

软件工程

第2讲 软件工程过程模型

贾西平

Email: jiexp@126.com

1

课程主要内容

面向过程的软件工程

可行性研究

需求分析

结构化软件设计

软件编码

软件测试

面向对象的软件工程

面向对象概述

面向对象分析

面向对象设计

动态建模

面向对象测试

软件工程项目管理

软件度量

项目计划

风险管理

质量保证

2

内容提纲

- 过程
- 软件工程过程
- 软件工程过程模型
 - 瀑布模型
 - 原型模型
 - RAD模型
 - 演化模型
 - 敏捷开发
- 软件重用
- 08:42 • 框架

3 of 43

3

过程

- 看视频，思考下面问题(12')：
 - 过程含义
 - 过程管理的作用
 - 什么是过程方法
 - 软件过程包括哪些活动
- <https://www.bilibili.com/video/av88249030?p=20>

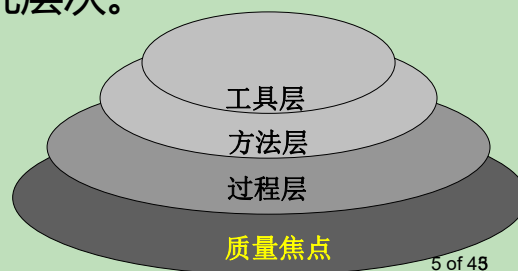
08:42

4 of 43

4

软件工程过程

- **软件工程过程**是用以开发或维护软件及其相关产品的一系列活动，包括**软件工程活动**和**软件管理活动**。
- 软件工程以**质量焦点**为基础，分**过程**、**方法**和**工具**三个研究层次。



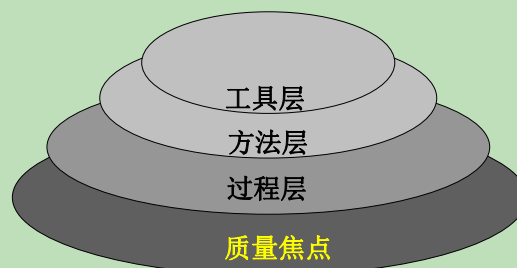
08:42

5 of 48

5

软件工程过程—质量焦点

- 软件工程必须以有组织的软件质量保证为基础，对**质量**的关注构成了软件工程的根基。



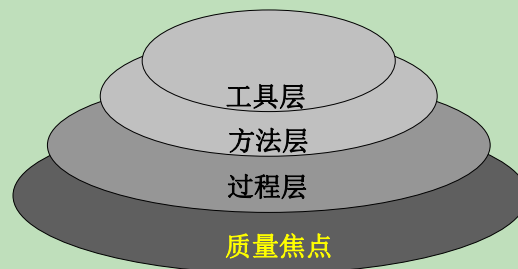
08:42

6 of 43

6

软件工程过程—过程层

- 软件工程**过程**指出了如何把各种方法和工具进行综合才能使软件开发合理、及时的进行。
- 定义了一组关键过程域(KPAs)，作为对软件工程项目进行管理与控制的基础。



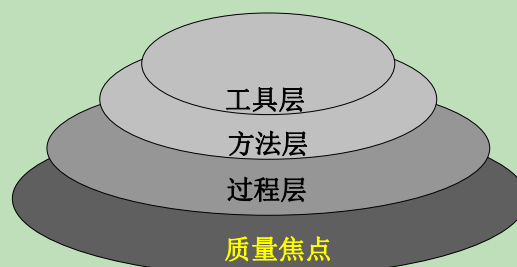
08:42

7 of 43

7

软件工程过程—方法层

- 软件工程**方法**涵盖了需求分析、设计、编程、测试、维护等各个环节，它指明了完成这些任务在技术上应当“如何做”。



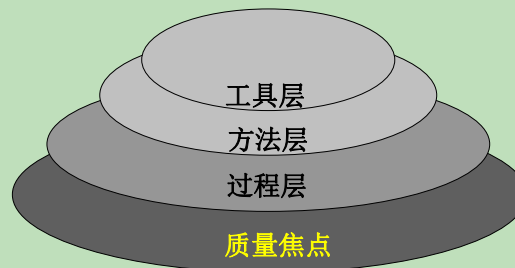
08:42

8 of 43

8

软件工程过程—工具层

- **工具层**对过程和方法提供支持，使得工程活动、管理活动得以自动、半自动的进行。



08:42

9 of 43

9

软件工程过程

- **软件工程过程**包括**四种基本过程活动**
 - **软件规格说明**：规定软件的功能，性能及其运行限制。
 - **软件开发**：产生满足规格说明的软件，包括设计，编码等工作。
 - **软件确认**：确认软件能满足客户提出的要求，对应于软件测试。
 - **软件演进**：为满足客户变更要求，软件必须在使用过程中演进，以求尽量延长软件的生命周期。

08:42

10 of 43

10

软件工程过程特点

一个好的软件工程过程应当具备如下特点：

- (1) **易理解性**。
- (2) **可见性**：每个过程活动都以得到明确的结果而告终，保证过程的进展对外可见。
- (3) **可支持性**：容易得到CASE工具的支持。
- (4) **可接受性**：比较容易被软件工程师接受和使用。
- (5) **可靠性**：不会出现过程错误，或者出现的过程错误能够在产品出错之前被发现。
- (6) **健壮性**：不受意外发生问题的干扰。
- (7) **可维护性**：过程可以根据开发组织需求的改变而改进。
- (8) **高效率**：从给出软件规格说明起，就能够较快地完成开发而交付使用。

08:42

11 of 43

11

软件过程定义模版

项目	目的
1. 目标	该过程的目的是什么？
2. 所有者	谁是负责该过程？谁负责文档、交流、维护和持续改进此过程？
3. 输入	该过程的输入是什么？这些输入来自何处？这些输入有什么约束和依赖？
4. 输出	该过程的输出是什么？这些输出去向如何？这些输出有什么约束和依赖？
5. 入口准则	该过程的启动要求是什么？
6. 出口准则	该过程的结束要求是什么？
7. 任务	实现该过程目标需要什么任务？
8. 依赖 / 约束	该过程任务或步骤中有什么依赖或约束？
9. 确认	该过程的度量标准是什么？如何知道任务是否达到预期？如何知道目标是否满足？

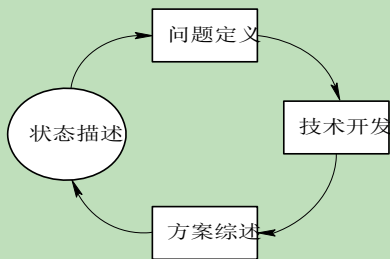
08:42

12 of 43

12

循环解决问题过程

- 宏观上，所有的软件开发过程都可看成是一个循环解决问题的过程。
- 包括四个阶段
 - **状态描述**表示了事物的当前状态；
 - **问题定义**标识了要解决的特定问题；
 - **技术开发**通过应用某些技术来解决问题；
 - **方案综述**提交解决结果(如文档、程序、数据、新产品等)给一开始就需要方案的人



08:42

13 of 43

13

软件工程过程模型

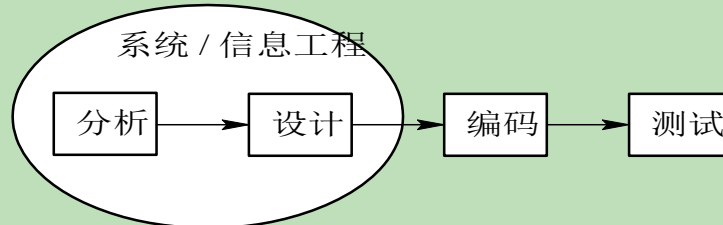
- **软件工程过程模型**：由软件工程师在实际工程活动中设计、提炼出的**工程开发策略**，用以覆盖软件过程中的基本阶段，确定所涉及的过程、方法、工具。
- **典型过程模型**
 - 线性顺序模型（瀑布模型）
 - 原型模型
 - 快速应用开发模型
 - 增量模型
 - 螺旋模型
 - 敏捷开发
 -

08:42

14 of 43

14

瀑布模型



- 1970年W.Royce提出瀑布模型

- **特征**

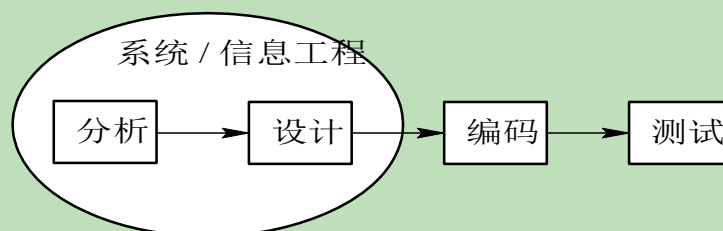
- 接受上一阶段的结果作为本阶段的输入
- 利用这一输入实施本阶段应完成的活动
- 对本阶段的工作进行评审
- 将本阶段的结果作为输出，传递给下一阶段
- 适用于需求确定，变更少的项目

08:42

15 of 43

15

瀑布模型



- **缺点**

- 缺乏灵活性，难以适应需求不明确或需求经常变化的软件开发
- 开发早期存在的问题往往要到交付使用时才发现，维护代价大

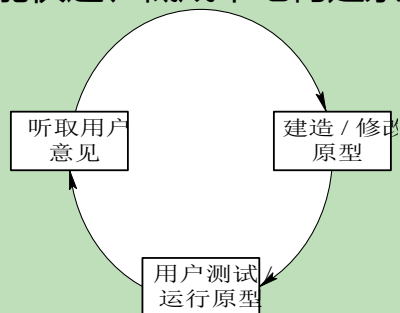
08:42

16 of 43

16

原型模型

- **原型** (prototype) 是预期系统的一个可执行版本，它反映了系统性质（如功能、计算结果等）的一个选定的子集。
 - 原型不必满足目标软件的所有约束
 - 目的是能快速、低成本地构建系统的一个**示范版本**

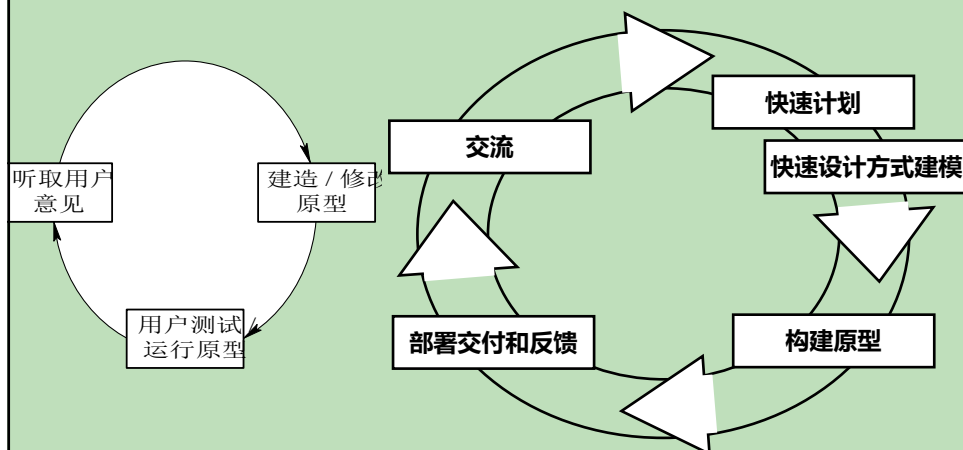


08:42

17

17

原型模型



08:42

18 of 43

18

原型类型

- **探索型 (exploratory prototyping)**
 - **目的**：弄清目标系统的要求，确定所希望的特性，并探讨多种方案的可行性
- **实验型 (experimental prototyping)**
 - **目的**：验证方案或算法的合理性，在大规模开发和实现前，用于考核方案是否合适，规格说明是否可靠。

08:42

19 of 43

19

原型的使用策略

- **废弃 (throw away) 策略**
 - 主要用于定义需求
 - 原型通常被废弃，然后根据探索或实验的结果用良好的结构和设计思想重新设计目标系统。
- 原型可作为单独的过程模型使用，也常被作为一种方法或实现技术应用于其它过程模型中。

08:42

20 of 43

20

原型模型特点

◆特点

- 通过迭代的过程快速建立与用户需求一致的原型
- 软件产品开发过程基本是线性顺序进行的

◆优点

- 减少不明确需求所带来的开发风险

◆缺点

- 开发人员需要掌握建立快速原型的技术和工具

08:42

21 of 43

21

原型模型使用范围

• 适用范围

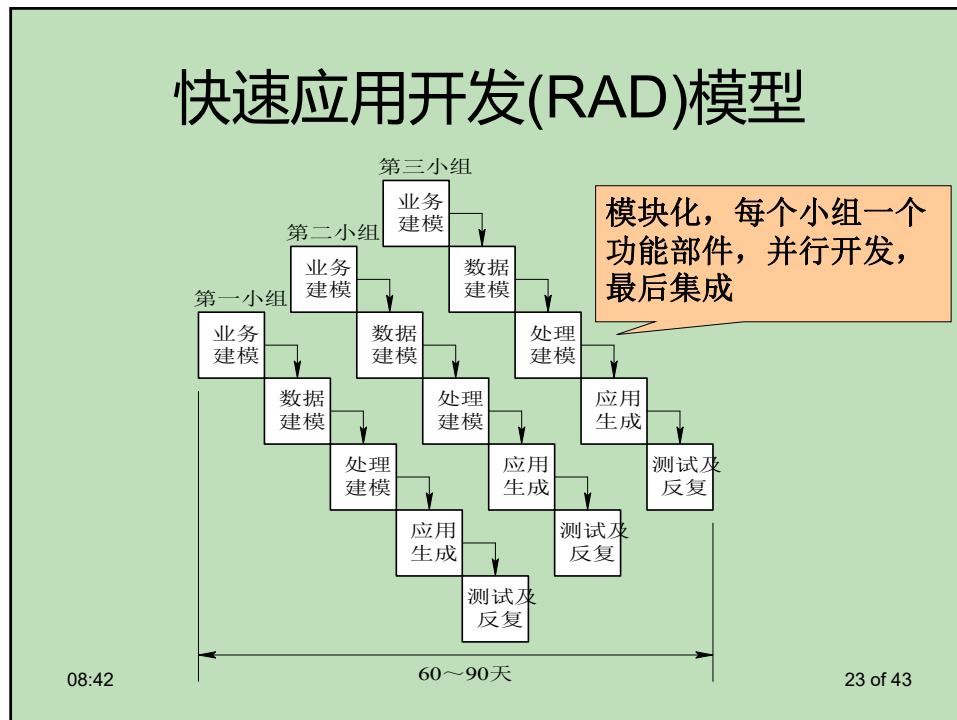
- 小型或者中等规模交互软件
- 大型系统的部分开发，例如：用户界面
- 生命周期较短的软件项目

08:42

22 of 43

22

快速应用开发(RAD)模型



23

RAD模型的开发阶段

- 业务建模
 - 业务活动中的信息流被模型化
 - 说明什么信息驱动业务流程、生成什么信息、谁负责生成该信息、该信息流向何处、谁处理它等。
- 数据建模
 - 标识出每个数据对象的特征属性
 - 定义这些对象之间的关系。
- 处理建模
 - 创建处理描述以便增加、修改、删除或获取某个数据对象。

08:42

24 of 43

24

RAD模型的开发阶段

- 应用生成
 - 复用已有的程序构件或是创建可复用的构件
 - 使用自动化工具辅助软件建造。
- 测试及反复
 - 测试所有新创建的构件、所有的接口

08:42

25 of 43

25

RAD模型特点

- 特点
 - 支持多小组并行工作
 - 强调程序构件复用
 - 开发周期短
- 不足
 - 对构件复用依赖程度高
 - 不适用技术风险很高，要采用较多新技术的项目

08:42

26 of 43

26

快速应用开发(RAD)模型

- 是线性顺序模型的一个“高速”变种，强调极端的开发周期。
- 通过使用基于构件的建造方法达到快速开发的效果。
- 适用情况
 - 需求得到很好理解
 - 项目的范围约束明晰
- 主要用于信息应用软件的开发

08:42

27 of 43

27

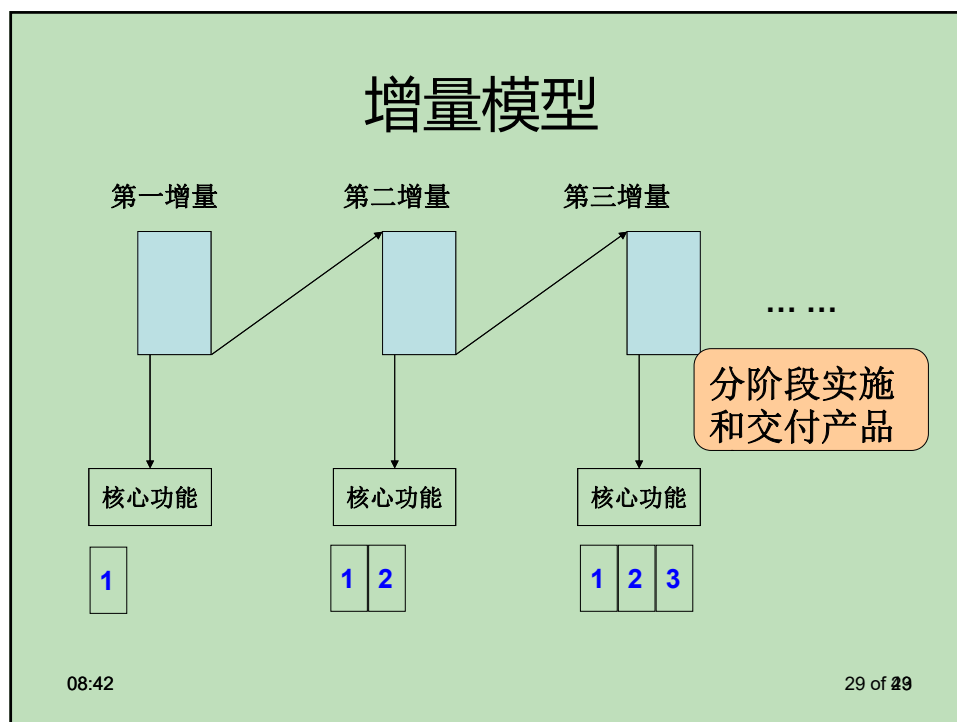
演化软件过程模型

- **基本思想**
分期完成，分步提交（先提交有限功能版本，再完善）
- **特点**：利用“迭代”方法，渐进开发，生产出逐步完善的软件产品
- **适用于对软件需求缺乏准确认识的情况。**
- **比较**
 - 瀑布模型：线性序列开发完成后能够交付一个完整系统
 - 原型模型：用于引导用户需求，验证算法，一般不交付最终产品
- **分类**
 - 增量模型
 - 螺旋模型

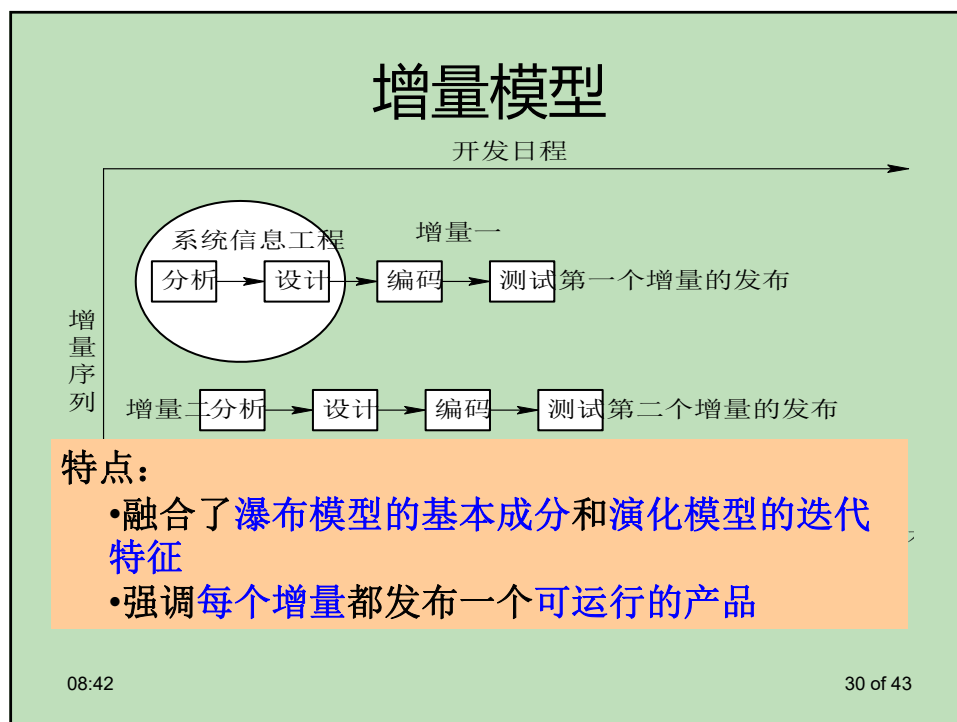
08:42

28 of 28

28



29



30

增量模型

• 优点

- 在较短时间内向用户提交部分工作的产品；
- 使用户可以看到所开发软件的中间版本；
- 早期增量作为原型为后期增量开发提供需求；
- 重要功能被首先交付，可以获得最多的测试；
- 能有计划地管理技术风险

• 缺点

- 需要软件设计开放式的体系结构
- 增量的有效集成比较困难
- 容易退化为边学边做，使软件过程的控制失去整体性

08:42

31 of 43

31

增量模型

• 适用情况

- 需求经常变化
- 市场急需而开发人员和资金不能在设定的市场期限之前实现一个完善的产品

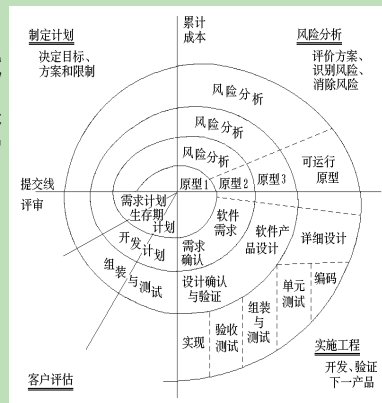
08:42

32 of 43

32

螺旋模型

- 瀑布模型和演化模型的结合，增加了**风险分析**
- 螺旋模型沿着螺线旋转，在四个象限上分别表达四个方面的活动，即：
 - 制定计划
 - 风险分析
 - 工程实施
 - 客户评估



08:42

33 of 43

33

螺旋模型

- **制定计划**：确定软件目标，选定实施方案，弄清项目开发的限制条件
- **风险分析**：评价所选的方案，识别风险，消除风险
- **工程实施**：实施软件开发，验证工作产品
- **客户评估**：评价开发工作，提出修正建议

08:42

34 of 43

34

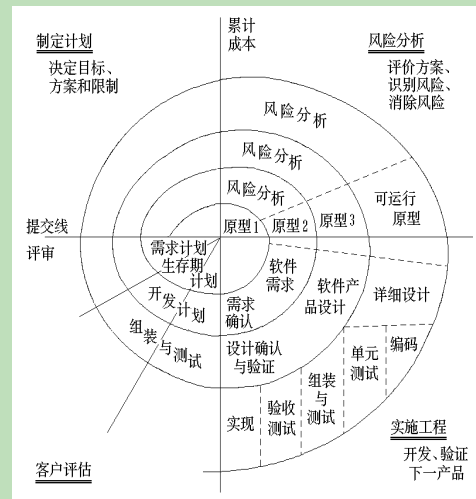
螺旋模型

• 优点

- 将质量目标放在首位
- 关注早期错误消除
- 风险驱动型

• 缺点

- 需要事先制订过程模型和发布产品
- 需要风险评估的经验



08:42

35 of 43

35

敏捷开发过程

- <https://www.bilibili.com/video/av88249030?p=22> (14')

08:42

36 of 43

36

微软开发过程

- <https://www.bilibili.com/video/av88249030?p=23> (8')
- <https://www.bilibili.com/video/av88249030?p=24> (17')
- <https://www.bilibili.com/video/av88249030?p=25> (15')

08:42

37 of 43

37

软件重用

- “重用” 是提升软件财富价值的有效途径。
- 重用包括**知识重用**、**方法重用**、**软件成分重用**三个层次
 - **知识重用**:使我们能够高效率的开发、维护一个又一个的软件项目；
 - **方法重用**:帮助我们有章可循的解决各类工程问题；
 - **软件成分重用**:使我们在需求分析、系统设计和编码事项的过程中能够事半功倍

08:42

38 of 43

38

软件成分重用

- 软件成分重用又可以分为**分析结果重用**、**设计结果重用**和**代码重用**三个层次。
- 在大部或局部雷同的项目中重复地使用已经被前驱项目证明是正确的部分**分析结果**
- **设计重用**：
 - 在开发类同项目的软件，尤其是在软件移植过程中能够极大地减少工作量，提高工作效率。
 - 设计结果重用包括体系结构设计重用和详细设计重用两重内涵。
- **代码重用**
 - 最直接的重用
 - 包括基于“宏”的重用、基于函数库的重用和基于继承的重用三种不同的方法

08:42

39 of 43

39

框架

- 框架是一个**可复用的设计构件**，它规定了应用的体系结构，阐明了整个设计、协作构件之间的依赖关系、责任分配和控制流程，**表现为**一组抽象类以及其实例之间协作的方法，它为构件复用提供了上/下文(Context)关系。
- 框架的最大好处就是重用
- 框架能重用代码，使构件间的通信趋于简单。

08:42

40 of 43

40

框架的好处

- 框架的最大好处就是重用
- 重用分析
- 重用设计
- 重用代码

08:42

41 of 43

41

主要内容回顾

- 软件工程过程
- 软件工程过程模型
 - 瀑布模型
 - 原型模型
 - RAD模型
 - 演化模型
 - 敏捷开发
- 软件重用
- 框架技术

08:42

42 of 43

42

思考题

- 瀑布模型的主要特点是什么？
- 原型模型的主要特点是什么？
- RAD模型的主要特点是什么？
- 增量模型的主要特点是什么？
- 螺旋模型的主要特点是什么？
- 重用的三个层次？
- 软件成分重用的三个层次？

08:42

43 of 43

43

谢谢大家！
感谢清华大学刘强老师的视频资源！

14:33

44