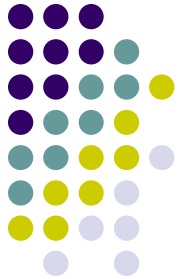


# Managing Software Development

## Day 3 Requirement specification

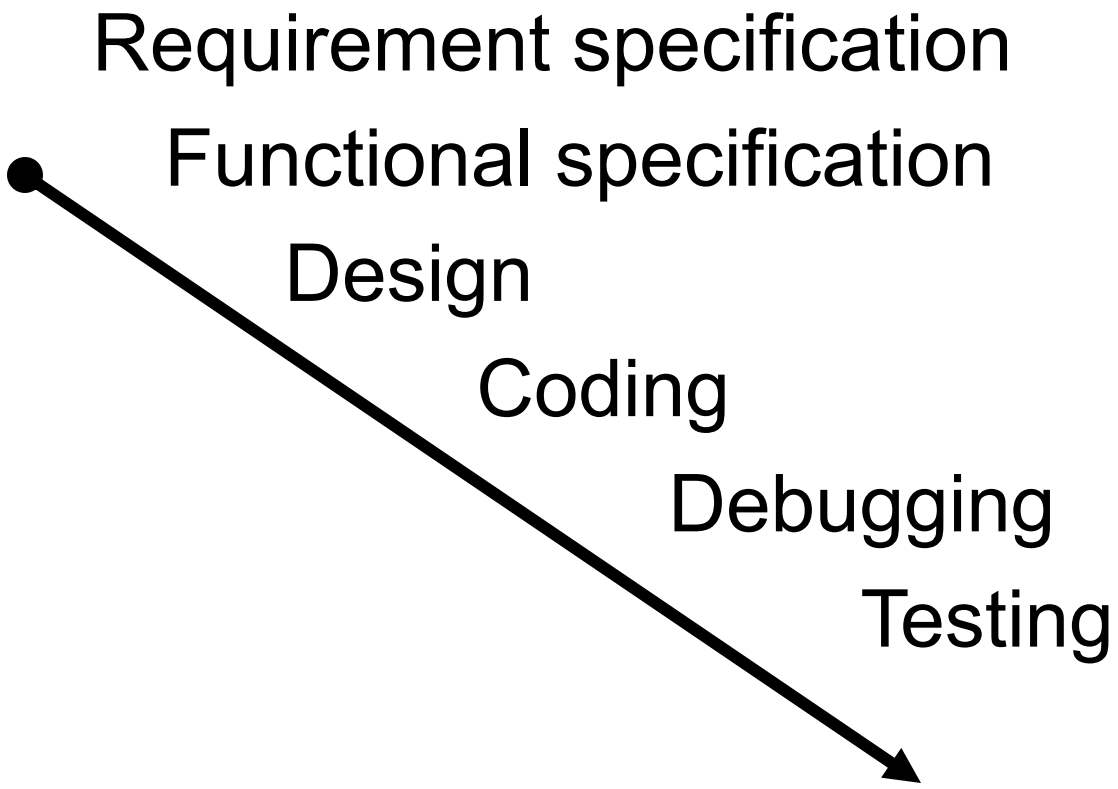
**Charoen Vongchumyen**

**Email : charoen.vo@kmitl.ac.th**

**1/2021**

**Thank to Tsuneo Yamaura**

# Waterfall model




Life cycle of software development

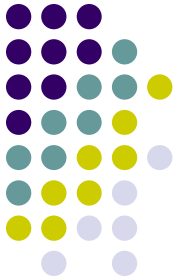
# What's Engineering?

Engineering is the way to make things again and again

Like cook book recipe

Development process

# Goal of software engineering



- Quicker → Schedule
  - Better → Quality
    - Cheaper → Cost

# CMM

- American standard for development process
- Capability Maturity Model → From Carnegie melon University
- 5 Level, 1 is lowest
  5 is Top

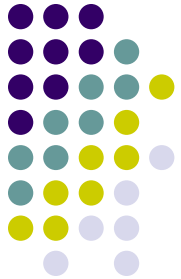
# CMM

Level 1 : Company have no development process.

No design, No document, All in their brain

Level 2 : Company has written Eng. Recipe, Development process (Some project)

# CMM

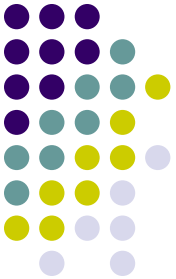


Level 3 : Development process spread all over the company (U.S. government required)

Level 4 : Engineer in project measure quality of project (error, MTBF, cost, schedule)

Level 5 : They can guide the project to better way

# CMM level in U.S. and India



(20 – 30 from 6,000 companies are level 5)

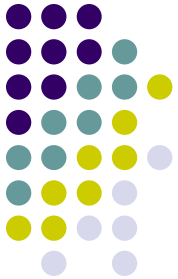
(5,000 from 6,000 companies are level 4)

(500 from 6,000 companies are level 3)

(200 from 6,000 companies are level 2)

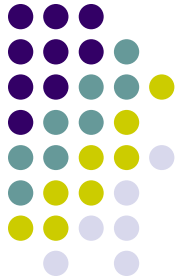
(100 from 6,000 companies are level 1)

# CMM in the real world



- India has the most level 5 but plenty of bugs
- CMM → say about the process not the players are good or not
- Level 5 but should see their outcome

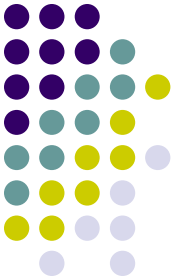
# CMM in restaurant



- In restaurant, CMM level 5 have great cook book but, if they use un-experience chef, It is not Oishi!!

- But in restaurant level 1, No cook book but use top chef, how to cook is in his brain, So go food come out.

# Development methodology



No general model methodology, Have to choose each time project start

- Structure development
- Object oriented
- Etc.

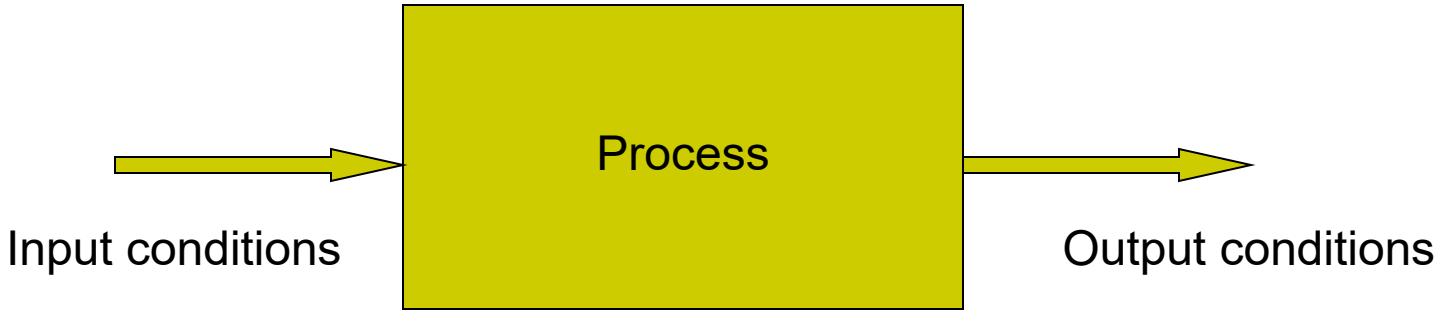
Depend on Budget, Man power, Time, Prj size

# Requirement specification

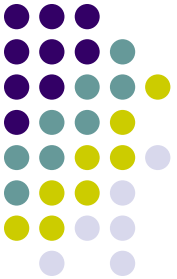
- Like operation manual

  What is input, process and output

  like cell phone manual, how to use it?



# Real world project

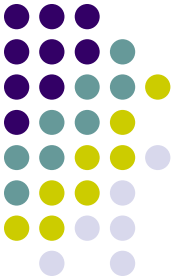


- 50%  unsuccessful projects
- 50% Semi-successful

  10% Successful and satisfy the Cost, Schedule and Quality

So, Why project fails????????

# Why project fails?



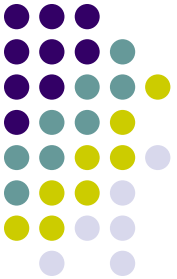
- Requirement is not fired.

  48% all moving, not stable

  How to deal with it?

  1. Separate contractor each step.
  2. Req. Spec. sight and change are charge

# Why project fails?



- Misestimation (Under estimation) 47%
  1. Quality — Plenty of bugs
  2. Cost — 20 – 50 % over budget
  3. Schedule — 100% delay

Too optimistic!!

# Three ashes

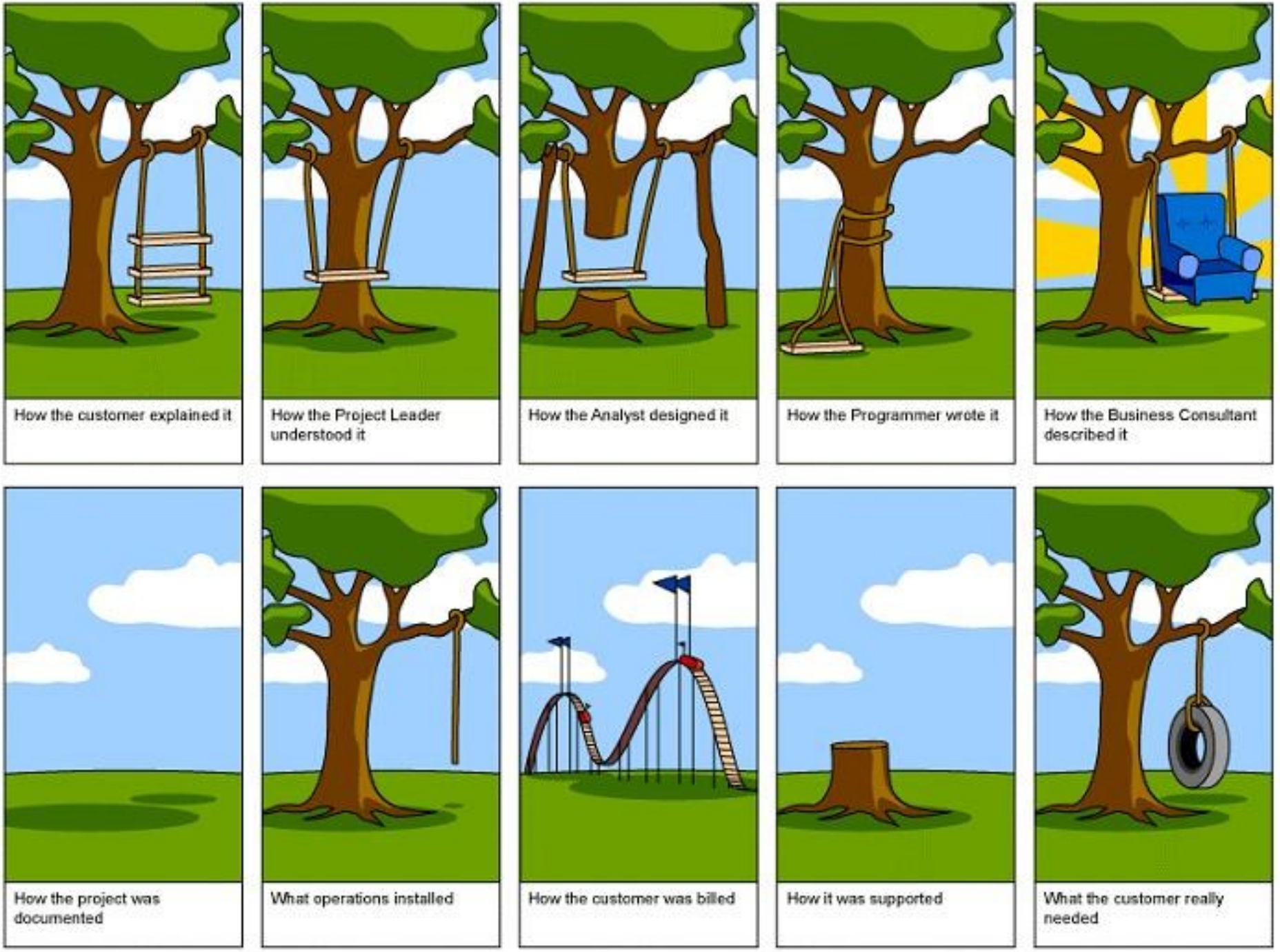
"If we can't build SW in conditions,

So pick up some important function"

Must be
Should be
Desired

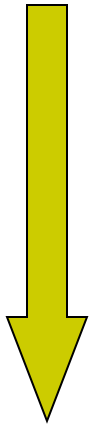
Like put priority
on injured soldier
in hospital
who will be
taken care first

**20% modules = 80% functions**


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

# Cost of phases

- 8% Requirement and function specification
- 5% Design
- 7% Coding
- 5% Debugging
- 7% Testing
- 60% Maintenance

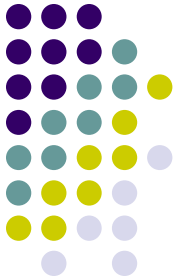


**1st version**

**2nd version**

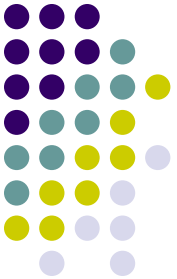
2 – 5 years past style and from now on

# Type of maintenance



- 20% Corrective maintenance (bugs fix)
- 60% Enhancement (Add more functions)
- 20% Adaptive maintenance (Make I work on some OS different, Environment → Porting)

# Difficulty of Req. Spec



- Ambiguous (Doubt)
- Incomplete
- Contradictory
- Difficult to understand

# Short practice
## (Group of 3)

**Rock, paper, scissor** **เป่ายิ้งฉุบ**

**Jan-ken-pon**

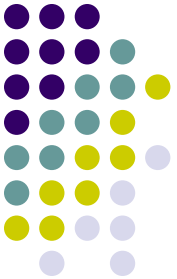


# Software Engineering

- The word “Software engineering” was born in 1968, NATO conference.
- Just word, No development process. Then the function , procedure was created.

CASE = Computer Aid Software Engineering (tool)

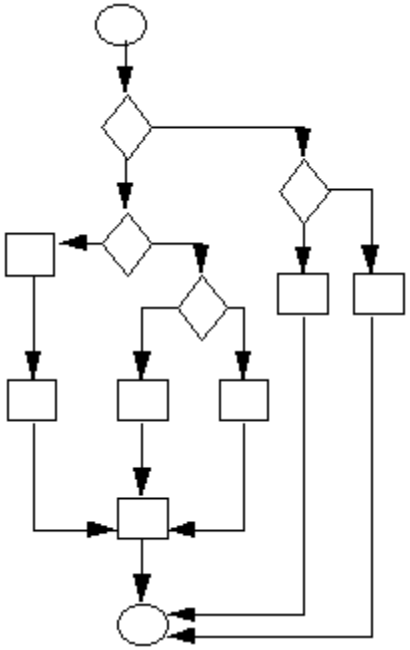
# Software Metric



- 1978, The first of software metric paper was released.

**“If you can’t measure, You can’t control ”**

# Cyclomatic number



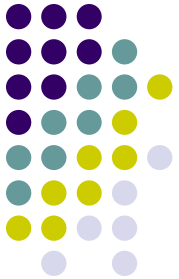
Measure complexity of project

$$C = E - N + 2P$$

E = Arch,
N = Node,
P = 1,
C = Complexity

- Don’t work for real life
- Just close area

# Software Metric



Measure something of software

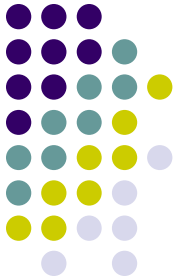
- Metrics after the fact (why you spend so much money, Why delay)

  Do after coding (such as Baseball, why we loose the game)

- Metrics before the fact

  Do before coding (In Baseball, Whose gonna be next putter)

# Conclusion



SW metric Is the measurement sth in SW to meet quality control

Such as the test cases and bugs

**“You can’t manage what you can’t measure”**

# Group exercise

Enter the length of sides

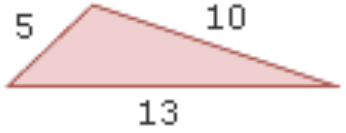
Side 1

Side 2

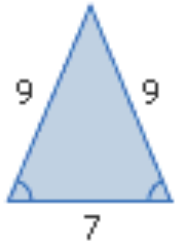
Side 3

Result

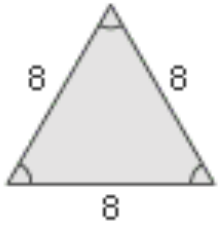
Enter



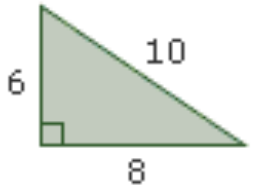
scalene triangle



isosceles triangle



equilateral triangle



right triangle

Define hours you spend, person (man-power) and measure your result