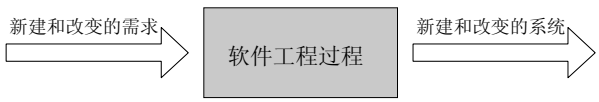
火龙果●整理
uml.org.cn

软件流程实施方案选择

什么是软件工程过程？

- 任何工业化生产都离不开过程
  - 大规模、有组织、有序的
- 软件工程过程
  - 合理使用软件生产制造技术，定义谁(Who)什么时候(When)做什么(What)，以及如何(How)达到确定目标
  - 将用户需求转化为软件系统的所有活动的集合
    - 开发一个新的产品
    - 增强已开发产品的功能(修复bug，或者添加功能)



火龙果●整理
uml.org.cn

什么是有效的过程?

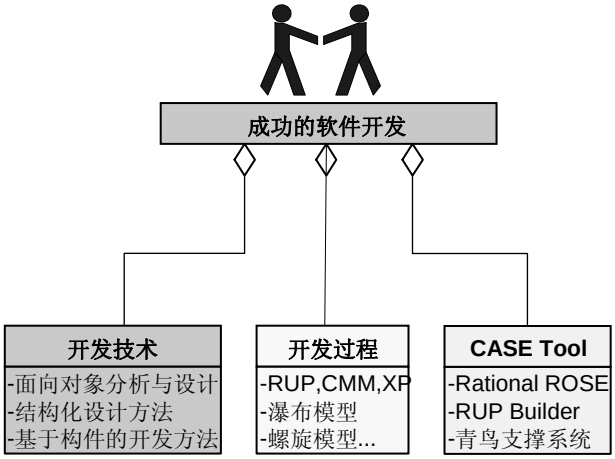
- 保证产品质量
- 迅速减少项目风险(需求、技术、政治风险等)
- 保证项目的可预测性(进度、成本、功能等)
- 能够捕捉和提供最佳实践方法(Best Practice)
- 促进领域内(软件开发)的共识和相互理解

议题

- RUP、EUP与XP
- 微软MSF与MOF
- Agile与CMMI

火龙果●整理
uml.org.cn

成功的软件开发



开发方法=建模语言+开发过程

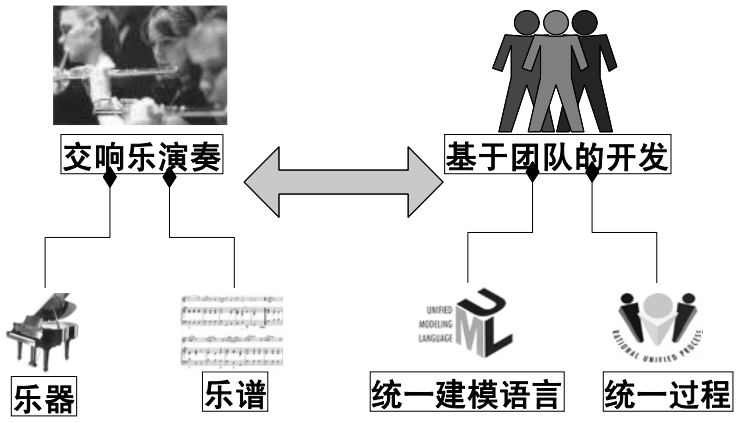
- 建模语言 (Modeling Language)
  - 表述设计方法的表示法
  - 主要是图形化的
- 开发过程 (Development Process)
  - 对开发中所采取步骤的指导
  - 将建模语言置于特定语境(开发环境)之中

火龙果●整理
uml.org.cn

广义的开发过程

- 虽然我们这里说过程(Process)，实际上它涵盖了软件项目管理技术的各个方面，包括：
  - 软件度量
  - 项目估算
  - 进度控制
  - 人员组织
  - 配置管理
  - 项目计划等

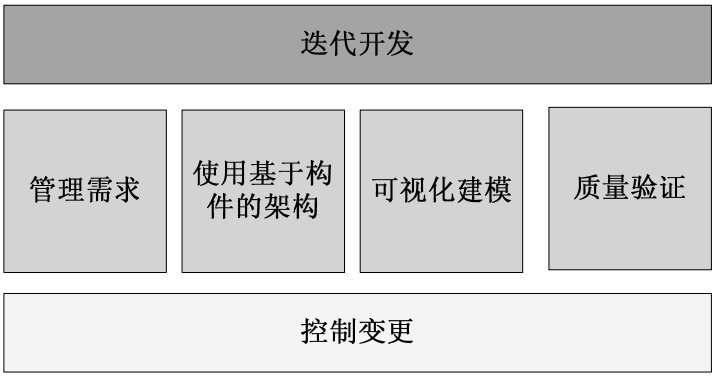
UML+RUP=最佳软件开发方法



火龙果●整理
uml.org.cn

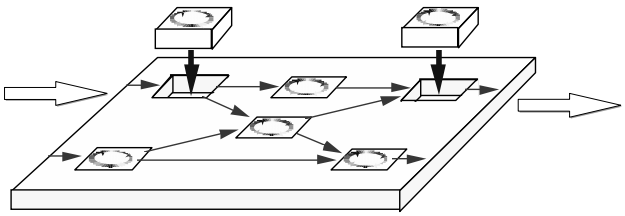
RUP囊括了6项最佳实践方法

被证明可解决软件开发过程中的根本问题



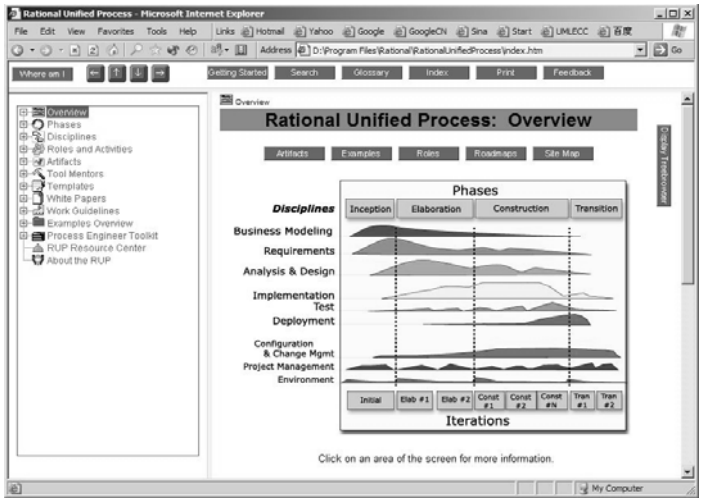
RUP是一个通用过程框架

- RUP是可定制的(Customizable)通用过程框架
  - 开发软件的种类
  - 开发软件的规模
  - 开发软件的应用领域
  - 开发团队的组织形式

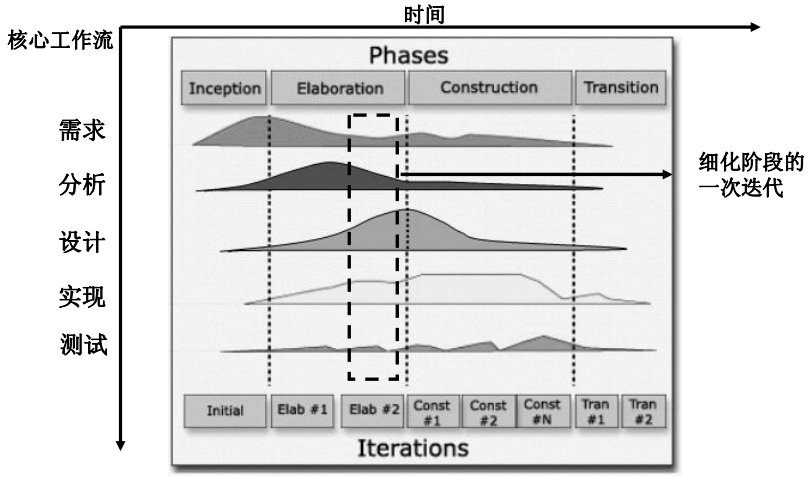


火龙果●整理
uml.org.cn

RUP Version 2002.05.00



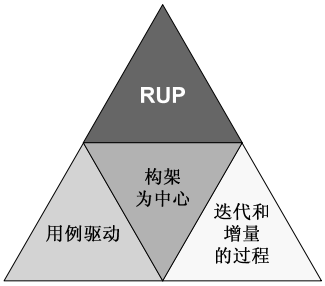
统一过程中的迭代



火龙果●整理
uml.org.cn

RUP的3大核心技术特点

- 用例(Use Case)/需求驱动的
- 以构架(Architecture)为中心的
- 迭代(Iterative)和增量的(Incremental) 开发方式



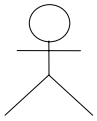
用例驱动的**RUP**

火龙果●整理
uml.org.cn

什么是用例(Use Case)

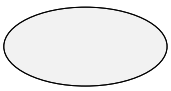
- 软件系统的特征
  - 服务于特定用户
- 用例描述用户(User)需求
  - 用户：与所开发系统交互的人或其他系统
  - 所有用例构成系统的全部功能和边界
- UML规范说明中的定义
  - 用例：在不展现系统或子系统内部结构的情况下，对系统或子系统的某个连贯的功能单元的定义和描述。

Use Case 基本模型元素符号



Actor

- Actor(参与者)
  - 表示与系统交互的任何事物



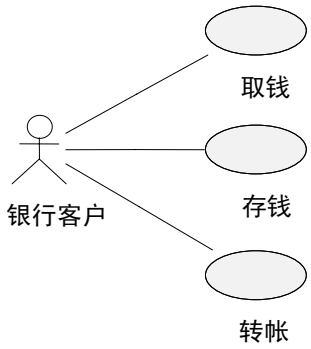
Use Case

- Use Case(用例)
  - 定义了系统的一系列动作，并向参与者提供有价值的结果

火龙果●整理
uml.org.cn

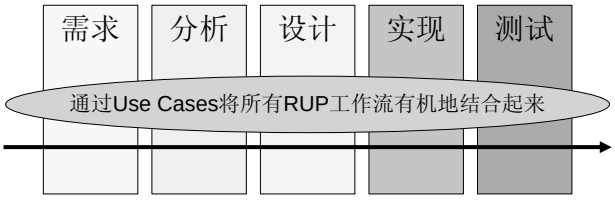
Use Case 举例

银行个人自动交易系统的Use Case图



RUP是用例驱动的

- 需求分析阶段用户需求用Use Case来表达
- 设计初期很多类根据Use Case来发现
- 构造阶段开发的管理和任务的分配按照Use Case来组织
- 测试阶段的实例根据Use Case来生成



火龙果●整理
uml.org.cn

用例在RUP中的其他功能

- 是有序(planned)迭代开发的基础
- 为编写最终用户手册提供索引(index)
- 抽取可单独出售的系统单元(Business Component)

以构架为中心的 **RUP**

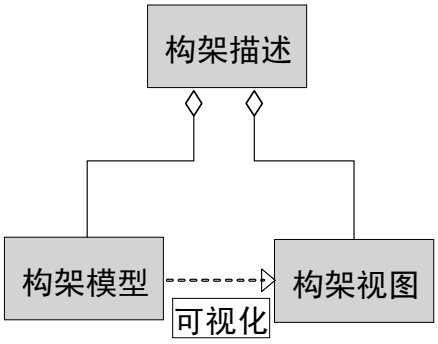
火龙果●整理
uml.org.cn

什么是构架

- 与建筑施工图的作用相似
  - 从各个角度刻画系统的总体设计(结构、服务设施、供暖、供水、供电…)
  - 去掉细节，突出系统的重要特征
- 软件构架(UML用户指南中的定义)
  - 系统的重要元素，如子系统、类、构件和节点，以及它们之间通过接口实现的协作
  - 软件系统的组织(结构与行为)
  - 构架风格(数据流风格、调用/返回风格等…)
  - 功能、性能、适应性、重用性、技术约束、美学考虑

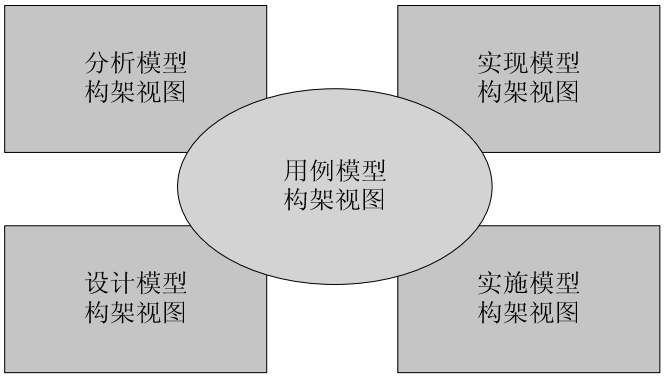
构架模型+构架视图=构架描述

- 模型是概念和知识
- 视图是表达模型的方式



火龙果●整理
uml.org.cn

构架视图



对照 Kruchten 提出的 4+1 view of software architecture

统一过程以构架为中心

- 构架为用户和开发人员提供系统的整体视图
- 构架是系统实现的基础
- 为项目管理提供基本指导
- 构架描述是软件系统的主要制品

火龙果●整理
uml.org.cn

以构架为中心的优点

- 创建可重用的框架，使构架级的重用成为可能
- 从构架可以方便地得到其他制品
  - 设计指南(包括使用模式和术语)
  - 形成产品结构系列(企业版、家用版…)
  - 开发队伍结构
- 简化基于构件的开发(Component-Based Development)

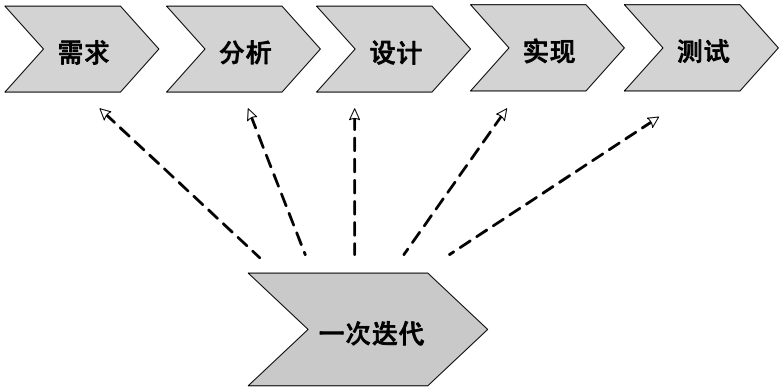
迭代和增量的 **RUP**

火龙果●整理
uml.org.cn

统一过程是迭代和增量的过程

- 迭代
  - 是统一过程中最小的开发时间单位，但它却包括了软件开发的所有工作流，因此也可以被看作是“袖珍瀑布模型”
- 增量
  - 是每次迭代所产生的，可增加系统功能的构造块
- 迭代&增量
  - 迭代是开发方式，增量是开发结果

迭代工作流



火龙果•整理
uml.org.cn

迭代是什么

- 是一组明确活动(activities)的集合
- 是有序的、受控的开发方式
- 是能够产生内部版本的袖珍项目
- 是项目开发过程中具体执行的工作流

迭代不是什么

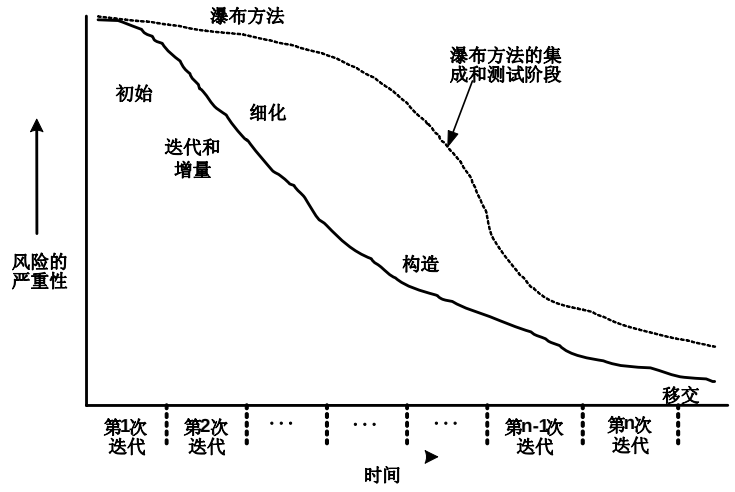
- 不是任意的探路
- 不是重复设计同样的东西
- 不是只影响开发人员
- 不是项目失败的理由

火龙果●整理
uml.org.cn

为什么采用迭代和增量的开发方法

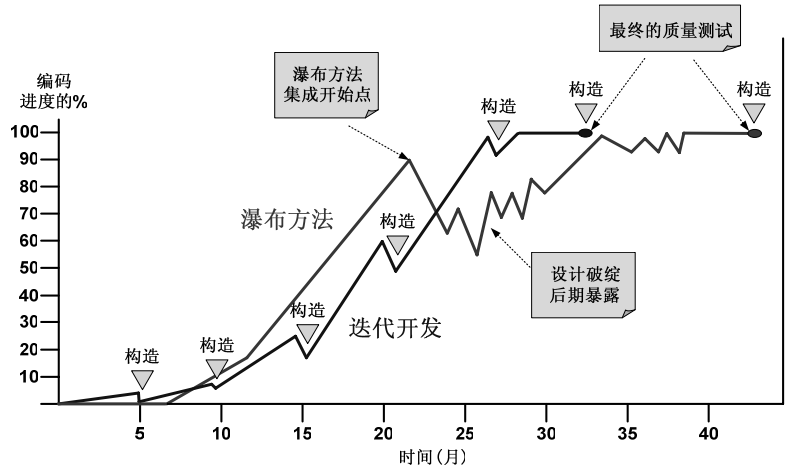
- 降低风险
  - 尽早处理关键风险和重要风险，同时将项目开发的总风险分散到每次的迭代中
- 小步前进
  - 允许系统灵活改变，更易于处理不断变化的需求
- 受控增长
  - 避免回头式开发，允许连续的系统集成
- 增强信心
  - 使用户和开发人员都可以看到系统中不断增加的可运行功能，对项目进度充满信心

迭代开发对风险的处理



火龙果●整理
uml.org.cn

迭代开发实现持续的构造集成



用例驱动、构架中心和迭代与增量

- 用例驱动系统构架的设计
- 系统构架影响用例的实现
- 用例体现了系统的功能(function)
- 构架反映了系统的表现形式(form)
- 用例设定了目标，构架建立了模式，开发人员根据目标和模式规划产品迭代开发的顺序

火龙果●整理
uml.org.cn

概述
Agile方法的产生

针对上述问题，产生了一系列轻载(Lightweight)方法，如XP，SCRUM等。

2001年2月，新方法的一些创始人在美国犹他州成立Agile 联盟(http://www.agilealliance.org/)

Lightweight ➡ Agile

概述
Agile方法的含义

Agile方法是在保证软件开发有成功产出的前提下，尽量减少开发过程中的活动和制品的方法。笼统的讲就是，“刚刚好”（Just enough），即开发中的活动及制品既不要太多也不要太少。

火龙果•整理
uml.org.cn

概述

Agile方法的实践效果

我预言XP对当今时代的作用可与CMM在八十年代和九十年代初的作用相媲美

-- Tom DeMarco, Cutter Trends Report

新方法在实践中取得了巨大的成功

•IONA公司的Obix技术支持小组在采用了XP方法后，软件生产率提高了67%

• SPG(software productivity group) 的Capers Jones则称，SCRUM方法可提高生产率6倍

Agile方法的核心理念

- 基于适应而非预测
- 以人为导向而非过程导向

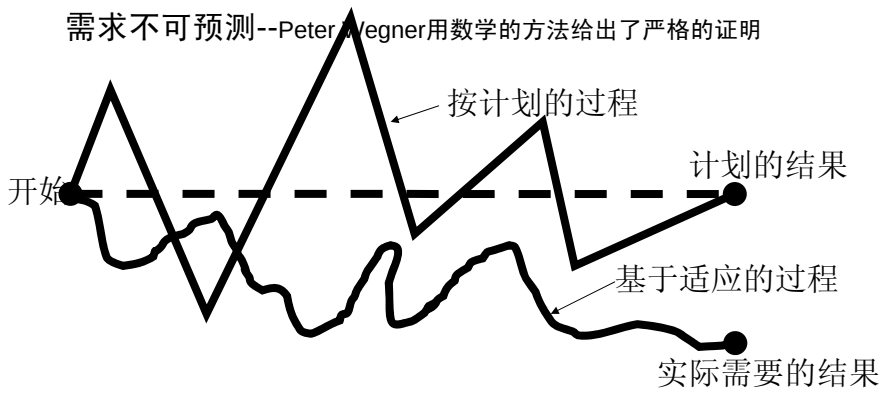
--Martin Fowler “New Methodology”

火龙果●整理
uml.org.cn

Agile方法的核心理念及特点

适应而非预测

需求不可预测--Peter Wegner用数学的方法给出了严格的证明



理论上来说，软件开发应是一个自适应的跟踪过程

Agile方法的核心理念及特点

适应而非预测

自适应系统是一个强反馈系统

- 在软件开发中，需求的获取和分析、软件设计、编码等实质上均为前馈环节，真正的反馈环节应该是用户对可运行软件的使用、使用中的判断及判断后与开发人员的信息交流。
- 反馈和前馈这一回路的响应速度应大于被跟踪（或被适应）的系统的变化速度，这就要求软件开发有快速的产出能力 。

火龙果●整理
uml.org.cn

Agile方法的核心理念及特点
适应而非预测

特点

Agile方法通过快速、短迭代式的开发，不断产出和演化可运行软件，根据用户的反馈信息作适应性调整，然后进入下一轮快速短迭代式开发。

Agile方法的核心理念及特点
以人为导向而非过程导向

软件开发中，人的因素是第一位的

- 人是过程的主体，而人的工作承受力是有限的
- 软件开发中的大部分是需要创造力的设计工作，软件人员是创造性的工作者
- 软件人员有主观上做好工作的意愿
- 软件开发的目的是为人提供方便，应首先着眼于有用的可执行的软件，也就是首先考虑商务目标，而不是为过程而过程

火龙果•整理
uml.org.cn

Agile方法的核心理念及特点

以人为导向而非过程导向

强调软件开发中相关人员间信息交流

- 软件开发中的中间过程和制品（文档），追根究底其目的是为了交流
- 项目失败的原因最终都可追溯到某个信息没有及时准确地传递到应该接收它的人 -- Alistir Cockburn
- 人特别擅长面对面的交流，面对面交流的成本要远远低于文档交流的成本 -- Alistir cockburn

Agile方法的核心理念及特点

以人为导向而非过程导向

Agile方法的一个共同特点:

努力营造诚信、开放的组织氛围，根据项目中信息流通的具体情况，按高内聚、松耦合的原则，将项目组划分为若干个小组（每个小组以不超过10人为宜，组员均在一个工作间内工作），通过小组内各种渠道的沟通，来减少中间制品的工作负担，提高应变能力。

火龙果●整理
uml.org.cn

Agile方法的核心理念及特点

Agile方法的价值系统和指导原则

- 任何软件开发方法都有一个相应的价值系统(Value system)，方法通过价值系统对过程予以指导，方法只有在其应用周境（context）与价值系统相吻合时才能发挥真正效力
- Agile联盟提出了“四个价值”、“十二个指导原则”

Agile方法的核心理念及特点

Agile方法的价值系统

- 较之于**过程和工具**，更注重**人及其相互作用**的价值
- 较之于**无所不及的各类文档**，更注重可**运行的软件**的价值
- 较之于**合同谈判**，更注重**与客户合作**的价值
- 较之于**按计划行事**，更注重**适应需求变化**的价值

火龙果●整理
uml.org.cn

Agile方法的核心理念及特点

Agile方法的指导原则

（1）在快速不断地交付用户可运行软件的过程中，将使用户满意放在第一位
（2）以积极的态度对待需求的变化（不管该变化出现在开发早期还是后期）
（3）以几周到几个月为周期，尽快、不断地交付可运行的软件供用户使用
（4）在项目过程中，业务人员和开发人员最好能一起工作
（5）以积极向上的员工为中心建立项目组，给予他们所需的环境和支持，对他们的工作予以充分的信任
（6） 在项目组中，最有用、最有效的信息沟通手段是面对面的交谈

Agile方法的核心理念及特点

Agile方法的指导原则

（7）测量项目进展的首要依据是可运行的软件
（8） Agile过程高度重视可持续开发
项目发起者、开发者和用户应能始终保持步调一致
（9） 应时刻关注技术上的精益求精和设计的合理，这样能提高软件的快速应变力
（10）简单化（尽可能减少不必要工作的艺术）
（11）最好的框架结构、需求和设计产生于自组织的项目组
（12）项目组要定期对其运作情况进行反思，提出改进意见，并进行相应的微调

火龙果●整理
uml.org.cn

Agile方法的核心理念及特点
Agile方法的适用范围

Martin Fowler认为：新方法不是到处可适用的

适合采用Agile方法的情况：

- 需求不确定、易挥发（Volatile, 意指今天的要求明天就不需要了）
- 有责任感和积极向上的开发人员
- 用户容易沟通并能参与

Agile与CMM

CMM的视角（ Mark Paulk 的看法）

(1) CMM更注重管理问题（组织过程的有效性和过程的系统化改进），Agile更注重技术和效率

(2) CMM提供了一个高度抽象的框架，有广泛的适用范围，Agile适用于小组织和需求不定、有用户紧密参与的情况(在高可靠性要求和大型项目组，或虚拟项目组中不宜采用)

(3) CMM与Agile方法虽有一些公共的特点，但Agile方法只是满足或部分满足了CMM2到3级中部分KPA（关键过程区域）的要求

(4) Agile方法提出了在某些周境（context）下非常有效的实践，与CMM方法有一定的互补性，CMM着重于“应该做什么”，Agile则侧重于“如何做”

火龙果●整理
uml.org.cn

Agile与CMM

XP的视角（ Ron Jeffries的看法）

- XP方法从某种意义上来说是CMM2到5级的一个垂直切片（满足了CMM2到5级中的部分KPA目标要求）
- 若将之应用于整个组织则还需更多的度量工作，但他同时指出，XP方法中更多的度量不是不可以做，而是要根据投入回报分析决定是否有必要

Agile与CMM

个人观点（仅供参考）

- CMM更注重质量，Agile更注重生产率
- CMM强调过程的可观测性，Agile强调可观测的结果（可运行软件）
- CMM注重管理和过程，Agile注重技术和效率
- CMM注重组织，Agile注重个人
- CMM无所不包（Universal)，Agile有明确的适用范围
- 它们都包含了一些软件工程的好的实践（Practices)

结合点在哪里，如何Just Enough?

火龙果●整理
uml.org.cn

Agile 具体方法介绍

- Extreme Programming (简称XP)
- SCRUM
- Crystal Methodologies (简称Crystal)
- Feature Driven Development(简称FDD)
- Dynamic Systems Development Methodology(简称DSDM)
- Adaptive Software Development(简称ASD)
- Pragmatic Programming等

什么是XP

Kent Beck 1996

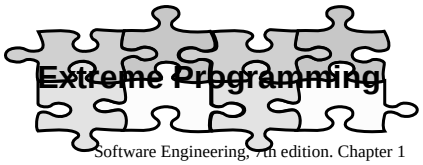
XP is a lightweight methodology for small to medium sized teams developing software in the face of vague or rapidly changing requirements.

-- Kent Beck.

XP是勇气，交流，反馈和简单。

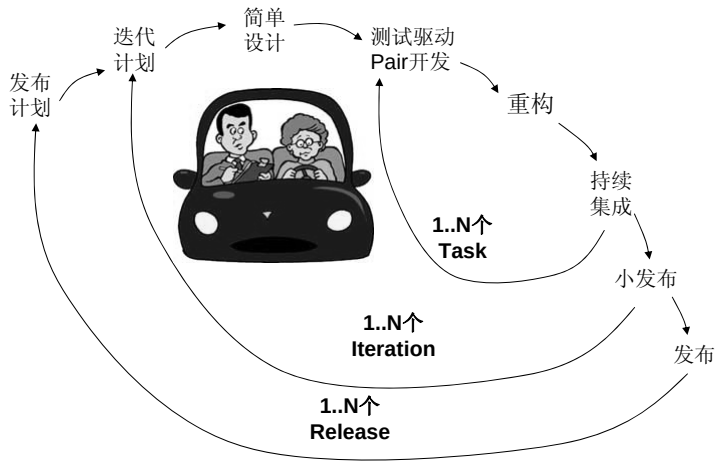
XP是软件开发过程中的纪律，它规定你：必须在编程前些测试，必须两个人一起编程，必须遵守编程规范……。

XP是把最好的实践经验提取出来，形成了一个崭新的开发方法。



火龙果●整理
uml.org.cn

XP的增量过程



XP价值观

交流(Communication)

简单(Simplicity)

反馈(Feedback)

勇气(Courage)

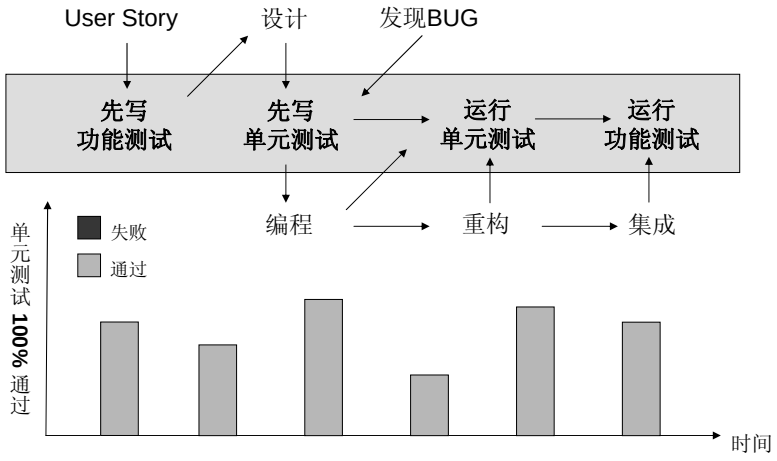
简单即为在管用能完成事情的前提下，做最简单的事；交流即整个开发过程应该都需要及时交流，这里交流侧重口头交流和能简单明了说明问题的文档交流，屏弃烦琐杂乱的文档和计划等；反馈即整个项目任何时候都需要客户和内部人员的反馈，以保证整个软件不断处于设计与编程与修复BUG的状态中；勇气即要求你必须有足够信心对自己的代码乃至别人的代码进行重构。

火龙果●整理
uml.org.cn

十二条惯例和规则

- On-Site Customer
（现场客户）
- 计划项目
（Planning Game）
- 频繁地小规模发布软件
（Small Releases）
- 简单设计
（Simple Design）
- 测试驱动开发
（Test Driven Development）
- 持续集成
（Continuous Integration）
- 集体拥有代码
（Collective Code Ownership）
- 编程规范
（Coding Standards）
- 重构
（Refactoring）
- System Metaphor
（系统隐喻）
- Pair Programming
（结对编程）
- 平稳的工作效率
（Sustainable Pace）

测试驱动开发(Test-Driven Develop, TDD)



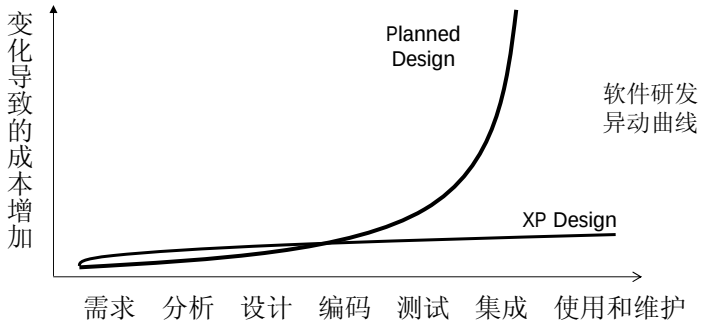
火龙果●整理
uml.org.cn

重构

- 减少重复设计，优化设计结构，提高技术上的重用性和可扩展性。
- 在Metaphor指引下的重构，是为商业模型服务的。不要把重构变成不断的盲目精简代码。
- 重构和编程前的计划型设计(Planned Design)结合，使XP的简单设计可行有效。
- XP提倡毫不留情的重构(Refactor mercilessly)。
- 任何人可以重构任何代码，前提是重构后的代码一定要通过100%测试单元测试后才能被Check-in。
- 可以根据需要，将一个迭代的全部目标定为重构。
- 不要太在意什么是最简单的设计 —— 愿意在最后重构，比知道如何做简单的设计重要得多。

简单设计

XP中的演进设计（Evolutionary-design）



- 如果没有它和众多惯例规则之间的耦合，XP的演化设计就蜕化成CODE-FIX。
- XP的演化设计是在Up-front design和Refactoring之间找到新的平衡。

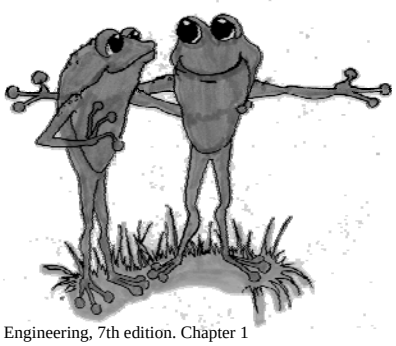
火龙果●整理
uml.org.cn

编程规范

- 规定了程序的风格，包括注释如何写，变量命名的规范，代码的格式等等。
- Teamwork 的前提之一，其它众多惯例和规则(如Pair Programming，Collective Code Ownership等)的前提之一。

集体拥有代码

- “我们”的代码，而不是“我”的代码。
- 任何人可以改动任何一段代码，但改动后的代码必须通过所有相关的测试。
- 简单设计，编程规范和Pair Programming，使阅读和修改Team内其他人的代码变得实际可行。



火龙果●整理
uml.org.cn

Pair Programming

两个程序员使用同一台电脑共同开发。**XP的必须组成部分，XP中最有争议的规则之一。**

频繁地小规模发布软件

- 发布过程应该尽可能地自动化、规范化。
- 不断地发布可用的系统可以告诉客户你在做正确的事情。
- 客户使用发布的系统，可以保证频繁地反馈和交流。
- 保证客户有足够的依据调控开发过程(增加、删除或改变User Story)。
- 降低开发风险。
- 随着开发的推进，发布越来越频繁。
- 所有的发布都要经过功能测试。

火龙果●整理
uml.org.cn

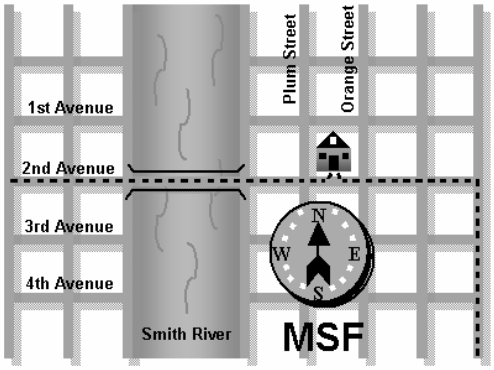
平稳的工作效率

平稳的工作效率指Team和个人在很长的时期内保持一定的开发效率。

- 保证了项目速度和计划过程的有效性和准确性；
- 保证了程序员可以持续地完成任务，Team可以持续地向客户交付可运行的系统(见敏捷开发宣言)；
- 加班多导致开发效率和质量下降，简洁增加了开发成本；
- Pair Programming已经加大了工作强度，并且和其它XP的规则一起提高了工作效率，使少加班和维持平稳的工作效率可能而且可行。
- 提倡平稳的工作效率，体现了XP以人为本的价值观。

MSF是微软解决方案框架

- 框架不是具体的软件程序或解决方案
- 框架只是一组模型、原则、概念方法，以帮助使用者开发良好解决方案为目的



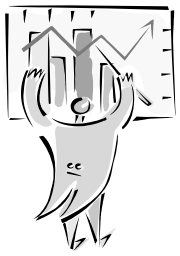
火龙果●整理
uml.org.cn

MSF主旨和观点

- MSF对环境、采用的技术、项目的大小有较大的适应性，这是MSF的第一个主旨
- MSF对软件开发还持以下独到的见解
  - 用户的需求是变动的
  - 需求是未来的，而不是当前的
  - 资源永远的匮乏的
  - 风险普遍存在
  - 开发小组成员是协作的平等关系
  - 认识是渐进的，过程是迭代的
  - 技术模型也可以影响业务模型

MSF概念

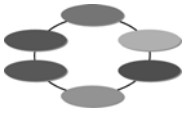
- MSF
  - 是一系列的指导方针
  - 目的在于用更快的速度、更少的人力、更小的风险、更高的质量，成功交付信息技术解决方案。



火龙果●整理
uml.org.cn

两种模型

- 小组模型
  - MSF提供了一个有机的建组模型，由不同业务职能人员形成开发组，人员以角色承担组内应负的职责
  - 角色之间是协商和合作关系。小组的扩大与缩减也都以角色承担工作量为准



过程模型

MSF提供了一个既分阶段又有反复迭代的软件开发模型

吸收了瀑布模型的经验和螺旋模型的优点

三种管理准则

- MSF有三种管理准则：项目管理准则、风险管理和就绪管理。
- MSF的项目管理
  - 不设项目经理
  - 项目管理活动尽量以当事人的意见为主
  - 强调项目管理与风险管理紧密结合
  - MSF项目管理非常重视实践

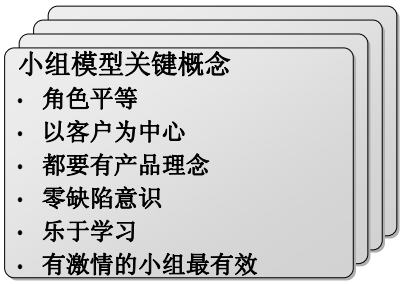
火龙果●整理
uml.org.cn

MSF的基本原则

- MSF八大基本原则
  - 职责明确、责任共享
  - 在一个共享的远景下工作
  - 小组成员有职有权
  - 重点是交付业务价值
  - 灵活敏捷、主动应变
  - 促进坦率沟通
  - 在质量上投资
  - 从各种经验中学习

关键概念

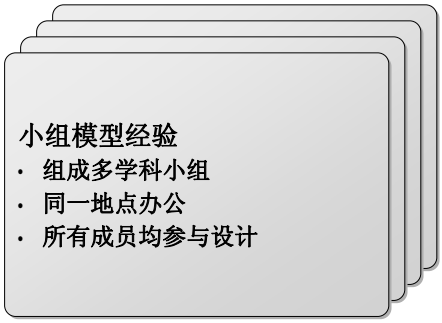
- MSF在形成自己的模型和准则时，为了体现MSF的基本原则形成一些关键的概念，即支持MSF思想的理念。
- MSF为每一种模型，每一种准则给出了关键概念



火龙果●整理
uml.org.cn

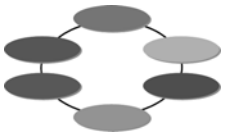
经实践检验正确的经验

- 除了基本原则和关键概念之外。MSF还为每种模型，每种准则提供了采用模型，实施管理时的实际经验，这些经验是经过实际验证切实可行的。

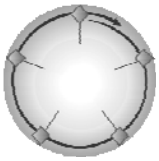


总结：MSF基本要素

小组模型



过程模型



项目管理准则
管理和满足承诺

风险管理准则
增加成功的潜力

就绪管理准则
在正确的时间拥有正确的技能

火龙果●整理
uml.org.cn

CMM

CMM的基本概念

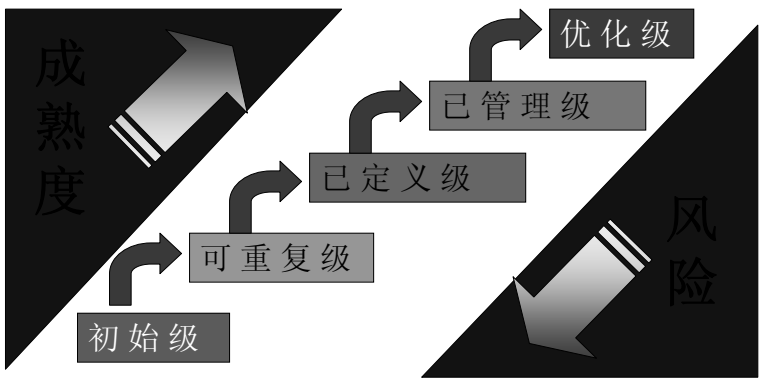
• 能力成熟度模型（Capability Maturity Model）

• 80年代，在美国国防部资助下，由卡内基梅隆大学软件工程研究所(SEI)建立，用于评价 软件开发组织的软件过程能力成熟度

• 后来此模型被用于软件开发组织内部的软件过程改进。

CMM

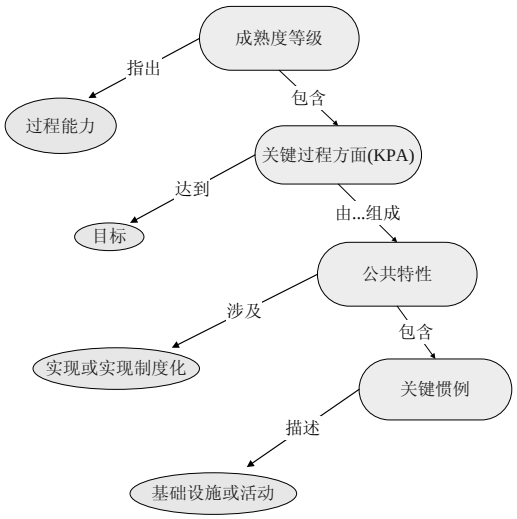
CMM的基本概念



火龙果●整理
uml.org.cn

CMM

CMM的基本概念



CMM结构图

CMM

CMM的基本概念

美国曾在1995年做过软件产业成熟程度的调查，发现在美国的软件产业中：

- CMM成熟度等级为初始级的竟占70%，其特征是软件开发过程不能预测，风险度高；
- 为可重复级的占15%，其特征是软件开发过程需小心谨慎方能避免失败；
- 为定义级的所占比例小于10%，其特征是软件开发过程相当稳定，进展顺利且可以预测；
- 为管理级的所占比例小于5%，其特征是软件过程预测准确、值得信赖；
- 为优化级的所占比例小于1%，其特征是软件过程能持续改善。

火龙果•整理
uml.org.cn

CMM

PSP的基本概念

个体软件过程PSP(Personal Software Process)，为基于个体和小型群组软件过程的优化提供了具体而有效的途径，例如

- 如何制订计划
- 如何控制质量
- 如何与其他人相互协作等等

CMM

PSP采用的方法

- PSP保障软件产品质量的一个重要途径是提高设计质量
- 在软件设计阶段，PSP的着眼点在于软件缺陷的预防，其具体办法是强化设计结束准则，而不是设计方法的选择

火龙果•整理
uml.org.cn

CMM

PSP的效用

研究表明：

- 绝大多数软件缺陷是由于对问题的错误理解或简单的失误所造成的，只有很少一部分是由于技术问题而产生的
- 如果在设计阶段注入一个差错，则这个差错在编码阶段要引发3～5个新的缺陷，要修复这些缺陷所花的费用要比修复这个设计缺陷所花的费用多10倍

CMM

PSP的效用

研究表明：

- 根据对参加培训的104位软件人员的统计数据表明，在应用了PSP后，软件中总的缺陷减少了58.0%，在测试阶段发现的缺陷减少了71.9%，生产效率提高了20.8%

火龙果•整理
uml.org.cn

CMM

TSP的基本概念

群组软件过程TSP(Team Software Process)，

- 指导项目组中的成员如何有效地规划和管理所面临的项目开发任务
- 告诉管理人员如何指导软件开发队伍始终以最佳状态来完成工作
- TSP实施集体管理与自己管理自己相结合的原则，最终目的在于指导一切人员如何在最少的时间内，以预定的费用生产出高质量的软件产品

CMM

TSP所采用的方法

- 对群组软件开发过程的定义、度量和改进
- 实施TSP的先决条件有3条：
  - ✓需要有高层主管和各级经理的支持，以取得必要的资源
  - ✓项目组开发人员需要经过PSP的培训并有按TSP工作的愿望和热情
  - ✓整个软件企业在总体上应处于CMM二级以上

火龙果•整理
uml.org.cn

CMM

TSP所采用的方法

在实施TSP的过程中，

- 要有明确的目标
  - ✓ 开发人员在每一阶段开始，要做好工作计划
  - ✓如果发现未能按期按质完成计划，应分析原因，以判定问题是由于工作内容不合适或工作计划不实际所引起，还是由于资源不足或主观努力不够所引起。

CMM

TSP所采用的方法

在实施TSP的过程中，

- 开发小组应随时追踪项目进展状态并进行定期汇报
- 开发小组应经常评审自己是否按PSP的原理工作

火龙果•整理
uml.org.cn

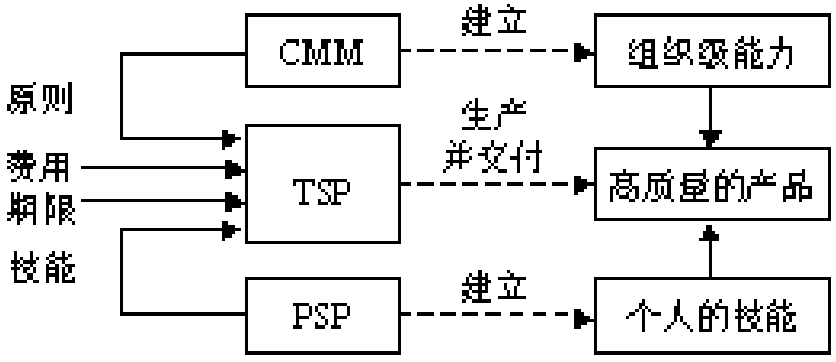
CMM

TSP所采用的方法

在实施TSP的过程中，

- 开发人员应按自己管理自己的原则管理软件过程，如发现过程不合适，应及时改进，以保证用高质量的过程来生产高质量的软件
- 项目开发小组则按集体管理的原则进行管理，全体成员都要参加和关心小组的规划、进展的追踪和决策的制订等项工作

CMM



火龙果•整理
uml.org.cn

经验借鉴

国外某些大公司软件工程管理经验：

- 软件工程管理方面的投资一般占软件开发费用的10%左右
- 软件过程的重大修改也必须由高层管理部门启动
- 软件过程的改善还有待于全体有关人员的积极参与，否则不仅他本人将失去从软件过程改善中获得提高的机会，甚至还会成为过程改善的阻力

经验借鉴

- 软件过程成熟度的升级本身就是一个过程，有一个生命周期。因此，过程改善工作需要循序渐进
- 要将CMM/PSP/TSP引入软件企业，最有效的途径是要对单位主管和主要开发人员进行系统的培训
- 中小企业可以以CMM为框架，先从PSP做起，然后在此基础上逐渐过渡到TSP，以保证CMM/PSP/TSP在企业中根深蒂固