**Chương 1: Tổng quan về lý thuyết.**

1. **Software Testing Overview**
2. **Introduction to Software Testing.**
3. ***Software Testing là gì?***

* Kiểm thử phần mềm là một quá trình thực thi một chương trình hoặc ứng dụng với mục đích tìm ra lỗi phần mềm. Và được sử dụng để xác định tính đúng đắn, đầy đủ và chất lượng của phần mềm máy tính đã phát triển.

1. ***Objectives of Testing***

* Để tìm và ngăn ngừa lỗi trong sản phẩm phần mềm.
* Giảm rủi ro về chất lượng phầm mềm.
* Để có được niềm tin của khách hàng bằng cách cung cấp cho họ sản phẩm chất lượng.
* Để đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu của doanh nghiệp và người dùng.

1. **Software Development Process.**

* Quy trình phát triển phần mềm là một tập hợp các hoạt động tổ chức mà mục đích của chúng là xây dựng và phát triển phần mềm.
* Có 4 thao tác là nền tảng của hầu hết các quy trình phát triển phần mềm:
* Đặc tả phần mềm.
* Quy trình phát triển phần mềm.
* Đánh giá phần mềm.
* Tiến hóa phần mềm.

1. **Principle of testing.**
2. nguyên tắc của Testing:

* Testing shows presence of defects
* Exhaustive testing is impossible
* Early testing
* Defect clustering
* Pesticide paradox
* Testing is context dependent
* Absence – of – errors fallacy

1. **Error / Fault / Failure.**

* Error:
* Đề cập đến sự khác biệt giữa sản phẩm thực tế và sản phẩm mong đợi.
* Các nhà phát triển và kỹ sư kiểm tra tự động hóa nêu ra lỗi.
* Sự cố trong mã dẫn đến lỗi.

VD: Lỗi cú pháp, giao diện người dùng, phần cứng,…

* Fault:
* Là một tình trạng khiến phần mềm không thực hiện được chức năng cần thiết của nó.
* Sai lầm của con người gây ra lỗi.

VD: Lỗi Logic, chức năng, bảo mật, hiệu suất,…

* Failure:
* Là hệ thống hoặc thành phần không có khả năng thực hiện chức năng cần thiết theo đặc điểm kỹ thuật của nó.
* Người dùng cuối phát hiện ra một vấn đề trong sản phẩm.

VD: trong ứng dụng ngân hàng nếu mô-đun **Chuyển số tiền** không hoạt động cho người dùng cuối khi người dùng cuối cố gắng **chuyển tiền** , nút gửi sẽ không hoạt động. Do đó, đây là một **thất bại**

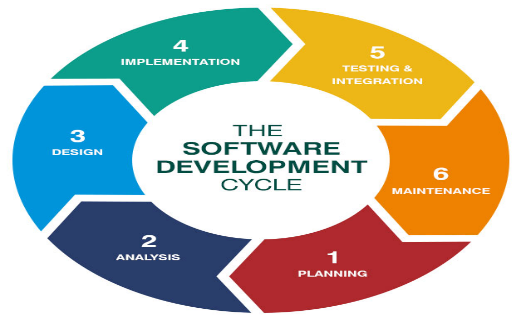
1. **What is Verification & Validation.**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Xác minh giải quyết mối lo ngại:  “Bạn có đang xây dựng nó đúng không?” | Xác thực giải quyết mối quan tâm:  “ Bạn có đang xây dựng đúng không?” |
| Đảm bảo rằng hệ thống phần mềm đáp ứng tất cả các chức năng. | Đảm bảo rằng các chức năng đáp ứng hành vi dự định. |
| Việc xác minh diễn ra trước tiên và bao gồm việc kiểm tra tài liệu, mã,… | Việc xác nhận xảy ra sau khi xác minh, chủ yếu liên quan đến việc kiểm tra tổng thể sản phẩm. |
| Do nhà phát triển thực hiện. | Thực hiện bởi người thử nghiệm. |

1. **Difference between QA & QC.**

|  |  |  |
| --- | --- | --- |
|  | **QA (Quality Assurance)** | **QC ( Quality Control)** |
| ***Khái niệm*** | Người chịu trách nhiệm đảm bảo chất lượng sản phẩm thông qua việc đưa ra quy trình giữa các bên liên quan. | Người chịu trách nhiệm thực hiện kiểm tra chất lượng phần mềm.  Vị trí: manual QC (ko cần kỹ năng lập trình), auto QC (cần lập trình) |
| ***Nhiệm vụ*** | - Đưa ra quy trình phát triển sản phẩm.  - K.tra, audit việcc thực thi quy trình của bộ phận trong nhóm làm sản phẩm có đúng quy trình QA đề ra không.  - Nhắc nhở đội ngũ phát triển sản phẩm việc tuân thủ theo quy trình đưa ra.  - Điều chỉnh, thay đổi quy trình phù hợp. | -Tìm hiểu, phân tích tài liệu mô tả về hệ thống và thiết kế test case,và thực hiện việc test phần mềm trước khi giao cho khách hàng.  - Lên kế hoạch kiểm thử  - Viết Script cho automation test (nếu có áp dụng kiểm thử tự động)  - Sử dụng các test tool để tạo và thực hiện các test case/script chi tiết.  - Phối hợp với nhóm lập trình trong việc fix bug và báo cáo chi tiết cho Project Manager hoặc các bên liên quan khác tuỳ dự án. |
| ***Kỹ năng*** | -Hiểu sâu về kiến trúc hệ thống phần mềm.  - Kỹ năng tổ chức, tư duy logic, phân tích, làm trên số liệu tốt  - Kỹ năng giao tiếp trong team  - Hiểu rõ về chứng chỉ CMMI, ISO.. trong phần mềm để xây dựng quy trình chuẩn. | - Code  - Giám sát (nhah chóng phát hiện các lỗi)  - Giao tiếp (phản hồi với dev, pro manager)  - Quản lý nhân viên (đôn đốc nhân viên) |

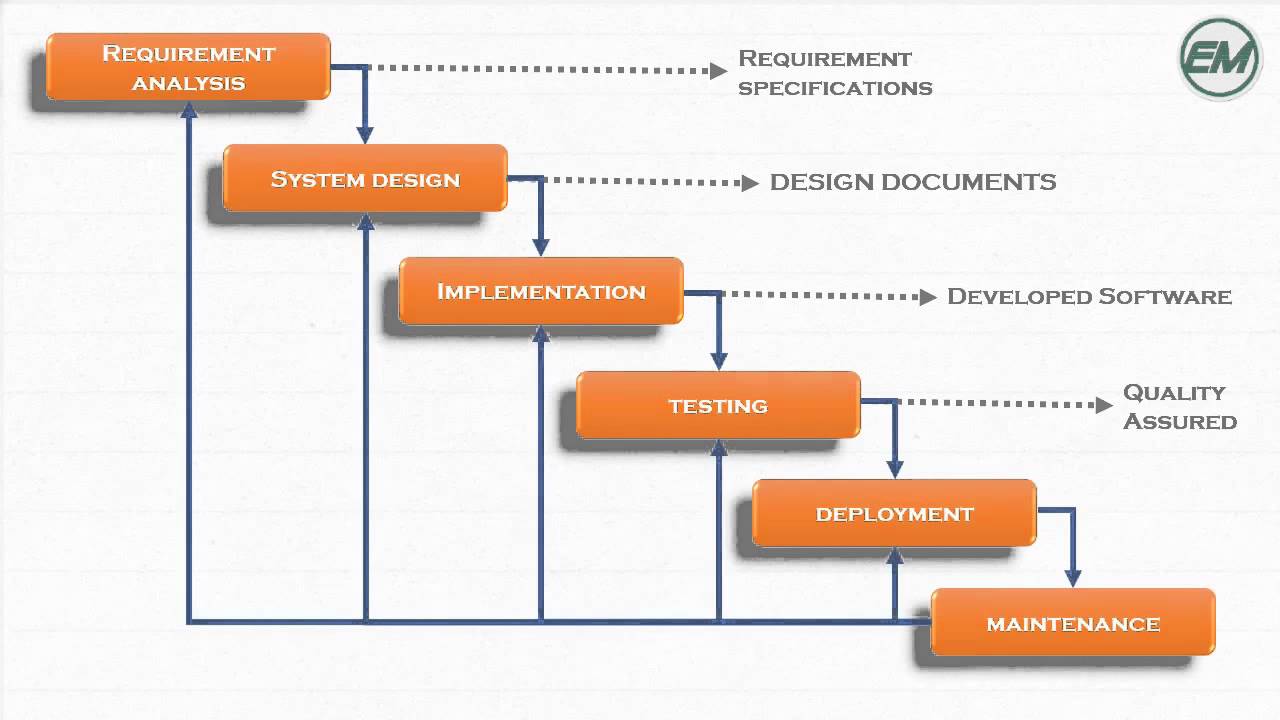
1. **Software Development Life Cycle (SDLC)**
2. **SDLC Phases.**



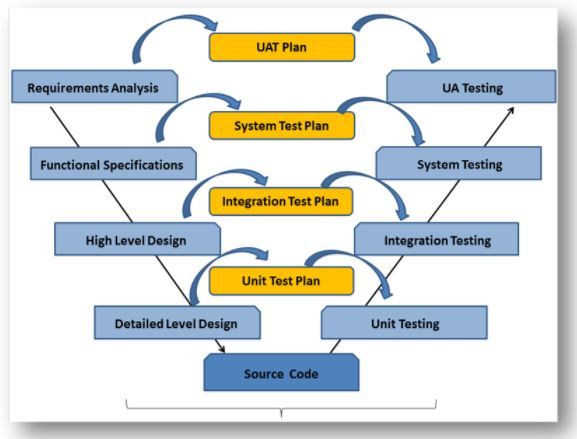
* SDLC, Vòng đời phát triển phần mềm là một quá trình được ngành công nghiệp phần mềm sử dụng để thiết kế, phát triển và kiểm tra phần mềm chất lượng cao.
* Mục tiêu của SDLC là tạo ra một phần mềm chất lượng cao đáp ứng hoặc vượt quá mong đợi của khách hàng, hoàn thành trong thời gian và chi phí ước tính.
* SDLC có 6 giai đoạn:
  + Requirement gathering and analysis
  + Design
  + Implementation or coding
  + Testing
  + Deployment
  + Maintenance

1. **Các mô hình phổ biến của SDLC.**

* Water Fall.



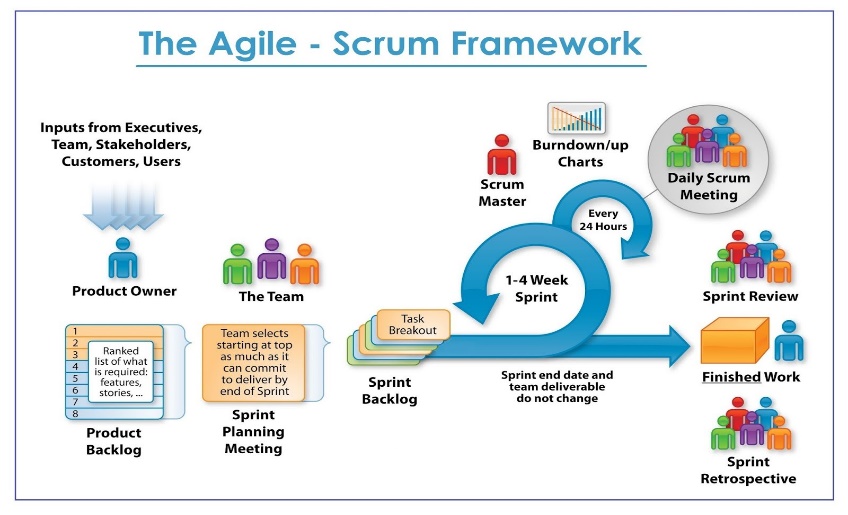
* V Model.



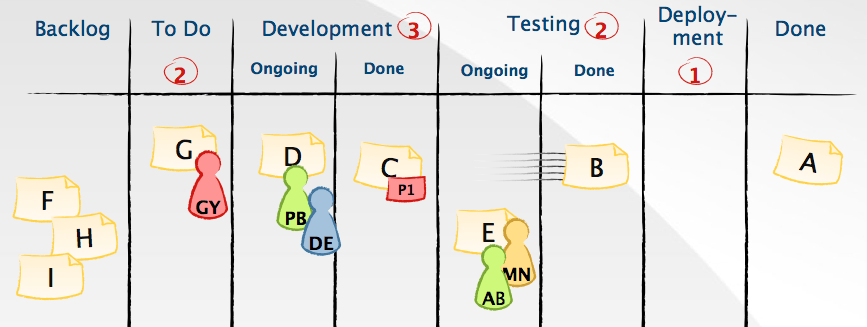
* Agile Model.



1. **Scrum Methodology.**



1. **Kanban Methodology.**



1. **Software Testing Type**
2. **Manual testing.**

* Kiểm thử thủ công bao gồm kiểm tra phần mềm theo cách thủ công, tức là không sử dụng bất kỳ công cụ tự động nào hoặc bất kỳ tập lệnh nào. Trong loại này, người kiểm tra đảm nhận vai trò của người dùng cuối và kiểm tra phần mềm để xác định bất kỳ hành vi hoặc lỗi không mong muốn nào.
* Có các giai đoạn khác nhau để kiểm tra thủ công như kiểm thử đơn vị, kiểm thử tích hợp, kiểm tra hệ thống và kiểm thử chấp nhận người dùng.

1. **Automation testing.**

* Kiểm thử tự động, còn được gọi là Test Automation, là khi người kiểm thử viết các kịch bản và sử dụng một phần mềm khác để kiểm tra sản phẩm. Quá trình này liên quan đến việc tự động hóa quy trình thủ công. Kiểm thử tự động được sử dụng để chạy lại các kịch bản kiểm tra đã được thực hiện theo cách thủ công, nhanh chóng và lặp đi lặp lại.
* Ngoài kiểm thử hồi quy, kiểm thử tự động hóa cũng được sử dụng để kiểm tra ứng dụng từ quan điểm tải, hiệu suất và căng thẳng. Nó làm tăng phạm vi kiểm tra, cải thiện độ chính xác và tiết kiệm thời gian và tiền bạc so với kiểm tra thủ công.

1. **Software Testing Methodologies.**
2. **Static Testing.**

* Thử nghiệm tĩnh là loại kiểm tra trong đó code không được thực hiện. Nó có thể được thực hiện bằng tay hoặc bằng một bộ công cụ. Loại kiểm tra này thực hiện kiểm tra code, tìm ra lỗi, các lỗi code và mã độc tiềm ẩn trong ứng dụng phần mềm.
* Static test bắt đầu sớm hơn trong vòng đời phát triển, và do đó nó còn được gọi là thử nghiệm xác minh (verification testing).

#### Các kỹ thuật Thử nghiệm tĩnh bao gồm:

* Inspection
* Walk-through
* Technical reviews
* Informal reviews

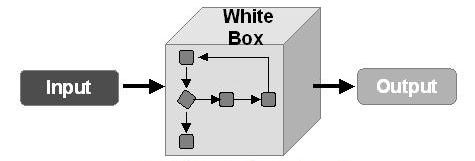
1. **Dynamic Testing**

* Thử nghiệm động được thực hiện khi code đang ở chế độ thực thi. thực thi chạy chương trình ứng dụng. Khi code được thực thi, thì đầu vào được truyền một giá trị, kết quả hoặc đầu ra của việc thực hiện được so sánh với kết quả dự kiến ban đầu đã đưa ra.
* Thử nghiệm dynamic còn được gọi là thử nghiệm xác nhận (Validation testing), đánh giá sản phẩm.

#### Các loại kỹ thuật Thử nghiệm động

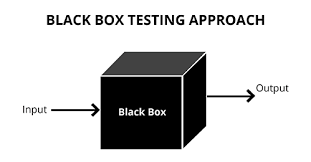
* Unit Testing
* Integration Testing
* System Testing
* Acceptance Testing

1. **White Box Testing.**



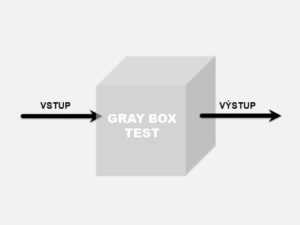
* Kiểm thử hộp trắng là việc điều tra chi tiết về logic và cấu trúc bên trong của mã. Thử nghiệm hộp trắng còn được gọi là thử nghiệm kính hoặc thử nghiệm hộp mở.
* Để thực hiện kiểm tra hộp trắng trên một ứng dụng, người kiểm tra cần biết các hoạt động bên trong của mã.

1. **Black Box Testing.**



* Kỹ thuật kiểm tra mà không có bất kỳ kiến thức nào về hoạt động bên trong của ứng dụng được gọi là kiểm thử hộp đen
* Người kiểm tra không biết gì về kiến trúc hệ thống và không có quyền truy cập vào mã nguồn

1. **Gray Box Testing.**



* Sự kết hợp giữa black box test và white box test.

1. **Software Testing Levels**
2. **Unit Testing.**

* Đơn vị là phần nhỏ nhất có thể kiểm tra được của một ứng dụng như hàm, lớp, thủ tục, giao diện. Kiểm thử đơn vị là một phương pháp mà các đơn vị mã nguồn riêng lẻ được kiểm tra để xác định xem chúng có phù hợp để sử dụng hay không.
* Các bài kiểm tra đơn vị về cơ bản được viết và thực hiện bởi các nhà phát triển phần mềm để đảm bảo rằng mã đáp ứng thiết kế và yêu cầu của nó và hoạt động như mong đợi
* Unit Test được thực hiện bằng cách sử dụng phương pháp Kiểm thử Hộp Trắng

1. **Integration Testing.**

* Kiểm thử tích hợp kiểm tra sự tích hợp hoặc giao diện giữa các thành phần, tương tác với các phần khác nhau của hệ thống như hệ điều hành, hệ thống tệp và phần cứng hoặc giao diện giữa các hệ thống
* Các hình thức test:
  + Bottom up:
  + Top down
  + Hybrid

1. **System Testing.**

* Bao gồm các thử nghiệm dựa trên rủi ro và / hoặc đặc tả yêu cầu, quy trình nghiệp vụ, trường hợp sử dụng hoặc các mô tả cấp cao khác về hành vi của hệ thống, tương tác với hệ điều hành và tài nguyên hệ thống.
* Kiểm tra hệ thống thường là kiểm tra cuối cùng để xác minh rằng hệ thống được cung cấp đáp ứng các đặc điểm kỹ thuật và mục đích của nó..
* Kiểm thử hệ thống theo kiểm thử chức năng và phi chức năng:
  + Kiểm thử chức năng:
    - Requirement-based testing: Thử nghiệm dựa trên yêu cầu
    - Business-process-based testing: Thử nghiệm dựa trên quy trình kinh doanh
  + Kiểm thử phi chức năng:
    - Sanity testing
    - Re-test
    - Regression Test
    - Security Test
    - Recovery Test
    - Usability Test
    - Performance Test
    - Load Test
    - Stress test:

1. **Acceptance Testing.**

* Sau khi kiểm tra hệ thống đã sửa chữa tất cả hoặc hầu hết các khiếm khuyết, hệ thống sẽ được giao cho người dùng hoặc khách hàng để kiểm tra chấp nhận.
* Kiểm tra chấp nhận về cơ bản được thực hiện bởi người dùng hoặc khách hàng mặc dù các bên liên quan khác cũng có thể tham gia.

Có 2 loại:

* + Alpha test
  + Beta test

1. **Test case Design Technique.**
2. **Khái niệm:**

* Trường hợp thử nghiệm là một tài liệu, trong đó có một tập hợp dữ liệu thử nghiệm, điều kiện tiên quyết, kết quả mong đợi và điều kiện hậu kỳ, được phát triển cho một kịch bản thử nghiệm cụ thể xác minh sự tuân thủ theo một yêu cầu cụ thể.
* Test Case đóng vai trò là điểm bắt đầu cho quá trình thực thi thử nghiệm và sau khi áp dụng một tập hợp các giá trị đầu vào, ứng dụng sẽ có kết quả cuối cùng và rời khỏi hệ thống tại một số điểm kết thúc hoặc còn được gọi là điều kiện hậu thực thi.

1. **Các loại kỹ thuật Testcase:**

* Specification-based techniques.
  + Boundary Value Analysis (BVA) Phân tích giá trị biên
  + Equivalence Partitioning (EP) Phân vùng tương đương (EP)
  + Decision Table Testing Kiểm tra Bảng Quyết định
  + State Transition Diagrams Sơ đồ chuyển đổi trạng thái
  + Use Case Testing Kiểm tra trường hợp sử dụng
* Experience-based techniques.
* Error Guessing (Đoán lỗi)
* Exploratory Testing (Thử nghiệm thăm dò)

1. **WebDriver Commands**

* Browser Commands
* Browser Navigation Commands
* WebElement Commands
* FindElements Commands
* Check Box & Radio buttons
* Drop Down & Multiple Selects
* WebTable Handling

1. **Tools test:**

* Postman
* Jmeter
* **Cơ hội nghề nghiệp:**
* ***Có nhu cầu tuyển dụng cao:***

Nghề nghiệp ổn định, khi theo đuổi nghề tester sẽ thường xuyên được cập nhật những công nghệ mới và được tiếp xúc với các dự án khác nhau, được học được nhiều thứ mới lạ vì thế sẽ không nhàm chán.

Đặc biệt nghề này có quy định về nghề nghiệp và cơ hội thăng tiến rõ ràng: Tester càng nhiều kinh nghiệm thì lại càng được quan tâm và chú trọng.

* ***Nghề Tester không có tuổi:***

Có thể thấy rằng đối với nghề lập trình sự nhạy bén của tuổi trẻ rất quan trọng, tuy nhiên đối với nghề Tester thì kinh nghiệm đã được tích lũy nhiều năm mới là điều quan trọng nhất.

Nếu như một người Tester giỏi tiếng anh thì lại càng có nhiều cơ hội để làm ở các công ty phần mềm lớn với các dự án outsourcing của nước ngoài với mức lương rất cao.

Nhu cầu tuyển dụng cao.

Nghề nghiệp ổn định, thường xuyên được cập nhật những công nghệ mới và tiếp xúc với những dự án khác nhau, học được nhiều thứ mới lạ nên không nhàm chán.

Có quy định về nghề nghiệp và cơ hội thăng tiến rõ ràng. Tester càng nhiều kinh nghiệm càng được quan tâm và chú trọng.

* **Mô tả vị trí việc làm về testing:**
* Nghiên cứu yêu cầu và thiết kế của dự án.
* Lập kế hoạch test và kịch bản test, chuẩn bị dữ liệu test.
* Viết test case, Test các dự án phần mềm.
* Quản lý, phân tích và theo dõi kết quả test, báo cáo kết quả test.
* Thực hiện các công việc khác theo sự phân công của Quản lý trực tiếp.
* **Các kỹ năng cần có:**
* Rèn luyện khả năng sắp xếp công việc.
* Viết báo cáo lỗi chi tiết.
* Viết test case phù hợp.
* Giao tiếp với đồng nghiệp.
* Tự đặt câu hỏi.
* **Mức lương khởi điểm dành cho Fresher.**
* Với sinh viên mới ra trường chưa có kinh nghiệm làm việc, mức lương sẽ tương đối thấp. Trung bình tiền lương sẽ dao động từ **4 – 8 triệu đồng**.
* Tuy nhiên, đó chỉ là mức khởi điểm ban đầu cho vị trí nhân viên thử việc hoặc thực tập sinh. Khi trở thành nhân viên chính thức mức lương sẽ được điều chỉnh theo quyết định của Ban giám đốc.