

I 根概念 {  
    计算机软件  
        ↓  
    软件系统  
        ↓  
    软件工程

II 影响软件开发的因素(2)

✓ 影响软件开发的因素(2)

VI. 软件退化 {  
    理想曲线  
    实际曲线  
    如何减少软件退化

III 专业软件发展

{ 活动: 商业/为用户使用, 团队开发, 预先调研, 测试, 定期维护, 具有  
    程序规范化  
性质(C)  
特点(C4)

IV 软件产品类别

{ 通用  
    定制

## Software Engineering 1

Soon Phei Tin

VI. 软件工程 {  
    因素(C3)  
    具体应用类型  
    软件过程(C4)/不同系统需要不同软件过程  
    基本原则  
    影响软件开发的因素(C3)  
    为什么软件工程重要  
    网络  
    伦理



# Objectives

- understand what software engineering is and why it is important;
- understand that the development of different types of software systems may require different software engineering techniques;
- understand some ethical and professional issues that are important for software engineers;



# Introduction

- We can't run the modern world without software.
  - National infrastructures and utilities are controlled by computer-based systems and most electrical products include a computer and controlling software.
  - Industrial manufacturing and distribution is completely computerized, as is the financial system.
  - Entertainment, including the music industry, computer games, and film and television, is software intensive.

# Introduction

难以形容的

- Software systems are abstract and intangible.
  - They are not constrained by the properties of materials, governed by physical laws, or by manufacturing processes.
- Implication? (Benefits and drawback)
  - This simplifies software engineering, as there are no natural limits to the potential of software.
  - However, because of the lack of physical constraints, software systems can quickly become extremely complex, difficult to understand, and expensive to change.



# Introduction

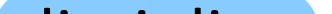
- What is a computer software?
  - *Computer Software* is a collection of instructions, data, or computer programs that are used to run machines and carry out particular activities. Software products may be developed for a particular customer or may be developed for a general market.

# Introduction

- What is Software System?

- A software system is a system that consists of a number of separate computer software, configuration files, system documentation and user documentation. ("Wikipedia, Software system")
- The term "software system" should be distinguished from the terms "computer software". The term computer software generally refers to a set of instructions that perform a specific task. However, a software system generally refers to a more encompassing concept with many more components. ("Wikipedia, Software system")
- The use of the term software system is at times related to the application of system theory approaches in the context of software engineering.

# Software Engineering

- What is software engineering?   
  - Software engineering is an engineering discipline that is concerned with all aspects of software production.
  - As a software engineer, you apply software development process when developing a software system.

# Software Engineering

- Engineering discipline - Engineers make things work. They apply theories, methods, and tools where these are appropriate. However, they use them selectively and always try to discover solutions to problems even when there are no applicable theories and methods. Engineers also recognize that they must work to organizational and financial constraints so they look for solutions within these constraints.
- All aspects of software production - Software engineering is not just concerned with the technical processes of software development. It also includes activities such as software project management and the development of tools, methods, and theories to support software production.

# Question

Which of the following are the activities when you are developing a software system?

- A. Test the software
- B. Write program
- C. Estimate the time needed to complete the program
- D. Write a user documentation

# Software Failure & Success

- Software engineering is criticized as inadequate for modern software development.
- Software failures are a consequence of two factors:
  - 2.1 • Increasing demands → 系统复杂度↑ → bug
  - 2.2 • Low expectations (you don't expect the software to grow into a giant beast) [软件初期简单开发团队目标低] → 软件发展发觉存在性能问题和设计缺陷

# Question

Which of the following are considered to be the causes to the failure of software engineering?

- A. Changing software requirements
- B. Adding software budget
- C. Starts software development immediately to save time

错误

软件开发需要充足规划和需求分析

# Software Failure & Success

- Of course, we still have problems developing complex software but, without software engineering, we would not have explored space, would not have the Internet or modern telecommunications.
- Software engineering has contributed a great deal and its contributions in the 21st century will be even greater.

# Professional software development

- Amateur

- People in business write spreadsheet programs to simplify their jobs
- scientists and engineers write programs to process their experimental data
- hobbyists write programs for their own interest and enjoyment

- Professional

企业管理系统,移动应用,

操作系统

嵌入

- developed for specific business purposes, for inclusion in other devices, or as software products such as
- intended for use by someone apart from its developer
- is usually developed by teams rather than individuals
- It is maintained and changed throughout its life.

专业软件设计目的开发者  
以外的用户使用

现代软件系统复杂需要高效  
协作(需求分析,设计,编码,  
测试,维护)

软件开发并不意味着完成就结束,定期修复漏洞,改进  
性能,添加新功能,甚至做出架构调整

# Professional software development

- Software engineering is intended to support professional software development, rather than individual programming. **设计**
- It includes techniques that support program specification, design, validation, and evolution. **验证** ← **演化**
- Professional software usually has the following properties: -
  - Strict user requirements **严格用户需求**
  - Required accuracy and data integrity **(数据完整性): 财务, 医疗, 科学研究**
  - Higher security standard **(敏感信息): 用户数据, 财务数据, 医疗记录**
  - Stable performance for heavy load **高负载: 电子商务平台, 社交媒体, 云计算**
  - Required technical support, etc.

# Question

Software engineering include the following activities: -

- A. Software team forming
- B. ✓ Software specification
- C. Software selection
- D. ✓ Software validation

部分 GPT 选择我偏向定义上没写入选

# Professional software development

Product characteristics	Description
Maintainability  可维护性	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security  可靠性和安全性	Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency  (针对系统利用) 效率	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilization, etc.
Acceptability  用户能理解你的设计 可接受性	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use.

# TWO kinds of software products

通用

- Generic software products

- These are systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.

定制

- Customized software products 委托

- These are systems that are commissioned by a particular customer. A software contractor develops the software especially for that customer. 承包商

- The distinction between these system product types is becoming increasingly blurred

# Question

Which of the following can be considered as a generic software?

- A. Image processing software like Adobe Photoshop
- ~~B. Logistics software for the Chinese Army~~
- C. Language translation mobile app in the Apple Store

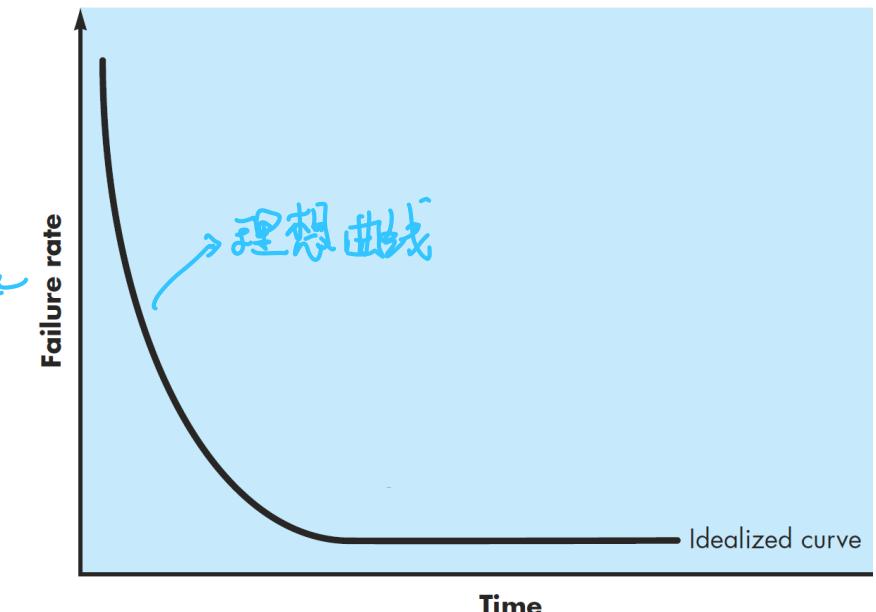


# Software Development challenges

- Why does it take so long to get software finished? 长时间 ← 功能复杂性 / 需求变化
- Why are development costs so high? 成本高 ← 技术人才成本 / 开发工具设备资金 / 测试维护 测试耗时长
- Why can't we find all errors before we give the software to our customers? bug 多 ← 测试局限性无法覆盖所有场景 / 软件复杂性 / 项目期限限制 成本
- Why do we spend so much time and effort maintaining existing programs? 维护现有项目需要大量时间精力 ← 软件需要兼容新操作系统 / 需求变化 / 相比更换系统更新
- Why do we continue to have difficulty in measuring progress as software is being developed and maintained? 系统更容易  
开发和维护过程很难衡量进度 ← 软件开发非线性 (设计 / 测试需要大量时间), 不确定性 (需求变更), 测试工具不足

# Software Deterioration 软件退化

- Software is a logical rather than a physical system element. Therefore, software has one fundamental characteristic that makes it considerably different from hardware:  
*Software doesn't "wear out."*
- In theory, therefore, the failure rate curve for software should take the form of the "idealized curve" shown in Figure

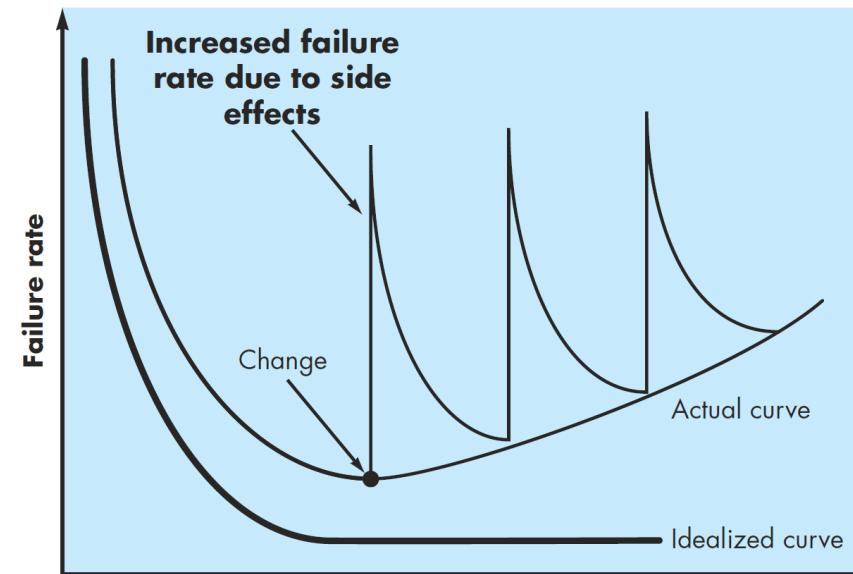


# Software Deterioration

- Software doesn't wear out. But it does **deteriorate** **但软件会退化**
- During its life, software will undergo change. As changes are made, it is likely that errors will be introduced, causing the failure rate curve to spike as shown in the "actual curve" ( Figure 1.2 ).
- Before the curve can return to the original steady-state failure rate, another change is requested, causing the curve to spike again.

软件工程方法目标是减小这些尖峰幅度和频率，更接近于理想曲线

Software engineering methods strive to reduce the magnitude of the spikes and the slope of the actual curve



实际曲线随着时间逐步上升 ← 频繁修改引入新问题 / 复杂度增加 / 过往缺陷(技术债务) / 环境变化

# Software Deterioration (如何减少软件退化)

- To reduce software changes: -

- Work closely with the stakeholder to ensure requirements are correctly defined  
利益相关者
- Improve requirement study approach to achieve better requirements study

- To reduce side effects after changes: -

- The software should be modular so that changes will not have a lot of side effects to other part of the software
- The software must be maintainable
- Comprehensive testing should put in place to reduce errors.

全面的  
↓

- Work closely with stakeholder for fully understanding requirements
- achieve better requirement study

# Software Engineering Factor<sup>^</sup>

- Software engineering is a systematic approach to the production of software that takes into account practical cost, schedule, and dependability issues, as well as the needs of software customers and producers.
- How this systematic approach is actually implemented varies dramatically depending on: -
  - The organization developing the software
  - The type of software
  - The people involved in the development process.

# Type of Software Engineering Applications

- Perhaps the most significant factor in determining which software engineering methods and techniques are most important is the type of application that is being developed.

→ 独立安装运行无需联网，用户界面是关键（文本编辑器）

• Stand-alone applications 独立应用

• Interactive transaction-based applications 交互式事务应用

• Embedded control systems 嵌入式控制系统

• Batch processing systems 批处理系统

多个用户交互（查询/输入/结果快速返回）（图书馆管理

系统/银行系统/电商平台）

→ 音频、视频、用户体验  
(游戏、流媒体)

• Entertainment systems

• Systems for modeling and simulation 模型（交通流量/气象）

• Data collection systems 数据采集系统（工厂/医疗监测）

→ 集成在硬件的软件系统，不需要用户操作（汽车引擎，空调温度控制系统）

→ 处理大量数据分批执行（工资计算系统，银行对账系统）

# Software Engineering Approaches

Software Process

- The systematic approach that is used in software engineering is sometimes called a software process.
- A software process is a sequence of activities that leads to the production of a software product.
- There are four fundamental activities that are common to all software processes: -
  - Software specification 软件规范
  - Software development 软件开发
  - Software validation 软件验证
  - Software evolution 软件演进

# Software Engineering Approaches

- Different types of systems need different development processes.
- For example, embedded control system in an aircraft has to be completely specified before development begins. In e-commerce systems, the specification and the program are usually developed together.

有些系统对安全性可靠性要求高必须严格规划和验证(飞机嵌入式)

有些系统需要快速开发灵活调整适应市场需求(电商平台)

# Software engineering fundamentals

- Some fundamental principles apply to all types of software system, irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process.
  - Dependability and performance are important for all types of system.
  - Understanding and managing the software specification and requirements (what the software should do) are important.
  - Where appropriate, you should reuse software that has already been developed rather than write new software.

→ 节省时间和成本 / 减少错误 / 提高效率

# General issues that affect most software

- **Heterogeneity** 异构性 (软件需要适配多种软硬件平台)

- Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.

- **Business and social change** → 快速迭代/新功能开发 → 满足需求

- Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

- **Security and trust**

- As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

# Why software engineering important?

- More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.
- It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of systems, the majority of costs are the costs of changing the software after it has gone into use.

大多数系统的成本主要是后期的维护成本

# Software engineering and the web

- The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems. 用户只需通过浏览器访问，无需关心底层操作系統和硬件环境
- Web services allow application functionality to be accessed over the web. 允许不同应用程序跨平台跨设备交互
- Cloud computing is an approach to the provision of computer services where applications run remotely on the ‘cloud’.
  - Users do not buy software but pay according to use.

Web 万维网 建立在 Internet 应用层上 基于 http 协议，利用 html 编写网页，CSS 定义网页视觉样式，JavaScript 实现网页交互功能，利用 Web 浏览器加载 Web 内容  
通过超链接从一个网站跳转到另外一个网站

按需付费

云服务

# Web-BASED software engineering

- ① • Software reuse is the dominant approach for constructing web-based systems.  
    • 框架重用：Django/React 快速搭建核心功能和落地界面（处理数据，用户认证，端点）而不是从头写代码  
    • API 使用：使用 PayPal 提供的 API 处理付款，使用 Google Maps API 嵌入地图
  - When building these systems, you think about how you can assemble them from pre-existing software components and systems.
- ② • Web-based systems should be developed and delivered incrementally.  
    • development 将系统拆分成小的可管理的部分，每次发布一个工作版本，确保用户/开发者快速看到效果，及时调整
- ③ • User interfaces are constrained by the capabilities of web browsers.  
    • 用户界面受限于 Web 浏览器能力
  - Technologies such as AJAX allow rich interfaces to be created within a web browser but are still difficult to use. Web forms with local scripting are more commonly used.

Asynchronous JavaScript and XML

difficult 指实现异步请求，处理数据，更新页面内容

# Question

Why a web-based systems should be developed and delivered incrementally?

- 技术不影响力开发方式
- A. It is normally run on a web browser
  - B. The user is normally diverse, requirements can be unpredictable
  - C. To allow the developers to learn from the feedback as they are developing the system
  - D. The web is too large  
Web 开发的确拆分成多个模块, 但增量开发来源需求的快速变化

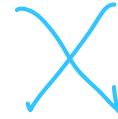
# Web-based software engineering

- Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.

※ Web 系统本质上是复杂的分布式系统 (前端运行在用户浏览器上, 后端服务分布在不同服务器上, 数据库存储在独立系统)

↓ 需求分析 / 模块化设计 / 版本控制与协作 / 测试 / CI/CD 流水线 →  
持续集成与交付 / 维护

# Software Engineering Ethics



- Like other engineering disciplines, software engineering is carried out within a social and legal framework that limits the freedom of people working in that area.
- As a software engineer, you must accept that your job involves wider responsibilities than simply the application of technical skills.



*"With great power comes  
great responsibility"*  
~Voltaire



Compiled by NetSysCon find more inspiring quotes at  
<http://netsyscon4hr.wordpress.com/category/quotes-by-the-greats/>



**With Great Power**



**Comes  
great responsibility**

# Software Engineering Ethics

- Ethics, as understood here, addresses any intentional action that impacts negatively or positively the lives and values of others.
- The ethical activity involved in technical decisions should be based on an understanding of the impact of those decisions.
- Software engineers have obligations to the users of their products, which include not only the implemented system but also includes other products such as requirements, software project management plans, specifications, designs, documentation, test suites, programs, user manuals, and training materials. (Software Engineering Ethics, 2002, Donald Gotterbarn)



# Software Engineering Ethics

- You must also behave in an ethical and morally responsible way if you are to be respected as a professional engineer.
- You should not use your skills and abilities to behave in a dishonest way or in a way that will bring disrepute to the software engineering profession.

# Software Engineering Ethics

- There are areas where standards of acceptable behavior are not bound by laws but by the more tenuous notion of professional responsibility. Some of these are:
  - Confidentiality 保密性
  - Competence (未具备相关能力的人员不应对承担超出自己专业范围的工作)
  - Intellectual property rights 知识产权
  - Computer misuse (未经授权的访问,篡改数据,发布恶意软件)

# Software development risk

- 1) • Many software development projects run into difficulties
  - Does not work as expected 不满足客户需求
  - Over budget 超出预算
  - Late delivery 延迟交付
- 2) • Much of the functionalities in the software is wasted 功能浪费
  - Wrong requirement
  - User dislike it
  - There are no customer
  - etc

# Software Development Risk

- Many software projects fail because the software developers build the wrong software. The software development team must: -
  - Fully understand requirement
  - Validate requirement
- The developer will often add technical insights and suggestions, but remember the client satisfaction is the primary measurement of success in software project.

# Software Development Risk – The danger

破产

- Failures of software development projects can bankrupt companies.
- What are the consequences if the development of a software: -
  - Late?
  - Over budget?
  - Does not work or full of bugs?

# Software Engineering vs Computer Science

## Software Engineering

- Concerned with the practicalities of developing and delivery professional software system

## Computer Science

- Focuses on the theory and fundamentals

The End  
Thank you