Chapter 9

Software development process model

- Software process model

- Predictive model

- Adaptive model

- Agile process model

- Scrum framework

Software Process Model

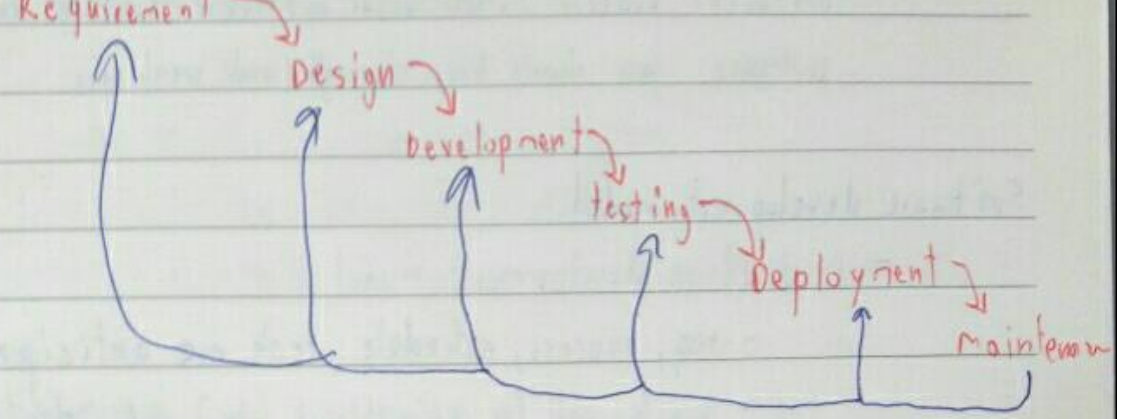
- an abstract representation of software process
- Different process model is for different software development purpose you should know strength and weakness

Software development model

- Predictive development model
  - req, process, schedule, tech are anticipated
  - Work well for project that clear and Identify
  - use when client needs clarity on target delivery
  - Waterfall model, Incremental, V model
- adaptive development model
  - give opportunities to change direction
  - Work well with changing or uncertain req
  - Use when goal not clear
  - Use when project calls for experimental design
  - Prototypes, Spiral, Agile
- Code-and-fix model
  - code and get immediate feed back
  - fix

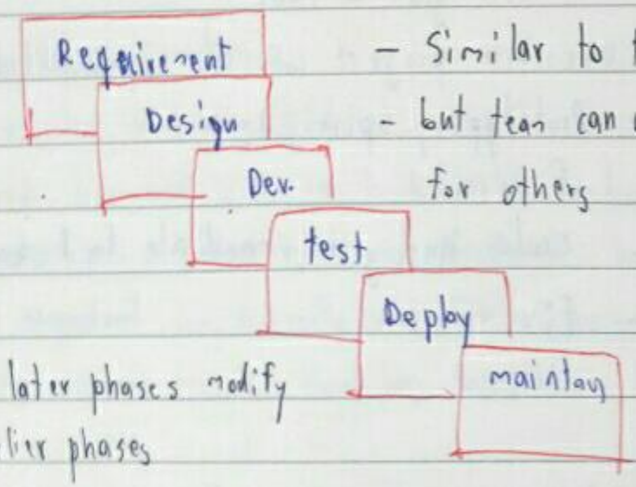
Predictive model

Water fall model



- use in government project and quanlity control project
- each phase requires fornal review and intensive doc
- * costly work

Water fall with overlapping phase

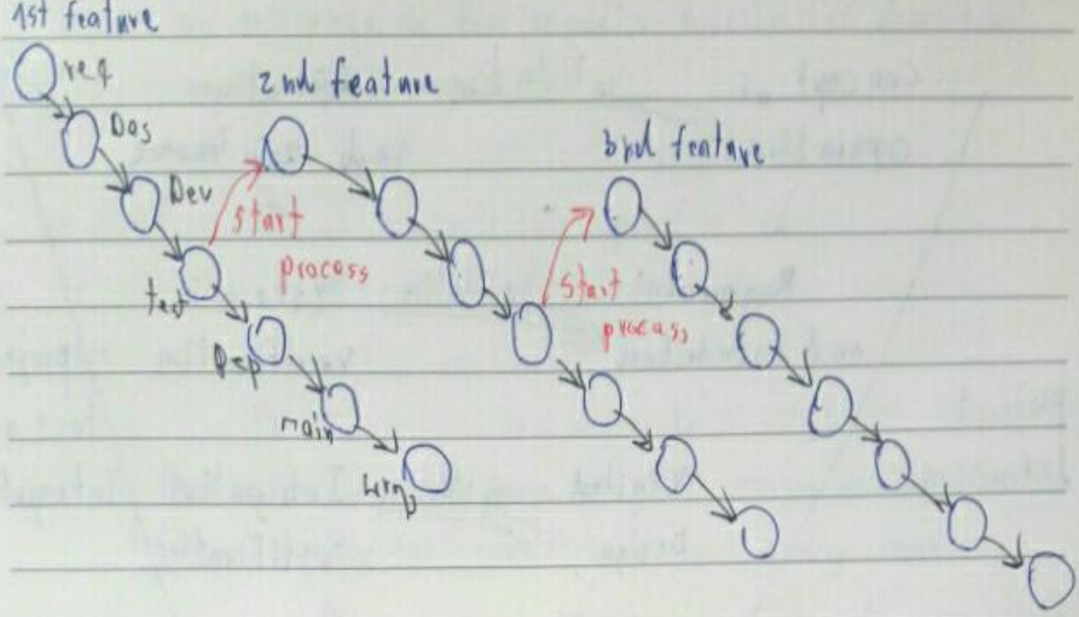


- Similar to the water fall model
- but tean can work without waitng for others

- allow later phases modify earlier phases

- progess is more difficult to track

Incremental Waterfall model



Model is designed, inplemented and tested incrementally until product is finished

advantage

- Working software is quickly produced early
- model more fexible as change to scope
- easier to test and debug
- Customer can respond to each increment
- error easy to identified

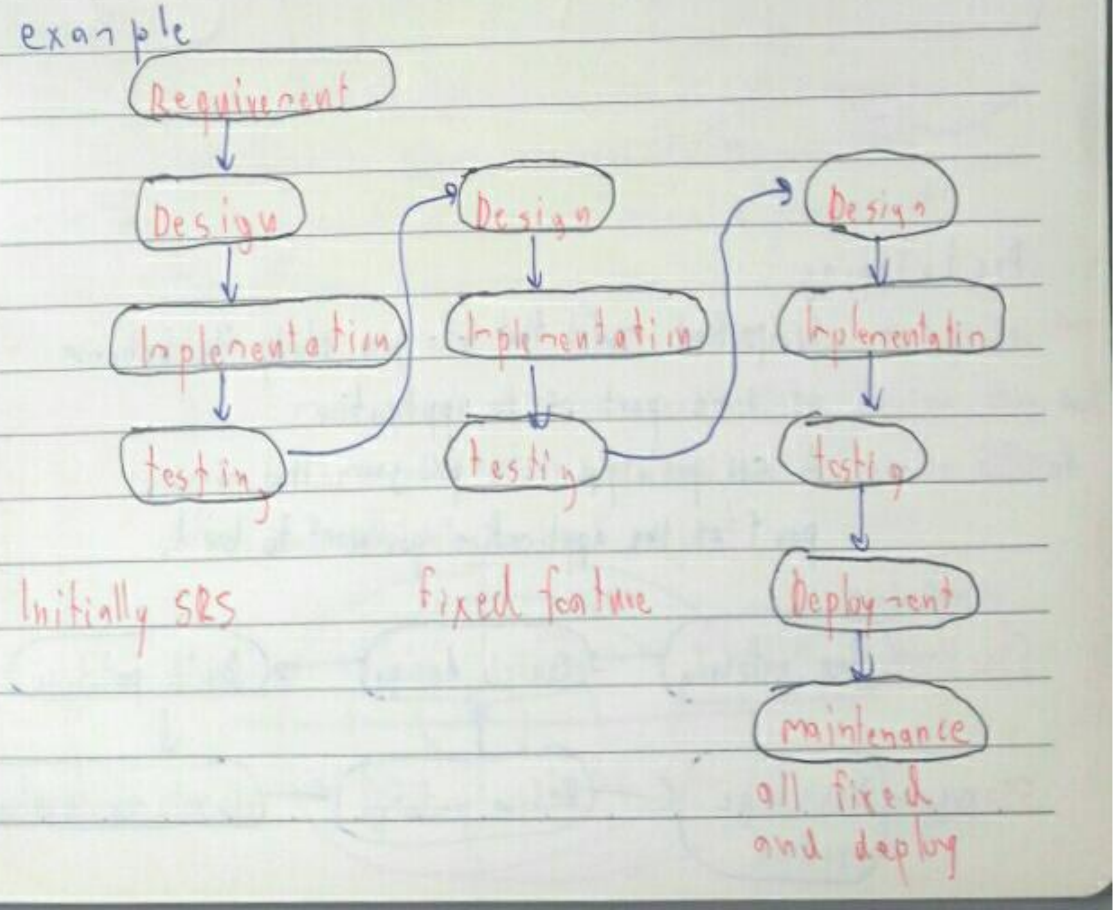
disadvantage

- need good planning / design
- need clear and complete definition of whole system
- costly than std. waterfall

Adaptive model (Iterative model)

- an approach that repeats a series of development again and again
  - Iteration 1: Initially provide some of the software specification
  - Repeat the development to get new version
  - later iteration: continue clage until reach the development of accurate application
- Pro: Suitable of large project, major requirement
- Con: Costly system architecture or design issues

example

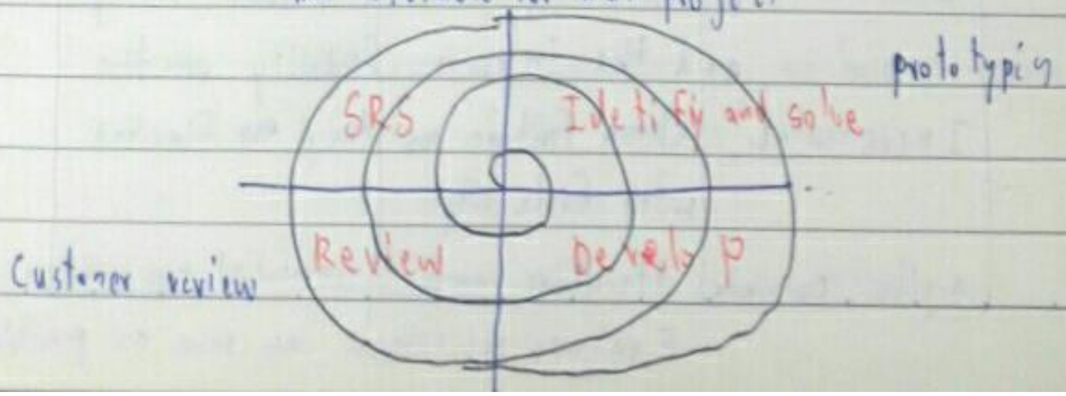


Type of prototypes

- Throwaway prototype
  - prototype to study sone aspect of the systen and then you throw it away and write code From scratch
- Evolutionary prototype
  - The prototype denostrates sone of the applicatin featu A s project progress, you refine those featue and add new ons until the prototype transfom in to the finished applicatin

Spiral

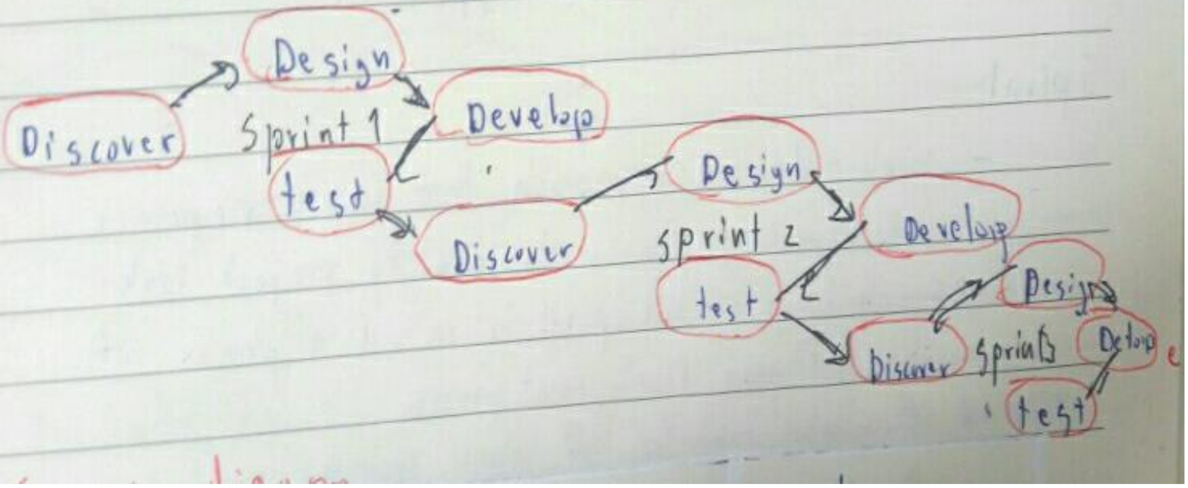
- risk - driven approach that uses a sequence of repeating phases to identify project risk
- Each loop of the spiral is called a phase of the software development process
- The exact number of phases can be varied by the Pm
- Pro: many point of reviewig and making desion very well
- Con: It's complicated risk analysis can be difficult not stutable for small project

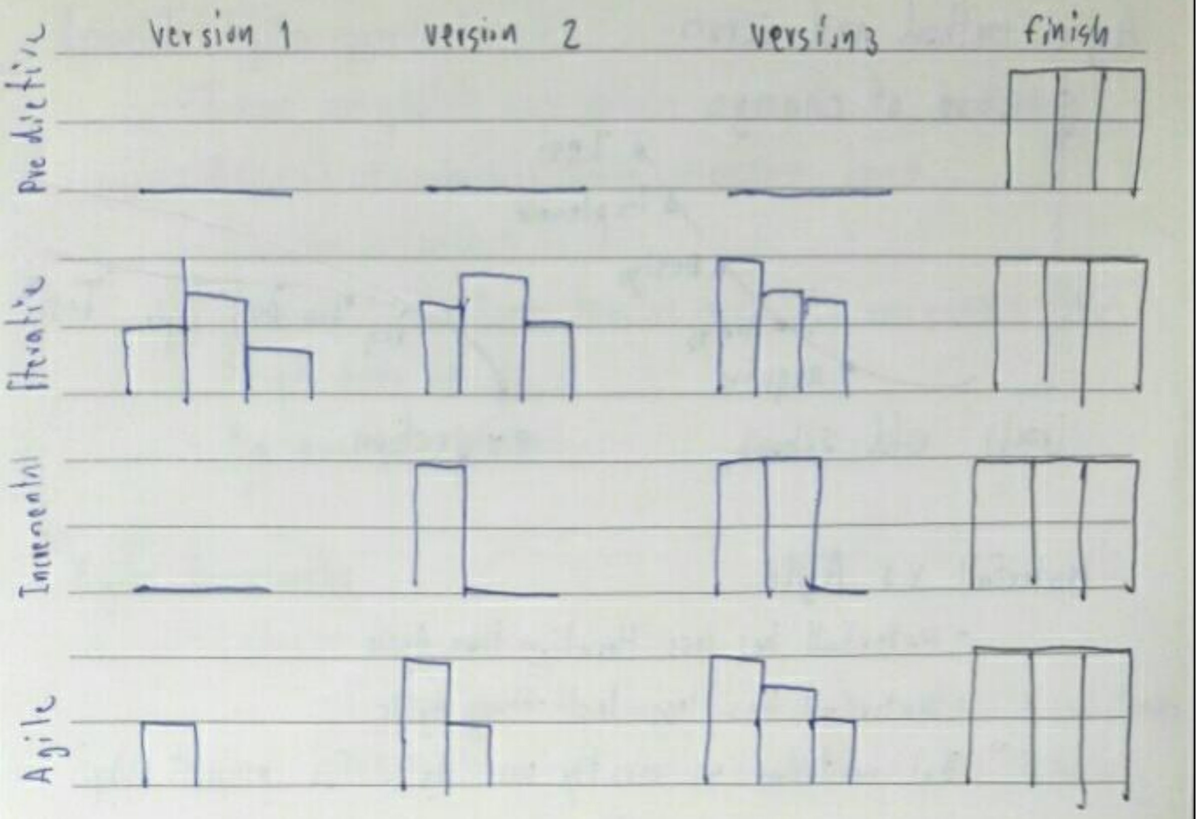


agile

- combination of both incremental and iterative model and promotes more involvement of customer during the Software development
- Pro : Rapid development, Allow change easily, focus on user/cus
- Con : Mainly depends on the customer, Not suitable For large projects, Lack of documentation

example of agile





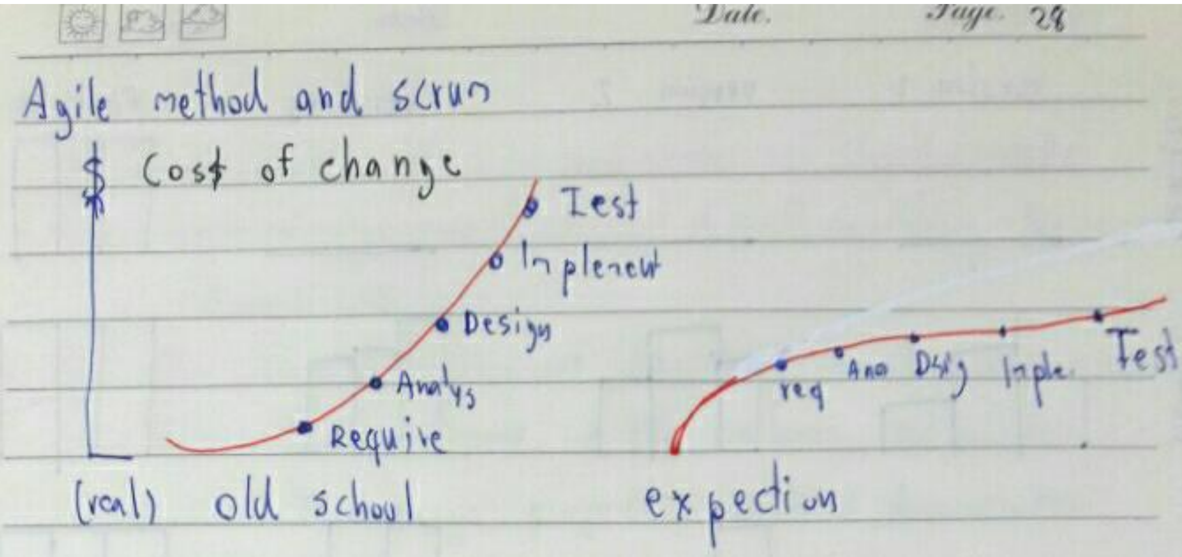
Choosing a model

appropriate model: Assure the success of a project

wrong model: Constant source of problem

How to choose the right model
- aware of what level of understanding we have of requirement
- expect of lifetime of the project
- risk - schedule constrints
- Interaction with customer - Expertise

Agile method and scrun



Waterfall vs Agile

- Waterfall has less Iteration than Agile
- Waterfall has less test than Agile

" But in today we mostly use Agile for projekt why? "

Agile method (This method found in 2001 in "Manifesto for agile SD"

- set of gaideline that help developers to generate as nuch high-quality code as possible / as quick as possibl

here is short discription

Individuals and interactions over procces and tool

Working software: over comprehensive document

Customer collaboration: over contract negotiation

Responding to change: over following a plan

Lean & Agile principle

- focus on quality and deliver real value to customer
- Accept uncertainty Build, measure, learn
- Iterative approach
- Team and Customer Involvement It's everyone's job
- Simple code design
- No one gets blamed

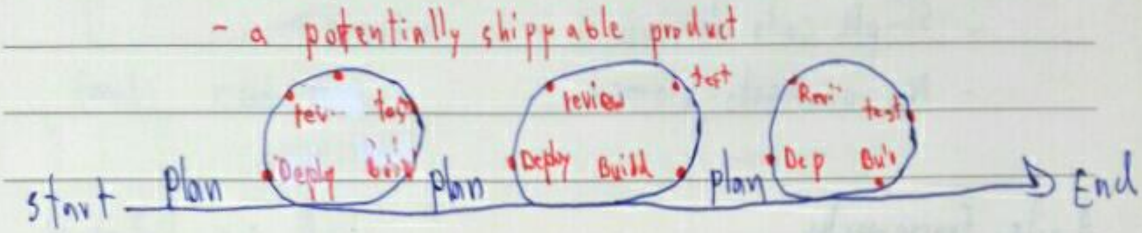
Agile frameworks

- Extreme Programming (XP)
  - lightweight methodology for small to medium sized team
  - XP Practice - Small release, Simple design, Test, Refactoring
    Pair, Programming
- kanban
  - work flow management method design to help you
    visualize the work
  - backlog, To do, In process, Done
- Scrum
  - Project management methodology with goal to
    delivering new software for 2-4 weeks
  - backlog, To do, In process, verify, done

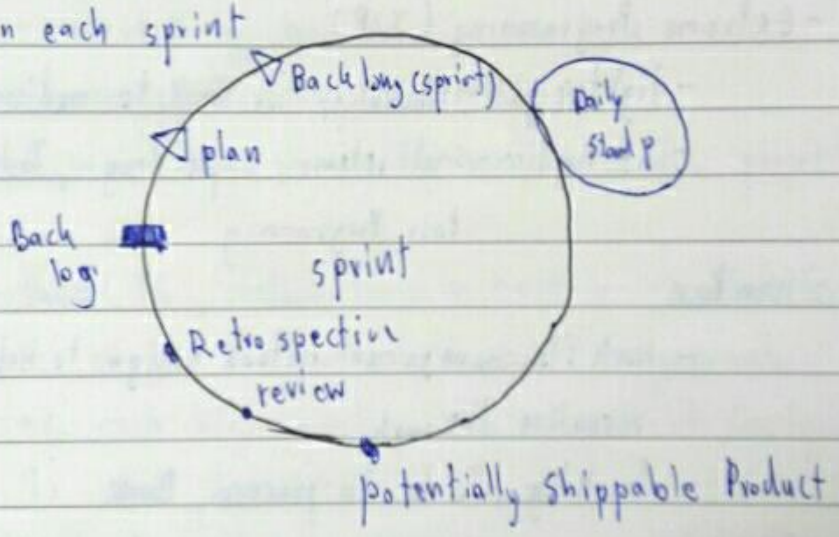
Introduction to scrum (most Agile framework popular)

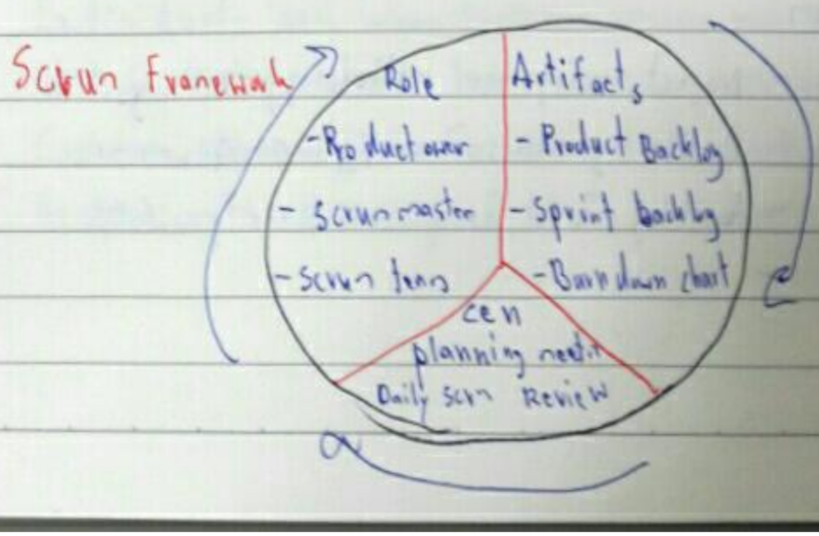
start with

- Planning a minimun of feature set
- building what has planned
- testing and review small feature set
- a potentially shippable product



In each sprint





Scrum Roles

- Product owner
  - Describe product goals
  - Define the feature
  - Prioritize feature
  - Decide on release date and content
  - be responsible for the profitability of the product
  - Adjust features and priority every iteration as needed
  - Accept or reject work result
- Scrum master
  - Represents management to the project (not PM)
  - Responsible for enacting scrum values and practices
  - Removes obstacles for the team
  - Ensure that the team is fully functional and productive
  - Enable close cooperation access all role and functions
  - shield the team from external interferences
- Scrum team
  - 5-10 people
  - Cross - functional etc tester, Programmer, UX/UI
  - members often play multiple roles
  - Team are self organizing

Artifacts

Product Backlog

- requirements for a systen, express as a prioritized list
  - list of all desire and prioritized work
  - User stories : As a [user], I want [what] so that
- Is managed and owned by a Product Owner
- Usually is created during the Sprint Planning
- The highest priority Product Backlogs go to sprint
- Can be changed and re-prioritized be for planning

Sprint backlog

- subset of Product Backlog items, which define the work for sprint
- Tean estimates the size of the task from each product Backlog
- Is created only by team member; Product Owner is not allow to doi
- Should be updated everyday by team member
- No more than 300 tasks in the list
- if a task requires more than 16 hours, it should be broken down
- changes add new tasks, remove unnecessary tasks, only by team
- Estimate

Burn down chart

- Depict the total sprint back log hours remaining per day
- Shows the amount of work remaining for the whole project
- Ideally should burn down to zero to end of the sprint
- Actually is not a straight line

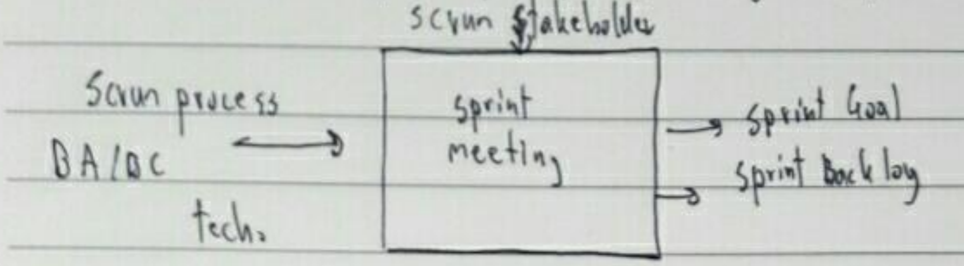
Story points = the score of task effort

velocity = the amount of estimated work the tean can perforn within each sprint

Scrum Ceremonies

Sprint Plannig Meeting

- product owner, Scrummaster and team meet to discuss the user stories and estimate their relative sizes
- the output of sprint planning meeting are Sprint Goal / Backlog



- meeting can up to 4 hour
    - Create Goal (Product Backlog)
    - Creat Back log

Daily Scrum

- Parameters
    - Daily
    - 10 - 15 minutes
    - Standup
    - Not for problem solving
- 3 question
    - What did you do since the last
    - What do you hope befor next
    - What obstables do you see in your way

Sprint Review

- occur at the end
- present complete work
- tag the task that not finish

Sprint Retrospective meeting

- Occur at the end
- discuss how to improve
- what went well and how to make it again
- what went poorly and how to avoid them
- How can we improve the next sprint