**­­­­­PART 1: SOFTWARE TESTING OVERVIEW**

**Chapter 1: Software Development Life Cycles and Testing**

Software Development Life Cycles (SDLC): Quy trình phát triển phần mềm

1. **Gather Requirement (thu thập yêu cầu)**

– Phân tích yêu cầu, **nghiệp vụ (business)** từ phía khách hàng

* Tìm hiểu thêm về Lĩnh vực mà mình đang làm: y tế, giáo dục, bán hàng…

Kí đóng dấu, xác nhận

1. **Analyze and Design (phân tích và thiết kế)**

* Kiến trúc – UI Xác nhận
* Kết cấu
  + Databases; (dữ liệu)
  + Design classes (thiết kế lớp)

1. **Code:**
2. **Test**

* **Đúng yêu cầu của khách hàng**
* **Đạt chất lượng**

1. **Deploy**

Khách hàng kiểm tra 🡪 Chấp nhận hay không?

**Agile(nhanh nhẹn)**

1 Sprint: 2 – 4 weeks( chyaj nước rút )

PO = Product Owner writes **User Stories** (yêu cầu của từng chức năng)

App 🡪 many features( 1 áp nhiều tính năng)

Each feature 🡪 1 or many stories(mỗi tính năng có 1 hoặc nhiều câu chuyện)

Ex: Login

U1: As a user, I can login with valid account.

Là người dùng, tôi có thể đăng nhập bằng tài khoản hợp lệ.

**Scrum Team:** Scrum Master + Team members

Product Backlog (UserStories)

Feature 1

Sprint 1 (2-4 weeks)

2

Sprint 2

Update feature 1

Sprint 3

Feature 2

|  |  |  |
| --- | --- | --- |
| **Testing** | **Water Fall** | **Agile**  **Scrum/ SAFE** |
| Documentation | Many | Little |
| Domain knowledge  Testing specific skills | Docs  A lot of Time to learn | Docs  **Team** |
| Time to test | Times: few  Time to test: Long | Times: A lot  Time to test: short |
| Regression (test lại) | few | A lot |
| Automation |  | Apply |

**Build**

6

2: F3

1

Sprint 1: F1,2

2-4 weeks

6 bước: Phân tích yêu cầu, thiết kế, code, test, deloy

4

3

F4

Test lai 123

5

**Testing Phase**

1. **Phase:** Giai đoạn
2. **Milestone: cột mốc để chuyển phases – move to another phase**

Phase 1 – M1 – Phase 2 – M2 – Phase

1. **Input Criteria**: tiêu chuẩn đầu vào
2. **Output Criteria**: tiêu chuẩn đầu ra

**Testing Phases:**

1. **Unit test –** test code - code ra: Dev

Java – Junit

VS – N Unit

100% Pass

99% + 1% đã chứng thực

Shift left testing

1. **Integration test:** Dev, tester support

Đưa ra 1 số Scenarios

1. **System test**: Tester sẽ làm
2. User Acceptance: khách hàng test chấp nhận / thuê tester để test
3. Static testing and Dynamic Testing

**Chapter 2: Software Testing Overview**

Testing

1. **QA = Quality Assurance 🡪 Process**

**Do the thing right = Validation**

ISO, CMMI,….

Quy trình làm ra sản phẩm phần mềm

Audit đội dự án có làm đúng hay không?

QA giúp kiểm tra lại những yêu cầu của khách hàng đã làm đầy đủ chưa?

1. **QC = Quality Control 🡪 Product**

**Testing**

**Do the right thing = Verification**

Kiểm tra sản phẩm có đạt chất lượng hay không?

**Thuật ngữ:**

Onshore: cùng quốc gia

**Offshore: khác quốc gia**

In-house : cùng công ty

**Outsource: khác công ty**

1. **Coverage: Độ bao phủ test =**

**Tổng số đã test / tổng số cần phải test**

**Tổng số cần phải test = liệt kê ra tất cả các trường hợp phải được test trên ứng dụng**

**Chức năng: Login**

**Testing Phases**

1. **Unitest: tổng số dòng code đã test/ trên tổng số dòng code**
2. **Integration: tổng số feature đã tích ­­hợp/tổng số feature( tính năng)­- cần tích (tính theo scenarios)**
3. **System test: tổng số trường hợp test (test case) đã test/ tổng số test cases cần test cố gắng liệt kê toàn bộ trường hợp cần phải test**
4. **User acceptance: tổng số yêu cầu (user stories) đã test / tổng số yêu cầu cần phải test**

**Testing types:**

**Glass box = White box testing = Kiểm thử hộp trắng – trong - Dev làm (Unit test + Integration) – Software Developer in Test (SDIT) hỗ trợ**

**Black box: Kiểm thử hộp đen – kiểm thử bên ngoài – tester làm (System test + User Acceptance)**

***Input đầu vào và output đầu ra***

**End user**

**Gray box: Vận dụng kiến thức về lập trình/IT để làm kiểm thử hộp đen tốt hơn.**

* **SQL**
* **HTML/SQL injection**
* **API**
* **Programing scripts = lập trình**
* **Automation**
* **Testing**
* **English**

**DevOps = Development and Operation**

**CI/CD**

CI/CD = Continuous Integration – Continuous Testing – Continuous Deployment - Continuous Delivery

**Process: Build testable = build verification**

CI/CD = Tích hợp liên tục – Thử nghiệm liên tục – Triển khai liên tục – Phân phối liên tục

Quy trình: Bản dựng có thể kiểm tra được = xác minh bản dựng

Code ->Unitest 🡪Không lỗi Merge master🡪 Tich hop -> Integration Test -> **Build Dev** 🡪 Test lại OK (smoke test) 🡪 Deploy trên môi trường test(Staging, qa) 🡪 System test (manual + auto) 🡪 OK, chon RC (Release candidate) 🡪 Production, Go Live

Code ->Unitest ◊Không lỗi Merge master◊ Tich hop -> Integration Test -> Build Dev ◊ Test lại OK (smoke test) ◊ Deploy trên môi trường test(Staging, qa) ◊ System test (manual + auto) ◊ OK, chon RC (Ứng viên phát hành) ◊ Production, Go Live

**Process nhanh thì phải áp dụng DevOps = Agile +  
‘’’’’’’’’’’’**

**;,**Tools + Skills : Xây dựng kiến trúc hạ tầng (DevOps Engineers) + AutomationTim hieu: Jenkins