\* Quy trình làm phần mềm – Solftware Engineering process/ Methodology/ Life cycle/ Model/ Method/

Công ty làm phần mềm: FPT Solfware, VINAGAME, KMS, TMA, ...

Công ty có nhu cầu làm phần mềm: ... (CUSTOMER)

FPT Solfware, KMS, TMA || CT phần mềm Nhật, Mĩ, .. || Customer

Outsourcing 🡨 Đẩy Sang làm Cust pf FPT solf

CT “Công nghệ” MOMO, Lazada, ... nuôi sẵn 1 nhóm DEV chỉ để làm phần mềm cho Cty đang nuôi họ. => Cty này dựa trên 1 vài phần mềm họ tự viết để kinh doanh

\*Chốt hạ: Làm phần mềm bắt đầu từ Customer/User, Bắt đầu từ Requirements, những tính năng cần có của phần mềm là gì ??

\* Làm phần mêm có những công việc gì?

- Requirements -> Cus and User muốn cái gì

Nhiệm vụ của DEV team: Tìm ra reqs từ KH

- Design(UI, Database, Kiến trúc(Cách bố trí chia code) - MVC)

- Implement -> Viết Code

- Validation/Testing : Kiểm thử phần mềm

- Deployment/ Mainternance: bàn giao/ nâng cấp

Cách phối hợp (Có trước có sau) 4 – 7 bước này thì gọi là:

Quy trình làm phần mềm/ vòng đời làm phần mềm – SLDC – Soft.DEV.Life Cyclel; Soft.Dev.Methodology(pp luận) Soft.Dev.Process/Model

\*5 công đoạn này có nhiều cách phối hợp vs nhau, trộn 5 bước trên

2 Trường phái lớn:

- Tradition (Waterfall, V-model, Sashimi, ...)

- Agile (Scrum, XP, Kanban, Lean, Lean Startup)

Graphical user interface, application

Description automatically generated

\*\*\*\*\* Quy trình làm phần mềm theo waterfall (Mô hình thác nước)

\*The Whole (Luôn làm trên tất cả - Hiểu hết bài toán rồi đi tiếp, DESIGN hết ms đi tiếp, CODE full toàn bộ, TEST luôn cả app)

REQS -> DESIGN ->CODE -> TEST -> Đưa cho K/H xài full ko che tính năng – Working Soft

- Mô hình chia ko đều về thời gian

\*\*\*\*\* Quy trình thực dụng (linh hoạt): AGILE (ITERATIVIE & INCREMENTAL – Lặp & Tăng trưởng dần dần)

*Ra đời năm 2000. Thời kì hoàng kim 2010.*

*2001 -> 17 người sếp cty lớn p/m, giáo sư đại học, kiến trúc sư trưởng cty p/m -> Tuyên ngôn AGILE -> Quy trình đc chính thức khai sinh*

\*Divide & Conqueure: chia để trị.

\*Tách từng modul ra để làm, có phần mềm sớm, lấy feedback để sửa

🡪 Hướng tới sự hài lòng của Cus/End-User tối thượng.

- Bên AGILE những Reqs được gọi là USER STORY

Business Analyst (BA): Nhân viên phân tích

Bridge Engineer (BR): Kĩ sư cầu nối

Customer Supplier

Fap ---------------------------------------------------🡪 Dev

PDT, GV, ... BA(Dev)

Mẹo làm việc K/H:

1. Xác định xem App viết cho ai dùng (User Types/ Roles/ Classify).

VD1: Fap ai dùng?

- Student.

- Lecture.

- Academic Staff.

- Parent

VD2: Phần mềm bán hàng (PMBH)

- Thu ngân/ Cashier

- Sales

-...

VD3: Quản lý thư viện

- Librian/ Bookeeper

- Reader (Tùy)

2. Vơi mỗi người dùng/ Role, Liệt kê ra những tính năng (Màn hình = Ô nhập + nút nhấn + xử lý đi kèm -> ra đc 1 K/quả mà người dùng mong đợi/ kq họ phải làm hàng ngày khi chưa có PM). What, Requirement, Goal, Target, Objective, User Story.

->>>>>>> Các tính năng (màn hình) là những thứ mà user phải làm hàng ngày khi chưa có phần mềm hoặc p/mềm cũ.

VD1: Phần mềm quản lý bán hàng

-> Cashier (Nhân viên thu ngân)

- Create an order

- Scan khách hàng thân thiết để tích lũy điểm mua hàng

-> Sale (Admin) Adminustrator (Quản trị - người giữ, quản lý thông tin phía sau app khiến cho thông tin luôn sẵn sàng cho mọi role khác)

- Add a new food Item.

- Remove a food item.

- Update a food item (price, avatar, description).

- View report (revenue)

\*\*\*\*\*Cách viết REQ như ở trên: Câu đi kèm 2 thứ: User Role - What (ai làm đc j với app, app cung cấp tính năng gì cho ai?) 🡪 Viết REQS kiểu truyền thống – Water fall

\*\*\*\*\* Agile/Scrum...Viết theo Template(“MÀU MÈ”)

AS A... I WANT TO ... SO THAT...

AS AS CASHIER, I WANT TO CREATE AN ORDER, SO THAT I CAN RECORD WHAT OUR CUSTOMER BROUGHT.

AS AS BOOKEEPER, I WANT TO ADD NEW BOOK, SO THAT IT IS AVAILABLE TO ANY BORROWER. (Đây là User Story)

Source PE: Cho 1 bài toán, tình huống KH đang cần có app nào đó để giúp cho c/v của họ:

- Quản lý xyz gì đó.

- Quản lý học vụ/đào tạo của 1 trường đh, cao đẳng.

- Quản lý H/động của 1 bệnh viện.

- Quản lý H/động của 1 công ty du lịch.

Đề nghị KH mô tả họ đang làm việc như nào:

Mô tả cách trường, BV, cty, vận hành khi chưa có app/ có app cũ cần nâng cấp

Nhặt ra trong bài toán, câu truyện của KH cái REQS – User Story – Functional Requirements.

VD: Text

Description automatically generated

\*\* Trong các REQS từ khách hàng, do ta thu lượm đc, có 1 loại REQ đc gọi là: Non- Functional Requirements – không viết theo template mà viết tự do.

Non- Functional Requirements: UX, trải nghiệm người dùng, nhanh chậm tốt , xấu ... Tính từ chất lượng của tính năng.

VD: App chạy nhanh, phản hồi người dùng trong vòng 3s đổ lại.

App chọn tông màu hường nam tính.

Với danh sách REQS đã cho, và còn nhiều hơn nữa

Với mô tả bài toán đã cho.

Với đặc trưng của dev team: Giỏi công nghệ, chưa quen công nghê.

\*\*\* Đầu bài yêu cầu bạn chọn phương pháp phát triển phần mềm nào tùy ý, nhưng phải giải thích tại sao??? 7 điểm

\*\*3 điểm còn lại liên quan đến cách ta tổ chức các user Story, cách ta cài đặt chúng và bàn giao chúng cho KH

Việc quản lý tiến độ/ phân công công việc, danh sách các User Stories, từng chặng sprint (1-4w) làm những vc j có thể dùng.

**Tuyên ngôn về Phát triển Phần mềm Agile**

Chúng tôi đang khám phá ra những cách tốt hơn để phát triển  
phần mềm bằng cách thực hiện nó và giúp những người khác làm điều đó.  
Thông qua công việc này, chúng tôi đã nhận ra giá trị:

Các cá nhân và tương tác qua các quy trình và công cụ  
Làm việc phần mềm trên tài liệu toàn diện  
Sự hợp tác của khách hàng trong quá trình đàm phán hợp đồng  
Đáp ứng sự thay đổi so với việc tuân theo một kế hoạch

Có nghĩa là, trong khi có giá trị trong các mục ở  
bên phải, chúng tôi đánh giá các mục ở bên trái nhiều hơn.

**Manifesto for Agile Software Development**

We are uncovering better ways of developing  
software by doing it and helping others do it.  
Through this work we have come to value:

Individuals and interactions over processes and tools  
Working software over comprehensive documentation  
Customer collaboration over contract negotiation  
Responding to change over following a plan

That is, while there is value in the items on  
the right, we value the items on the left more.