

SE202

Introduction to Software Engineering

Chapter 1 – What is Software Engineering?

Dr. Pathathai Na Lumpoon
2-2019

Agenda

- What is Software Engineering?
- Why Software Engineering? Software crisis & its solution
- Why Software Engineering is popular?
- Key questions about SE
- Things to keep in mind as Software Engineer

What do you think '*Software Engineering*' is about?

Software

Computer programs + associated documentation + configuration data.

A professionally developed software system is often more than a single program.

Engineering

A systematic, scientific, and well-defined **process** to produce a good quality product.

Software engineering is an engineering discipline that is concerned with all aspects of software production and intended to support professional software development, rather than individual programming.

Types of Software products

- **Generic**

- Sold on open market
 - Decisions on software change are made by the developer.
- Often called
 - COTS (Commercial Off The Shelf)
- Cheaper and more reliable comparing to Custom
- E.g. Software for PCs such as databases, word processors and drawing packages.

- **Custom**

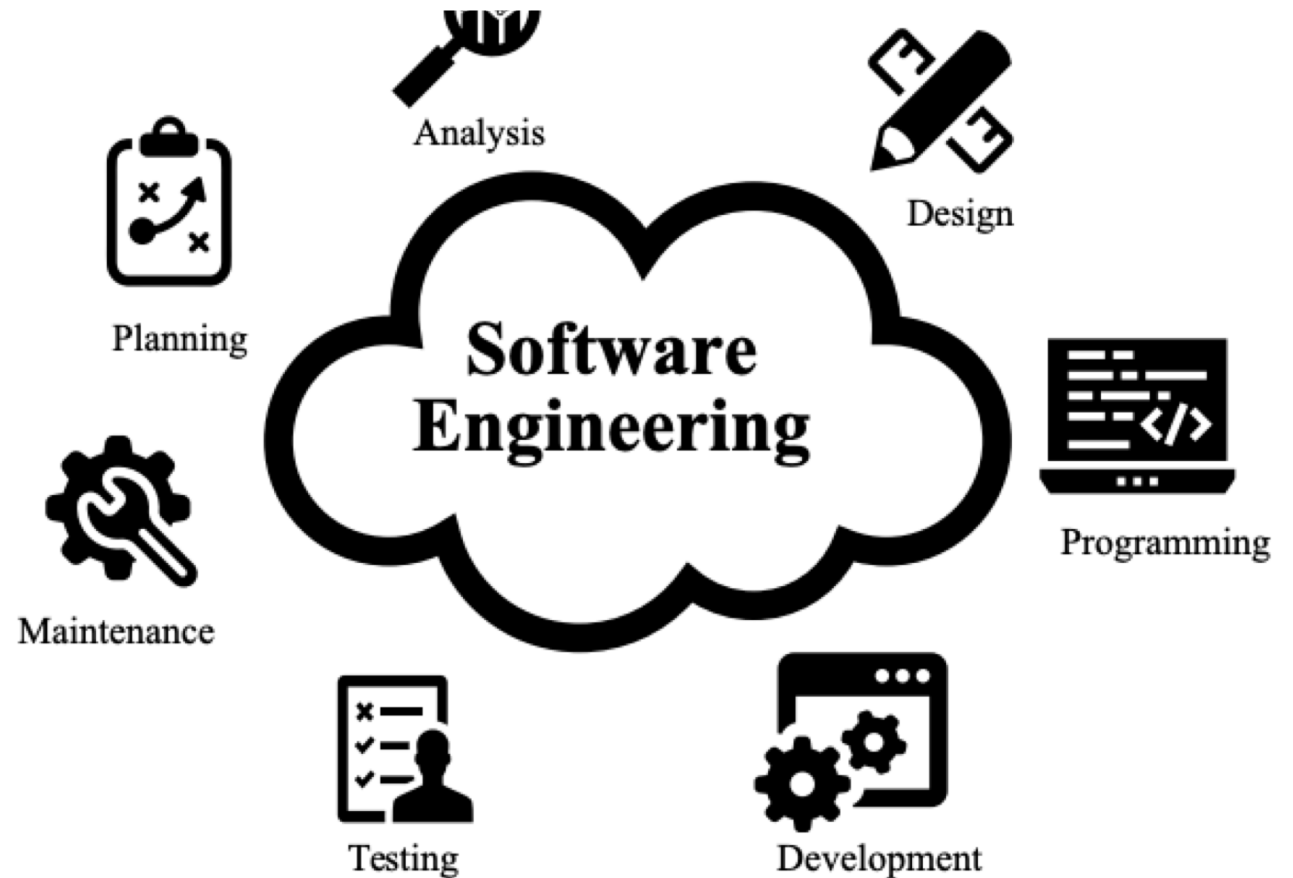
- For a specific customer, specific purposes
- E.g. Systems written to support a particular business process and air traffic control systems

Seven broad categories of computer software

- System Software
- Application Software
- Engineering and Scientific Software
- Embedded Software
- Product-line Software
- Web Application
- Artificial Intelligence Software

What is Software Engineering?

- A **process** of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.



Other definitions of software engineering

- IEEE, in its standard 610.12-1990, defines software engineering as the application of a systematic, disciplined, which is a computable approach for the development, operation, and maintenance of software.
- Fritz Bauer defined it as 'the establishment and used standard engineering principles. It helps you to obtain, economically, software which is reliable and works efficiently on the real machines'.
- Boehm defines software engineering, which involves, 'the practical application of scientific knowledge to the creative design and building of computer programs. It also includes associated documentation needed for developing, operating, and maintaining them.'

Why Software Engineering?

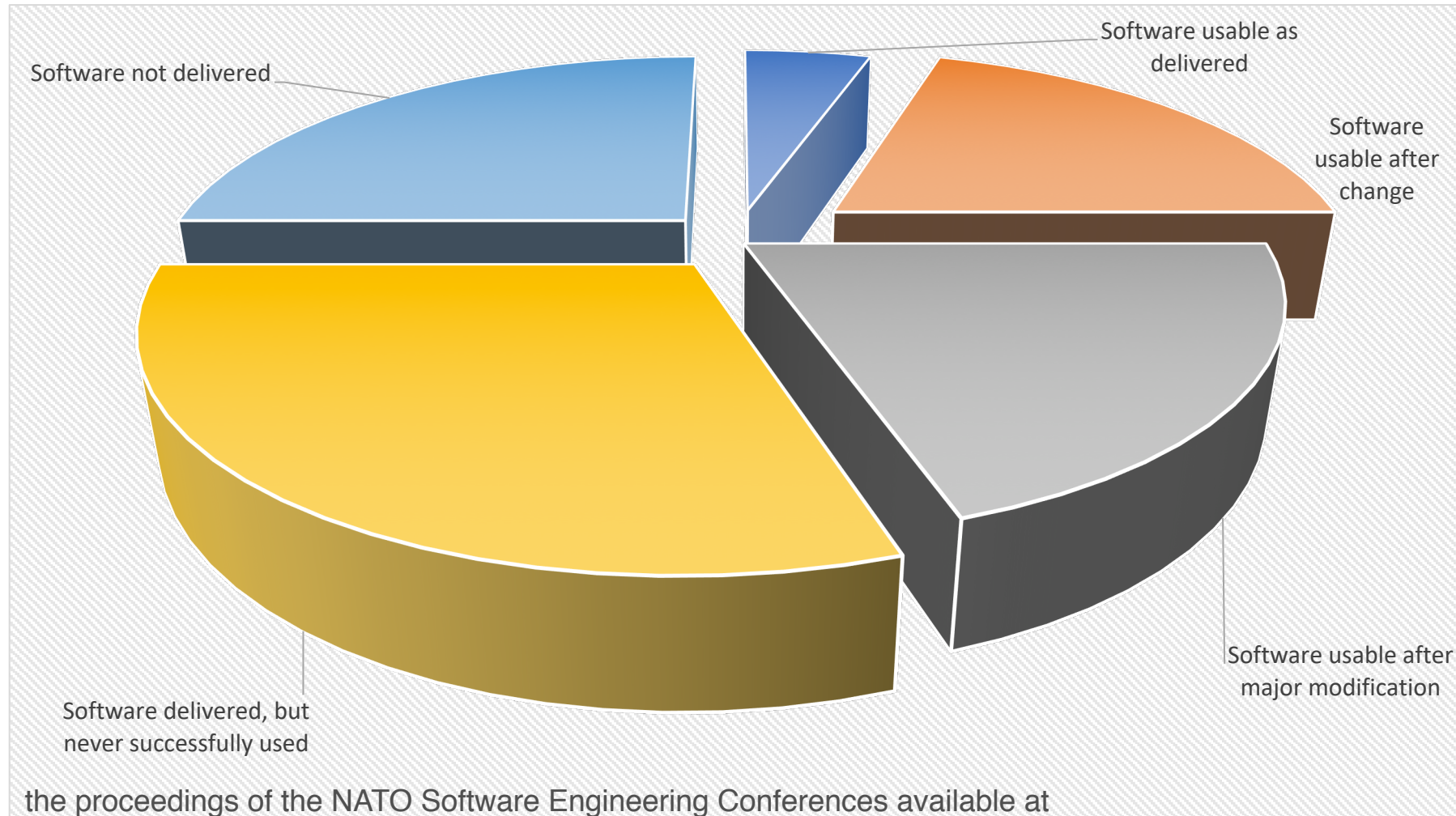
Software Crisis & its Solution

What was the Software Crisis?

- It was in the late 1960s when many software projects failed.
- Many software became over budget. Output was an unreliable software which is expensive to maintain.
- At that time, program developments were based on individual approaches
- Larger software was difficult and quite expensive to maintain.
- Lots of software not able to satisfy the growing requirements of the customer.
- Complexities of software projects increased whenever its hardware capability increased.
- Demand for new software increased faster compared with the ability to generate new software.

Evidence of Software crisis

Study of 9 Software Development contracts (Davis, 1990)



the proceedings of the NATO Software Engineering Conferences available at
<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>

The Solution

- Solution was to the problem was transforming unorganized coding effort into a software engineering discipline. These engineering models helped companies to streamline operations and deliver software meeting customer requirements.
- The notion of '*software engineering*' was first proposed *in 1968*
- A conference held to discuss about the '**software crisis**' (Naur and Randell, 1969)
- The late 1970s saw the widespread uses of software engineering principles.
- In the 1980s saw the automation of software engineering process and growth of (CASE) Computer-Aided Software Engineering.
- The 1990s have seen an increased emphasis on the 'management' aspects of projects standard of quality and processes just like ISO 9001

Why Software Engineering is Popular?

- **Nature of software**

- Software is complex, large, tangible
- Software is adaptability
- Software is changeability
- Software is scalability
- Software is costly

- **Quality Management**

- Offers better method of software development to provide quality software products.



People get smarter..
So do the technologies..
Users get even smarter..
They ask for something
smarter, faster, easier to use..
Software Engineers' life
becomes tougher T-T..



Key questions about SE

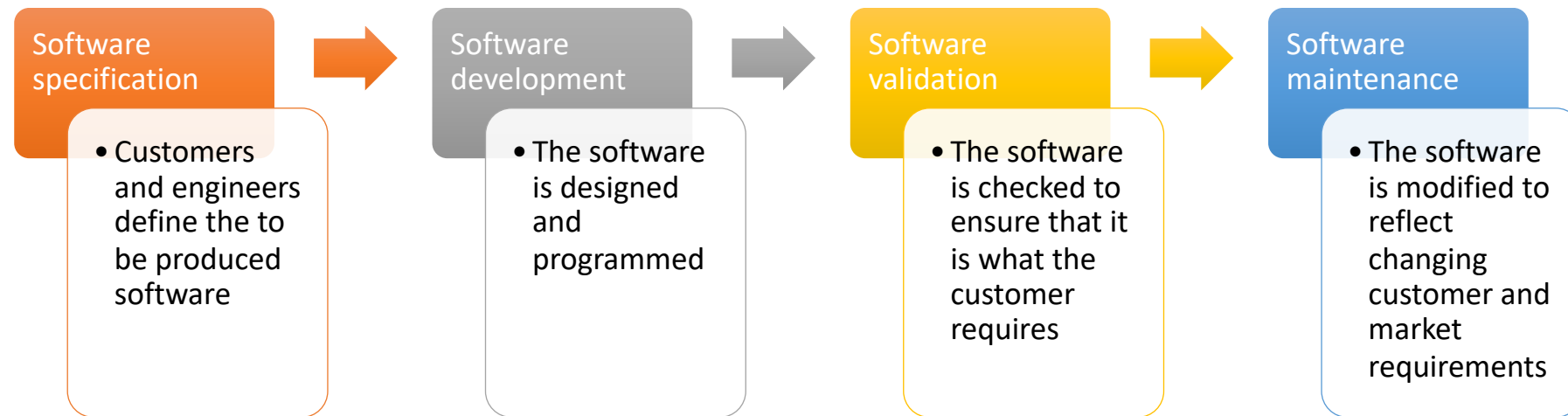
1. What are software engineering methods?

“... a **structured approach** to software development whose aim is to facilitate the production of high-quality software in a cost-effective way.”

e.g. function-oriented methods, object-oriented methods, Unified Modeling Language (UML)

But Software Engineering doesn't mean everything needs to be formal and rigid. **Less formal** approaches are more efficient in certain cases.

What are the fundamental software engineering activities?



2. What are the attributes of good software?

- **Functionality:** deliver required functions and performance
- **Maintainability:** accommodate changes and evolution of software
- **Dependability:** trustworthy (reliability, security and safety)
- **Efficiency:** does not waste system resources
- **Usability:** usable by the intended users

3. What are the key challenges facing software engineering?

- **Legacy system:** need to maintain and update old but valuable systems
- **Diversity** : develop and maintain communication among the diversity of software systems
- **Complexity** : deal with the increased complexity of software need for new applications.
- **Delivery:** develop a quality product in short delivery time
- **Trust:** develop a product that can be trusted by its users

4. What are the costs of software engineering?

- Costs vary depending on the type of software being developed, software size, software complexity and the requirements of system attributes such as performance and system reliability.
- Maintenance typically consumes 40 to 80% (60% average) of software costs
- In safety-critical areas such as space, aviation, nuclear power plants, etc. the cost of software failure can be massive because lives are at risk.

Things to keep in mind as
Software Engineer

Licensing Professional Software Engineers

- Engineering is a licensed profession
 - In order to protect the public, safety, health, or welfare
 - Engineers design artifacts following well accepted practices which involve the application of science, mathematics and economics
 - *Ethical practice is also a key tenet of the profession*
- In many countries, much software engineers does not require an engineering license, but is still an engineer.

Ethics in Software Engineering: IEEE/ACM code of ethics

- Software engineers shall
 - Act consistently with *public* interest
 - Act in the best interests of their *clients*
 - Develop and maintain with the highest *standards* possible
 - Maintain integrity and independence
 - Promote an ethical approach in management
 - Advance the integrity and reputation of the profession
 - Be fair and supportive to *colleagues*
 - Participate in *lifelong learning*