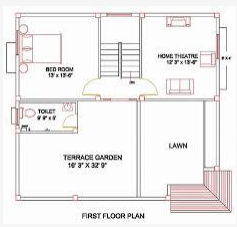
CHAPTER 1
INTRODUCTION TO
SOFTWARE &
SOFTWARE ENGINEERING

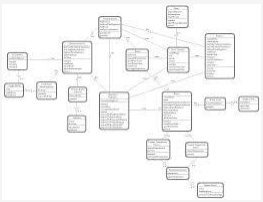
Software Engineering

Engineering



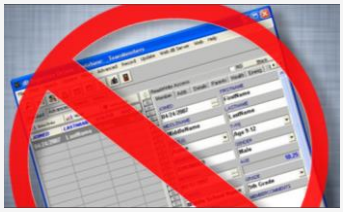
Design | Build | Product

Software Engineering

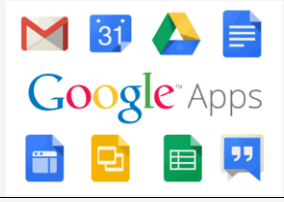


Software is dead…..!

- The **old School view of Software**
  - You buy it
  - You own it &
  - It's your job to manage it
  - That is coming to an end



- Because of **web 2.0** & extensive **computing power**, there is a different generation of software
  - It is delivered via Internet
  - It looks exactly like it's residing on each user's computing device
  - Actually it reside on far away server



3

What is Software?

- Software is
  1) **Computer program** that when executed provide desired features, function & performance
  2) **Data Structure** that enable programs to easily manipulate information
  3) **Descriptive information** in both hard and soft copy that describes the operation and use of programs

Computer Program

\+

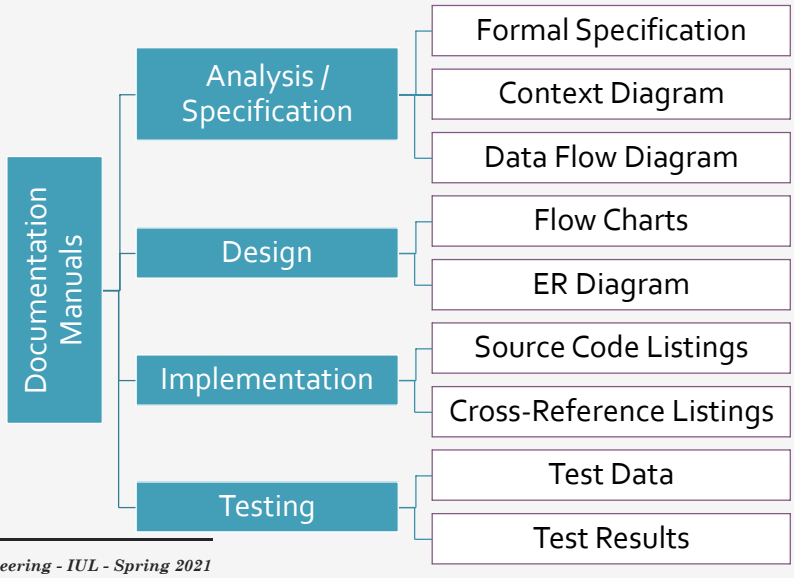
Data Structure

\+

Documents Soft & Hard

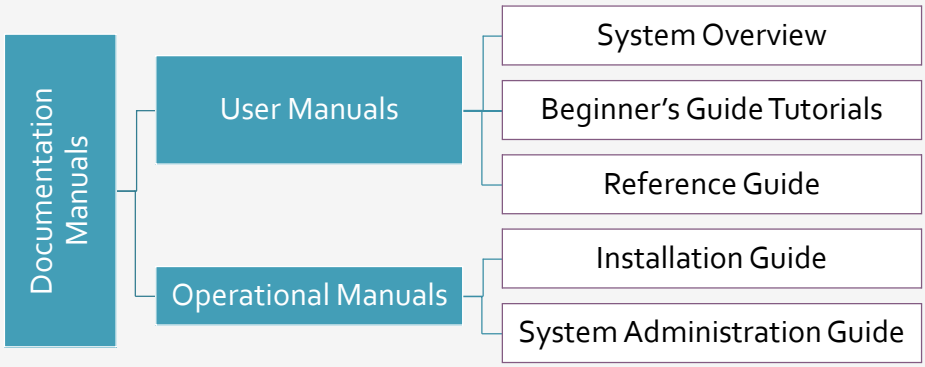
4

List of Documentation Manuals



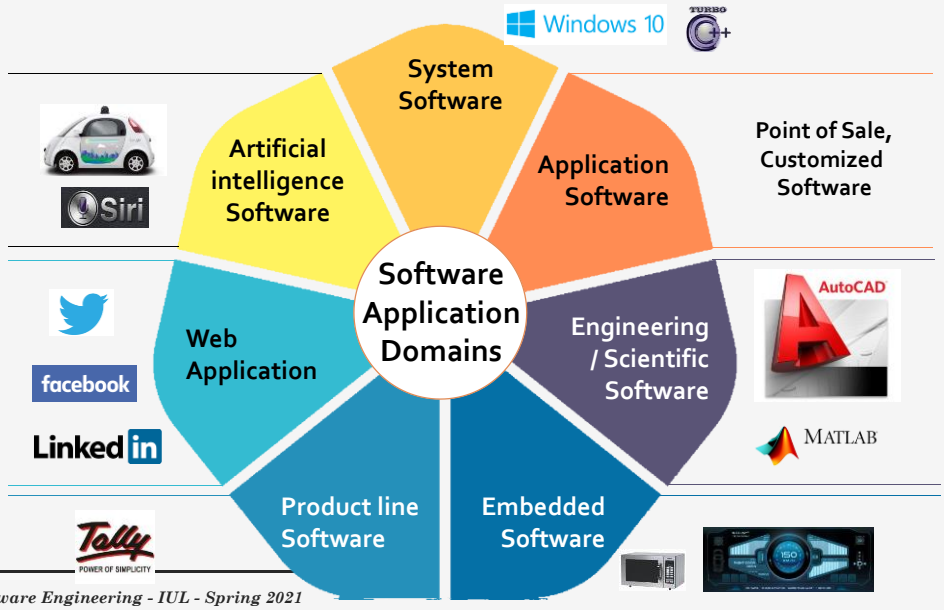
5

List of Documentation Manuals



6

Software Application Domains



Software Engineering

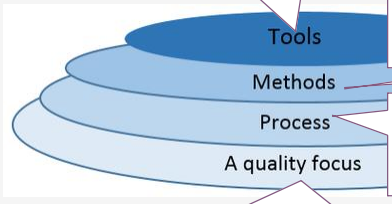
Software engineering is the establishment and use of **sound engineering principles** in order to obtain **economically software** that is **reliable and works** efficiently in **real machines**.

Software Engineering is the science and art of building (designing and writing programs) a software systems that are:

1) on time
2) on budget
3) with acceptable performance
4) with correct operation

Software Engineering Layered Approach

Software Engineering Tools **allows automation of activities** which helps to perform systematic activities. A system for the support of software development, called **computer-aided software engineering** (CASE). **Examples:** Testing Tools, Bug/Issue Tracking Tools etc…



It provides **technical how-to's** for building software, it encompasses **many tasks** including communication, requirement analysis, design modeling, program construction, testing and support

It is a foundation of Software Engineering, It is the **glue** the holds the **technology layers**, It **defines** a **framework activities**

Defines continuous **process improvement principles**

Software Engineering Cont.

- Software Engineering is a layered technology
- **Quality**
  - Main principle of Software Engineering is Quality Focus.
  - An engineering approach must have a **focus on quality**.
  - Total Quality Management **(TQM), Six Sigma, ISO** 9001, ISO 9000-3, CAPABILITY MATURITY MODEL **(CMM), CMMI** & similar approaches encourages a continuous process improvement culture
- **Process layer**
  - It is a foundation of Software Engineering
  - It is the glue the holds the technology layers
  - It **defines** a **framework** with activities for effective delivery of software engineering technology

Software Engineering Cont.

- **Method**
  - It provides **technical how-to's** for building software
  - It **encompasses many tasks** including communication, requirement analysis, design modeling, program construction, testing and support
- **Tools**
  - Computer-aided software engineering (**CASE**) is the scientific application of a **set of tools** and **methods** to a software system which is meant to **result in high-quality, defect-free, and maintainable software products**.
  - CASE tools automate many of the activities involved in various life cycle phases.

Software Process

- A **process** is a collection of **activities, actions** and **tasks** that are performed when some work product is to be created
- A process is not a **rigid prescription** for how to build the software
- Rather it is **adaptable approach** that enables the people doing the work to **pick** and **choose** the **appropriate set of work actions** and tasks
- The **purpose** of software process is
  - to **deliver** software in **timely** manner and
  - within sufficient **quality to satisfy** those
    - Who has given proposal for software development and
    - Those who will use software
  - A **process framework** establishes the foundation for complete software engineering process, it encompasses five activities

Process Framework Activities

Communication 	Communication with Customers / stockholders to understand project requirements for defining software features	**Planning** 	Software Project Plan which defines workflow that is to follow. It describes technical task, risks, resources, product to be produced & work schedule
Modeling 	Creating models to understand requirements and shows design of software to achieve requirements	**Construction** 	Code Generation (manual or automated) & Testing (to uncover errors in the code)
Deployment 	Deliver Software to Customer Collect feedback from customer based on evaluation Software Support		

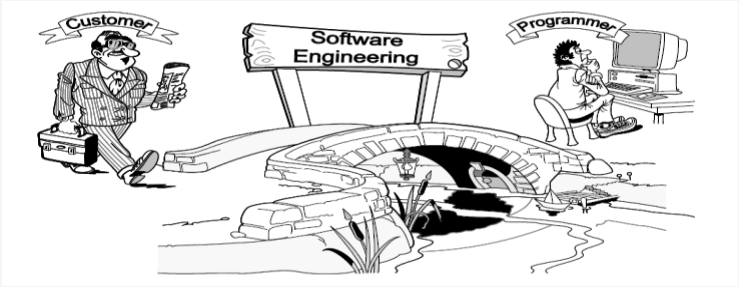
Software Process

- A **process** is a collection of **activities, actions and tasks** that are performed when some work product is to be created.
- **Each of these** activities, actions & tasks **reside within** a **framework** or model
- Figure in next slide represents "The Software Process"
- Each framework **activity is populated** by **set of** software engineering **actions**
- Each software engineering **action is defined by** a **task set** that identifies work to be completed, product to be produced, quality assurance points & milestones to indicate progress

Software Engineering

The purpose of software engineering is to develop software-based systems that let customers achieve business goals. The customer may be a hospital manager who needs patient-record software to be used by secretaries in doctors' offices; or, a manufacturing manager who needs software to coordinate multiple parallel production activities that feed into a final assembly stage.

Software engineer must understand the customer's business needs and design software to help meet them.



15

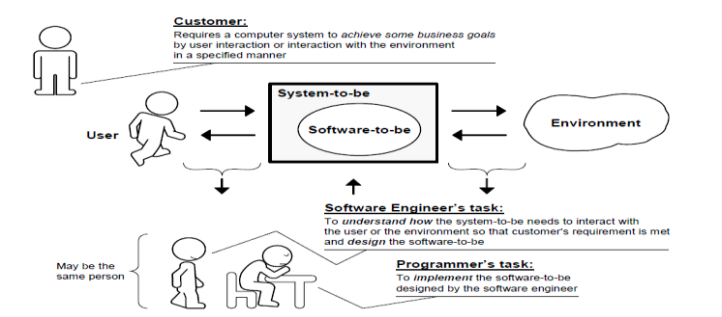
This task requires

The ability to quickly learn new and diverse disciplines and business processes

The ability to communicate with domain experts, extract an abstract model of the problem from a stream of information provided in discipline-specific jargon, and formulate a solution that makes sense in the context of customer's business.

The ability to design a software system that will realize the proposed solution and gracefully evolve with the evolving business needs for many years in the future.

The role for software engineering



16

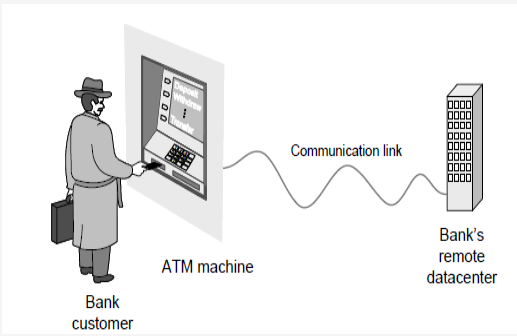
Software engineering is often confused with programming. Software engineering is the creative activity of understanding the business problem, coming up with an idea for solution, and designing the "blueprints" of the solution.

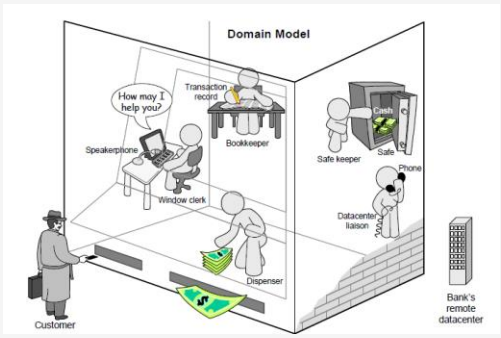
Programming is the craft of implementing the given blueprints.

Software engineer's focus is on *understanding* the interaction between the system-to-be and its users and the environment, and *designing* the software-to-be based on this understanding. Unlike this, programmer's focus is on the program code and ensuring that the code faithfully implements the given design.

This is not a one-way process, because sometimes the designs provided by the "artist" (software engineer) cannot be "carved" in "marble" (programming infrastructure) as given, and the "craftsman" (programmer) needs to work closely with the designer to find a workable solution. In an ideal world, both activities would be done by the same person to ensure the best result; in reality, given their different nature and demands, software engineering and programming are often done by different people.

Example: developing software for an Automatic Teller Machine (ATM).





Dynamic interactions of the imagined components during task accomplishment