

Table of Contents

Table of Contents	1
Learning Objectives	2
What is Computer Software?	2
Who is Software Engineer?	2
What is Computer Software Important?	2
How Computer Software are Build?.....	2
What is the Work Product?	2
Why Software Engineering?	2
Role of Software	3
Software Definition.....	3
Software Characteristics	3
Software Applications.....	5
System Software	6
Real-time Software	6
Business Software.....	6
Engineering and Scientific Software	6
Embedded Software	7
Personal Computer Software	7
Web-based Software	7
Artificial Intelligence Software.....	7
Reference for Further Reading.....	7

Learning Objectives

After completing this handout, you will be able to:

- Understand what is software engineering.
- Understand the role of software.
- Understand the definition of software from software engineering perspective.
- Learn the software characteristics.
- Learn the software application areas.
- Learn the software crisis.
- Learn the software myths.

What is Computer Software?

- Computer software is the product that software engineers design and build.
- It encompasses
 - programs
 - documents
 - data

Who is Software Engineer?

- Software engineers analyze, design, and develop the computer software.

What is Computer Software Important?

- Because it affects nearly every aspect of our lives and has become pervasive in our commerce, our culture, and our everyday activities.

How Computer Software are Build?

- Software engineers build computer software by applying a process that leads to a high-quality result that meets the needs of the people who will use the computer software. They apply a software engineering approach.

What is the Work Product?

- The work product is the programs, documents, and data that are computer software.

Why Software Engineering?

- Software's impact on our society and culture continues to be profound.

- As its importance grows, the software community continually attempts to develop technologies that will make it easier, faster, and less expensive to build high-quality computer programs.
- Some of these technologies are targeted at a specific application domain (e.g., Web-site design and implementation).
- Others focus on a technology domain (e.g., object-oriented systems).
- Still others are broad-based (e.g., operating systems such as LINUX).

Role of Software

- Software takes on a dual role. It is a product and, at the same time, the vehicle for delivering a product.
- As a product, software is an information transformer—producing, managing, acquiring, modifying, displaying, or transmitting information that can be as simple as a single bit or as complex as a multimedia presentation.
- As the vehicle, it is used to deliver the product, software acts as the basis for the control of the computer (operating systems), the communication of information (networks), and the creation and control of other programs (software tools and environments).
- Software delivers information.
- Software transforms personal data (e.g., an individual's financial transactions) so that the data can be more useful in a local context.
- Software manages business information to enhance competitiveness.
- Software provides a gateway to worldwide information networks (e.g., Internet).

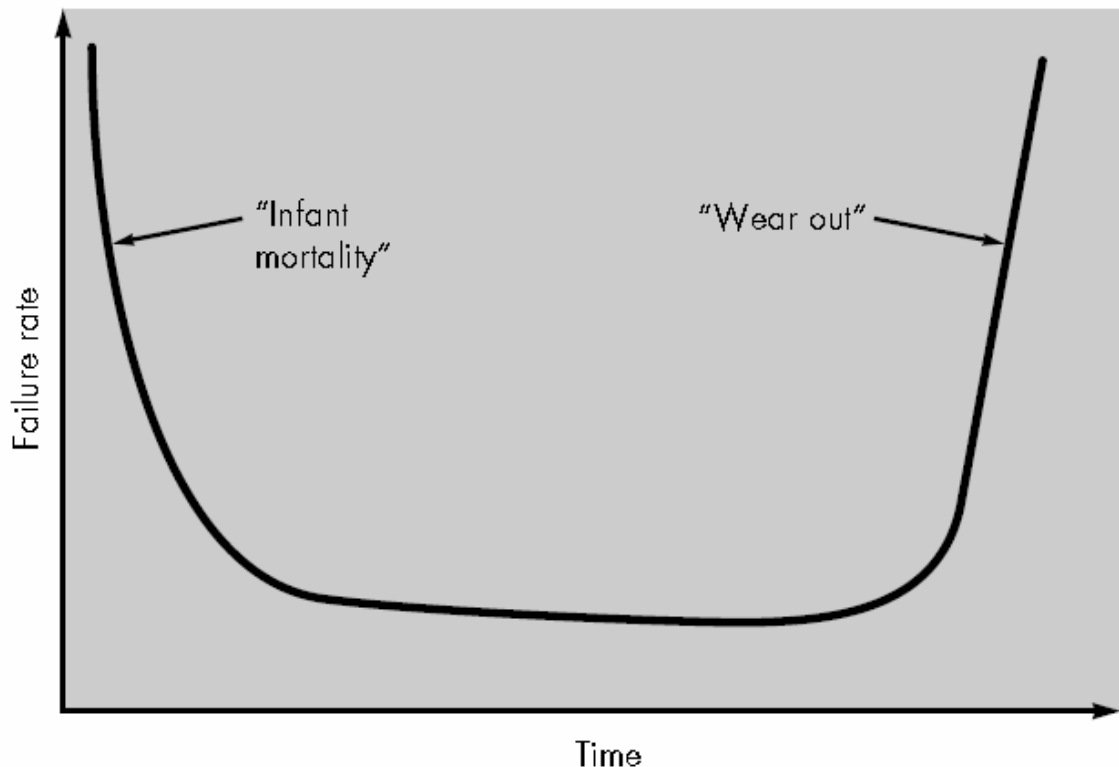
Software Definition

- Software is
 - instructions (computer programs) that when executed provide desired function and performance
 - data structures that enable the programs to adequately manipulate information, and
 - documents that describe the operation and use of the programs. There is no question that other, more complete definitions could be offered

Software Characteristics

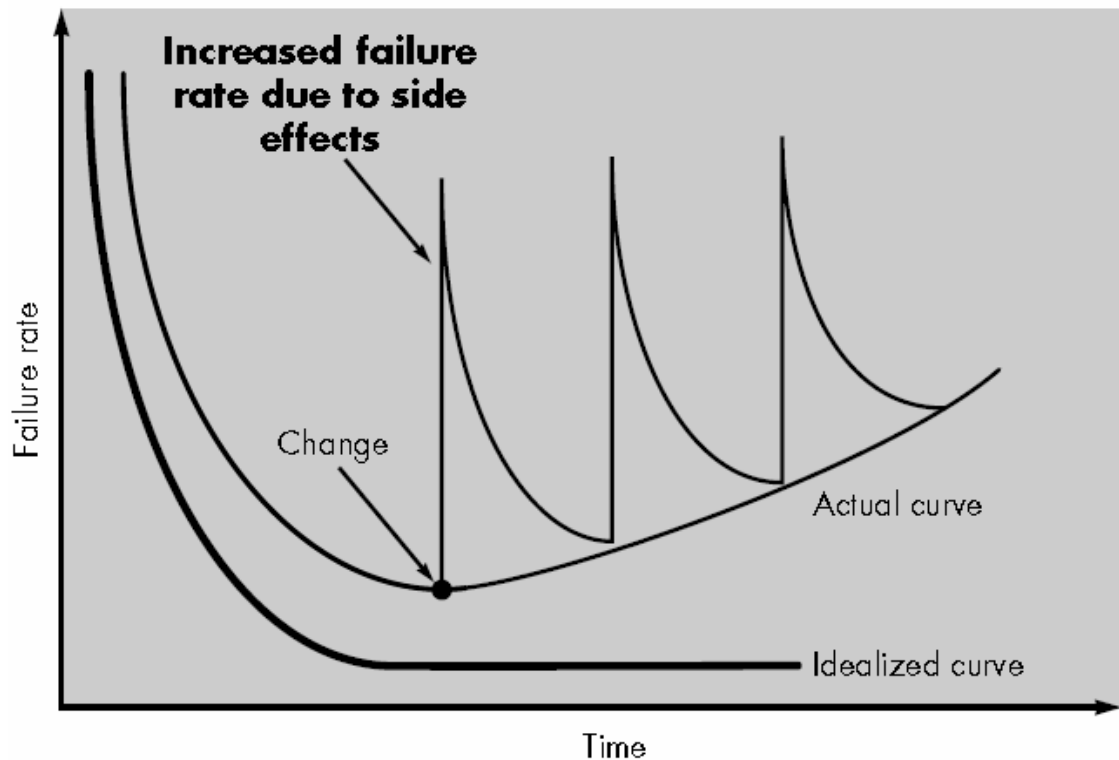
- Software is developed

- Software is developed or engineered; it is not manufactured in the classical sense.
- Some similarities exist between software development and hardware manufacture.
- In both activities, high quality is achieved through good design.
- Both activities are dependent on people, but the relationship between people applied and work accomplished is entirely different.
- Both activities require the construction of a "product" but the approaches are different.
- Software doesn't "wear out."
 - Hardware exhibits relatively high failure rates early in its life.
 - Defects are corrected and the failure rate drops to a steady-state level.
 - As time passes, the failure rate rises again as hardware components suffer from the cumulative affects of dust, vibration, abuse, temperature extremes, and many other environmental maladies.
 - Hardware begins to wear out.



- Software is not susceptible to the environmental maladies that cause hardware to wear out.

- Undiscovered defects will cause high failure rates early in the life of a program.
- Software doesn't wear out. But it does deteriorate!



- Software continues to be custom built.
 - A software component should be designed and implemented so that it can be reused in many different programs.
 - Programmer built scientific subroutine libraries that were reusable in a broad array of engineering and scientific applications.
 - Modern reusable components encapsulate both data and the processing applied to the data, enabling the software engineer to create new applications from reusable parts.
 - For example, today's graphical user interfaces are built using reusable components that enable the creation of graphics windows, pull-down menus, and a wide variety of interaction mechanisms.

Software Applications

- Information content and determinacy are important factors in determining the nature of a software application.

- Content refers to the meaning and form of incoming and outgoing information. For example, many business applications use highly structured input data (a database) and produce formatted "reports."
- Information determinacy refers to the predictability of the order and timing of information.

System Software

- System software is a collection of programs written to service other programs. Some system software (e.g., compilers, editors, and file management utilities) process complex, but determinate, information structures.
- The system software area is characterized by heavy interaction with computer hardware; heavy usage by multiple users; concurrent operation that requires scheduling, resource sharing, and sophisticated process management; complex data structures; and multiple external interfaces.

Real-time Software

- Software that monitors/analyzes/controls real-world events as they occur is called real time.
- Elements of real-time software include a data gathering component that collects and formats information from an external environment, an analysis component that transforms information as required by the application, a control/output component that responds to the external environment, and a monitoring component that coordinates all other components so that real-time response (typically ranging from 1 millisecond to 1 second) can be maintained.

Business Software

- Business information processing is the largest single software application area.
- Discrete "systems" (e.g., payroll, accounts receivable/payable, inventory) have evolved into management information system (MIS) software that accesses one or more large databases containing business information.
- In addition to conventional data processing application, business software applications also encompass interactive computing (e.g., point-of-sale transaction processing).

Engineering and Scientific Software

- Engineering and scientific software have been characterized by "number crunching" algorithms.

- Applications range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing.
- Computer-aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics.

Embedded Software

- Embedded software resides in read-only memory and is used to control products and systems for the consumer and industrial markets.
- Embedded software can perform very limited and esoteric functions (e.g., keypad control for a microwave oven) or provide significant function and control capability (e.g., digital functions in an automobile such as fuel control, dashboard displays, and braking systems).

Personal Computer Software

- Word processing, spreadsheets, computer graphics, multimedia, entertainment, database management, personal and business financial applications, external network, and database access are only a few of hundreds of applications.

Web-based Software

- The Web pages retrieved by a browser are software that incorporates executable instructions (e.g., CGI, HTML, Perl, or Java), and data (e.g., hypertext and a variety of visual and audio formats).

Artificial Intelligence Software

- Artificial intelligence (AI) software makes use of non-numerical algorithms to solve complex problems that are not amenable to computation or straightforward analysis.
- Expert systems, also called knowledge based systems, pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing are representative of applications within this category.

Reference for Further Reading

- Book: Software Engineering, A Practitioner's Approach, Fifth Edition, Roger S. Pressman, chapter 1.