# Introduction to Software Engineering

Session I

Kiran waghmare

# Reference

- Stephen Schach, *Classical and Object-Oriented Software Engineering with UML and Java*, Chapter 1, McGraw-Hill, New York, USA.
  - http://www.mhhe.com/engcs/compsci/schach5/samplech.mhtml

# Software



Importance : Software can have
a huge impact
in any aspect of Society


# Introduction

**Program:**

- A program is a **set of instructions** --- -----> executes.
- A single program can have **multiple file.**
- Programs are written for **knowledge** and consists **of coding.**

**Software:**

- It is a collection of **computer programs** and related data that

**provides the instructions** for telling a computer what to do and how to do it.

- Also, software is a **conceptual entity** which is a set of computer programs, procedures and associated documentation concerned with the operation of a data processing system.
- It is a set of
- **Programs + Procedures + Operating Manuals + Its Documentation**
-
-

# Program vs. Software

| Sr No | Program | Software |
|---|---|---|
| 1 | Programs are developed by individuals. They are small in size and have limited functionality | They are large in size |
| 2 | Here Programmer is a solo user | Many users are involved in the development |
| 3 | It consists of single program or set of instructions | It consists of number of programs |
| 4 | Single interface may not be important | Each and every interface is important, carefully handled and designed. |
| 5 | It requires small or little documentation | It requires complete documentation and operating manual. |
| 6 | It can be developed by individual style | It has to follow software or system development life cycle (SDLC) |

# Software Engineering

- Software Engineering is an ***engineering discipline*** which is concerned with *all aspects of software production* from the early stages of system requirements through to maintaining the system after is has gone into use.

| Computer Scientist | Engineer | Software Engineer |
| --- | --- | --- |
| Proves **theorems** about algorithms, designs, languages, define knowledge representation schemes. | **Develops a solution** for an application-specific problem for a client. Uses computers & languages, tools, techniques and methods. | Works in **multiple application domains** |

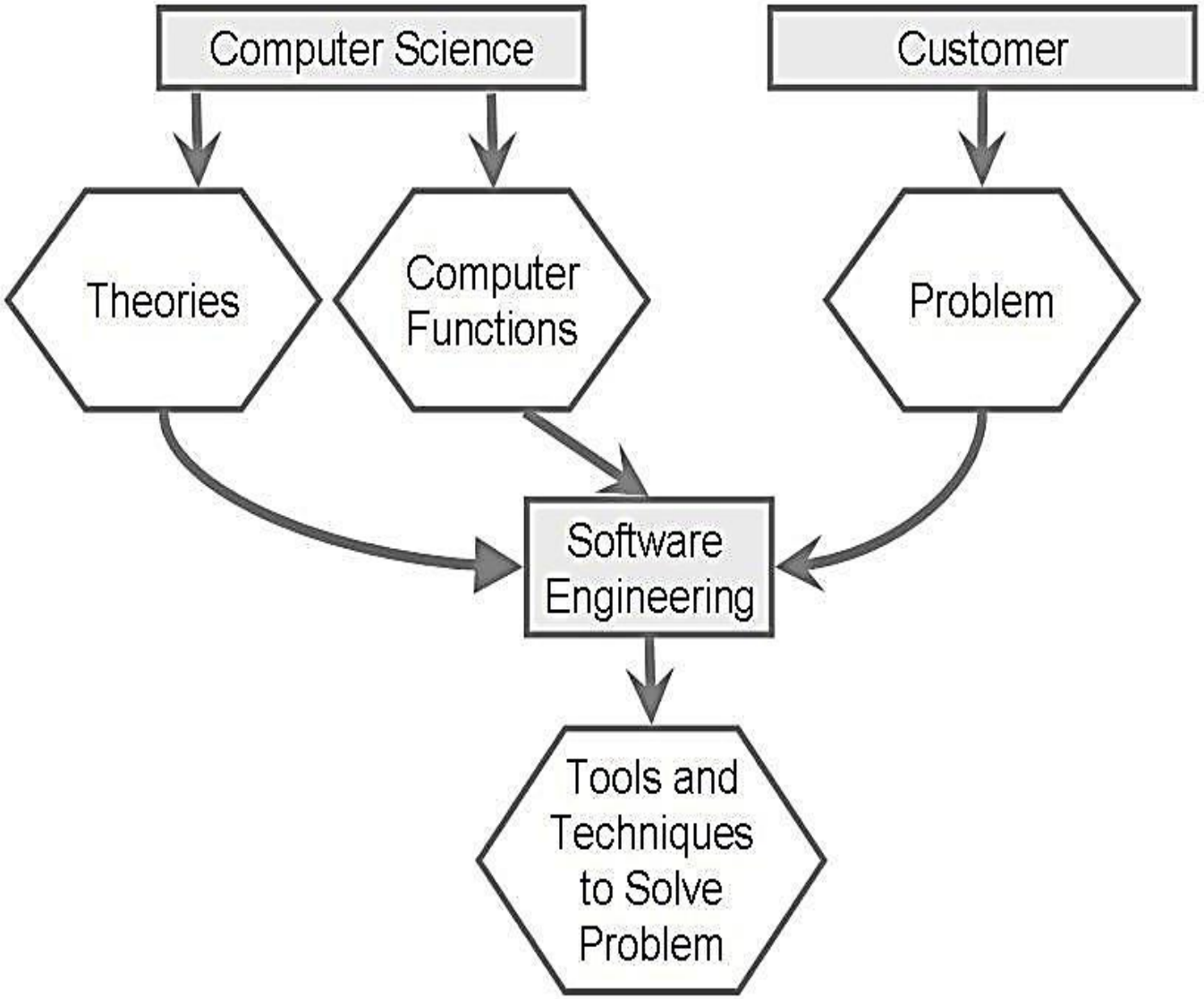



Fig. Software Engineering

# Software Engineering : Definition :

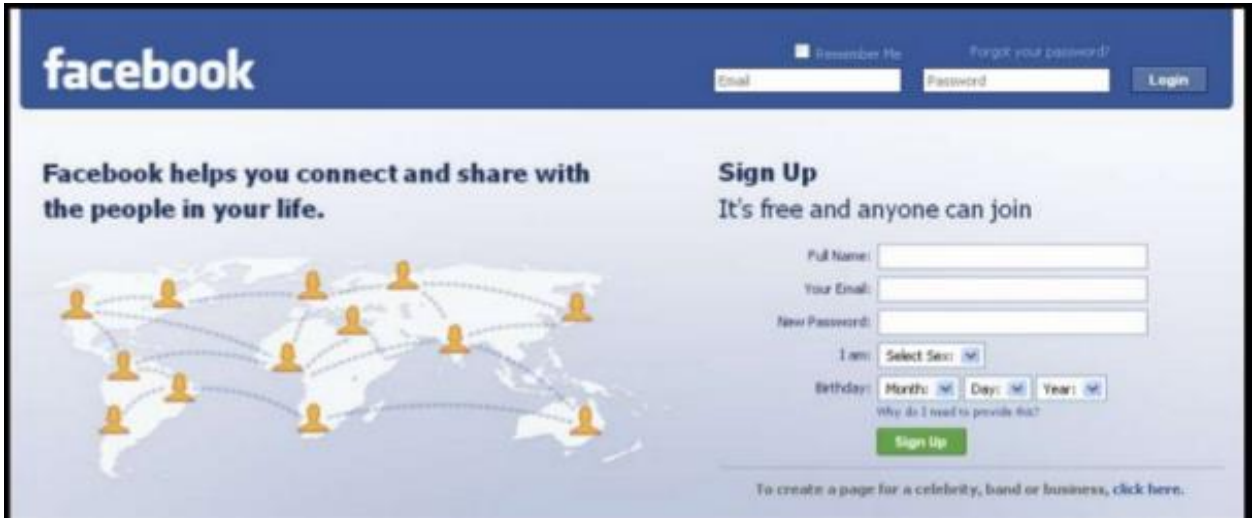
- Software Engineering is a collection of
  - **techniques, methodologies and tools** that help
    - with the **production** of
    - a **high quality** software system
    - with a given **budget**
    - before a given **deadline**
    - while **change** occurs.

- **Or**
- "Software engineering is the establishment and use of sound engineering principles in order to obtain **economically** software that is reliable and work efficiently on real machines"
- **Or**
- "Software engineering is a systematic approach to **development, operation, maintenance and retirement** of software"
- **Or**
- "Software engineering is the application of science and mathematics by which the capabilities of computer equipment are made useful to man via **computer programs, procedures and associated with documentation**."

# Where you can find software

# Some popular ones…





# And even in…

# Conclusion

- Software is Almost Everywhere…

**Problems in Software Development**

# Common Issues

- The final software **doesn't fulfill the needs of the customer.**
- **Hard to extend and improve:** if you want to add a functionality later is mission impossible.
- **Bad documentation.**
- **Bad quality**: frequent errors, hard to use,…
- **More time and costs** than expected

# Ariane 5 Flight 501

- Cause: design errors in the software
- Video: Ariane 5 Flight 510

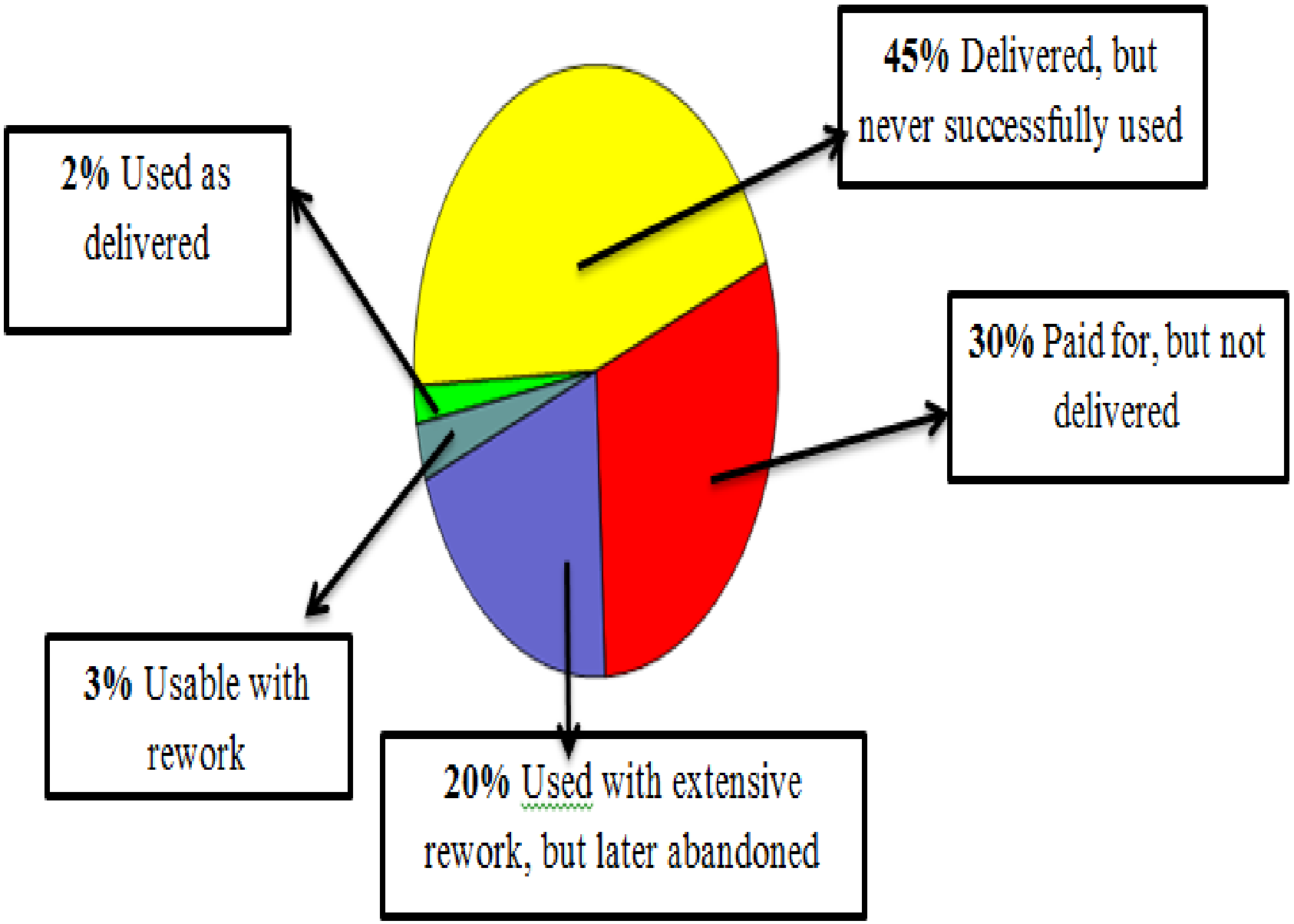
A software bug caused European Space Agency's Ariane 5 rocket to crash 40 seconds into its first flight in 1996 (***cost: half billion dollars***)

A software exception occurred during execution of a data conversion from 64-bit floating point to 16-bit signed integer value

# Mars Path Finder

A few days into its mission, NASA's Mars Pathfinder computer system started rebooting itself

- Cause: Priority inversion during preemptive priority scheduling of threads


45% Delivered, but never successfully used
2% Used as delivered
30% Paid for, but not delivered
3% Usable with rework
20% Used with extensive rework, but later abandoned

# Software Crisis:

- The software crisis means ***problem encountered*** during the software development. The software crisis has the following characteristics:
  - **Expensive delivery**
  - **Extremely late**
  - **Unsatisfactorily software system**
  - **Viability to complete**
  - **Over budget**
  - **Not according to the requirements of the user**

- **45%** Delivered, but never successfully used

- People in software industry bothered about the software crisis for last 25 years, but we have successfully avoided actual clap of doom until now.

# Reasons for Software Crisis:

- There are three main reasons occurs for the software crisis are as follows:

1. **Costs** of Software is more than hardware.
2. **Lack of communication** between software engineer and user.
3. **Growing size** and complexity of the programs.

# What is Software Engineering?

- Software engineering is a discipline whose aim is
  - the production of fault-free software,
  - that is delivered on time,
  - within budget, and
  - satisfies the user's needs.

# Software Engineering is a Layered Technology

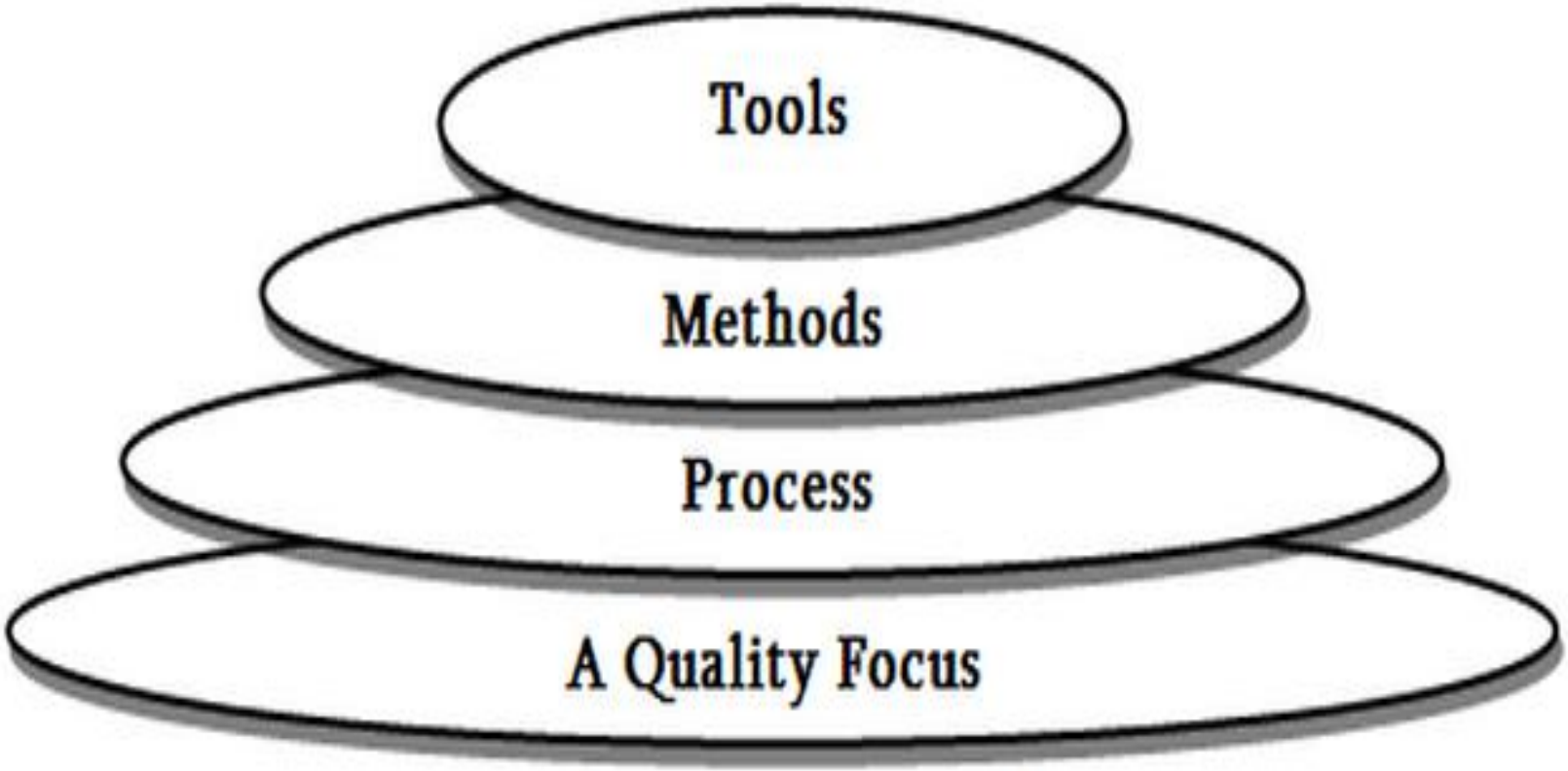



Figure: Flowchart of the Layers of Software Development

# Quality, Process, Methods, and Tools

- **A quality focus:**
  - **Continuous process** improvement
  - **Bedrock** that supports software engineering

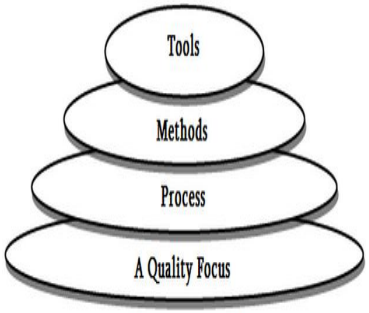


Figure: Flowchart of the Layers of Software Development

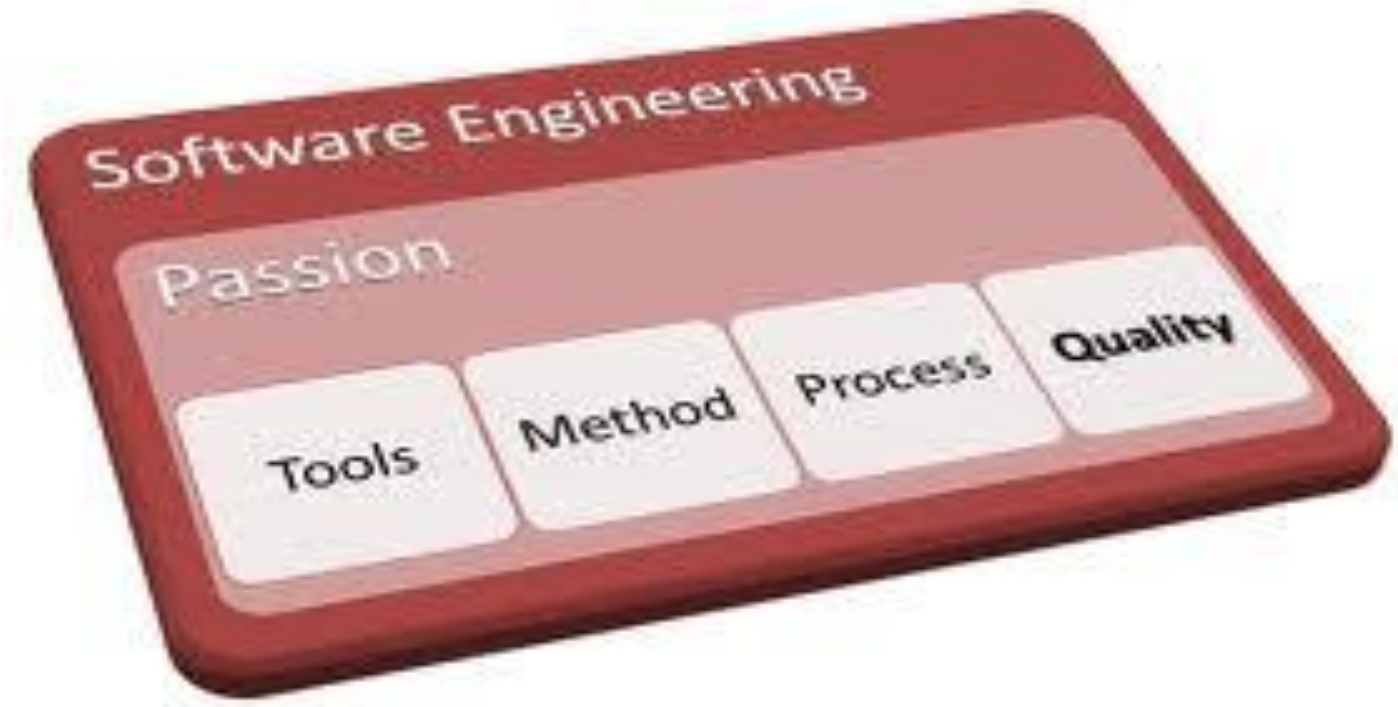
- **Process**
  - **Provides the glue** that holds the layers together;
  - Enables **rational and timely development;**
  - **Provides a framework** for effective delivery of technology;
  - Forms **the basis for management;**
  - **Provides the context** for technical methods, work products, milestones, quality measures, and change management

- **Methods**
  - **Provide the technical** "how to" for building software;
  - Rely on a **set of basic principles**;
  - **Encompass a broad array** of tasks;
  - Include **modeling activities**
- **Tools**
  - **Provide automated or semi-automated** support for the process and methods (i.e., CASE tools)

- **Several techniques have been suggested to help solve the software crisis.**
  - **~1975-1985:**
  - **Structured Paradigm**
    - **Structured Systems** Analysis, Composite/Structured Design, Structured Programming, Structured Testing
    - Lead to **major improvements** for software industry.
    - But only **good for small programs** (say, 5,000-50,000 lines of codes)
    - **Not scale up** well with today larger programs (say, 500,000-5,000,000 LOC)
    - **Not so good in software maintenance aspects,** (for instance, because of the separation of action-oriented and data-oriented in structured paradigm).
  - **Object-Oriented Paradigm**
    - An object is a **unified software component** that incorporates both data and actions that operate of those data. --> More Promising!

# The Software Process

Kiran waghmare



# Software Engineering

- Software engineering is a discipline whose aim is
  - the production of **fault-free** software,
  - that is **delivered on time**,
  - **within budget**, and
  - **satisfies** the user's needs.

# Software

- Software consists of **not just code** in machine-readable **form but also all the documentation** that is an intrinsic component of every project.
  - The **specification** document
  - The **design** document
  - **Legal and accounting** documents of all kinds
  - The software **project management plan** and other management documents
  - All types of **manuals.**

# What is a Process?

- (Webster)
  - A system of operations in producing something;
  - a series of actions, changes, or functions that achieve an end or a result

- (IEEE)
  - A sequence of steps performed for a given purpose

# The Software Process

- **A structured set of activities required to develop a software system**
  - Specification
  - Analysis, design and implementation.
  - Validation
  - Evolution

- **A software process model is**
  - **an abstract representation of a process**
  - it presents a description of a process from some particular perspective

# Client Developer and User

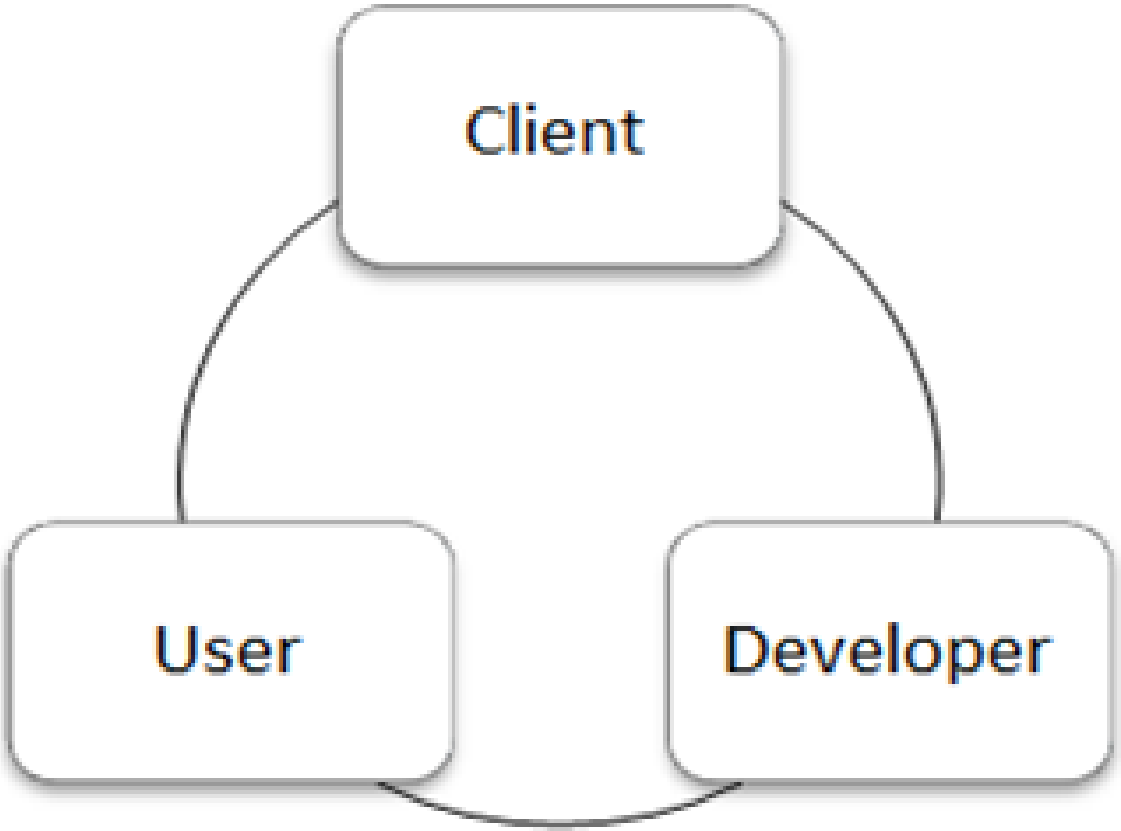



Fig. Relationship of Client Developer and User

# Client Developer & User

- Client
  - Individual or organization that wants a **product to be developed.**
- Developer
  - Are the members of the organization **responsible for building software.**
- User
  - Person on whose behalf the client has commissioned the product & **who will utilize the software.**

# Three Types of Software: Based on the Functionality

- **Custom software:**
  - It is written for **one client**.
- **Commercial off-the-shelf (COTS) software:**
  - It has **multiple copies and the copies are sold** at much lower prices to a large number of buyers.
  - It is developed for "the market".
  - That is, there are no specific clients or users until the software has been developed and is available for purchase.
  - Shrink-wrapped software, Clickware
- **Open-source software:**
  - It is developed and **maintained by a team of volunteers and may be downloaded** and used free of charge by anyone.