GROUP 6 QUESTIONS (IS ELECTIVE)

1. Are executable instructions or code that automate the execution of test cases.
   a. Test scripts
   b. Test results
   c. Documenting test
   d. Test case

2. It is a document the outcomes of individual test executions.
   a. Test scripts
   b. Test results
   c. Documenting test
   d. Test case

3. This involves documenting the various tests performed on software and applications to ensure they meet specified requirements and quality standards.
   a. Test scripts
   b. Test results
   c. Documenting test
   d. Test case

4. It is a detailed description of individual tests that need to be performed on the system.
   a. Test scripts
   b. Test results
   c. Documenting test
   d. Test case

5. Outline the overall strategy for testing the system.
   a. Test scripts
   b. Test Plan
   c. Documenting test
   d. Test case

6. It provides a structured approach to define the objectives and goals of automated testing within the context of the organization's overall quality assurance strategy.
   a. Clear objectives and goals
   b. Efficiency and time savings
   c. Resource allocation and planning
   d. Risk assessment and mitigation

7. The document helps in identifying the resources required for implementing and maintaining automated testing practices.
   a. Clear objectives and goals
   b. Efficiency and time savings
   c. Resource allocation and planning
   d. Risk assessment and mitigation

8. It enables teams to identify potential risks associated with automated testing, such as tool limitations, environmental dependencies, or scalability issues.
   a. Clear objectives and goals
   b. Efficiency and time savings
   c. Resource allocation and planning
   d. Risk assessment and mitigation

9. The document includes metrics and key performance indicators (KPIs) for evaluating the effectiveness and efficiency of automated testing.
   a. Clear objectives and goals
   b. Efficiency and time savings
   c. Measurement and reporting
   d. Risk assessment and mitigation

10. A well-defined testing strategy document helps in optimizing the automated testing process to improve efficiency and save time.
    a. Clear objectives and goals
    b. Efficiency and time savings
    c. Resource allocation and planning
    d. Risk assessment and mitigation

11. Start by defining your infrastructure requirements using tools like Terraform, AWS CloudFormation, or Azure Resource Manager.
    a. Implement Configuration Management
    b. Infrastructure as Code (IaC)
    c. Containerize Applications
    d. Leverage Cloud-Based Testing Environments

12. Employ configuration management tools such as Ansible, Puppet, or Chef to automate the setup and maintenance of your testing environments.
    a. Implement Configuration Management
    b. Infrastructure as Code (IaC)
    c. Containerize Applications
    d. Leverage Cloud-Based Testing Environments

13. Take advantage of cloud services like AWS, Azure, or Google Cloud Platform to provision on-demand testing environments.
    a. Implement Configuration Management
    b. Infrastructure as Code (IaC)
    c. Containerize Applications
    d. Leverage Cloud-Based Testing Environments

14. Use containerization technologies such as Docker to encapsulate application dependencies and configuration
    a. Implement Configuration Management
    b. Implement Test Data Management
    c. Containerize Applications
    d. Leverage Cloud-Based Testing Environments

15. Develop strategies for managing test data effectively, including data generation, masking, and refreshing.
    a. Implement Configuration Management
    b. Implement Test Data Management
    c. Containerize Applications
    d. Leverage Cloud-Based Testing Environments

16. Collect all requirements from various sources such as project documentation, stakeholder interviews, and user stories.
    a. Define RTM Template
    b. Identify Requirements
    c. Document Requirements
    d. Link Requirements

17. Determine the structure of your RTM, including columns for requirements, their sources, statuses, and links to test cases or other relevant documents.
    a. Define RTM Template
    b. Identify Requirements
    c. Document Requirements
    d. Link Requirements

18. Input the identified requirements into the RTM template. Each requirement should have a unique identifier for easy reference.
    a. Define RTM Template
    b. Identify Requirements
    c. Document Requirements
    d. Link Requirements

19. Establish links between requirements and related artifacts such as design documents, test cases, and source code.
a. Define RTM Template
b. Identify Requirements
c. Document Requirements
d. Link Requirements

20. Utilize automation tools or scripts to streamline the process of linking requirements to other artifacts.
a. Define RTM Template
b. Identify Requirements
c. Document Requirements
d. Automate Linkage

21. Develop a system for managing changes to requirements and their associated artifacts.
a. Implement Change Management
b. Integrate with Project Management Tools
c. Document Requirements
d. Automate Linkage

22. Integrate the RTM with project management tools such as Jira or Trello to facilitate collaboration and tracking across teams.
a. Implement Change Management
b. Integrate with Project Management Tools
c. Document Requirements
d. Automate Linkage

23. Regularly review and update the RTM to reflect changes in requirements, project scope, and stakeholder feedback.
a. Implement Change Management
b. Integrate with Project Management Tools
c. Document Requirements
d. Regular Maintenance

24. Continuously assess and improve the automation process to enhance efficiency and accuracy in managing requirements traceability.
a. Implement Change Management
b. Continuous Improvement
c. Document Requirements
d. Regular Maintenance

25. Make sure the RTM is accessible to all relevant stakeholders, with appropriate permissions and visibility settings in place.
a. Implement Change Management
b. Continuous Improvement
c. Ensure Accessibility
d. Regular Maintenance

26. Automatically populating defects enhances traceability by providing a clear record of which requirements are affected by identified issues.
a. Efficiency
b. Accuracy
c. Timeliness
d. Traceability

27. Automation reduces manual effort by automatically capturing defects linked to specific requirements.
a. Efficiency
b. Accuracy
c. Timeliness
d. Traceability

28. Automation reduces the risk of human error in populating defects, ensuring that all identified issues are accurately recorded and linked to the appropriate requirements.
a. Efficiency
b. Accuracy
c. Timeliness
d. Traceability

29. Defects are promptly reflected in the RTM as soon as they are identified, allowing stakeholders to quickly assess the status of requirements and take appropriate action.
a. Efficiency
b. Accuracy
c. Timeliness
d. Traceability

30. Automated processes may lack the contextual understanding necessary to accurately determine the severity and impact of defects on requirements.
a. Lack of Contextual Understanding
b. Maintenance Overhead
c. Privacy and Security Concerns
d. False Positives

IS ELECTIVE QUIZ (GROUP 7)

1. The individual steps or actions that make up a test case to ensure they are clear, accurate, and cover all necessary scenarios.
   - Code
   - Logic
   - **Test Steps**
   - Coverage

2. Ensure that all parts of the code have been tested and that there are no untested areas that could potentially contain defects.
   - Code
   - **Coverage**
   - Logic
   - Test Steps

3. Used in the code to ensure they are correct, efficient, and meet the intended requirements.
   - Coverage
   - **Logic**
   - Code
   - Test Steps

4. Conducting a code review to check for coding standards compliance, potential bugs, readability, and maintainability.
   - Test Steps
   - Coverage
   - Logic
   - **Code**

5. A test or formats used to ensure they are comprehensive, consistent, and align with testing standards.
   - **Template**
   - Monitoring
   - Test scripts
   - Data

6. Evaluates system performance under normal and peak load conditions.
   - Stress Testing
   - Spike Testing
   - **Load Testing**
   - Endurance Testing

7. Evaluates system performance over an extended period to identify any issues related to resource leaks or degradation.
    - **Endurance Testing**
    - Stress Testing
    - Load Testing
    - Spike Testing

8. Assesses how the system handles sudden spikes or surges in user traffic.
    - Endurance Testing
    - Load Testing
    - Stress Testing
    - **Spike Testing**

9. Measures the system's ability to scale up or down in terms of users, transactions, or data volume.
    - Reliability Testing
    - **Scalability Testing**
    - Load Testing
    - Volume Testing

10. Tests the system's performance with a large volume of data to ensure it can handle the expected data load.
    - Scalability Testing
    - Load Testing
    - Reliability Testing
    - **Volume Testing**

11. It is a activity that ensures that an end product stakeholder's true needs and expectations are met.
    - Scalability Testing
    - Load Testing
    - Reliability Testing
    - **Validation Testing**
12. Ensure that all software requirements are clear, complete, and well-defined.
    - Code Coverage Analysis
    - Test Coverage Analysis
    - Requirement Traceability
    - **Requirements Verification**

13. Evaluate the extent to which the automated tests cover the functionalities, features, and scenarios specified in the requirements.
    - **Test Coverage Analysis**
    - Code Coverage Analysis
    - Requirement Coverage
    - Volume Testing

14. Ensure that the automated tests align with the specified requirements and provide adequate coverage for each requirement.
    - Test Coverage Analysis
    - Code Coverage Analysis
    - **Requirement Coverage**
    - Requirement Traceability

15. Continuously monitor the testing process and outcomes to identify areas for improvement and ensure the ongoing alignment with requirements and coverage goals.
    - Test Coverage Analysis
    - **Monitoring**
    - Test Scripts Review
    - Feedback Loop

16. Unit testing helps in detecting bugs and errors in individual units of code, allowing developers to fix them before they propagate to other parts of the system.
    - **Bug Detection**
    - Regression Testing
    - Code Quality
    - Documentation

17. Unit tests serve as living documentation for the codebase, providing insights into how individual components should behave and how they interact with each other.
    - Bug Detection
    - Regression Testing
    - Code Quality
    - **Documentation**

18. Writing unit tests encourages developers to write modular, well-structured, and reusable code.
- Bug Detection
- Regression Testing
- **Code Quality**
- Documentation

19. Unit tests serve as a safety net during code changes and refactoring by ensuring that existing functionality remains intact.
- Bug Detection
- **Regression Testing**
- Code Quality
- Documentation

20. Type of testing conducted to evaluate how a system behaves under different workloads and to determine its responsiveness, stability, and scalability.
- Bug Detection
- **Performance testing**
- Code Quality
- Documentation

21. Performance testing ensures that the system meets user expectations in terms of speed and responsiveness, thus enhancing user experience.
- **User Experience**
- Reliability
- Scalability

22. Performance testing assesses how the system handles increasing workloads and user traffic, allowing for scalability planning and optimization.
- **Scalability**
- User Experience
- Reliability

23. It helps in identifying and fixing issues related to system crashes, timeouts, or unresponsiveness, ensuring the reliability of the application.
- **Reliability**
- Scalability
- User Experience

24. A well-performing application enhances the reputation of the business and increases customer satisfaction, leading to higher retention rates and improved brand image.
- Cost-Effectiveness
- **Business Reputation**
- Compliance

25. In certain industries, compliance regulations may require performance testing to ensure that the system meets specified performance criteria.
- Cost-Effectiveness
- Business Reputation
- **Compliance**

26. By identifying performance issues early in the development lifecycle, performance testing helps in reducing the cost of fixing issues post-production.
- **Cost-Effectiveness**
- Business Reputation
- Compliance

27. Is an open-source tool developed by the Apache Software Foundation.
- **JMeter**
- LoadNinja
- WebLOAD

28. Is a popular tool that offers robust load testing capabilities, including concurrency testing.
- JMeter
- LoadNinja
- **WebLOAD**

29. Is a cloud-based tool focusing on ease of use and scalability.
- JMeter
- **LoadNinja**
- WebLOAD

30. Is a type of non-functional software testing that evaluates the behavior and performance of an application under sustained or prolonged usage.
- **Soak Testing**
- LoadNinja
- WebLOAD

GROUP 8

1. Refers to the process of monitoring and analyzing various aspects of the testing process and the software under test.
A. Program Tracking
B. Identifying Defects Early
C. Test Prioritization
D. Continuous Feedback Loop

2. Program tracking allows AST tools to monitor code changes and trigger relevant test cases automatically.
A. Continuous Feedback Loop
B. Test Prioritization
C. Identifying Defects Early
D. Program Tracking

3. With program tracking, AST tools can prioritize test execution based on the impact of code changes.
A. Identifying Defects Early
B. Test Prioritization
C. Program Tracking
D. Adaptability to Change

4. Program tracking facilitates real-time feedback between development and testing activities.
A. Root Cause Analysis
B. Adaptability to Change
C. Test Prioritization
D. Continuous Feedback Loop

5. In case of test failures, program tracking provides insights into the root causes of issues.
A. Adaptability to Change
B. Root Cause Analysis
C. Program Tracking
D. Identifying Defects Early

6. What is the primary goal of meticulously analyzing and validating requirements in defect prevention?
A. To identify defects during testing
B. To ensure software requirements are met
C. To conduct code reviews
D. To prioritize software features

7. Which activity is essential for maintaining a comprehensive understanding of defects after analysis and review?
A. Root cause analysis
B. Pair programming
C. Defect logging and documentation
D. Test-driven development

8. What is the main benefit of implementing pair programming according to the presentation?
A. Accelerating the development process
B. Reducing the likelihood of defects in later stages
C. Minimizing the need for code reviews
D. Decreasing the need for automated testing

9. How does Test-Driven Development (TDD) contribute to defect prevention as mentioned in the slides?
A. By automating the testing process
B. By writing tests before coding
C. By ensuring compliance with requirements
D. By identifying errors early in development

10. Which technique helps ensure that established standards are followed during development based on the presentation?
A. Training and skill development
B. Pair programming
C. Test-Driven Development (TDD)
D. Utilizing checklists

11. Which characteristic of good automated testing metrics refers to the degree to which something is connected or pertinent to a particular topic, situation, or context.
A. Actionability
B. Relevance
C. Measurability
D. Interpretability

12. Which characteristic of good automated testing metrics involves determining whether an attribute, phenomenon, or outcome can be assessed or evaluated using specific criteria, metrics, or instruments to track its progress, performance, or impact.
A. Actionability
B. Relevance
C. Measurability
D. Interpretability

13. Which characteristic of good automated testing metrics refers to the quality of being actionable, meaning that something can be acted upon or used to take practical steps or make decisions.
A. Actionability
B. Relevance
C. Measurability
D. Interpretability

14. Which characteristic of good automated testing metrics refers to the ease with which something can be understood, explained, or interpreted.
A. Actionability
B. Relevance
C. Measurability
D. Interpretability

15. In the characteristics of good automated testing metrics, which aspect is not directly connected to relevance?
A. Alignment with business objectives
B. Minimizing irrelevant results
C. Aligning with project requirements
D. Maximizing relevant outcomes

16. Which metric quantifies the number of defects relative to the size of the module or release, providing insight into the quality of the codebase?
A. Defect Leakage
B. Defect Density
C. Defect Removal Efficiency
D. Defect Category

17. What does defect leakage measure in the testing process before user acceptance testing (UAT)?
A. The effectiveness of defect removal methods
B. The distribution of defects based on various quality criteria
C. The number of defects missed by the testing team
D. The impact of defects on the software's quality and efficiency

18. What does Defect Removal Efficiency (DRE) evaluate in the testing phases?
A. The effectiveness of defect removal methods
B. The efficiency of test cases in detecting defects
C. The impact of defects on the software's quality
D. The completeness of testing activities

19. What do defect category metrics provide a breakdown of?
A. Defect severity levels
B. Testing process effectiveness
C. Defects missed by the testing team
D. Defects based on various quality criteria

20. What does test coverage assess in testing activities?
A. The efficiency of test case preparation efforts
B. The completeness of testing activities
C. The impact of defects on the software's quality
D. The distribution of defects based on various quality criteria

21. AST metrics help measure the extent to which automated tests cover different parts of the software under test.
A. Test Execution Time
B. Test Coverage
C. Resource Utilization
D. Regression Testing

22. AST metrics allow teams to track the effectiveness of automated tests in detecting defects.
A. Defect Detection
B. Test Coverage
C. Test Execution Time
D. Regression Testing

23. AST metrics provide insights into the efficiency of automated test execution.
A. Defect Detection
B. Test Coverage
C. Resource Utilization
D. Test Execution Time

24. AST metrics help teams evaluate the resource utilization of their automated testing infrastructure.
A. Test Execution Time
B. Rsource Utilization
C. Test Coverage
D. Defect Detection

25. AST metrics assist in measuring the effectiveness of automated regression testing.
A. Defect Detection
B. Test Execution Time
C. Regression Testing
D. Test Coverage

26. RCA plays a crucial role in enhancing the reliability of automated tests and the accuracy of test results.
A. Optimizing Test Efficiency
B. Issue Resolution
C. Improving Test Reliability
D. None of the above

27. RCA allows teams to analyze the efficiency of their automated testing processes and identify areas for optimization.
A. Optimizing Test Efficiency
B. Improving Test Reliability
C. Issue Resolution
D. None of the above

28. RCA in AST helps pinpoint the underlying causes of test failures, anomalies, or performance bottlenecks.
A. Improving Test Reliability
B. Issue Resolution
C. Optimizing Test Efficiency
D. None of the above

29. What strategic approach empowers teams to enhance their testing efficiency through RCA?
A. Issue Resolution
B. Improving Test Reliability
C. Optimizing Test Efficiency
D. None of the above

30. What pivotal function does RCA serve in elevating the reliability of automated tests, ultimately leading to enhanced accuracy in test results?
A. Improving Test Reliability
B. Optimizing Test Efficiency
C. Issue Resolution
D. None of the above

GROUP 9

1. It involves evaluating and selecting tools that align with organizational testing needs.
   A. Decision to automate testing
   B. Test Tool Acquisition
   C. Test Planning, Design, and Development
   D. Test Program Review and Assessment

2. The process of introducing automated testing to a new project team constitutes the third phase of the ATLM.
   A. Execution and Management of Tests
   B. Test Tool Acquisition
   C. Automated Testing Introduction Process
   D. Decision to Automate Testing

3. This phase involves managing expectations, highlighting benefits, and proposing a test tool to gain management support.
   A. Test Program Review and Assessment
   B. Execution and Management of Tests
   C. Test Tool Acquisition
   D. Decision to Automate Testing

4. This phase involves assessing the current testing process and identifying areas where automation can be beneficial.
   A. Assessment and Planning
   B. Tool Selection and Setup
   C. Test Case Design and Development
   D. Maintenance and Enhancement

5. Continuous maintenance and enhancement of automated tests are performed to keep pace with changes in the software application and testing requirements.
   A. Test Case Design and Development
   B. Assessment and Planning
   C. Maintenance and Enhancement
   D. Execution and Analysis

6. Test cases are executed automatically using the chosen automation tools and frameworks.
   A. Execution and Analysis
   B. Assessment and Planning
   C. Test Case Design and Development
   D. Maintenance and Enhancement

7. It's essential to understand the needs and expectations of stakeholders regarding the automated testing process.
A. Identifying Testing Objectives
B. Understanding Stakeholder Needs
C. Identifying Automation Opportunities
D. Defining Test Requirements

8. Test cases are documented to describe the scenarios, conditions, and expected outcomes of each test.
A. Addressing Non-functional Requirements
B. Documenting Test Cases
C. Identifying Testing Objectives
D. Collaboration and Communication

9. During the requirements gathering phase, opportunities for test automation are identified.
A. Collaboration and Communication
B. Identifying Testing Objectives
C. Documenting Test Cases
D. Identifying Automation Opportunities

10. Test cases ensure that the software meets its specified requirements by covering various functionalities and scenarios.
A. Verification of Requirements
B. Assurance of Quality
C. Risk Mitigation
D. Early Defect Detection

11. Test cases uncover defects or bugs within the software, allowing for early detection and resolution.
A. Assurance of Quality
B. Verification of Requirements
C. Early Defect Detection
D. Risk Mitigation

12. that software products meet high-quality standards before release.
A. Risk Mitigation
B. Assurance of Quality
C. Early Defect Detection
D. Verification of Requirements

13.Once the defect is found and documented, it needs to be mapped to all the entities it relates to.
A. Activity 1- System integration testing
B. Activity 2 - Reporting the defects
C. Activity 3- Mapping the defects
D. Activity 4 - Re-testing the failed tests

14. Activity helps explore a lot of defects in the application, especially if it is big and complex.
A. Activity 1- System integration testing
B. Activity 2 - Reporting the defects
C. Activity 4 - Re-testing the failed tests
D. Activity 5-Regression Testing

15. We take the time to re-test each bug that was found in Activity 1.
A. Activity 1- System integration testing
B. Activity 2 - Reporting the defects
C. Activity 3- Mapping the defects
D. Activity 4 - Re-testing the failed tests

16. involves testing the application with its modules integrated as a unit
a. Activity 1- System integration testing
b. Activity 2 - Reporting the defects
c. Activity 4 - Re-testing the failed tests
d. Activity 5-Regression Testing

17. perform regression testing to ensure that the application is still stable and functioning correctly
a. Activity 1- System integration testing
b. Activity 2 - Reporting the defects
c. Activity 4 - Re-testing the failed tests
d. Activity 5-Regression Testing

18. is a crucial element of an automation framework architecture, ensuring that the architecture can handle growing demands and increasing workloads without compromising performance
A. Flexibility
B. Security
C. Scalability
D. Modularity

19. Program review and assessment are fundamental aspects of quality assurance in software development.
A. Quality Assurance
B. Compliance and Standards
C. Risk Mitigation
D. Error Detection and Correction

20. involve evaluating whether the automated testing processes comply with industry standards, best practices, and organizational guidelines.
A. Quality Assurance
B. Compliance and Standards
C. Risk Mitigation
D. Error Detection and Correction

21. It refers to checkpoints or milestones in a project or process that are managed and evaluated primarily through digital or virtual means, rather than physical meetings or reviews.
A. Quality Assurance
B. Compliance and Standards
C. Risk Mitigation
D. Virtual Quality Gates

22. teams can identify and mitigate risks associated with testing, such as incomplete test coverage, unreliable test cases, or dependencies on specific environments or configurations.
A. Error Detection and Correction
B. Performance Evaluation
C**. Risk Mitigation**
D. Virtual Quality Gates

23. It helps in evaluating their performance metrics such as execution time, resource utilization, scalability, and reliability.
A. Error Detection and Correction
B. Performance Evaluation
C. Risk Mitigation
D. Virtual Quality Gates

24. Through program review and assessment, potential errors, flaws, or inefficiencies in the automated testing scripts or frameworks can be identified.
A. Error Detection and Correction
B. Performance Evaluation
C. Risk Mitigation
D. Virtual Quality Gates

25. It allows teams to assess how well the automated testing process is performing in terms of efficiency, effectiveness, and reliability.
A. Performance Evaluation
B. Resource Allocation
C. Identifying Improvement Areas
D. Decision Making

26. By analyzing process metrics, teams can identify bottlenecks, inefficiencies, or areas for improvement in the automated testing process.
A. Quality Assurance
B. Resource Allocation
C. Identifying Improvement Areas
D. Decision Making

27. Process measurement helps ensure that the automated testing process is consistently delivering high-quality results and adhering to predefined quality standards.
A. Quality Assurance
B. Resource Allocation
C. Identifying Improvement Areas
D. Decision Making

28. It helps in allocating resources effectively by identifying areas that require more attention or investment.
A. Quality Assurance
B. Resource Allocation
C. Identifying Improvement Areas
D. Decision Making

29. Process measurement provides valuable insights for making informed decisions related to tool selection, process optimization, and resource allocation.
A. Quality Assurance
B. Resource Allocation
C. Identifying Improvement Areas
D. Decision Making

30. By measuring key performance indicators (KPIs) over time, teams can track progress and continuously improve the automated testing process.
A. Quality Assurance
B. Resource Allocation
C. Identifying Improvement Areas
D. Continuous Improvement

GROUP 10

1. This involves explaining the need for developing a document that outlines the strategy for automated testing.

    A) automated test strategy document
    B) automated standards assessment
    C) automated software testing metrics
    D) automated testing tools

2. Identifies the importance of assessing and adhering to automated standards in testing processes.

    A) automated test strategy document
    B) automated standards assessment
    C) automated software testing metrics
    D) automated testing tools

3. It involves recommending and utilizing specific metrics to measure the effectiveness and quality of automated testing.

    A) automated test strategy document
    B) automated standards assessment
    C) automated software testing metrics
    D) automated testing tools

4. Criteria for selecting appropriate automated tools and the benefits of having a well-defined business case for automation.

    A) automated test strategy document
    B) automated standards assessment
    C) automated software testing metrics
    D) automated testing tools

5. Focuses on tracking progress, incorporating defect prevention, and utilizing set of metrics to monitor effectiveness of automation.

    A) astf maintenance and progress tracking
    B) requirements determination
    C) software testing type
    D) root cause analysis

6. Focuses on tracking progress, incorporating defect prevention, and utilizing set of metrics to monitor effectiveness of automation.

   A) astf maintenance and progress tracking
   B) requirements determination
   C) software testing type
   D) root cause analysis

7. A various software testing types and the role of automated testing in production testing.

   A) astf maintenance and progress tracking
   B) requirements determination
   C) software testing type
   D) root cause analysis

8. The importance of root cause analysis in AST development and execution.

   A) astf maintenance and progress tracking
   B) requirements determination
   C) software testing type
   D) root cause analysis

9. Skills required for roles like program management, systems engineering, systems development, configuration management and quality assurance in AST projects.

   A) skills and roles
   B) leadership skills
   C) schedule and timeline management
   D) efficiency management

10. Ability to develop a test automation strategy and ensure the final quality of the product.

   A) skills and roles
   B) leadership skills
   C) schedule and timeline management
   D) efficiency management

11. This involves creating a detailed project plan, setting realistic milestone and continuously monitoring progress to identify and mitigate delays.

   A) skills and roles
   B) leadership skills
   C) schedule and timeline management
   D) efficiency management

12. Refers to the improvement of productivity or the reduction of wasted time, effort, or resources in completing tasks or processes.

   A) skills and roles
   B) leadership skills
   C) schedule and timeline management
   D) efficiency management

13. Refers to the process of helping individuals or groups navigate through challenges or obstacles by providing support, guidance and resources to reach effective solutions.

   A) problem-solving facilitation
   B) creativity and innovation
   C) requirements management
   D) facilitating refactoring

14. Refers to the generation of new ideas, concepts, or solutions that are novel, valuable and useful.

   A) problem-solving facilitation
   B) creativity and innovation
   C) requirements management
   D) facilitating refactoring

15. Is a set of techniques for documenting, analyzing, prioritizing and agreeing on requirements so that engineering teams always have current and approved requirements?

   A) problem-solving facilitation
   B) creativity and innovation
   C) requirements management
   D) facilitating refactoring

16. Unit test frameworks support refactoring efforts by providing a safety net for making changes to code.

   A) problem-solving facilitation
   B) creativity and innovation
   C) requirements management
   D) facilitating refactoring

17. It enables tracking changes made to configurations over time.

   A) traceability
   B) consistency
   C) scalability
   D) automation

18. Configuration management ensures that all systems and environments are configured consistently.

   A) traceability
   B) consistency
   C) scalability
   D) automation

19. Configuration management it becomes easier to scale infrastructure and applications by automating the provisioning and configuration of new resources.

   A) traceability
   B) consistency
   C) scalability
   D) automation

20. Tools that allow for automation of configuration tasks, which saves time and reduces the likelihood of human error.

A) traceability
B) consistency
C) scalability
D) automation

21. Facilitates collaboration among team members by providing a centralized platform for managing configurations.

A) collaboration
B) compliance and auditing
C) defect tracking
D) issue identification

22. Support compliance efforts by providing documentation of configurations and changes.

A) collaboration
B) compliance and auditing
C) defect tracking
D) issue identification

23. It is a systematic process of identifying, recording, monitoring, and managing defects or issues in a product or system throughout its development lifecycle.

A) collaboration
B) compliance and auditing
C) defect tracking
D) issue identification

24. It is a system provide centralized platform for recording and documenting software issues, bugs and enhancement requests.

A) collaboration
B) compliance and auditing
C) defect tracking
D) issue identification

25. It is a type of software testing that uncovers vulnerabilities in the system and determine that the data and resources of the system are protected from possible intruders.

A) security testing
B) server-side application security
C) client-side application security
D) system software security

26. To ensures that the server encryption and its tools are sufficient to protect the software from any disturbance.

A) security testing
B) server-side application security
C) client-side application security
D) system software security

27. This will make sure that any intruders cannot operate on any browser or any tool that is used by customers.

A) Security testing
B) server-side application security
C) client-side application security
D) system software security

28. It will check the weaknesses of the network structure, such as policies and resources.

A) security testing
B) server-side application security
C) client-side application security
D) system software security

29. Evaluate the vulnerabilities of the application based on different software, such as the operating system, database system etc.

A) security testing
B) server-side application security
C) client-side application security
D) network security

30. It is a well-regarded security assessment framework that provides a structured approach to testing the security of computer systems and networks.

A) security testing
B) server-side application security
C) client-side application security
D) open-source security testing methodology manual