SWR302  
Software Requirement

* SRS: software requirement Specofocation (bản đặc tả các yêu cầu của phần mềm) ~ tên các màn hình, tên các chứng năng, CRUD item, list of fuctional, Reqs, Non-funct reqs. Gồm các project capstone

Quy trình làm phần mềm/ SoftwareDevelopment process/ Model/ Method

1. Requirements Gathering – tìm ra các yêu cầu phần mềm (liệt kê các tính năng) – Chuyên viên đi khách (IT BA: Bussiness analyst/ BrSE: Bridge S.E)

Vd: app có màn hình login = Gmail

App có màn hình/ tính năng thêm/tạo mới đơn hàng, thanh toán online

App có tính năng theo dõi tiến trình xử lý đơn hàng (giống giao hàng shoppe)

-> SRS – software Requirements SPECIFICATION (góc nhìn nhà F thì là ½ cuốn KLTN)

-> SWR302

2. Design – thiết kế

UI (Front-end Design), FIGMA

Database (ERD)

Software Architectural design, kiến trúc của phần mềm: MVC, Micro Services, API, …

>>>SWD391 kì 7 mức lương từ 4000$/ tháng ở VN.

3. Implementation / Development/ Viết code

4. Validation/Testing/QC (QC Tester)

5. Go live/ Production/ Deployment/ Enhancement/ Upgrading/ Fixing/ Pach/ Update

Bàn giao sản phẩm, đưa vào sử dụng, nâng cấp, fix bug, thêm mới tính năng

1 -> 2 -> 3 -> 4 -> 5: Tuyến tính và tuần tự, xong gọn từng phần công việc -> TRADITIONAL/WATERFALL

Nếu đi cùng lược -> Agile/ Incremental & Iterative

>>> Scrum, KanBan, XP, LEAN (START-UP)

Why Requirements Engineering (kỹ thuật)?

* Bức tranh 1: sự khác ngôn ngữ của con người khiến product mãi ko thành
* Bức tranh 2: tam sao thất bổn hiểu sai bussiness muốn gì dẫn đến sai xót
* Bức tranh 3: bên cung cấp dịch vụ và bên yêu cầu
* App thu nhập cá nhân: nếu đang độc thân -> mỗi cá nhân mỗi tháng trung bình xài 11tr nếu làm lương cao qua thì phần dư sẽ bị tính thuế

RequirementEngineering: (kỹ thuật tìm ra các yêu cầu phần mềm, tại sao phải có Requirements)

* I. Ví dụ về SotfWare Requirement
  + A statement, một câu phát biểu cửa customer hoặc của User về tính năng/tên màn hình của phần mềm họ cần (App phải làm cho họ 1 điều gì đó)
  + Ví dụ:
    - app cho phép tôi add một công thức làm bánh.
    - App có tính năng thêm mới một sản phẩm nước yến
    - Add thêm gói thuê xe
* II. Why REQS Engineering? Tại sao phải học các kỹ thuật lấy yêu cầu phần mềm
  + DEV team
    - Tam sao thất bổn, có thể hiểu sai ý nha
  + Customer/ User
    - Không diễn đạt chính xác nhu cầu
    - Emerging – Phát sinh thêm, nói thiếu
    - Khách hàng thay đổi yêu cầu
    - Complict: các REQS tương phản, mâu thuẩn nhau, bất hợp lý
    - Dư thừa tính năng: Làm theo yêu cầu của khách dù nó có dư thừa thì không sao
  + Phí tổn phải trả cao khi REQs trực trặc
    - 56% REQs thì tốn 82% sửa
* Nồi cơm điện kim cương: web điện tử nhưng lại có mục giải trí
  + Reqs: khách hàng muốn có mục giải trí và muốn thì có
* III. App và nơi tim ra chúng, noi xuất hiện ra app, app xuất hiện như thế nào? -> do dev team viết, dev team ở đâu?
  + Công ty chuyên làm phần mềm nhưng gọi là Product-Based
    - Đặc trưng của công ty
      * các sản phẩm là phần mềm được viết cho đám đông User
      * Phần mềm đa dụng, phổ thông -> Generic App
      * Công ty phần mềm tự đi khảo sát thị trường, tự viết ra và hy vọng bán được sản phẩm
        + BA (Business Analyst): đi nghiên cứu thị trường
      * Tự bỏ tiền túi và tự đi bán -> lỗ hoặc lãi
    - Nhánh con Start-up:
      * Làm 1 sản phẩm chưa từng có trên thị trường hoặc thị trưởng sở tại
      * Phải khảo sát thị trường, tìm cái gì là điều chưa ai làm
      * “PACE” học viện đào tạo doanh nhân, start-up thì phải hoc trước
      * Câu truyện:
        + 4 anh chàng lập trình viên muốn làm giàu cho chính mình nghĩ về cắt tóc -> Shine30 ra đời -> có 1 app quản lý
        + Cũng anh khác thấy sửa máy lạnh phiền quá -> gọi thợ nhưng éo biết thợ nào -> app gọi thợ sửa máy lạnh ra đời -> app thới giới thợ
    - Vidu:
      * microsoft, windows, AZURE, SQL Sever
      * EASport: FIFA, Nianticlots + Nintendo: Pokemon
      * Tên công ty +???: gà rán
      * Misa: App quản lý hoạt động của doanh nghiệp
      * Adobe
  + Công ty chuyên làm phần mềm nhưng gọi là Sevice-Based
    - Đặc sản của công ty này
      * Customized/Bespoke app
      * Công ty phần mềm làm app theo đơn đặt hàng của cá nhân, doanh nghiệp, tổ chức
      * Ví dụ:
        + Fsoft: App: quản lí ngân hàng TPBank, DeustchBank, QLBV chợ Rẫy An Sinh, Siêu thị, bộ ngành, …
        + ELCA, NashTech, Robert Bosch, TMA, LogiGear
      * Ai đó đưa $, đặt hàng làm app
      * Không lo bán app, ế app
      * App được viết theo nhu cầu riêng biệt của từng đơn bị khavhs hàng ví dụ như ngân hàng, quán ăn, khách sạn, công ty du lịch, shop, công ty sản xuất, …
      * Có tiền nếu có đặt hàng
    - OUT-SOURING: Gia công pần mềm
      * công ty phần mềm 🡨-------BA--------🡪 một công ty phần mền (trong/ngoài nước) 🡨------BA---------🡪 Bệnh viện, ngân hàng, Cty sản xuất, …
      * nói chuyên dân IT với IT nói chuyện dễ với nhau nhưng chém gió về công nghệ đc
      * IF số 2 là Japan thì BA cở công ty phần mềm 1 sẽ đc gọi là BrSE (Bridge Software Engineer – kỹ sư cầu nối) yêu cầu phải biết code để làm vc đc hết với bên nhật luôn
  + App đến từ dev team của một công ty ko phải công ty IT/ phần mềm
    - Ví dụ:
      * Grab, Uber (Chết rồi), Gojek, Be, …
      * GHN, GHTK, Ahamove
      * Beamin, ShoppeFood, GrabFood, …
      * MoMo, MoCa, ZaloPay
      * Bia Sài Gòn, Hoàng Anh Gia Lai, NovaLand, TH true Milk
      * ACB, VCB, TPBank
      * Shoppe, Lazada, Sendo, Tiki
      * TGDD
    - Họ bán một món đồ nào đó nhưng họ cần một phần mềm xịn sò nào để bán
    - Các công ty này đều tuyển nhân viên lập trình
    - Có sự xuất hiện của PO, làm việc với các user là các phòng ban
    - Đặc trưng của app:
      * App dùng cho nội bộ
      * App dùng cho mô hình kinh doanh
      * Có thể xếp vào app Generic
    - Inhouse dev sẽ bảo trỉ, nâng cấp tính năng
* IV. Phân loại một số khái niệm dễ gây hiểu nhầm
  + As IS SYSTEM, To Be SYSYEM
    - As is system/Legacy system: hiện trạng hiện nay, khách hàng hiện nay đang sài app gì trong công việc của họ. Hoặc đang làm bằng tay mọi thứ như thế nào
      * ATM truyền thống là một ví dụ (chủ yếu là rút tiền xem số dư)
      * Thẻ tích điểm khách hàng một số nơi dùng giấy và mộc
    - To Be SYSTEM: App sẽ làm để fix các problems của as if/Legacy, ...
  + Product vs project
    - Product:
      * App được làm xong và đem xài
      * Nói về kết quả và đích đến
      * Nói về các tính nang/ chức năng
      * Ví dụ:
        + AppBrowser:

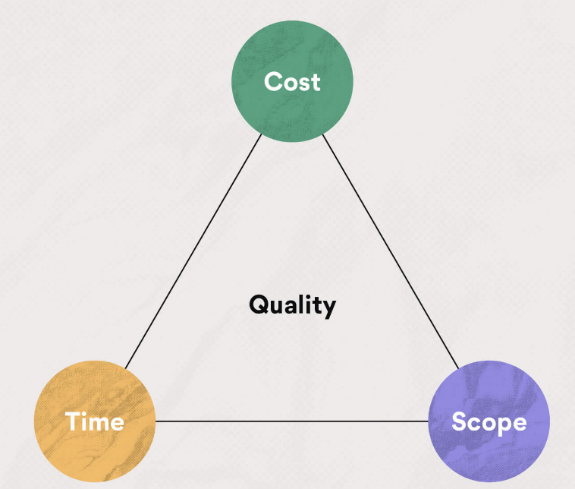
chức năng duyệt web

Bookmark

Clear history

Duyệt ẩn danh

Download(resume)

* + - * Quản lý, tìm ra các tính năng này đó là BA/BrSE(làm theo yêu cầu), PO (Product Owner)/Project manager(sáng tạo)
    - Project:
      * Nói về hành trình làm ra app
      * Những công việc cần làm để ra được cái App/Product
      * Kết thúc một dự án ta được một product
      * Người quản lý để tạo ra app/product là PM (Project Manager): GOLDEN TRIANGLE OF PROJECT MANAGEMENT
        + Time
        + Cost
        + Scope
      * Có ScrumMaster nữa: cần 2 cái chứng chỉ lv1 và lv2 để làm sếp quản lý
  + Project Manager, ProductManager, Product Owner (đã note phần trên)
  + Business Analyst, Bridge Software Engineer (đã note phần trên)
  + Product Requirements, Project Requirement
    - Đều là những câu phát biểu (Staments) yêu cầu ai đó phải làm gì về điều gì đó
    - Product Reqs: MÔN NÀY TẬP TRUNG VÀO CÁI NÀY
      * Những câu phát biểu liên quan đến tính năng/màn hình/xử lý mà app cần phải có
      * Mô tả, liệt kê tính năng của app
      * Ví dụ:
        + “Admin tính năng CRUD users”
        + “tạo mới đơn hàng thì nhập sản phẩm theo ba cách: mã vạch, QR, gõ số”
    - Project Reqs:
      * Những câu phát biểu liên quan đến quá trình làm app, quá trình quản lý dự án
      * Liên quan đến cái tam giác trên hình trên
      * Ví dụ:
        + “Ngày mai xong phần admin để test”
        + “Tối này cả team tăng ca”
        + “Cần mua ngay cái máy để thử tính năng”
* V. SURE OF R/E (BA, BrSE, PO, Product Manager ) - SCOPE
  + WHY? 🡪 Goc nhìn Benefit -> BRD
    - CÂU HỎI?
      * Tại sao ta lại làm cái app này?
      * Ví dụ
        + Tại sao lại phải làm cái app FiFa
        + Tại sao lại làm app BV. Chợ Rẫy?
    - CÂU TRẢ LỜI:
      * Tiền
    - App GERNERGIC:
      * Công ty bỏ tiền túi ra làm app và hy vọng bán được nhiều
      * 🡪LÀM APP ĐỂ KIẾM TIỀN!!! từ đám đông
      * Người đặt ra câu hỏi why chính là sếp công ty, Founder
        + Muốn lấy tiền từ bá tánh thì phải đem lại Benefits cho bá tánh
    - App Customized/Bespoke
      * Ai đóa đưa tiền cho công ty phần mềm để nhận về một cái app
      * Người trả lời câu hỏi WHY là khách hàng
  + WHAT? \*\*\* TRỌNG TÂM CỦA MÔN HỌC
    - App có tín năng, chức năng gì? 🡪 Software Reqs <-> Product Reqs
    - Phân loại Requirements (Những câu phát biểu về 1 điều gì đó phải làm/ liên quan đến tính năng của app)
      * Business Reqs:
        + Cuốn sách BRD
      * User Reqs
        + SRS
        + User Story

BackLog

* + - * System Reqs
        + Ghi trong FRS/FRD/SRS
        + Functional reqs
        + Non-Functional reqs
      * …
  + WHO?
    - Ai sẽ xài app?
    - Thiết bị nào, app nào khác cùng tham gia vào việc vận app sẽ viết/sẽ làm
* VI: STAKE HOLDER:
  + Những người bên liên quan đến việc làm dự án, làm app, to be system
    - Development team
      * PM
      * Scrum Master’
      * Designer
      * Developer
      * QA/QC
    - Nhóm khác
      * BA – business Analyst
      * BrSE
      * PO
      * Product Manager
    - Customer
      * Ai đó đưa ra yêu cầu về tính nân p/ mềm mà họ cần và dev phải implement
      * Generic app
        + Công ty phần mềm làm cho bá tánh

Survey tìm strend, needs

* + - * Customize app
        + Nhận yêu cầu từ một ai đó cụ thể:

Nhận công việc từ BV, nhà hàng, quán ăn, …

* + - * + Công ty có nuôi nhóm dev để làm app:

Các phòng ban khác

* + - User: người xài app
      * App lưu data gì? info cần lưu! Không care Table/ERD
      * Đừng bắt các sếp làm admiin là người đưẹoc mướn để CRUD data. ADMIN có quyền sinh sát trên data. Không cần gắn vái account tên là admin
      * Direct user:
        + FAP: Student, Parents, Lecturer
        + Là những ai mở app lên, nhập data input và nhấn nút chờ kết quả -> dùng cho công việc của họ
      * Indirect user:
        + Người không cần xài app mà vẫn cần lấy từ thông tin từ app mà không phải là hacker
        + Không dùng màn hình app mà nhờ direct lấy giúp
      * Ví dụ
        + Ba má dẫn mình đi mua con xe, ba má dùng
        + Ba má dẫn mình đi mua con xe, cho minh dùng
        + -> Tiệm xe là nơi cung cấp dịch vụ -> Ba má xuống tiền: Customer -> là người dùng
    - UserPersonal
      * Chân dung người dung -> chùng cho app generic
      * Thuật ngữ của marketing, phòng phát triển sản phẩm
      * Mọt bộ hồ sơ giả
      * Nói về một user nào đó đai diện chọ một tập user ngoài đơeì thật
      * Đi khảo sát ngoài thị trường
      * Generic:
        + Bá tánh x 3,14
        + Chia nhóm user theo đặc trưng nào đó
        + Chia phân khúc thành tường nhóm
        + Chọn một nhóm và hướng đến
      * Ngồi liệt kê ra các cá tính và hô sơ của một người. giúp dễ dàng nói chuyện trong một team
      * -> phân tích thị trường
  + BA:
    - Thuật ngữ
      * BA, BMA, BA
      * BA (Business Analyst): Chuyên viên phân tích nghiệp vụ khách hàng
    - Task:
      * Tìm ra user role, ai xài app
      * Tìm ra Reqs, chức năng màn hình của app
      * Viết document BRD, SRS, FRS, FRD
      * Quản lý sự thay đổi của Reqs: Confluence (Jira)
    - Skills: Kỹ năng cần có của BA
      * Softkills chủ đạo
      * Kỹ năng đặt câu hỏi/ phỏng vấn k/hàng và lắng nghe
      * Phân tích, đánh giá vấn đề
        + Ví dụ:

App thu ngân

2 máy barcode: một cố định, một cầm tay

Đồ nặng ký -> nhấn barcode

Không touch được vì 10k món

App trà sữa:

TouchScreen

Tùy bài toán nghiệp vụ (cách user làm việc các phương án đè xuất REQS)

Conveience Store:

Barcode

* + - * System Thinking
        + Tư duy hệ thống
        + App to be phải đưẹoc nhìn trong tổng thể hejej thống
        + App mình viết có giao tiếp thích hợp với app khác hay không ví dụ: ví, chay, qlys kho, chăm sóc KH
      * OBSERVATION: kỹ năng quan sát
        + Xem các app khác trên thi trường
        + Xem, chú y đến khách hàng làm việc
        + Tìm học, clone Reqws
        + App BH ở quán nhậu
      * Facilitation Organization: điều phối vf tổ chức
        + Làm việc với n User cùng lúc -> t chức phối hợp
      * Modeling: Mô hình hóa
        + Vẽ, hiểu được luồng xử lý c/việc của KH

ss

OJT Sharing

Github/facebook: tienhuynh-tn

1. Prepare

* CV
* Github
* Kỹ năng mềm
* Kiến thức

2. Interview

* HR interview
* Technical Interview
  + bài test
  + phỏng vấn trực tiếp về kiến thức
* Director interview
  + Có thể gọp với phần trên
  + Xử lý tình huống
* Offering
  + Lương, thời gian làm việc hoặc kết quả

3. OJT tại FPT Software

PHỎNG VẤN:

* 4 Nguyên lí của OPP, giải thích từng cái
  + Trừu tương/ Abstraction
  + Đóng gói/ Encapsulation
  + Kế thừa/ Inheritance
    - Reuse nhưng ko phải copy and paste
  + Đa hình/ Polymorphism
* Tính đa hình được hiện thực hóa như thế nào
* Phân biệt giữa Abstract class và Interface
* Phân biệt OverRide và OverLoad
  + OverLoad có khả năng xảy ra trong bất kỳ class nào khi có các hàm trùng tên khác phần tham số
  + OverRide là thằng con độ lại hàm của thằng cha (tính đa hình)
* Access Modifier/Specifier trong Encapsulation là gì
  + Gồm 4 cái
* 5 nguyên lý thuyết kế nâng cao: SOLID (tham khảo tôi đi code dạo)
  + High Cohesion Loose coupling
  + S: Single Reponsbility
  + D: Dependency Injection
* Database: ERD là cái gì? PK, FR? JOIN (Dạng chuẩn NF – Nomalization Form)?
* PHÂN BIỆT THƯ VIÊN VỚI FAMEWORK:
  + CODE nào ép viết đúng cú pháp thì chính là famework
  + Thư Viện chỉ gọi ra, thoải mái