Task-1->  
After coding, the software goes through rigorous testing to identify and fix bugs, check performance, and ensure it meets the specified requirements.  
Testing is critical to ensure that the software is reliable, functional, and meets user expectations. There are various types of testing, each focusing on a different aspect of the software. Types of Testing:

1. Unit Testing This is the first line of defense, where individual components or functions of the software are tested in isolation to ensure they work as intended. Unit tests are typically automated.
2. Integration Testing After unit testing, integration testing checks how different components of the system work together. It ensures that interactions between modules perform as expected.
3. System Testing System testing verifies that the entire system works as intended when all components are integrated. It covers functionality, performance, security, and other system-wide aspects.
4. Performance Testing This testing evaluates how the software performs under different conditions, such as varying loads and stresses. It includes load testing, stress testing, and scalability testing.
5. Security Testing This testing identifies vulnerabilities in the software and ensures that sensitive data (e.g., passwords, payment details) is protected from unauthorized access.

Task-2->

Agile is a set of principles for software development under which requirements and solutions evolve through collaboration between self-organizing teams. It advocates for flexible planning, iterative development, and continuous improvement. Key Principles:

• Customer Collaboration Over Contract Negotiation

• Responding to Change Over Following a Plan

• Working Software Over Comprehensive Documentation

• Individuals and Interactions Over Processes and Tools

Task-3->  
Test-driven development can improve code quality, maintainability, and efficiency, resulting in code that meets the conditions, is simple to test, and is less prone to errors. TDD also provides a faster feedback loop, allowing developers to identify and remove stumbling blocks early in development. TDD integrates with Agile concepts to improve software development and provide higher-quality solutions.

Task-4->  
Unit testing is a crucial part of software development as it helps catch defects early in the process, reduce costs, improve code quality, and facilitate refactoring.

Failing to write Unit test can affect->  
Early Detection of issues  
Code Quality  
Development speed  
Safety of changing code

Task-5->

Task-6->

Integration Testing will help to verify the connection between two or more modules while Unit Testing is a testing method by which individual units of code are tested.

Issues found in Unit Testing can be instantly fixed but issues found in Integration Testing take a longer time and the cost to fix is higher, as it becomes difficult to find where exactly a bug has arisen.

Unit Testing is a kind of white box testing, whereas Integration Testing is a kind of black-box testing.

Task-7->