软件工程概述

清华大学软件学院 刘强

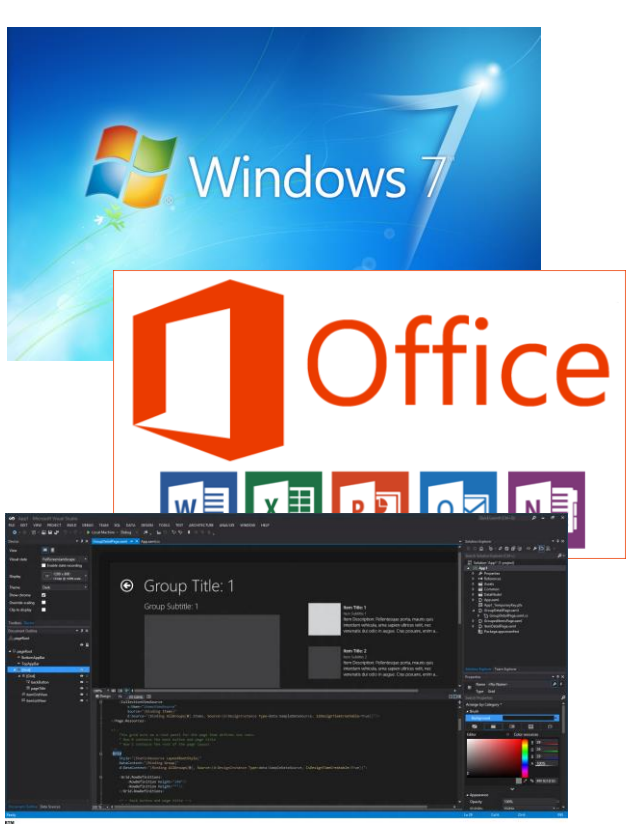
教学提纲

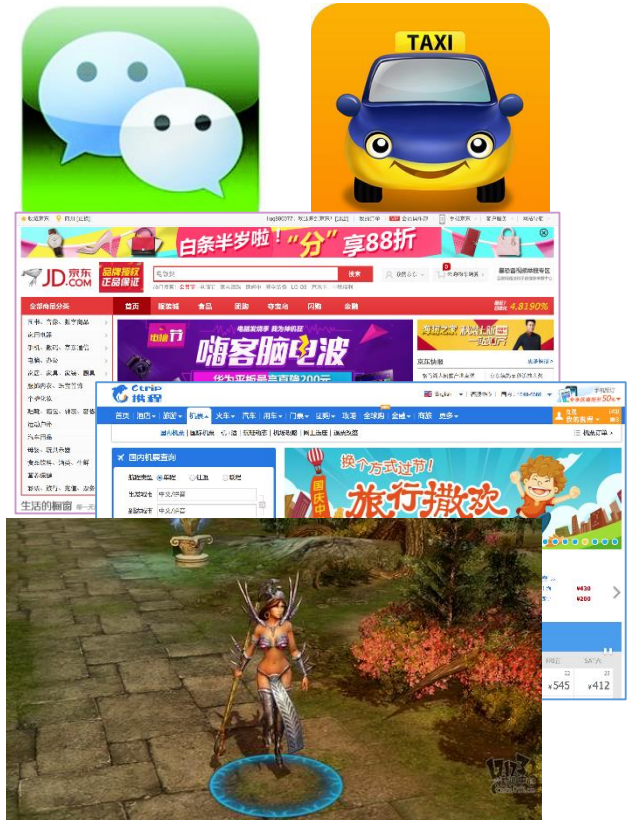
1. 软件工程的产生与发展
2. 软件工程的基本概念
3. 软件质量实现
4. 软件工程学科发展

软件无处不在

软件无处不在，为我们的生活创造了无限精彩。在当今的信息时代，世界正在变得更加“智慧”，万事万物间感知化、互联化和智能化的程度不断加深。

什么是软件


Windows 7
Office
Group Title: 1
Group Subtitle: 1


TAXI
白条半岁啦！"分"享88折
嗨客脑电波
旅行撒欢

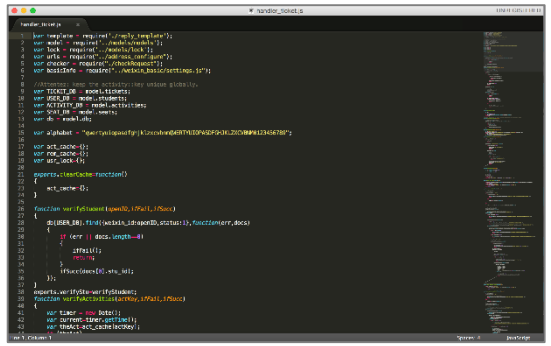
什么是软件

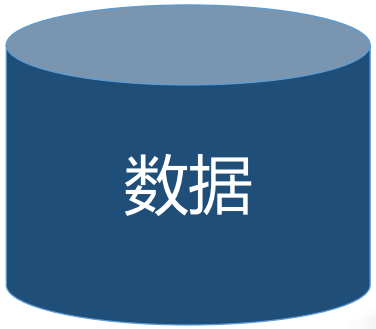
Software

软件的定义

软件 = 程序 + 数据 + 文档

- 程序：计算机可以接受的一系列指令，运行时可以提供所要求的功能和性能。
- 数据：使得程序能够适当地操作信息的数据结构。
- 文档：描述程序的研制过程、方法和使用的图文资料。





讨论：软件的定义

- 举例1：程序设计基础 —— 打飞机游戏
- 举例2：数据结构与算法 —— 二叉树遍历算法
- 举例3：Web前端技术实训课程 —— 微信小游戏
- 举例4：清华大学软件学院学生节 —— 弹幕微信墙

你认为上面的作品是软件吗？为什么？

如何理解软件的真正含义？

讨论：软件的定义

网购商品推荐

商品推荐系统是帮助用户在海量商品信息中发现对自己有价值的信息，并且让信息能够展现在对其感兴趣的用户面前。从软件的角度，说一说你对这个产品的看法。

软件的定义

软件 = 程序 + 数据 + 文档

- 程序：计算机可以接受的一系列指令，运行时可以提供所要求的功能和性能。
- 数据：使得程序能够适当地操作信息的数据结构。
- 文档：描述程序的研制过程、方法和使用的图文资料。

软件 ≠ 程序

- 程序是一个纯计算机的概念，它是为了解决特定的问题而用计算机语言编写的一系列代码。
- 软件更主要的是一个商业概念，它是一种商业产品，要有市场和能够盈利的商业模式，被一定数量的用户接受和使用。

软件的本质特性

一架客机由数百万个单独的部件组成，需要上千人组装，但通常都能够按时按预算交付使用。

微软于1989年11月发布的Word最初版本，花费了55人年，大约有249,000行源代码，却晚了4年交付使用。

软件的本质特性

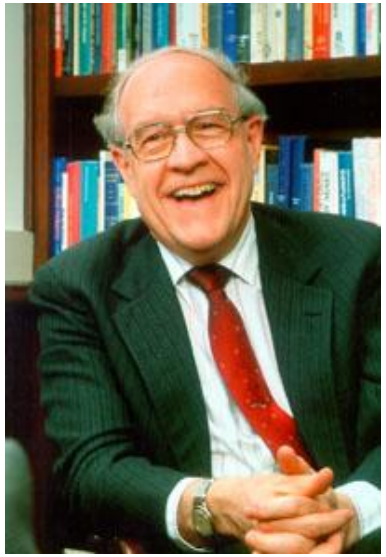
举例：王老师的儿子现在上小学二年级，老师要求家长每天出30道加减法算术题让学生练习。王老师决定写一个程序实现，很快就做好了。

学校老师听说之后称赞有加，其他老师也想要类似的程序，希望二年级到四年级都能用。

- 题目避免重复
- 可定制数量和打印方式
- 设置一系列参数：是否有乘除法；是否有括号；数值范围；加减有无负数；除法有无余数；是否支持分数；是否支持小数；打印每行的间隔等

王老师熬夜完成了该软件的初始版本，学校教导主任看到很满意，提出应该放到学校网站上，让老师和家长在网上定制各类的四则运算作业，还可以生成试卷，并且网站随时可以访问。

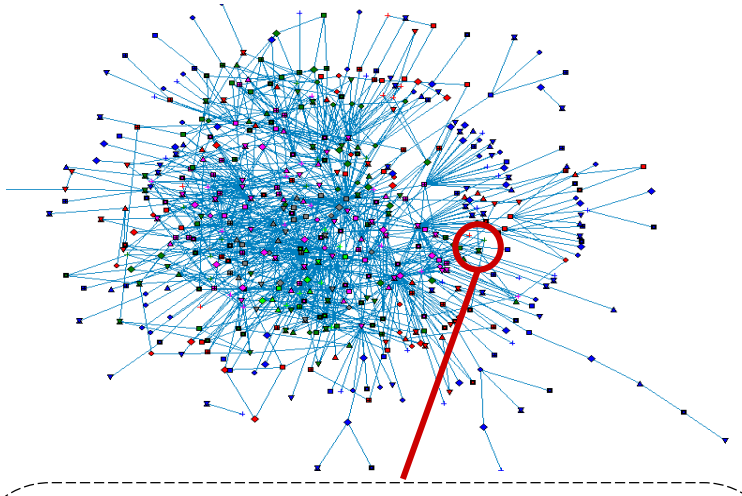
软件的本质特性



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。

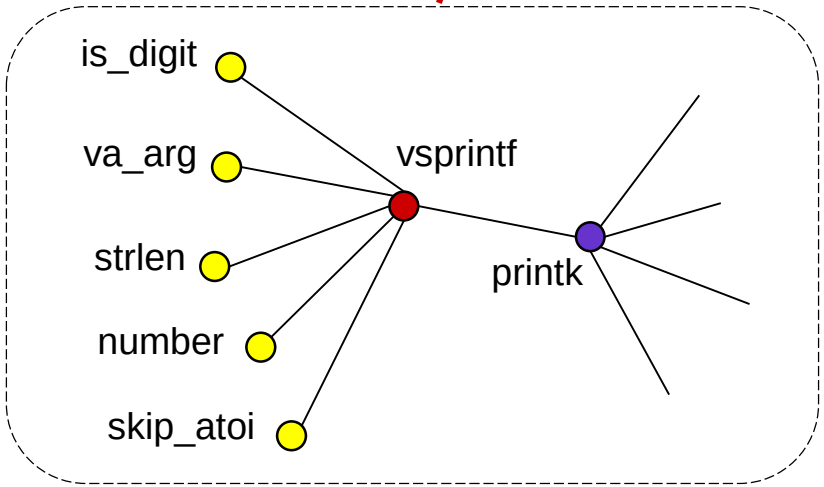
Brooks, F. P., “No silver bullet: essence and accidents of software engineering”, *IEEE Computer* , Vol. 20, No. 4, pp.10-19, 1987

软件的本质特性：复杂性



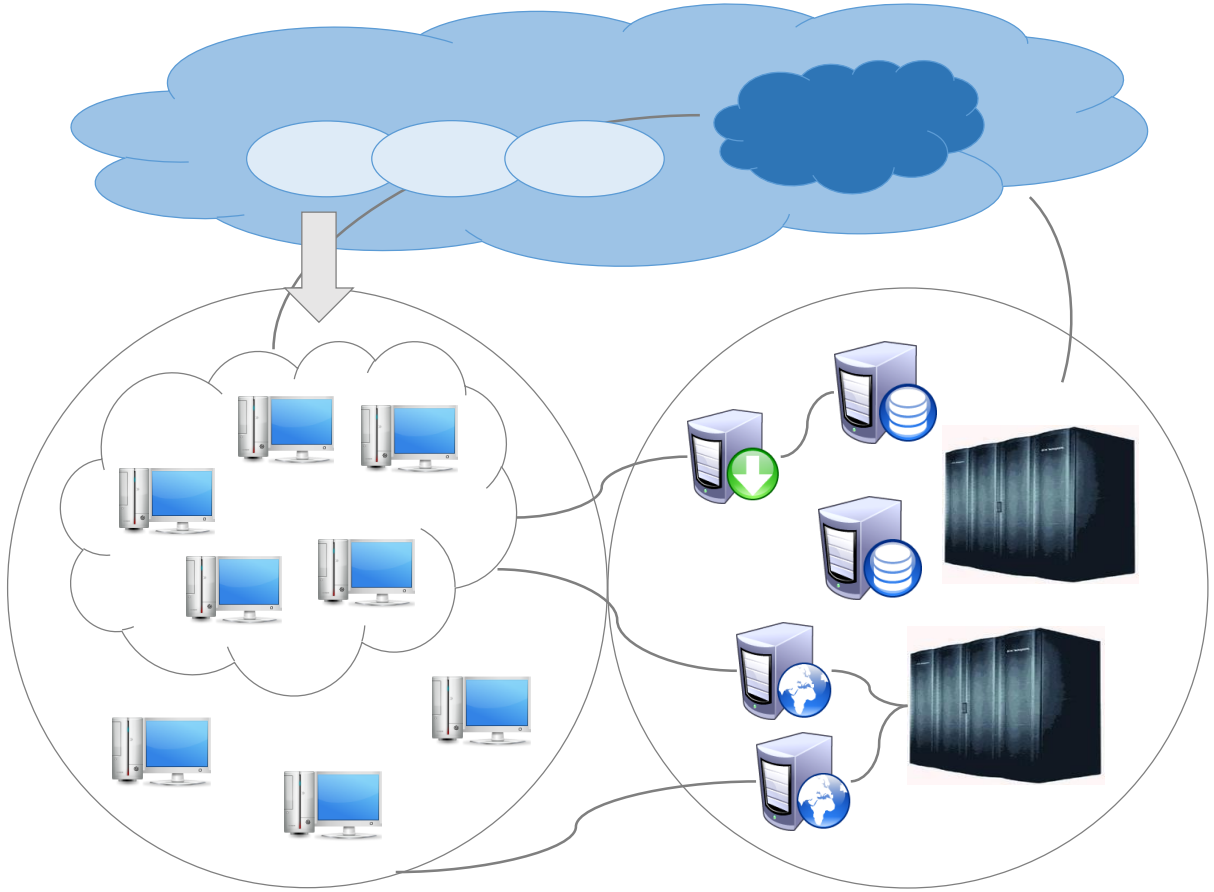
Linux内核有630个函数，存在1814个函数调用。

注：图中将函数表示为节点，调用关系表示为边。



函数 vsprintf 向控制台输出字符、数字和字符串，它调用了 is_digit、strlen、number 等多个函数，函数在动态执行过程中呈现出更为复杂的状态。

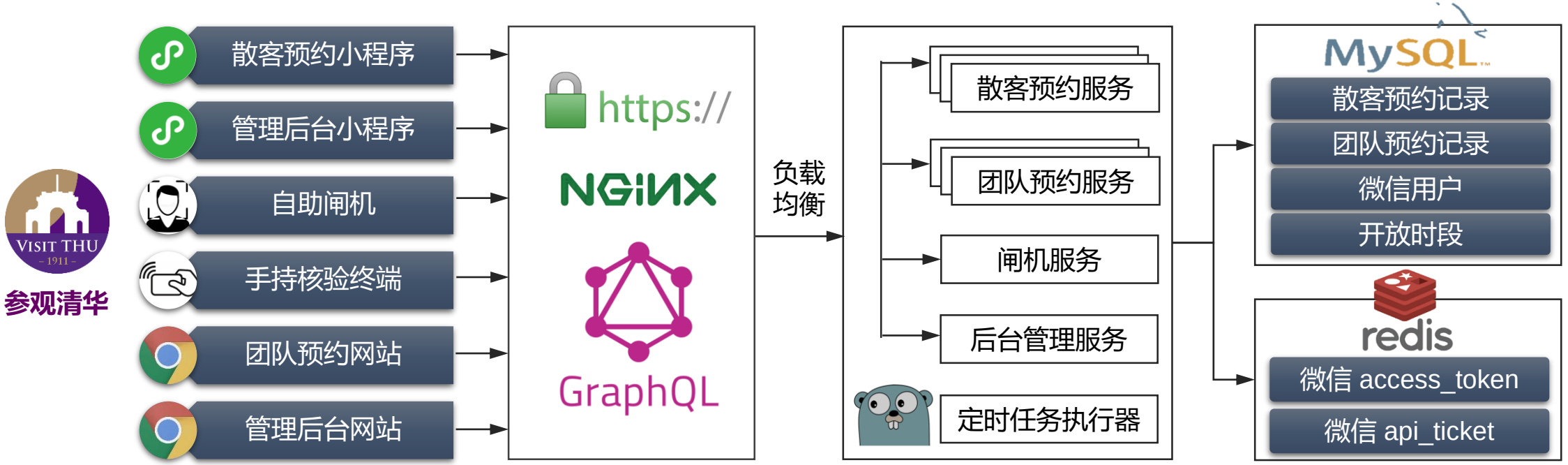
软件的本质特性：复杂性



- Google搜索引擎建立在遍布全球30多个站点、超过100万台服务器的云计算设施上。
- Amazon拥有28个云计算中心，在全球的服务器总量超过150万台。
- 阿里云是国内最大的云计算平台，拥有近百万台服务器，分布在北京、上海、深圳、香港和美国等。

软件的本质特性：一致性

- 软件不能独立存在，需要依附于一定的环境（如硬件、网络以及其他软件）
- 软件必须遵从人为的惯例并适应已有的技术和系统
- 软件需要随接口不同而改变，随时间推移而变化，而这些变化是不同人设计的结果



软件的本质特性：演化性

2011

- 2011.1.21，微信1.0发布（文字短信、QQ好友、头像）
- 2011.2，支持多人会话
- 2011.3，增加图片功能
- 2011.5.10，微信2.0发布（语音对讲、QQ邮箱提醒）

2010.11.20
微信正式立项

2012

- 2012.4.19，微信4.0发布（朋友圈、开放API、地理位置）
- 2012.7.19，微信4.2发布（视频、网页版、朋友圈回复）
- 2012.8.18，微信公众平台开放
- 2012.9.25，微信4.3发布（摇一摇传图、解绑、扫一扫）

- 2011.8，微信2.5发布（查看附近的人、语音记事本）
- 2011.10.1，微信3.0发布（摇一摇、漂流瓶）
- 2011.12，微信3.5发布（二维码扫描、自定义表情）

2013

- 2013.1.21，微信4.5发布（语音聊天室、摇一摇搜歌、语音提醒、位置导航）
- 2013.8.5，微信5.0发布（折叠公众账号、游戏中心、新版扫一扫、微信支付）

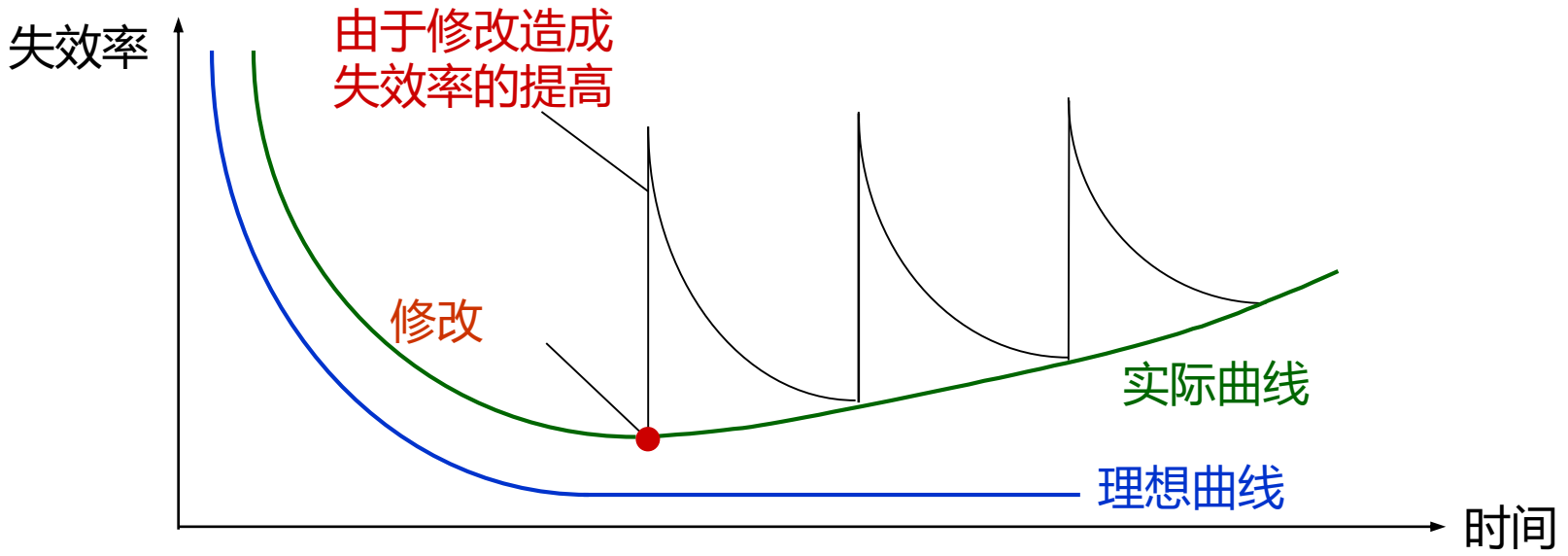
2014

- 2014.1.26，微信5.2发布（聊天记录搜索、银行卡新增服务、图片墙、@提醒）
- 2014.5.8，微信5.3发布（面对面建群、收藏内容添加标签、外文翻译）
- 2014.8.14，微信5.4发布

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

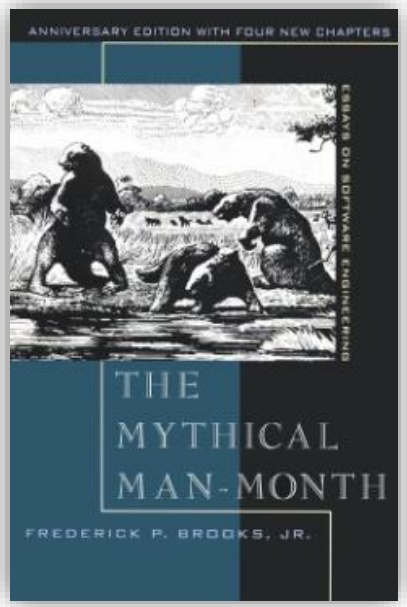
软件的失效率曲线



软件的本质特性：不可见性

- 软件是一种“看不见、摸不着”的逻辑实体，不具有空间的形体特征
- 开发人员可以直接看到程序代码，但是源代码并不是软件本身
- 软件是以机器代码的形式运行，但是开发人员无法看到源代码是如何执行的

软件的本质特性：不可见性



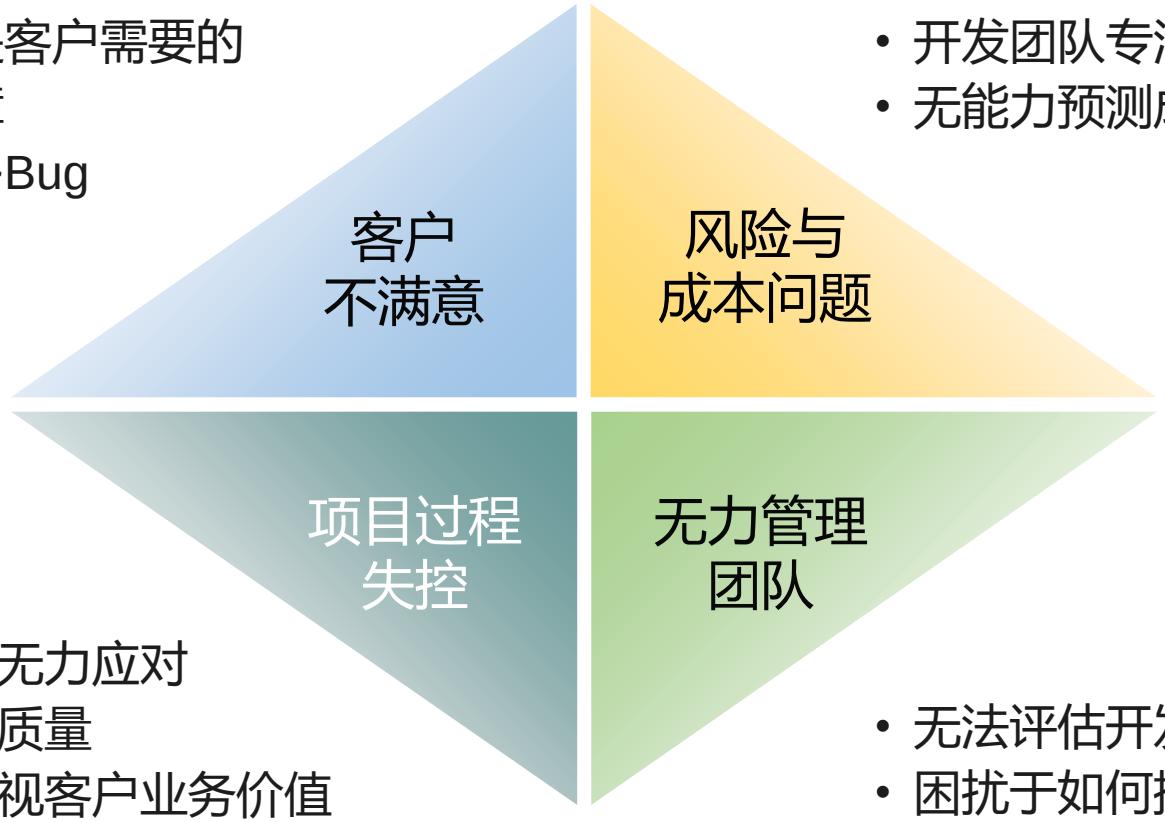
软件人员太像“皇帝的新衣”故事中的裁缝了。当我来检查软件开发工作时，所得到的回答好象对我说：我们正忙于编织这件带有魔法的织物。只要等一会儿，你就会看到这件织物是及其美丽的。但是我什么也看不到，什么也摸不到，也说不出任何一个有关的数字，没有任何办法得到一些信息说明事情确实进行得非常顺利，而且我已经知道许多人最终已经编织了一大堆昂贵的废物而离去，还有不少人最终什么也没有做出来。

软件开发面临的挑战

软件所具有的复杂性、一致性、可变性、不可见性等特性，使得软件开发过程变得难以控制，开发团队如同在焦油坑中挣扎的巨兽。

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

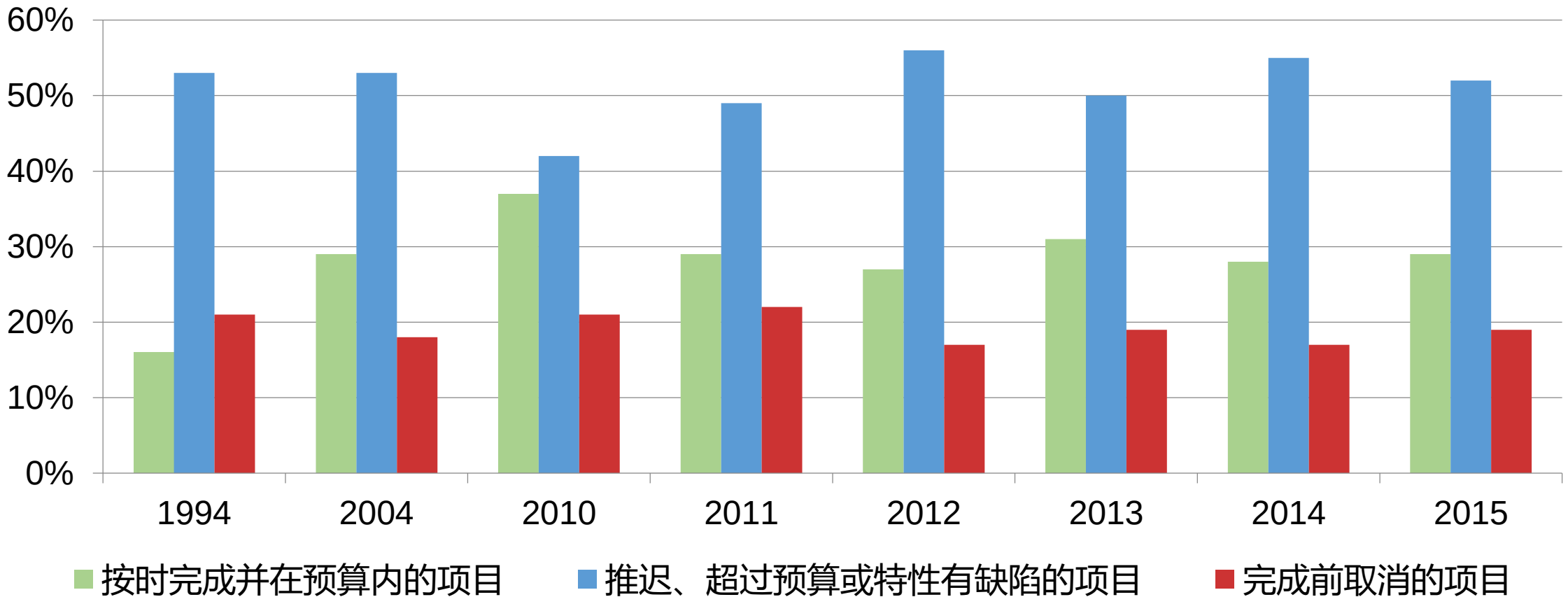


- 开发团队专注技术，忽视风险
- 无能力预测成本，导致预算超支

- 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

- 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

美国Standish集团的调查报告



美国Standish集团的调查报告

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	28%	4%
	Waterfall	44%	45%	11%

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：微软 Windows 系统

Vista

- 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。

举例2：微软 Windows 系统

网友选择：Win8超越Vista成最差系统！

举例3：12306 网络购票系统

- 12306网络购票系统历时两年研发成功，耗资3亿元人民币，于2011年6月12日投入运行。
- 2012年1月8日春运启动，9日网站点击量超过14亿次，出现网站崩溃、登录缓慢、无法支付、扣钱不出票等严重问题。
- 2012年9月20日，由于正处中秋和“十一”黄金周，网站日点击量达到14.9亿次，发售客票超过当年春运最高值，出现网络拥堵、重复排队等现象。

举例4：丰田汽车意外突然加速

2007年9月，美国奥克拉荷马高速公路出口闸道上，一辆05款的凯美瑞汽车突然自动加速，造成一死一伤的惨剧。美国国家公路交通安全管理局表示，从2000年至2010年5月中旬，接到超过6200起丰田轿车突然加速问题的投诉，其中涉及89起死亡事故及57起受伤事故。

事故原因：

- 油门踏板位置感测系统故障：汽车本身因为软件缺陷不会产生诊断故障代码，导致汽车无预警加速。
- 车载系统本身故障：有可能会在驾驶员控制油门的情况下，擅自连续进行燃油喷射和点火。

软件工程的诞生

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。

软件工程的发展历史

- 软件开发没有方法可循
- 软件设计是在开发人员头脑中完成的隐藏过程
- 60世纪中期的软件危机

史前时代

1956 - 1967

- 1968年提出“软件工程”
- 结构化开发方法
- 瀑布式软件生命周期模型成为典型

瀑布过程模型

1968 - 1982

- 面向对象开发方法
- 软件过程改进运动
- CMM/ISO9000/SPICE等质量标准体系

质量标准体系

1983 - 1995

- 敏捷开发方法流行
- 更紧密的团队协作
- 有效应对需求变化
- 快速交付高质量软件
- 迭代和增量开发过程

20世纪90年代至今

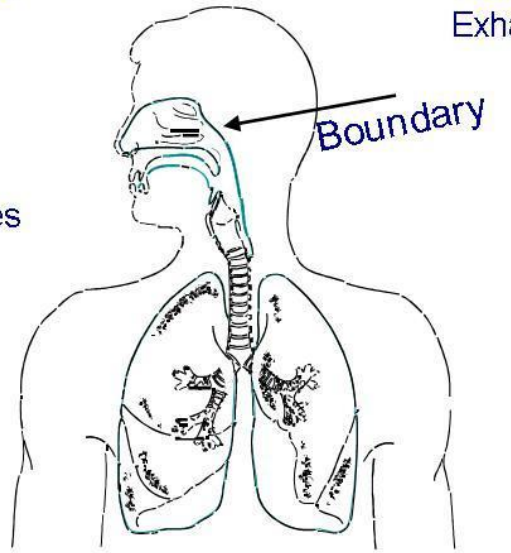
教学提纲

1. 软件工程的产生与发展
2. 软件工程的基本概念
3. 软件质量实现
4. 软件工程学科发展

系统的本质

ENTITIES:
Particulate matter
Oxygen
Carbon dioxide
Water
Nitrogen
Nose
Mouth
Trachea
Bronchial tubes
Lungs
Alveoli

ACTIVITIES:
Inhale gases
Filter gases
Transfer molecules to/from blood
Exhale gases



系统的特征：

- 系统是相互联系的一组元素的集合
- 系统是具有特定功能的有机整体
- 系统是有边界的
- 系统需要与其他系统交互
- 一个系统可能包含另一个系统
- 系统是逐渐演变形成的

系统的本质

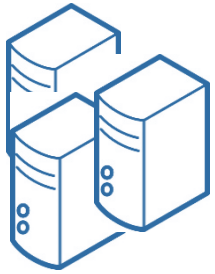
散客预约小程序

人证核验闸机系统

预约管理系统

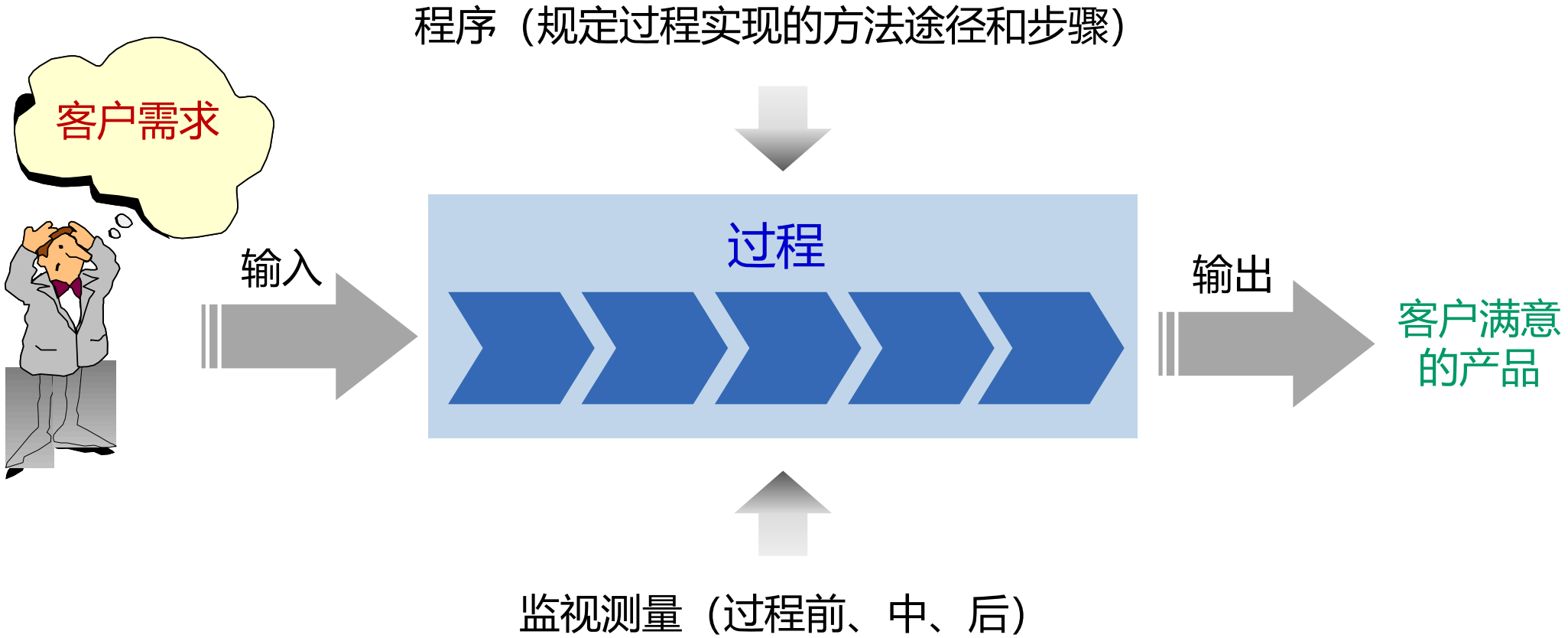
参观清华

团队预约系统



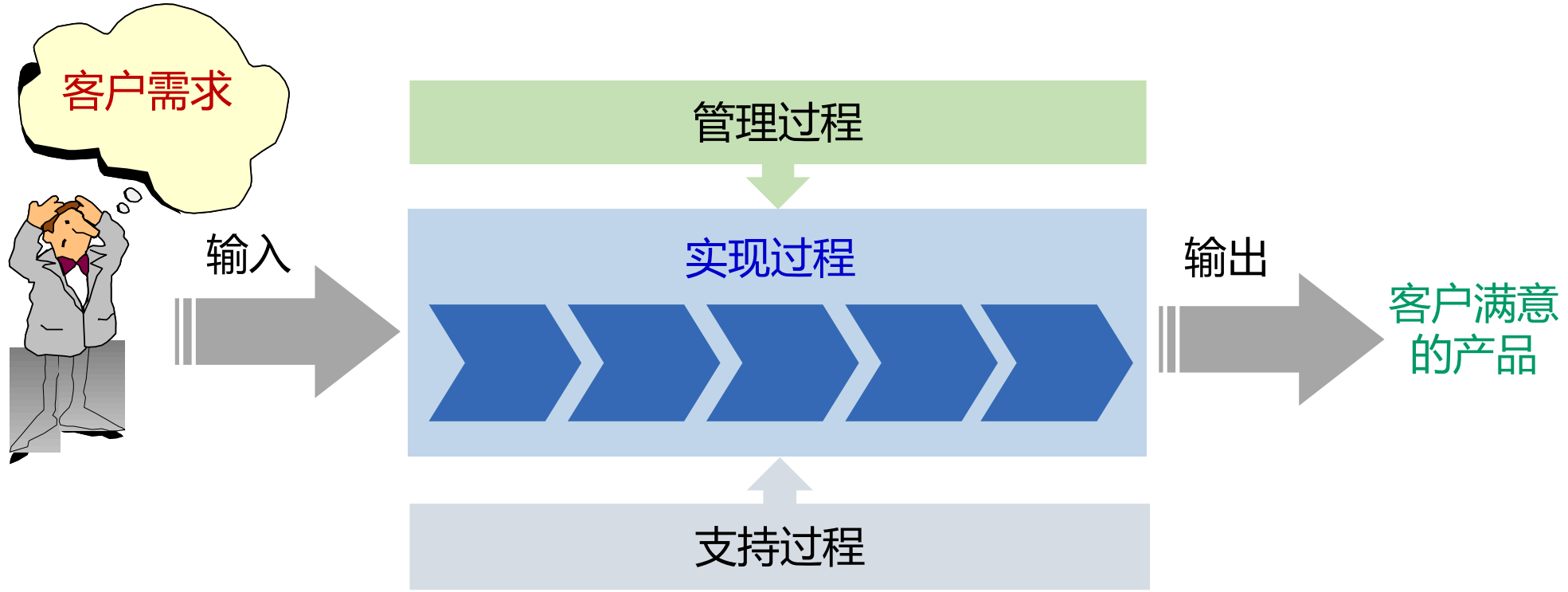
数据管理系统

工程的方法



工程的方法

过程方法是系统地识别和管理组织内所使用的过程，保证更有效地获得期望的结果。



什么是软件工程?

软件工程是 ① 将系统性的、规范化的、可定量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。

软件工程的目标 —— 创造“足够好”的软件

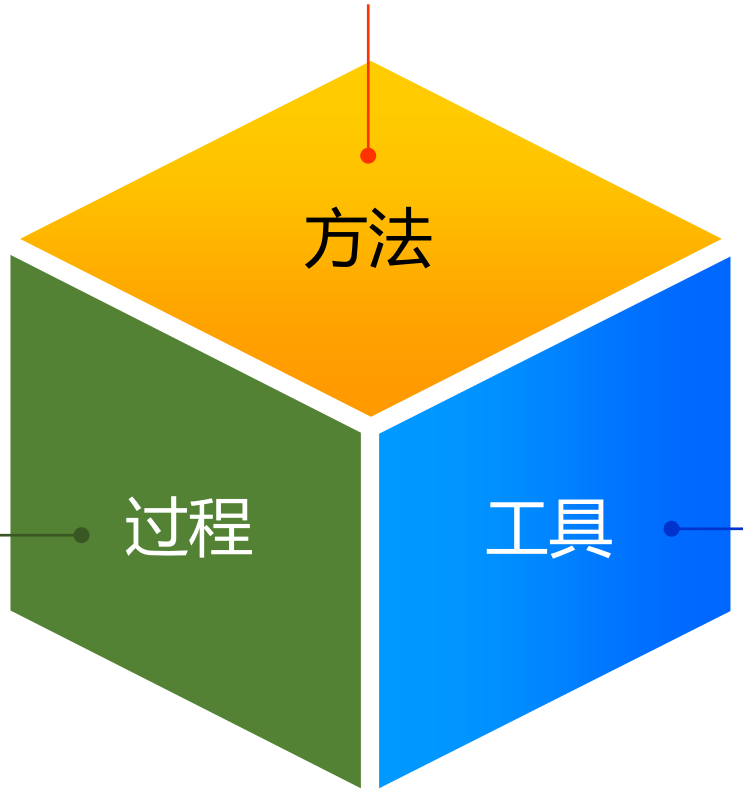
什么是软件工程?

软件工程是 ① 将系统性的、规范化的、可定量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。

- 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
- 具有良好性能、可靠性、可扩展性、可移植性等
- 软件维护费用低

软件工程的基本要素

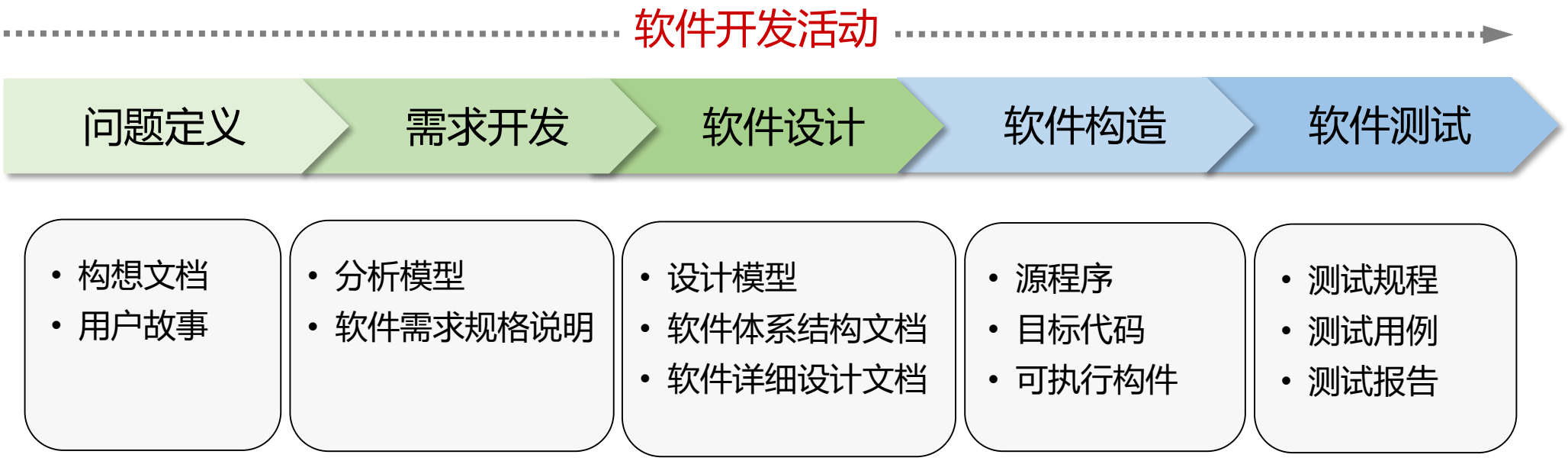
完成软件开发任务的技术手段



支持软件开发各个环节的控制和管理

为软件开发方法提供自动的或半自动的软件支撑环境

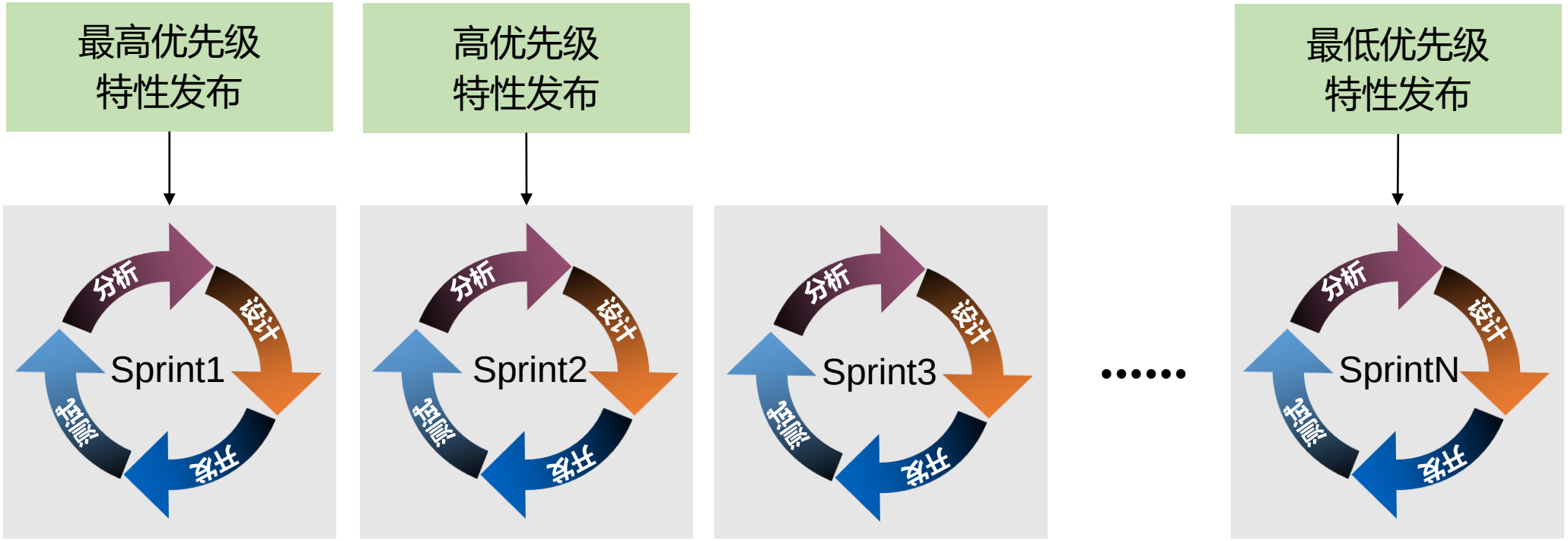
软件工程过程



软件开发管理与支持活动

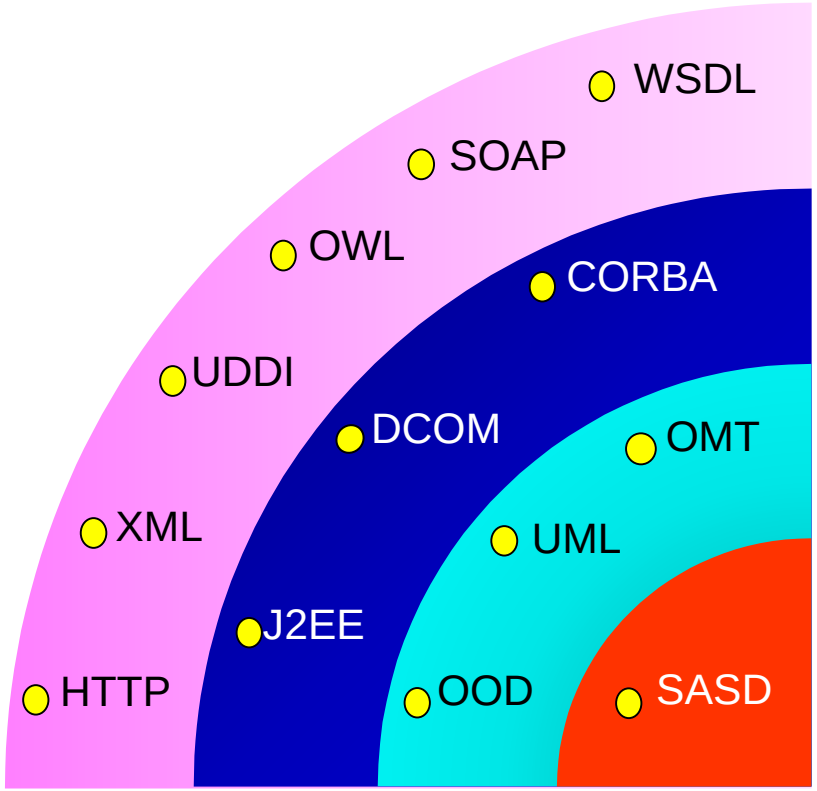
（软件项目管理计划、软件配置管理计划、软件质量保证计划、评审记录……）

软件工程过程



根据团队的开发速率和特性优先级，决定每个迭代应实现的特性

软件工程方法



面向服务：在应用表现层次上将软件构件化，即应用业务过程由服务组成，而服务由构件组装而成。

面向构件：寻求比类的粒度更大的且易于复用的构件，期望实现软件的再工程。

面向对象：以类为基本程序单元，对象是类的实例化，对象之间以消息传递为基本手段。

面向过程：以算法作为基本构造单元，强调自顶向下的功能分解，将功能和数据进行一定程度的分离。

软件工程工具

需求开发	软件设计	软件构造	软件测试	软件维护	开发管理
• 软件建模工具 • 数据库设计工具		• 程序编辑器 • 程序编译器 • 程序解释器 • 程序调试器 • 集成开发环境	• 单元测试工具 • 静态分析工具 • 自动化测试工具 • 性能测试工具 • 缺陷跟踪工具	• 代码重构工具 • 逆向工程工具	• 需求管理工具 • 项目管理工具 • 配置管理工具 • 测试管理工具 • 持续集成工具

Microsoft Visio

MySQL Workbench

软件开发的基本策略

软件复用

- 构造一个新的系统不必从零做起，直接复用已有的构件进行组装
- 构件是经过反复使用验证的，由其组成的新系统具有较高的质量

分而治之

- 将一个复杂的问题分解成若干个简单的问题，然后逐个解决
- 来源于人们生活与工作的经验，完全适合于技术领域

逐步演进

- 软件开发是自底向上逐步有序的生长过程
- 小步快跑：每走完一步再调整并为下一步确定方向，直到终点

优化折中

- 优化：优化软件的各个质量特性，如运行速度、资源利用、用户体验
- 折中：通过协调各个质量特性，实现整体质量的最优

软件开发的基本策略：软件复用

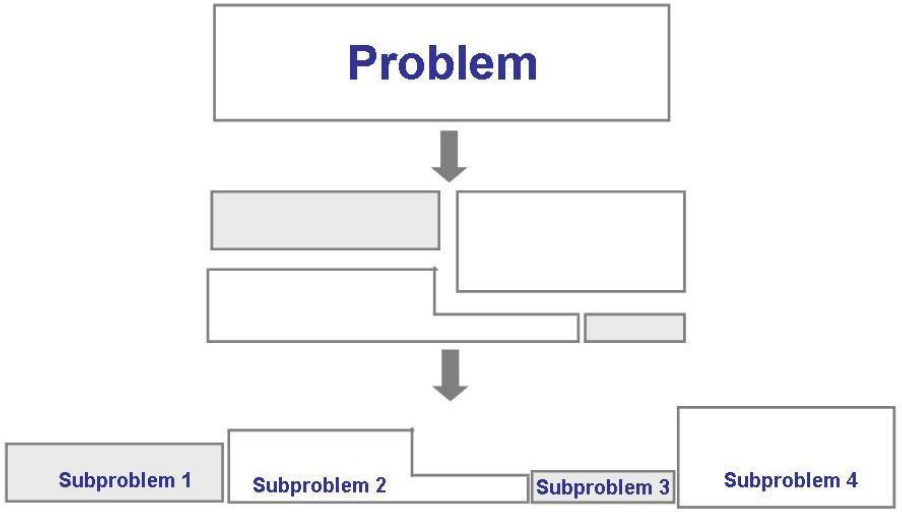
软件复用是利用将已有的软件制品，直接组装或者合理修改形成新的软件系统，从而提高开发效率和产品质量，降低维护成本。

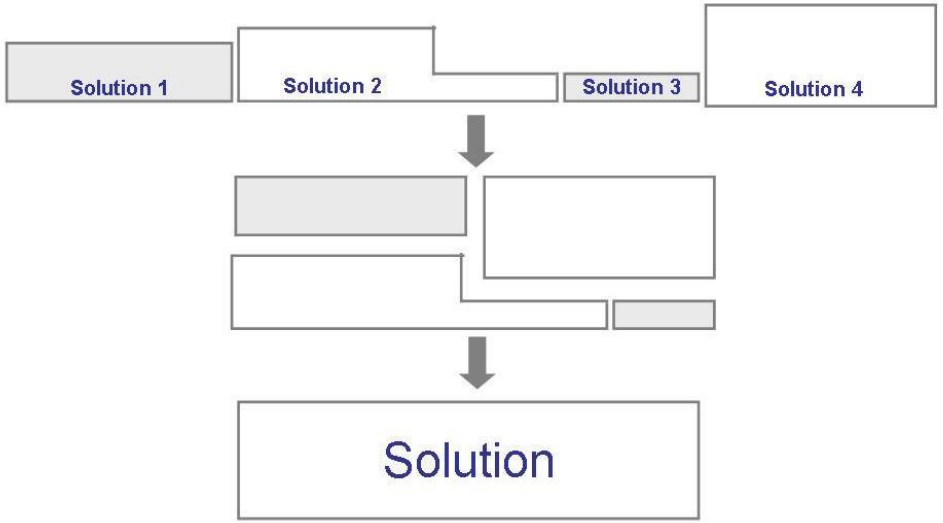
软件复用不仅仅是代码复用

- 库函数、类库
- 模板（文档、网页等）
- 设计模式
- 组件
- 框架

软件开发的基本策略：分而治之

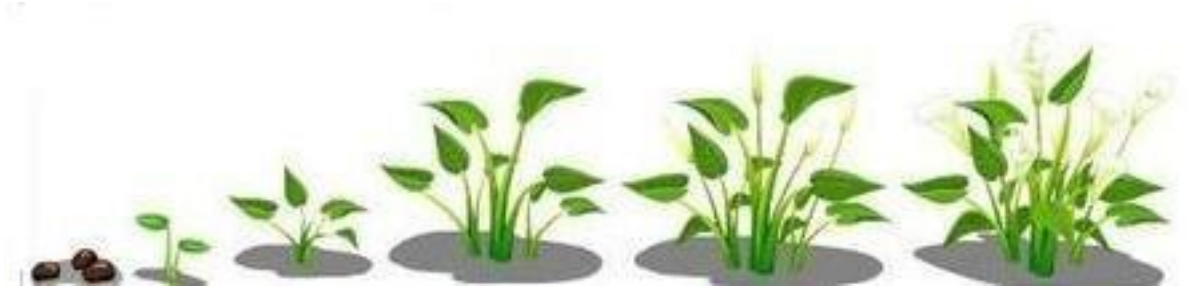
软件工程是一项解决问题的工程活动，通过对问题进行研究分析，将一个复杂问题分解成可以理解并能够处理的若干小问题，然后再逐个解决。





软件开发的基本策略：逐步演进

软件更像一个活着的植物，其生长是一个逐步有序的过程。软件开发应该遵循软件的客观规律，不断进行迭代式增量开发，最终交付符合客户价值的产品。



软件开发的基本策略：优化折中

软件工程师应当把优化当成一种责任，不断改进和提升软件质量；但是优化是一个多目标的最优决策，在不可能使所有目标都得到优化时，需要进行折中实现整体最优。

在编写C程序代码时，对文件的访问是影响程序速度的一个重要因素，那么如何提高文件的访问速度呢?

软件开发的基本策略：优化折中

使用内存缓冲区方法，
读取1468802字节文件

缓冲大小	用户CPU（秒）	系统CPU（秒）	时钟时间（秒）	循环次数（秒）
1	23.8	397.9	423.4	1468802
2	12.3	202.0	215.2	734401
4	6.1	100.6	107.2	367201
8	3.0	50.7	54.0	183601
16	1.5	25.3	27.0	91801
32	0.7	12.8	13.7	45901
64	0.3	6.6	7.0	22951
128	0.2	3.3	3.6	11476
256	0.1	1.8	1.9	5738
512	0.0	1.0	1.1	2869
1024	0.0	0.6	0.6	1435
2048	0.0	0.4	0.4	718
4096	0.0	0.4	0.4	359
8192	0.0	0.3	0.3	180
16384	0.0	0.3	0.3	90
32768	0.0	0.3	0.3	45
65536	0.0	0.3	0.3	23
131072	0.0	0.3	0.3	12

教学提纲

1. 软件工程的产生与发展
2. 软件工程的基本概念
3. 软件质量实现
4. 软件工程学科发展

什么是好的软件

质量就是软件产品对于某个（或某些）人的价值。

—— 杰拉尔德·温伯格

正确的软件 → 一个软件要能够满足用户的需求，为用户创造价值。这里的价值可以体现在两个方面，即为用户创造利润和减少成本。

软件运行正确 → 软件没有或者有很少缺陷，具有很强的扩展性、良好的性能以及较高的易用性等。

什么是好的软件

曾经在全球风靡一时的谷歌眼镜，从2015年1月19日开始不再接受订单。与此同时，谷歌还将关闭其“探索者（Explorer）”软件开发项目，整体转入另外一个部门。

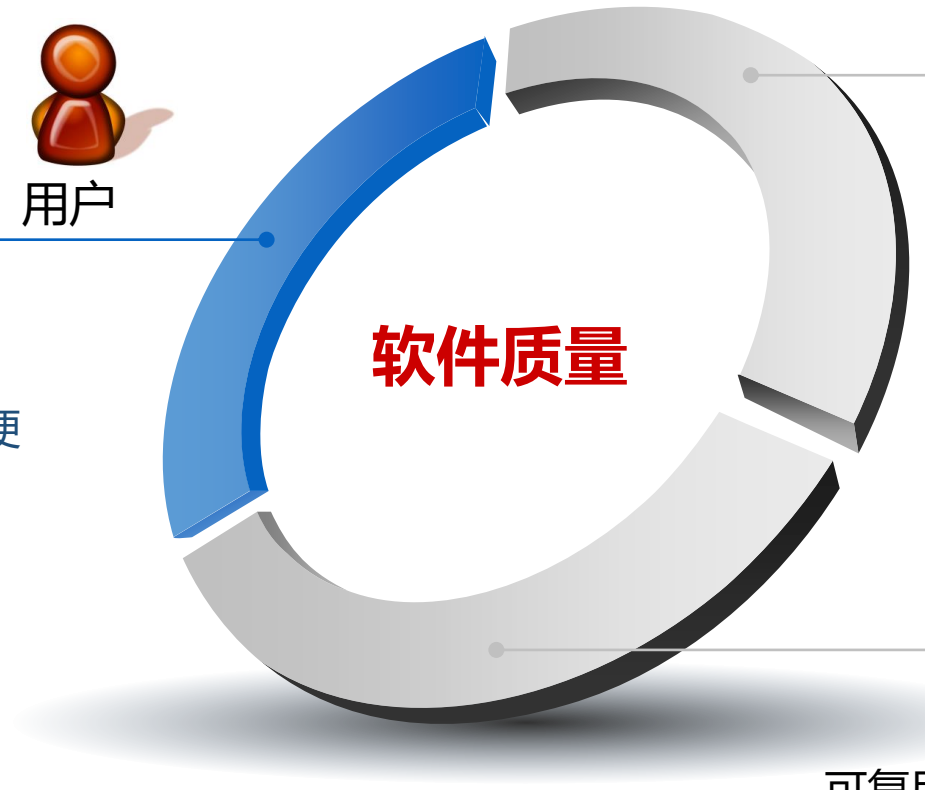
Vista推出后，由于运行效率、兼容性和可靠性等诸多问题，很多用户在安装之后又卸载Vista而退回使用XP，最终微软不得不在两年之后用Win 7取代了Vista。

什么是好的软件

高质量的软件产品：

- 做了用户想要它做的事情
- 正确有效地使用计算机资源
- 易于用户学习和使用
- 设计良好、代码良好且易于测试和维护

什么是好的软件



用户

功能质量

软件符合指定需求
软件几乎没有缺陷
软件性能正常
软件容易上手，操作方便

开发人员

结构质量

代码可测试性
代码可维护性
代码可读性
代码效率：高效管理资源
代码安全：可预防常见威胁

投资者

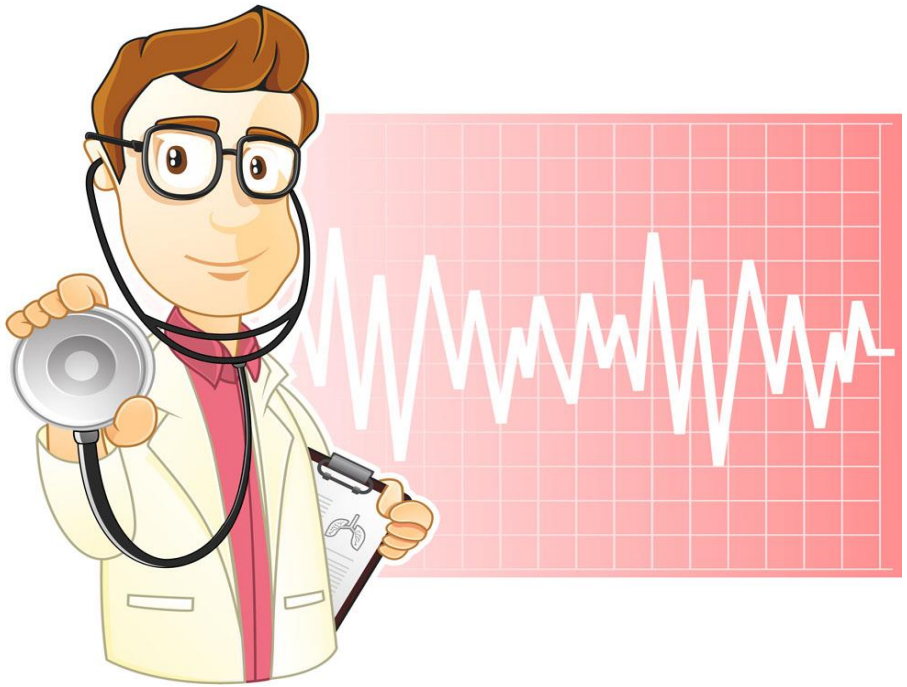
过程质量

软件按时交付
软件满足预算
可复用的开发过程，确保交付质量

产品质量维度

如何判断人是健康的？

产品质量维度



健康体检档案

姓名　　性别　　年龄

婚姻状况　联系电话

家庭住址：

眼科	眼科	辨色力	正常		
		裂隙灯检查右眼	未见异常		
		裂隙灯检查左眼	未见异常		
		其他	屈光不正		
彩超	彩超(肝胆)	肝	肝脏大小形态正常，表面光滑，内部回声分布均匀，肝内胆管显示清晰，血管走向正常，门静脉不宽。		
		胆	胆囊形态大小正常，壁光滑，透声清晰。胆总管不增宽。		
	彩超(双肾)	双肾	双肾大小形态正常，被膜光滑，皮、髓质分界清，集合系统未见分离。		
	彩超(脾)	脾	脾脏大小形态如常，被膜光滑，内部回声均匀，脾门静脉不宽。血流未见异常。		
	彩超(胰)	胰	胰腺大小形态如常，边界清晰，内部回声均匀，主胰管不扩张。		
心电图	12导心电图	12导心电图	窦性心动过缓(心率58次/分)		
		胸部正位片	两侧胸廓对称，气管居中，两肺纹理清晰，未见明显活动性病变，两侧肺门影不大，两侧肋膈角锐利，心脏及大血管未见明显异常。		
		白细胞	4.96	10^9/L	3.97-10.00
		粒细胞比值	54.3	%	40.0-68.0
		淋巴细胞比值	37.2	%	20.0-40.0
		单核细胞比值	5.8		3.0-9.0
		嗜酸粒细胞比值	2.5	%	0.5-8.0
		嗜碱粒细胞比值	0.2	%	0.0-1.2
		粒细胞绝对值	2.69	10^9/L	2.00-6.8
		淋巴细胞绝对值	1.85	10^9/L	0.80-4.00
		单核细胞绝对值	0.29	10^9/L	0.12-8.00
		嗜酸粒细胞绝对值	0.12	10^9/L	0.02-0.8
		嗜碱粒细胞绝对值	0.01	10^9/L	0.00-0.10
		血小板	241	10^9/L	100-300
		血小板平均体积	8.4	fL	7.6-13.2
		血小板分布宽度	16.0		7.0-19.0
		大血小板比率	0.202		0.080-0.430
		红细胞	4.92	10^12/L	3.50-5.50
		红细胞压积	43.4	%	38.0-50.8
		血红蛋白	150	g/L	120-170
		红细胞分布宽度CV	13.1		11.0-16.0
		红细胞分布宽度SD	44.8		35.0-56.0
		红细胞平均体积	88.2	fL	80.0-100.0
		平均血红蛋白含量	30.5	pg	27.0-34.0
		平均血红蛋白浓度	346	g/L	310-378
		比重	1.015		1.005-1.03
		PH值	7.0		5.0-8.0
		亚硝酸盐	阴性		阴性
		尿蛋白	阴性		阴性
		葡萄糖	阴性		阴性
		潜血	阴性		阴性
		尿胆红素	阴性		阴性
		尿胆原	阴性		阴性

ISO9126 质量模型

ISO9126 质量模型

功能性

- 适合性：当软件在指定条件下使用，其满足明确和隐含要求功能的能力。
- 准确性：软件提供给用户功能的精确度是否符合目标。
- 互操作性：软件与其它系统进行交互的能力。
- 安全性：软件保护信息和数据的安全能力。

可靠性

- 成熟性：软件产品避免因软件中错误发生而导致失效的能力。
- 容错性：软件防止外部接口错误扩散而导致系统失效的能力。
- 可恢复性：系统失效后，重新恢复原有的功能和性能的能力。

ISO9126 质量模型

易用性

- 易理解性：软件显示的信息要清晰、准确且易懂，使用户能够快速理解软件。
- 易学习性：软件使用户能学习其应用的能力。
- 易操作性：软件产品使用户能易于操作和控制它的能力。
- 吸引性：软件具有的某些独特的、能让用户眼前一亮的属性。

效率

- 时间特性：在规定的条件下，软件产品执行其功能时能够提供适当的响应时间和处理时间以及吞吐率的能力。
- 资源利用：软件系统在完成用户指定的业务请求所消耗的系统资源，诸如CPU占有率、内存占有率、网络带宽占有率等。

ISO9126 质量模型

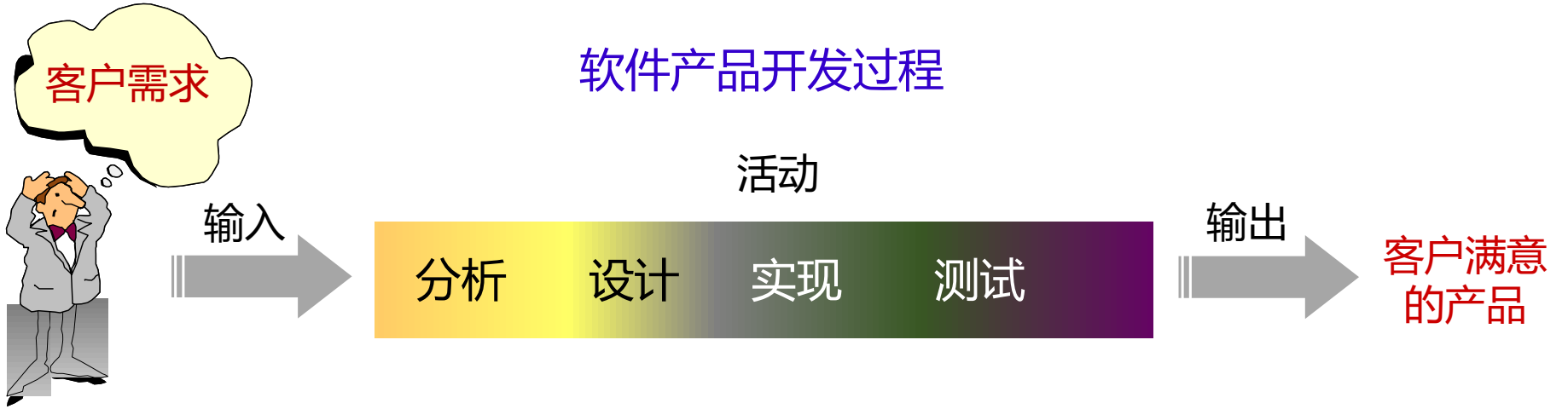
可维护性

- 易分析性：软件提供辅助手段帮助开发人员定位缺陷原因并判断出修改之处。
- 易改变性：软件产品使得指定的修改容易实现的能力。
- 稳定性：软件产品避免由于软件修改而造成意外结果的能力。
- 易测试性：软件提供辅助性手段帮助测试人员实现其测试意图。

可移植性

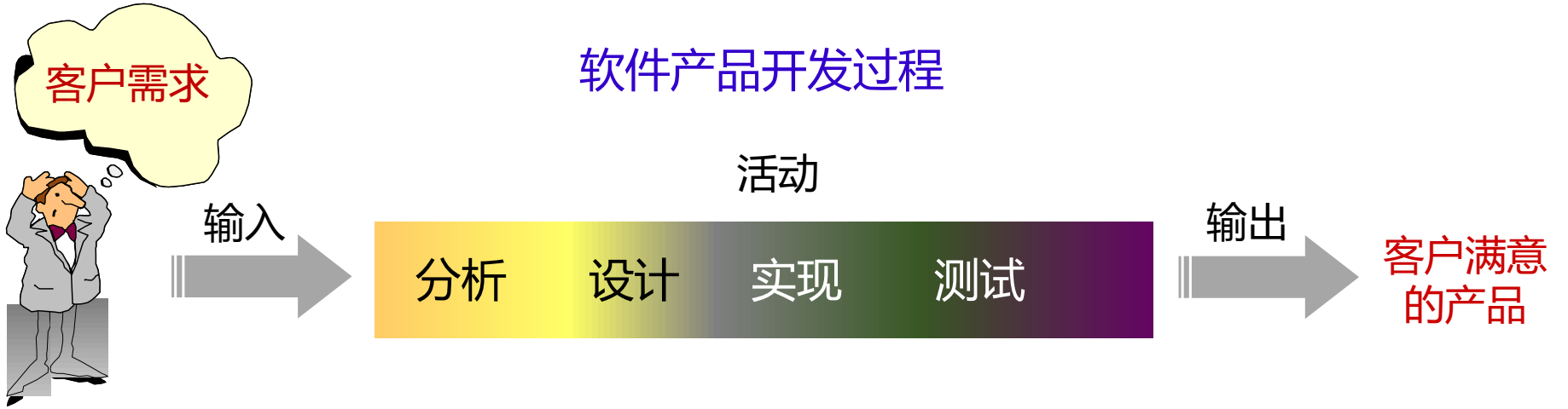
- 适应性：软件产品无需做任何相应变动就能适应不同运行环境的能力。
- 易安装性：在平台变化后，成功安装软件的难易程度。
- 共存性：软件产品在公共环境与其共享资源的其他系统共存的能力。
- 替换性：软件系统的升级能力，包括在线升级、打补丁升级等。

实现软件质量



软件质量是如何实现的？如何才能有效地提高质量？

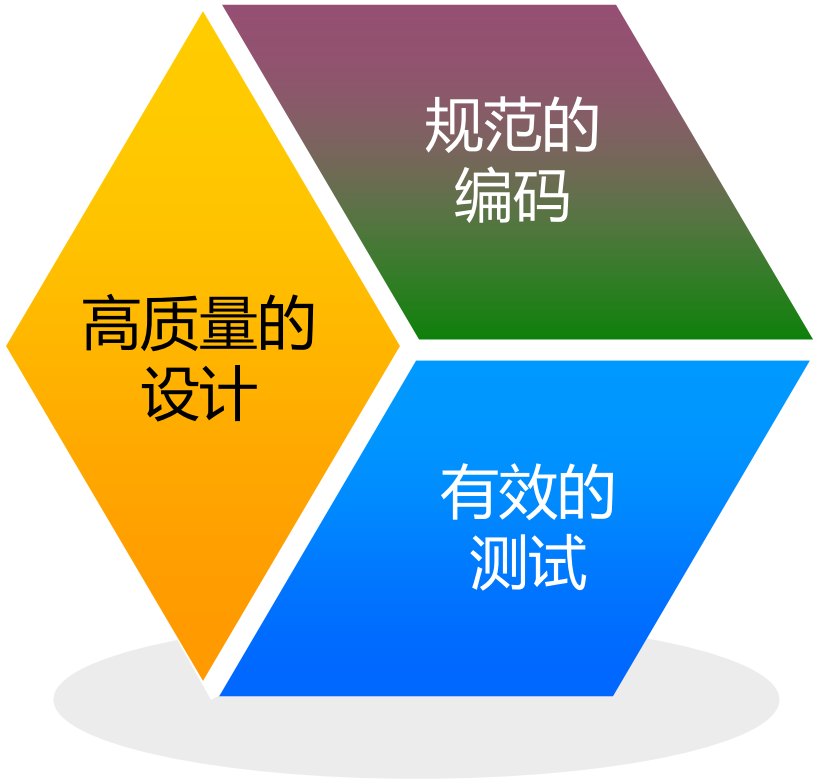
实现软件质量



- 质量不是被测出来的，而是在开发过程中逐渐构建起来的
- 虽然质量不是测出来的，但是未经测试也不可能开发出高质量的软件
- 质量是开发过程的问题，测试是开发过程中不可缺少的重要环节

实现软件质量

高质量软件
开发之道



商业环境下的软件质量

软件质量的重要性是无容置疑的，
那么是不是质量越高就越好？

软件产品是否应该追求“零缺陷”？

商业环境下的软件质量

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都签了生死状，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

商业目标决定质量目标：

- 商业目标决定质量目标，不应该把质量目标凌驾于商业目标之上
- 质量是有成本的，不可能为了追求完美的质量而不惜一切代价
- 理想的质量目标不是“零缺陷”，而是恰好让广大用户满意

$oftware

教学提纲

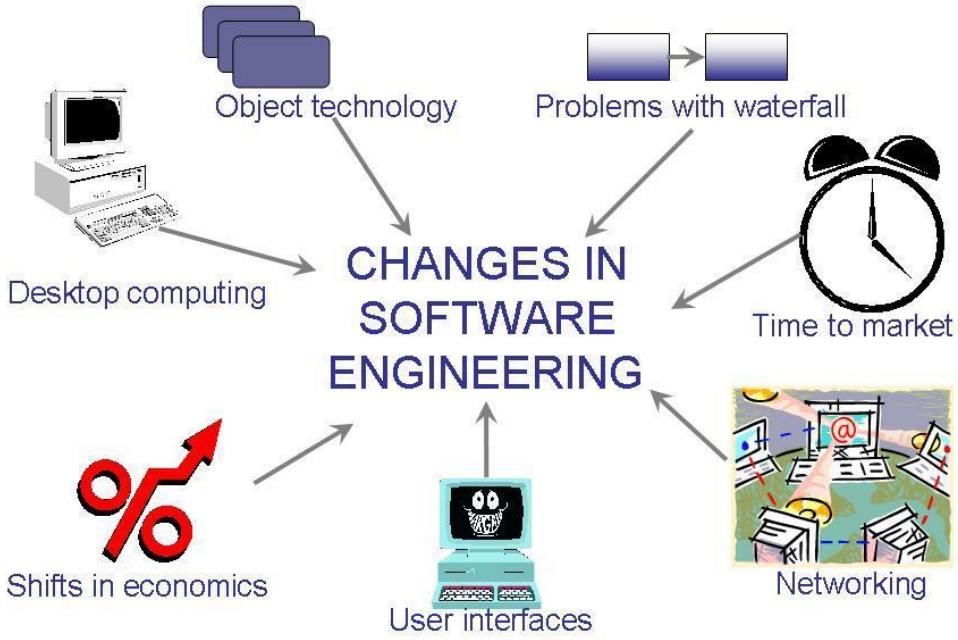
1. 软件工程的产生与发展
2. 软件工程的基本概念
3. 软件质量实现
4. 软件工程学科发展

软件工程学科发展

Wasserman, Anthony I., “Toward a discipline of software engineering”, *IEEE Computer* , Vol. 13, No. 6, pp.23-31, 1996

改变软件工程的关键因素：

- 交付时间的重要性
- 计算技术在经济中的转变
- 功能强大的桌面计算
- 互联网络
- 面向对象技术
- 图形用户界面
- 瀑布模型的不可预知性



软件工程学科发展

抽象

软件建模方法

用户界面原型化

软件体系结构

Wasserman's Discipline of Software Engineering

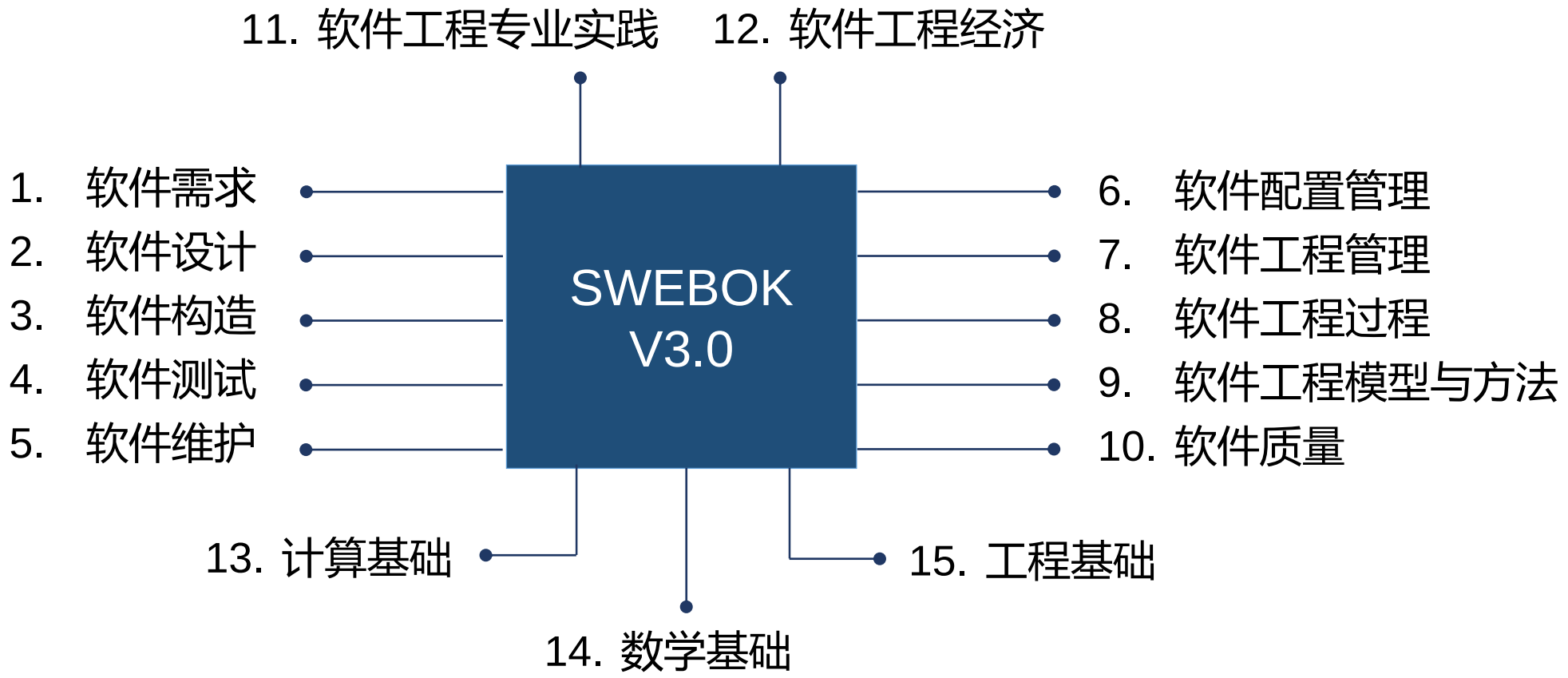
软件过程

软件复用

度量

工具与集成环境

软件工程知识领域


11. 软件工程专业实践
12. 软件工程经济
1. 软件需求
2. 软件设计
3. 软件构造
4. 软件测试
5. 软件维护
SWEBOK
V3.0
6. 软件配置管理
7. 软件工程管理
8. 软件工程过程
9. 软件工程模型与方法
10. 软件质量
13. 计算基础
14. 数学基础
15. 工程基础

软件工程职业道德规范

IEEE / ACM 职业道德准则

1. 公众：软件工程人员应始终与公众利益保持一致。
2. 客户和雇主：在与公众利益保持一致的原则下，软件工程人员应满足客户和雇主的最大利益。
3. 产品：软件工程人员应当确保他们的产品及其改进符合尽可能高的专业标准。
4. 判断：软件工程人员应当具备公正和独立的职业判断力。
5. 管理：软件工程管理者和领导者应拥护和倡导合乎道德的有关软件开发和维护的管理方法。
6. 职业：在与公众利益一致的原则下，软件工程人员应当提高职业的信誉。
7. 同行：软件工程人员对其同行应持平等和支持的态度。
8. 自我：软件工程人员应当终身学习专业知识，促进合乎道德的职业实践方法。

CMU ICS 课程条例

不被允许的使用他人信息的行为：

- 剽窃代码：包括复制、上网搜索、重新输入、窃视他人文档和代码。
- 重用原先的课程或者网上的代码和解决方案。

不被允许的提供自己信息的行为：

- 分享代码
- 口头为他人描述代码（注：描述具体的代码段）
- 为他人提供非常细节的指导

CMU ICS 课程条例

不属于作弊甚至应该鼓励的行为：

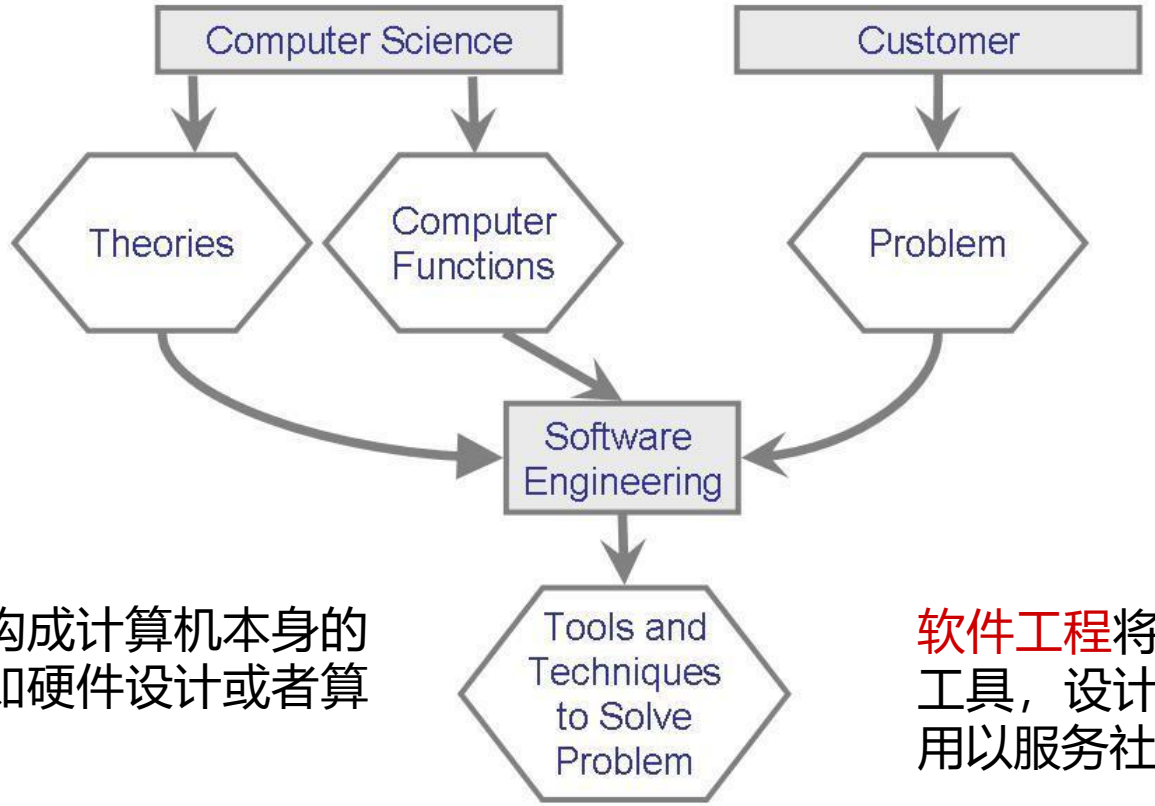
- 帮助他人如何使用各种工具，包括但不限于系统、网络、编译器、调试器。
- 帮助他人关于高层次的设计问题，例如探讨整个问题的框架。如果需要讨论的问题要用代码来描述，就不是高层次的设计问题。
- 帮助他人关于高层次的debug，例如你有一个流程图，但思路受阻，可以请教他人，他人指出你流程图中的某一个流程不对。
- 向老师和助教请求帮助，使用老师或助教或课程提供的代码和页面。

软件工程课程条例

- 应用 CMU ICS 课程条例，违犯条例的结果：
  （1）最好的情况是当次作业成绩为0分
  （2）最坏的情况是课程成绩不及格
- 尊重他人的知识产权，如果你使用开源或他人的成果（诸如代码、文档、设计方案等），需要获得允许并按照协议使用，明确注明并给出具体来源，课程将仅评价你的贡献
- 个人应该有责任参与行业发展和技术进步，鼓励大家对外发布博客文章进行技术和经验分享

软件工程与计算机科学的关系

科学是发现世界上已经存在的事物，回答“是什么”和“为什么”的问题。
工程是创造世界上从未存在的事物，回答“做什么”和“怎么做”的问题。



计算机科学研究构成计算机本身的理论和结构，诸如硬件设计或者算法的理论证明等。

软件工程将计算机作为问题求解的工具，设计和实施尚未存在的方案用以服务社会。

科学思维与工程思维

认知和探索

科学思维

筹划和构建

工程思维

科学思维与工程思维

人工智能的科学与工程

科学目标：理解智能行为，无论其是表现在机器内，还是人或其他生物内

工程目标：开发与人一样出色，甚至比人更出色的，具有智能行为的机器

谢谢大家！

THANKS