# City University

## SE 415: Software Engineering

### Lecture 1 & 2

Supta Richard Philip

supta.philip@gmail.com

---

# Chapter 1: The Nature of Software

## Definition of Software

Software is:

- (1) instructions (computer programs) that when executed provide de- sired features, function, and performance;
- (2) data structures that enable the pro- grams to adequately manipulate information, and
- (3)descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.

## Software Application Domains

There are seven broad categories of computer software and continuing challenges for software engineers:

1. System software
2. Application software

3. Engineering/scientific software
4. Embedded software
5. Product-line software
6. Web/Mobile applications
7. Artificial intelligence software

## Legacy Software

The older programs(software)—often referred to as legacy softwar is very difficult to maintenance and changes.

**Four broad categories of software are evolving to dominate the industry.**

- **WebApps**

> *WebApps have evolved into sophisticated computing tools that not only provide stand-alone function to the end user, but also have been integrated with corporate databases and business applications.*

- **Mobile Applications**

> *The term app has evolved to connote software that has been specifically de- signed to reside on a mobile platform (e.g., iOS, Android, or Windows Mobile).*

- **Cloud Computing**

> *Cloud computing encompasses an infrastructure or “ecosystem”*

> *that enables any user, anywhere, to use a computing device to share computing resources on a broad scale.*

# Chapter 2: Software Engineering

## Definition of Software Engineering

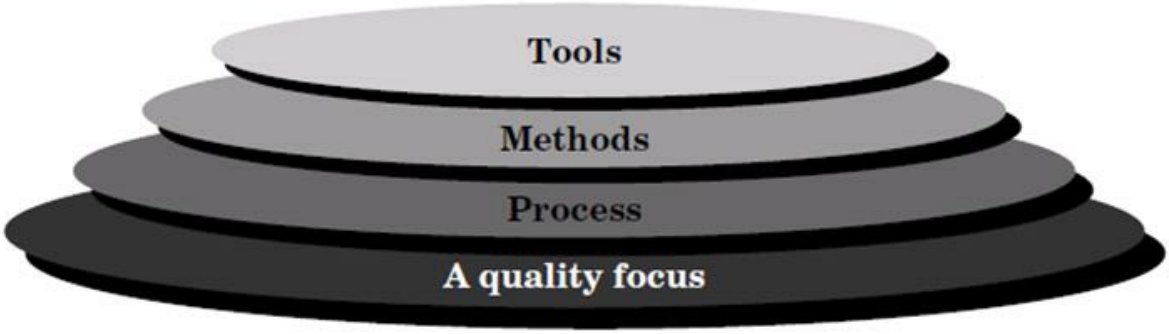
Software Engineering :

(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

(2) The study of approaches as in (1).

## software engineering layers

- Software engineering is a layered technology.
- Software engineering must rest on an organizational commitment to quality.
- The foundation for software engineering is the process layer.
- Process defines a framework that must be established for effective delivery of software engineering technology.
- Software engineering methods provide the technical how-to’s for building software; a broad array of tasks that include communication, requirements analysis, design modeling, program construction, testing, and support.
- Software engineering tools provide automated or semi-automated support for the process and the methods.




## software Engineering Process

- A process is a collection of activities, actions, and tasks that are performed when some work product is to be created.

> *The Process Framework*

- A process framework establishes the foundation for a complete software engineering process by identifying a small number of framework activities

  > *Communication*
  > *Planning*
  > *Modeling*
  > *Construction*
  > *Deployment*

## Software Engineering Umbrella Activities

- Software engineering process framework activities are complemented by a number of umbrella activities.
- In general, umbrella activities are applied throughout a software

project and help a software team manage and control progress, quality, change, and risk.

> *Software project tracking and control*
>
> *Risk management*
>
> *Software quality assurance*
>
> *Technical reviews*
>
> *Measurement*
>
> *Software configuration management*
>
> *Reusability management*
>
> *Work product preparation and production*

## Software Engineering Pratice

1. Understand the problem (communication and analysis).
2. Plan a solution (modeling and software design).
3. Carry out the plan (code generation).
4. Examine the result for accuracy (testing and quality assurance).

## Software Engineering General Principles

- The First Principle: The Reason It All Exists
- The Second Principle: KISS (Keep It Simple, Stupid!)
- The Third Principle: Maintain the Vision
- The Fourth Principle: What You Produce, Others Will Consume
- The Fifth Principle: Be Open to the Future
- The Sixth Principle: Plan Ahead for Reuse
- The Seventh Principle: Think!

## Referrence

Software Engineering A Practitioner’s Approch, Roger S. Pressman, Bruce R. Maxim, Eight Edition