

软件工程

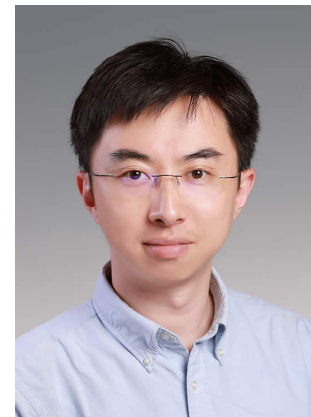
10102302

赵君峤 长聘副教授

计算机科学与技术学院

同济大学

课程团队简介



- 赵君峤 长聘副教授

- cs1.tongji.edu.cn/~Junqiao
- 主持同济大学“TiEV途灵无人驾驶系统软件”设计与开发
- 主持同济大学-上汽、华为、中兴、理想等联合项目软件系统设计与开发
- 参与国家重点863计划“三维城市地理信息系统与示范应用”软件设计与开发

- 叶晨 教授级高级工程师

- 主持同济大学“基于深度学习的颈部肌张力障碍痉挛责任肌肉诊断软件”、“基于实时通信浮标的海底设施智能预警与防护系统”、“面向生活场景的汉字字体生成软件”
- 参与国家“核高基”重大专项“汽车电子实时嵌入式操作系统的系统服务与开发环境研制”
- 参与科技部国际合作项目“物联网信息服务关键技术与应用示范研究”、“物联网环境监测与保护应用联合研发”
- 参与铁道部重点课题“铁路信号系统安全风险智能感知及防护系统”
- 参与上海市科委重大项目“城市智能交通协同监管与实时服务平台研发”
- 参与同济大学“TiEV途灵无人驾驶系统软件”

课程说明

- 课程目标
 - 熟练掌握软件工程的基本理论、方法和工具，能够在软件项目中自觉运用软件工程思想和方法，提升团队软件开发的质量
- 课程网站
 - <http://canvas.tongji.edu.cn/courses/>
- 教学资源参考
 - 清华大学国家级精品课程 《软件工程》 主讲人 刘强 副教授 刘璘 副教授
 - https://www.icourses.cn/sCourse/course_3016.html
 - https://www.xuetangx.com/course/THU08091000367/5883555?channel=learn_title

课程大纲

课程项目

- 尽可能的模拟软件项目的整个实施流程
- 以可商用的软件产品作为导向
- 敏捷开发实践
- 中国高校计算机大赛www.C4best.cn
- 详见《课程分组项目说明》

[illegible]

评分说明

- 平时作业 20%
- 课程项目 50%*
- 期末考试 30%*

推荐教材

- Ian Sommerville, Software Engineering (12th edition), Pearson
 - <https://iansommerville.com/software-engineering-book/static/>
- Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering: Theory and Practice (4th edition), Pearson
- 软件工程导论 第6版 张海藩

相关技术

- UML统一建模语言
 - https://www.w3cschool.cn/uml_tutorial/uml_tutorial-c1gf28pd.html
- Git分布式版本控制系统
 - <https://www.codecademy.com/learn/learn-git/modules/learn-git-git-workflow-u/cheatsheet>
- DevOps
 - <https://huaweicloud.com/devcloud/>
 - <https://gitlab.com>
- Apifox
 - <https://apifox.cn>
- LLM
- Google test framework*
 - <https://github.com/nordlow/gtest-tutorial>
- Google programming style*
 - <https://google.github.io/styleguide/>

- Questions?

软件工程概述

1	软件工程的产生与发展
2	软件工程基本概念
3	软件质量属性
4	软件过程模型

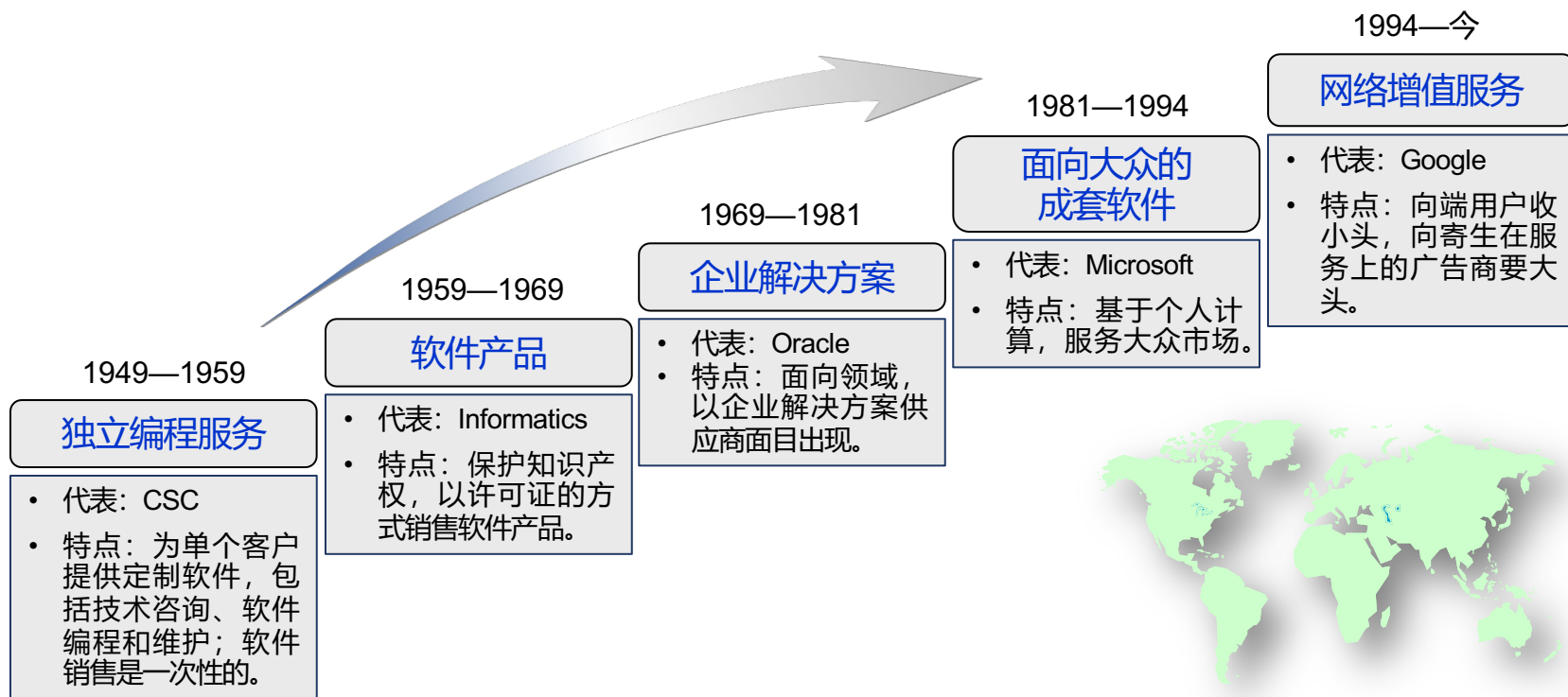
软件工程概述

1

软件工程的产生与发展

- 软件的发展历程
- 软件开发面临的挑战
- 软件的本质特性
- 软件开发问题的解决途径

软件发展历程



软件开发面临的挑战

- 交付的许多功能不是客户需要的
- 交付的日期没有保障
- 客户使用时发现许多Bug

客户
不满意

- 开发团队专注技术，忽视风险
- 无能力预测成本，导致预算超支

风险与
成本问题

- 客户需求变化频繁，无力应对
- 无法预见软件的交付质量
- 对流程盲目遵从，忽视客户业务价值

项目过程
失控

无力管理
团队

- 无法评估开发人员能力及工作进度
- 困扰于如何提升团队的能力与效率

举例1：ARIANE 5 火箭

1996年6月4日，Ariane 5火箭在发射37秒之后偏离其飞行路径并突然发生爆炸，当时火箭上载有价值5亿美元的通信卫星。



原因：

- 程序中试图将64位浮点数转换成16位整数时产生溢出
- 缺少对数据溢出的错误处理程序
- 备份软件通过复制而成

举例2: Windows Vista系统



- 该系统从2001年开始研发，整个过程历时5年半，先后有9000位开发人员投入其中，耗资60亿美元，代码规模超过5000万行。
- 按照微软公司最初的计划，该系统面世时间应该在2003年，之后推迟到2004年下半年再到2005年初，最终在取消一些高级功能后于2006年11月正式发布。
- 由于整个系统过于庞杂，其开发管理相当混乱，以致于很多时间用在互相沟通和重新决定上。
- 从Vista Beta 1进入公开测试以来，程序错误总数已经超过2万个，这其中还不包括微软内部未公开的一些错误。



<https://www.zhihu.com/question/19861013>

一个值得思考的问题

为什么一个看似简单的东西，却很有可能变成一个落后进度、超出预算、存在大量缺陷的怪物？



Brooks, F. P., *The Mythical Man Month*, Addison-Wesley, 1975

《人月神话》源于Brooks在IBM公司担任OS/360系统项目经理时的实践经验

软件的本质特性

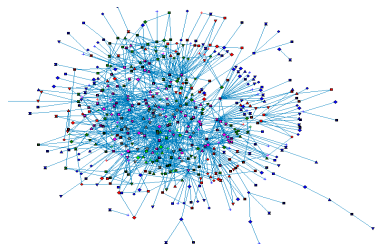


软件具有**复杂度、一致性、可变性和不可见性**等固有的内在特性，这是造成软件开发困难的根本原因。

Brooks, F. P., "No silver bullet: essence and accidents of software engineering", *IEEE Computer*, Vol. 20, No. 4, pp.10-19, 1987

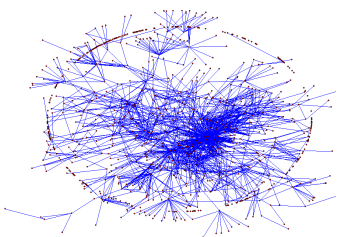
软件的本质特性： 复杂性

软件是人类思想的外延，人们将自己的思想传送给计算机，当产生的可执行文件被激活运行时，软件便重现人类的意图。



Linux内核有630个函数，存在1814个函数调用。

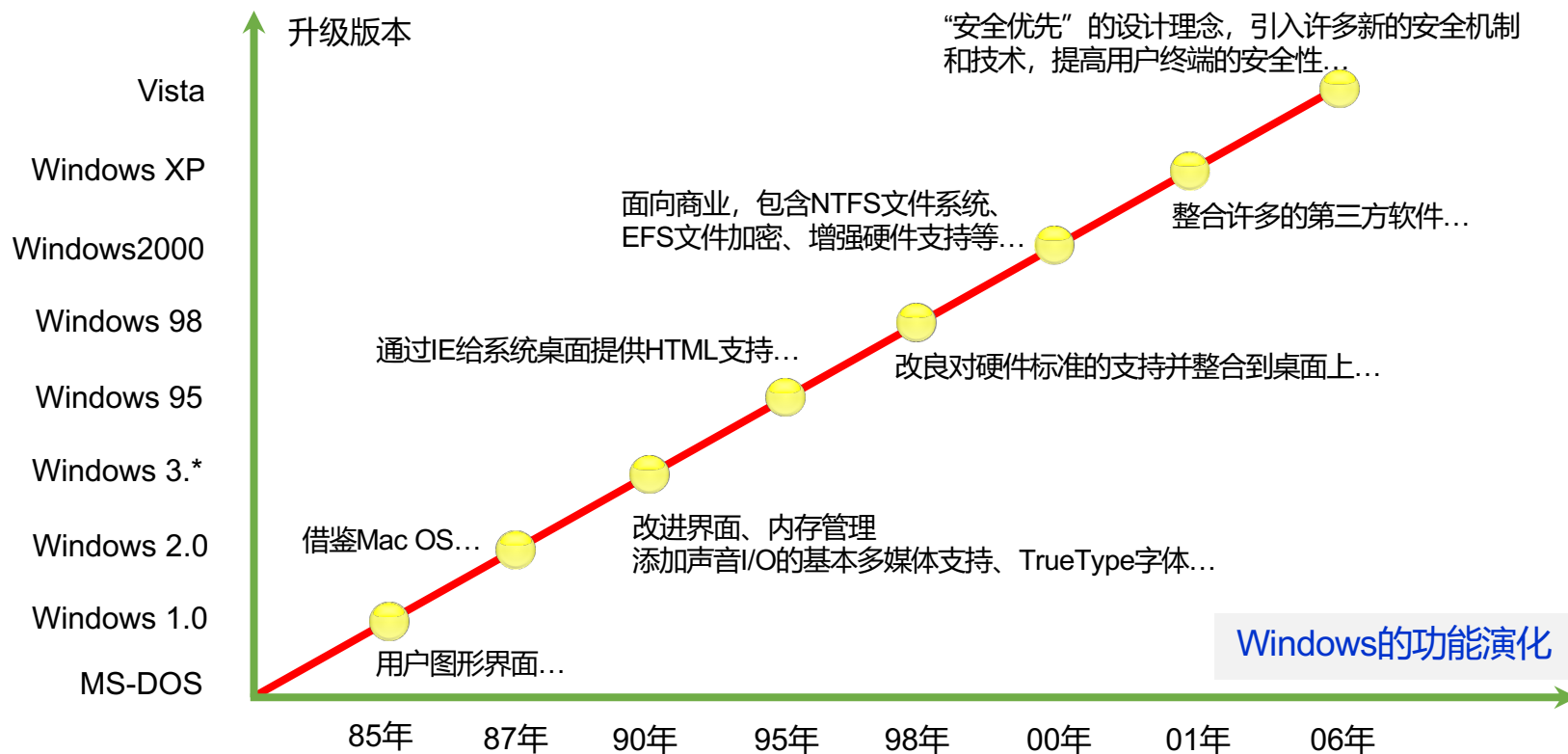
注：图中将函数表示为节点，调用关系表示为边。



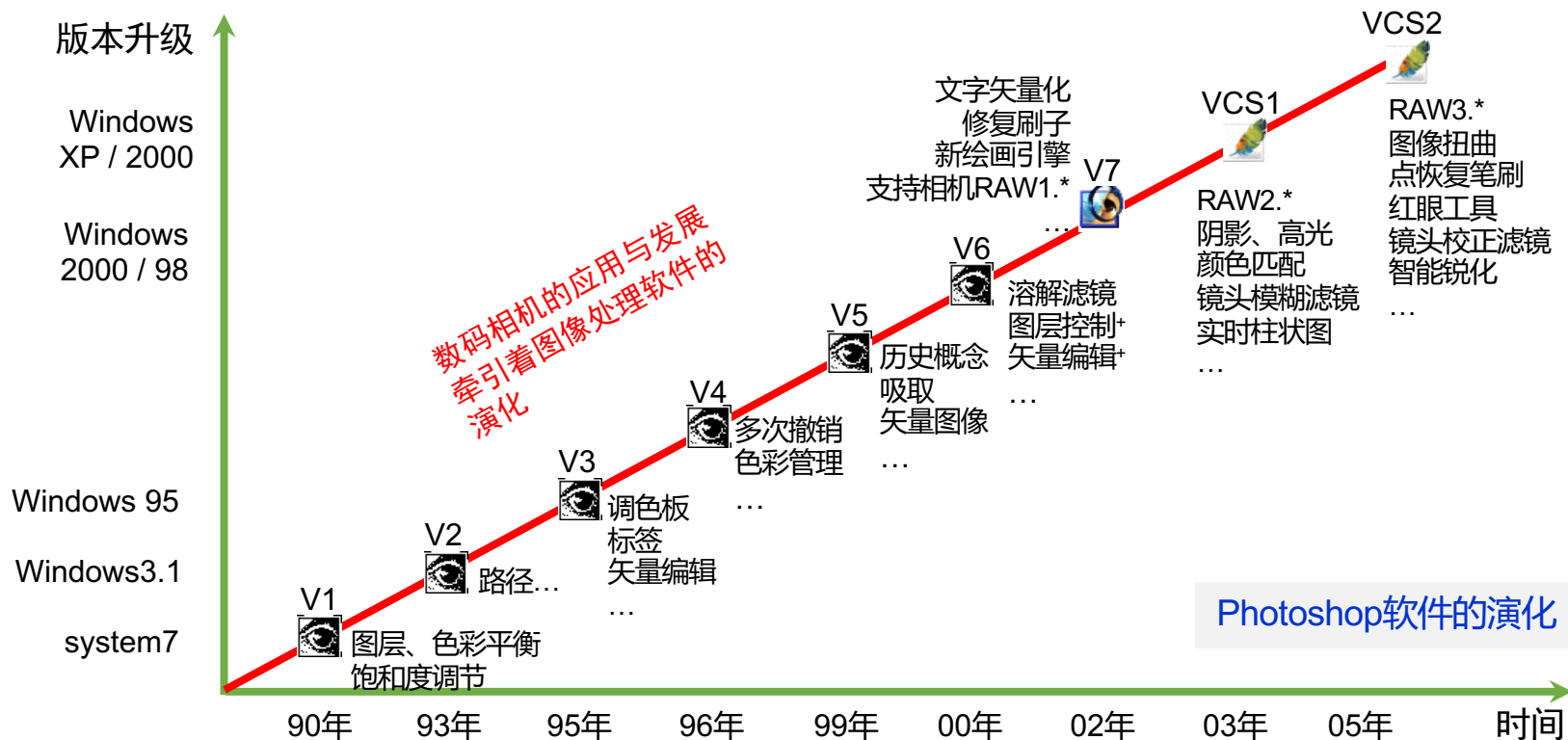
Tomcat 软件含1019个类，类与类之间有2109个继承或聚合关系。

注：图中将类表示为节点，关系表示为边。

软件的本质特性：演化性

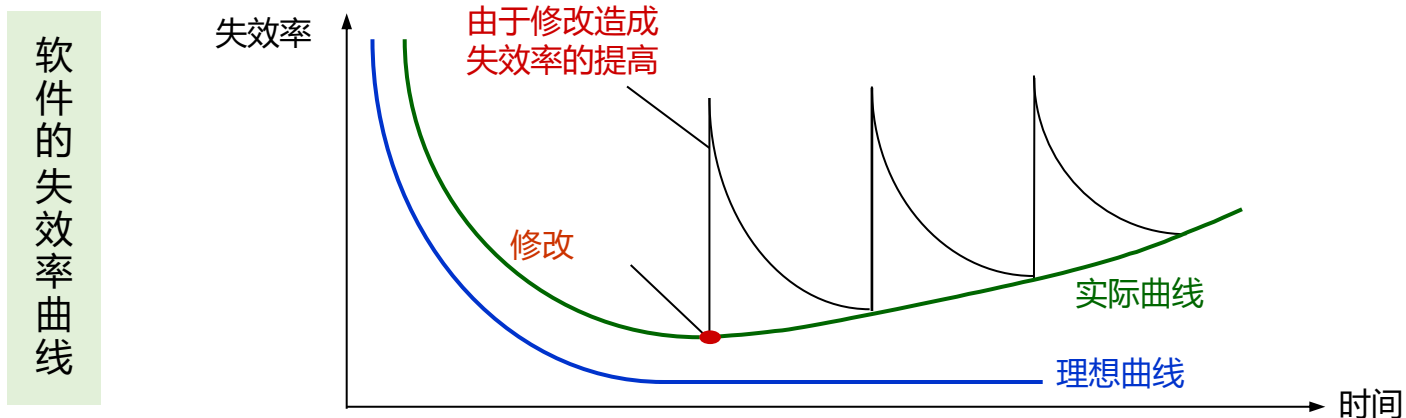


软件的本质特性：演化性

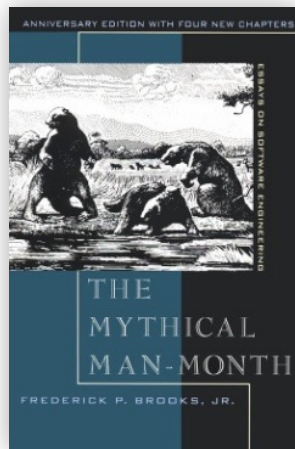


软件的本质特性：演化性

- 人们总是认为软件是容易修改的，但忽视了修改所带来的副作用
- 不断的修改最终导致软件的退化，从而结束其生命周期



软件的本质特性： 不可见性



软件人员太像“皇帝的新衣”故事中的裁缝了。当我来检查软件开发工作时，所得到的回答好象对我说：我们正忙于编织这件带有魔法的织物。只要等一会儿，你就会看到这件织物是及其美丽的。但是我什么也看不到，什么也摸不到，也说不出任何一个有关的数字，没有任何办法得到一些信息说明事情确实进行得非常顺利，而且我已经知道许多人最终已经编织了一大堆昂贵的废物而离去，还有不少人最终什么也没有做出来。

软件的本质特性

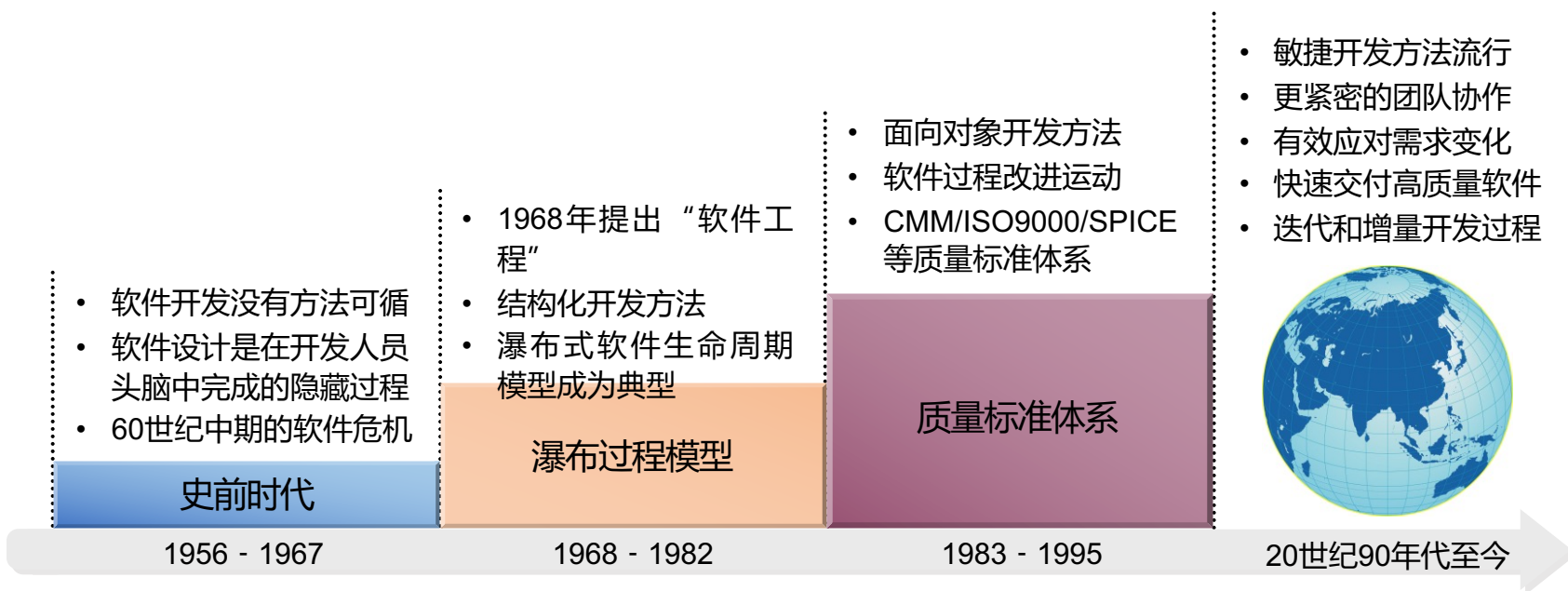


软件具有**复杂度、一致性、可变性和不可见性**等固有的内在特性，这是造成软件开发困难的根本原因。

Brooks, F. P., "No silver bullet: essence and accidents of software engineering", *IEEE Computer*, Vol. 20, No. 4, pp.10-19, 1987

软件开发问题的解决途径

软件工程诞生：1968年北大西洋公约组织（NATO）召开国际会议，提出“**软件工程**”概念和术语。



软件工程概述

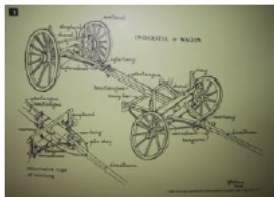
2

软件工程基本概念

- 系统的本质与工程的方法
- 什么是软件工程
- 软件工程的基本要素

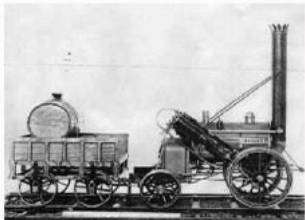
工程的方法

工程是将理论和所学的知识应用于实践的科学，以便经济有效地解决实际问题。



- craft
 - personal skill
 - experience
 - flexible materials

- engineering
 - tables and tools
 - recorded knowledge
 - controlled materials



■ 规模上的差异

- 花园小道 vs. 汽车高速公路
- 树上小屋 vs. 摩天大楼
- 加法程序 vs. 医院档案系统

■ 手工 (Craft) : 小规模的设计与建造

- 简单问题与单一目标
- 个人控制与个人技能

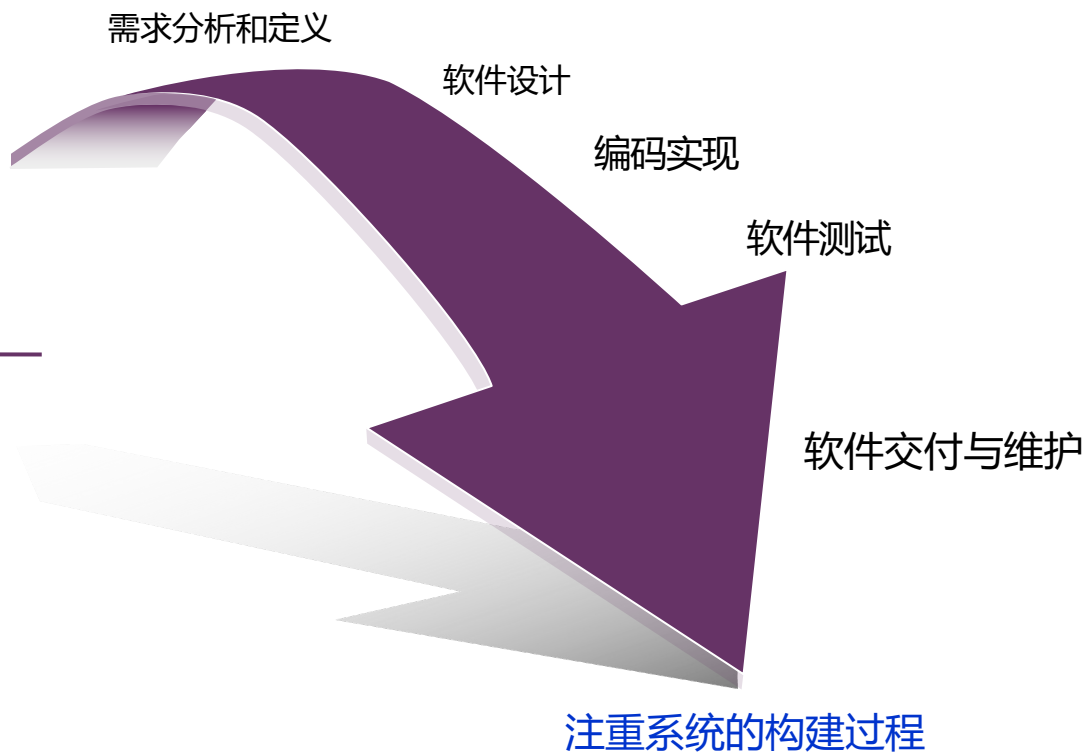
■ 工程 (Engineering) : 大规模的设计与建造

- 复杂问题与目标分解
- 团队协作，需要考虑运营、管理、成本、质量控制、安全等

工程的方法

工程的特征

- 平衡与决策
- 度量与验证
- 训练有素的过程
- 团队协作与角色分工
- 系统地运用工具
- 工程原则、标准和实践
- 重用设计和设计制品



什么是软件工程

软件工程是 ① 将系统性的、规范化的、可量化的方法应用于软件的开发、运行和维护，即工程化应用到软件上；② 对①中所述方法的研究。

软件工程的基本目标：

- 较低的开发成本
- 按时完成开发任务并及时交付
- 实现客户要求的功能
- 所开发软件具有良好的性能
- 较高的可靠性、可扩展性、可移植性
- 软件维护费用低



举例：麦当劳的过程管理



- 流水线作业：机械化、批量化、标准化
- 四无主义：无论何人，无论何时，无论何地，无品质差异
- 质量控制：在过程细节上精益求精

精益求精的细节管理：

- 面包直径均为17cm，这种尺寸入口最佳
- 面包中的气泡全部为0.5cm，这种尺寸味道最佳
- 牛肉食品的品质检查有40多项内容，从不懈怠
- 肉饼成分讲究，由83%肩肉与17%五花肉混制而成
- 牛肉饼重量在45克时其边际效益达到最大
- 汉堡包从制作到出炉时间严格控制在5秒钟
- 一个汉堡包净重1.8盎司，其中洋葱重量为0.25盎司
- 汉堡包出炉后超过10分钟，薯条炸好后超过7分钟，不准售出
- 汉堡包饼面上如有人工手压轻微痕迹，不准售出
- 与汉堡包一起卖的可口可乐必须是4 °C，这个温度最可口
- 柜台高度92cm，绝大多数顾客付账取物最方便
- 避免顾客等待30秒以上，这是人与人对话时产生焦虑的临界点

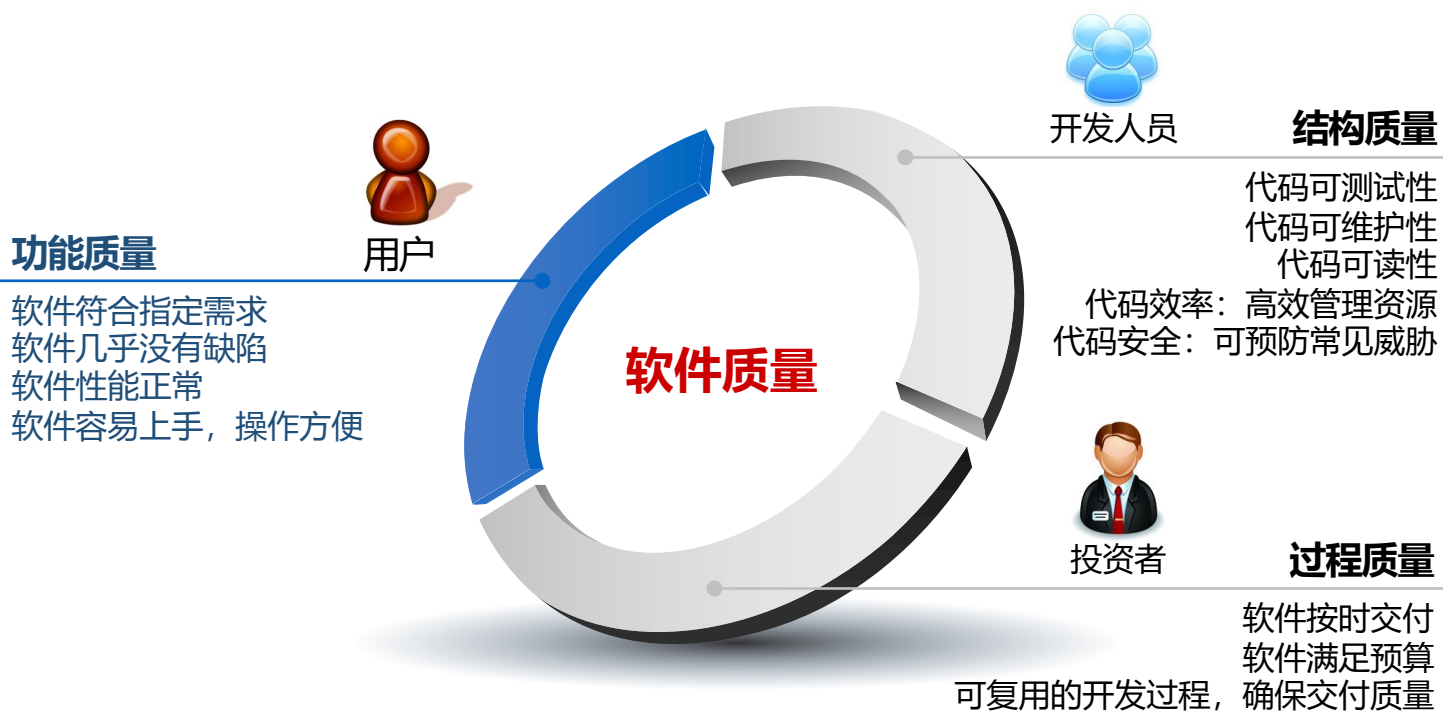
软件工程概述

3

软件质量属性

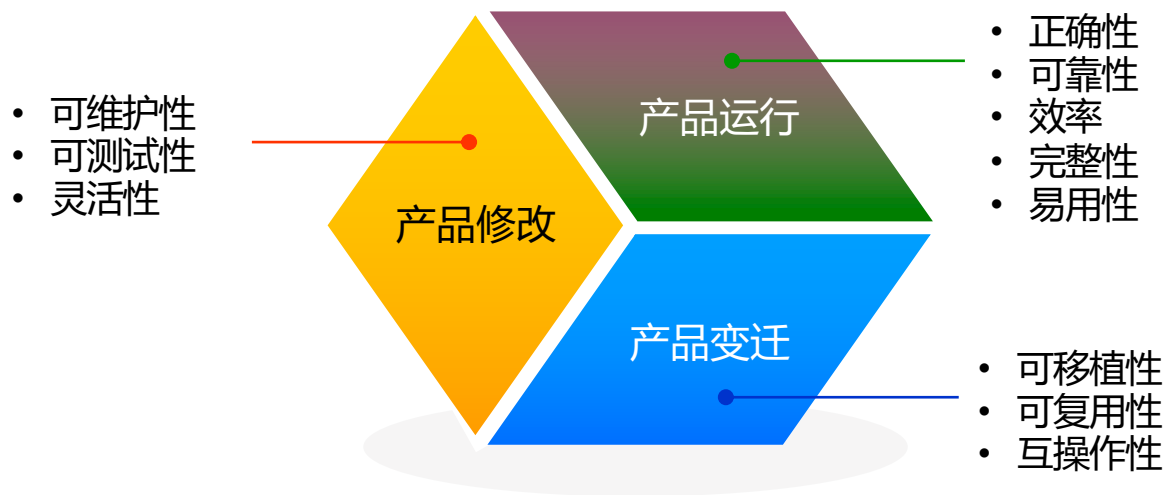
- 什么是好的软件
- McCall 质量模型
- ISO9126 质量模型
- 商业环境下的软件质量

什么是好的软件



McCall 质量模型

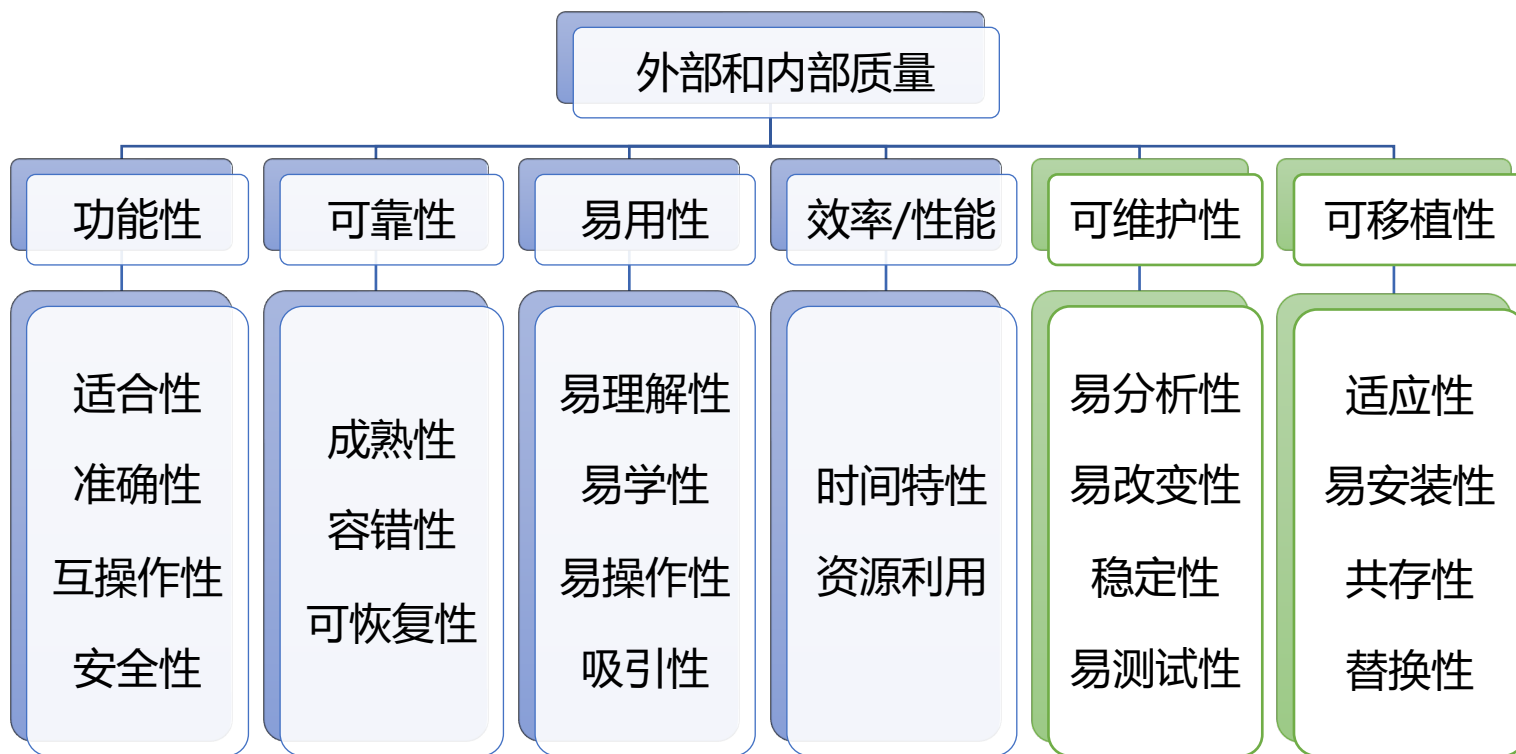
软件质量是许多质量属性的综合体现，各种质量属性反映了软件质量的不同方面。人们通过改善软件的各种质量属性，从而提高软件的整体质量。



ISO9126 质量模型

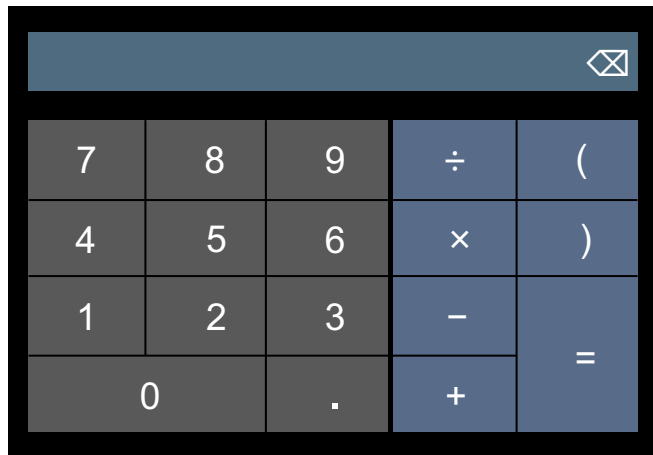


ISO9126 质量模型



案例分析

某学生使用Java/C++语言开发了一个计算器程序，可以实现加、减、乘、除四种操作。请阅读和分析该程序，你如何评价该程序的质量？

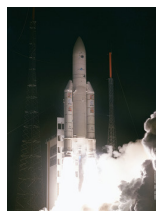


```
public class Calculator {  
    public static void main(String[] args) throws Exception {  
        String A;  
        String B;  
        String oper;  
        String result="";  
  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
  
        System.out.println("请输入数字A:");  
        A=br.readLine();  
  
        System.out.println("请输入运算符(+、-、×、/): ");  
        oper=br.readLine();  
  
        System.out.println("请输入数字B:");  
        B=br.readLine();  
  
        if(oper.equals("+")) {  
            result=String.valueOf(Double.parseDouble(A)+Double.parseDouble(B));  
        }  
  
        if(oper.equals("-")) {  
            result=String.valueOf(Double.parseDouble(A)-Double.parseDouble(B));  
        }  
  
        if(oper.equals("*")) {  
            result=String.valueOf(Double.parseDouble(A)*Double.parseDouble(B));  
        }  
  
        if(oper.equals("/")) {  
            result=String.valueOf(Double.parseDouble(A)/Double.parseDouble(B));  
        }  
  
        System.out.println("-----");  
        System.out.println("运算的结果是: "+A+oper+B+"="+result);  
        System.out.println("-----");  
    }  
}
```

商业环境下的软件质量

讨论思考：

软件质量的重要性是无容置疑的，那么是不是质量越高越好？
软件产品是否应该追求“零缺陷”？



举例：在航天器发射之前，只要发现任何异常，就会立即取消发射指令，直到异常被消除为止。前苏联甚至做得更过分，许多重大武器系统的负责人签了生死状，系统研制成功则获得英雄勋章，失败则被枪毙。



许多互联网软件（例如新浪微博、百度导航等）在产品仍然存在一定缺陷的情况下就发布上线，之后再不断更新版本修复已有的缺陷。这种系统为什么不像航天系统一样，在发布前应修改所发现的任何缺陷？

商业环境下的软件质量

讨论思考：

软件质量的重要性是无容置疑的，那么是不是质量越高越好？
软件产品是否应该追求“零缺陷”？

商业目标决定质量目标：

- 为了提高用户对产品的满意度，企业必须提高产品的质量，但是不可能为了追求完美的质量而不惜一切代价。当企业为提高质量所付出的代价超过收益时，这个产品就没有商业价值了。
- 企业必须权衡质量、效率和成本三个因素，产品质量太低了或者太高了，都不好。理想的质量目标不是“零缺陷”，而是恰好让广大用户满意，并且将提高质量所付出的代价控制在规定的预算之内。

软件工程概述

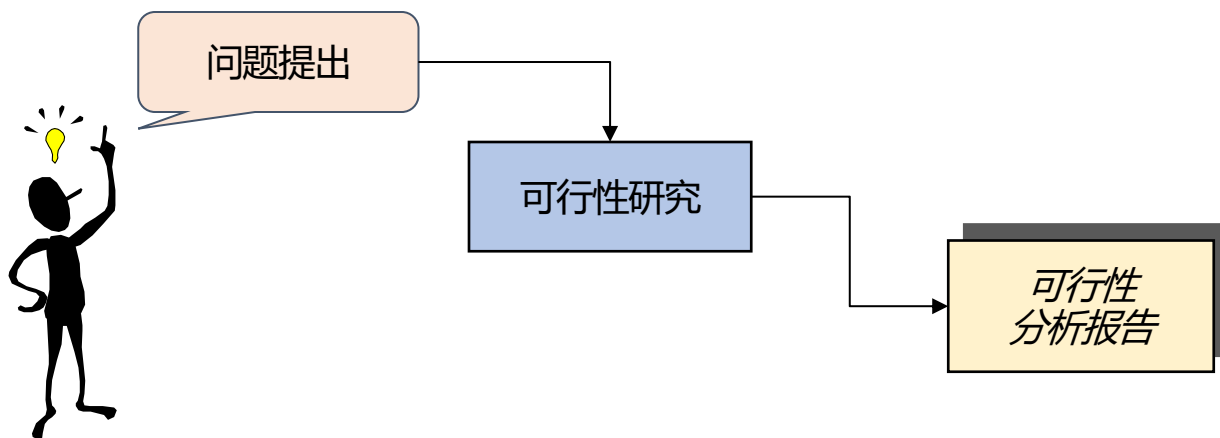
4

软件过程模型

- 软件开发活动
- 瀑布模型
- 原型化模型
- 阶段化开发
- 可转换模型

软件开发活动

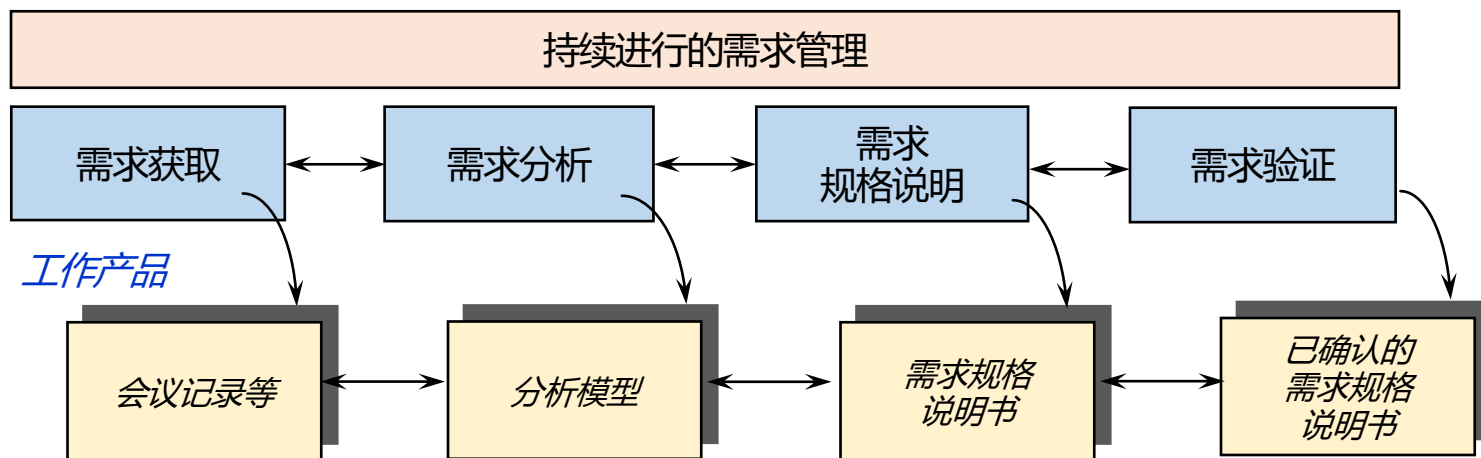
问题定义：人们通过开展技术探索和市场调查等活动，研究系统的可行性和可能的解决方案，确定待开发系统的总体目标和范围。



软件开发活动

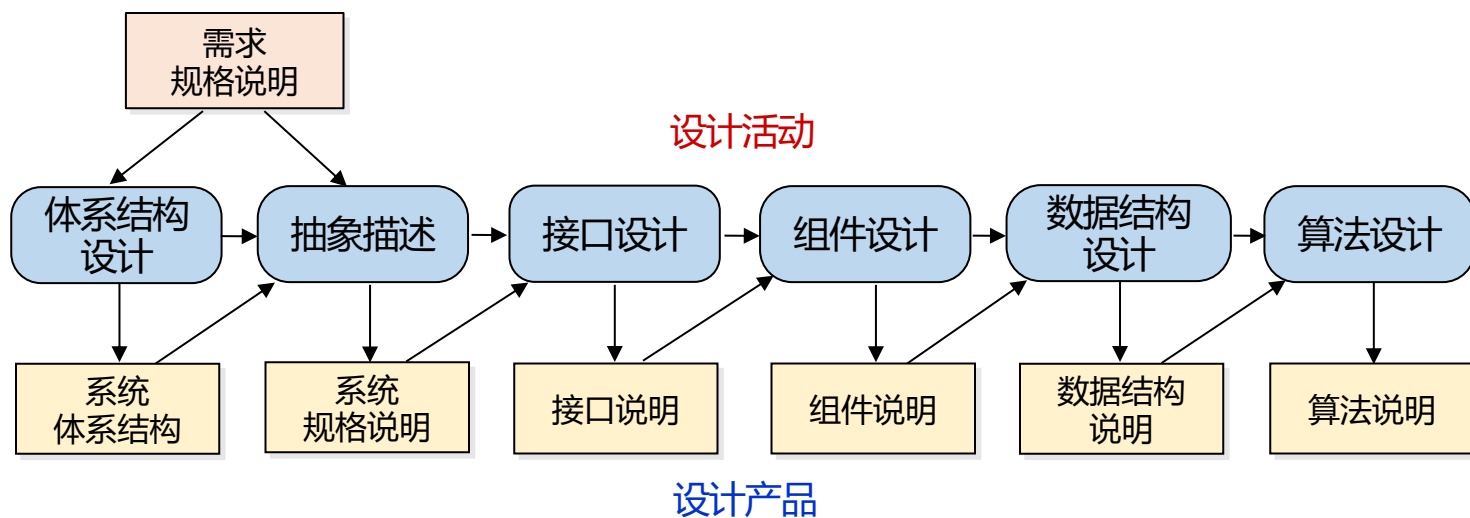
需求开发：在可行性研究之后，分析、整理和提炼所收集到的客户需求，建立完整的需求分析模型，编写软件需求规格说明。

活动



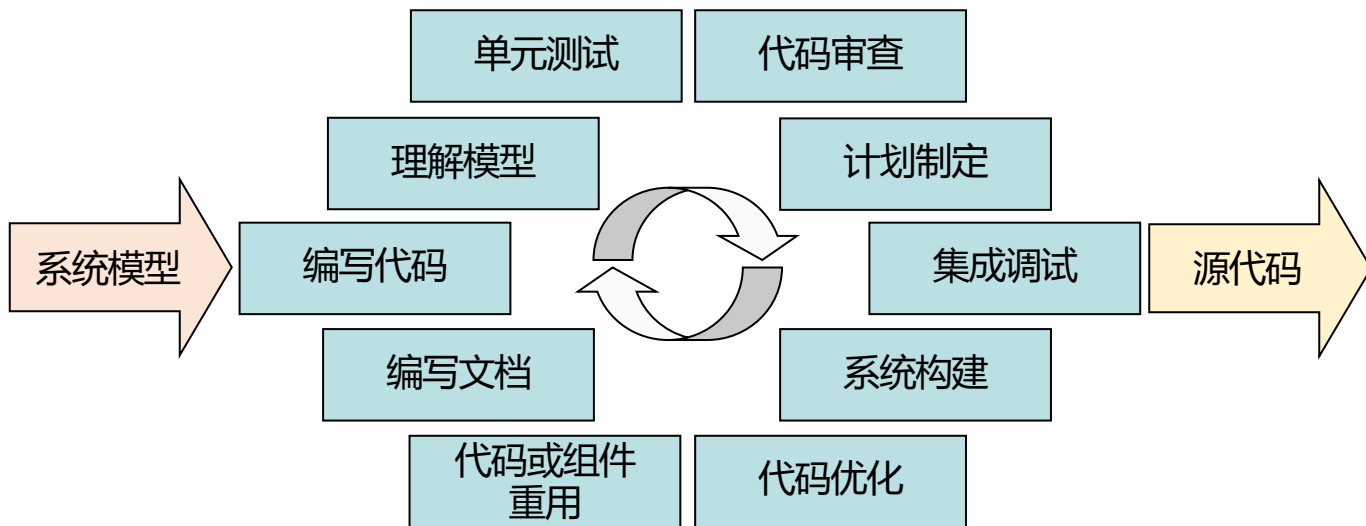
软件开发活动

软件设计：根据需求规格说明，确定软件体系结构，进一步设计每个系统部件的实现算法、数据结构及其接口等。



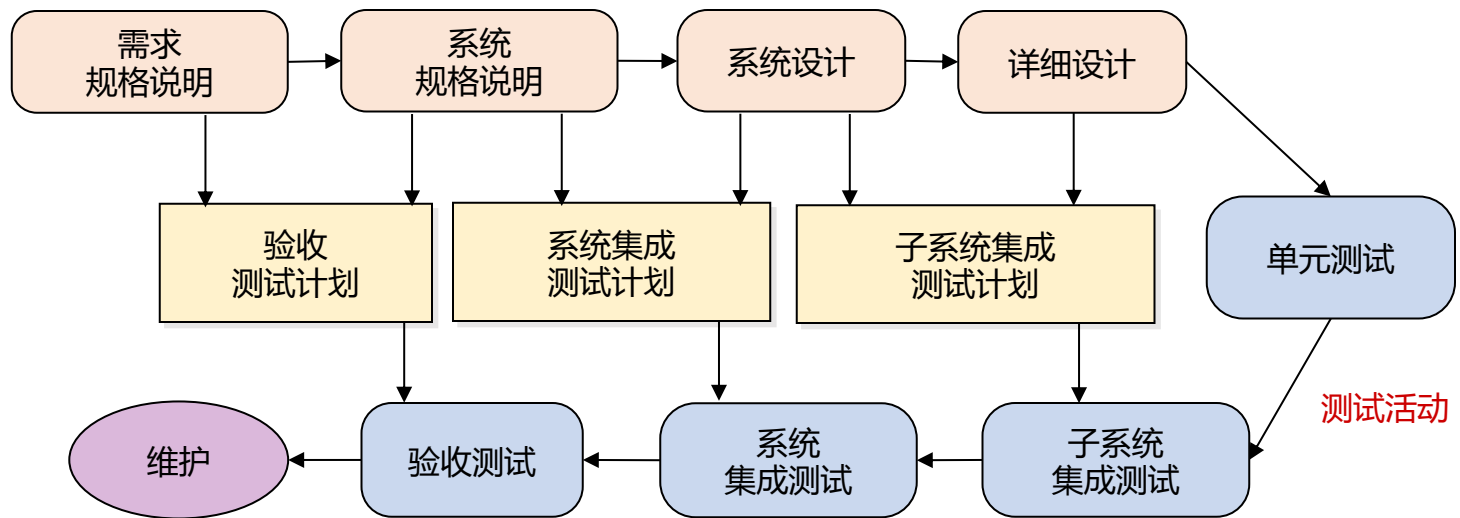
软件开发活动

软件实现：概括地说是将软件设计转换成程序代码，这是一个复杂而迭代的过程，要求根据设计模型进行程序设计以及正确而高效地编写和测试代码。



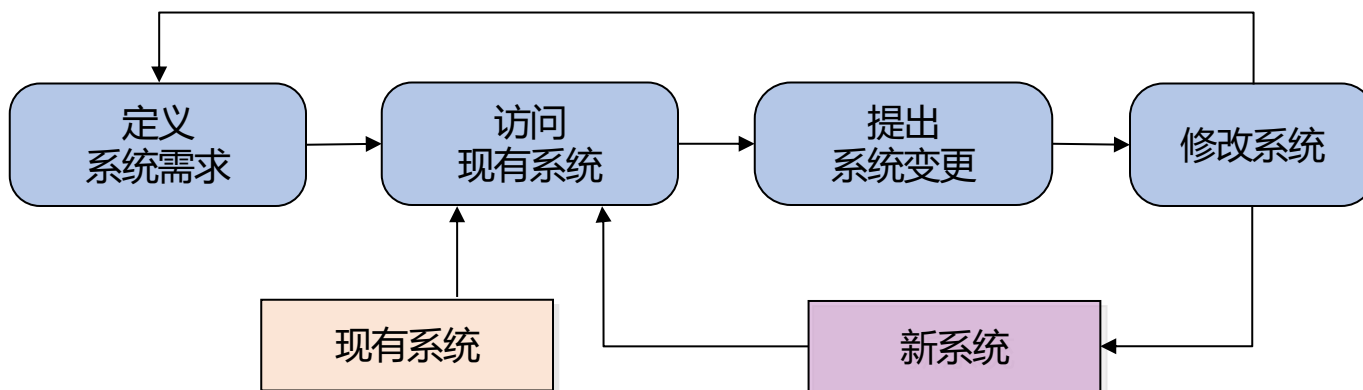
软件开发活动

软件测试：检查和验证所开发的系统是否符合客户期望，包括单元测试、子系统测试、集成测试和验收测试等。



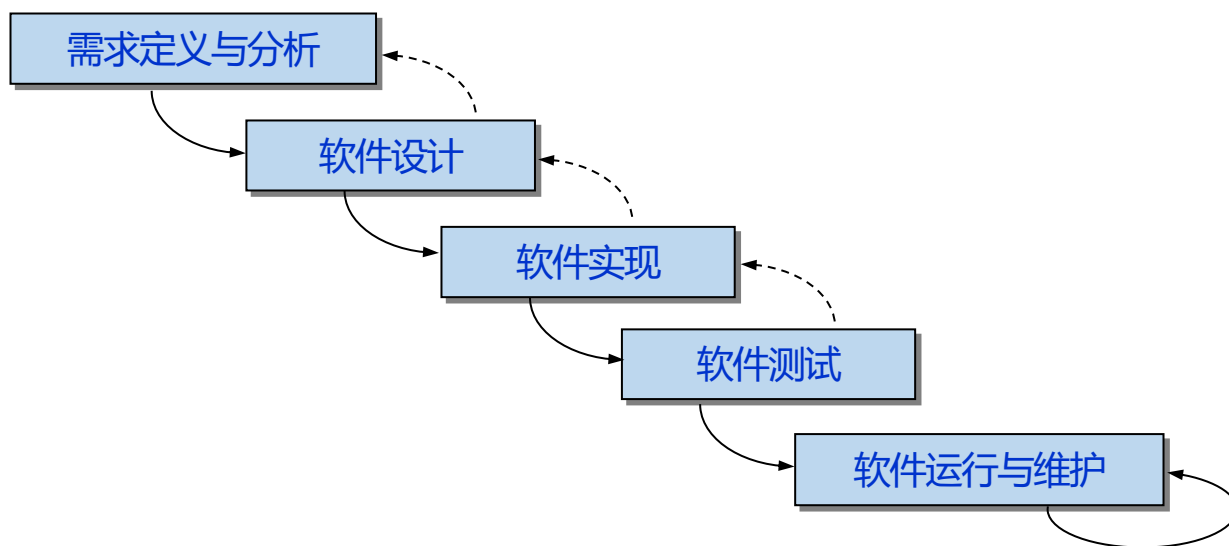
软件开发活动

软件演化：系统投入使用后对其进行改进，以适应不断变化的需求。完全从头开发的系统很少，将软件系统的开发和维护看成是一个连续过程更有意义。



瀑布模型

瀑布模型的开发阶段严格按照线性方式进行，每一个阶段具有相关的里程碑和交付产品，且需要确认和验证。



瀑布模型

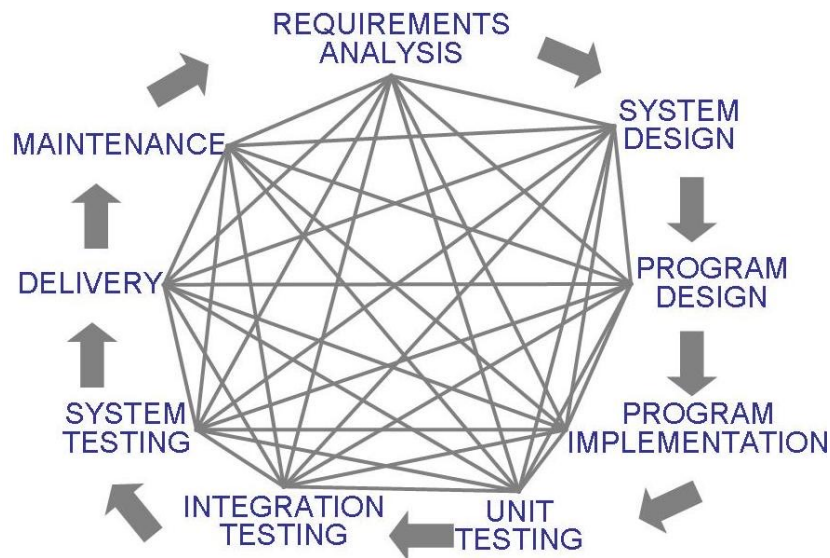
讨论：瀑布模型是否反映了实际的软件开发过程？软件开发作为一个问题求解过程，应当具备什么特点？



瀑布模型产生于硬件领域，它是从制造业的角度看待软件开发的。制造业是重复生产某一特定的产品，而软件开发并不是这样的，随着人们对问题的逐步理解以及对可选方案的评估，软件在不断演化。因此，软件开发是一个创造的过程，而不是一个制造的过程。

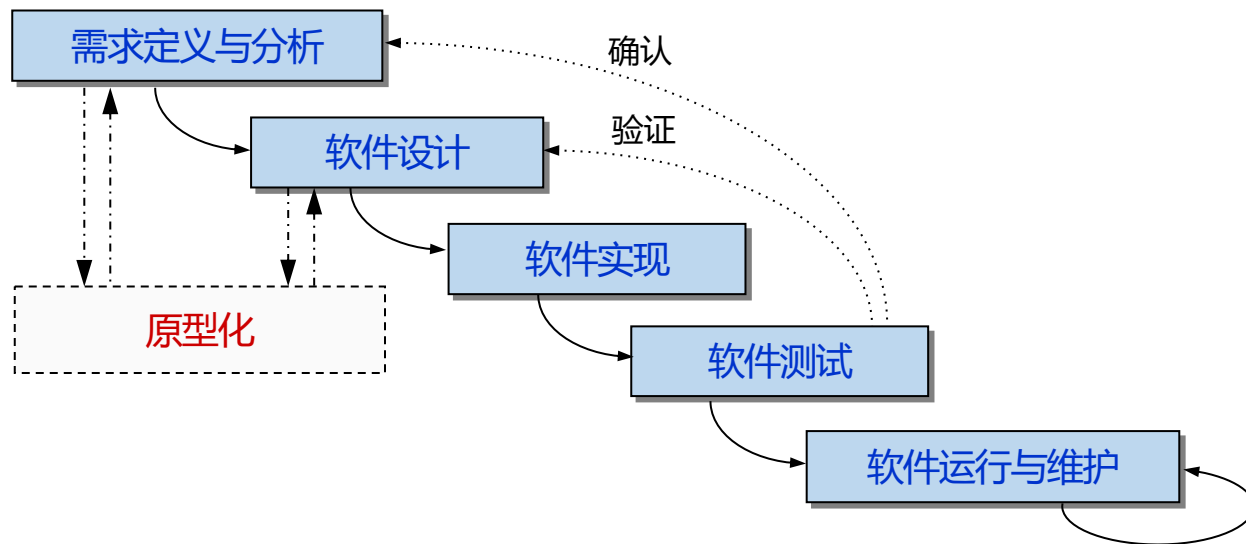
软件开发的迭代性

软件开发具有迭代性，需要不断地反复尝试，通过比较和选择不同的设计，最终确定令人满意的问题解决方案。



原型化模型

原型化模型需要迅速建造一个可运行的软件原型，它使用户和开发人员对系统的相关方面进行检查，以决定是否合适和恰当。



需求原型化

纸上原型是一种原型设计方法，通常在多张纸张和卡片上手绘或打印并裁剪成模块，用以表示不同的界面元素，再组合拼凑并粘贴到背景板上，构造成模拟真实产品界面的原型。这种方法显然构建更快、修改更方便。



https://www.bilibili.com/video/BV1Av4y1Q7iZ/?spm_id_from=333.337.search-card.all.click&vd_source=661261e4a0302a3aee1753693ae9ddc9

需求原型化

Balsamiq Mockups是一款免费的手绘风格的产品原型设计软件，比纸质的产品原型设计图更加方便存储，而且是简约清爽的手绘风格。它的兼容性也非常好，可使用XML语言保存元素，也可以导出PNG图片。

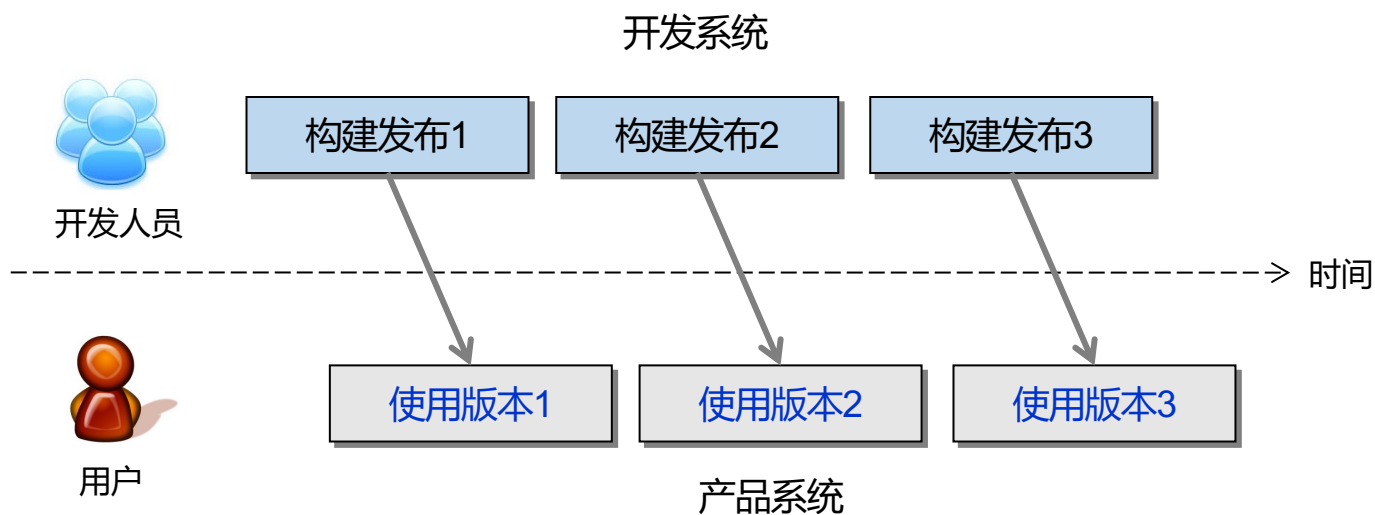


<http://www.balsamiq.com>

https://www.bilibili.com/video/BV1pV4y1K7A6/?spm_id_from=333.337.search-card.all.click&vd_source=661261e4a0302a3aee1753693ae9ddc9

阶段化开发

今天的商业环境需要快速地推出新产品，阶段化开发使得软件系统能够一部分一部分地交付，从而缩短软件开发周期。

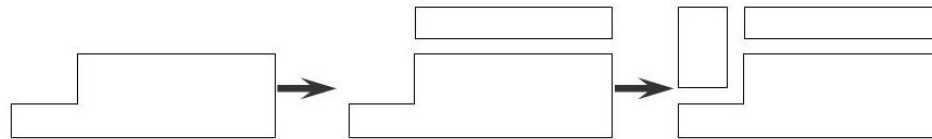


阶段化开发

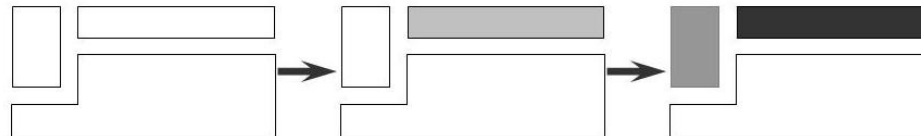
增量模型：在每一个新的发布中逐步增加功能直到构造全部功能。

迭代模型：一开始提交一个完整系统，在后续发布中补充完善各子系统功能。

INCREMENTAL DEVELOPMENT

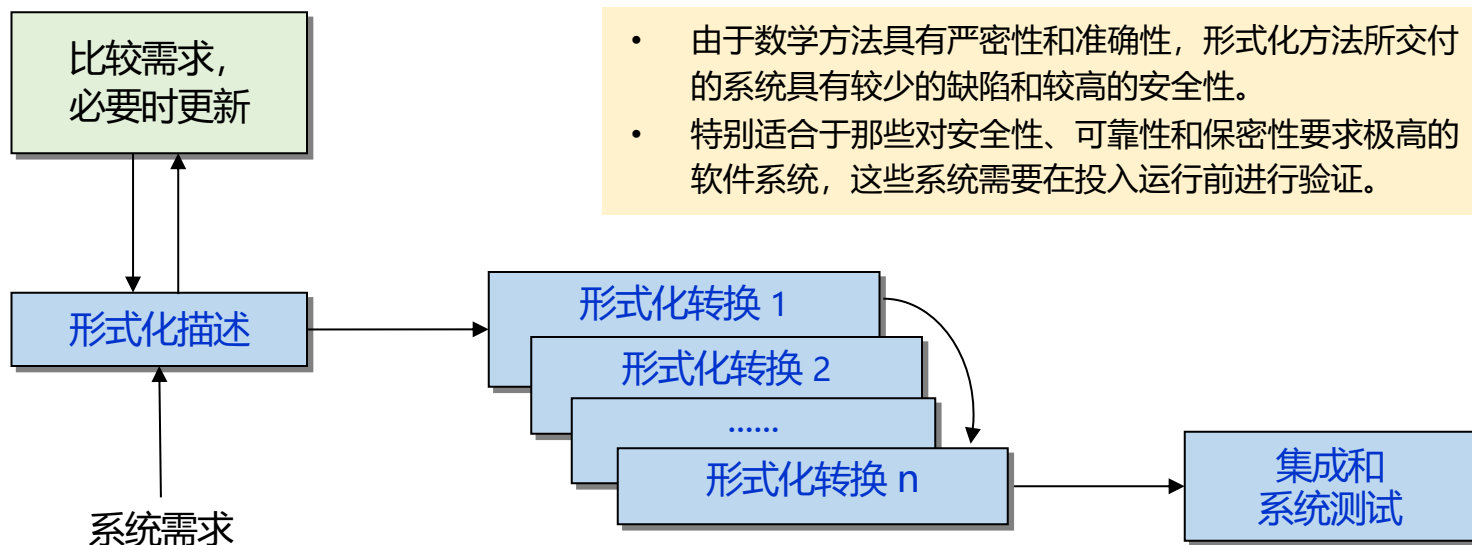


ITERATIVE DEVELOPMENT



可转换模型

可转换模型是采用形式化的数学方法描述系统，并利用一系列转换将形式化的需求规格说明变为可交付使用的系统。



案例分析

实例一：汽车制动防抱系统

- 汽车制动防抱系统（Anti-locked Braking System，简称 ABS）是一种具有防滑、防锁死等优点的汽车安全控制系统。
- 该系统通过汽车微电脑控制，以非常快的速度精密地控制制动液压力的收放，从而达到防止车轮抱死，确保轮胎的最大制动力以及制动过程中的转向能力，使车辆在紧急制动时具有躲避障碍的能力。



案例分析

实例一：汽车制动防抱系统

- 汽车制动防抱系统 (Anti-locked Braking System, 简称 ABS) 是一种具有防滑、防锁死等优点的汽车安全控制系统。
- 该系统通过汽车微电脑控制, 以非常快的速度精密地控制制动液压力的收放, 从而达到防止车轮抱死, 确保轮胎的最大制动力以及制动过程中的转向能力, 使车辆在紧急制动时具有躲避障碍的能力。

实例分析：

- 嵌入式控制系统
- 该系统对安全性和可靠性要求极高, 需要在投入运行前进行验证
- 适合采用形式化方法



案例分析

实例二：网络公开课程网站

- 某公司准备开发一个大规模在线公开课程网站，支持学校将自己的课程录像、课件及参考资料等公布在网上，学生可以进行自主学习。
- 该系统将教育、娱乐和社交网络结合在一起，创造了一种新型的网络教育模式，对传统的高等教育模式带了很大的冲击。



案例分析

实例二：网络公开课程网站

- 某公司准备开发一个大规模在线公开课程网站，支持学校将自己的课程录像、课件及参考资料等公布在网上，学生可以进行自主学习。
- 该系统将教育、娱乐和社交网络结合在一起，创造了一种新型的网络教育模式，对传统的高等教育模式带了很大的冲击。

实例分析：

- 系统需求会经常发生变化，业务模式存在不确定性
- 系统应该易于维护和修改
- 适合采用增量模型或迭代模型





谢谢大家！

THANKS

