Software Engineering Overview

Inst. Nguyễn Minh Huy

Contents

- Basic concepts.
- Software quality.
- Brief history.
- Software engineering career.

Contents

- **Basic concepts.**
- Software quality.
- Brief history.
- Software engineering career.

Basic concepts

- What is software?
  - Software vs. Program?
    - Software = Program**S** + Data + Documents.
  - Software vs. Hardware?
    - Control hardware.
    - Flexibility:
      - Adapt to changes.
      - Can be updated.
    - Ubiquitous computing.

Basic concepts

- Types of software:
  - Users: general, specific.
  - Purposes: utility, business, game.
  - Hardware access level: system, tool, app.
  - Computer devices: server, desktop, mobile.
  - Internet: web, desktop.

Basic concepts

- What is software engineering?
  - Professional vs. Amateur.
  - Engineering = Scientific methods + Making products.
  - Software engineering:
    - ➢ Apply scientific methods.
    - ➢ Professional approaches.
    - ➢ Well-organized activities.
    - ➔ To make software.
  - Software engineering vs. Computer science?
    - ➢ Learn to build vs. Build to learn.
    - ➢ Computer science goes behind software engineering!!

Basic concepts

- Software engineers – Who are you?
  - Who are trained to join the field.
  - Have knowledge and professional skills.
  - Software engineer vs.:
    - Programmer, developer.
    - Hardware engineer?
    - System engineer?
  - Social position??

Contents

- Basic concepts.
- **Software quality**.
- Brief history.
- Software engineering career.

Software quality

- How is good software?
  - As users:
    - Correctness.

      ➔ Most important.
    - Reliability:
      - Availability: ready at anytime.
      - Scalability: ready at any circumstances.
    - User-friendly.
    - Security.
    - Fault tolerance.

Software quality

- How is good software?
  - As developers:
    - Maintainability: easy to fix.
    - Extensibility: easy to update.

Contents

- Basic concepts.
- Software quality.
- **Brief history.**
- Software engineering career.

Brief history

- Stage 1: Baby…
  - 1950 – 1970.
  - Code & Fix process:
    - No planning.
    - No design.
    - Just do it, then fix errors.

  ➔ Chaotic and risky!!

Brief history

- Stage 2: Teenager…
  - NATO conference 1968.
  - Apply scientific methods:
    - Discipline.
    - Professional approaches.
    - Well-organized activities.
  - Software process:
    - Well-defined step-by-steps.
    - Predictable.
    - Technical documents.

Brief history

- Stage 3: Grown-up…
  - User needs:
    - ➢ More complex.
    - ➢ Frequently change.
  - Heavy-weight process.

    ➔ Rapid, iterate methods.
  - Rapid application development (RAD):
    - ➢ Split development into loops.
    - ➢ Get feedback frequently.
    - ➢ Reduce documents.
    - ➢ People oriented.

Contents

- Basic concepts.
- Software quality.
- Brief history.
- **Software engineering career.**

Software engineering career

- The big picture:
    - Standish Group, 2011 - 2015:
        - 20% FAILED!
        - 50% HAVE PROBLEMS!
        - 30% SUCCESSFUL!
    - No silver bullet…
        - Software process.
        - High level language (2nd, 3rd).
        - Object Oriented Programming.
        - 4th-generation programming language.

Software engineering career

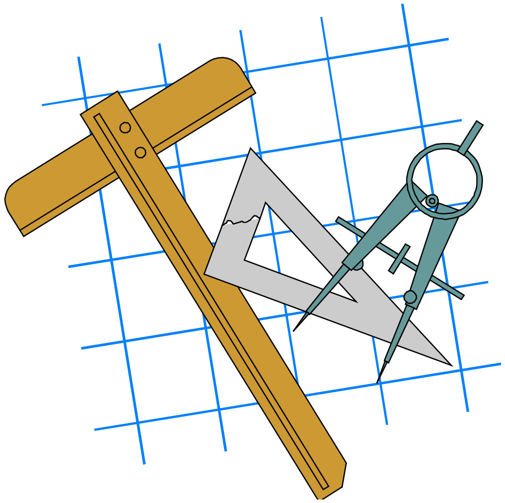
- Work characteristics:
  - Ad-hoc (often).
  - High work pressure:
    - Overtime work (OT).
    - “Worker in the office”.
  - Obsolete knowledge and technologies.
    - High competitive.
    - “No country for old men”.

Software engineering career

- What engineer needs:
  - Learning mind.
  - Responsibility.
  - Passion.
  - Copyright respect.
  - Small fish in big pond vs. Big fish in small pond.

Project

- Project preparation:
  - Registration: team members & project name.
  - Working rules: all members must follow.
  - Workspace:
    - Team website:
      - Team information: members, rules, project description.
      - Project schedule: weekly plan, report, and resources.
    - Working place.
  - Kick-off meeting.



Discussion

- Career path:

  Think about your “CAREER PATH”:

  - Draw time arrow, 4 milestones: graduate,5-year,10-year, 20-year.
  - Answer 3 questions at each milestone:
    + Still work in software engineering? If no, why not?
    + Expected work position?
    + Expected salary?
  - Describe final goals of your career path.