Introduction to Software Engineering

- What is software engineering?
  Using **engineering approaches** to developing software products.

- What is (conventional) engineering?
  For solving **customers' problems**,
  using **systematic approaches**,
  to produce **quantifiable high-quality** products,
  within **limited** budget and resources.

  Keywords:
  - Engineering is for **solving customers' problems**, compared to science for discovering facts and knowledge.
  - Systematic approaches means having **repeatable and widely appliable processes** to products production projects.
  - Engineering always produces products with **highest quality** which should be **measurable by numerical metrics**.
  - Engineering produces products with **highest-possible** quality based on given limited (**not unlimited**) budget and resources.

- What is software engineering as a discipline of engineering?
  For solving customers' problems,
  using systematic approaches,
  to **develop** quantifiable high-quality **software** products,
  within limited budget and resources.

- What are differences between software engineering and conventional (hardware-based) engineering?
  - **Complexity**
    Software products are often larger and more complex than hardware products.
    Lines of code vs. number of hardware parts.
    Relationships between lines of code vs. relationships between hardware parts.

  - **Changeability**
    Software is often requested/required to change and can be changed during and after development.
    Hardware products cannot be changed after production and at later stage of production.
    Dealing with frequent changes is a major challenge and activity in software engineering, which is not in conventional engineering

  - **Conformability**
    Software products are often required to be used in variety of environments, such as different computer capacities, different devices/platforms, and different physical environments.
    Hardware products are usually used in pre-determined, specific physical environments.

  - **Intangibility**

All hardware products are tangible. Production stages of hardware products can be physically observed. All hardware products follow laws of physics and continuous math.
Software products are not tangible. Progress of software development cannot be physically observed, e.g. % of code and/or documents completed may not reflect progress until code is working. All software products do not follow laws of physics and are based on discrete math.

- Why software engineering?
  - (Ad-hoc) craft (doghouse) vs. engineering (large building) software products
  - Software development is a creative activity (innovation, computer science) vs. disciplined activity (handbook/standard, software engineering)
  - Size and complexity of software products
  - Programming in small (single/few modules, single person development) vs. programming in large (multiple modules, team development)

- Aspects of software engineering
  - Software development phases / major activities
    - Requirements engineering
      Figure out customer's needs, by business or system analysts.
      Produce software requirements specification, or user stories.
    - Design
      - High-level design: software architecture design
        Based on overall functional and non-functional (quality) requirements, produce architecture design or blueprint of the software system, by software architect.
      - Detailed design (e.g. detailed class design)
        Implement or realize architectural components and their relations, as well as detailed requirements in detailed software modules, by software designers.
    - Implementation
      Implement or code detailed design in programming language(s).
      Produce source code of the software system, by programmers.
    - Test
      Test implemented software systems using test cases, by software testers.
    - Maintenance and evolution (after release)
      Fix errors
      Improve quality
      Modify system functionality
      Add new functionality
      Reengineering
  - Software process models
    How software development phases or major activities are organized and scheduled.

Different software process models describes different ways of organization and scheduling.
Development process of a software system is based on a selected software process model.
Software life cycle = software development process (requirements, design, implementation, test) + maintenance/evolution.

- Software development project management
  - Project management: planning, scheduling, progress monitoring, development control, quality assurance, risk management, etc.
  - Software configuration management and version control.

- Software metrics (numerical)
  - Software quality metrics
  - Development quality metrics

- Textbook contents