



Ch.0 About the Course

(Software Engineering, SE)

Zhangye Wang

zywang@cad.zju.edu.cn

QQ:1282842242

2015春软工QQ群: 372424807

Zhejiang University

Mar. 9, 2015





Teacher

Zhangye Wang(王章野)

---E-mail: zywang@cad.zju.edu.cn

---Office: Rm. 521, Building Mengminwei(蒙民伟楼),
Zijingang Campus, Zhejiang Univ.

---Phone: 15858127288(Cell); 88206681ext.519(Office)

---Website: <http://www.cad.zju.edu.cn/home/zywang/>

---Course: Time-- Class 1,2, Monday, 8:00am—9:35am;
Room—Building 7-604

---Experiment: Time-- Class 3,4, Sunday , 9:50am—11:25am (Single week)
Room--曹西-503





About myself

- Research Interests:
--- CG (Computer Graphics)





About myself

- Research Interests:
 - CG (Computer Graphics)
 - VR (Virtual Reality)





About myself

- Research Interests:
 - CG (Computer Graphics)
 - VR (Virtual Reality)
 - Game Design





About myself

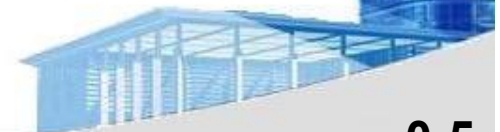
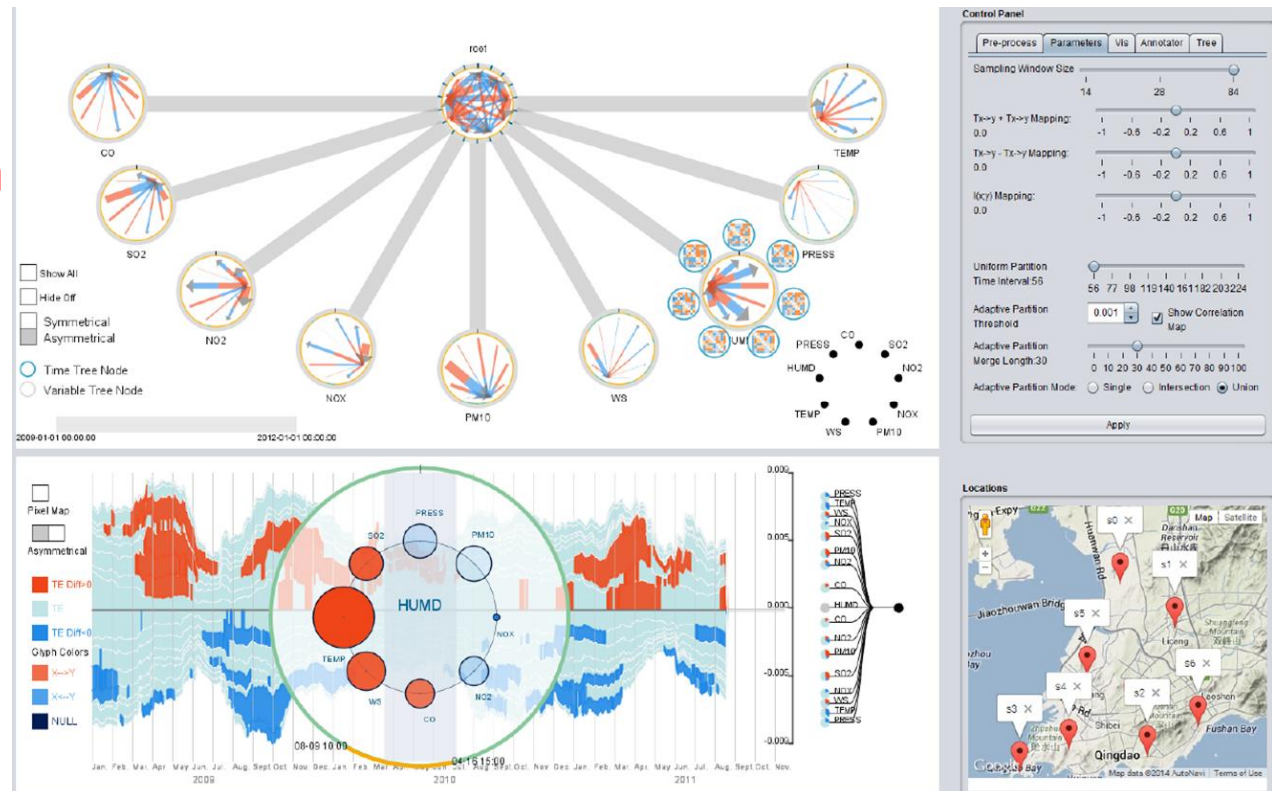
- Research Interests:

- CG (Computer Graphics)

- VR (Virtual Reality)

- Game Design

- Big Data Visualization





About myself(2)

- Publication & Honors

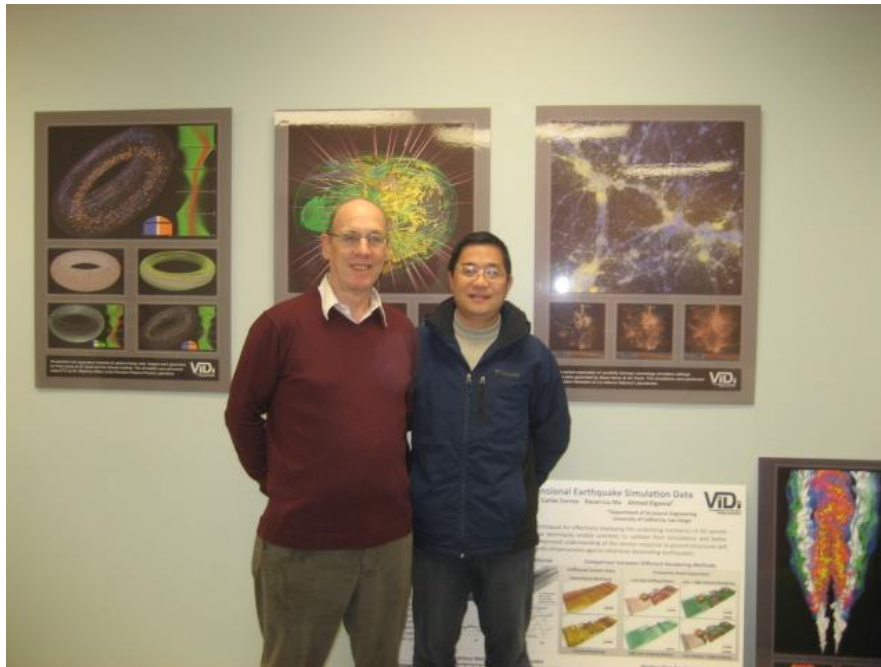
- Published About 100 academic papers;
- Won Best Paper Award of ChinaGraphic three times;
- Won Lu Zengyong High-Tech. Prize of CAD&CG(2003)





My Visiting Experience

---Sept. 2010-Sept. 2011, as a
Visiting Scholar of Computer
Science Department, UC Davis,
USA



With Prof. Gouraud at Vidi Lab., UC Davis



At Facebook





About Assistant (本课程助教介绍)

Shaoxiang Zhang(张少雄)

E-mail: zsx19931129@163.com

手机: 18757135685

- 个人简介:
- 2014级计算机应用专业硕士生。
- 本科软件工程专业毕业。
- 发表论文: L_0 梯度极小化耦合梯度保真的结构图像平滑, 中国科学(信息科学), 2014, 第44卷,第11期, pp.1370-1384





Web Source

- Course website: <http://www.cs.zju.edu.cn/se/>



The screenshot shows the homepage of the Zhejiang University Software Engineering Course Website. The header is blue with the university logo and name on the left, the site title '软件工程课程网站' in the center, and a red 'PPT 下载' button on the right. A navigation bar below the header contains links for '首页', '教学大纲', '课堂教学', '课堂录像', '课程实验', '课后讨论', and '课后习题'. A '趣文推荐' section is also present. The main content area is titled '课程导航' and lists three course features: 1. Practice and assessment interaction, 2. Research and teaching interaction, and 3. Network and on-site learning interaction. Each feature is followed by a brief description of its goals and methods.

课程导航

课程特色之一：实践与考核互动，以实际案例和创新性激励机制带动学生学习积极性

针对课程教学很容易流于形式的问题，可以考核方法改革与教研结合为手段，以大型实验项目模拟真实环境，激发学生自主性学习，达到全面培养现代软件工程师能力的目的。

课程特色之二：科研与教学互动，科研为教学及时补充前沿理论，教学为科研输送得力干将

针对软件工程知识点更新快的问题，解决的重点在于使教学队伍年轻化，教师注重将最新科研成果融入教学工作，以科研促进教学，才能不断更新教学内容，紧跟学科发展的前沿。

课程特色之三：网络与现场学习互动，全方位满足学生即时沟通和个性化学习的需要

针对单一教学手段不能满足学生团队沟通与个性化学习需要的问题，我们认为网络化教学手段可以成为对传统教学手段的重

- Where including: Course PPT, Video, Schedule, Tests, Forum, etc.





Teaching materials & References

- 教材:

《软件工程——实践者的研究方法》（英文名：
Software Engineering – A Practitioner's
Approach）英文版第8版，Roger S.
Pressman，清华大学出版社，2015.2.11

教材（书号48950）网购地址（119元）：
<http://www.hzbook.com/Books/8284.html>





软件工程：实践者的研究方法（英文版·第8版）



 **购买**

作者: (美) Roger S. Pressman Bruce

语种: 简体中文

R. Maxim 著

开本: 16

ISBN: 978-7-111-48950-4

原书名: Software Engineering: A Practitioner's Approach

定价: 119.00

属性分类: 教材

页数: 968

所属丛书: 经典原版书库

出版日期: 2015年02月11日

责任编辑:

译者: 无

适用专业:

图书分类: 计算机 > 软件工程及软件方法学

绝版: 否

原出版社: McGraw-Hill

包含CD: 无

浏览量: 29

加入我的书架

发表书评

发表勘误

联系编辑

图书摘要 ○ 图书特色 ○ 上架指导 ○

本书自1982年发行第1版以来，一直受到软件工程界的高度重视，成为高等院校计算机相关专业软件工程课的重要教学参考书。近30年来，它的各个后继版本一直都是软件专业人士熟悉的读物，在国际软件工程界享有无可质疑的权威地位。它在全面而系统、概括而清晰地介绍软件工程的有关概念、原则、方法和工具方面获得了广大读者的好评。此外，本书在给出传统的、对学科发展具有深刻影响的方法时，又适当地介绍了当前正在发展的、具有生命力的新技术。



其它参考书

1. 《软件工程课程设计》，杨小虎等著, 浙江大学出版社 (2007)
2. 《Software Engineering, Theory and Practice》 Shari Lawrence Pfleeger Prentice-Hall, Inc. (1998)
3. 《软件工程导论》（第三版），张海藩，清华大学出版社（1997）
4. 《实用软件工程》（第二版），郑人杰、殷人昆、陶永雷，清华大学出版社（1996）
5. 《Fundamentals of Software Engineering》 Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Prentice-Hall, Inc. (1991)





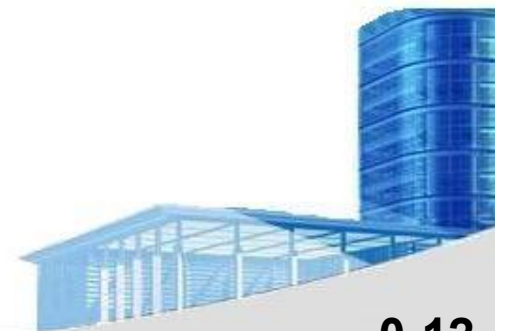
Post-courses & Trendency

---Some Post-courses

- 软件文档写作;
- 项目管理与案例分析;
- 软件需求分析与设计

---Trendency

- 2011年9月, 国务院学位委员会宣布: 软件工程成为一级学科, 与计算机科学并列。
- 软件工程系, 服务计算与云计算.....





Schedule

周次	教学内容	课时	课外作业	课外课时
1	Ch.1 The Nature of Software	2	习题集1	2
2	Ch.2-4 The Process	2	习题集2-4	2
3	Ch.31 Project Management; Ch.6 Human Aspects; Ch.7 Modeling Principles	2	习题集31, 6-7	2
4	Ch.8-9 Requirements: Concepts & Scenario	2	习题集8-9, 布置需求报告	2
5	Ch.10-11 Requirements: Class & others	2	习题集10-11	2
6	Ch.19 Quality Concepts; Ch.12 Design Concepts	2	习题集19-20	2
7	Ch.13 Architectural Design	2	习题集13, 收需求; 布置设计报告;	2
8	Ch.17 WebApp Design; Ch.18 MobileApp Design; Ch.20-21 Review & SQA	2	习题集17-18, 20-21	2
9	Ch.14 Component-level Design ; Ch.15 UI Design ; Ch.16 Pattern-based Design	2	习题集14-16, 布置设计模式研究报告	2
10	总体设计报告演讲	2		2
11	Ch.29 Configuration Management; Ch.22 Testing Strategies	2	习题集12, 29, 布置测试报告	2
12	Ch.23-24 Testing Conventional & OO Apps	2	习题集23-24, 布置v1.0; 收设计模式研究报告	2
13	Ch.25-26 Testing for WebApp & Mobile App	2	习题集25-26, 收测试	2
14	Ch.27 Security Engineering; Ch.28 Formal Methods*; Ch.36 Maintenance	2	习题集27-28, 36; 收v1.0; 布置v2.0	2
15	Ch.34 Scheduling; Ch.35 Risk; Ch.30 Product Metrics	2	习题集34-35, 30, 进行Web Speech演讲	2
16	Ch.5 Agile; Ch.32 Project Process Metrics; Ch.33 Estimation	2	习题集5, 32-33; 收v2.0; 布置合并版	2



Grading & Requirement

- **Grading**

1. **Class Quiz**: 20%
2. Project: 40%
3. Final Exam: 40%

- **Requirement**

- Preview (English edition preferred)
- Practice & Cooperation
- Active & communicative





About Project

- All students (About 57) are divided into A , B Teams;
- **Each Team** includes 6 groups;
- **Each group** has 4~5 members.
- **Note:** Elect Team leaders **Laboratory Project Title: Teaching Service System (TTS)**
- (软件工程课程实验项目:教学服务系统)

The **Teaching Service System** is composed of 6 subsystems:

The **Teaching Service System** is based on the university network to provide service for the teaching activities. The system is composed of 6 subsystems:

- 1)information management,
- 2)automatic course arrangement,
- 3)course selection,
- 4)course resource share,
- 5)online testing,
- 6) score management.





About Project (2)

1. 基础信息管理子系统
2. 自动排课子系统
3. 选课子系统
4. 资源共享子系统
5. 在线测试子系统
6. 成绩管理子系统





Ch.1 The Nature of Software





1.1 The Nature of Software

- The Evolution

In the early days: User \leftrightarrow Computer

Software = “Place a sequence of instructions together to get the computer to do something useful”.



In late 1950's: User \leftrightarrow **Programmer** \leftrightarrow Computer

Computer became cheaper and more common.

High level languages were invented.

In early 1960's: **Hacker** \neq **Cracker**

Very few large software projects were done by some experts.





1.1 The Nature of Software

Case 1. 美国IBM公司在1963年至1966年开发的IBM360机的操作系统。这一项目花了**5000人-年**的工作量，最多时有**1000**人投入开发工作，写出了近**100万**行源程序。据统计，这个操作系统每次发行的新版本都是从前一版本中找出**1000**个程序错误而修正的结果。

这个项目的负责人**F. P. Brooks**事后总结了他在组织开发过程中的沉痛教训时说：“...正像一只逃亡的野兽落到泥沼中做垂死的挣扎，越是挣扎，陷得越深，最后无法逃脱灭顶的灾难。...程序设计工作正像这样一个泥潭，...一批批程序员被迫在泥沼中拼命挣扎，...谁也没有料到问题竟会陷入这样的困境...”。

IBM360操作系统的历史教训成为软件开发项目的典型事例为人们所记取。而Brooks博士随后写出了软件工程领域的经典著作《**人月神话**》(The Mythical Man-Month)，至今畅销不衰。





1.1 The Nature of Software

Software = Product (information transformer)
Vehicle for delivering a product (OS, network, tools)

? The same questions are still asked today:

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





1.1 The Nature of Software

- **What is Software?**

Software is a set of items or objects that form a **configuration** that includes

- ☞ **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.

AND MORE ...

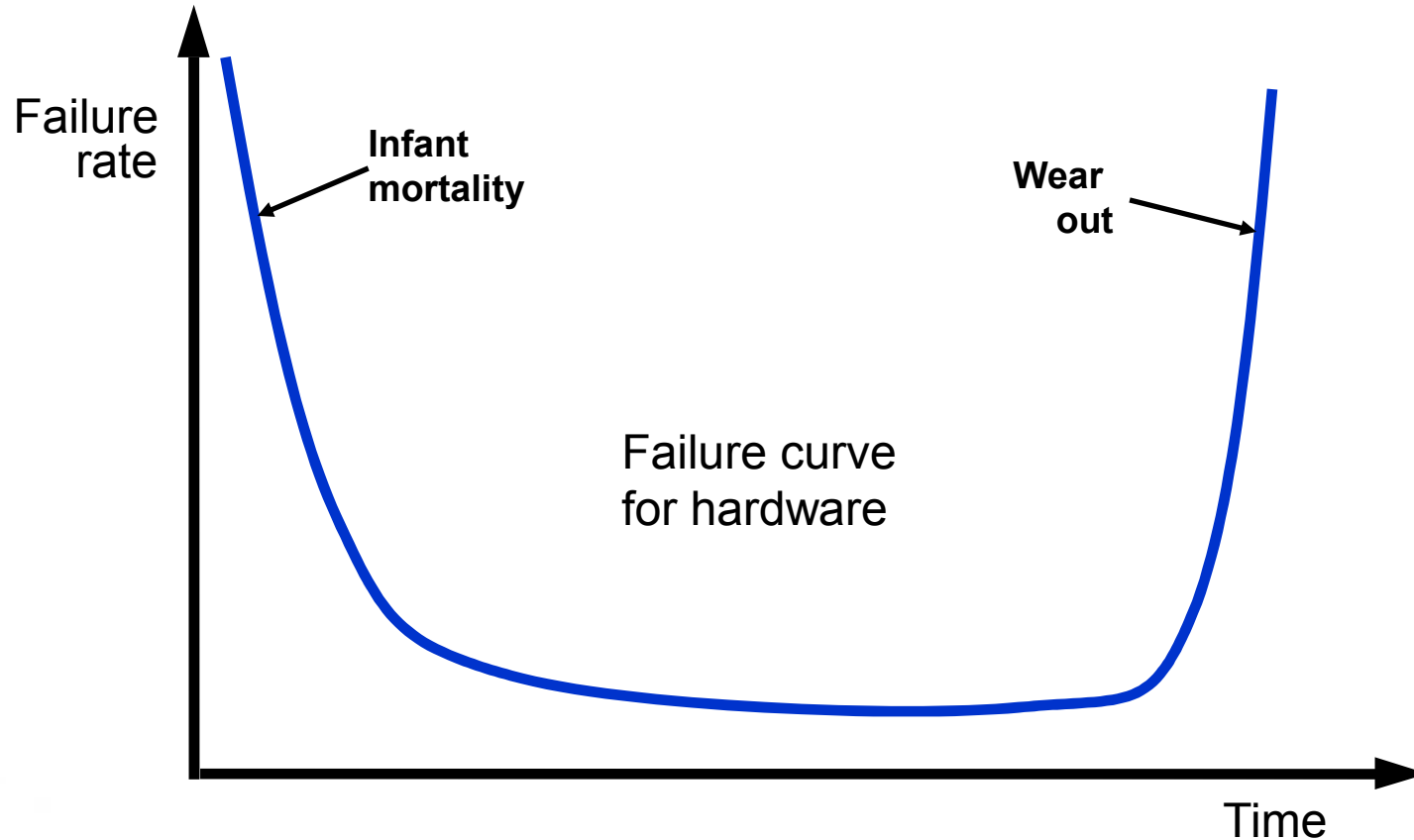
- **Software is developed or engineered, it is not manufactured in the classical sense.**



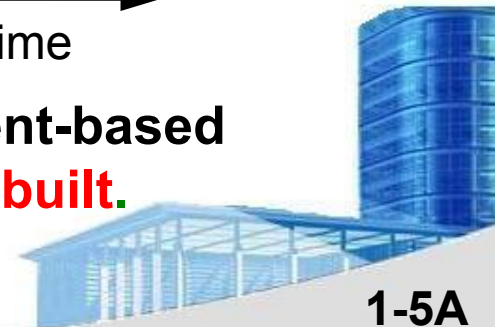


1.1 The Nature of Software

- Software doesn't **wear out**. But it does **deteriorate**!



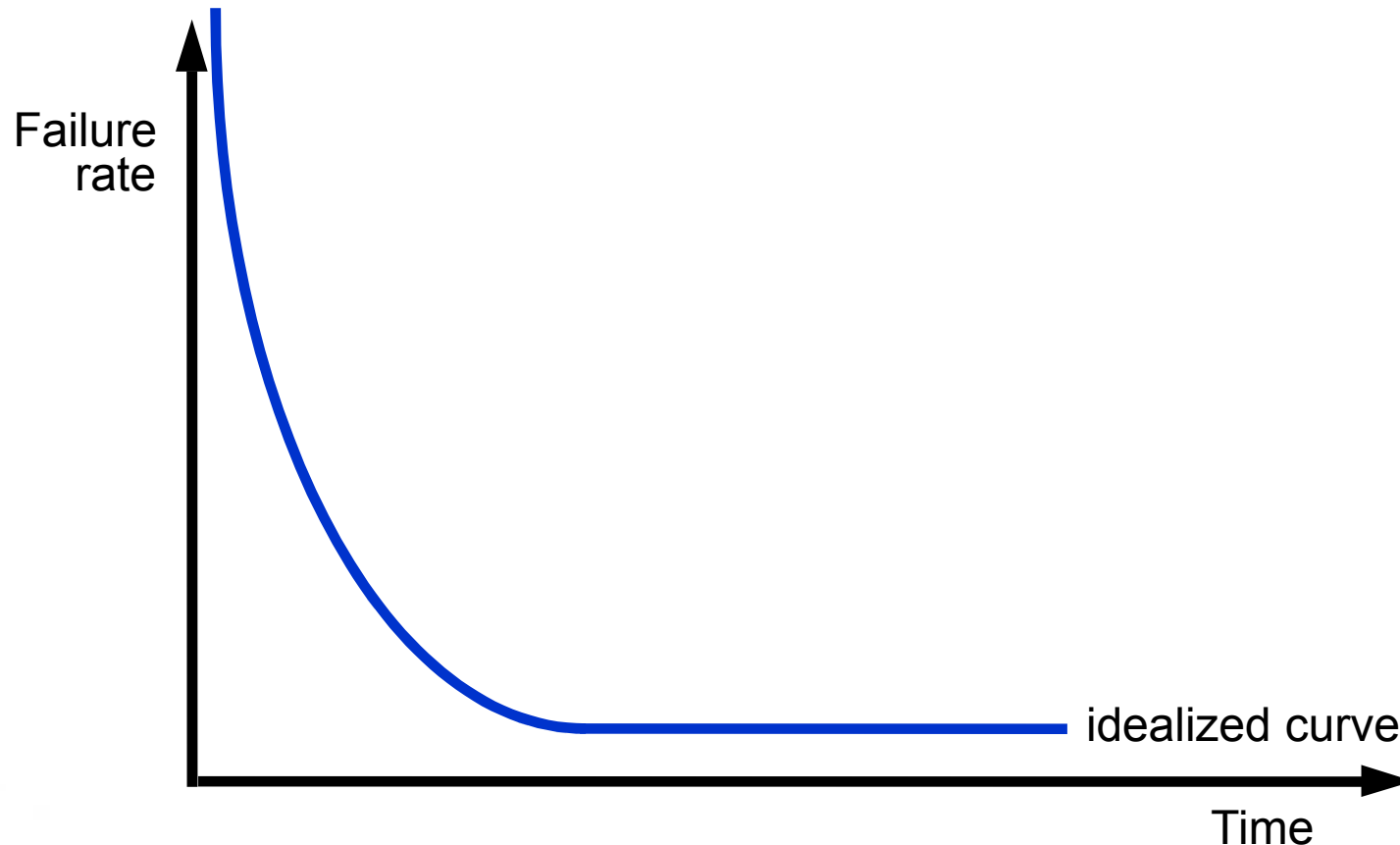
- Although the industry is moving toward component-based assembly, most software continues to be **custom built**.



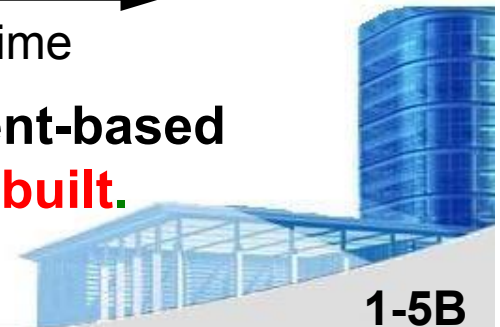


1.1 The Nature of Software

- Software doesn't **wear out**. But it does **deteriorate**!



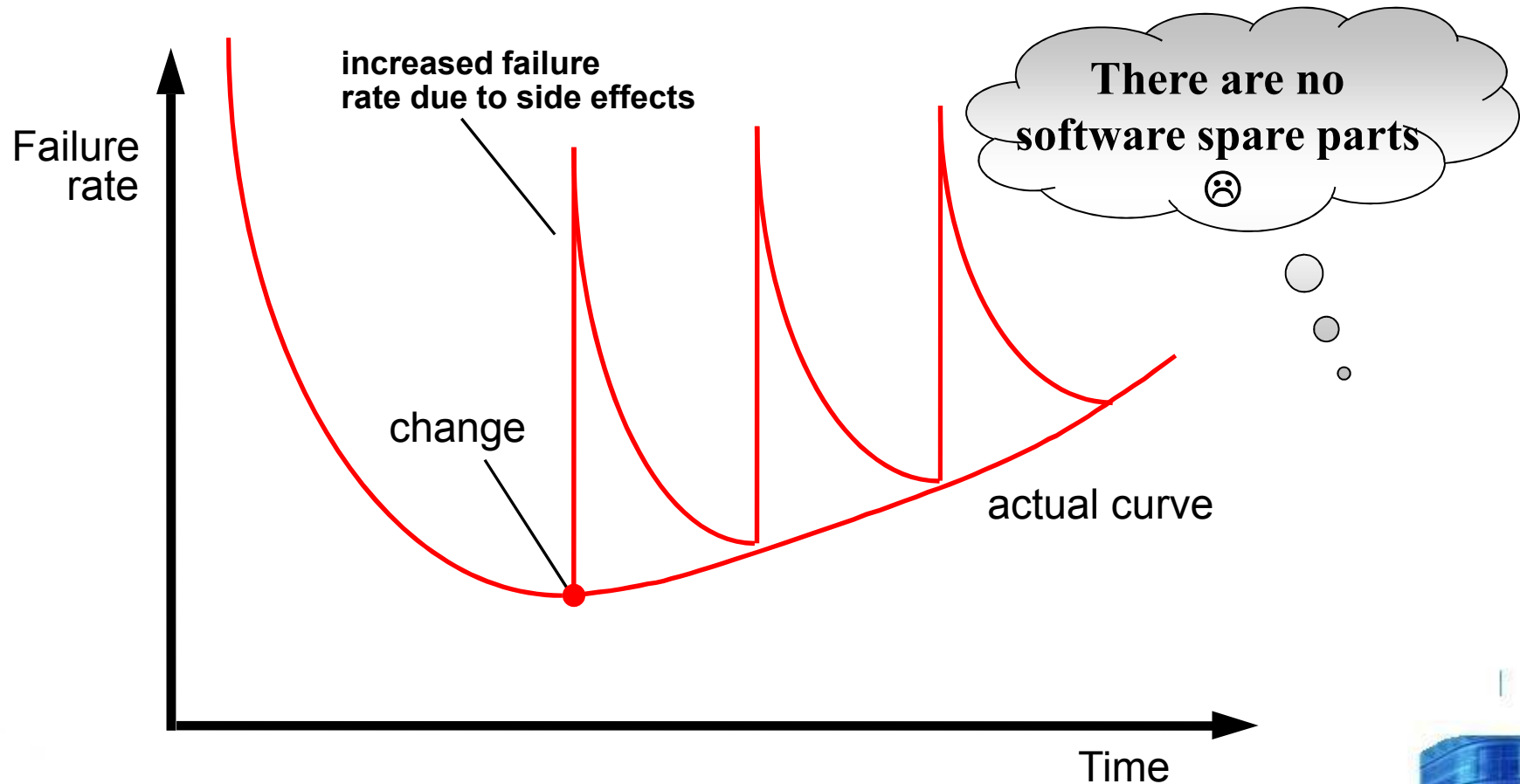
- Although the industry is moving toward component-based assembly, most software continues to be **custom built**.





1.1 The Nature of Software

- Software doesn't **wear out**. But it does **deteriorate**!



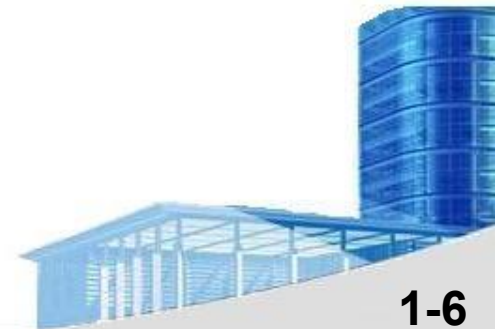
- Although the industry is moving toward component-based assembly, most software continues to be **custom built**.





1.1 The Nature of Software

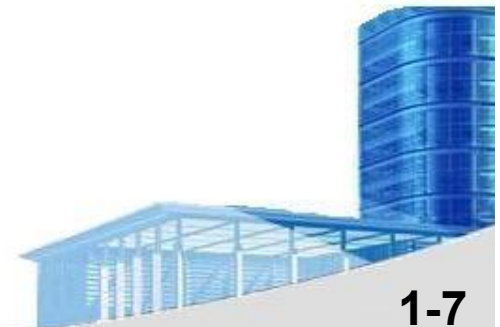
- **Software Application Types**
 - **System software**
 - **Application software**
 - **Engineering/Scientific software**
 - **Embedded software**
 - **Product-line software**
 - **Web-applications**
 - **Artificial intelligence software**





1.1 The Nature of Software

- **Legacy** Software — Why must it 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** (切实可行的) within a network environment.





1.2 The Changing Nature of 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** (脱机/独立的) 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.





1.2 The Changing Nature of Software

• Mobile Applications

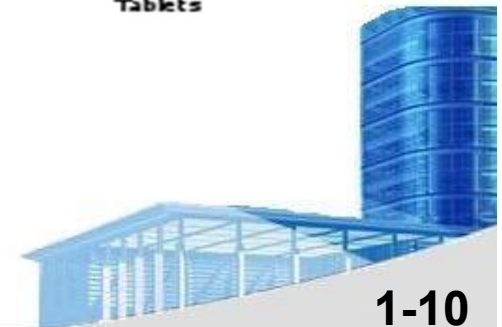
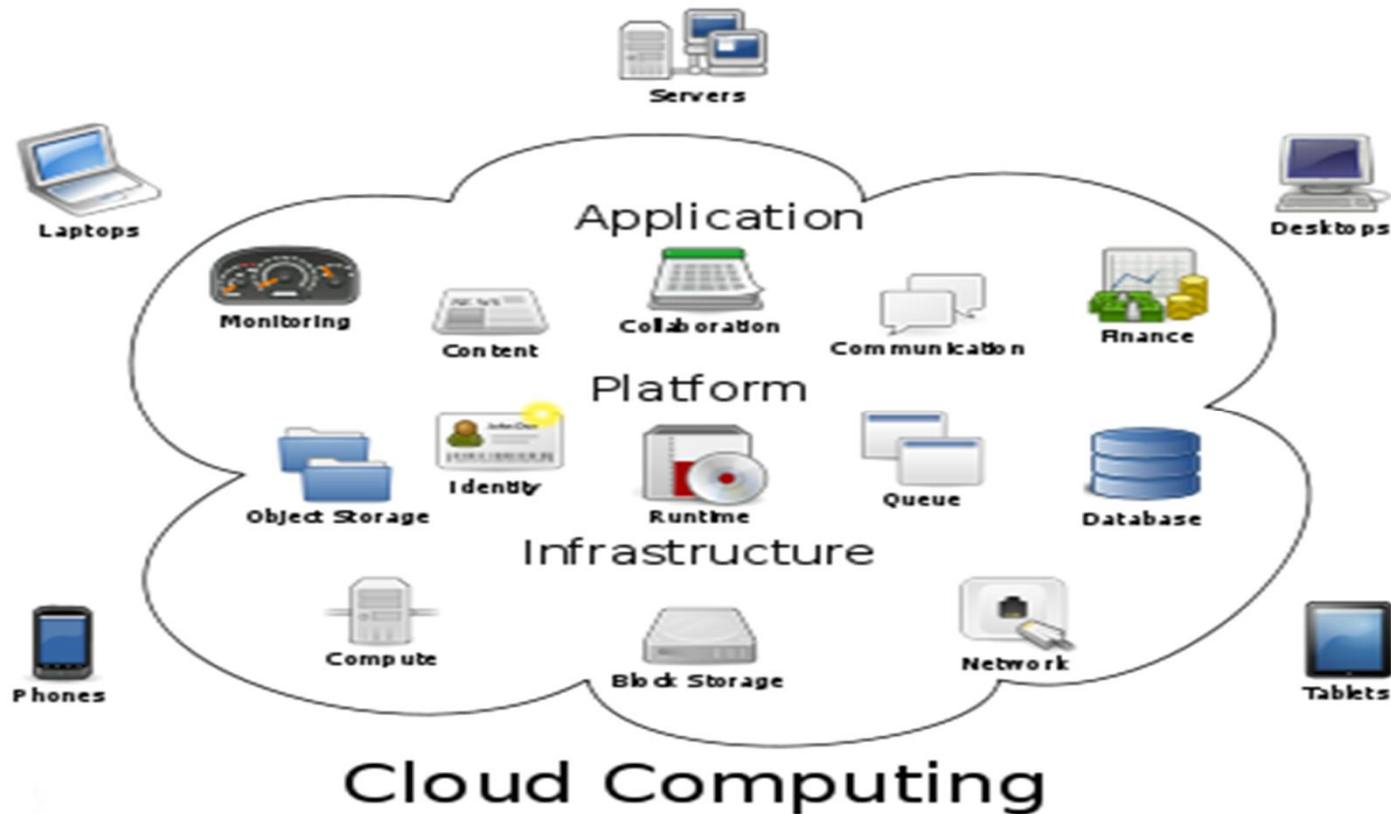
- Reside on mobile platforms (e.g. **iOS**, **Android**, Windows **Mobile**) such as cell phones or **tablets**
- Contain user interfaces that take both **device** characteristics and **location** attributes
- Often provide access to a combination of web-based resources and local device processing and 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 (**WebApps** vs. **mobile app**) will become blurred





1.2 The Changing Nature of Software

- Cloud Computing





1.2 The Changing Nature of Software

• 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 The Changing Nature of Software

• 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





Tasks

- Join /organize a group(4~5 persons) ;
- Elect group leaders!
- Review Ch. 1; Preview Ch. 2-4
- Finish “Problems and points to ponder” in pp.12 (Ch. 1)
- Join class QQ group:372424807 !

*张少雄: zsx19931129@163.com; 手机: 18757135685

