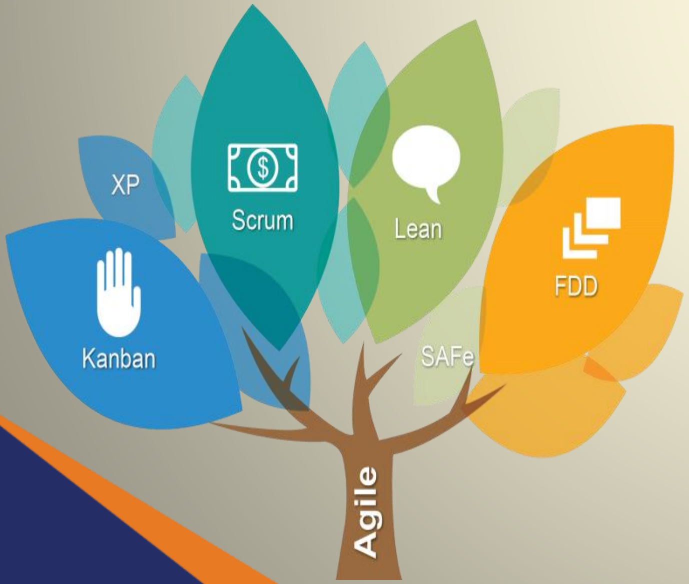
Software Process Model 2017

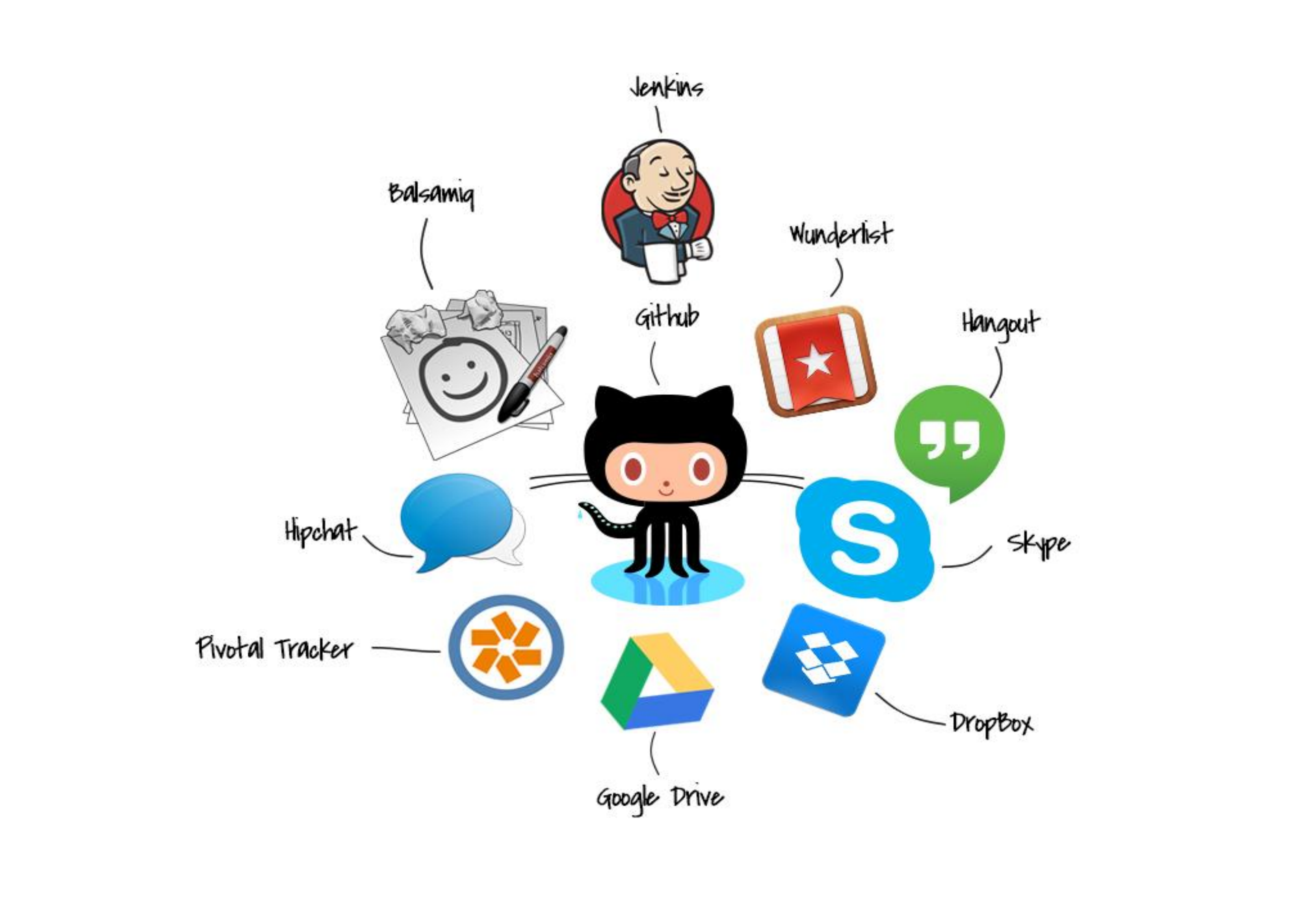
Introduction



Session outcomes

- Introduction to Module
- SPM- Introduction

MODULE INTRODUCTION



Module contents

- Course web
- General
  - Module outline
  - Notices
  - Marks
- Weekly updates
  - Lecture
  - Lab
  - Tutorial
  - Additional Reading

Learning outcomes

- Identify what software engineering is and relate it to other disciplines.
- List the characteristics and effects of different types of software engineering processes.
- Describe the requirement engineering process and to write a formal requirements document for a software project.
- Use UML to model system requirements through Use Case Diagrams, Use Case Scenarios and Activity Diagrams.
- Illustrate system requirements using modelling tool.
- Explain the fundamental principles and practices associated with Agile processes.
- Recognize and differentiate software design strategies.
- Describe and compare different testing strategies used in software development.
- Compare and contrast modern software development methodologies.

Assessment Criteria

Mid Term Examination	30%	LO1-LO4
Assignment I	10%	LO3-LO5
Assignment II	10%	LO4-LO5
Final Examination	50%	LO1-LO9

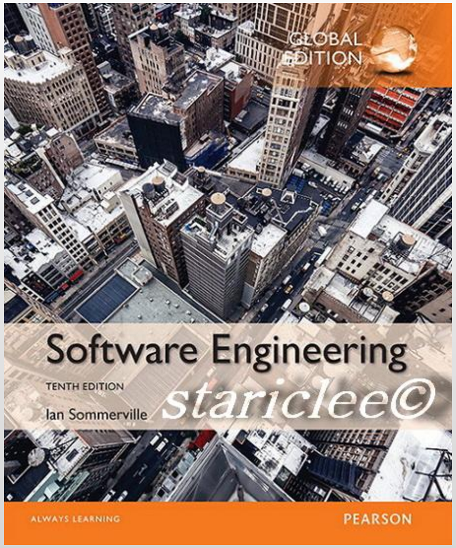
To pass this module, students need to obtain a pass mark in both "Continuous Assessments" and "End of the Semester Examination" components which would result in an overall mark that would qualify for a "C" grade or above.

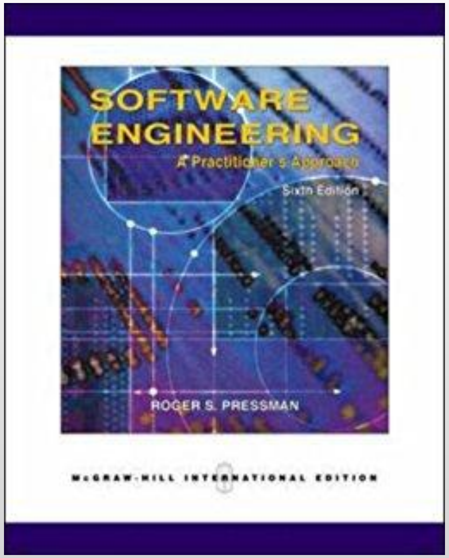
Assignments

- **Five members** should be in one group
- Randomly chosen case study
- Two submissions
  - Before Mid term
    - Based on requirements engineering and use case diagrams
  - After Mid term
    - Based on Activity Diagram

Recommend Texts

- ***Ian Sommerville, "Software Engineering", Pearson Education Limited, 10th edition, 2016***
- ***R. Pressman, "Software Engineering: a practitioner's approach", McGraw-Hill Education; 8th edition, 2014***
- ***K.S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley, 2012***
- *SWEBOK, Guide to the Software Engineering Body of Knowledge, 2014*

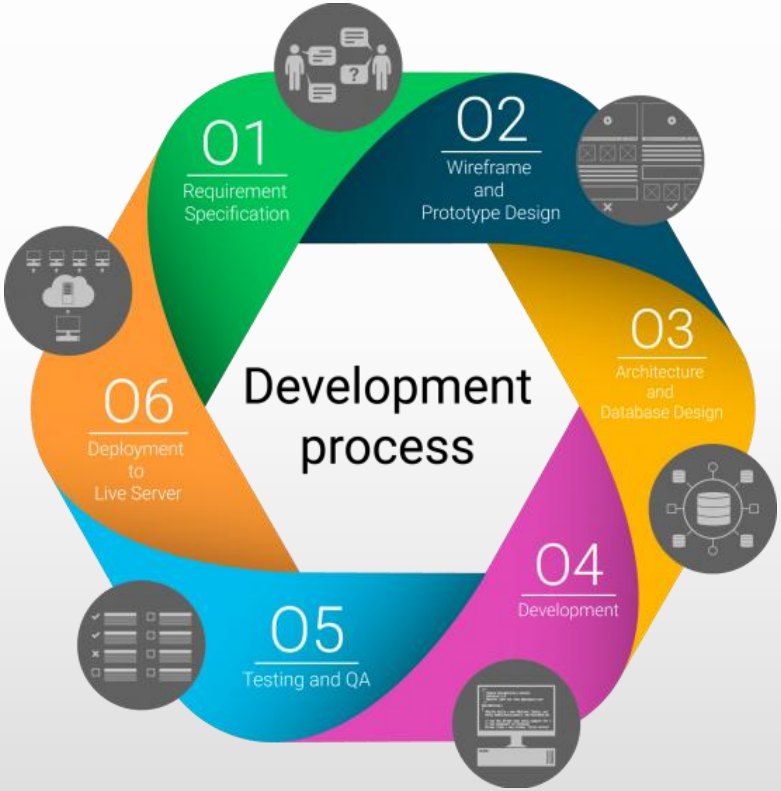




Lab Schedule

Time	Lessons	Labs
Week 1	Introduction to Software Process Modeling	PS
Week 2	SDLC Models	PS
Week 3	SDLC Models	PS
Week 4	Requirements Engineering	PS
Week 5	Use Case Diagram	SPM - Use Case Diagram
Week 6	Use Case Diagram	SPM - Use Case Diagram
Week 7	Activity Diagram	SPM- Activity Diagram
Week 8	**Midterm Examination**	
Week 9	Software Design	SPM- Activity Diagram
Week10	Implementation and Testing	PS
Week 11	Implementation and Testing	PS
Week 12	Modern Software Development Methodologies	SPM- Agile
Week 13	Modern Software Development Methodologies	SPM- Agile

SPM- INTRODUCTION


01
Requirement Specification
02
Wireframe and Prototype Design
03
Architecture and Database Design
04
Development
05
Testing and QA
06
Deployment to Live Server
Development process

Outline

1. What is a Software
2. What is Software Engineering
3. Software Process
4. Software Process Activities
5. Software process model
6. Software Development Life Cycle
7. Some frequent Questions
8. Software Engineering Ethics

What is Software?

Software is not only the computer programs, but also associated documentation and configuration files, needed to make the programs operate correctly.

+

Software product can be

- Generic products

These are stand alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.

- Customized

These are systems that are developed for a particular customer requirements

Popular Software

W
WinRAR®
Linux™
Ps
android
TM
Microsoft®
.NET
MySQL®
nero
Burning ROM

1st Year 1st Semester IP

```
/* adding two numbers*/
#include <stdio.h>

int main(void)
{
            int no1, no2;
            int sum;

            no1 = 25;  //  assign value to no1 variable
            no2 = 12;  //  assign value to no2 variable

            sum = no1 + no2;  // add numbers

            printf( " Sum is %d\n",  sum);    // print sun
            return 0;
} // end of main function
```

```
/* adding two numbers*/
#include <stdio.h>

int main(void)
{
            int no1, no2;
            int sum;

            printf("Enter first number:  ");   /* prompt */
            scanf("%d", &no1);  /* read the value */

            printf("Enter second number:  ");    /* prompt*/
            scanf("%d", &no2);  /* read the value */

            sum =  no1  + no2;    /* assign total to sum */

            printf( " Sum is %d\n",  sum);    /* print sum */

            return 0;
} // end of main function
```

- Are these Software ?
- What are things that you need to do to develop Software?

Programs Vs. Software Products

Program

- Small
- Single developer
- Small in size
- Limited Functionality
- Single user (author)
- Simple user interface
- Sparse documentation
- No user manual
- Ad hoc development

Software Product

- Large
- Team of developers
- Multiple users (customer)
- Complex user interfaces
- Detailed documentation
- User manual
- Systematic development

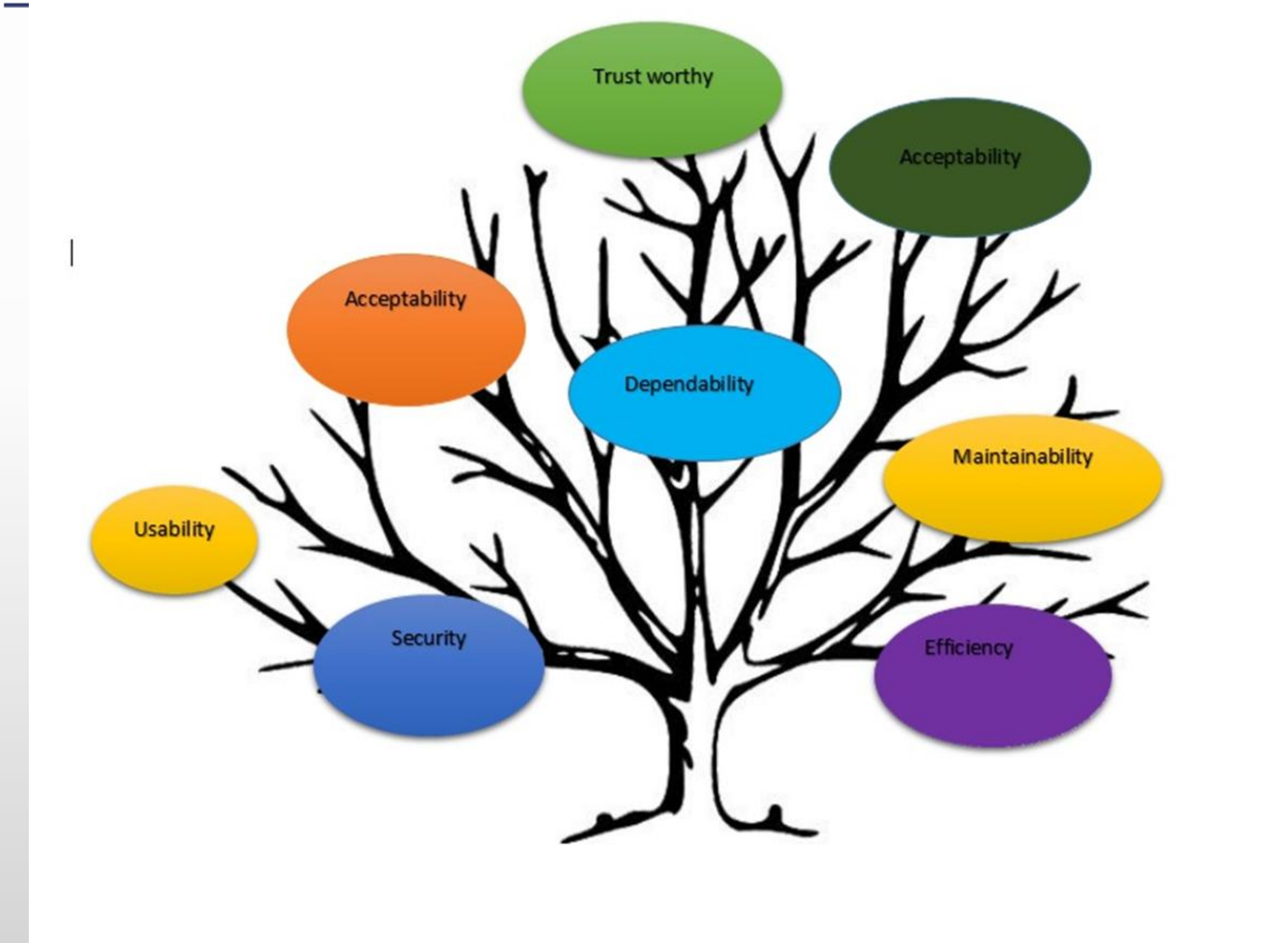
How do we develop a real software?

- There will be a real user (Customer) who would need to use the software.
  1. Feasibly study (whether it is technical feasible and financially worthwhile)
  2. You have to find out what the customer wants (Requirements Gathering)
  3. Analyze the problem
  4. Develop a solution (Design)
  5. Code the solution
  6. Test and Debug
  7. Maintenance

Essential attributes of good software

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable, efficient and usable.

Essential attributes of good software cont.



Suggest Something Innovative?

- Do not think about technical barriers
- You can think beyond of the reality

“New Ideas will lead you to highest point of the Software Engineering”

Software Engineering

- IEEE Definition of Software Engineering:

The application of a *systematic, disciplined, quantifiable* approach to the development, operation, and maintenance of software;

that is, the application of engineering to software.

IEEE Standard 610.12-1990, 1993.

Software Engineering Cont.

- Engineering discipline

  make things work by applying theories, methods and tools where these are appropriate and also try to discover solutions to problems even when there’s no proper theories/methods.

- All aspects of software production

  Not only technical processes of software development, but also project management and development of tools, methods and theories to support S/W production.

Goals in Software Engineering

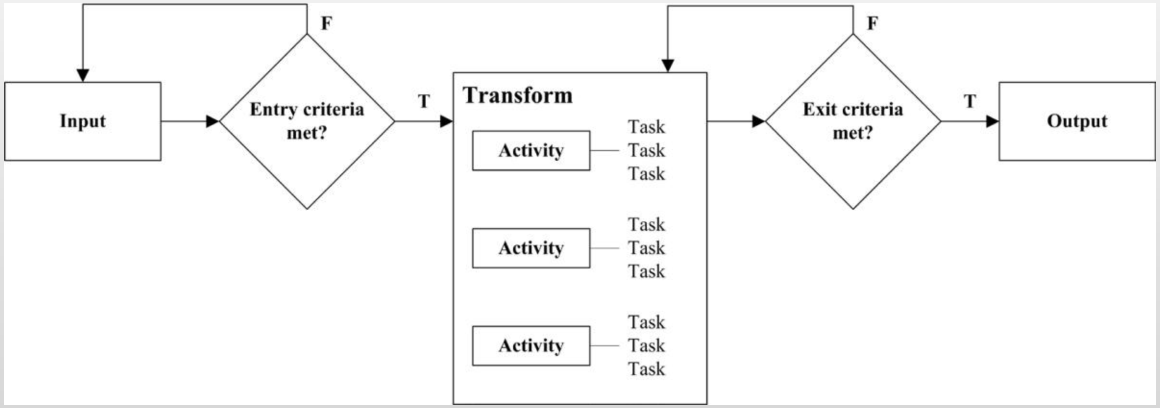
- To deliver the software to the customer at the agreed time
- To keep overall costs within budget
- To deliver software that meets the customer’s expectations
- To maintain a coherent and well functioning development team

Key Challenges

- The product is intangible
- Large software projects are often “one-off” projects
- Software processes are available and organization – specific
- Deliver Quality software to the customer at the agreed time
- Keep overall costs within budget

Software Process

- A Software process is a set of activities and associated results that leads to the production of a software product.
- A software process is a set of interrelated activities and tasks that transform input work products into output work products. (SWEBOK V3)



Making A Cup of Tea

- Ingredients : Tea Leaves, Sugar, Milk Powder, Boiled Water
- Process

  Boil the water

  Pour boiled water into cup

  Put a tea bag inside a cup

  Leave it few minutes

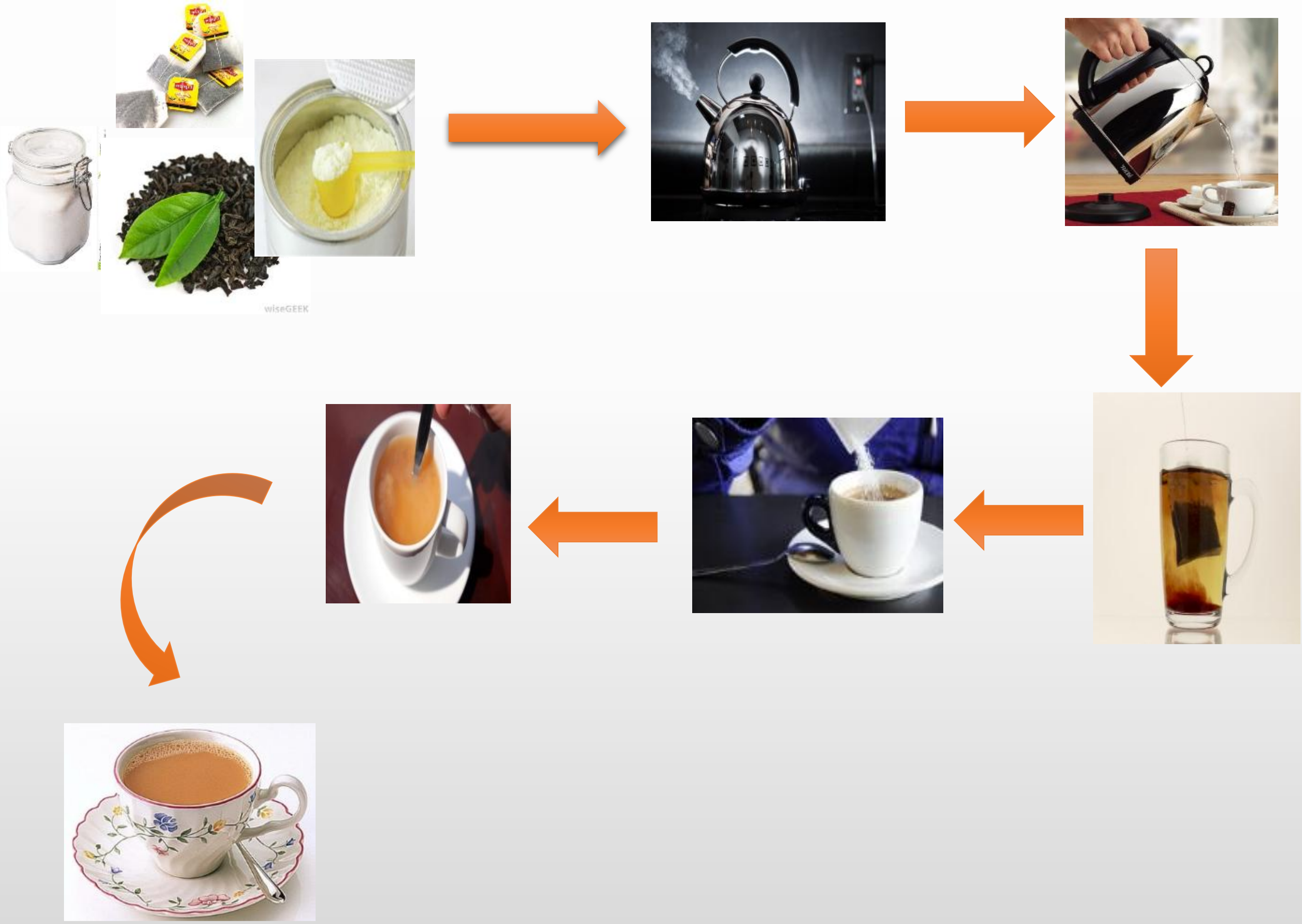
  Put Sugar and Milk (if necessary)

  Stir few seconds

  Arrange it nicely

- Output: Tea

Making A Cup of Tea



Software Process Activities

- Software Specification
- Software Development
- Software Validation
- Software Evolution

Software Process Activities

- Software Specification

The functionality of the software and constraints on its operation must be defined.

- Software Development

The software is designed and programmed. The software should meet the specification agreed earlier.

- Software Validation

The software must be validated to ensure that it does what the customer wants

- Software Evolution

The software must evolve to meet changing customer needs.

Software Processes

"There is no universal process that is right for all kinds of software"

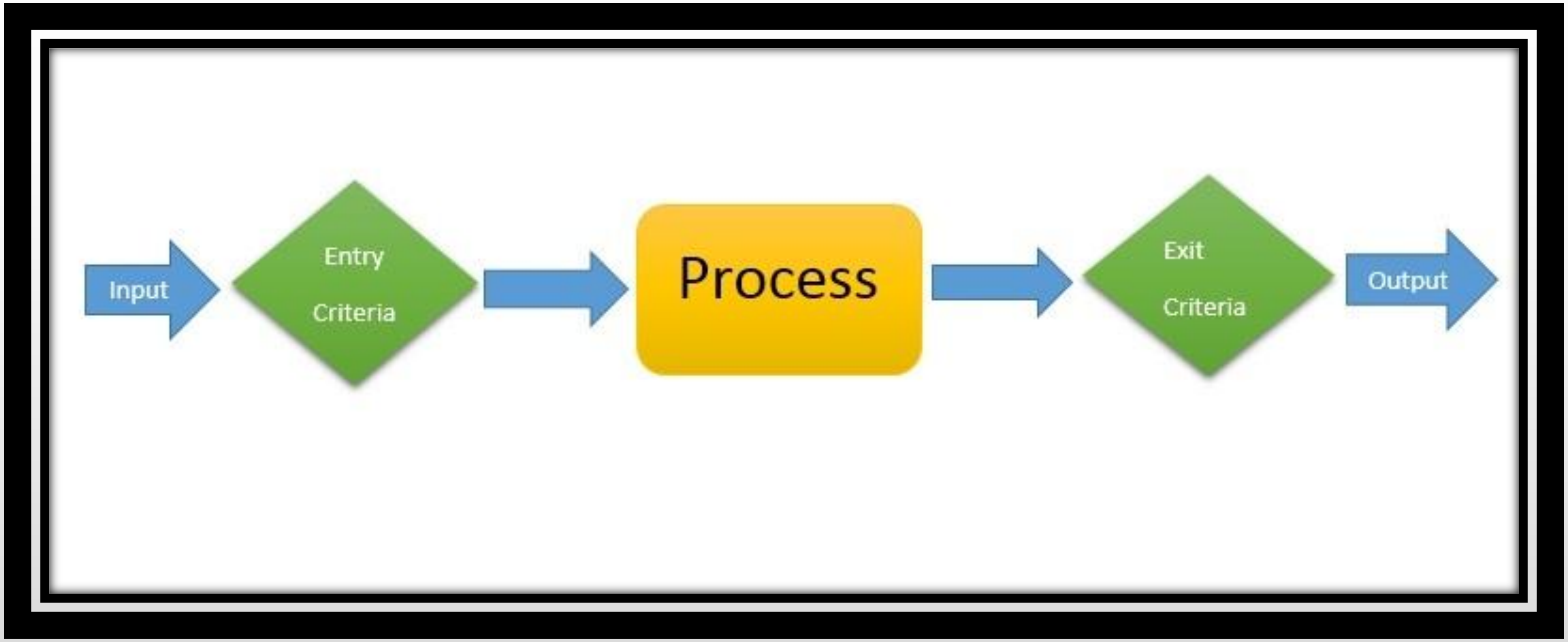
Most companies have developed their own development processes. Different Software needs different methodologies.

Ex:

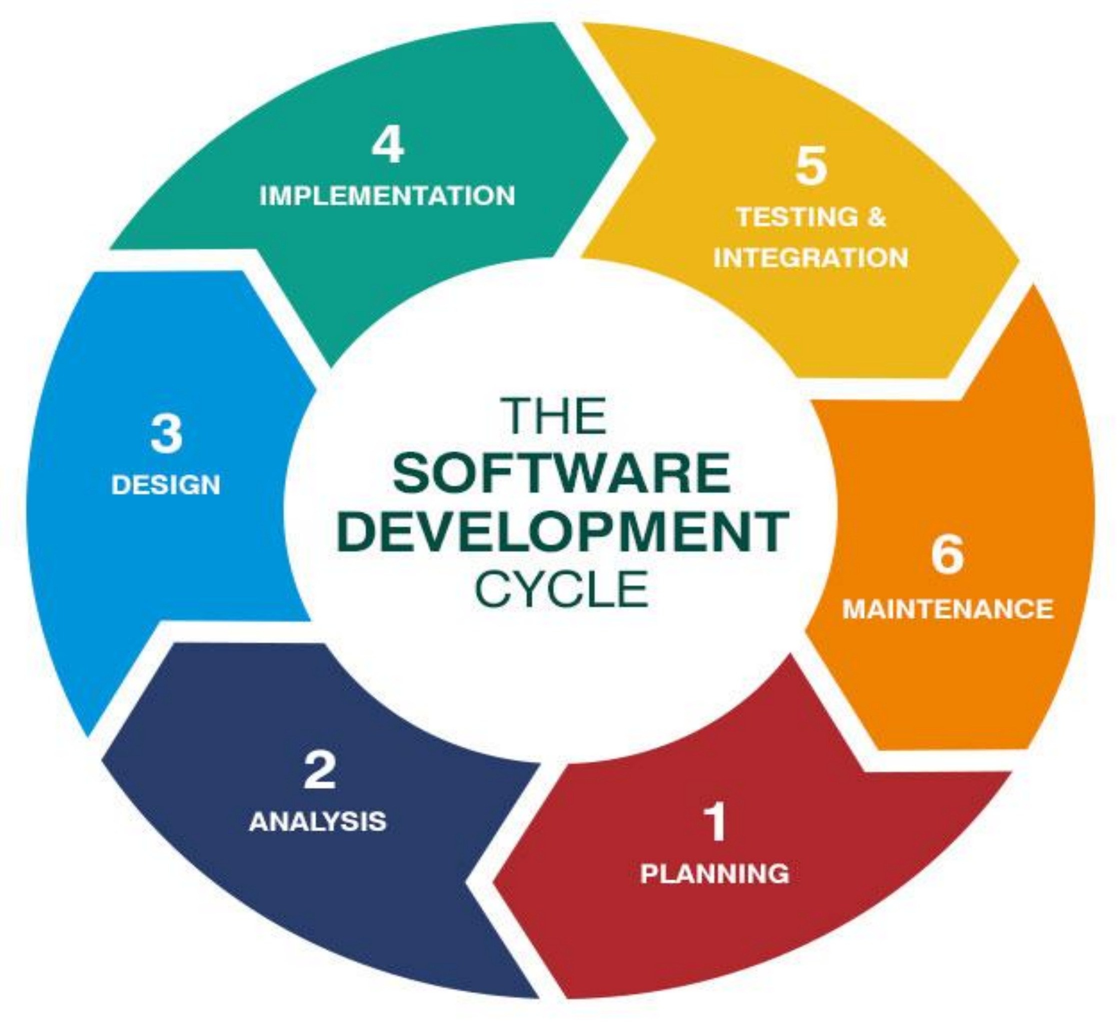
- For safety-critical systems, a very structured development process is required where detailed records are maintained.
- For business systems, with rapidly changing requirements, amore flexible ,agile process is likely to be better

Software process model

- It is a simplified representation of software process.



Software Development Life Cycle



Software Development Life Cycle Models

- A software development life cycle is the series of identifiable stages that a software product undergoes during its life time.
- A software life cycle model is descriptive and diagrammatic representation of the software life cycle.
- Since a software life cycle has somewhat similar implications with regard to software development, a software life cycle model is often referred as software process model.

Software Development Life Cycle Models Cont.

- A life cycle model represents all the activities required to make a software product transit through its life cycle phases.
- Different life cycle models may map the basic development activities to phases in different ways.
- Thus, no matter which life cycle model is followed, the basic activities are included in all life cycle models though the activities may be carried out in different orders in different life cycle models.

General Software Process Models

- Waterfall Model
  - Classic
  - Iterative
- Prototyping
- Evolutionary Model
  - Incremental
  - Spiral
- Rapid application development.
- Agile development.

Software Engineering Ethics

As a Professional Software Engineer,

- You should accept that your work involves wider responsibility than simply application of technical skills
- You should behave in an ethical way and morally responsible way
- You should not use your skills and abilities to behave in a dishonest way that will bring disrepute to the software engineering profession

Software Engineering Ethics Con.

Standards

- Confidentiality
- Competence
- Intellectual Property rights
- Computer misuse

Case Studies

- Library Management System
- Effective Doctor Patient Management System

Library Management System

- Sri Lanka Institute of Information Technology (SLIIT) is the largest degree awarding institute in Sri Lanka with degree programs diversified to computing, business and engineering. In order to cater to its growing need of knowledge the institute maintains a Library Information System connecting Malabe, Metropolitan and Matara campuses. Each holds a latest collection of books and periodicals, particularly in the field of Information Technology, business management, engineering, general English, architecture and quantity surveying. The library of the Malabe Campus acts as the main resource center through which all library development activities are coordinated. SLIIT libraries are open to SLIIT students daily including weekends from 7.30 AM to 7.00 PM.

Tasks carried out at the library

- Add library materials
- Manage Library membership
- borrow books
- return books
- Pay fine on overdue materials
- Refund library deposit
- Replace lost library material
- Search library materials
- Generate reports

Effective Doctor Patient Management System

National Health Service (NHS) is the recognized doctor patient handling and management system that helps doctors in their work and also patients to book doctor appointments and view medical progress. The system allows doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name. The system manages the appointment data for multiple doctors for various date and times. Each time a user visits a doctor his/her medical entry is stored in the database by doctor. Next time a user logs in he may view his/her entire medical history as and when needed. At the same time a doctor may view patients medical history even bore the patient visits him. This allows for an automated patient doctor handling system through an online interface. This system also consists of organ donor module. This module allows for organ donation registration as well as organ search. The module is designed to help urgent organ requirements through easy/instant searches.

Major Functional Requirements

- **Admin Login:** The system is under supervision of admin who manages the bookings made.
- **User login/registration:** Users have to first register themselves to login into the system.
- **Medical History:** System allows to update and view patient medical history.
- **Doctor Search:** System allows for doctor search through categories, name and location.
- **Appointment availability check:** User can click on spaces to view the availability.
- **Appointment booking online for date and time:** Users can book appointment for their required date and time.
- **Automatic cost calculation:** The system calculates the total cost incurred for parking based on the time that user has asked for booking.
- **Booking cancellation:** User may even cancel their bookings by login into the system anytime.
- **Email on appointment booking:** When user is successful in appointment confirmation and 'thank you' email regarding the slot booked.
- **Feedback:** The system has a feedback form, where user can provide feedback into the system.
- **Organ Donor Registration:** User who is going to donate organ has to register himself by filling the details.
- **Organ Donor Search:** User can search donor based on type of organ.

Next Lecture

Software Development Life Cycle Models