

Software Engineering

HUANG Jie

School of Software Engineering

Tongji University, 2021



同濟大學
TONGJI UNIVERSITY

Chapter 1

The Nature of Software

- ✓ What is Computer?
- ✓ What is Software?
- ✓ What is Software Engineering?
- ✓ Why Software Engineering is important?
- ✓ Challenges in building software projects & software products.
- ✓ Careers in software engineering.
- ✓ Software application domain.

What is Computer?

Engineering helped create large man-made structures using applied science.

Today, we use devices such as mobile phones, Internet and automobiles, most of these are driven or controlled by computers.

Question

Could you find a device without software in your daily life?

Question

*What is the **difference** between **science**, **engineering** and **technology**?*

Components of a Computer

Hardware & Software.

A Computer system also contains many types of software. Normally, in three types, Application Software, Middleware and System Software.

Question

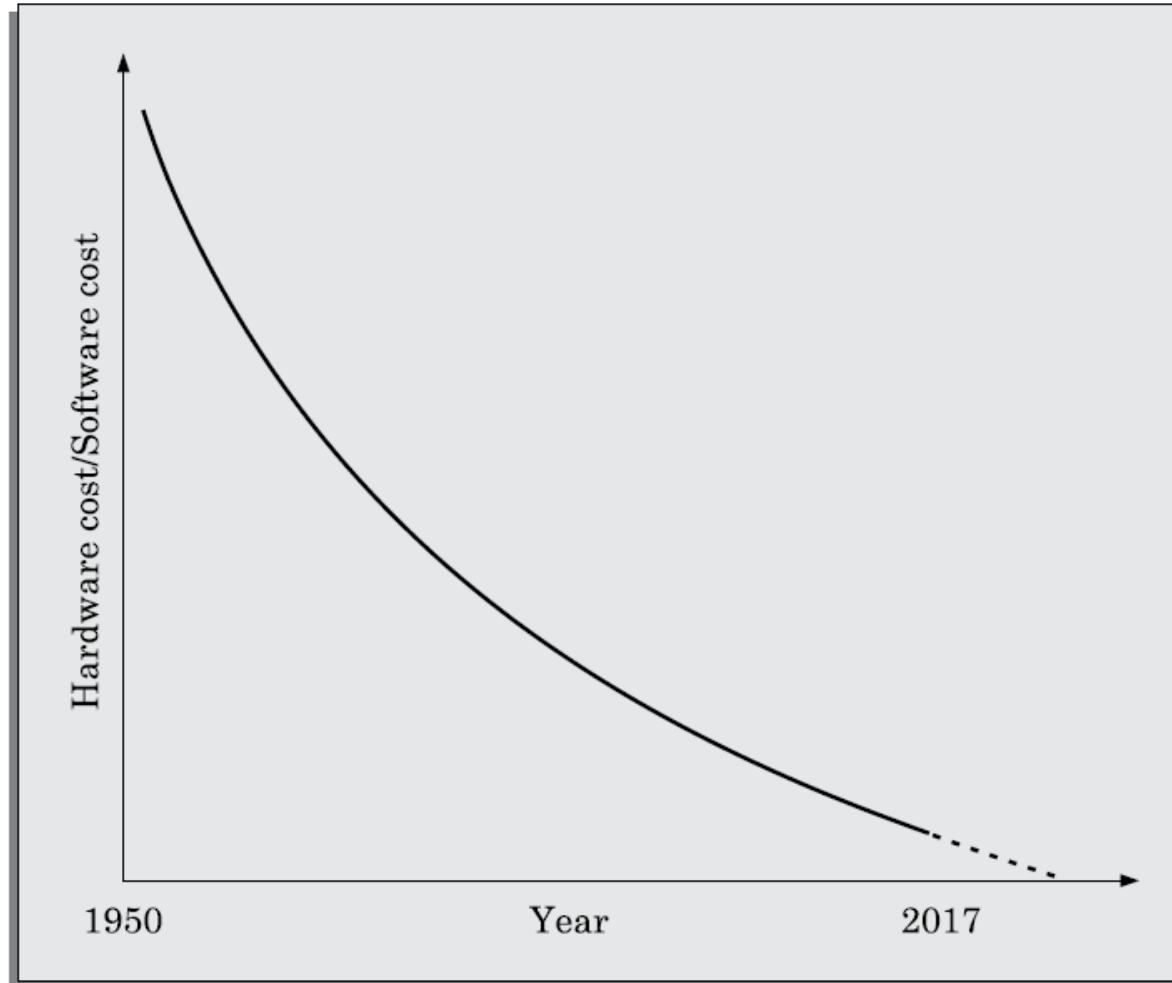
Give some examples of App. Middleware or System?

Almost all software products or application programs running on your computer have been developed by a team.

Example

Microsoft Word product.

Hardware cost vs Software cost



What is Software?

Software is

- (1) **instructions** (computer programs) that when executed provide desired features, function, and performance;
- (2) **data structures** that enable the programs to adequately manipulate information;
- (3) **documentation** that describes the operation and use of the programs.

What is Software?

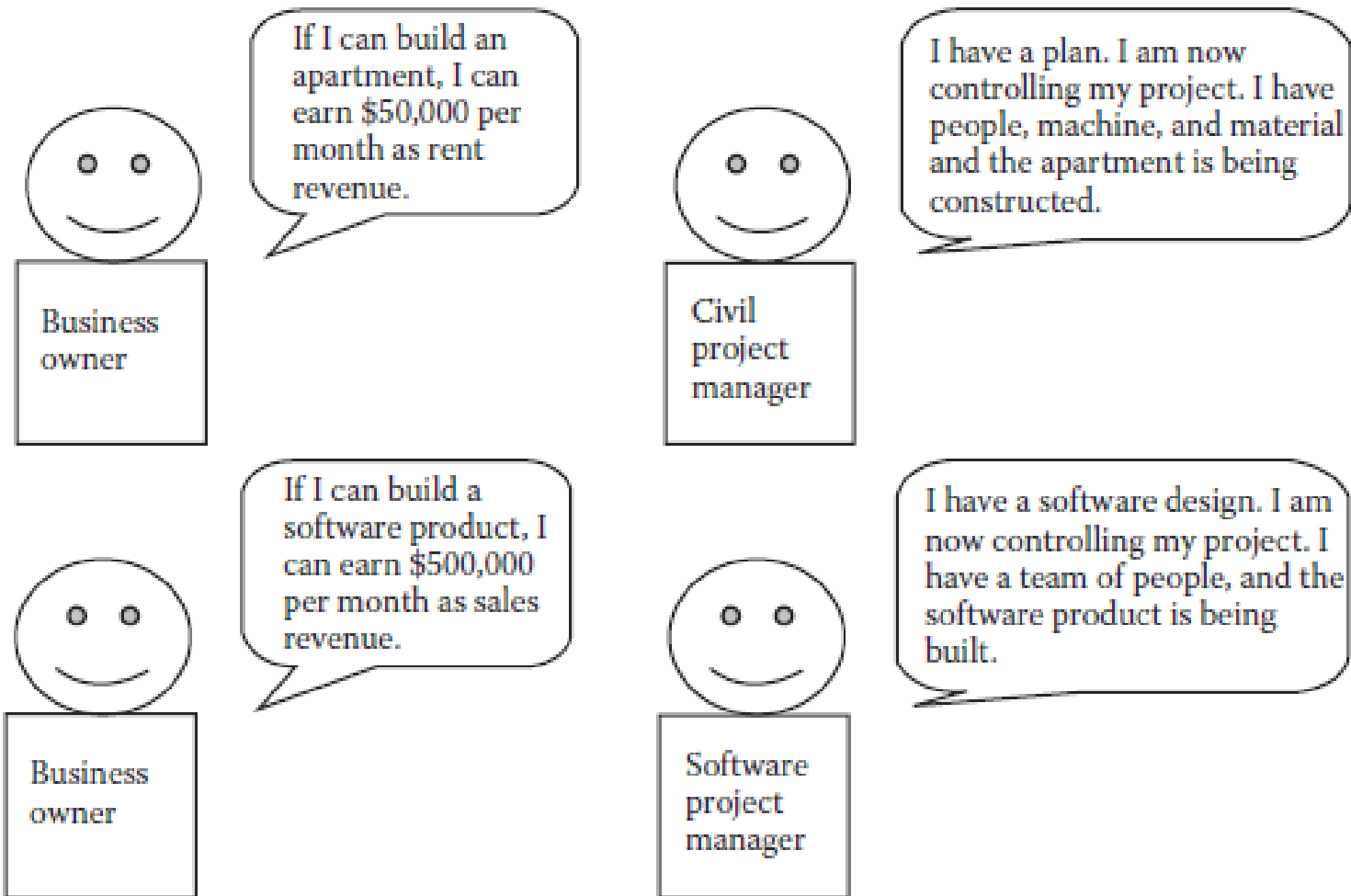
Any computer program, small or big, is qualified as a software product, that needs at least a group of people to develop it for several weeks or longer.

The terms **software**, **software project** and **software product** are used synonymously in this course.

Example: Comparing an apartment construction vs. a software product development.

Question: *Do you find the **difference** between the two words underlined in this example?*

What is Software?



What is Software?

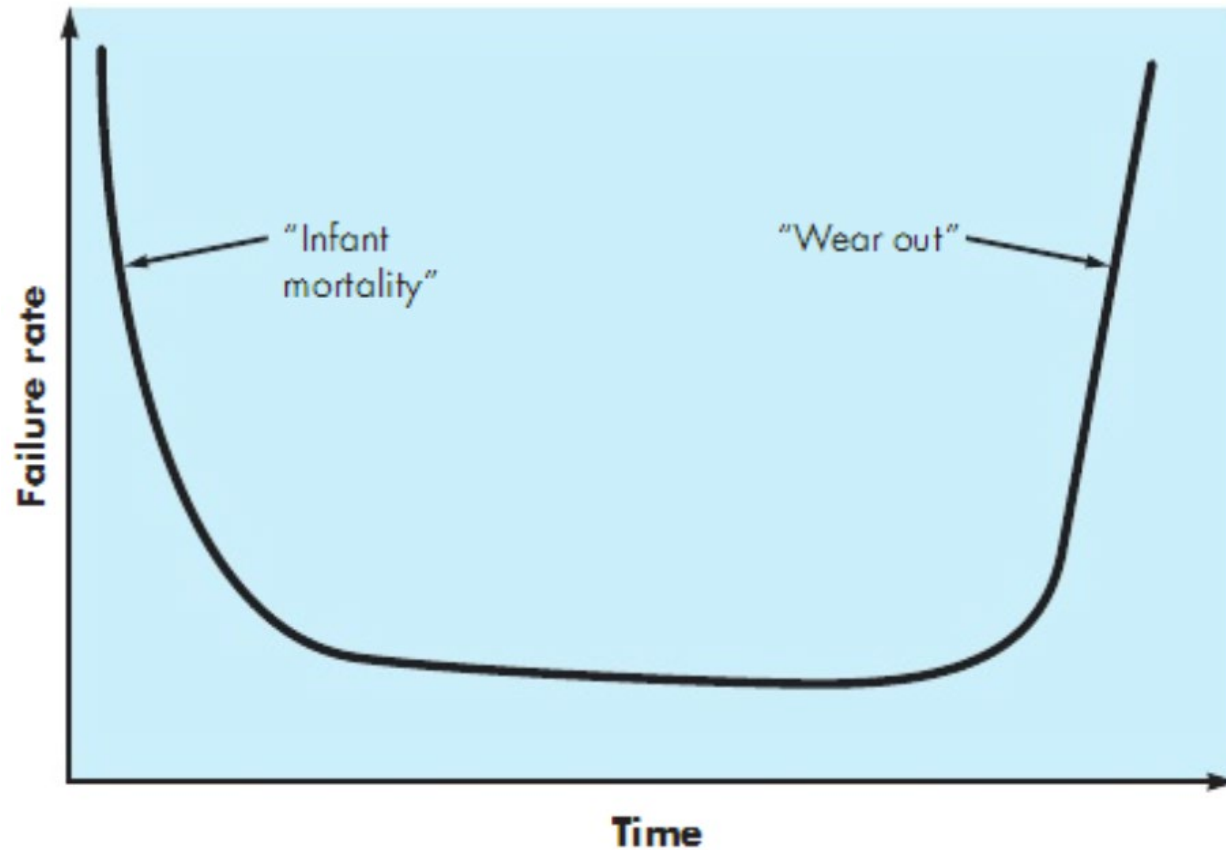
Building a software product is an activity similar to the construction of an apartment. Users must get **functions** !

We should also think the **non-functional** issues, similar to the measures taken while constructing the apartment, these considerations including security, performance and reliability.

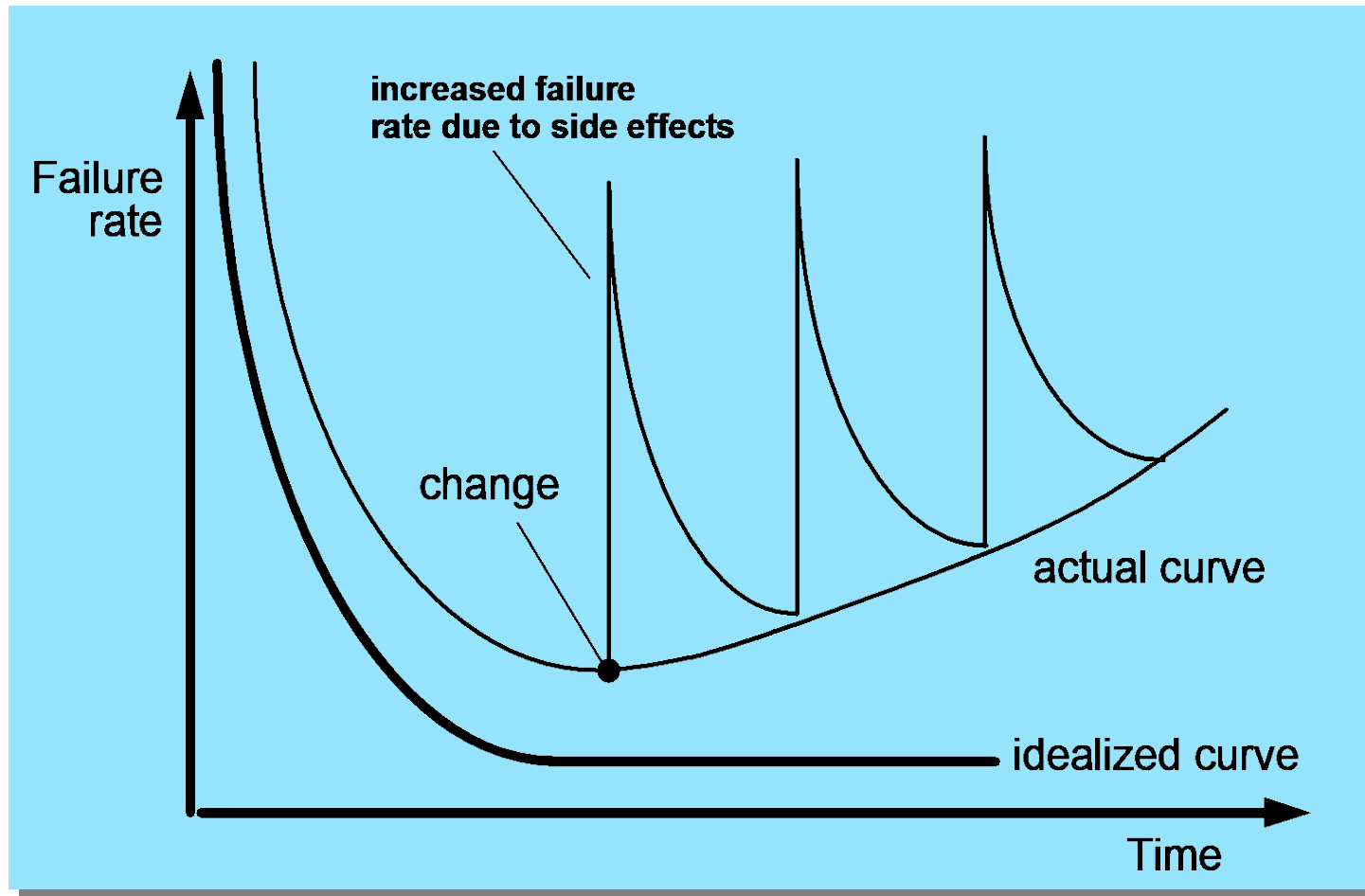
But, there are **differences** between hardware and software.

Example: Wear vs. Deterioration

Wear vs. Deterioration

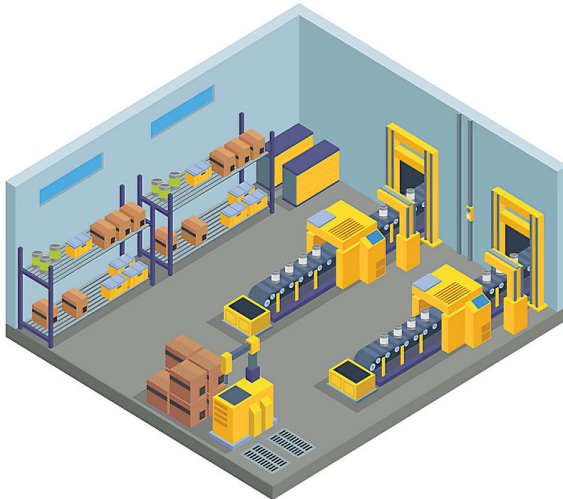


Wear vs. Deterioration



What is Software?

- Software doesn't **wear out**.
- Software is developed or engineered, it is not manufactured in the classical sense.
- Although the industry is moving toward component-based construction, most software continues to be custom-built.



What is Software Engineering?

■ Definition of Software Engineering

- "the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software". — **IEEE**
- "an engineering discipline that is concerned with all aspects of software production" — **Ian Sommerville**
- "the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines" — **Fritz Bauer**

■ Question

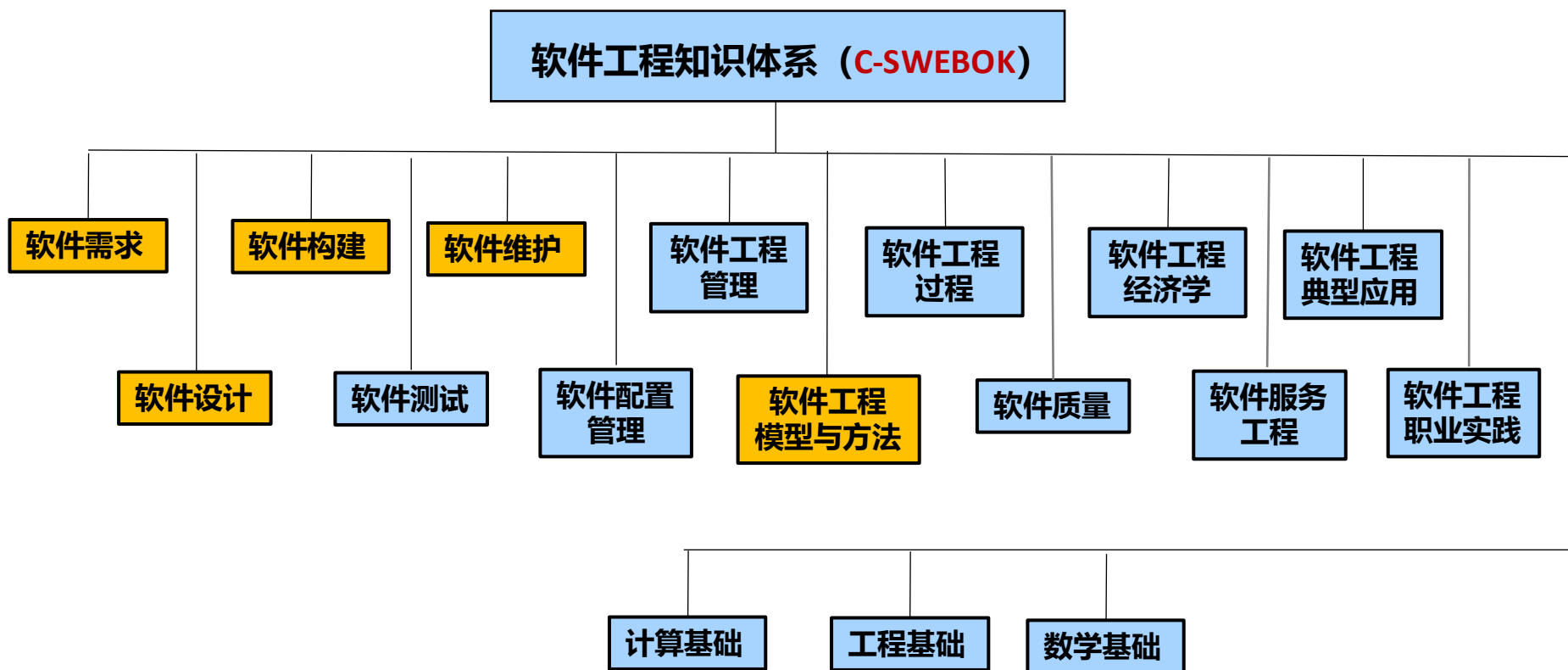
What is the definition of Software Engineering proposed by Roger S. Pressman?

What is Software engineering?

- Software engineering can be defined as the **study & application** of engineering to the design, development and maintenance of software product.
- Software engineering discusses **systematic** and **cost-effective** techniques to software development. These techniques have resulted from innovations as well as lessons learnt from past mistakes.
- Engineering always connotes **several** things:
 - **Designing** a product in detail.
 - A **methodical approach** to carrying out the design and production so that the product is produced with the desired qualities.
 - **Trade-off** in different alternatives.

The position of this course in **C-SWEBOK**

中国软件工程知识体系 (C-SWEBOK)



The main content of this course

- The course presents a framework that can be used by those who build computer software – people who must get it right.
- The framework encompasses
 - Software process and its models.
 - Requirement engineering.
 - Software design and methods.
 - Software configuration and maintenance.

Why Software Engineering important?

- Building large software products does require software engineering. Definitely, software engineering techniques make it possible to create useful software products with sufficient quality, within the time limits and within the budget if software engineering practices are adopted.
- Functions of software engineering
 - Reduction of development & maintenance **costs**.
 - Reduction of development **time**.
 - Increasing the **quality**: quality is the single most important ingredient in making any product successful.

Challenges in Software Engineering

- Developing a software product is always a challenging job. The reasons are numerous.
- The conversion of user requirements into a software product involves a long chain of processes.

- **E**xample

Suppose a user requires a software that is able to analyze the popularity of his/her personal web page.

- **Q**uestion

How are you going to build such a software?

Challenges in Software Engineering

- **Example (cont.)**
 - First, design a program counting the number of hits to that web page within a specified period of time, e.g. 24 hours.
 - Then, report this count to an e-mail address of the user.
 - Third, write the source code using a programming language.
 - Last, after thorough testing, you can deliver the product to the user.
- Building such a simple software is easy. However, consider that the user's requirements are not only to provide the number of hits to the web page, but also to categorize such hits according to the following criteria:
 - Number of hits for the last year.
 - Number of visitors by country per day, per week, and per month.
 - Number of visitors by different browsers used by the visitors.
 - Number of hits and the number of unique visitors.
 - Number of visitors by the operating system used.

Challenges in Software Engineering

- **E**xample (cont.)

- Now, if you try to design the software according to the new requirements of the user, you will realize that it is not easy to develop the required software this time.

- As the complexity and size of the user requirements increase, the design and implementation sizes of the product also increase. This increase in complexity and size is the main problem associated with the development of software products. This is the reality that most of software product development projects face.

Software application domains

- **Seven** broad categories of software
 - System software.
 - Application software.
 - Engineering/Scientific software.
 - Embedded software.
 - Product-line software.
 - Web/Mobile applications.
 - Artificial intelligence software.

Careers in Software Engineering

- Software engineering provides several careers to pursue.
 - A person having skills and interest in finding defects in software, can pursue a career in software testing.
 - Anyone who enjoys programming can become a software developer and write source code.
 - A person good in designing and architecture can become a software designer or architect.
 - A person who likes the task of collection and classification of information can become a business analyst.
 - A person who has gained experience in more than one area of software engineering can become a software project manager.
 - New careers are emerging as software engineering technology emerges.
- **E**xample: Career's salary table

Legacy Software

Why must legacy software change?

- Software must be **adapted** to meet the needs of new computing environments or technology.
- Software must be **enhanced** to implement new business requirements.
- Software must be **extended to make it interoperable** with other more modern systems or databases.
- Software must be **re-architected** to make it viable within a network environment.
- **Q**uestion: Give some examples of legacy software.

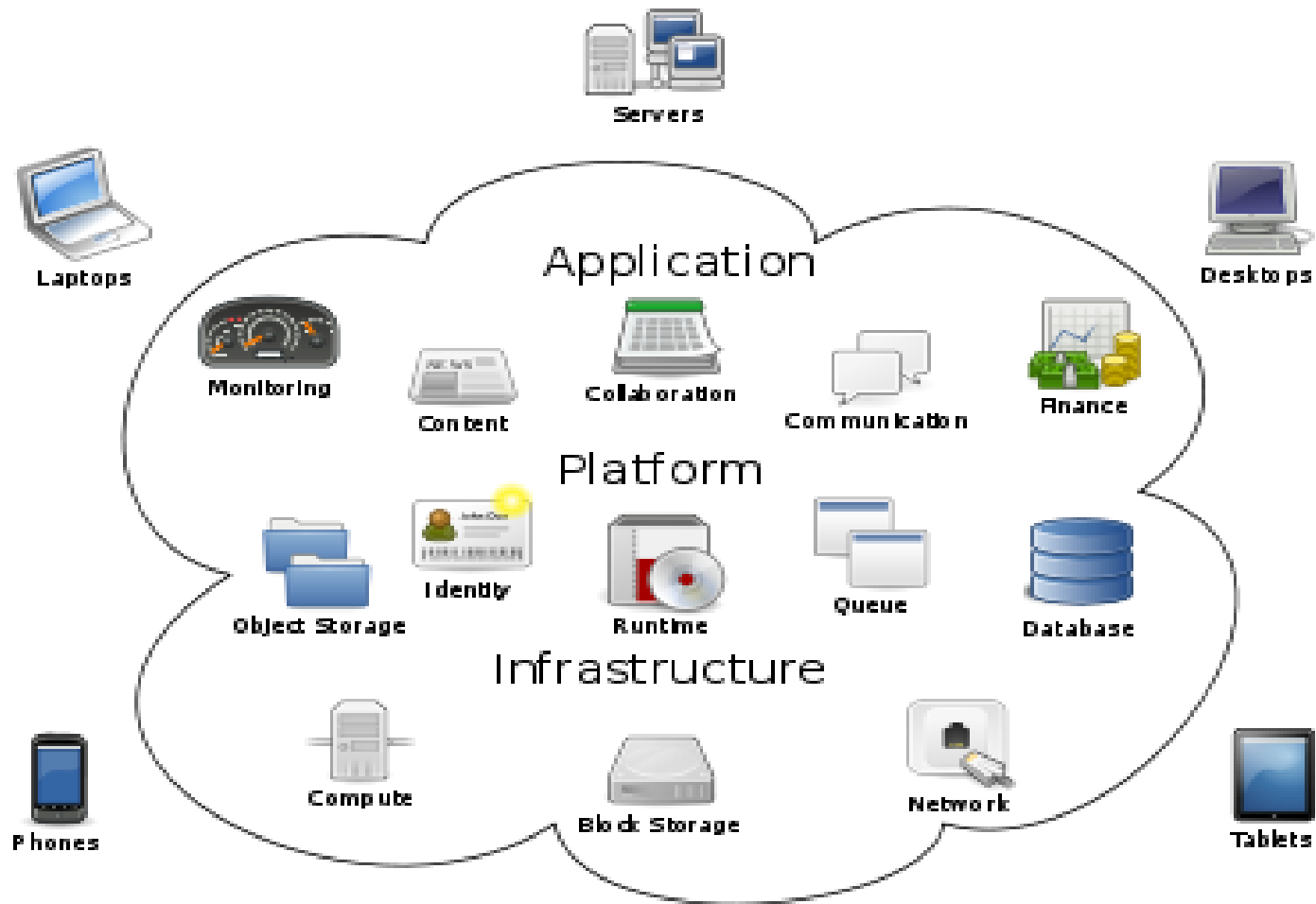
WebApps

- Modern WebApps are much more than hypertext files with a few pictures.
- WebApps are augmented with tools like XML and Java to allow Web engineers including interactive computing capability.
- WebApps may standalone capability to end users or may be integrated with corporate databases and business applications.
- Semantic web technologies (Web 3.0) have evolved into sophisticated corporate and consumer applications that encompass semantic databases that require web linking, flexible data representation, and application programmer interfaces (API's) for access.
- The aesthetic nature of the content remains an important determinant of the quality of a WebApp.

Mobile Apps

- Reside on mobile platforms such as cell phones or tablets.
- Contain user interfaces that take both device characteristics and location attributes.
- Often provide access to a combination of web-based resources and local device processing and storage capabilities.
- Provide persistent storage capabilities within the platform.
- A mobile web application allows a mobile device to access to web-based content using a browser designed to accommodate the strengths and weaknesses of the mobile platform.
- A mobile app can gain direct access to the hardware found on the device to provide local processing and storage capabilities.
- As time passes these differences will become blurred.

Cloud Computing



Cloud Computing

Cloud Computing

- Cloud computing provides distributed data storage and processing resources to networked computing devices.
- Computing resources reside outside the cloud and have access to a variety of resources inside the cloud.
- Cloud computing requires developing an architecture containing both frontend and backend services.
- Frontend services include the client devices and application software to allow access.
- Backend services include servers, data storage, and server-resident applications.
- Cloud architectures can be segmented to restrict access to private data.

Product Line Software

- Product line software is a set of software-intensive systems that share a common set of features and satisfy the needs of a particular market.
- These software products are developed using the same application and data architectures using a common core of reusable software components.
- A software product line shares a set of assets that include requirements, architecture, design patterns, reusable components, test cases, and other work products.
- A software product line allow in the development of many products that are engineered by capitalizing on the commonality among all products with in the product line.

Assignments

■ Chapter 1. Problems to ponder

- 1.2 Provide examples(both positive and negative) that indicate the impact of software on our society.
- 1.3 Develop your own answers to the five questions asked at the beginning of Section1.1 in the textbook.

Discuss with your classmates.

■ Questions

- What is the Definition of Software Engineering proposed by Roger S. Pressman?
- What is the principal aim of the software engineering discipline? What does the discipline of software engineering discuss?

Assignments

■ Reading

The Mythical Man-Month Essay on Software Engineering,
written by FrederickP.Brooks

Chapter 1. The Tar Pit

■ Preview

Software Engineering 8th edition, by Roger.S.Pressman

Chapter 2 Software engineering

Chapter 3 Software process structure

The Mythical Man-Month Essay on Software Engineering,
written by FrederickP.Brooks

Chapter 2. The Mythical Man-Month