b. Risk management
c. Customer feedback d. Project duration

58. **Which of the following is the primary consideration when selecting programming languages and tools for a software project?**
a. Project timeline b. Risk assessment
c. Technology Stack d. User Feedback

59. **Software testing and bug fixing are primarily categorized under which project management activity?**
a. Project Planning b. Technology Stack
c. Quality Assurance d. Risk Management

60. **UML stands for:**
a. Unified Markup Language b. Universal Mapping Logic
c. Unified Modeling Language d. Unique Model Layout

61. **What is the primary purpose of Unified Modeling Language (UML)?**
a. Coding algorithms b. Hardware design
c. Visualizing software design d. Running applications

62. **Which of the following is NOT a benefit of UML?**
a. Standardized visualization b. Improved communication
c. Automatic code generation d. Better system understanding

63. **Which UML diagram is primarily used to capture functional requirements from user perspective?**
a. Class diagram b. Sequence diagram
c. Use case diagram d. Deployment diagram

64. **Which UML diagram represents the interactions between users and system functionalities?**
a. Class diagram b. Sequence diagram
c. Use case diagram d. Activity Diagram

65. **The first step in identifying use cases is:**
a. Define user goals b. Outline system interactions
c. Identify actors d. Validate use cases

66. **Which term describes any entity that interacts with the system in UML Use Case diagrams?**
a. Class b. Object
c. Actor d. Component

67. **Actors in a use case can be:**
a. Only human users b. Only other systems
c. Only project managers d. Both a and b

68. **In UML Use Case Diagrams, actors are represented as:**
a. Circle b. Stick figure
c. Rectangle d. Arrow

www.ilmkidunya.com

69. **In a library system, which of the following could be an actor?**
a. Book b. Search book c. Librarian d. Borrow

70. **Which of the following is NOT part of a class diagram?**
a. Class name b. Attributes
c. Methods d. Actors

71. **Which UML diagram best represents the structural organization of a system?**
a. Use case diagram b. Sequence diagram
c. Class diagram d. Deployment diagram

72. **A class diagram primarily focuses on:**
a. User interactions b. System structure
c. Network latency d. Code optimization

73. **In UML class diagrams, what do the actions 'open' and 'close' usually represent?**
a. Users b. Attributes
c. Methods d. Relationships

74. **Which UML notation element is used to represent a class in class diagrams?**
a. Oval b. Stick figure c. Rectangle d. Diamond

75. **Which UML diagram is used to show how objects in a system interact with each other in a particular sequence?**
a. Class Diagram b. Use Case Diagram
c. Sequence Diagram d. Activity Diagram

76. **What do objects in a sequence diagram primarily exchange?**
a. Class definitions b. Messages
c. Memory addresses d. Return values

77. **Which UML diagram is specifically designed to represent workflows and sequences of actions in a process?**
a. Class diagram b. Activity diagram
c. Deployment diagram d. Component diagram

78. **Which UML diagram type is most similar to a flowchart?**
a. Class Diagram b. Use Case Diagram
c. Sequence Diagram d. Activity Diagram

79. **In an activity diagram, what does an arrow indicate?**
a. User input b. Message flow c. Control flow d. Use case

80. **A diamond shape in an activity diagram represents:**
a. Action b. Decision
c. Class d. Start node

81. **Which of the following is NOT a type of UML diagram?**
a. Class Diagram b. Use Case Diagram
c. Sequence Diagram d. Flowchart Diagram

82. **Which concept serves as reusable templates for solving common software design problems?**
a. Design Patterns b. Programming Languages
c. Flowcharts d. Debuggers

83. **Which design pattern ensures that only one instance of a class exists in the entire application?**
a. Factory Pattern b. Strategy Pattern
c. Observer Pattern d. Singleton Pattern

84. **Which design pattern notifies all dependent or interested objects automatically when the observed object changes state?**
a. Factory Pattern b. Strategy Pattern
c. Observer Pattern d. Singleton Pattern

85. **The Observer Pattern is primarily used for:**
a. Data compression b. User logins
c. Notification systems d. File storage

www.ilmkidunya.com

86. **Which design pattern functions like a workshop, manufacturing different types of objects while hiding their creation logic?**
a. Factory Pattern b. Strategy Pattern
c. Observer Pattern d. Singleton Pattern

87. **Which design pattern is analogous to selecting the right tool from a toolbox for a specific task?**
a. Factory Pattern b. Strategy Pattern
c. Observer Pattern d. Singleton Pattern

88. **Which of the following is NOT a design pattern?**
a. Strategy Pattern b. Observer Pattern
c. Loop Pattern d. Factory Pattern

89. **Which of the following is a design pattern commonly used in web development frameworks?**
a. FIFO b. MVC c. HTML d. CPU

90. **Which of the following frameworks use the MVC pattern?**
a. Adobe Photoshop b. Ruby on Rails and Angular
c. MS Word and Excel d. Oracle Database

91. **A mistake or error that causes unexpected behavior in software is called:**
a. Function b. Bug c. Syntax d. Module

92. **The process of finding and fixing errors in software is called:**
a. Compiling b. Designing
c. Debugging d. Encrypting

93. **Which tool is used to inspect variables and step through code during execution?**
a. Compiler b. Browser c. Debugger d. Server

94. **Which debugging technique allows developers to execute code one line at a time to observe behavior and identify errors?**
a. Stepping through code b. Code formatting
c. Breakpoint d. Logging

95. **Which of the following is the first and lower level in the software testing hierarchy?**
a. System Testing b. Acceptance Testing
c. Unit Testing d. Integration Testing

96. **What type of software testing is performed in isolation from other components?**
a. System Testing b. Acceptance Testing
c. Unit Testing d. Integration Testing

97. **Which type of testing focuses on verifying the communication and interaction between software modules?**
a. Regression Testing b. Acceptance Testing
c. Unit Testing d. Integration Testing

98. **Which type of testing is used to evaluate a software system's overall functionality?**
a. System Testing b. Acceptance Testing
c. Regression Testing d. Integration Testing

99. **Which of the following is used to test the entire software system as a whole?**
a. System Testing b. Acceptance Testing
c. Unit Testing d. Integration Testing

100. **Which type of testing checks whether the software meets user expectations and requirements?**
a. System Testing b. Acceptance Testing
c. Unit Testing d. Integration Testing

101. **Who typically performs acceptance testing?**
a. Developers b. End-users or clients
c. Database administrators d. Security analysts

102. **UAT stands for:**
a. Unit Acceptance Testing b. User Acceptance Testing
c. Unified Application Testing d. Universal Access Testing

www.ilmkidunya.com

103. Which of the following is NOT a software testing level?
a. System Testing
b. Acceptance Testing
c. Unit Testing
d. Code Formatting Testing

104. Tools designed primarily for writing and editing source code are called:
a. Compilers
b. Debuggers
c. Code Editors
d. Browsers

105. What is the purpose of a code editor?
a. Run hardware tests
b. Translate text
c. Write and edit code
d. Compress files

106. Another name for a language editor is:
a. Translator
b. Code editor
c. File explorer
d. Image editor

107. Which of the following is a code editor?
a. Notepad++
b. Chrome
c. Excel
d. C++

108. Which code editor is highly extensible and supports multiple programming languages?
a. Notepad
b. Excel
c. VS Code
d. Paint

109. _____ is used to convert high-level program into machine language.
a. Debugger
b. Editor
c. Translator
d. Repository

110. Which of the following is an example of a high-level language?
a. Machine code
b. UML
c. Python
d. Algorithm

111. Which type of language consists of binary code that computers can directly execute?
a. Assembly Language
b. High-Level Language
c. Machine Language
d. Markup Language

112. Which tool translates code line-by-line during program execution?
a. Compiler
b. Interpreter
c. Editor
d. Debugger

113. Which tool translates the entire code at once before program execution?
a. Compiler
b. Interpreter
c. Editor
d. Debugger

114. Which type of translator does Python primarily use to execute code?
a. Browser
b. Compiler
c. Interpreter
d. File viewer

115. C/C++ programs are converted into machine language with the help of:
a. Editor
b. Compiler
c. Debugger
d. Loader

116. GCC is an example of a:
a. Code editor
b. Compiler
c. Debugger
d. Loader

117. GCC stands for:
a. Global Code Compiler
b. GNU Code Converter
c. GNU Compiler Collection
d. General Computer Compiler

118. Which debugger is integrated with Visual Studio IDE?
a. Eclipse
b. VS Code Debugger
c. Visual Studio Debugger
d. NetBeans

119. Which of the following is a debugger for C/C++ programs?
a. Python Interpreter
b. VS Code
c. GDB
d. GCC

120. IDE stands for:
a. Integrated Debugging Environment
b. Internal Data Editor
c. Integrated Development Environment
d. Independent Developer Engine

121. Which of the following provides an integrated set of tools for writing, testing and debugging programs?
a. Repository
b. Debugger
c. IDE
d. Text editor

www.ilmkidunya.com

122. Which of the following is NOT included in an IDE?

a. Editor
b. Compiler
c. Debugger
d. Image editor

123. Which of the following IDEs is most widely used for Python development?

a. PyCharm
b. Visual Studio
c. Spyder
d. IDLE

124. Which of the following is an example of an offline development platform?

a. GitHub
b. Visual Studio
c. Gitpod
d. Repl.it

125. Which platform enables developers to store, manage, and track code changes using version control?

a. Code editor
b. Source code repository
c. Debugger
d. Compiler

126. Which platform is the most widely used for hosting open-source projects?

a. GitLab
b. Bitbucket
c. GitHub
d. Repl.it

127. Which platform is used for both public and private repositories?

a. VS Code
b. Bitbucket
c. Gitpod
d. Repl.it

Answers

1. b	2. d	3. a	4. b	5. c	6. c
7. d	8. c	9. c	10. c	11. b	12. c
13. c	14. a	15. a	16. c	17. b	18. d
19. b	20. b	21. c	22. a	23. b	24. a
25. c	26. d	27. c	28. c	29. b	30. c
31. c	32. a	33. d	34. a	35. c	36. b
37. c	38. c	39. c	40. c	41. c	42. c
43. c	44. c	45. d	46. c	47. c	48. b
49. b	50. c	51. c	52. c	53. d	54. b
55. c	56. c	57. c	58. c	59. c	60. c
61. c	62. c	63. c	64. c	65. c	66. c
67. d	68. b	69. c	70. d	71. c	72. b
73. c	74. c	75. c	76. b	77. b	78. d
79. c	80. b	81. d	82. a	83. d	84. c
85. c	86. d	87. b	88. c	89. b	90. b
91. b	92. c	93. c	94. a	95. c	96. c
97. d	98. a	99. a	100.b	101.b	102.b
103.d	104.c	105.c	106.b	107.a	108.c
109.c	110.c	111.c	112.b	113.a	114.c
115.b	116.b	117.c	118.c	119.c	120.c
121.c	122.d	123.a	124.b	125.b	126.c
127.b					

www.ilmkidunya.com