软件设计与体系结构

注意事项

- 1课堂点名3次不到,取消考试资格!
- 2 平时成绩(30分):
 - (1) 问卷调查: 3分。 (2) 期中检查报告: 3分
 - (3) 系统验收: 3分。(4) 实验报告: 21分
- 3 实验: 第2, 4, 6, 8, 10, 12, 14, 16周 星期五3-4节课(双周)。

实验报告

基本需求

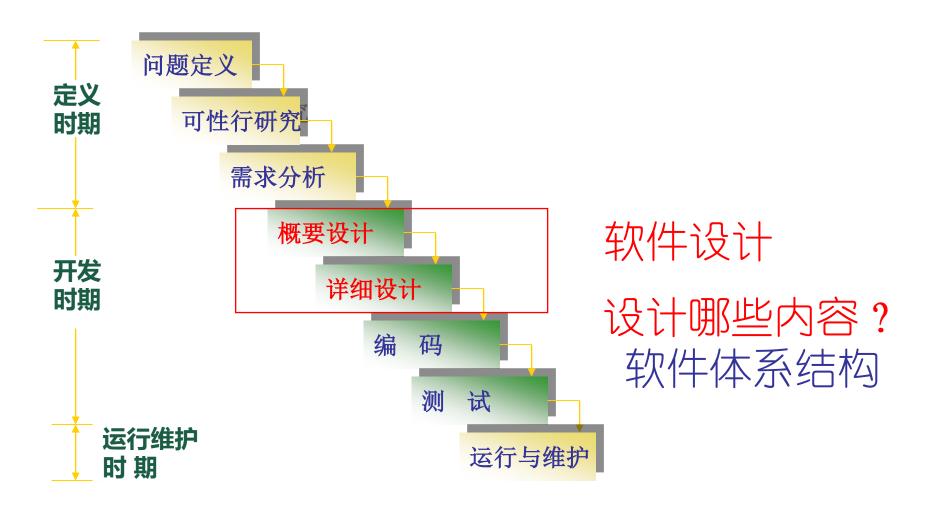
实现一个通用的用户权限系统

要求:

- (1) 语言: VC++(MFC), JavaEE, APP+JavaEE
- (2) 软件体系结构归档:用例文档、活动图文档、类图文档、顺序图文档、包图文档、部署文档。
 - (3) 《软件设计与体系结构-实验指导书(含封面)-2022.doc》实验指导书
 - (4) 一人一组

课程学习内容

软件生存周期模型 (瀑布模型 Waterfall Model)



课程名

面向软件体系结构的软件设计



教材:

1软件设计与体系结构 (第2版) 董威 文艳军 陈振邦 高等教育 出版社

参考教材:

- 1 《软件体系结构 原理、方法与实践》第2版 张友生著 清华大学出版社
- 2 软件设计和体系结构 秦航等编著 清华大学出版社
- 3 软件系统分析与体系结构设计 杨洋 刘全 东南大学出版社
- 4 软件工程导论 (第6版) 张海藩 清华大学出版社
- 5 《UML用户指南》 (第2版) 机械工业出版社]
- 6 软件体系结构 (第4版) 覃征等编著 清华大学出版社
- 7 《实战Hadoop大数据处理》 曾刚 清华大学出版社

课程章节安排

第1章 软件工程与软件设计:对应第1章、第3章 第2章 软件体系结构风格:对应第7章、第8章 第3章 23种java设计模式:对应第7章 第4章 软件体系结构描述语言:对应第2章

第5章 面向对象的软件分析与设计:对应第4章、第6章

课程学习目标:

- 1 掌握软件设计与软件体系结构概念
 - 1.1软件体系结构的风格 1.2设计模式 1.3框架
 - 1.4 软件体系结构描述语 (UML)
 - 1.5面向对象的软件分析与设计
- 2 掌握面向对象软件设计的基本思想
 - 2.1 软件设计指导思想:基于用例驱动的软件设计/建模。用例图、活动图、顺序图、类图、包图、部署图。
 - 2.2 应用MVC设计模式于分析与设计类的过程中。
- 3 掌握国家《计算机软件编制规范》GB-T8567-2006.doc 关于面向对象的软件系统建模规范文档SJ/T11291-2003《面向对象的软件系统建模规范第3部分:文档编制》
- 4 通过开发《用户权限系统》加深软件设计、软件体系结构以及《面向对象的软件系统建模规范第3部分:文档编制》的理解。

第1章软件工程与软件设计

内容提要

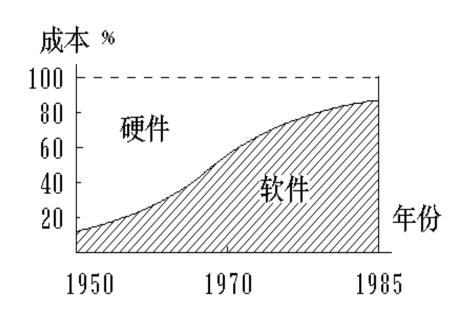
- 1.1 软件危机
- 1.2 软件工程
- 1.3 软件设计
- 1.4 软件体系结构
- 1.5 小结

1.1 软件危机

1.1.1 软件危机的表现

在计算机软件的开发和维护过程中所遇到的一系列严重问题。

- (1) 软件成本日益增长
- (2) 开发进度难以控制
- (3) 软件质量差
- (4) 软件维护困难



1.1.2软件危机的原因

- (1) 用户需求不明确
- (2) 缺乏正确的理论指导
- (3) 软件规模越来越大
- (4) 软件复杂度越来越高

1.1.3 如何克服软件危机

技术、管理

- 在技术上,应该采用基于重用的软件生产技术
- 在管理上,应该采用多维的工程管理模式



1.2 软件工程

1.2.1 软件工程定义

软件工程是指导计算机软件开发和维护的一门工程学科。采用工程的概念、原理、技术和方法来开发与维护软件,把经过时间考验并且证明正确的管理技术和当前最好的技术方法结合起来,以经济地开发出高质量的软件并有效地维护它,这就是软件工程。

软件工程: 技术+管理

1.2.2 软件工程方法学

- 软件工程方法学 (methodology, paradigm): 在软件 生命周期全过程中使用的一整套技术方法的集合。
- 软件工程方法学: 两大方法学

1、传统方法学(生命周期方法学或结构化范型)

• 它采用**结构化技术**(结构化分析、结构化设计和结构化实现)来完成软件开发的各项任务,并使用适当的软件工具或软件工程环境来支持结构化技术的运用。

2、面向对象方法学

 与传统方法相反,面向对象方法把数据和行为看成是同等 重要的,它是一种以数据为主线,把数据和对数据的操作 紧密地结合起来的方法。

> **软件设计**采用面向对象方法学指导: **软件体系结构**

1.2.2 软件工程方法学

- 软件工程方法学三要素: P5
 - ■方法
 - ■工具
 - ■过程

1.2.2 软件工程方法学

方法

工具

• 软件工程支撑环境.

过程

软件体系结构



软件工程方法学:工具

数据库: mysql, oracle,sql server.

软件开发环境: visual studio, eclipse

版本控制: vss(visual studio), svn(eclipse)

软件建模工具: rational rose

数据库建模工具: powerder design



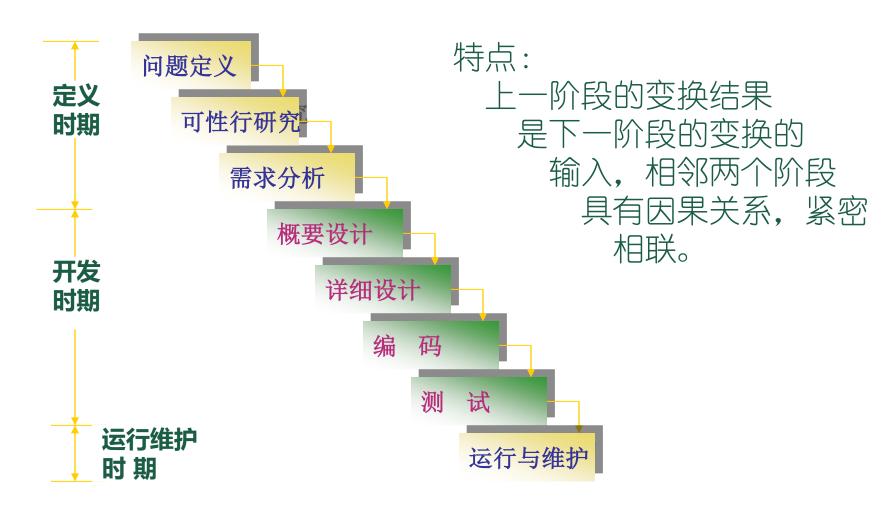
软件工程方法学:过程

包括多方面的任务,如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。

1.2.3 软件生命周期(SW life cycle)

- ■把软件从产生、发展到成熟、直至衰亡为止的全过程
- 软件生命周期划分为如下9个阶段(活动):
 - 问题定义
 - 可性行研究
 - 需求分析
 - 概要设计
 - 详细设计
 - 编码
 - 测 试
 - 运行
 - 维护

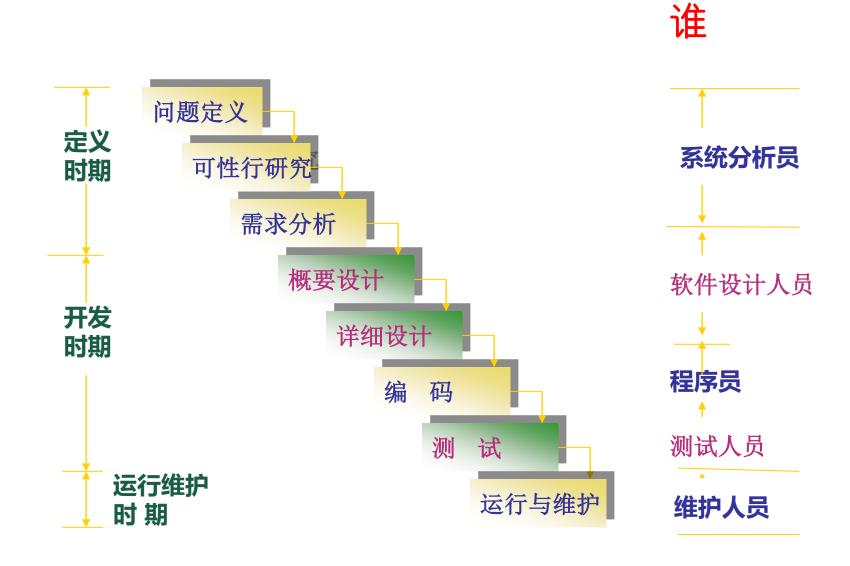




1.2.3 软件生命周期

阶段			关键问题	工作结果	文档
定义时期	软件计划	问题定义	是什么?	关于问题性质、工程 目标、规模的报告	计划任务书
		可行性分析	可行吗?	高层逻辑模型、 成本/效益分析	
	需求分析		做什么?	逻辑模型	需求规格说 明
开发时期	软件设计	概要设计	如何做?	求解方案、软件结构	设计说明书
		详细设计	具体怎样做?	编码规格说明	
	编码 (实现)		做!	源程序清单	程序清单
	测试	单元测试	可用吗?	单元测试方案、结果	测试报告
		集成测试		集成测试方案、结果	
运行维护 时期		维护	支持!	维护记录	维护记录







1.2.4 软件开发过程(软件过程)及 软件开发过程模型

软件开发过程(软件过程)是为了获得高质量软件 所需要完成的一系列任务的框架,它规定了**完成各项 任务的工作步骤**。

软件开发过程(软件过程)描述为了开发出客户需要的软件,什么人(who)、在什么时候(when)、做什么事(what)以及怎样(how)做这些事以实现某一个特定的具体目标。

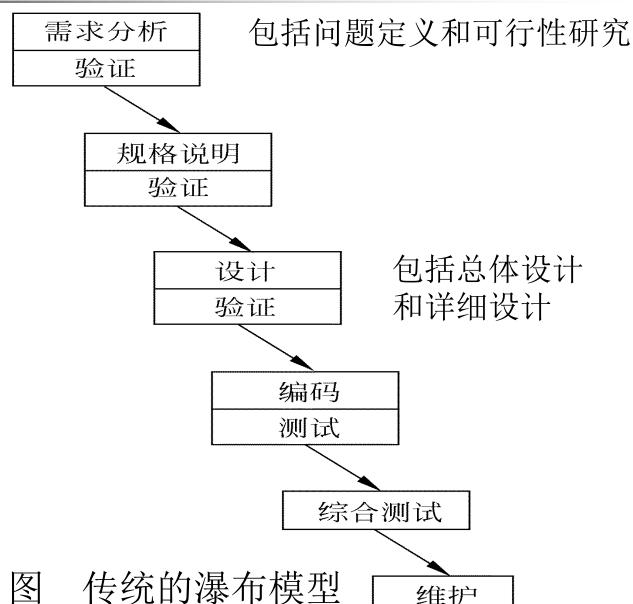


软件开发过程模型(软件过程模型、过程模型、 软件生命周期模型)

- 定义:描述软件开发过程中各阶段关系(软件生命周期各阶段的关系),是软件开发过程的概括。
- 作用
 - 为软件工程管理提供里程碑和进度表
 - 里程碑 (Milestone)
 - ◆ 软件生存期中各开发阶段末尾的特定点
 - 为开发过程提供原则和方法

1.2.4 软件开发过程(软件过程)及 软件开发过 程模型

瀑布模型

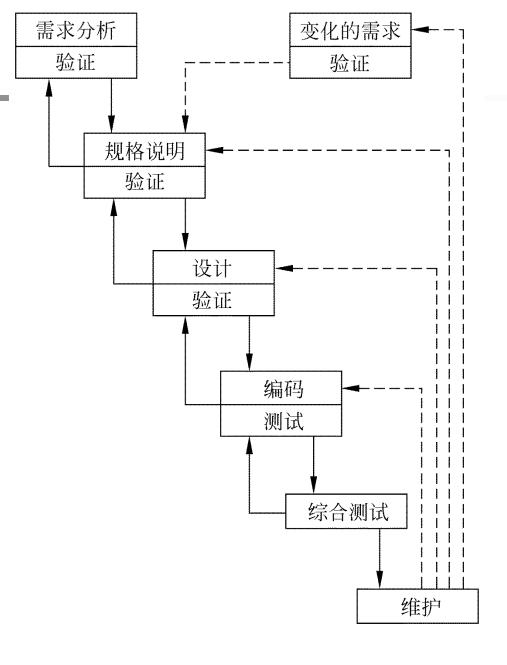


冬 传统的瀑布模型

传统的瀑布模型过于理想化了,事实上,人在工作过程中不可能不犯错误。实际的瀑布模型是带"反馈环"的,如图所示。

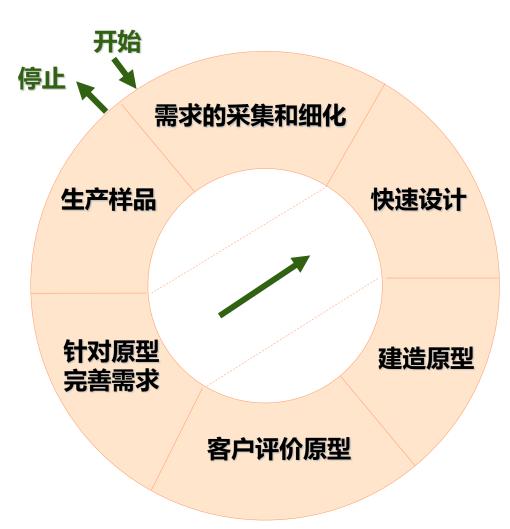


- 1、图中实线箭头表 示开发过程,虚线箭 头表示维护过程。、
- 2、实际的瀑布模型 当在后面阶段发现前 面阶段的错误时, 要沿图中左侧的反射 要沿图中左侧的段的 线返回前面阶段的产品 之后面阶段的任务。



(2) 快速原型模型(

Prototyping Model Aodel



- 1、原型系统仅包括未来系统的主要功能,以及系统的重要接□。
- 2、为了尽快向用户提供原型,开发原型系统时应型,开发原型系统时应尽量使用能缩短开发周期的语言和工具。

快速原型模型(Rapid Prototyping Model)

快速原型模型

优点: 不带反馈环, 基

本上是线性顺序进行。

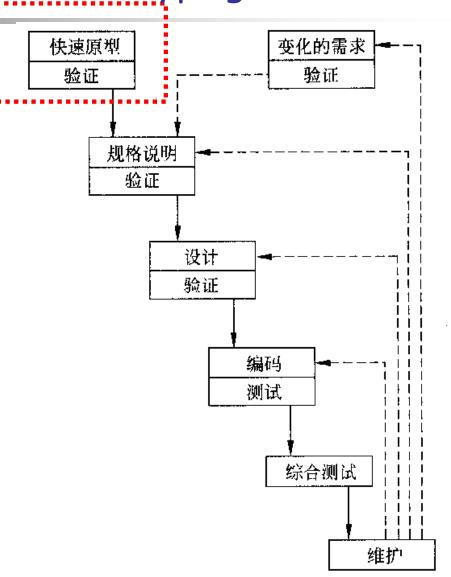
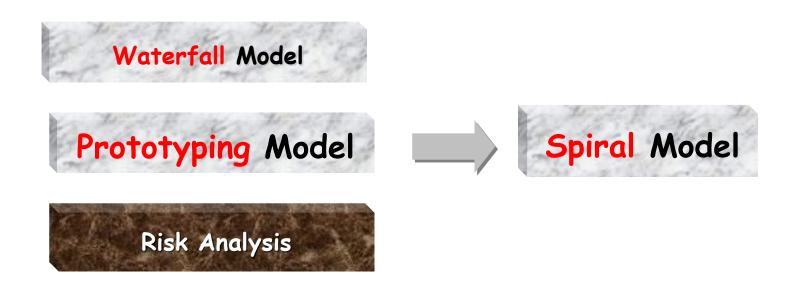
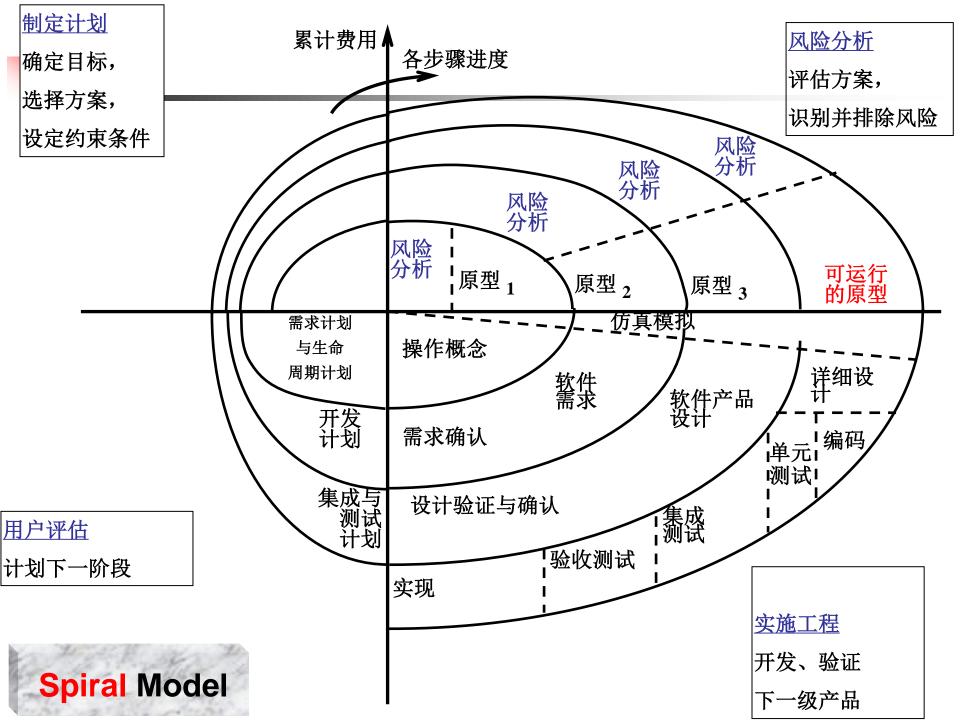


图 1.4 快速原型模型

- (3) 螺旋模型(Spiral Model)
- 1988,TRW公司BARRY W. Boehm提出
- 风险分析



螺旋模型由<u>制定计划、风险分析、实施工程、用户</u> 评估 四部分组成的迭代模型。



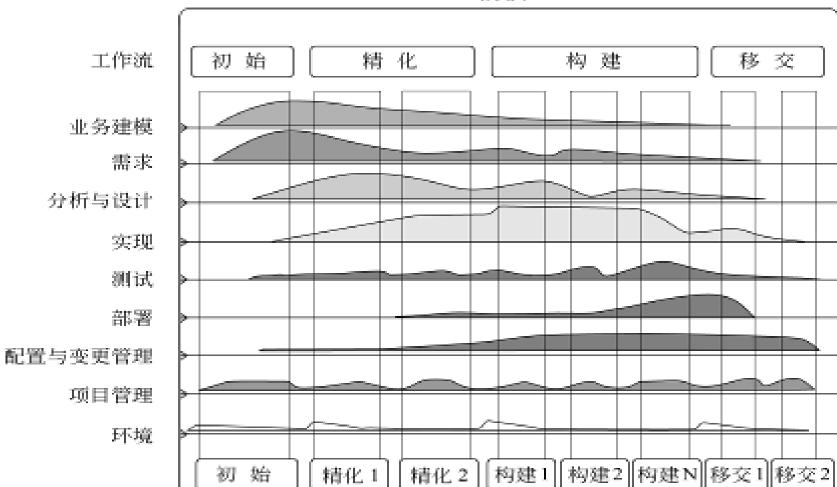
(4) Rational统一过程 (RUP)

- RUP总结了经过多年商业化验证的6条最有效的软件开发经验,这些经验被称为最佳实践:
 - ▶ 迭代式开发
 - ■管理需求
 - 使用基于构件的体系结构
 - ■可视化建模
 - 验证软件质量
 - 控制软件变更

阶段

迭代

移 交



■ RUP工作阶段

- 初始阶段:建立业务模型,定义最终产品视图, 并且确定项目的范围
- 精化阶段: 设计并确定系统的体系结构,制定项目计划,确定资源需求
- 构建阶段: 开发出所有构建和应用程序,把它们集成为客户需要的产品,并且详尽地测试所有功能。
- 移交阶段: 把开发出的产品提交给用户使用

■ RUP软件开发生命周期

- 核心工作流: RUP有9个核心工作流,前6个为核心过程工作流程,后3个位核心支持工作流程。
 - 业务建模
 - ■需求
 - 分析与设计
 - 实现
 - 测试
 - ■部署
 - 配置与变更管理
 - ■项目管理
 - 环境

(5) 敏捷软件开发 见华为敏捷软件开发.ppt



敏捷软件开发宣言

我们一直在实践中探寻更好的软件开发方法,身体力行的同时也帮助他人。由此我们建立了如下**价值观**

个体和互动 高于 流程和工具工作的软件 高于 详尽的文档客户合作 高于 合同谈判响应变化 高于 遵循计划

也就是说,尽管右项有其价值,我们更重视左项的价值。

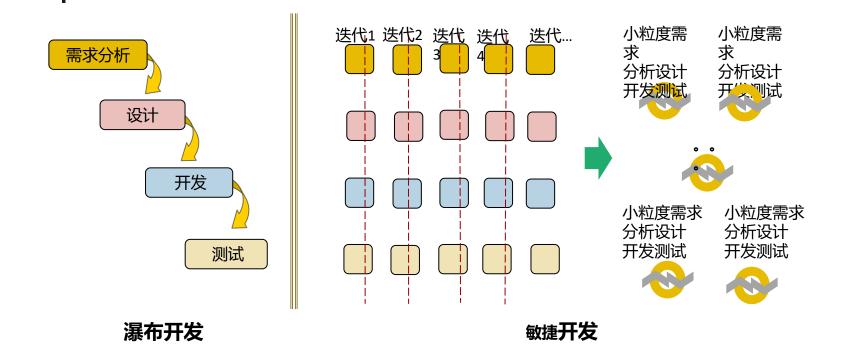
Kent Beck Mike Beedle Arie van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve

敏捷价值观 - 敏捷软件开发宣言 (2)

敏<mark>捷宣言遵</mark> 循的 原则

- 敏<mark>捷宣言遵</mark> 1,我们最重要的目标,是通过持续不断地及早交付有价值的软件使客户满意。
 - 2, 欣然面对需求变化,即使在开发后期也一样。为了客户的竞争优势,敏捷过程掌控变化。
 - 3, 经常地交付可工作的软件, 相隔几星期或一两个月, 倾向于采取较短的周期。
 - 4,业务人员和开发人员必须相互合作,项目中的每一天都不例外。
 - 5, 激发个体的斗志, 以他们为核心搭建项目。提供所需的环境和支援, 辅以信任, 从而达成目标。
 - 6,不论团队内外,传递信息效果最好效率也最高的方式是面对面的交谈。
 - 7, 可工作的软件是进度的首要度量标准。
 - 8, 敏捷过程倡导可持续开发。责任人、开发人员和用户要能够共同维持其步调稳定延续。
 - 9, 坚持不懈地追求技术卓越和良好设计, 敏捷能力由此增强。

敏捷开发与瀑布开发的外在区别



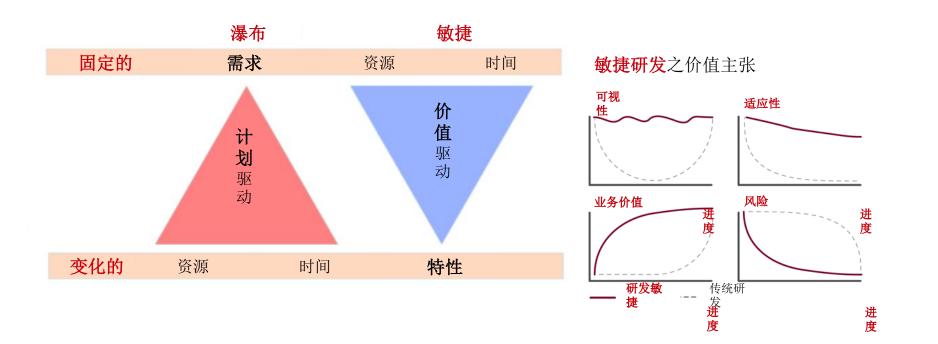


- · 软件更像一个活着的植物,软件开发是自底向上逐步有 序的生长过程,类似于植物自然生长
- 敏捷开发遵循软件客观规律,不断的进行迭代增量开发 ,最终交付符合客户价值的产品

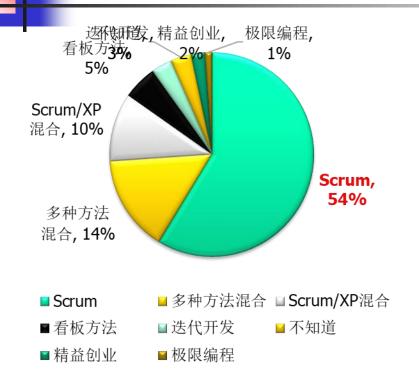


价值驱动 - 敏捷与传统瀑布型模式的最大区别

■ 敏捷基于这种方式,可以实现研发过程的持续高可 视性、高可适应性,更早且持续产出业务价值,更 早发现和解决风险。



敏捷常用的工程方法



根据2019年的第13届VersionOne版年度敏捷行业状态报告(如左图),以Scrum为基础的方法论(包括Scrum、Scrum/XP混合等)体系仍然居于主流地位,使用率最高。其他还有:看板方法、精益创业、极限编程等。

除此之外,还有**DSDM、FDD、RAD**等一些符合敏捷精神的小众方法论,以及一些规模化敏捷的方法论,在此未一一列出。

1.3 软件设计

软件设计: 对软件将如何开发出来的一种描述。



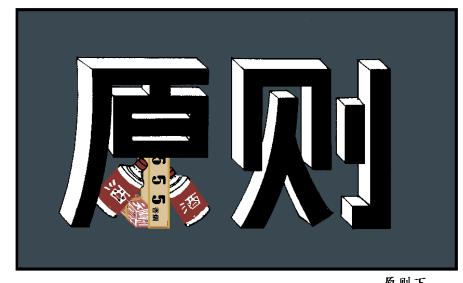
1.3.1 软件设计原则

抽象

体系结构 模式 模块化 信息隐蔽 功能独立 求精

重用

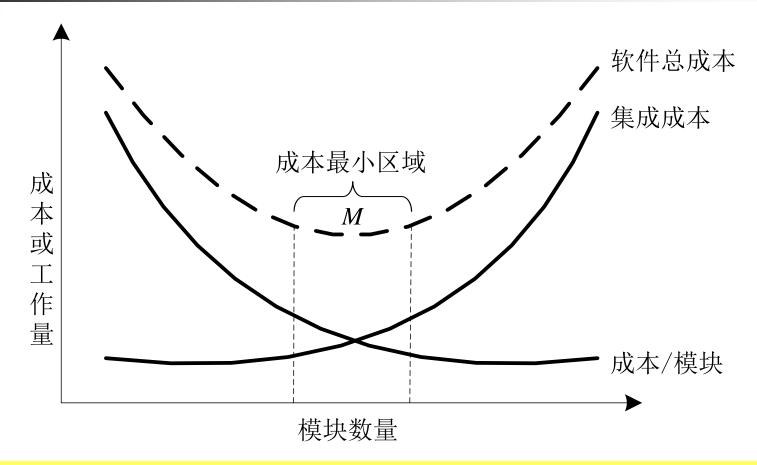
设计类



原则下...



模块化和软件成本



模块化时,位置需要保持在M附近,来避免过少、过多的模块化。 至于如何知道M的附近位置?如何将软件划分成模块?

7 +/- 2

1.3.2 设计过程和设计质量

在整个设计过程中,要使用一系列正式技术 评审,来评估设计的质量。标准如下:

- 设计必须实现所有分析模型中的明确需求,而 且满足客户期望的所有隐性需求。
- 对于生成代码的人、进行测试的人、维护软件的人,设计必须是可读的、可理解的指南。
- 设计必须提供软件的全貌,并从实现的角度说明数据域、功能域、行为域。



复习

软件工程方法学:

1、传统方法学

采用结构化技术 (结构化分析、结构化设计和结构化实现)来完成软件开发的各项任务。

基于数据流的软件需求分析与设计

2、面向对象方法学

采用面向对象技术(面向对象分析、面向对象设计和面向对象实现)来完成软件开发的各项任务。

基于软件体系结构的软件需求分析与设计

1.4 软件体系结构

■ 1.4.1 为什么要研究软件体系结构?

开发一个具有一定规模和复杂性的软件系统和编写一个简单的程序,是不一样的。借用《设计模式》的作者G. Booch的比喻,其中的区别如同建造一座大厦和搭建一个狗窝的差别。







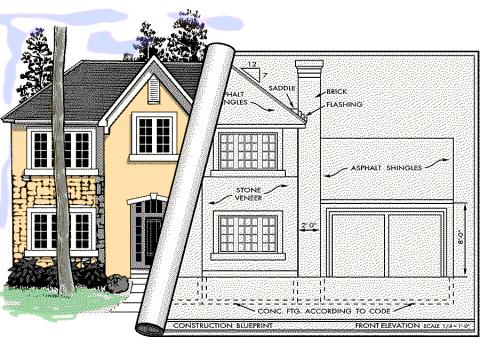
狗窝、 房子和 高楼



建筑师也充当建造者和客户之间的协调人!



房子的建模





1.4 软件体系结构

- 1.4.2 什么是软件体系结构
- 卡内基梅隆大学的软件工程研究所在网站上公开征集 软件体系结构的定义,至今已有百余种。

其中,较有影响力的定义包括:

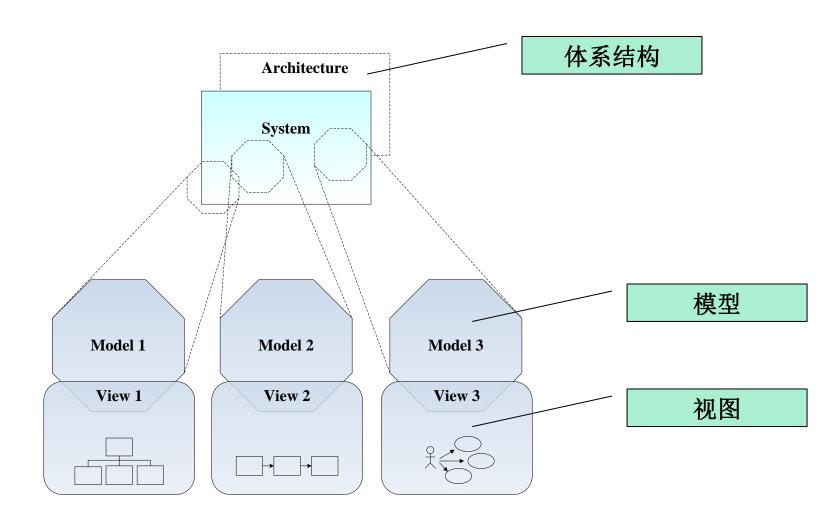
- 卡耐基梅隆大学的Garlan于1993年: 构件+连接件+ 约束
- 在2000年发布的IEEE Std1471-2000: 软件系统的基本组织,包含构件、构件之间、构件与环境之间的 关系,以及相关的设计与演化原则等。

1.4.3 几个概念

- 构件(Components组件)是功能单元,执行预定 义的服务并且与其他构件交互。
- 连接件(Connectors)定义交互协议与策略。 表示组件间的交互
- 约束(Constraints) 定义了系统必须服从的规则

- 构件(组件):三个主要流派
 - (1) OMG(Object Management Group 对象管理组织
-)的CORBA(Common Object Request Broker
- Architecture 通用对象请求代理结构)
 - (2) Oracle的JavaBean和EJB(Enterprise JavaBean)
- (3) Microsoft 的DCOM(Distributed Component Object Model分布式组件对象模型)





- (1) 软件体系结构实际上是不可见的
- (2) 软件体系结构是依附于系统存在的高层抽象
- (3) 需要对软件体系结构进行文档化: 软件体系结构描述形式化的描述语言: ADL(体系结构描述语言) C2, Unicon, Wright等非形式化的描述语言: UML
- (4) 针对特定的开发过程模型,合适的描述语言会有利于软件开发过程。

例如:在Rational统一过程(RUP)开发过程模型中,使用UML语言对软件体系结构进行描述。

(5) 该定义中没有涉及怎么建模,只是告诉我们应该对系统讲行建模。

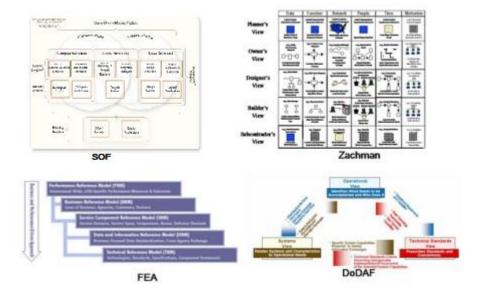


1.4.5 软件体系结构建模方法

怎么建模

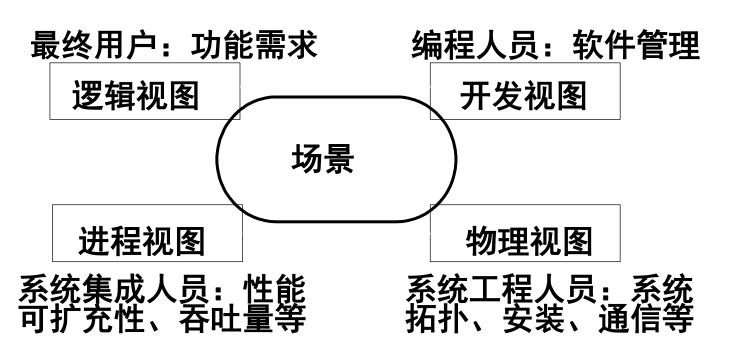
- **4+1**
- DoDAF
- MODAF
- TOGAF
- Zachman

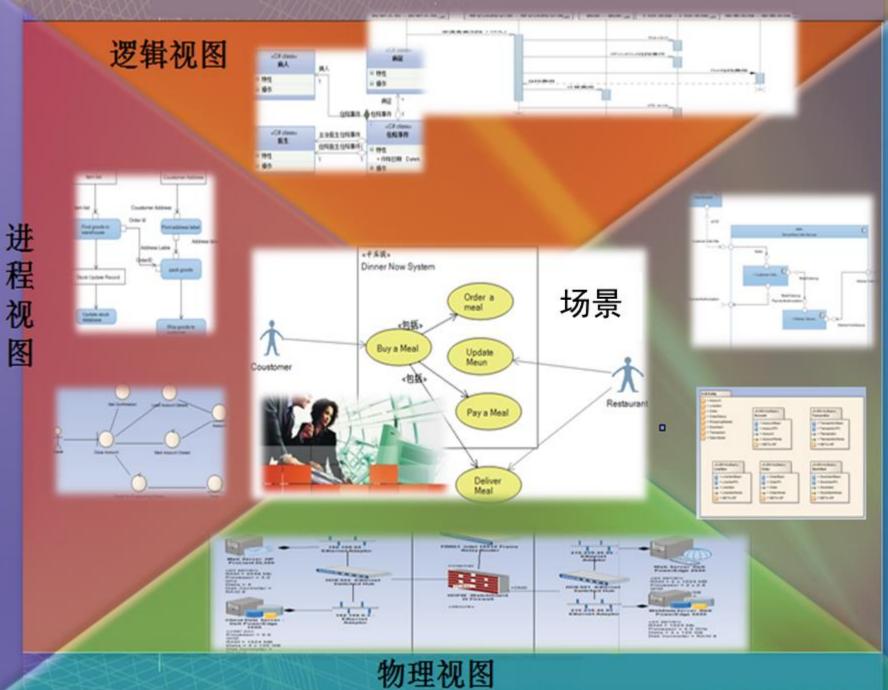
等等



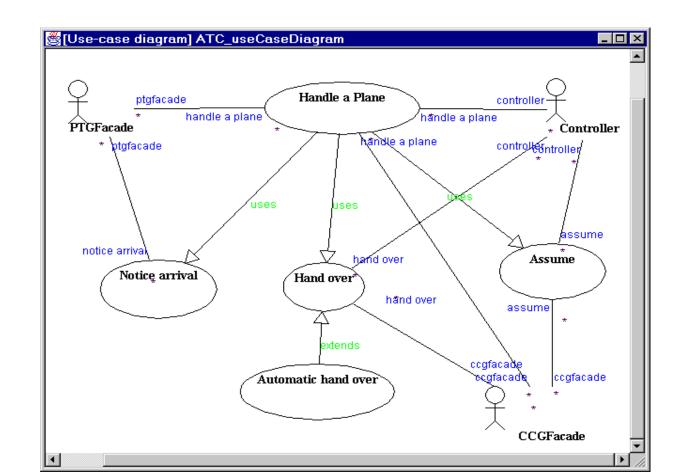
1.4.6 4+1模型

1995年,Philippe Kruchten在《IEEE Software》上发表了题为《The 4+1 View Model of Architecture》的论文,引起了业界的极大关注,并最终被RUP采纳。



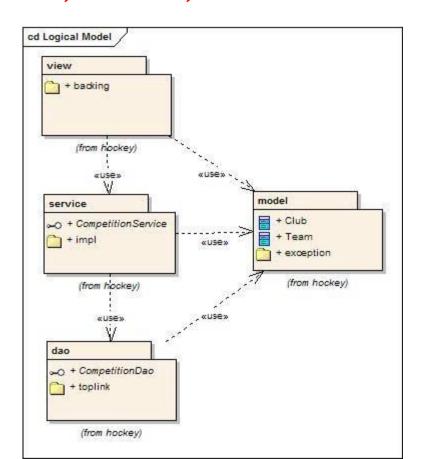


● 场景视图,关注最终用户需求,是**系统功能**的高层抽象,为整个技术架构的上线文环境.通常用UML用例图和活动图描述。

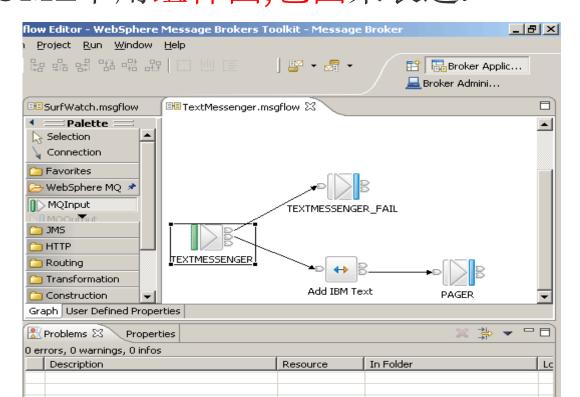




●逻辑视图(Logical view),主要是整个系统的结构及 实现方式的表述,关注系统提供最终用户的功能, 通 常在UML中用类图,通信图,时序图来表述。



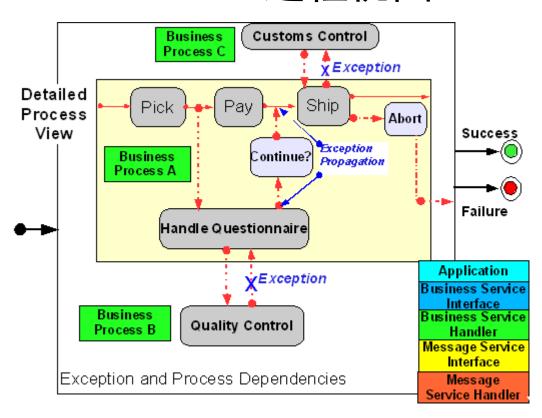
● 开发视图(Development View), 描述软件在开发环境下的静态组织, 开发视图关注程序包, 不仅包括要编写的源程序, 还包括可以直接使用的第三方SDK和现成框架、类库, 以及开发的系统将运行于其上的系统软件或中间件, 在UML中用组件图,包图来表述.





- 进程视图 (Process view): 关注系统动态运行时,主要 是进程以及相关的并发、同步、通信等问题。
 - 进程视图和开发视图的关系: 开发视图一般偏重程序包在编译时期的静态依赖关系, 而这些程序运行起来之后会表现为对象、线程、进程, 进程视图比较关注的正是这些运行时单元的交互问题.
 - ●在UML中通常用活动图表述。

4+1:进程视图



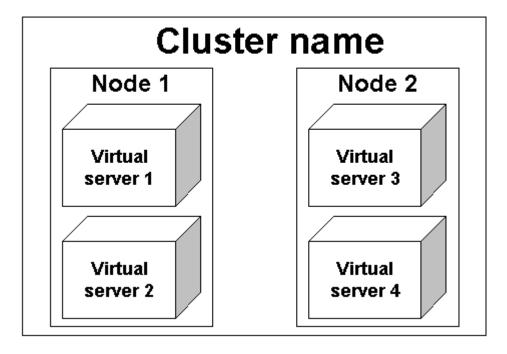


- ▶ 物理视图(Physical view): 关注软件的物理拓扑结构,以及**如何部署机器和网络**来配合软件系统的可靠性、可伸缩性等要求。
 - 物理视图和过程视图的关系:过程视图特别关注目标程序的动态执行情况,而物理视图重视目标程序的静态位置问题;物理视图是综合考虑软件系统和整个IT系统相互影响的架构视图。
 - 在UML中通常用部署图表述



4+1: 物理视图

Virtual servers (physical view)



Rational公司的Booch 从UML语言的包

Rational公司的Booch 从UML语言的角度出发给出了4+1模型的具体实现:

- (1) 用例视图(场景视图)
- (2) 设计视图 (逻辑视图)
- (3) 过程视图(进程视图)
- (4) 实现视图 (开发视图)
- (5) 部署视图 (物理视图)

1.5 小结

- 软件工程方法学三要素:
 - 方法: 传统方法学(结构化方法),面向对象方法
 - 工具: Rational Rose, PowerDesigner,Oracle...
 - 过程:软件过程(软件开发过程)
 - 软件开发过程中各阶段所执行的一系列活动,动作,任务的集合。
 - 阶段:问题定义可性行研究需求分析概要设计详细设计编码测试运行维护
 - 软件生命周期: 阶段
 - 软件开发过程模型(软件生命周期模型)
 - 描述软件开发过程中各阶段关系(软件生命周期 各阶段的关系)
 - 瀑布模型、快速原型模型、螺旋模型、Rational 统一过程(RUP)模型等
 - 软件设计: 概要设计 详细设计 2个阶段

1.5 小结(续)

- 软件设计: 概要设计 详细设计 2个阶段
 - 对软件将如何开发出来的一种描述
 - 软件设计包含一系列的转换过程。
 - 即把一种描述或者模型转换为另一种描述或者模型
 - 描述的结果可能是文字、图表,甚至数学符号、 公式等组成的文档或模型。
 - 内容:为了推动软件工程本科教育发展,ACM与IEEE/CS在2014年制定了计算教程软件工程卷(Computing Curriculum-Software Engineering, CCSE),其中关于软件设计的活动(内容)如下。(CCSE 2014)

- 软件设计内容(CCSE 2014):
 - 软件体系结构设计
 - 界面设计(人机交互设计)
 - 模块/子系统设计
 - 软件体系结构已经把软件划分为子系统
 - 此处指的是确定模块的具体接口定义并设计模块的内部结构,即设计包含其中的(更小粒度的)模块、构件和类,明确它们之间的协作关系,确保它们能够协同实现高层模块接口规定的所有功能和行为
 - 遵循"高内聚,低耦合"设计思想。
 - 过程/算法设计
 - 上述几种设计活动主要考虑软件体系结构及其模块 之间的控制分层关系,而不关心模块内各处理元素 和决策元素的顺序。
 - 过程/算法设计的任务就是对模块内部的执行过程 进行描述,给出有关处理的精确说明,例如事件 的顺序、选择、循环等操作以及数据的组成等。
 - 面向过程:流程图;面向对象:UML的活动图。
 - 数据模型设计:数据结构设计、数据库设计
- 软件设计一般先设计软件体系结构,然后再逐步精化进行更 详细的设计,一直到设计可以被实现的程度。



- I 软件体系结构
 - 定义: 构件+连接件+约束
 - 软件体系结构建模方法
 - **■** 4+1
 - 软件体系结构描述
 - 形式化描述语言: ADL
 - 非形式化描述语言: UML
 - 软件体系结构风格
 - 应用框架: SpringMVC, Struts等
 - 软件设计模式

后续章节安排

- 软件体系结构
 - 软件体系结构风格
 - 应用框架: SpringMVC, Struts等(选讲)
 - 软件设计模式
 - 软件体系结构描述语言(UML)
- 软件设计
 - 界面设计
 - 面向数据流的软件设计(选讲)
 - 面向对象的软件设计
 - 基于用例驱动的软件开发过程
 - MVC设计模式用来寻找类。
 - 用例图、活动图、类图、顺序图、包图、部 署图
 - 数据设计(数据库原理课程讲授)



thanks