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
      * 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
  + App đến từ dev team của một công ty ko phải công ty IT/ phần mềm

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)?