

# CP317-Software Engineering

Introduction to software engineering – week 1 -1

Shaun Gao, Ph.D., P.Eng.

# Agenda

- Overview of the course
- Software engineering
  - Concept
- The process of software engineering
  - Requirement gathering
  - System design – high-level design
  - Software requirement specification – low-level design
  - Development – programming/coding
  - Testing
  - Deployment
  - Maintenance
- Summary

# Overview of the course

- Course Name: CP317B: Software Engineering
- Instructor: Lunshan (Shaun) Gao, Ph.D., P.Eng.
- Email: lgao@wlu.ca
- Textbook (reference):
  - Beginning software engineering by Rod Stephens; Publisher: John Wiley & Sons, 2015
  - Software Engineering: A Practitioner's Approach by R.S. Pressman, 2010
- Evaluation:

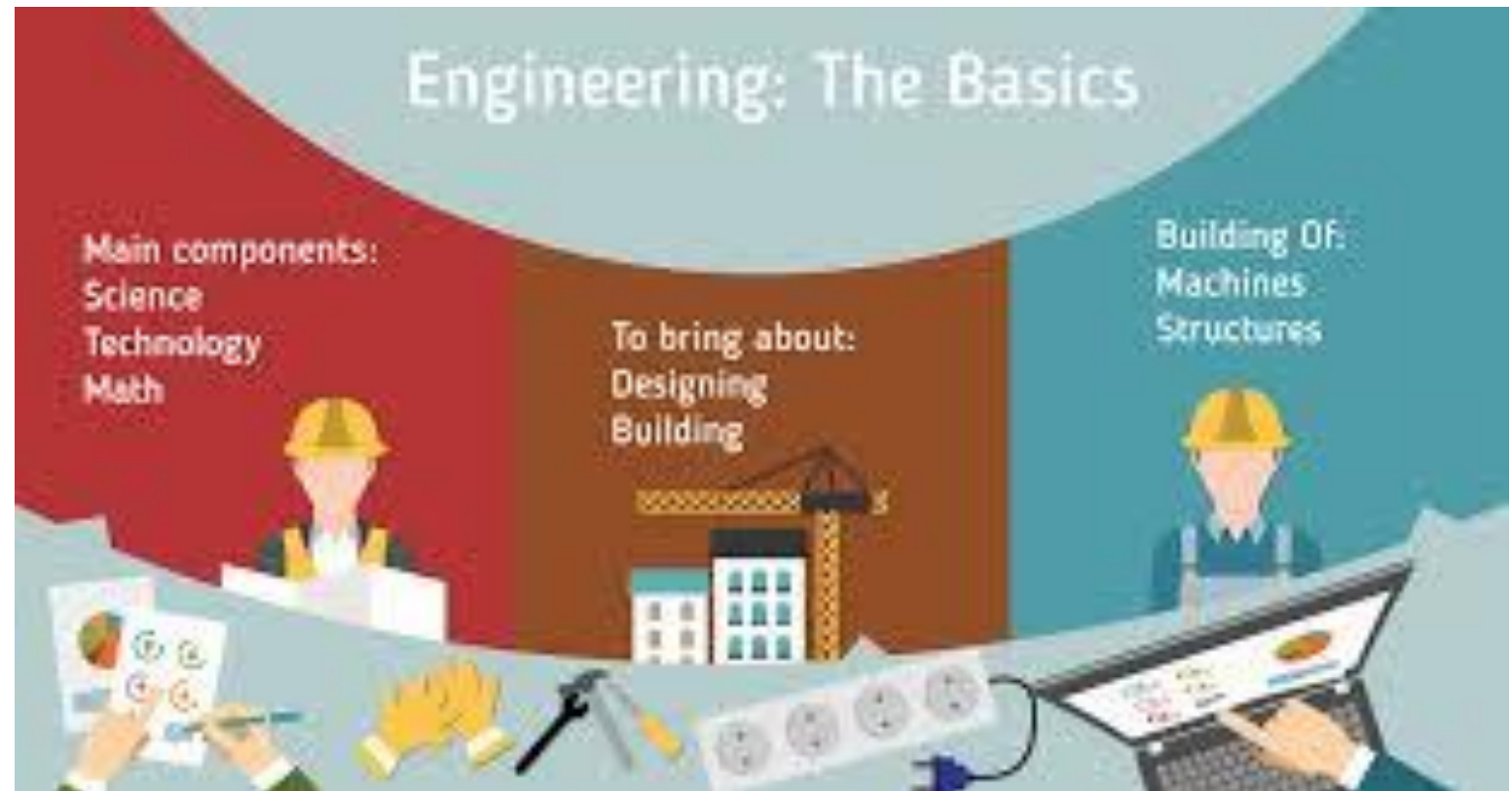
• Group project (2x15%)	30% (presentation + project report)
• Test 2x15%	30% close book
• Final exam	40% close book
- Rule 1: Academic plagiarism is prohibited. Project reports and materials for presentation must be delivered to MyLearningSpace.
- Rule 2: don't ask for increasing marks. Grade is earned not given.

# Overview of the course

- Week1: introduction to software engineering and project management
- Week2: Requirement gathering
- Week3: High-level design – architecture design
- Week4: Low-level design-1
- Week5: Low-level design-2 ----- test 1
- Week6: Development
- Week7: Testing and metrics
- Week8: Deployment and waterfall model
- Week9: Iterative models-1 ----- test2
- Week10: Iterative models-2 and RAD
- Week11-12: Project presentation

# Introduction

- Software engineering
- Engineering



# Introduction – cont.

- Engineering
  - Definition 1: Engineering is the study using scientific principles to **design and build** machines, structures, and other things, including bridges, roads, vehicles, and buildings: civil/electrical/mechanical engineering
  - Definition 2: Engineering is the branch of science and technology concerned with the **design, building**, and use of engines, machines, and structures.
- Question – **Think-Pair-Share** (TPS):
  - Engineering vs. Science

# Introduction – cont.

- Software engineering
  - IEEE definition of software engineering:
    - The study of approaches as in the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
  - **Fritz Bauer, a German computer scientist**, defines software engineering as:
    - “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.”
- Definition 1: Software engineering is the study using scientific principles to **design and build** software products for computer systems.
- Definition 2: Software engineering is an engineering branch associated with **designing and building** software products using scientific principles, methods, and procedures. The outcome of software engineering is an efficient and reliable software product.

# Introduction – cont.

- Differences between software engineering and other types of engineering

- The final products are intangible
- Short history
- No standard process



Intangible  
Product



No Standard  
Software  
Process

- SE vs. Computer engineering (CE)
  - CE deals with designing, developing, and operating computer systems. In contrast to SE, it **emphasizes solving problems in digital hardware and at the hardware-software interface.**

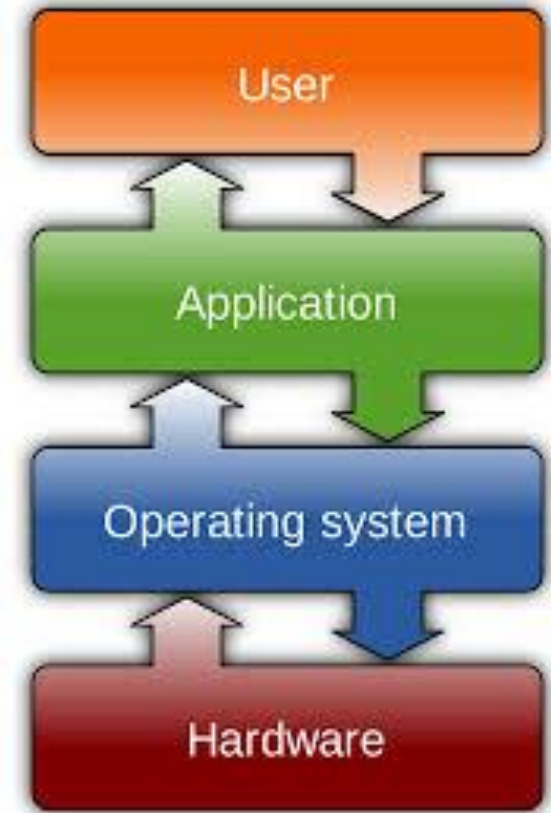


# Introduction – cont.

- Software products
  - Definition 1: A software product is **merchandise** consisting of a computer program that is offered for sale.
  - Definition 2: A software product is **a software system** delivered to the customer with the **documentation** that describes how to install and use the system. In certain cases, software products may be part of system products where hardware, as well as software, is delivered to a customer.

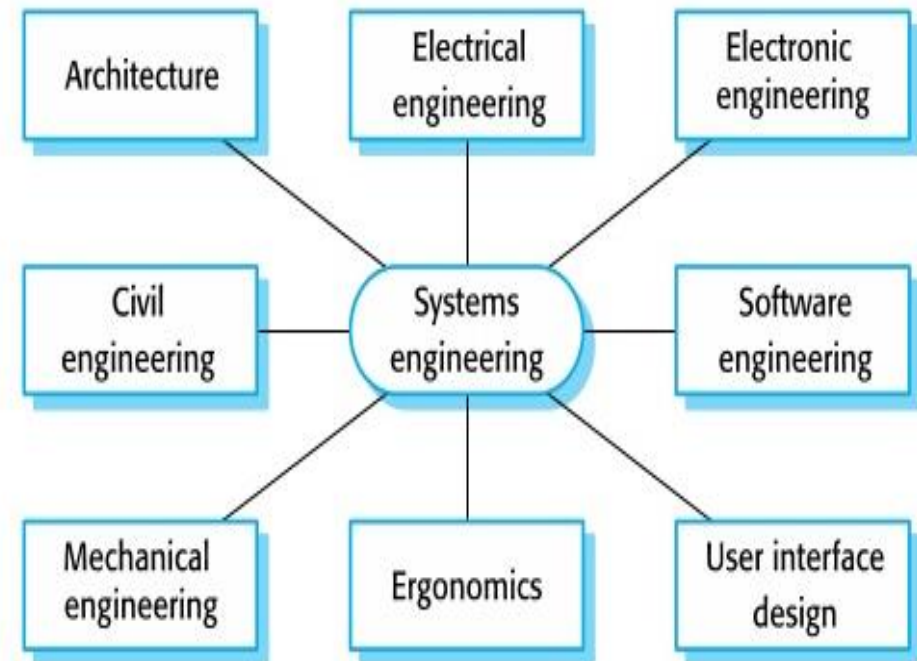
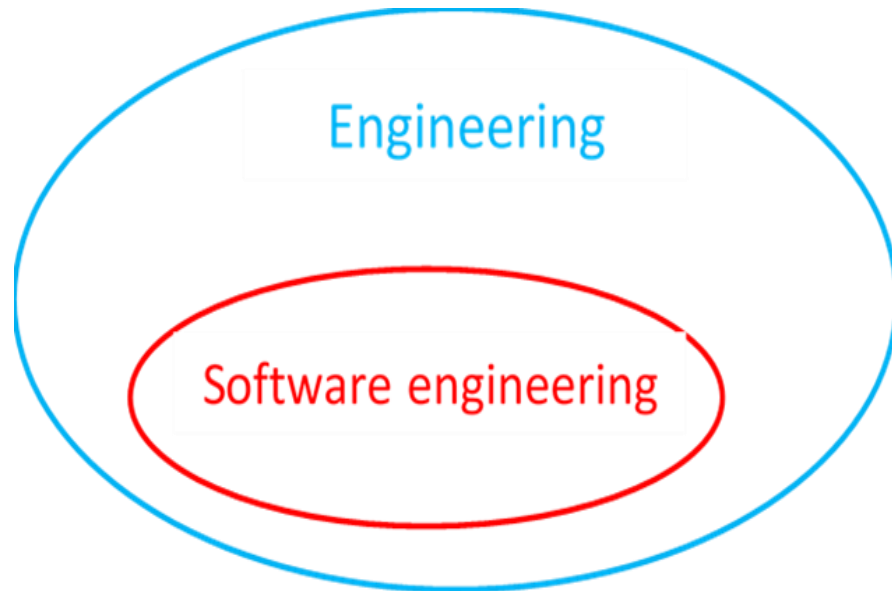
# Introduction – cont.

- Types of Software products
  - Application software – web application, user interface software
  - Middle ware – database, relational database management system
  - Embedded software – software that works in routers, smartphones
  - Real time embedded software – software that works in airplanes, rail train control systems, etc.



# Introduction – cont.

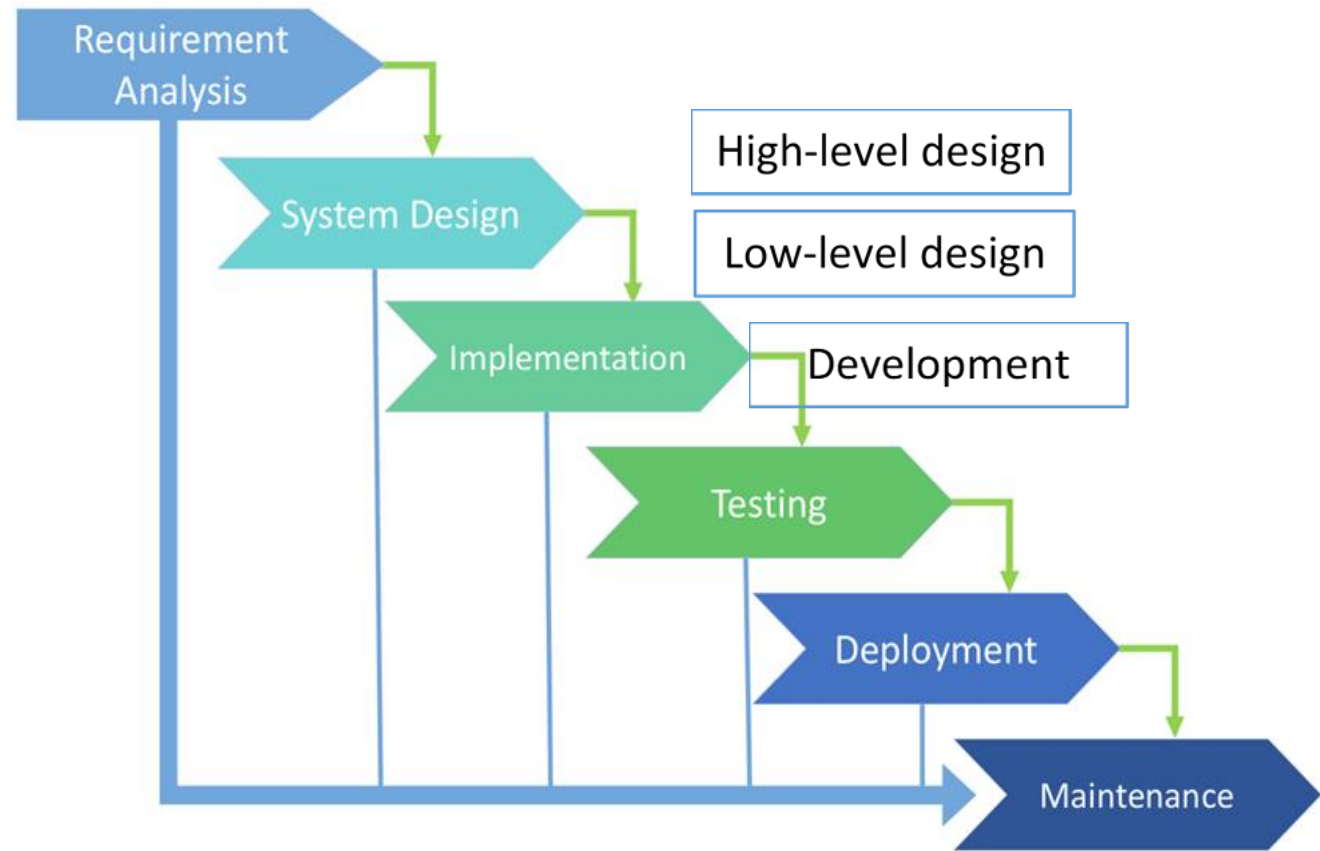
- Relationship between software engineering and other engineering



# The process of software engineering

- **Basic tasks** of software engineering

- Requirement gathering
- High-level design
- Low-level design
- Coding
- Testing
- Deployment
- Maintenance



# The process of software engineering –cont.

- Requirements: A requirement is a singular documented **physical or functional need** that a particular design, product or process aims to satisfy.
  - Understanding
  - Feasibility research
  - Negotiation
  - Requirements changes always happen
- System design
  - Requirements decomposition
  - Specification
  - Tools selection

# The process of software engineering –cont.

- Software design document
  - Data flow
  - Object-oriented design
  - UML (Unified Modeling Language)
- Implementation
  - Coding
  - Bug fixing
    - Fixing one bug often create a new bug: (a) the bug fix is incorrect; (b) the fix breaks other code that depends on the original buggy behavior; (c) the fix may change system behavior
- Testing
  - Foundation is requirements
  - Specification

# The process of software engineering –cont.

- Deployment:

- Deployment is the action of bringing resources into effective action.

- It involves the following:

- New or new version computer system
    - User training
    - On-site support while the users get familiar with the system
    - Bug fix

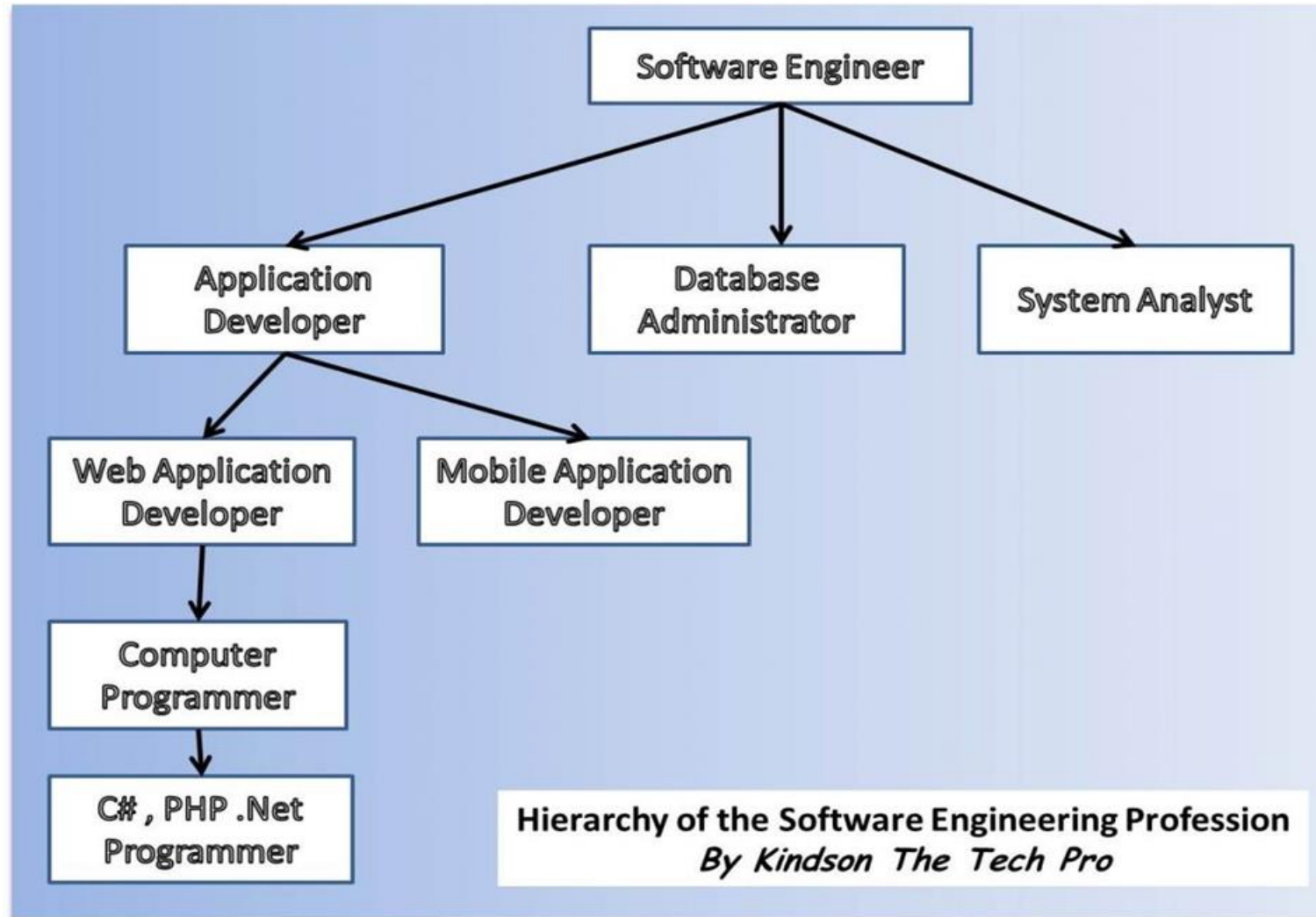
- Maintenance

- Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes.

- It involves the following:

- Customer support
    - Bug fix

# SE vs. Developer and Programmer





# SE vs. Developer and Programmer

- **Software Engineer:**

- Has a specific degree
- Some knowledge of engineering
- Capable of designing systems used by developers and programmers
- Lead developers and teams

- **Programmer:**

- Write code
- Understand an algorithm
- Follows specifications

- **Developer/Analyst:**

- Super type programmer
- Gathers requirements,
- designs and implements applications
- Writes technical documentation

# Skills required for software engineers

- Computer systems (hardware and software)
- Computer languages
  - Object-oriented design
- Software testing and debugging (familiar with tools)
- Problem solving and logical thinking
- Communication skills (both written and verbal)
- Team player
- Management

# Summary

- The concept of software engineering
- Relationship between software engineering and engineering
- Basic tasks of software engineering
  - Requirement gathering
  - High-level design
  - Low-level design
  - Coding
  - Testing
  - Deployment
  - Maintenance
- Skills required for software engineers
- **Question:** different and relationship between science and engineering

# Announcement

- 2-4 students become a group for the group project
  - 2 students focus on project presentation
  - 2 students focus on project report
- Please email me your group members by end of January
- Group project requirements
- Group project evaluation rubrics