Software Engineering

Chapter 1
The Nature of Software

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 8/e* (McGraw-Hill 2014). Slides copyright 2014 by Roger Pressman.

Reproduced by Ning Li , 2022

Software is everywhere!

Google

ctrip.com 携程旅行网

淘宝网 Taobao.com

amazon

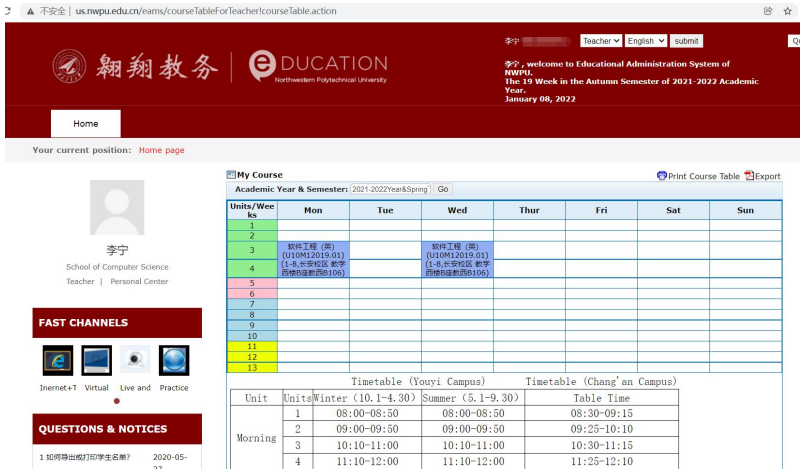
腾讯会议

WhatsApp

Tencent QQ

LinkedIn

Tripadvisor



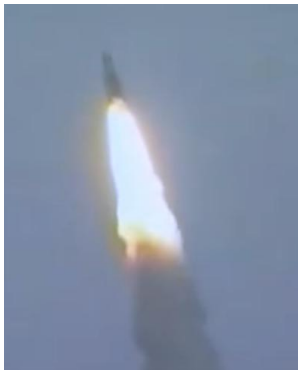
Software is everywhere!

Watch a video:

A software bug caused a disaster!

https://www.bilibili.com/video/av90036117?from=search&seid=15553121928885596131

Quiz



Ariane-5 501 (June,1996) rocket veered off its flight path about ( ) seconds after launch, beginning to disintegrate under high aerodynamic forces, and finally self-destructing by its automated flight termination system. ($370 million)

A) 20　　B) 37　　C) 45

software failure reason: SRI 16 bit integer overflow

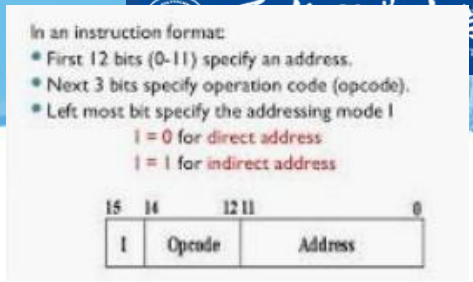
Contents

- 1.1 The nature of software
  - 1.1.1 Defining Software
  - 1.1.2 Software Application Domains
  - 1.1.3 Legacy Software
- 1.2 The Changing Nature of Software
  - 1.2.1 WebApps
  - 1.2.2 Mobile Applications
  - 1.2.3 Cloud Computing
  - 1.2.4 Product Line Software

1.1.1 What is Software?

Program = = software ?

1.1.1 What is Software?

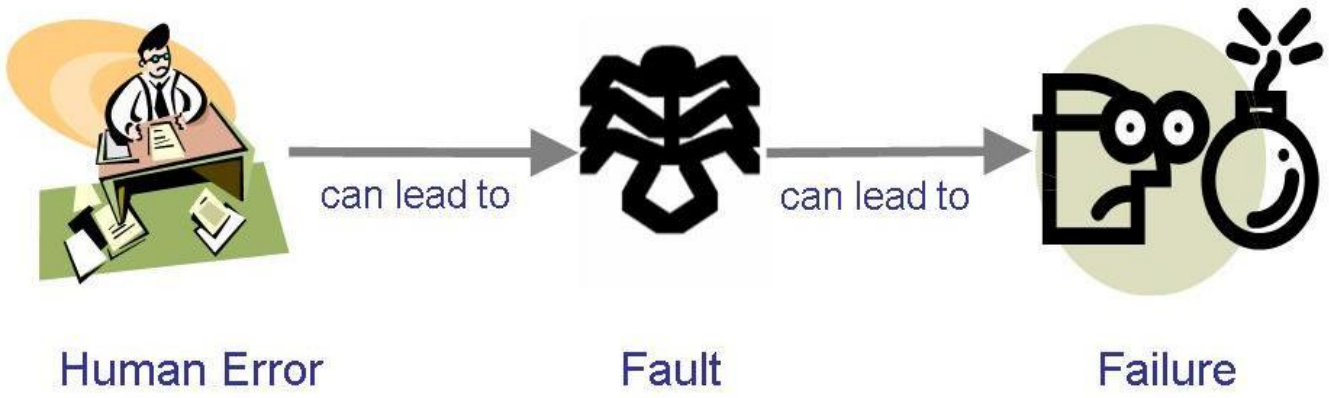


- ***Software*** *is (textbook definition):*
  - ***instructions*** *(computer programs) that when executed provide desired features, function, and performance;*
  - ***data structures*** *that enable the programs to adequately manipulate information*
  - ***documentation*** *that describes the operation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

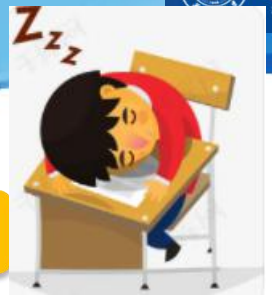
Program != software

Terminology for bugs

- **A fault**: occurs when a human makes a mistake, called **an error**, in performing some software activities (inside view)
- **A failure**: is a departure from the system's required behaviors (outside view)



Example: Fault, Failure



Count the number of “1”

Human Error

```
public class MyCount {

    public int numOne(int [] arr) {
        int count = 0;
        for (int i = 1; i < arr.length; i++)
            if (arr[i] == 1) {
                count++;
            }
        return count;
    }

    public static void main(String[] args) {
        MyCount myCount = new MyCount();
        System.out.println("number of 1: " + myCount.numOne(new int[]{1,2,3,4,5}));
        System.out.println("number of 1: " + myCount.numOne(new int[]{2,1,3,4,5}));
    }
}
```

Fault / Defect
i=1(should start from 0)

Run result

```
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
number of 1: 0
number of 1: 1
```

Failure: {1,2,3,4,5}

Success:{2,1,3,4,5}

主观题 10分

设置

What is the difference between hardware and software?

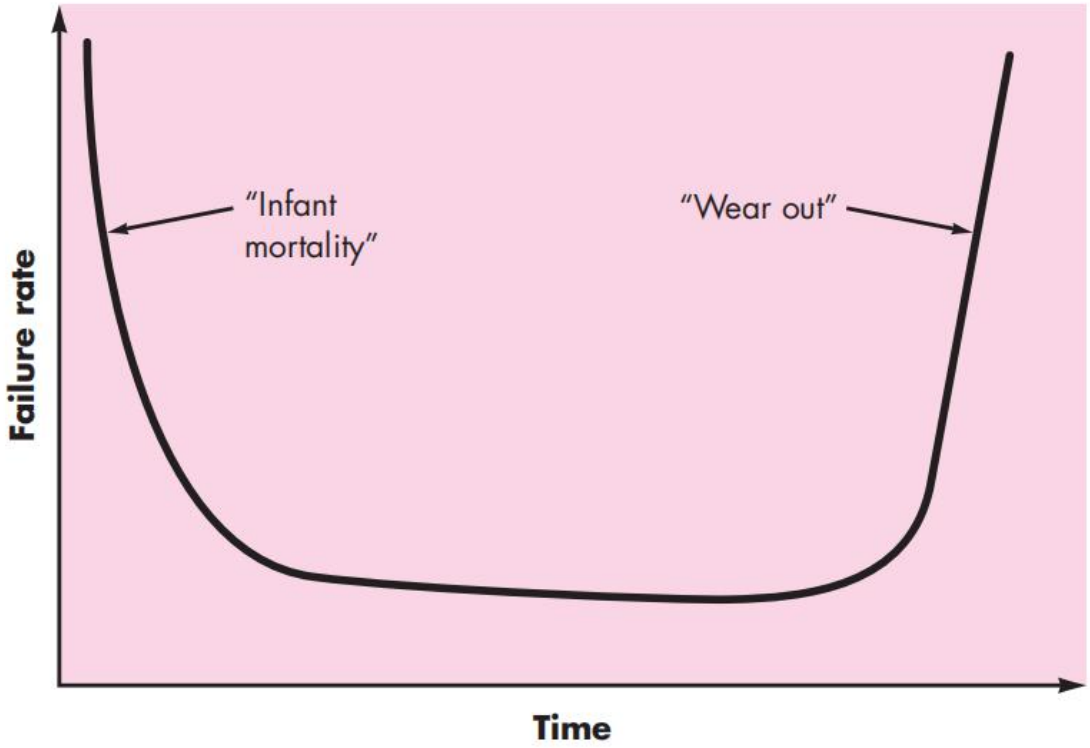
正常使用主观题需2.0以上版本雨课堂

作答

Wear vs. Deterioration (Hardware)

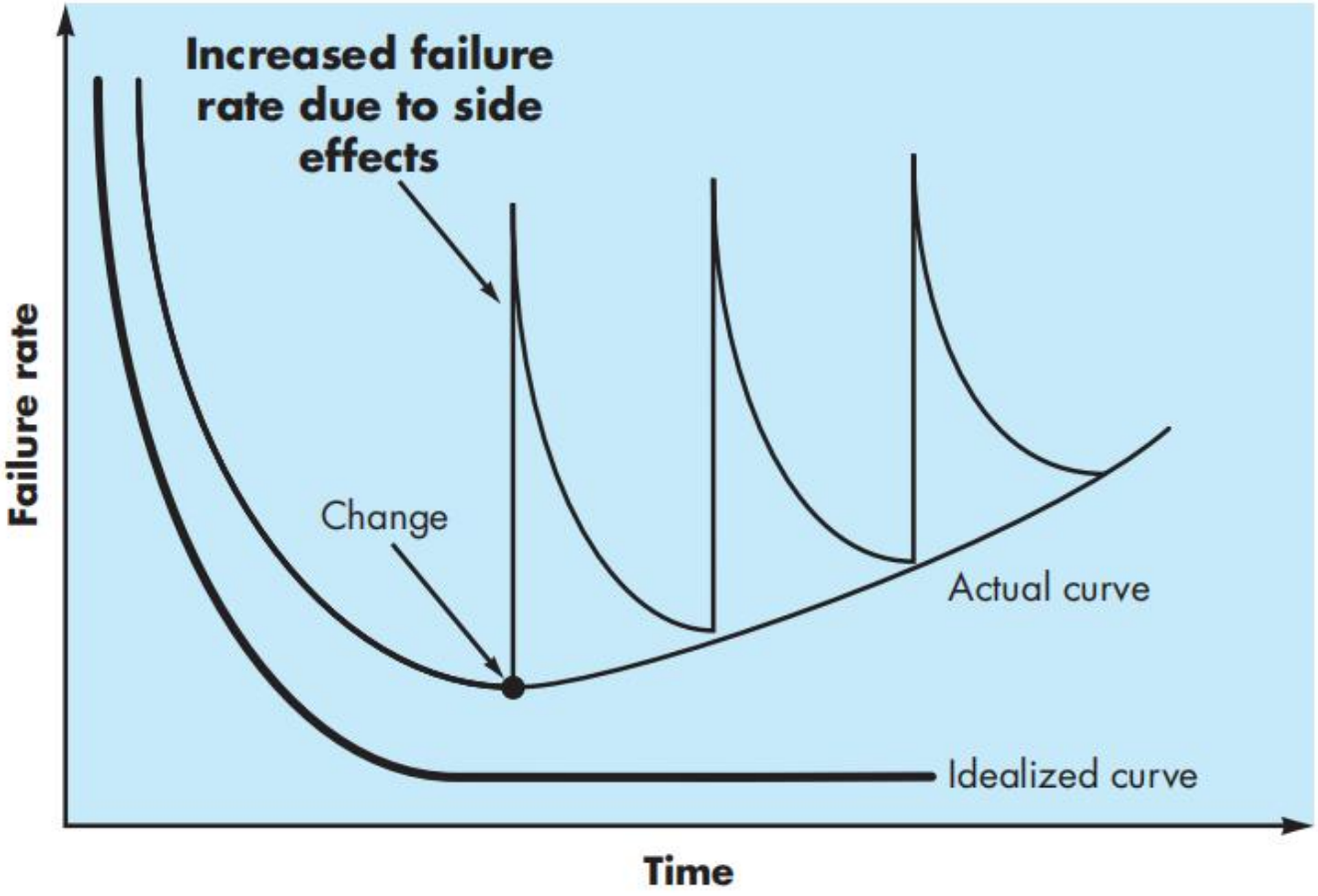
FIGURE 1.1

Failure curve for hardware



bathtub curve

Wear vs. Deterioration (Software)



1.1.1 What is Software?

- **Characteristics of software:**

1. ***Software is developed or engineered, it is not manufactured in the classical sense.***

2. ***Software doesn't "wear out." (磨损)***

3. Software **maintenance** is more complex than hardware *(replace with spare parts).*

1.1.2 Software Applications Domains

- System software (operating system, database)
- Application software
- Engineering/Scientific software
- Embedded software
- Product-line software
- Web/Mobile applications
- AI software (robotics, game playing)
- .....

多选题 1分

设置

此题未设置答案，请点击右侧设置按钮

When does software change happen?

A meet the needs of new computing environments

B implement new business requirements

C make it interoperable with other more modern systems or databases

D make it viable within a network environment

提交

1.1.3 Legacy Software - Do nothing

- *Why must 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.

Contents

- 1.1 The nature of software
  - 1.1.1 Defining Software
  - 1.1.2 Software Application Domains
  - 1.1.3 Legacy Software
- 1.2 The Changing Nature of Software
  - 1.2.1 WebApps
  - 1.2.2 Mobile/iPad Applications
  - 1.2.3 Cloud Computing
  - 1.2.4 Product Line Software

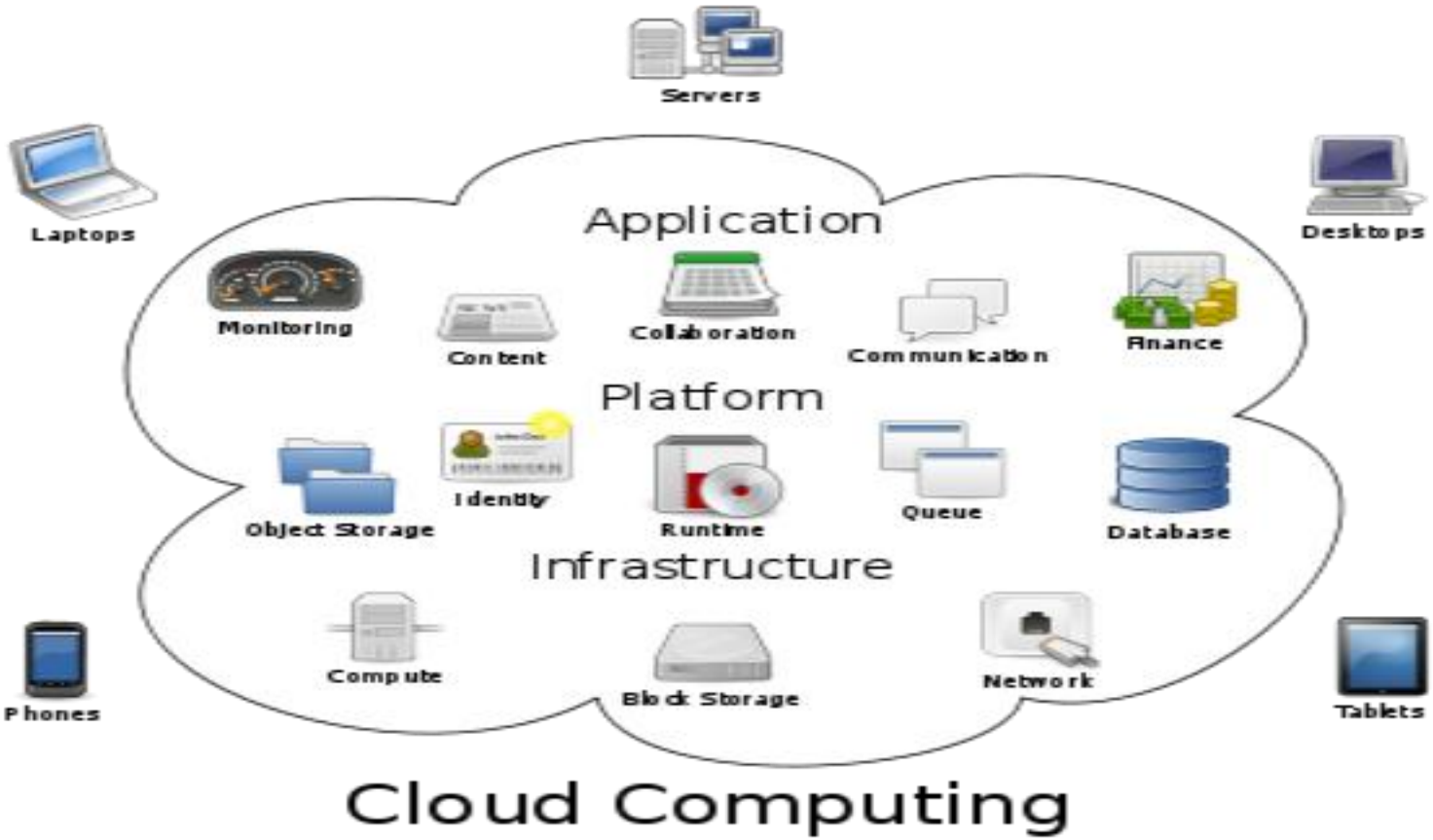
1.2.1 Characteristics of WebApps

- Modern WebApps are much more than hypertext files
- WebApps are augmented with tools like XML and Java to allow Web engineers including interactive capability
- WebApps may standalone capability to end users or may be integrated with corporate databases and business applications
- Semantic web technologies 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

1.2.2 Mobile Apps

- Contain user interfaces that take both device characteristics and location attributes
- Reside on mobile platforms such as cell phones or tablets
- 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
- 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

1.2.3 Cloud Computing



1.2.3 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

1.2.4 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.

Think: Software development

- Why does it take so long to get software finished?
- Why are development costs so high?
- Why can't we find all defects before we give the software to our customers?
- Why do we spend so much time and effort maintaining existing programs?
- Why do we continue to have difficulty in measuring progress as software is being developed and maintained?

Review

- What is software

  Definition: Instruction + data struct + documentation

  Nature: changing

- Characteristics of software:

  1. Software is developed or engineered, it is not manufactured in the classical sense.
  2. Software doesn't "wear out."
  3. Software maintenance is more complex than hardware *(replace with spare parts)*.

THE END