Chapter 2

Software Development Life cycle

(SDLC)

-SDLC

-Software Documentation

-Stakeholder

Lecture 2 - Software Development Life Cycle (SDLC)

- Core process of software development
- Activities to perform

Some basic problem of SE

Ad-hoc Software development

Creating software without any formal guidelines or process

it will cause

- important actions (test, design) may go ignored
- not clear when start or stop doing each task
- does not scale well to multiple people
- not easy to review or evaluate one's work

Then it lead to SDLC that is clear step, tangible doc
allow for view of work, specify action

Analysis → Design → Dev → Test → Deploy → Maintain → Wrap up

Requirement Analysis and Specification: what the sys should do

Design - define software structure that realises the specification

Develop, Intergated - coding / combining code for the system

Test - checking that it work correctly and does what the customer wa

Deploy - rolling the system out to the user

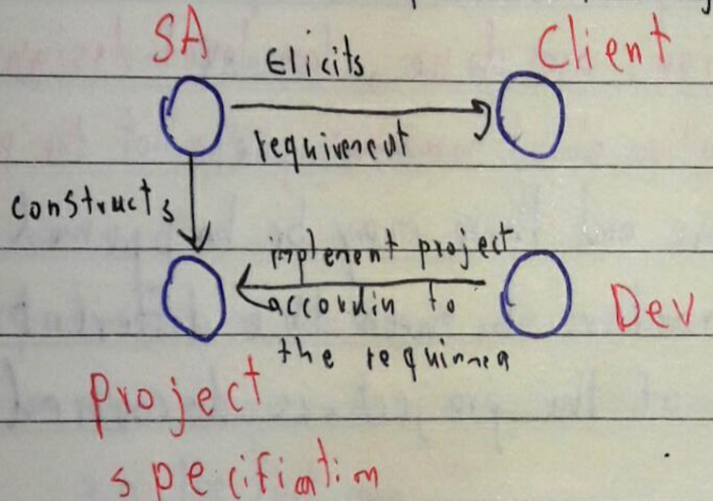
Maintain - changing the system in responce to chang"

Wrap up - evaluating

" can take mouths or years to complete

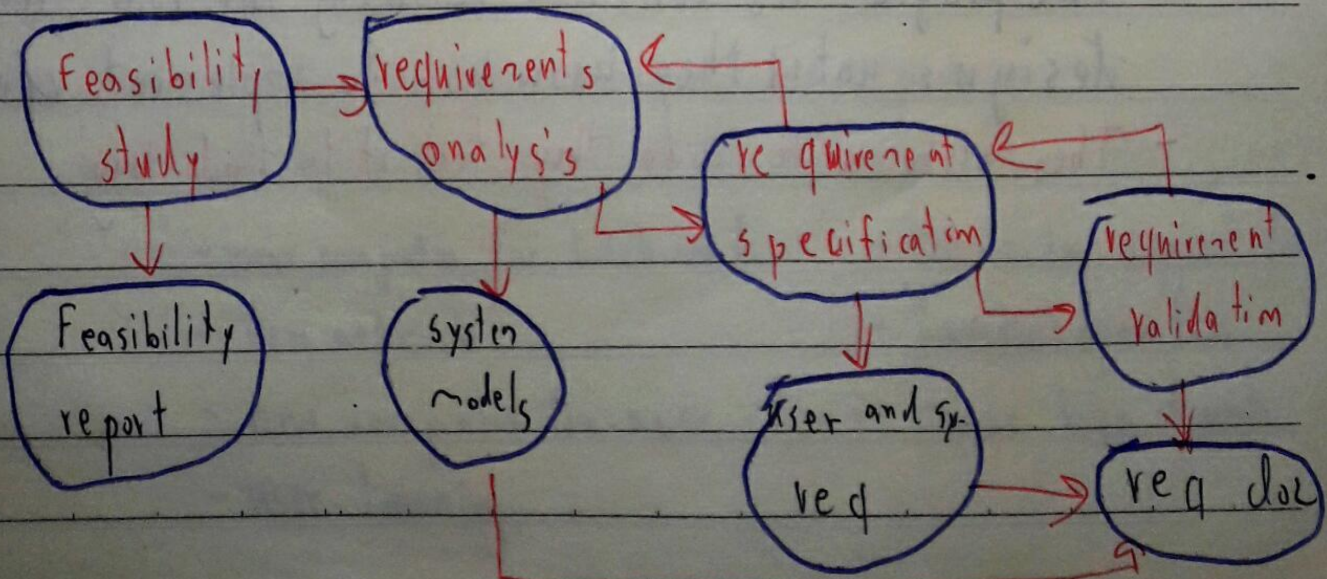
Requirement Analysis and Specification

- To solve problems by understanding
  - Customer's problems
  - Business environment
  - Avaible technologies
- The process of establishing what poducts and service are required and the constraint on the sys operation and dev
- key: identify customers and try to interact with then as much as possible to insight what the customers want/need



changes often happen

The requirements engineering process



Designing pharse

High level Design

- Deciding how the requirements should be structured
    - what platform to use
    - what data design to use
    - Interfaces with other systems

"make sure that the high-level design covers every aspect of the requirements"

Low level Design

- After high-level designs are done, low level designs begin
- it includes information about how that piece of the project should wo

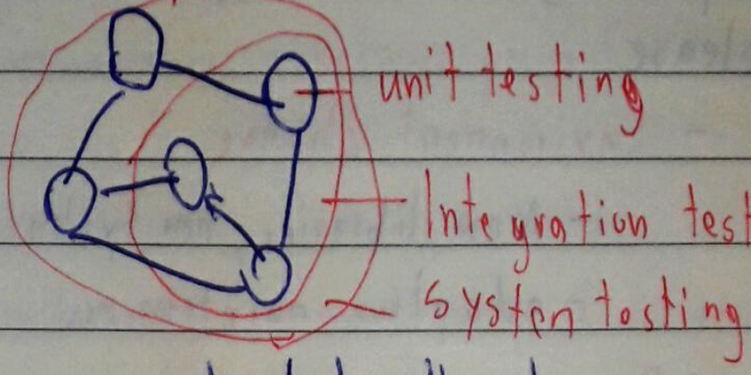
"Changing the design here and there may be happened since the interactions between the different pieces of the project is discovered"

Software development

- Translate design structure into an executable program;
- The programmers continue refining the low-level designs until they know how to implement design code
- Then after write code they test it to find bug

Software Testing

- checking the system does work correctly and does what customer want
- Systems are tested by tester who didn't write the code



Fixing a bug can lead to other bugs

- may be incorrect bug
- break some code
- change some behavior

"Every time you make a decision, the next task in the sequence include two more decision that depend on the first one."

Software deployment

- all of the activities that make a software system available for use.
- can be difficult, time consuming, and expensive.
  - might involve any or all of the following
    - new computer for back-end
    - new network
    - new computer for user
    - user training
    - On-site support for user
    - Parallel operation
    - more bug found

Software maintenance

- Software is inherently flexible and can change.
- Many things can be happened after the products release
  - Environment change
    - New libraries, new Systen
    - adaptive maintenanc
  - Requirements change
    - ~~bug~~ Feature requireme
    - Perfective maintenance
  - Bug report
    - problems with the software are found
    - corrective maintenance

Wrap Up

- At the end of project, team should evaluate the proj-
  - Figuring out how to make the thing that went well occur more often in the futur
  - Detemining how to prevent the thing that went badly in the Future

SDLC in nature

- dev model handle basic tasks - work sometime flow back ward
- basic task often occur at the same time
- longer bug remains the harder it is to fix

Software Documentation And Document Management System

Software dev include

- Documents: need for software developement
- People: involve in a software project

Software documentation

- document will produce inevery step of software process
- it is important at every step
- the main goal is to ensure that dev and stakeholder are understant the same goal
- Example in some software project
  - requirements
  - Use cases
  - design
  - test plans
  - User training material
  - etc
- Can be ≥100 or ≥1000
- Can has many version
- the most recent version of the requirements will be use
  - the most recent high-level and low-level design is use
  - the older version will collect as refferenct

Document management system (DMS)

- use for share current doc to the team
- only one person can edit doc at given tim
- can fetch recent version
- can fetch specific version
- can search for tag, keyword, or anything
- can compare two version of a document

example

ascensio system, sharepoint, Rubex, Ducuware

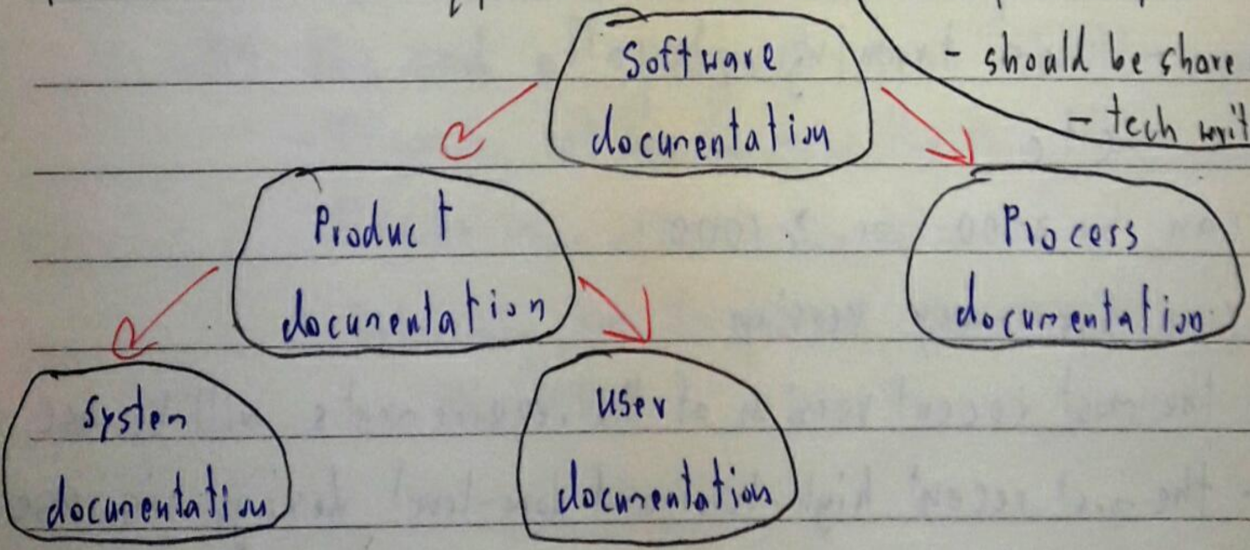
Program source code

- different from document
- share the code using version contrll
- it usually use "git hub"

key for write doc

- write just enough
- keep doc up-to-date
- should be shared and know fb
  - tech writer (take care doc)

Documentation types



We will talking in detail next page →

Software documentation

- process doc
  - Represents all doc produced during dev also maintenace that describe process
  - plans, test schedules, reports, meeting note, busn correspon
- production doc
  - describes the product that is being dev (system doc) and provide instruction on how to perform varions tasks with it (User doc)
- System doc
  - Represents doc that describe the system in each part
  - requirement, design, architecture, code, plan, guides
- User doc
  - Covers manuals that are mainly prepared for end-users of the product and sys admin
  - Tutorial, user guides, troubleshooting, installation

Why do we need Documentation

1. for communication
2. to perform all program management
3. describe to user how to operate
4. it will be use by maintenance engineers

Stakeholders of Soft ware project

- stakeholders is a
  - A persun
  - Group
  - Organizatin

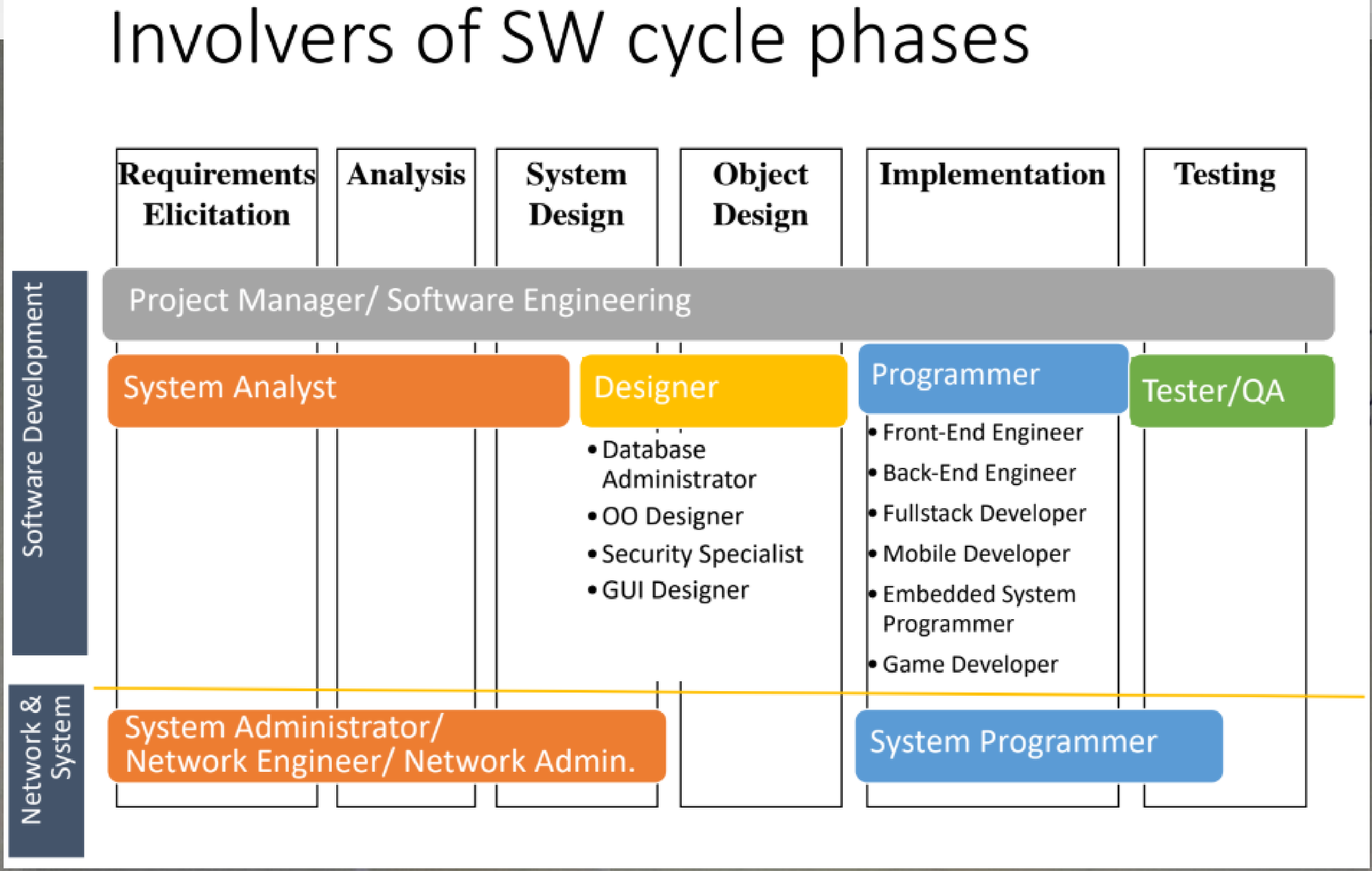
  } that is actively involved in a project

Stakeholders Categories

- Those who are involved in the project and work on it
  - project team
  - Management team
  - Third party con
  - Support team
- Those who are affected by the project
  - Customers
  - Head / employees of functional unit
  - End user
- Those who are not involved in the project but not work on it
  - Top manager
  - Owener of comp
  - Share holder / creditors
  - Regulatory structures

To success the project

- require good hard skill
- require the best soft skill



Involvers of SW cycle phases

Requirements Elicitation | Analysis | System Design | Object Design | Implementation | Testing

Software Development

Project Manager/ Software Engineering

System Analyst

Designer

- Database Administrator
- OO Designer
- Security Specialist
- GUI Designer

Programmer

- Front-End Engineer
- Back-End Engineer
- Fullstack Developer
- Mobile Developer
- Embedded System Programmer
- Game Developer

Tester/QA

Network & System

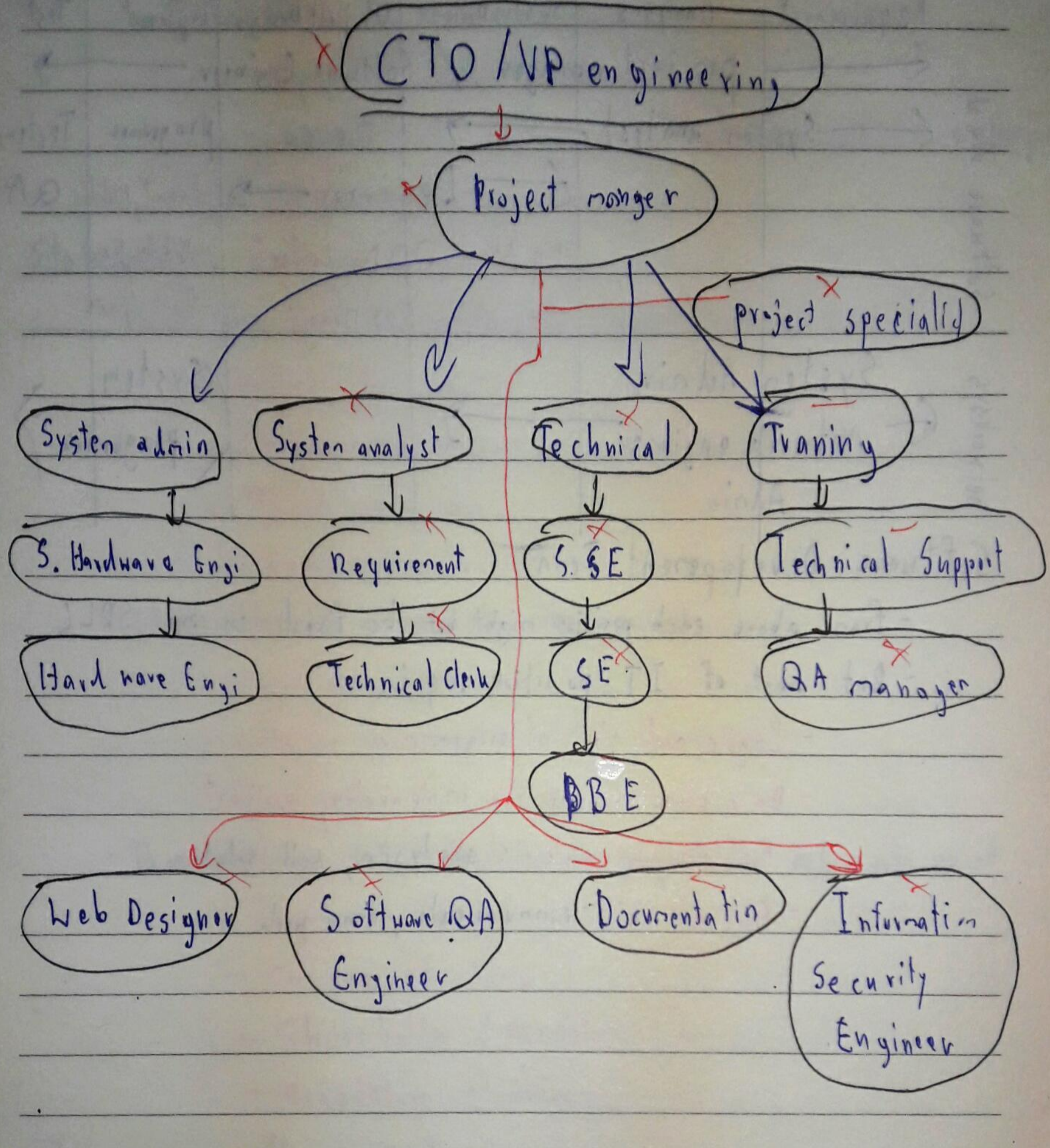
System Administrator/ Network Engineer/ Network Admin.

System Programmer

Software Development Team

- from above each position might not be fixed in one SDLC
- But most of IT position require
  - Logic and skill of programming
  - Data structure of the implementing project
  - Problem solving skill: analysis, self solution
  - Social skill: communicate, team work

Example of project Team Development.


CTO / VP engineering
Project manger
project specialist
Systen admin
Systen analyst
Technical
Traning
S. Hardware Engi
Requirenent
S. SE
Technical Support
Hard ware Engi
Technical Clerk
SE
QA manager
DBE
Web Designer
Software QA Engineer
Documentatin
Information Security Engineer