Software Engineering Practices:: CAP437

Lecture 0: The kick start session

(Course Introduction)

Structure of Lecture 0

- General Course Information/Overview
- Introduction into Software Engineering

Course Information/Overview

- LTP – 4 0 0 [Four lectures/week]

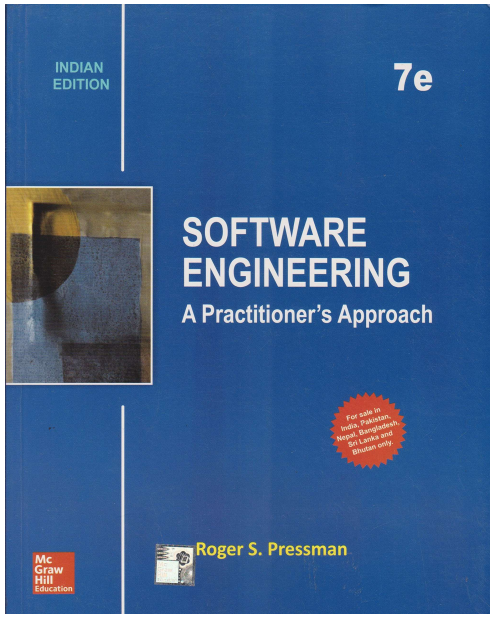
Text Books:

1. SOFTWARE ENGINEERING A PRACTITIONERS APPROACH by R.S. PRESSMAN, MCGRAW HILL EDUCATION

References:

1. FUNDAMENTALS OF SOFTWARE ENGINEERING by RAJIB MALL, PHI Learning

2. AN INTEGRATED APPROACH TO SOFTWARE ENGINEERING by PANKAJ JALOTE, NAROSA PUBLISHING HOUSE



Course Assessment Model

Marks break up

• Attendance	5
• Continuous Assessment(2 out of 3)	25
• MTE	20
• ETE	50
Total	100

The hitch…

The three BURNING questions in mind…

- Why are we learning Software Engineering?
- What would we do with it?
- What will be the course outcome?

Course Outcome

After the completion of the course the students will be able to

- **Represent the given project in various phases of a lifecycle.**
- **Select appropriate process model depending on the user requirements.**
- **Apply the knowledge, techniques, and skills in the development of a software product.**
- **Differentiate various processes used in all the phases of the software development.**
- **Illustrate various life cycle activities like analysis, design, implementation, testing and maintenance.**

Learning Outcomes

Upon successful completion of this course, you should be able to demonstrate basic knowledge of and skills in:

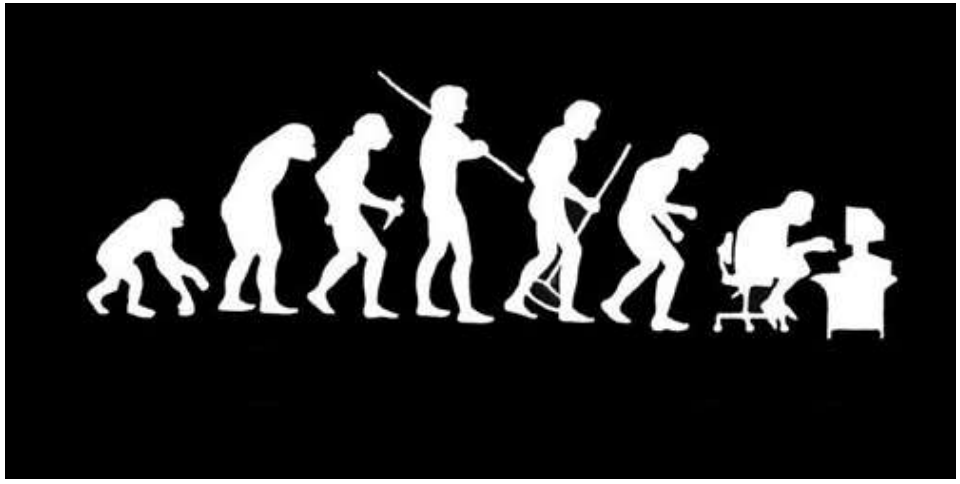
- software engineering paradigms;
- system analysis;
- requirements analysis;
- planning;
- implementation;
- quality assurance (verification and validation; testing);
- maintenance (evolution);
- Software Metrics;
- software processes and methodology.

Let us re-invent ourselves

To begin with basics…

Let us go to basics.

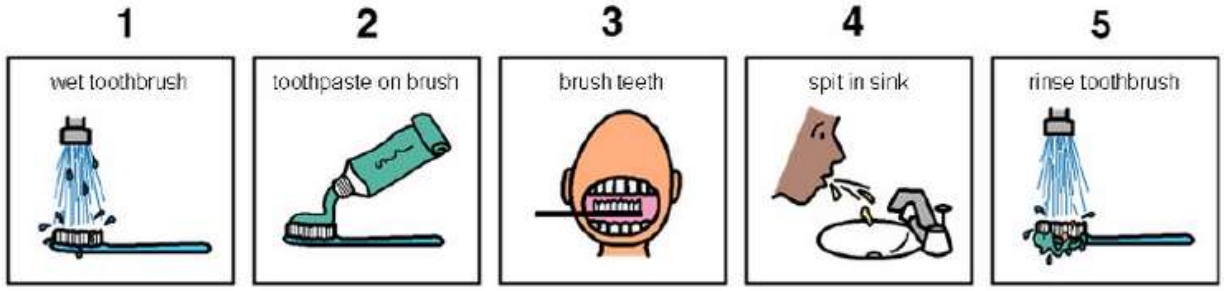
Let us begin from toddling to learn to walk



Get ready to be childish….

Daily routine

- Let us look around our daily routine…
- Let us see where all we use processes everyday
- Simple things we do to start the day



So there are steps you know to clean your teeth…

Daily routine

- Yet another example but more complex

Vision Sensor

Smell Sensor

Radiation Sensor

Touch Sensor

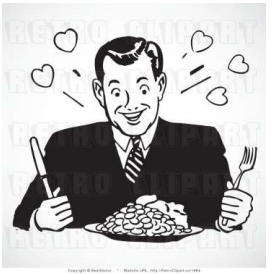
If all sensors ✓ = SIP

If even one is ✗ = WAIT!!!

So what does this mean?

- Take ANY activity of the day…
- It will have a set procedure
- It has to be done in a designated way
- If not done the specified way will yield wrong results.
- Success in doing it depends on how closer one is to the prescribed method.
- This clearly shows that to accomplish any task we need to follow

PROCESSES



What next?

If there are processes in accomplishing any task

- There has to be ways to different ways/ processes
- There has to be methodology to verify/validate the accomplishment
- There has to be methodology to ensure quality.
- There has to be measures to quantitively establish task accomplished as per scope, budget and time.
- And for all this…

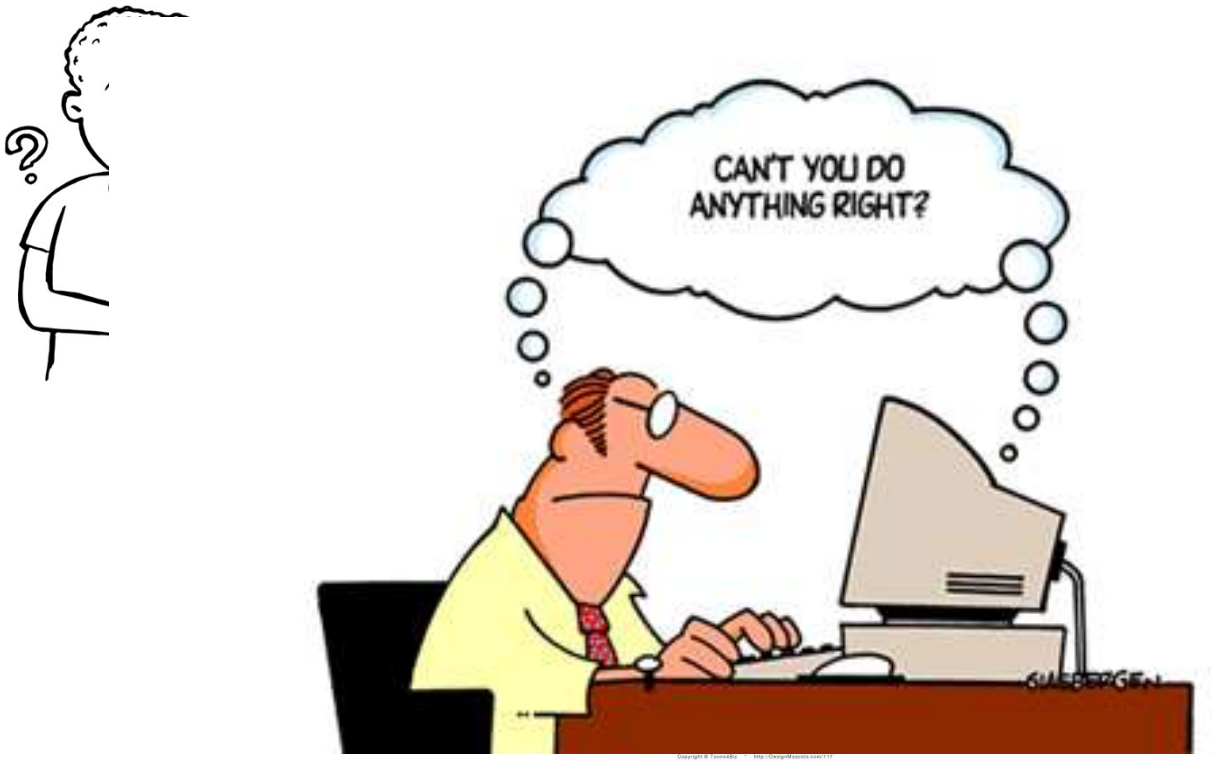
What next?

- There has to be processes to DEVELOP software

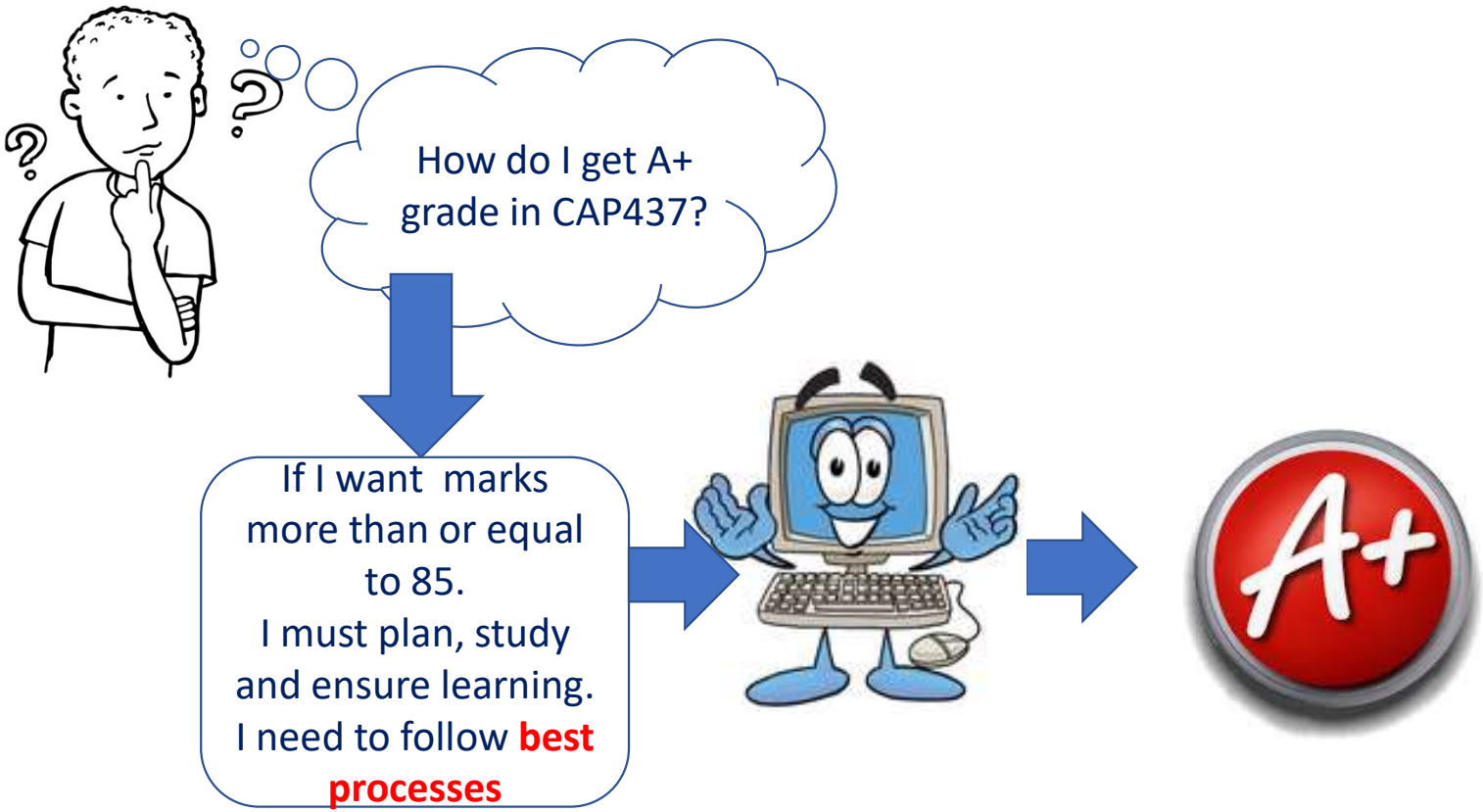
- There has to be processes to plan the software development
- There has to be processes to ensure quality of software
- There has to be processes to provide **EARLY WARNING** singles if things are not as per plan

Otherwise….

Diving deeper…



Diving deeper…



What do we need to know?

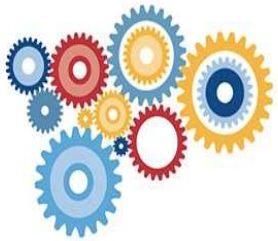
Unit 1

Tamil

தமிழ்

Punjabi

ਪੰਜਾਬੀ



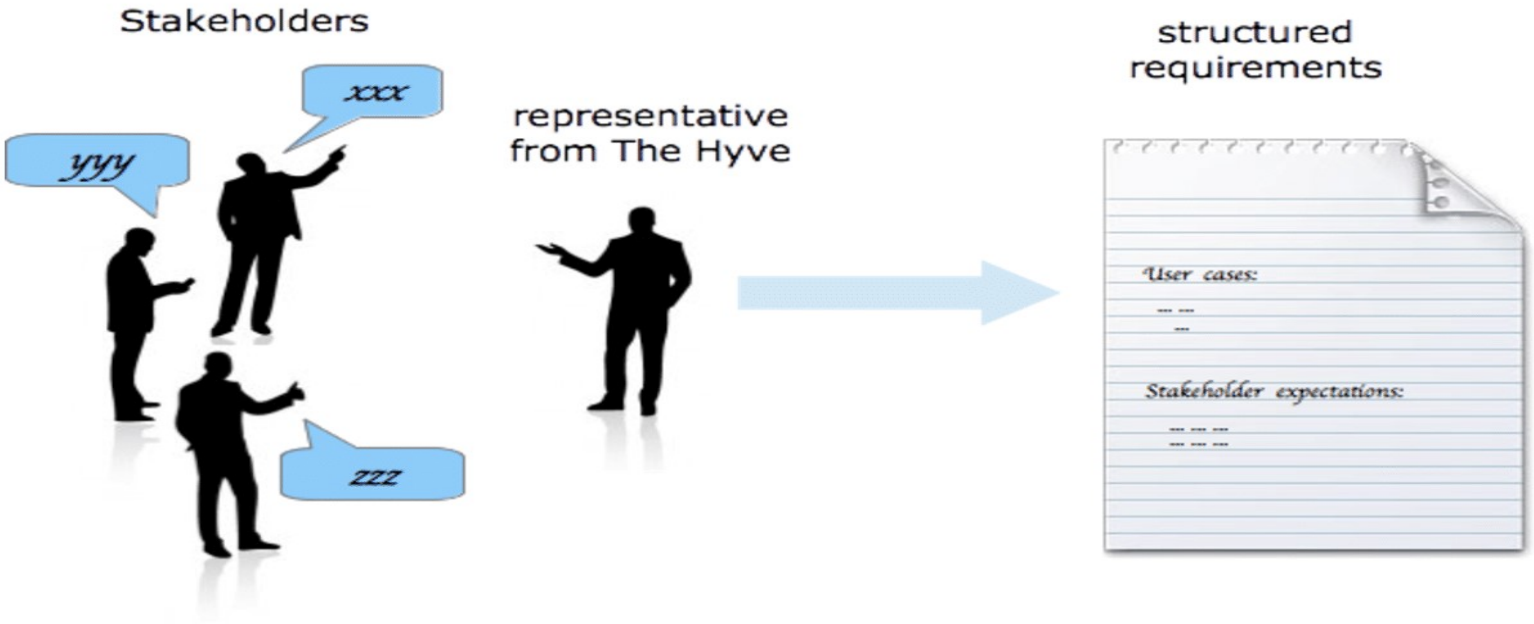
Processes

Need of Processes :: **Introduction to software engineering**

Understand. :: S**oftware process models**

What do we need to know?

Unit 2



Understand what is to be done:: Requirement gathering

What do we need to know?

Unit 3

Putting together individual requirement like pearls in a necklace:: **User interface design**

What do we need to know?

Unit 4

Appealing to users. :: **User interface design**

Can I do the way I like? :: **Standards**

What do we need to know?

Unit 5

How to ensure quality of software:: **Software Testing**

What do we need to know?

Unit 5

Defects in software:: **Bugs (term coined by** Grace Hopper

The Scenario Design and the Test Case Testing Life Cycle normally starts in parallel with the Development Cycle.

Bug Free Software

What do we need to know?

Unit 6

S.No.	Testing Metric	Data retrieved during test case development & execution
1	No. of Requirements	5
2	Avg. No. of Test cases written per Requirement	20
3	Total no. of Test cases written for all requirements	100
4	Total no. of Test cases Executed	65
5	No. of Test cases Passed	30
6	No. of Test cases Failed	26
7	No. of Test cases Blocked	9
8	No. of Test cases un executed	35
9	Total No. of Defects identified	30
10	Critical Defects count	6
11	High Defects Count	10
12	Medium Defects Count	6
13	Low Defects Count	8

How to ensure reliability and longevity? :: **Software maintenance** & **metrics**

Next : Introduction to software engineering