软 件 工 程

第一章 软件工程学概述

2020 / 09 / 21

本章概述

迄今为止，计算机系统已经经历了4个不同的发展阶段，但是，人们仍然没有彻底摆脱“**软件危机**”的困扰，软件已经成为限制计算机系统发展的瓶颈

为了更有效地开发与维护软件，软件工作者在20世纪60年代后期开始认真研究消除软件危机的途径，从而逐渐形成了一门新兴的工程学科——**软件工程**

内容提要

01 软件的定义

02 软件的本质特征

03 软件危机

04 软件工程

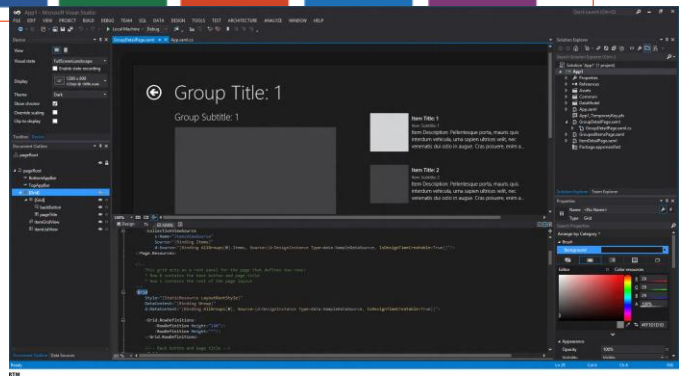
软件无处不在


Windows 10

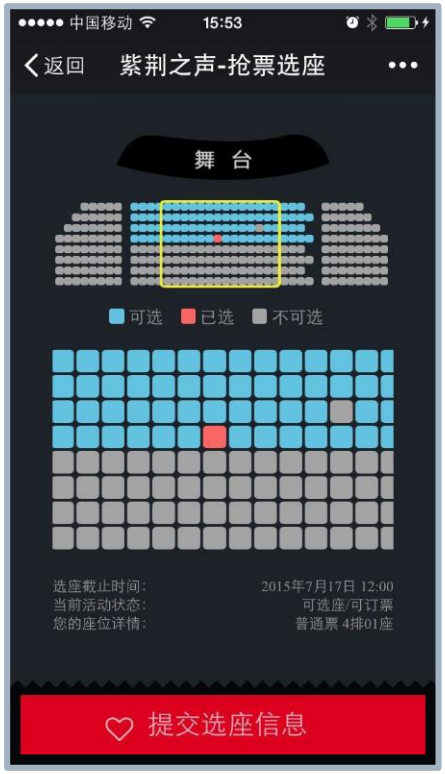

滴滴出行
滴滴一下美好出行


Office


白条半岁啦！"分"享88折
JD.COM 京东
旅行撒欢


Group Title: 1
Group Subtitle: 1

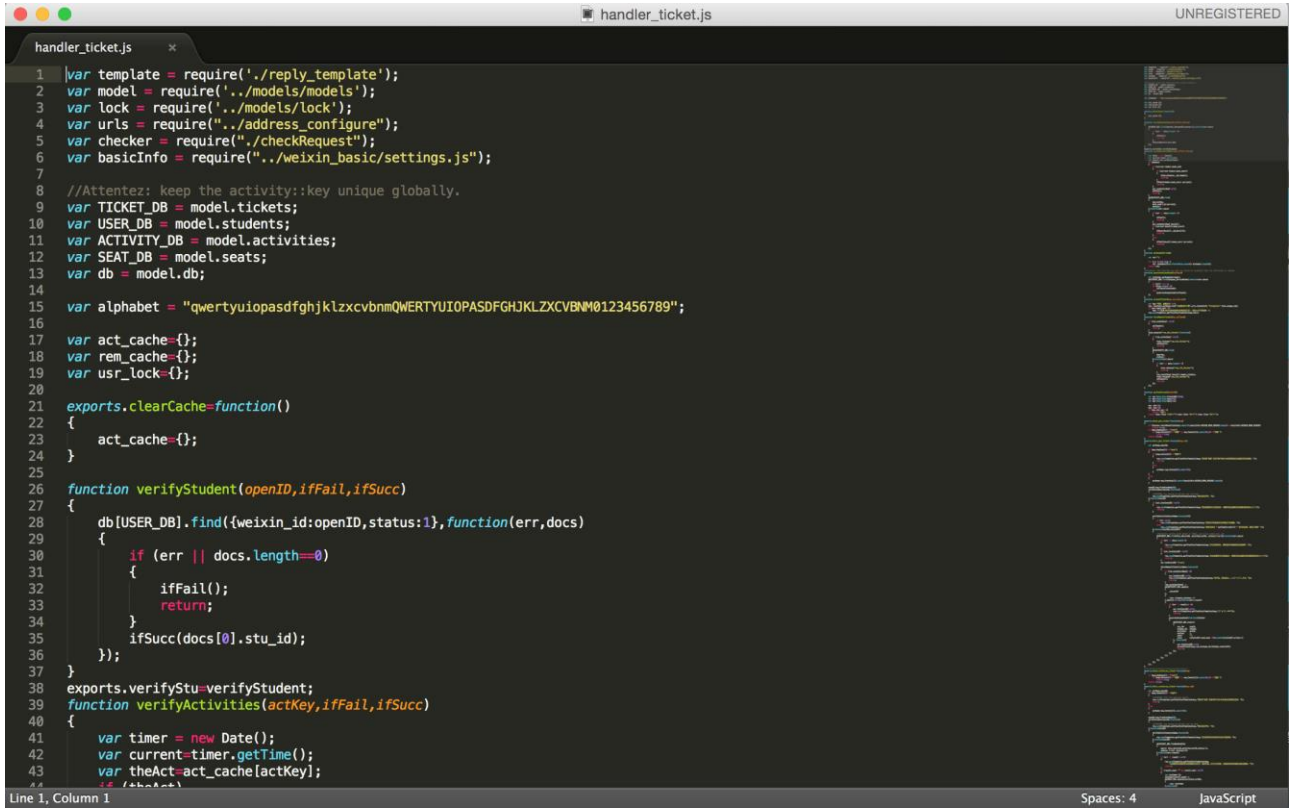
软件的定义



一个订电影票的微信应用，用户端看到的，就是图形化界面，

但是在计算机内部是如何执行的呢?

软件的定义



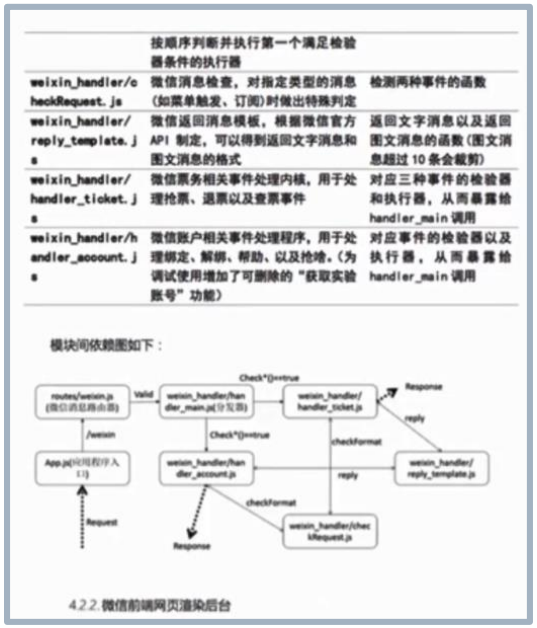
首先，计算机内部实际运行着**一些代码**，这些代码负责向计算机发出指令以实现订票、生成电子票等不同功能

软件的定义

key	String	每个活动的活动代称，文字抢票的活动关键字，有唯一性
place	String	活动地点
description	String	活动描述
pic_url	String	活动配图的url地址
start_time	Int	活动开始时间，存为从1970年1月1日经过的毫秒数
end_time	Int	活动结束时间，存为从1970年1月1日经过的毫秒数
book_start	Int	抢票开始时间，存为从1970年1月1日经过的毫秒数
book_end	Int	抢票结束时间，存为从1970年1月1日经过的毫秒数

第二，在计算机运行程序的过程中，还需要学生信息、电影信息、电子票等**相关数据**

软件的定义



	按顺序判断并执行第一个满足检验器条件的执行器	
weixin_handler/checkRequest.js	微信消息检查，对指定类型的消息（如菜单触发、订阅）时做出特殊判定	检测两种事件的函数
weixin_handler/reply_template.js	微信返回消息模板，根据微信官方API制定，可以得到返回文字消息和图文消息的格式	返回文字消息以及返回图文消息的函数（图文消息超过10条会被剪）
weixin_handler/handler_ticket.js	微信票务相关事件处理内核，用于处理抢票、退票以及查票事件	对应三种事件的检验器和执行器，从而暴露给handler_main调用
weixin_handler/handler_account.js	微信账户相关事件处理程序，用于处理绑定、解绑、帮助、以及抢啥。（为调试使用增加了可删除的“获取实验账号”功能）	对应事件的检验器以及执行器，从而暴露给handler_main调用

模块间依赖图如下：

4.2.2. 微信前端网页渲染后台

4.2.3. 后台管理模块

后台管理后端(routes/user_manage.js)：

查找数据库中所有暂存、已发布的活动并将它们的信息返回给活动列表页进行渲染；导出 xlsx，根据前端发送的活动 id 查找数据库中对应活动的所有有效票，并将每张票的持有者学号、支付状态、入场状态、座位号等信息汇总生成 xlsx 表格返回前端；删除活动，根据活动 id 将某个活动删除（status 改为-1）；根据活动 id 查找 activity、seat 数据库，并将具体信息返回前端给活动详情页面进行渲染；活动详情写入数据库，根据前端 post 的内容，检查活动内容是否完整，格式是否正确，内容是否符合时间段的特点（某些信息在某些活动阶段里不允许更改），然后选择是否录入数据库，并将成功或失败信息返回前端。

后端对要录入数据库的内容进行了严格的审查，相当于前端检查过一次的内容，后端也进行一次审核（双审核）。这样费周章的进行两次检验有必要吗？其实还是很有必要的，因为前端是很不靠谱的，用户可以改，甚至可以直接给后端 post 一些杂乱的内容，如果后端不做进一步的检查那么数据库很可能就会发生错误；即使用户并没有想恶意攻击，但也会有一些意外情况，比如用户的系统时间不正确，而前端是通过用户设备的时间来判断一个活动处于哪个阶段，进而判断哪些字段可以或不允许被更改，一旦用户时间和服务器时间不匹配，后端又不做检查，就可能发生一些严重错误——明明抢票开始了，总票数不允许更改，但用户系统时间是抢票前，导致该活动总票数在抢票中被改大或改小。因此后端的这个部分写了很长来进行严密的内容审查。

第三，为了后续的维护和开发，还需要一些描述技术，实现细节的**开发文档**

软件的定义

软件 = 程序 + 数据 + 文档

- 程序：计算机可接受的一系列指令
- 数据：使得程序能够操作信息的数据结构
- 文档：描述程序的研发过程、方法和使用的图文资料

软件的定义

现在思考一个问题：

程序数据和文件是否就代表了软件的真正含义呢?

内容提要

01 软件的定义

02 软件的本质特征

03 软件危机

04 软件工程

软件的本质

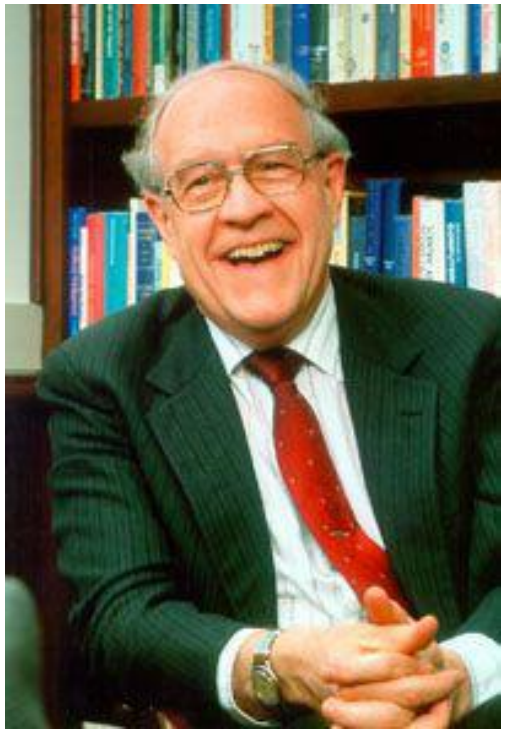
波音公司的客机由数百万个零件组成，需要上千人组装，但通常能够按预算交付

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

软件的本质

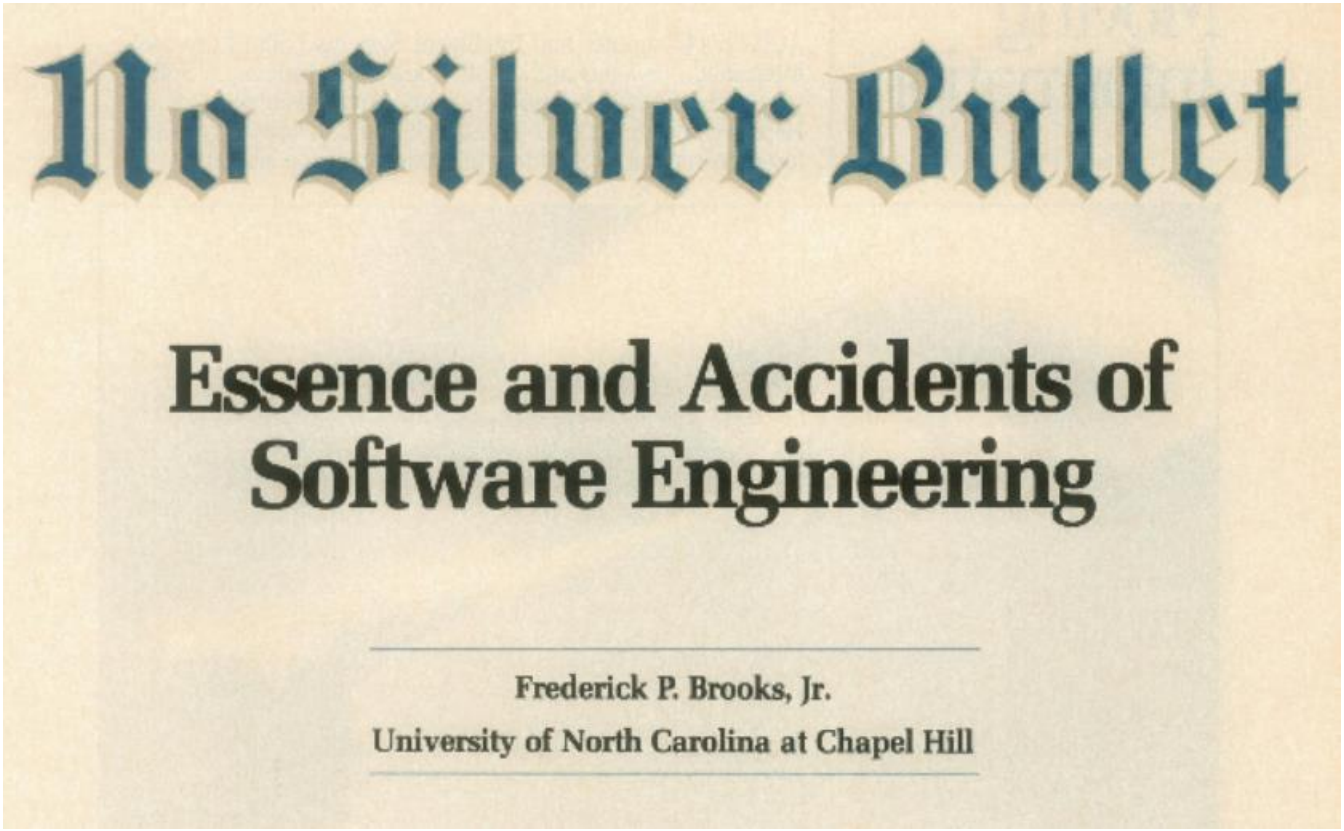
开发软件与开发客机有什么本质区别?

软件的本质



Fred Brooks，北卡罗莱纳大学的计算机科学教授，曾担任IBM OS360项目经理，在计算机体系结构、操作系统和软件工程方面做出了里程碑似的卓越贡献，于1999年获得了计算机领域最具声誉的图灵奖

软件的本质



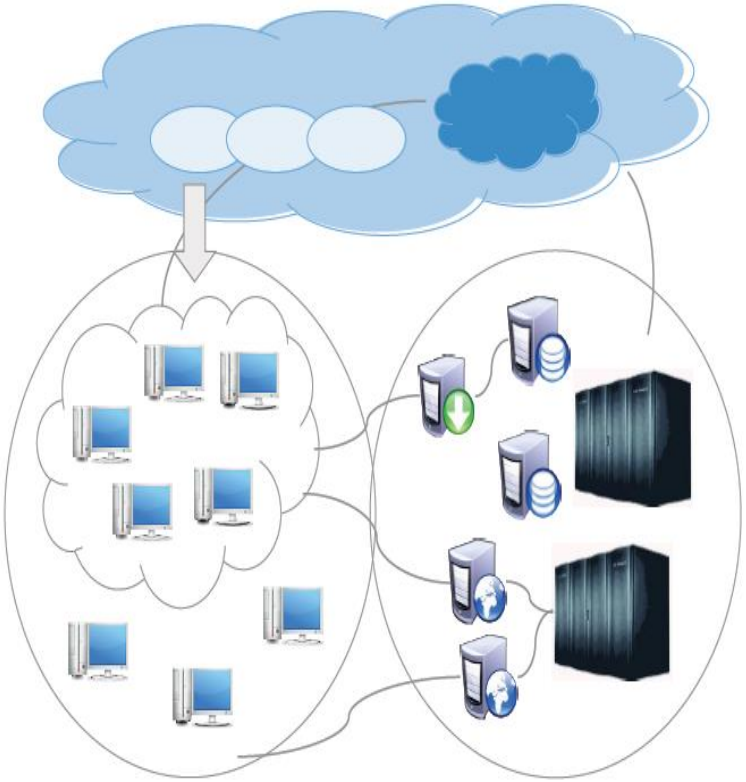
No Silver Bullet

Essence and Accidents of Software Engineering

Frederick P. Brooks, Jr.
University of North Carolina at Chapel Hill

他于1987年发表了一篇《没有银弹》的文章，指出软件具有**复杂性、一致性、可变性和不可见性**等内在特性，是造成软件开发困难的根本原因

软件的本质（复杂性）



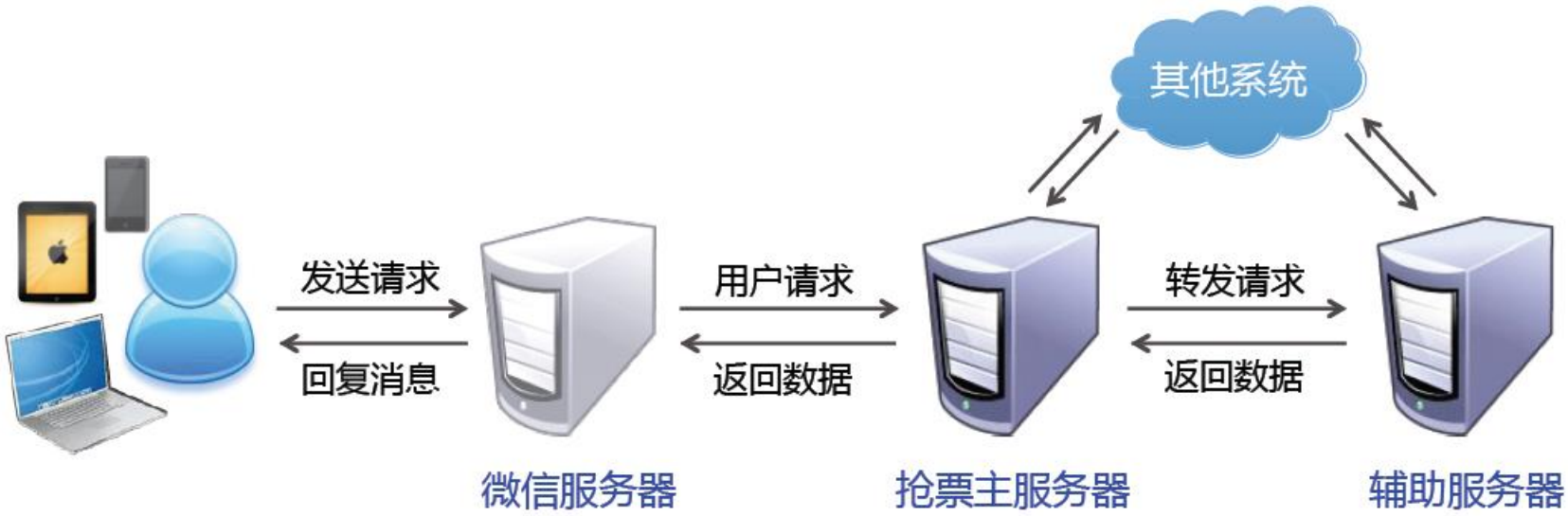
Google搜索引擎建立在全球30多个站点、超过100万台服务器的云计算设施上

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（一致性）

软件依赖于不同的硬件、网络以及其它软件，在这个过程中，接口会不断改变。但消息的发送、回复要保持一致性



软件的本质（可变性）

2010.11.20
微信正式立项

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）

2012

- 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）
- 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）

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（不可见性）

- 软件是一种“看不见，摸不着”的逻辑体
- 软件是以机器代码在运行，但开发人员无法看到源代码是如何执行的

单选题 1分

设置

软件会逐渐退化而不会磨损，其原因在于（　）

A 软件通常暴露在恶劣的环境下

B 软件错误在经常使用后会逐渐增加

C 不断的变更使接口之间引起错误

D 软件备件很难订购

提交

内容提要

01 软件的定义

02 软件的本质特征

03 软件危机

04 软件工程

软件危机

- **早期：上世纪60年代中期之前**
  - 软件为每个具体应用专门编写，规模小
  - 编写者和使用者往往是同一个人，只有程序，没有文档
- **第二个时期：上世纪60年代中期至 70年代中期**
  - 广泛使用软件产品，数量急剧增加
  - 个体化特性使程序不可维护
  - “软件危机”出现
  - “软件工程”被提出

软件危机

典型案例

- 1963年，IBM/360操作系统OS/360
- 100万行代码，4000多个模块
- 1000人：开发人员最多时1000人
- 4年：从1963年到1966年
- 数亿美元，约5000人年工作量
- 延期交付

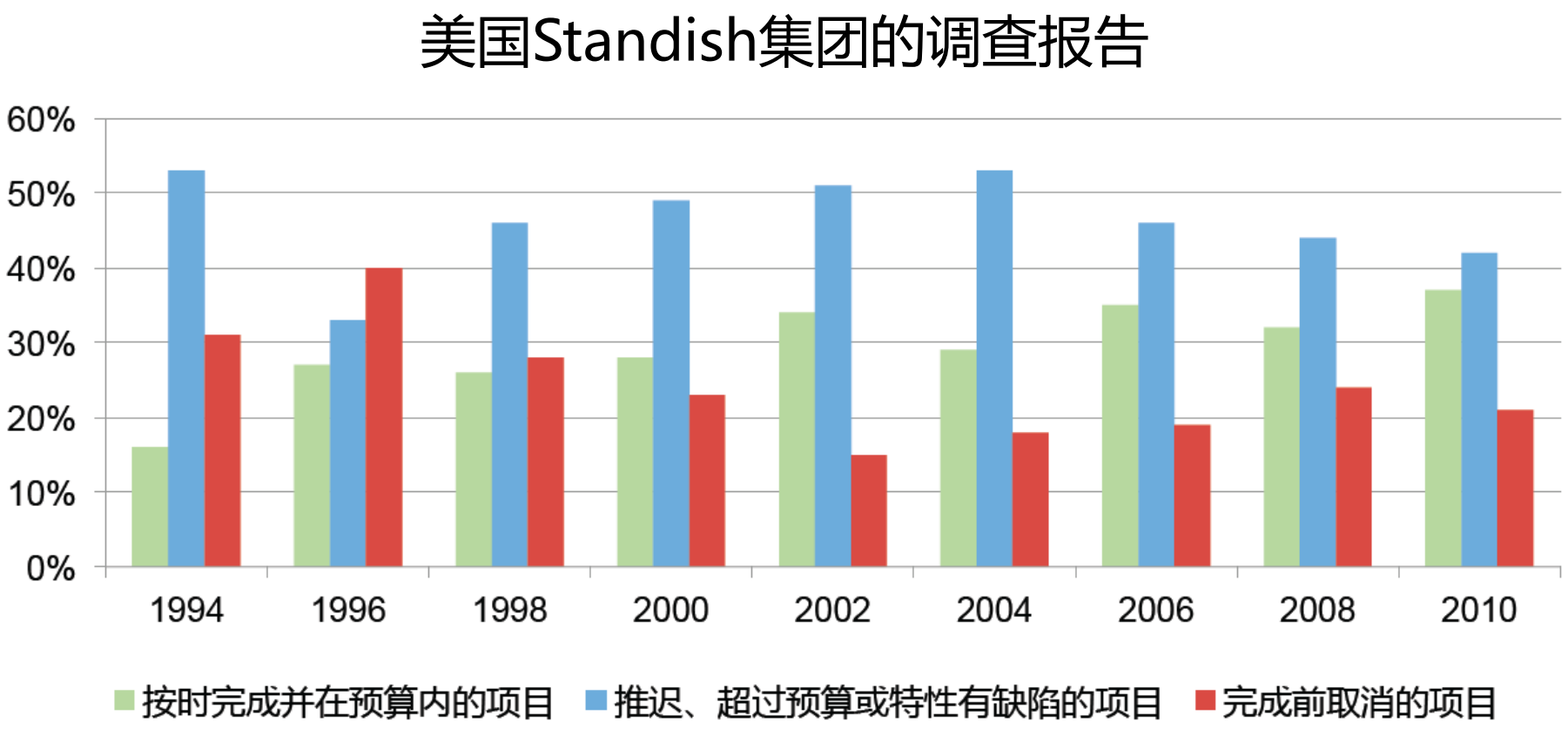
每一个新版本都是对前一个版本修复1000个BUG的结果！！！

软件危机

项目负责人Brooks事后总结了他的沉痛教训时说：

“正像一只逃亡的野兽落到泥潭中做垂死的挣扎，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潭中挣扎，谁也没有料到问题竟会陷入这样的困境”

软件危机



软件项目的成功率大约只有30%，其中一半左右超出预算和最后的期限，还有其中还有20%是即使投入了极大努力，也未能完成

软件危机 （案例1：ARIANE5）

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数亿美元的通信卫星

原因：

- 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

软件危机 （案例2：VISTA）

- 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
- 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
- 公开测试以来，程序错误总数已经超过2万个，还不包括微软未公开的一些错误

软件危机 （案例3：12306）

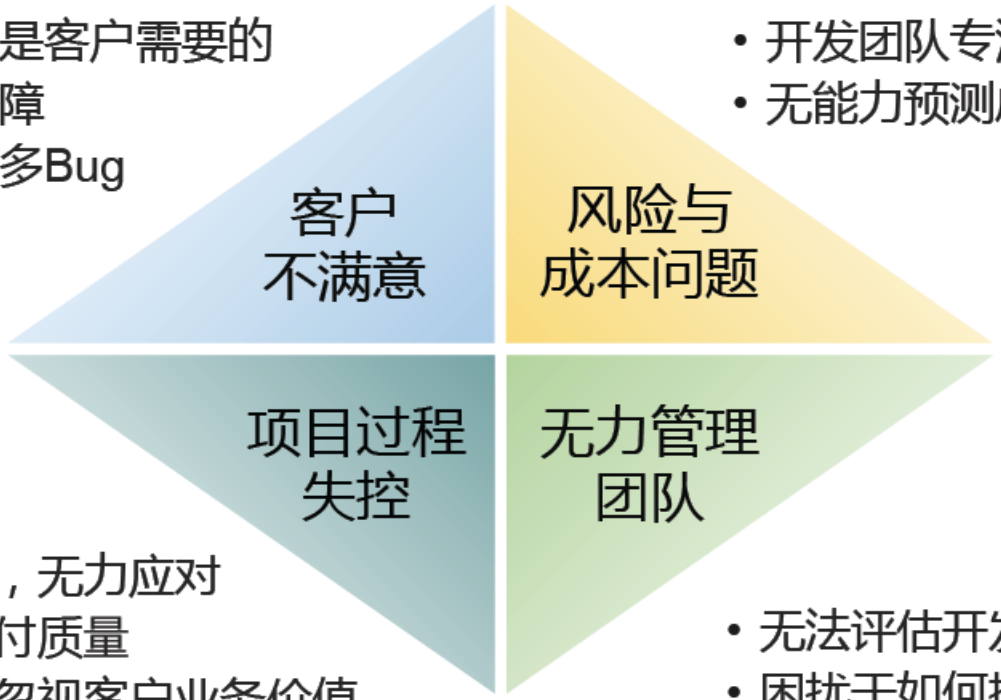
- 12306网络购票系统历时两年研发成功，耗资3亿元人民币，于2011年6月12日投入运行
- 2012年1月8日春运启动，9日网站点击量超过14亿次，出现网站崩溃、登录缓慢、无法支付、扣钱不出票等严重问题
- 2012年9月20日，由于正处中秋和“十一”黄金周，网站日点击量达到14.9亿次，发售客票超过当年春运最高值，网络拥堵

软件危机 （更多案例）

- 美国交警的雷达测速枪锁定战斗机，触发反导系统发射
- Therac-25放射治疗仪发出高剂量射线造成病人死亡，因为操作员按键太快导致程序出错
- 英国救护车呼叫系统，新系统布署前没有过渡，导致救护车调度混乱，病人抢救延误
- 1967年苏联“联盟一号”载人飞船由于软件忽略一个小数点，导致打不开降落伞，在进入大气层时烧毁

软件危机的特点

- 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

单选题 1分

设置

Ariane 5火箭发射失败的事例告诉我们（　）。

A 系统环境的变化可能影响软件采集的精度

B 软件后备系统可能通过复制产生

C 软件重用必须重新进行系统论证和测试

D 选项 A 和 C

E 选项 A、B 和 C

提交

内容提要

01 软件的定义

02 软件的本质特征

03 软件危机

04 软件工程

软件工程

古人云：诸事有道

软件工程一直致力于

探索软件开发问题的解决之道

软件工程

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**的概念和术语

软件工程

- 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

软件发展的四个阶段

软件工程

工程是将理论和知识应用于实践的科学，以便有效地解决问题

- 大规模的设计与建造
- 复杂问题与目标分解
- 团队协作与过程控制

软件工程

软件工程的定义

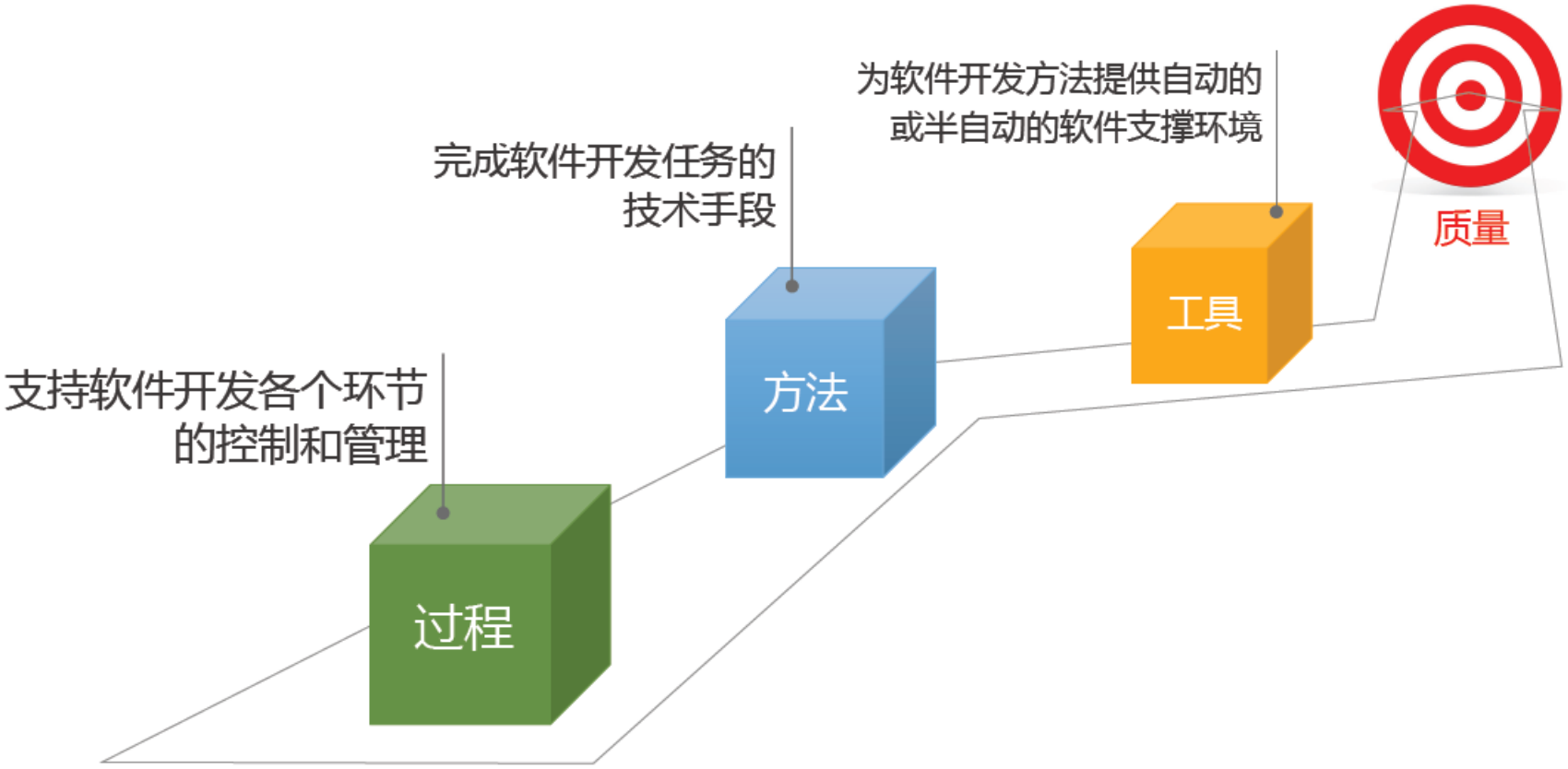
① 将系统性的、规范化的、可定量的方法应用于软件的开发、运行和维护，即工程化应用到软件上

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

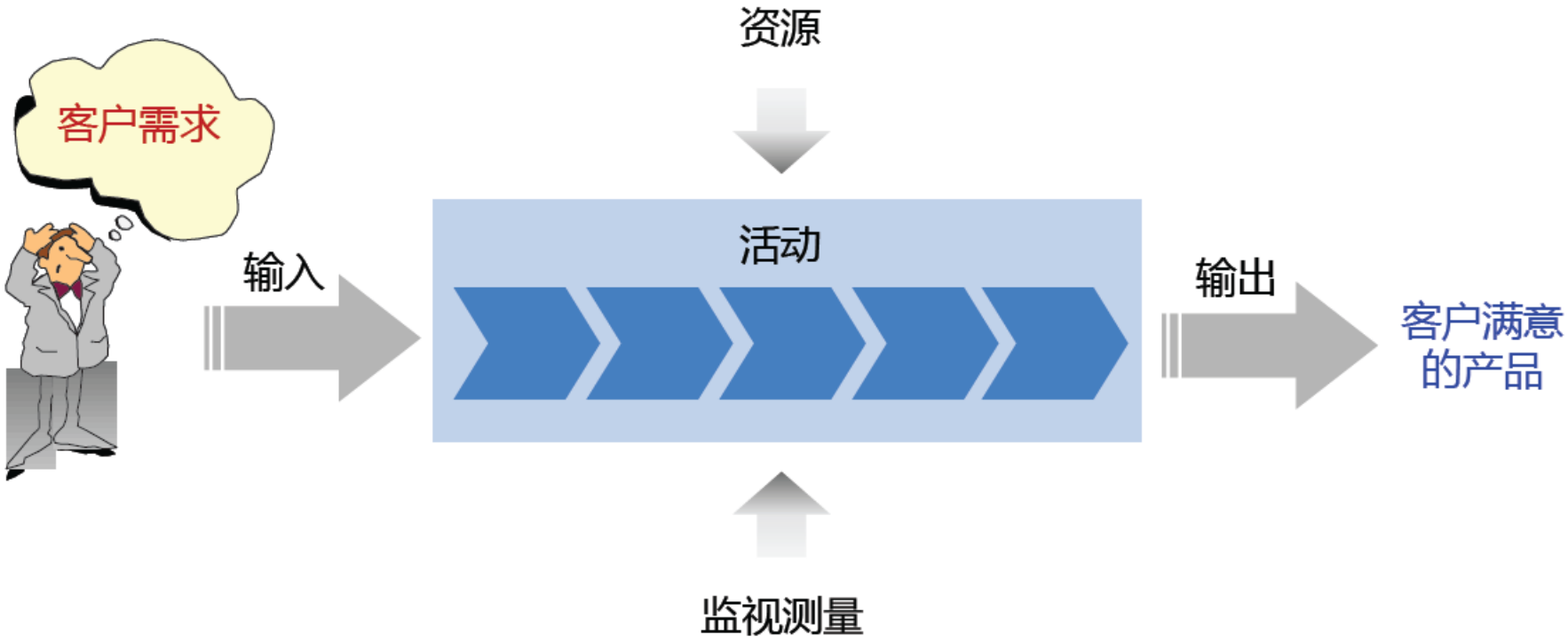
软件工程

软件工程的三个基本要素



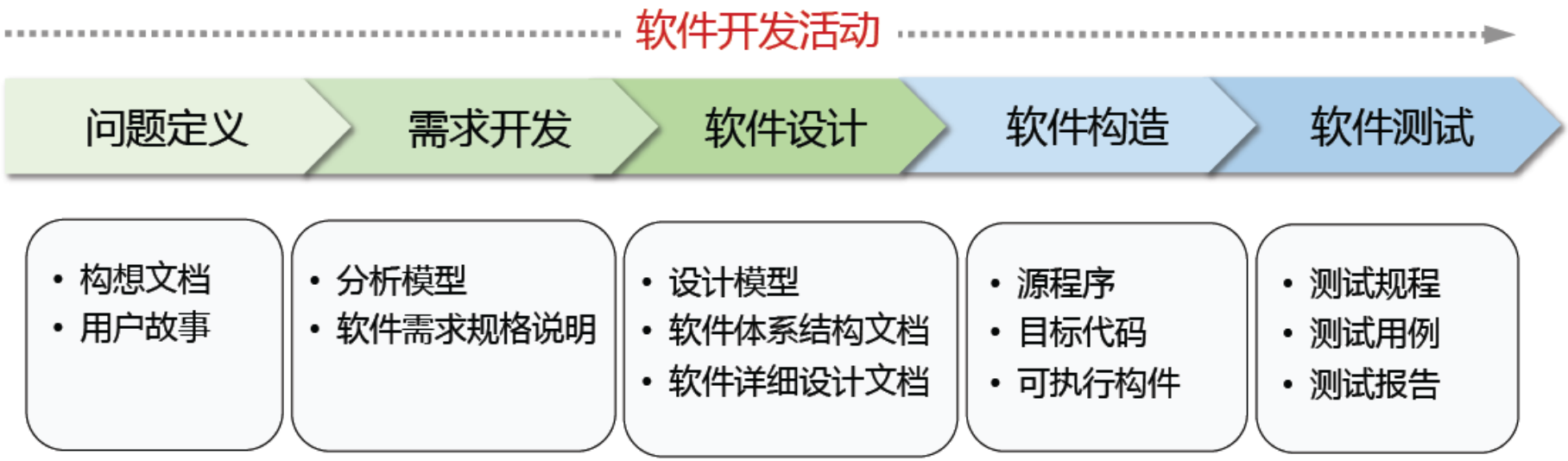
软件工程

软件工程的三个基本要素
(过程)



软件工程

软件工程的三个基本要素
(过程)

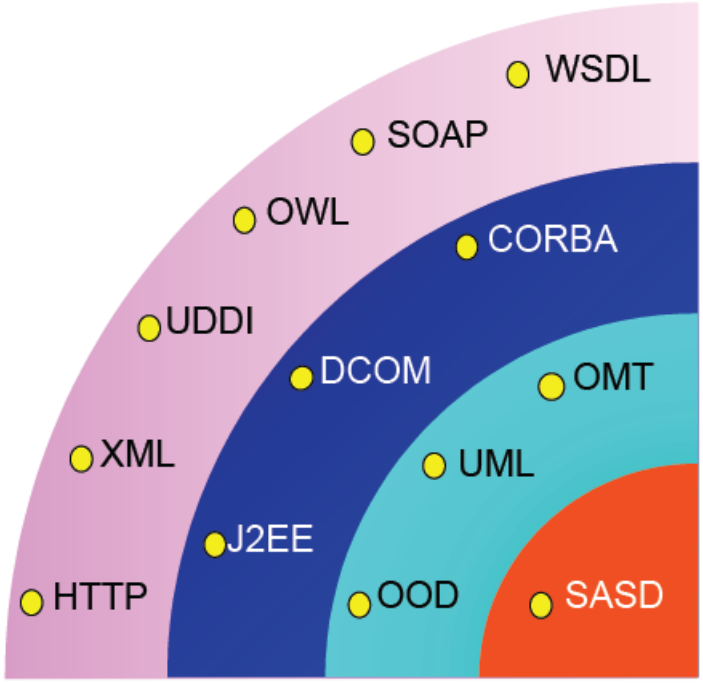


软件开发管理

(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

软件工程的三个基本要素
（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

工欲善其事
必先利其器

软件工程

软件工程的三个基本要素
（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
ENTERPRISE ARCHITECT Microsoft Visio MySQL Workbench	Notepad++ Visual Studio eclipse	Selenium xUnit BugFree Loadrunner Apache JMeter	Visual Studio eclipse ALTOVA umodel	Trello tower REDMINE git GitHub	

软件工程

软件开发的基本策略

软件复用	构造一个新系统不必从零做起，直接复用已有的构件进行组装即可
分而治之	将复杂的问题分解成若干个简单的问题，来源于人们的生活于工作经验
逐步演进	小步快跑，每走完一步调整并为下一步确定方向，直到终点
优化折中	优化软件的各个质量特性，同时通过协调各个质量特性，实现整体的最优

软件工程

软件开发的基本策略（软件复用）

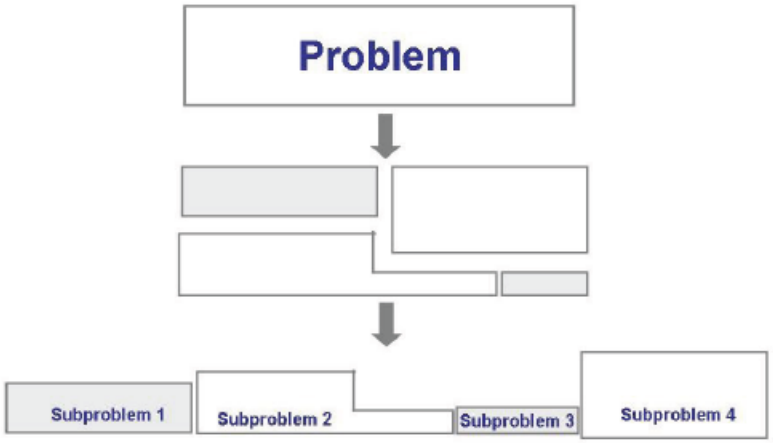
软件复用不仅仅是代码复用

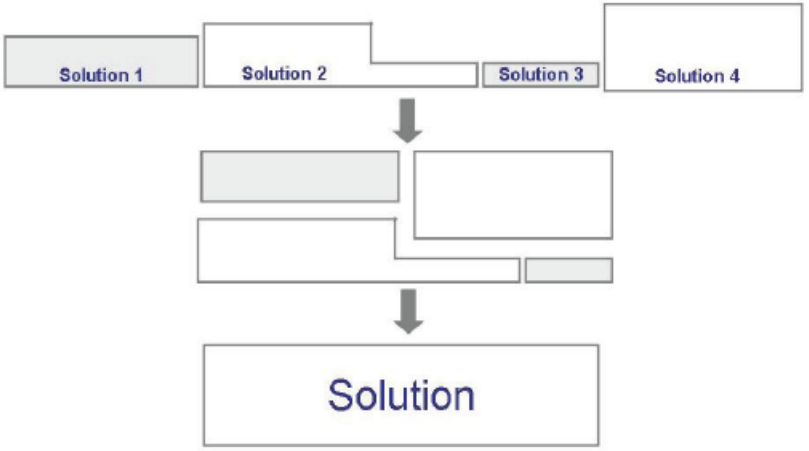
- 库函数、类库
- 模板（文档、网页等）
- 设计模式
- 组件
- 框架

软件工程

软件开发的基本策略（软件复用）

软件工程是一项解决问题的工程活动，通过分析问题，
将一个复杂问题分解若干小问题，然后再逐个解决





软件工程

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，通过不断进行迭代式增量开发，最终交付符合客户价值的产品

软件工程

软件开发的基本策略（优化折中）

优化是一种责任，但是优化是一个多目标的决策，
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，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

单选题 1分

设置

软件工程的基本目标是（ ）。

A 开发足够好的软件

B 消除软件固有的复杂性

C 努力发挥开发人员的创造性潜能

D 更好地维护正在使用的软件产品

提交

聆听　　总结

分享　　改进