

第2章 软件工程与需求工程

2.1 软件工程

- ▶ 软件危机的解决方法
 - 应用工程化的方法进行软件的开发和维护
- ▶ 软件工程的研究内容
 - 软件开发过程、软件开发和维护的方法和技术、软件开发和维护工具系统、质量评价和质量保证、软件管理和软件开发环境等。

4

第2章 软件工程与需求工程

- 2.1 软件工程
- 2.2 软件开发过程模型
- 2.3 需求工程与软件开发
- 2.4 软件需求的开发和管理过程

2

2.2 软件开发过程模型

1. 瀑布式模型
2. 快速原型模型
3. 渐增式模型
4. 螺旋式模型
5. 面向对象的开发模型
6. 敏捷开发模式简介

5

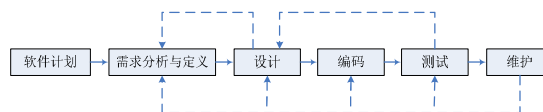
2.1 软件工程

- ▶ 软件危机
 - 是指人们难以控制软件的开发和维护。
- ▶ 表现
 - (1) 大型软件系统十分复杂，很难理解和维护；
 - (2) 软件开发周期过长；
 - (3) 大型软件系统的可靠性差；
 - (4) 软件费用往往超出预算。

3

2.2.1 瀑布式模型

依据软件生命期而提出的软件开发模型，将软件的开发过程被分为六个阶段，每个阶段都有明确的分工和任务，并产生一定的书面结果。各阶段之间是紧密相关的，后一阶段的工作是依据前一阶段的工作结果而开展的。



6

2.2.1 瀑布式模型

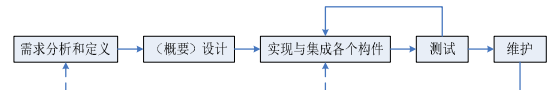
不足

- （1）要求用户一开始就提出**清晰完整的需求**；
- （2）阶段间**移交信息**（文档）的过程中，由于个人的理解不同，容易产生误解；
- （3）用户的**参与程度不够**。

7

2.2.3 渐增式模型

渐增式模型的基本思想是从核心功能开始，通过不断地改进和扩充，使得软件系统能**适应用户需求**的变动和扩充，从而获得柔性较高的软件系统。



渐增式模型表明，必须在实现各个构件之前就**全部完成需求分析和概要设计工作**。

10

2.2.2 快速原型模型

快速原型模型的基本思想是快速建立一个实现了若干功能的（不要求完全）可运行模型来启发、揭示和**不断完善用户需求**，直到满足用户的全部需求为止。



8

2.2.3 渐增式模型

特点

- （1）能在短时间向用户提交可完成部分功能的产品。
- （2）能逐步增强产品功能以使用户有较充裕的时间学习和适应新的软件系统。

不足

- （1）在把每个新增的构件或功能集成到现有的软件系统中时，必须不破坏该软件系统。
- （2）在设计软件系统的体系结构时，要充分考虑到其开放性，而且加入新构件的过程必须简单和方便。

11

2.2.2 快速原型模型

目的

- （1）
- （2）
- （3）

特点

- （1）
- （2）
- （3）

不足

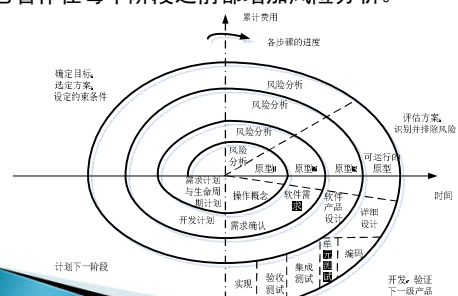
- （1）
- （2）快速原型系统对于软件系统的开发环境要求较多，在一定程度上也影响了其使用的范围和实用价值。



9

2.2.4 螺旋式模型

将瀑布式模型与快速原型模型结合到一起，并加上风险分析。理解这种模型的一个简便方法是把它看作在每个阶段之前都增加风险分析。



12

2.2.4 螺旋式模型

特点

- （1）适用于软件开发机构内部开发大规模软件项目。
- （2）对于可选方案和约束条件的强调有利于已有软件的重用，也有助于把软件质量作为软件开发的一个重要目标。
- （3）减少过多测试或测试不足所带来的风险。

局限性

- （1）风险分析执行的困难使螺旋模型仅适用于大规模软件项目

13

2.2.5 面向对象的开发模型

特点

（1）有一部分分析工作必须在设计之前进行，而另外一些分析工作则需与其他部分的设计与实现工作并行地进行，因而呈现出非线性工作方式。

（2）软件系统的表达形式在整个开发模型中都是相同的，即面向对象方法中把类及其结构作为系统的表达单元，无论哪一个阶段都以渐增的方式不断地进化或细化这些表达单元。

（3）开发模型支持软件的重用。

16

风险分析简介

风险辨识

- 哪些风险
- 风险的主要因素
- 风险所造成的后果

风险评估与主要风险因子识别

风险来源分析

风险应对策略

- 风险回避策略
- 风险转移策略
- 风险分散策略
- 风险承担策略

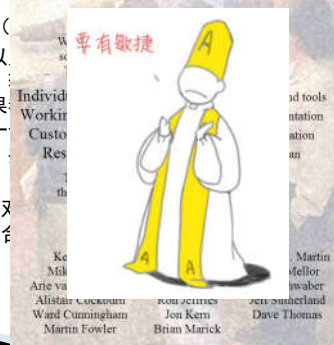
14

敏捷开发 Manifesto for Agile Software Development

敏捷开发（

- 是一种以开发中，目的成果就是把一小项目，态

- 敏捷是绝对效沟通和

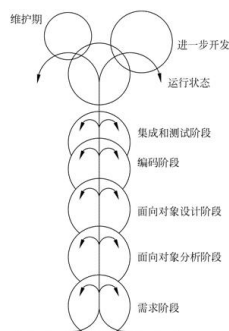


去。在敏捷各个子项E。简言之，独立运行的二可使用状

化，以高

2.2.5 面向对象的开发模型

所谓面向对象就是应用对象、类、继承、封装、消息、对象或类之间的关系等面向对象的概念对问题进行分析和求解的软件开发技术，或者说，是以对象（类）为数据中心、对象之间的动态行为模式作为运行机制的一种问题求解方法。



15

敏捷开发简介

目的:

- 降低需求变化的成本

提倡:

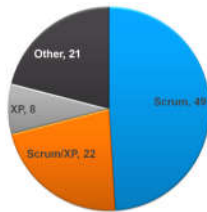
- 互动交流、反馈、简单、勇气、团队

核心:

- 小规模，频繁的版本发布，短迭代周期
 - 测试驱动开发Test-driven development
 - 结对编程Pair programming
 - 持续集成Continuous integration
 - 每日站立会议Daily stand-up meeting
 - 同拥有代码Collative code ownership.
 - 系统隐喻System metaphor

敏捷开发简介

- ▶ 极限开发方法
 - 极限编程 (XP)
 - Scrum
 - 精益开发 (Lean Development)
 - 动态系统开发方法 (DSDM)
 -
- ▶ Scrum创始人
 - Jeff Sutherland



19

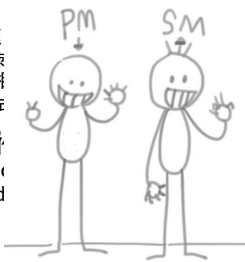
与传统开发模式对比

- ▶ 对比瀑布式开发
 - 两者没有很多的共同点
 - 瀑布模型式是最典型的预见性的方法，严格遵循预先计划的需求、分析、设计、编码、测试的步骤顺序进行。
 - 相对来讲，敏捷方法则在几周或者几个月的时间内完成相对较小的功能，强调的是能将尽早将尽量小的可用的功能交付使用，并在整个项目周期中持续改善和增强。
- ▶ 对比迭代方法
 - 相比迭代式开发两者都强调在较短的开发周期提交软件，敏捷开发的周期可能更短，并且更加强调队伍中的高度协作。
 - 适应性的方法集中在快速适应现实的变化
 - 当项目的需求起了变化，团队应该迅速适应

22

敏捷开发简介

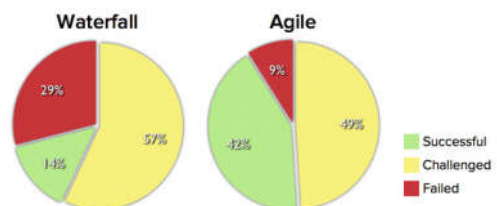
- ▶ 相关概念
 - agile: 迅速
 - sprint: 冲刺
 - scrum: 英式橄榄球
 - product backboard: 团队是球员，目标进行工作
 - story board: “待开发”的区块
 - burn down: 时间进展



反馈结果
次冲刺，一个
：大家一拥而上，扣，围绕着产品
表
化窗口，一般有
“待发布”几
过燃尽图获知时

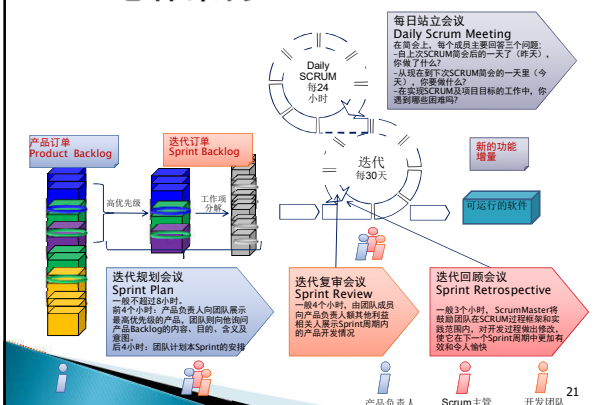
20

Agile Succeeds Three Times More Often Than Waterfall



23

Scrum总体架构



21

2.3 需求工程与软件开发

1. 需求工程对软件开发的影响
2. 需求工程面临的困难

24

2.3.1 需求工程对软件开发的影响

- ▶ 需求工程对软件开发的影响如下：
 - （1）需求是制定项目计划的基础
 - （2）需求工程所产生的最终产物——需求规格说明——是软件设计和软件实现的基础
 - （3）需求规格说明也是测试工作和用户验收软件系统的依据
 - （4）需求规格说明也是软件维护工作的依据

25

2.3.2 需求工程面临的困难

- ▶ 需求分析
 - 需求分析是业务分析
 - 需求分析是一种分解活动
 - 需求分析是一种提炼与整合活动
 - 需求分析是一种规格化活动
- ▶ 存在的问题
 - 需求间的冲突和矛盾的检查以及解决；
 - 需求是否完整的确定；
 - 合适的需求建模的方法和技术。

28

2.3.2 需求工程面临的困难

需求工程是人们通过不断地认识和深入研究而形成的结果。而且需求工程对软件开发的影响是很大的。随着软件系统日益大型和复杂化，软件需求的开发和管理也日益复杂，而且需求工程自身也面临诸多有待解决地问题。

26

2.3.2 需求工程面临的困难

- ▶ 需求描述语言和规范化的困难性
 - 怎样规范化用户需求；
 - 规范化哪些用户需求；
 - 非形式化和形式化描述语言的使用。

29

2.3.2 需求工程面临的困难

- ▶ 需求获取（需求捕获）中存在的问题：
 - 捕获范围不足
 - 缺乏计划性
 - 缺乏科学性
 - 捕获对象不明确
 - 捕获手段不足

27

2.3.2 需求工程面临的困难

- ▶ 需求验证的困难性
 - 需求规格说明正确性的确认和验证；
 - 验证的方法和技术；
 - 如何进行自动验证。

30

2.3.2 需求工程面临的困难

- ▶ 需求管理的困难性
 - 需求规格说明书的质量保证；
 - 需求规格说明书的版本管理；
 - 需求变更的控制。

31

需求人员所需要的技能

技能类型	培养要点	说明
业务能力	类比	例如，在非销售企业也能找到“产、销、供”的线索
	宏观思考	过于陷入细节就会影响宏观理解
技术能力	溯源	分析技术的发展历史，可以更好地了解其作用
	优/缺点	了解优缺点就能够在正确的地方应用正确的技术
沟通能力	思维模式	通过不断改变和训练可以提高思维模式

32