INTRODUCTION TO SOFTWARE ENGINEERING

Dr.Lê Hùng Tiến

VLU

Introduction To
Software Engineering

A Carnegie Mellon University
The Practical Software Engineering Series

The Professional Software Engineer

Course Objective

- Upon completion of this course, students will have the ability to:
  - Understand the discipline and principles of Software Engineering.
  - Understand the evolution of software in industry and the global competitive trends.
  - Understand software process, product and services.
  - Understand software modeling & techniques.
  - Demonstrate an appreciation for the breadth of software engineering.

Lecture Learning Objectives

- Upon completion of this lecture, student will be able to:
  - Understand what is a Software Professional.
  - Appreciate the discipline of Software Engineering.
- Outcomes:
  - To be able to describe the key areas of Software Engineering’s body of knowledge.
  - Understand the steps to become a software professional.

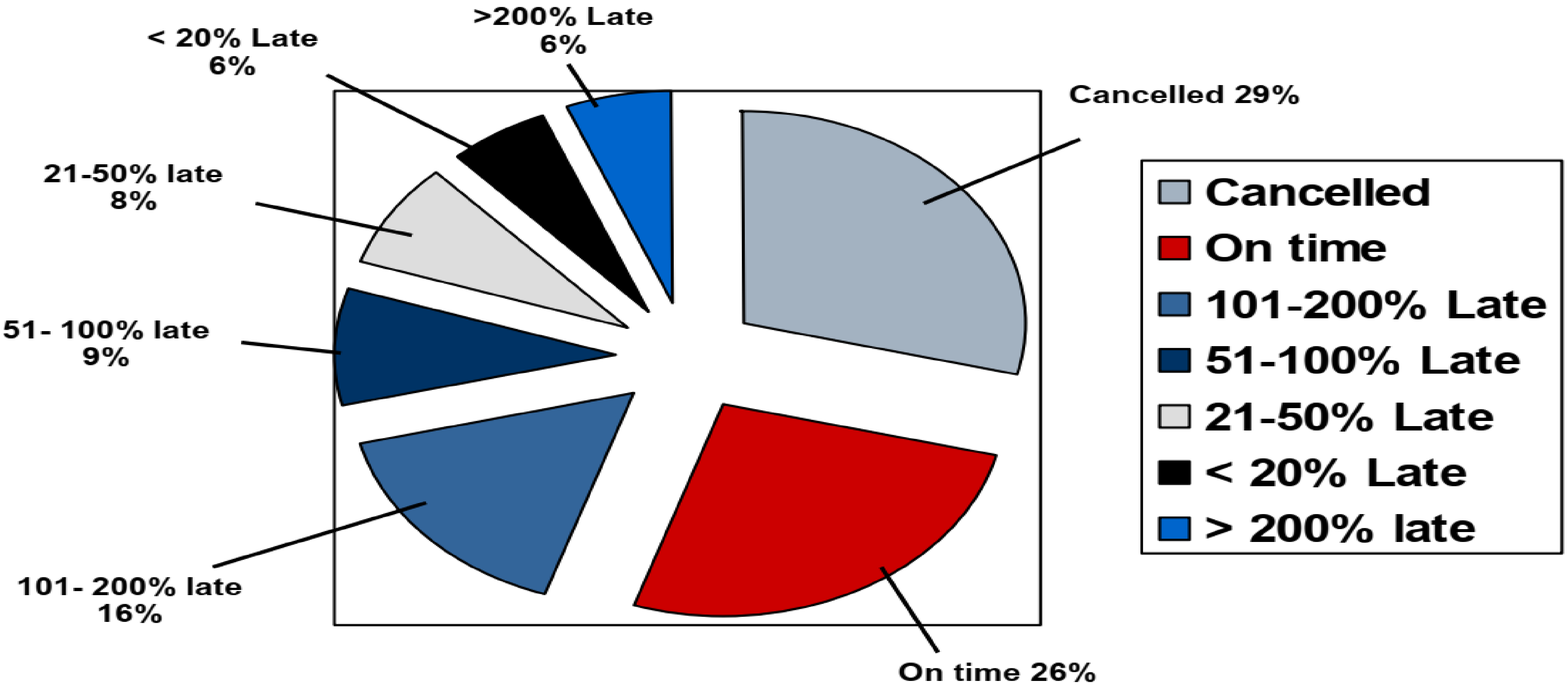
The Best Software Organizations

- The best software organizations control their projects to meet quality goals. They predict software delivery accurately and deliver software products within budget, and their productivity is constantly improving, their people are happy and their customers are highly satisfied.

- The practices needed to create good software have been well established and available but many software organizations do not use them.

- There is a large gap between the best and the average and many organizations are still using outdated techniques. To build the most sophisticated and complex software without changing their habits, programmers are still practicing what they learned in school many years ago.

Disappointing Software Project Outcomes



Source: The Standish Group's Chaos Study 2006

State Of Software Project in U.S

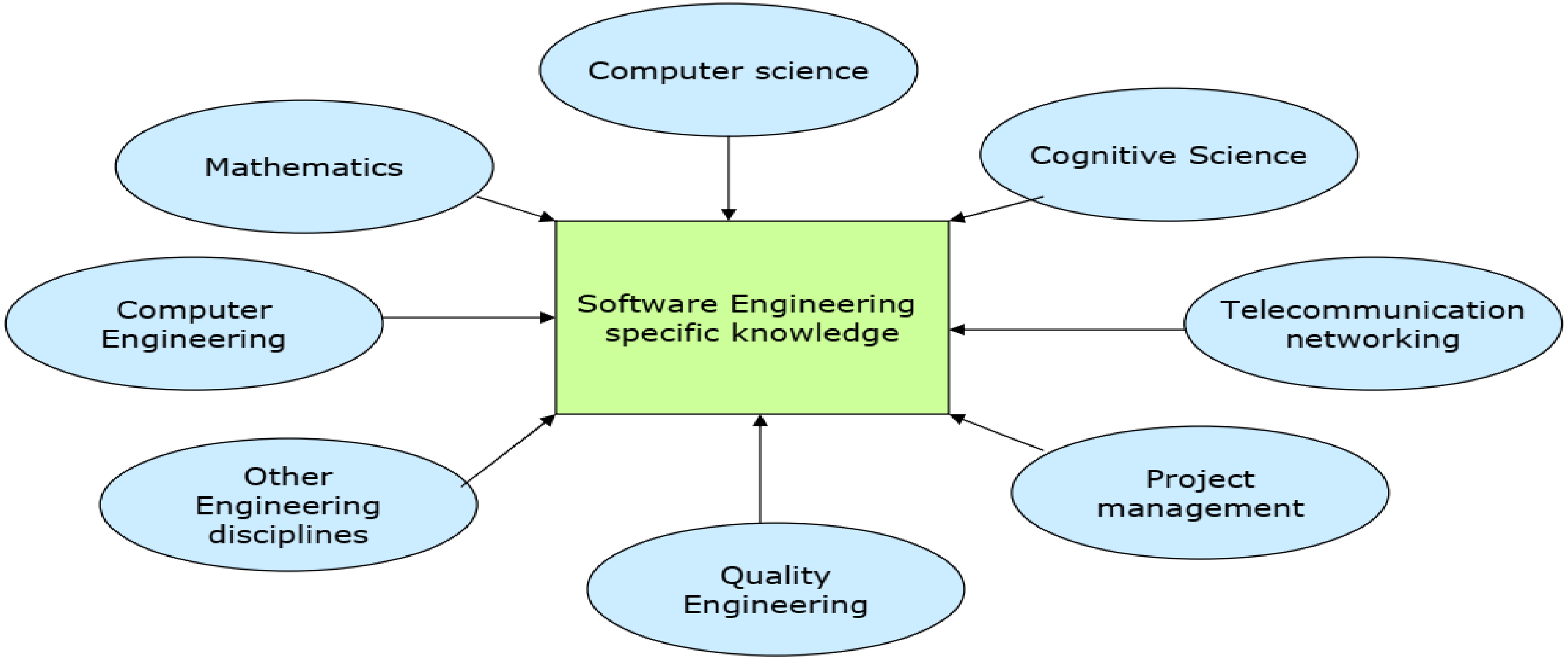
- Much of the $250 billion in annual U.S. software development spending is wasted, late, incomplete, or canceled projects:
  - 53% ($132.5 billion) are considered over budget, delayed, and less functional than planned.
  - 31% ($77.5 billion) are considered impaired and must be canceled.
  - 16% ($40 billion) are completed within budget, on time, and with all functions included.

Source: The Standish Group's chaos study (2006)

Software Engineering

- □ In 1968, NATO held the first software engineering conference and described "Software Engineering" as the body of knowledge needed to build quality software.
- □ Software Engineering has made significant knowledge since 1968 with hundred of thousands of papers on this topic. Software engineering professional groups host hundreds of conferences and workshops every year. Knowledge has been documented in thousands of pages of standards and many universities offer degrees in software engineering.
- □ Software Engineering is a discipline to create cost-effective solutions to practical problems, preferentially applying scientific knowledge to the building of software systems.
- □ The knowledge areas of a competent professional software engineer are documented in the Software Engineering Body of Knowledge (SWEBOK).

Software Engineering's Body Of Knowledge



Software Engineering’s Body Of Knowledge

The Software Engineering’s body of knowledge has identified the following areas as the core competencies:

- **Requirements Engineering**: The discovery, documentation, and analysis of the functions to be implemented in software.
- **Software Architecture**: The definition of the basic structure or the “Blueprint” of software system at the high level.
- **Software Design**: The detailed levels of software dividing into modules, interfaces, choice of algorithms.
- **Software Construction**: The implementation of the software including coding, debugging, unit testing, technical reviews and performance.
- **Software Testing**: All activities associated with executing software to detect defects and evaluate features. Testing includes test planning, test case design, and specific kind of tests from component tests, integration tests, system tests, regression tests, stress tests, acceptance tests.

Software Engineering’s Body Of Knowledge

- □ **<u>Software Maintenance</u>**: The revision and enhancement of existing software, related documentation and tests.
- □ **<u>Software Configuration Management</u>**: The identification, documentation and change control of all property generated on a software project including source code, content, requirements, designs, test materials, estimates, plans and user documentation.
- □ **<u>Software Quality</u>**: All activities associated with providing confidence that a software item conforms to technical requirements. Quality engineering includes planning, measurements, reliability, testing, technical reviews, verification and validation.
- □ **<u>Software Engineering Management</u>**: The planning, tracking and controlling of a software project.
- □ **<u>Software Tools and Methods</u>**: The development and support for the automation of software.
- □ **<u>Software Process Engineering</u>**: All activities related to improving software development quality, timeliness, and productivity.

Software Engineering’s Body Of Knowledge

- □ When the Software Engineering’s Body Of Knowledge was published, many people complained that it is a lot to expect people to learn just to write a computer program.

- □ The basis of the argument is the role of the software engineer as compared to the programmer and the curriculum of computer science and software engineering.

- □ From the aspect of Software Engineering, programming is only a subset of Software Engineering because the responsibilities of a software engineer are aimed at the purposeful creation of software that satisfies a wide range of technical, business, and regulatory requirements, not the ability to create code.

- □ The field of Software Engineering will continue to evolve as the technology changes.

What Is A Profession?

□ A profession must have:

1. A requirement for extensive learning and training.
2. A code of ethics imposing standards higher than those normally tolerated in the marketplace.
3. A disciplinary system for professionals who breach the code.
4. A primary emphasis on social responsibility strictly over individual gain and corresponding duty of its member to behave as members of a disciplined and honorable profession.
5. A prerequisite of a license prior to practicing.

A Guide Toward Software Engineering Profession

- A professional's development typically follows these steps:
  - **Basic Education**: A professional generally begins their professional life by completing a university program in their chosen field.
  - **Accreditation**: The university programs are accredited by an oversight board that determine whether the program provides adequate education. The Accreditation Board for Engineering and Technology (ABET) oversees engineering programs in the United States and several countries.
  - **Skills Development**: Education is only the beginning. It is not sufficient to develop full professional capabilities. Professional need practice applying their knowledge before they take full responsibility for performing work in the field.
    - For example: Medical professionals must have several years of residency.

A Guide Toward Software Engineering Profession

- **Certification**: After completion of education and skill development, a professional is required to pass one or more exams that ensure they have attained a minimum level of knowledge.
- **Licensing**: Licensing is similar to certification except that it is mandatory and administered by a government authority.
- **Professional Development**: Many professionals are required to keep their professional education current by attending on-going professional education seminars, or workshops to maintain or improve their knowledge and skills.
- **Professional Society**: Professionals see themselves as part of a community who put their professional standards above their individual self interest.
- **Code of Ethics**: Each profession has a code of ethics to ensure that its practitioners behave responsibly.

Professional Software Engineer

- □ Software industry supports career development of software professionals and encourages their people to focus on improving their skills and knowledge.

- □ In reality, many software people move from company to company and from one position to another without any structured improvement of their skills.

- □ The requirements of a truly software engineer professional are still being debated among professionals, academia, and companies.

- □ The software field is changing rapidly with new technologies, new methods and new areas. It is difficult to narrowly define the area into a specific one.

Professional Development - 1

- **Entry Level:**
  - At the entry level, the engineer is generally a graduate from a university and just beginning to learn the principles of software engineering.
  - The major effort during the first few years (1–3) is to understand how the company operates and learn the specific software development methods, and processes of the company.
  - This person usually works under close supervision of a senior engineer or a supervisor.

Professional Development - 2

- □ **Intermediate Level:**
  - At the intermediate level, the software engineer already has some background in software engineering and its application in the company.
  - The software engineer has worked in a full software development project from initial to release and has 2 to 4 years of experience.
  - The software engineer must understand all major areas of software engineering and focus on having competencies in at least 3 areas.
  - This person is usually capable of performing work with limited supervision.

Professional Development - 3

□ **Advanced Level:**

- At the advanced level, the software engineer has a strong background in software engineering and its application in the company.

- The software engineer has already completed at least one or more projects, already has experience in each of the basic software development life cycle steps needed to release a product and has 4 to 6 years of experience.

- At this level, the software engineer must understand all key areas and competence in 5 areas and specialize in at least one area (technical lead in one area).

- This person usually capable of performing work independently.

Professional Development - 4

- **Associated Technical Level:**
  - At this level, the software engineer has consistently a strong record of success in proving the technical leadership in several projects.
  - The software engineer works a a technical lead or project manager in small to medium size projects and has been essential to the project’s success.
  - At this level, the software engineer already has competency in all key areas and provides technical leadership in at least 5 areas with 6 to 10 years of working experience.
  - This person is usually providing guidance and supervises others.

Professional Development - 5

- **Technical Leader Level:**
  - At this level, the software engineer is a person who already comprehends all aspects of software development and is able to handle additional responsibilities in both the internal and external aspects of managing software project.
  - The software engineer works as a project manager and takes total ownership of the project with pride. This leader contributes significantly to the success of the company and has mastered all key areas of the software engineering professional.
  - At this level, the software technical leader begins to work as mentor to others and promote the software engineer professional ethics.

Professional Development - 6

- **Senior Technical Level:**
  - At this level, the software engineer is a major resource to both the company and the software community. This person consistently overcomes very difficult technical challenges and makes key decisions on the goals and structured of the company.
  - This person has mastered all the key areas of software engineering and continues to contribute to the advancement of software engineering via research, and technical contribution to professional publications.
  - The senior technical area of expertise extends beyond a company to an industry level of influence.
  - At this level, the senior leader already has a career-long commitment to the field with significant contribution to the industry at both the national and international level.

Professional Development - 7

- **Mastery Level:**
  - At this level, the senior engineer is a key influencer on major issues confronting the software engineering community. This person consistently works to design and produce significant breakthrough technology and create world-class products.
  - This person continues to contribute to the advancement of software engineering via researches, and technical contribution to professional publications.
  - The mastery level of expertise extends beyond national to the international level of influence.

My Questions

- The software engineering field is changing rapidly and already has several good practices.
- The issue is that too few people know about the practices and even fewer people use them.
- Researchers have found that it typically takes 10 to 15 years for new ideas and best practices to flow-down the technology learning path.
- However, many of these ideas have been around for more than 15 years and why don’t people use them?

Is It Possible?

- □ Reduce the operating cost of software products & services?
- □ Improve time to market software products?
- □ Improve quality of software products & services?
- □ Improve the developing cost of software products?
- □ Improve the timeliness of software products & services?
- □ Improve the reliability of software products & services?
- □ Be the “Best-in-Class” in the industry?
- □ Operate the business as economically as possible?

The answer is: YES, YES, and YES !

My recommendation

Study Software Engineering
and
become a
Software Professional!