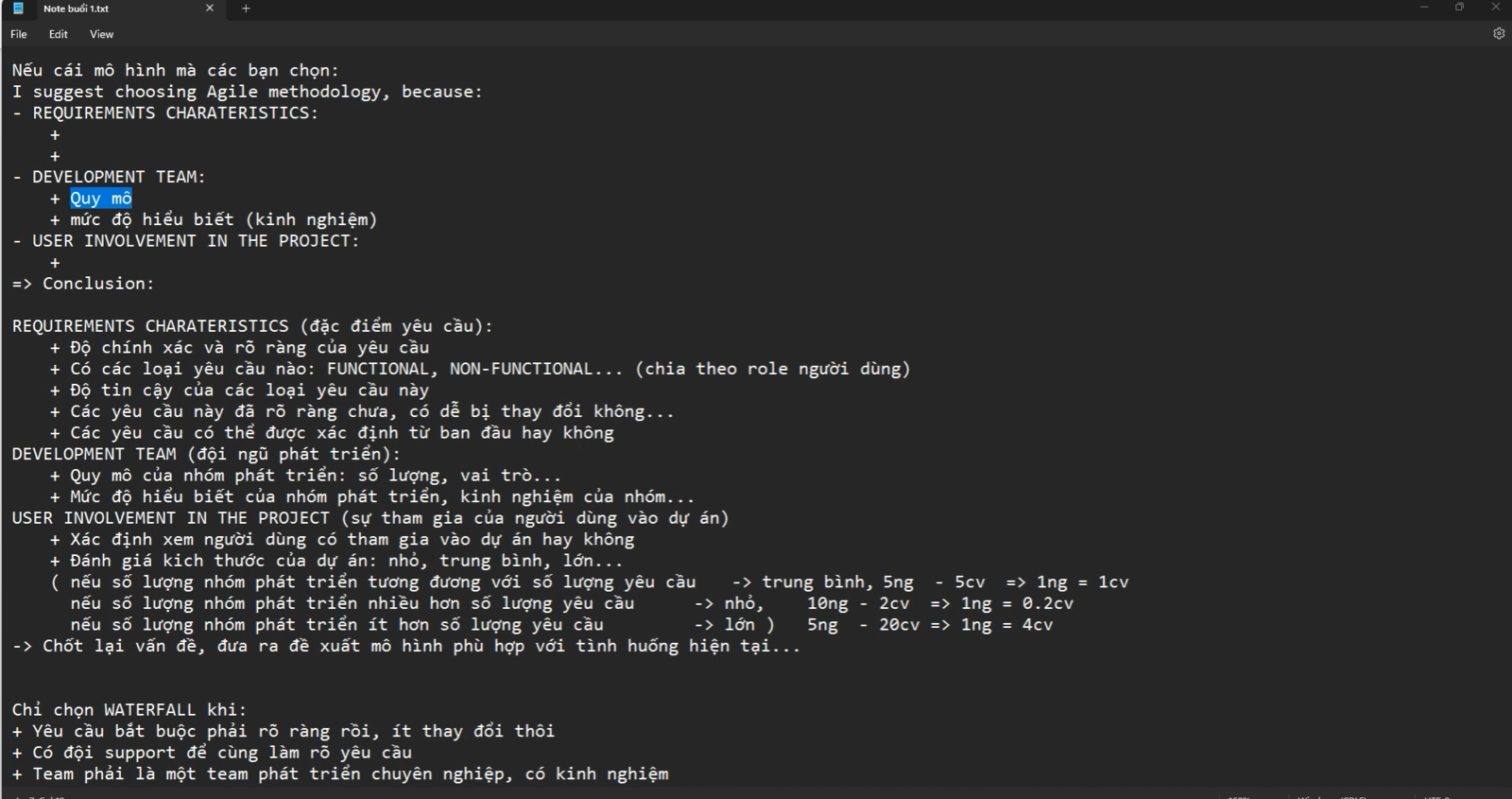
**CÂU 1:**

****

* Chỉ chọn WATERFALL khi:

+ Yêu cầu bắt buộc phải rõ rang rồi, ít thay đổi thôi

+ Có đội support để cùng làm rõ yêu cầu

+ Team phải là một team phát triển chuyên nghiệp, có kinh nghiệm

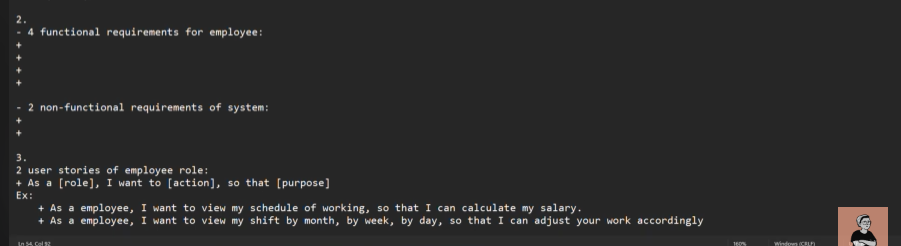
+ Quy mô dự án: nhỏ, trung bình

* Còn lại: Chọn AGILE

**CÂU 2:**

* Functional Requirement là những function phải sử dụng đến data
* Non-functional Requirement là những thứ đạt được khi phải thực thi
* Nếu hỏi quá số lượng fuction đã cho, thì phải bịa dựa trên hệ thống

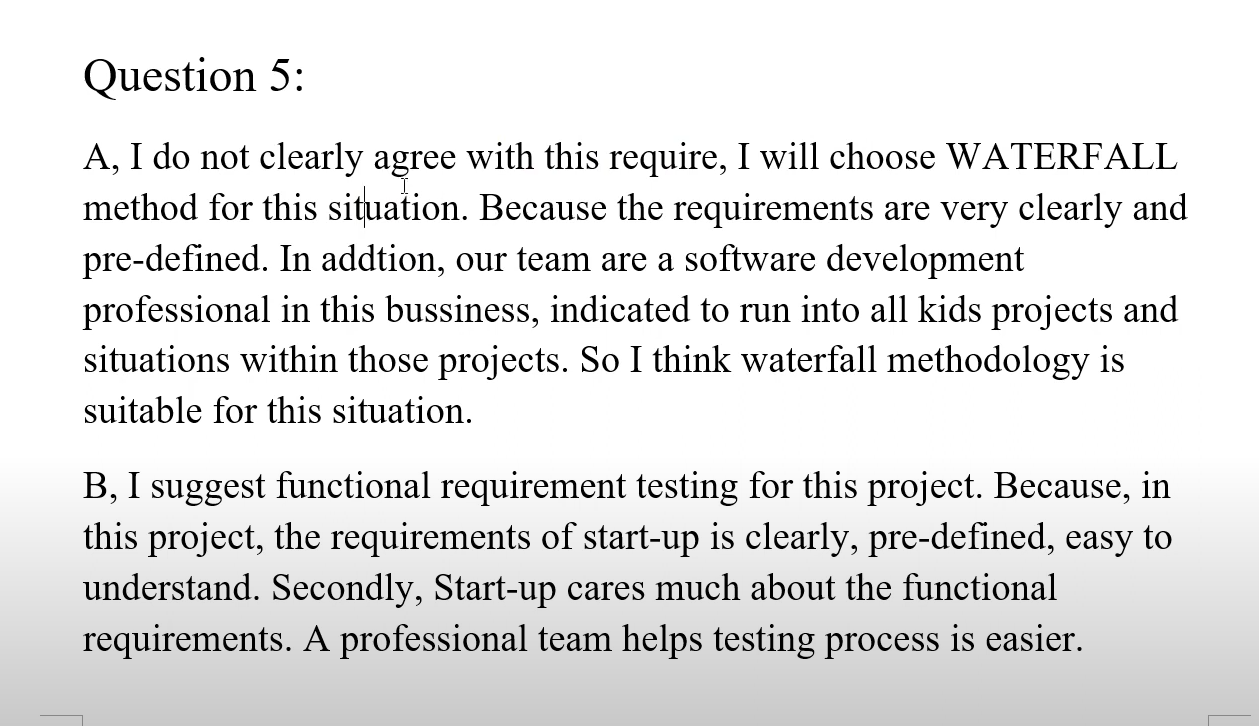
**CÂU 3:**

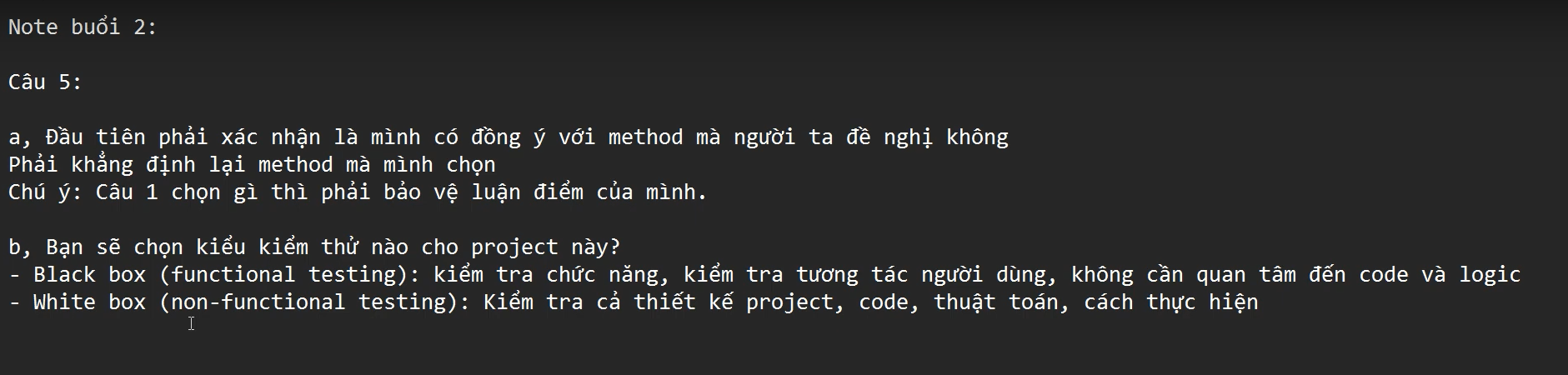


**CÂU 4:**



**CÂU 5:**





**When to use agile vs. waterfall**

Waterfall project management best suits well-defined projects with clearly specified requirements, limited complexity, and a definitive timeline. Waterfall works well when the customer's demands are precise and when there are no significant changes in scope or technology during the project.

The waterfall model fits projects with the following characteristics:

* Simple scope and requirement gathering
* Clear and linear sequence of tasks
* Predictable deliverables based on set deadlines
* Structured processes
* Rigid quality control measures
* Long-term commitment from all parties involved

Agile project management is a good fit in cases where the end goal may be unclear or difficult to define, when complex systems require frequent feedback loops, or when timelines and budgets are tight. It is also particularly effective for developing software applications since it allows for quick iteration and testing along the way. Additionally, agile can be helpful when dealing with teams who need to collaborate intensively, such as those located in different locations.

Agile project management can be an excellent fit for many teams and projects, especially those that meet the criteria below:

* Complex deliverables
* Frequent iterations and refinements
* Rapid delivery times
* Emergent requirements
* Collaborative environments
* Multiple stakeholders

Khi nào nên sử dụng agile so với waterfall

Quản lý dự án theo phương pháp Waterfall phù hợp nhất với các dự án được xác định rõ ràng với các yêu cầu được chỉ định rõ ràng, độ phức tạp hạn chế và mốc thời gian xác định. Waterfall hoạt động tốt khi nhu cầu của khách hàng chính xác và không có thay đổi đáng kể nào về phạm vi hoặc công nghệ trong suốt dự án.

Mô hình Waterfall phù hợp với các dự án có các đặc điểm sau:

Phạm vi và thu thập yêu cầu đơn giản

Trình tự nhiệm vụ rõ ràng và tuyến tính

Các sản phẩm có thể dự đoán được dựa trên thời hạn đã đặt

Quy trình có cấu trúc

Các biện pháp kiểm soát chất lượng nghiêm ngặt

Cam kết dài hạn từ tất cả các bên liên quan

Quản lý dự án theo phương pháp Agile phù hợp trong các trường hợp mục tiêu cuối cùng có thể không rõ ràng hoặc khó xác định, khi các hệ thống phức tạp yêu cầu vòng phản hồi thường xuyên hoặc khi mốc thời gian và ngân sách eo hẹp. Phương pháp này cũng đặc biệt hiệu quả để phát triển các ứng dụng phần mềm vì nó cho phép lặp lại và thử nghiệm nhanh chóng trong suốt quá trình. Ngoài ra, phương pháp Agile có thể hữu ích khi giải quyết các nhóm cần cộng tác chặt chẽ, chẳng hạn như các nhóm ở các địa điểm khác nhau.

Quản lý dự án linh hoạt có thể phù hợp tuyệt vời với nhiều nhóm và dự án, đặc biệt là những nhóm và dự án đáp ứng các tiêu chí dưới đây:

Các sản phẩm phức tạp

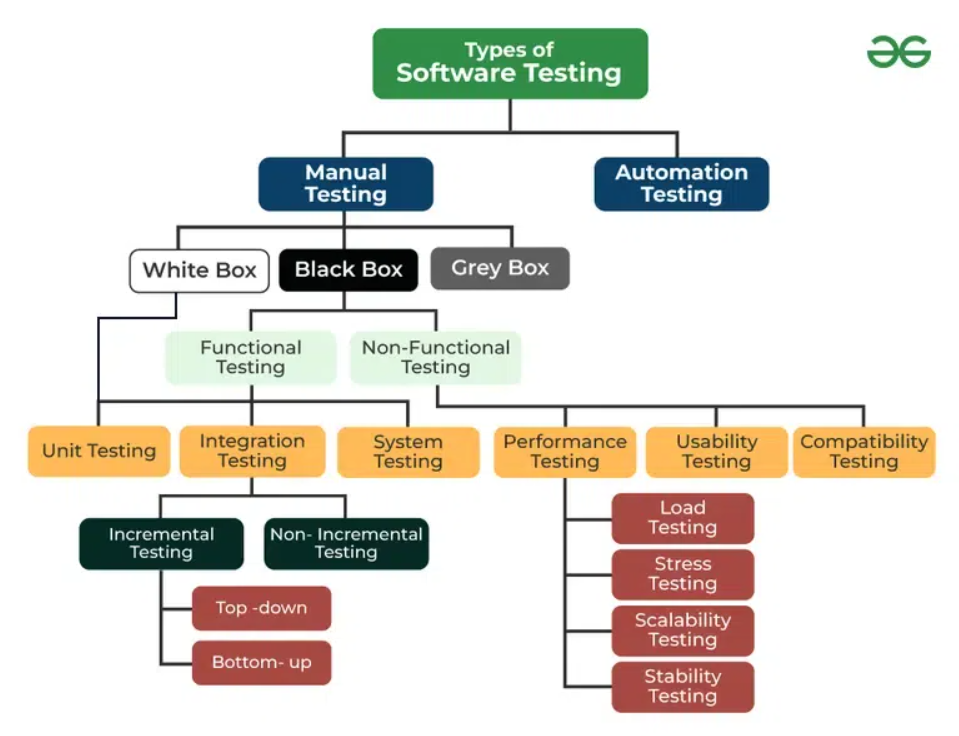
Lặp lại và tinh chỉnh thường xuyên

Thời gian giao hàng nhanh chóng

Các yêu cầu khẩn cấp

Môi trường cộng tác

Nhiều bên liên quan



**1. White Box Testing**

***White box testing****techniques analyze the internal structures the used data structures, internal design, code structure, and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing clear box testing or structural testing. White Box Testing is also known as transparent testing or open box testing.*

White box testing is a software testing technique that involves testing the internal structure and workings of a software application. The tester has access to the source code and uses this knowledge to design test cases that can verify the correctness of the software at the code level.

**Advantages of White box Testing:**

* **Thorough Testing**: White box testing is thorough as the entire code and structures are tested.
* **Code Optimization:**It results in the optimization of code removing errors and helps in removing extra lines of code.
* **Early Detection of Defects:**It can start at an earlier stage as it doesn’t require any interface as in the case of black box testing.
* **Integration with SDLC:**White box testing can be easily started in the Software Development Life Cycle.
* **Detection of Complex Defects:**Testers can identify defects that cannot be detected through other testing techniques.

**2. Black Box Testing**

*Black-box testing is a type of software testing in which the tester is not concerned with the internal knowledge or implementation details of the software but rather focuses on validating the functionality based on the provided specifications or requirements.*

**Advantages of Black Box Testing:**

* The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing.
* It is efficient for implementing the tests in the larger system.
* Tests are executed from the user’s or client’s point of view.
* Test cases are easily reproducible.
* It is used to find the ambiguity and contradictions in the functional specifications.

**1. Kiểm thử hộp trắng**

***Kỹ thuật kiểm thử hộp trắng****phân tích cấu trúc bên trong, cấu trúc dữ liệu được sử dụng, thiết kế bên trong, cấu trúc mã và hoạt động của phần mềm thay vì chỉ chức năng như trong kiểm thử hộp đen. Nó còn được gọi là thử nghiệm hộp kính, thử nghiệm hộp trong hoặc thử nghiệm cấu trúc. Kiểm thử hộp trắng còn được gọi là kiểm thử minh bạch hoặc kiểm thử hộp mở.*

Kiểm thử hộp trắng là một kỹ thuật kiểm thử phần mềm liên quan đến việc kiểm tra cấu trúc bên trong và hoạt động của một ứng dụng phần mềm. Người kiểm tra có quyền truy cập vào mã nguồn và sử dụng kiến ​​thức này để thiết kế các trường hợp kiểm thử có thể xác minh tính chính xác của phần mềm ở cấp độ mã.

**Ưu điểm của kiểm thử hộp trắng:**

* **Kiểm tra kỹ lưỡng** : Kiểm tra hộp trắng là kỹ lưỡng vì toàn bộ mã và cấu trúc đều được kiểm tra.
* **Tối ưu hóa mã:** Nó giúp tối ưu hóa việc loại bỏ mã và giúp loại bỏ các dòng mã bổ sung.
* **Phát hiện sớm các khiếm khuyết:** Nó có thể bắt đầu ở giai đoạn sớm hơn vì nó không yêu cầu bất kỳ giao diện nào như trong trường hợp kiểm thử hộp đen.
* **Tích hợp với SDLC:** Kiểm thử hộp trắng có thể dễ dàng bắt đầu trong Vòng đời phát triển phần mềm.
* **Phát hiện các khiếm khuyết phức tạp:** Người kiểm thử có thể xác định các khiếm khuyết không thể phát hiện được thông qua các kỹ thuật kiểm thử khác.

**2. Kiểm tra hộp đen**

*Kiểm thử hộp đen là một loại kiểm thử phần mềm trong đó người kiểm thử không quan tâm đến kiến ​​thức nội bộ hoặc chi tiết triển khai của phần mềm mà tập trung vào việc xác thực chức năng dựa trên các thông số kỹ thuật hoặc yêu cầu được cung cấp.*

**Ưu điểm của kiểm thử hộp đen:**

* Người kiểm thử không cần phải có thêm kiến ​​thức chức năng hoặc kỹ năng lập trình để thực hiện Kiểm thử hộp đen.
* Nó hiệu quả để thực hiện các thử nghiệm trong hệ thống lớn hơn.
* Các thử nghiệm được thực hiện theo quan điểm của người dùng hoặc khách hàng.
* Các trường hợp thử nghiệm có thể dễ dàng tái tạo.
* Nó được sử dụng để tìm ra sự mơ hồ và mâu thuẫn trong các đặc tả chức năng.

**Types of Black Box Testing**

1. **Functional Testing**
2. **Non-Functional Testing**

**1. Functional Testing**

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on the simulation of actual system usage but does not develop any system structure assumptions. The article focuses on discussing function testing.

**Benefits of Functional Testing**

* **Bug-free product:**Functional testing ensures the delivery of a bug-free and high-quality product.
* **Customer satisfaction:**It ensures that all requirements are met and ensures that the customer is satisfied.
* **Testing focused on specifications:**Functional testing is focused on specifications as per customer usage.
* **Proper working of application:**This ensures that the application works as expected and ensures proper working of all the functionality of the application.
* **Improves quality of the product:**Functional testing ensures the security and safety of the product and improves the quality of the product.

**2. Non-Functional Testing**

**Non-functional Testing**is a type of Software Testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects that are not tested in functional testing. Non-functional testing is a software testing technique that checks the non-functional attributes of the system. Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing.

**Benefits of Non-functional Testing**

* **Improved performance:**Non-functional testing checks the performance of the system and determines the performance bottlenecks that can affect the performance.
* **Less time-consuming:**Non-functional testing is overall less time-consuming than the other testing process.
* **Improves user experience:**Non-functional testing like Usability testing checks how easily usable and user-friendly the software is for the users. Thus, focus on improving the overall user experience for the application.
* **More secure product:**As non-functional testing specifically includes security testing that checks the security bottlenecks of the application and how secure is the application against attacks from internal and external sources.

**Types of Functional Testing**

1. **Unit Testing**
2. **Integration Testing**
3. **System Testing**

**1. Unit Testing**

Unit testing is a method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended. Unit tests are usually automated and are designed to test specific parts of the code, such as a particular function or method. Unit testing is done at the lowest level of the software development process , where individual units of code are tested in isolation.

*Note: Unit Testing basically Included in both White Box Testing and Black Box Testing.*

**Advantages of Unit Testing:**

Some of the advantages of Unit Testing are listed below.

* It helps to identify bugs early in the development process before they become more difficult and expensive to fix.
* It helps to ensure that changes to the code do not introduce new bugs.
* It makes the code more modular and easier to understand and maintain.
* It helps to improve the overall quality and reliability of the software.

***Note:****Some popular frameworks and tools that are used for unit testing include****JUnit****,****NUnit,****and****xUnit.***

* *It’s important to keep in mind that Unit Testing is only one aspect of software testing and it should be used in combination with other types of testing such as integration testing, functional testing, and acceptance testing to ensure that the software meets the needs of its users.*
* *It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.*

**Example:**

1. *In a program we are checking if the loop, method, or function is working fine.*
2. *Misunderstood or incorrect, arithmetic precedence.*
3. *Incorrect initialization.*

**2. Integration Testing**

Integration testing is a method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined. Integration testing is typically done after unit testing and before functional testing and is used to verify that the different units of the software work together as intended.

**Different Ways of Performing Integration Testing:**

Different ways of Integration Testing are discussed below.

* Top-down integration testing: It starts with the highest-level modules and differentiates them from lower-level modules.
* Bottom-up integration testing: It starts with the lowest-level modules and integrates them with higher-level modules.
* Big-Bang integration testing: It combines all the modules and integrates them all at once.
* Incremental integration testing: It integrates the modules in small groups, testing each group as it is added.

**Advantages of Integrating Testing**

* It helps to identify and resolve issues that may arise when different units of the software are combined.
* It helps to ensure that the different units of the software work together as intended.
* It helps to improve the overall reliability and stability of the software.
* It’s important to keep in mind that Integration testing is essential for complex systems where different components are integrated.
* As with unit testing, integration testing is only one aspect of software testing and it should be used in combination with other types of testing such as unit testing, functional testing, and acceptance testing to ensure that the software meets the needs of its users.

The **objective**is to take unit-tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.

**Integration testing is of four types: (i) Top-down (ii) Bottom-up (iii) Sandwich (iv) Big-Bang**

**Example:**

1. ***Black Box testing:****It is used for validation. In this, we ignore internal working mechanisms and focus on “what is the output?”*
2. ***White box testing:****It is used for verification. In this, we focus on internal mechanisms i.e. how the output is achieved.*

**3. System Testing**

System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution. It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users. This type of testing is performed after the integration testing and before the acceptance testing.

***System Testing****is a type of software testing that is performed on a completely integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated.*

**Advantages of System Testing:**

* The testers do not require more knowledge of programming to carry out this testing.
* It will test the entire product or software so that we will easily detect the errors or defects that cannot be identified during the unit testing and integration testing.
* The testing environment is similar to that of the real-time production or business environment.
* It checks the entire functionality of the system with different test scripts and also it covers the technical and business requirements of clients.
* After this testing, the product will almost cover all the possible bugs or errors and hence the development team will confidently go ahead with acceptance testing.

**Các loại thử nghiệm chức năng**

1. **Kiểm tra đơn vị**
2. **Thử nghiệm hội nhập**
3. **Thử nghiệm hệ thống**

**1. Kiểm tra đơn vị**

Kiểm thử đơn vị là phương pháp kiểm thử các đơn vị hoặc thành phần riêng lẻ của một ứng dụng phần mềm. Nó thường được thực hiện bởi các nhà phát triển và được sử dụng để đảm bảo rằng các đơn vị riêng lẻ của phần mềm hoạt động như dự định. Kiểm thử đơn vị thường được tự động hóa và được thiết kế để kiểm tra các phần cụ thể của mã, chẳng hạn như một chức năng hoặc phương pháp cụ thể. Kiểm thử đơn vị được thực hiện ở cấp độ thấp nhất của quy trình phát triển phần mềm , trong đó các đơn vị mã riêng lẻ được kiểm thử riêng biệt.

*Lưu ý: Kiểm thử đơn vị về cơ bản được bao gồm trong cả Kiểm thử hộp trắng và Kiểm thử hộp đen.*

**Ưu điểm của thử nghiệm đơn vị:**

Một số ưu điểm của Kiểm tra đơn vị được liệt kê dưới đây.

* Nó giúp xác định sớm các lỗi trong quá trình phát triển trước khi chúng trở nên khó sửa và tốn kém hơn.
* Nó giúp đảm bảo rằng những thay đổi đối với mã không gây ra lỗi mới.
* Nó làm cho mã trở nên mô-đun hơn, dễ hiểu và dễ bảo trì hơn.
* Nó giúp cải thiện chất lượng tổng thể và độ tin cậy của phần mềm.

***Lưu ý:****Một số khung và công cụ phổ biến được sử dụng để kiểm tra đơn vị bao gồm****JUnit****,****NUnit****và****xUnit.***

* *Điều quan trọng cần lưu ý là Kiểm thử đơn vị chỉ là một khía cạnh của kiểm thử phần mềm và nó nên được sử dụng kết hợp với các loại kiểm thử khác như kiểm thử tích hợp, kiểm thử chức năng và kiểm thử chấp nhận để đảm bảo rằng phần mềm đáp ứng nhu cầu của người dùng. .*
* *Nó tập trung vào đơn vị thiết kế phần mềm nhỏ nhất. Trong phần này, chúng tôi kiểm tra một đơn vị riêng lẻ hoặc một nhóm các đơn vị có liên quan với nhau. Nó thường được lập trình viên thực hiện bằng cách sử dụng đầu vào mẫu và quan sát đầu ra tương ứng của nó.*

**Ví dụ:**

1. *Trong một chương trình, chúng ta đang kiểm tra xem vòng lặp, phương thức hoặc hàm có hoạt động tốt hay không.*
2. *Hiểu sai hoặc không chính xác, ưu tiên số học.*
3. *Khởi tạo không chính xác.*

**2. Kiểm tra tích hợp**

Kiểm thử tích hợp là một phương pháp kiểm tra xem các đơn vị hoặc thành phần khác nhau của ứng dụng phần mềm tương tác với nhau như thế nào. Nó được sử dụng để xác định và giải quyết mọi vấn đề có thể phát sinh khi các đơn vị khác nhau của phần mềm được kết hợp. Kiểm thử tích hợp thường được thực hiện sau kiểm thử đơn vị và trước kiểm thử chức năng và được sử dụng để xác minh rằng các đơn vị khác nhau của phần mềm hoạt động cùng nhau như dự định.

**Các cách khác nhau để thực hiện kiểm tra tích hợp:**

Các cách kiểm thử tích hợp khác nhau sẽ được thảo luận dưới đây.

* Kiểm thử tích hợp từ trên xuống: Nó bắt đầu với các mô-đun cấp cao nhất và phân biệt chúng với các mô-đun cấp thấp hơn.
* Kiểm thử tích hợp từ dưới lên: Nó bắt đầu với các mô-đun cấp thấp nhất và tích hợp chúng với các mô-đun cấp cao hơn.
* Thử nghiệm tích hợp Big-Bang: Nó kết hợp tất cả các mô-đun và tích hợp tất cả chúng cùng một lúc.
* Thử nghiệm tích hợp tăng dần: Nó tích hợp các mô-đun trong các nhóm nhỏ, kiểm tra từng nhóm khi nó được thêm vào.

**Ưu điểm của việc tích hợp thử nghiệm**

* Nó giúp xác định và giải quyết các vấn đề có thể phát sinh khi kết hợp các đơn vị khác nhau của phần mềm.
* Nó giúp đảm bảo rằng các đơn vị khác nhau của phần mềm hoạt động cùng nhau như dự kiến.
* Nó giúp cải thiện độ tin cậy và tính ổn định tổng thể của phần mềm.
* Điều quan trọng cần ghi nhớ là thử nghiệm tích hợp là điều cần thiết cho các hệ thống phức tạp nơi các thành phần khác nhau được tích hợp.
* Giống như kiểm thử đơn vị, kiểm thử tích hợp chỉ là một khía cạnh của kiểm thử phần mềm và nó nên được sử dụng kết hợp với các loại kiểm thử khác như kiểm thử đơn vị, kiểm thử chức năng và kiểm thử chấp nhận để đảm bảo rằng phần mềm đáp ứng nhu cầu của người dùng.

Mục **tiêu** là lấy các thành phần đã được kiểm thử đơn vị và xây dựng cấu trúc chương trình đã được thiết kế quy định. Kiểm thử tích hợp là kiểm thử trong đó một nhóm các thành phần được kết hợp để tạo ra đầu ra.

**Kiểm thử tích hợp có bốn loại: (i) Từ trên xuống (ii) Từ dưới lên (iii) Sandwich (iv) Big-Bang**

**Ví dụ:**

1. ***Kiểm tra hộp đen:****Nó được sử dụng để xác nhận. Trong phần này, chúng tôi bỏ qua các cơ chế làm việc nội bộ và tập trung vào “đầu ra là gì?”*
2. ***Kiểm thử hộp trắng:****Nó được sử dụng để xác minh. Trong phần này, chúng tôi tập trung vào các cơ chế nội bộ, tức là cách đạt được đầu ra.*

**3. Kiểm tra hệ thống**

Kiểm thử hệ thống là một loại kiểm thử phần mềm nhằm đánh giá chức năng và hiệu suất tổng thể của một giải pháp phần mềm hoàn chỉnh và được tích hợp đầy đủ. Nó kiểm tra xem hệ thống có đáp ứng các yêu cầu đã chỉ định hay không và liệu nó có phù hợp để phân phối cho người dùng cuối hay không. Loại thử nghiệm này được thực hiện sau thử nghiệm tích hợp và trước thử nghiệm chấp nhận.

***Kiểm tra hệ thống****là một loại kiểm thử phần mềm được thực hiện trên một hệ thống tích hợp hoàn chỉnh để đánh giá sự tuân thủ của hệ thống với các yêu cầu tương ứng. Trong kiểm thử hệ thống, các thành phần đã vượt qua kiểm thử tích hợp được lấy làm đầu vào. Mục tiêu của thử nghiệm tích hợp là phát hiện bất kỳ sự bất thường nào giữa các đơn vị được tích hợp.*

**Ưu điểm của kiểm tra hệ thống:**

* Người kiểm thử không cần thêm kiến ​​thức về lập trình để thực hiện việc kiểm thử này.
* Nó sẽ test toàn bộ sản phẩm hoặc phần mềm để chúng ta dễ dàng phát hiện những lỗi hoặc khiếm khuyết không thể xác định được trong quá trình unit testing và Integration testing.
* Môi trường thử nghiệm tương tự như môi trường sản xuất hoặc kinh doanh thời gian thực.
* Nó kiểm tra toàn bộ chức năng của hệ thống bằng các tập lệnh thử nghiệm khác nhau và cũng bao gồm các yêu cầu kỹ thuật và kinh doanh của khách hàng.
* Sau quá trình thử nghiệm này, sản phẩm sẽ gần như loại bỏ được tất cả các lỗi hoặc lỗi có thể xảy ra và do đó nhóm phát triển sẽ tự tin tiến hành thử nghiệm chấp nhận.