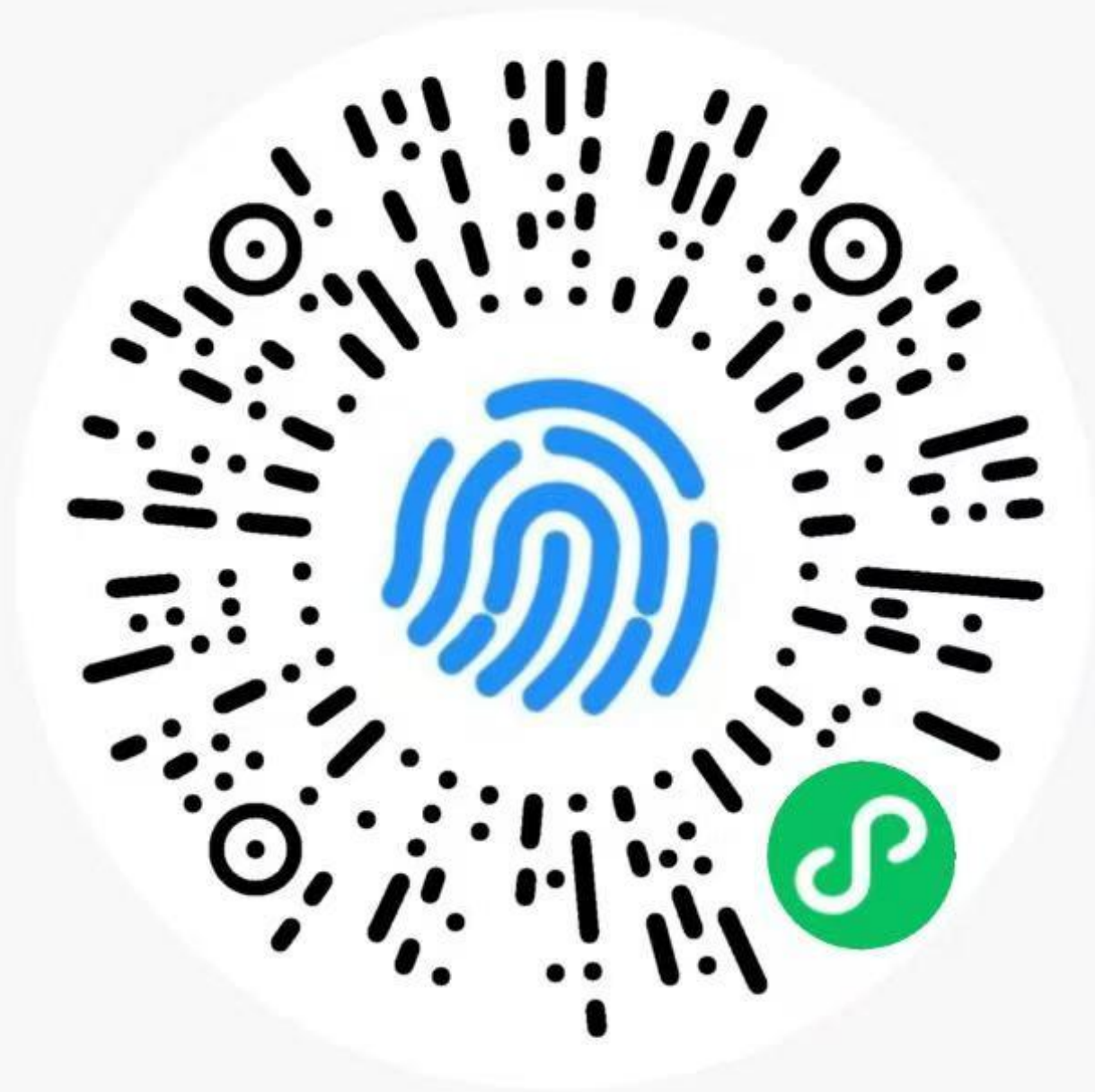


敏捷软件开发方法



微信扫码签到

内容

1. 何为敏捷开发方法

- ✓ 基本思想和原则
- ✓ 特点和应用

2. 具体敏捷开发方法

- ✓ 极限编程
- ✓ 测试驱动开发方法
- ✓ Scrum方法



1.1 传统重型软件开发方法的特点和不足

□遵循严格的过程和计划

- ✓ 定义广泛适用的过程并通过团队来执行该过程，从而指导软件开发
- ✓ 以预测性为主，倾向于预先制定详细的计划，并通过该计划来指导软件项目的实施，并期望软件开发过程与计划之间的偏差越少越好

□以文档为中心

- ✓ 通过文档记录各个阶段的成果，将文档作为交互的媒介
- ✓ 需对文档进行持续改进和评审
- ✓ 难以有效应对软件需求的变化

□等到开发后期才能得到可运行软件

- ✓ 编码是开发的后期工作

1.2 什么是敏捷（Agile）方法？

□ 一种轻量级软件开发方法

✓ 相对于重量级的软件开发方法而言

□ 主张软件开发要以代码为中心，快速、轻巧和主动应对需求变化，持续、及时交付可运行的软件系统

✓ 轻便、轻巧

□ 提供了一组思想和策略，指导快速响应用户需求的变化，快速交付可运行的软件制品

1.3 敏捷开发方法的基本观点

- 较之于过程和工具，应更加重视**人和交互**的价值
- 较之于面面俱到文档，应更加重视**可运行软件系统**的价值
- 较之于合同谈判，应更加重视**客户合作**的价值
- 较之于遵循计划，应更加重视**响应用户需求变化**的价值

敏捷方法体现的思想

□强化可运行的软件，弱化文档

- ✓以可运行软件为中心来开展软件开发

□以适应变化为目的来推进开发

- ✓针对变化不断进行优化和调整任务、产品和计划等

□以人为本

- ✓敏捷软件开发是面向人的而不是面向过程的，让方法、技术、工具、过程等来适应人，而不是让人来适应它们

敏捷意味着：轻盈、灵巧；无过多的负担；迅速响应变化

1.4 敏捷准则(1/2)

- **尽早和持续地交付**有价值的软件，以使用户满意
- 即使到了软件开发后期，也**欢迎用户需求的变化**
- **不断交付**可运行的软件系统，**交付周期**可以从几周到几个月
- 在整个软件项目开发期间，用户和开发人员最好能**每天一起工作**
- 由**积极主动**的人来承担项目开发，给他们提供所需环境和支持，信任他们的能力
- 团队内部最有效的信息传递方式是**面对面的交谈**

敏捷准则(2/2)

- 将**可运行软件**作为衡量软件开发进度的首要标准
- 可持续性的开发，出资方、开发方和用户方应当保持**长期、恒定的开发速度**
- 关注**优秀的技能和良好的设计**会增强敏捷性
- 简单化**
- 最好的架构、需求和设计出自于**自组织的团队**
- 软件开发团队应定期就如何提高工作效率的问题进行**反思**，并进行相应的**调整**

敏捷软件开发对技术提出的要求

- 如何快速开发出可运行的软件系统？
- 当需求改变时，如何快速应对变化？
- 如何给出可有效应对变化的软件设计？
- 在文档缺乏情况下如何保证软件质量？
- 如何提高软件开发的效率？

1.5 支持敏捷软件开发的技术

- 极限编程
- 测试驱动开发
- Scrum方法
- 敏捷设计
- 模式运用
- 软件重构
- MDA-模型驱动开发
- CASE工具

敏捷开发方法是一大类方法的统称，它们遵循敏捷思想

内容

1. 何为敏捷开发方法

- ✓ 基本思想和原则
- ✓ 特点和应用

2. 具体敏捷开发方法

- ✓ 极限编程
- ✓ 测试驱动开发方法
- ✓ Scrum方法



2.1 极限编程的基本思想

□由Kent Beck提出的一种特殊的敏捷软件开发方法

□四条核心思想

✓**交流**，强调基于口头（而非文档、报表和计划）的交流

✓**反馈**，通过持续、明确反馈来获得软件状态

✓**简单**，用最简单的技术来解决问题

✓**勇气**，快速开发并在必要时具有重新进行开发的信心

□将经过数十年检验的准则结合在一起，定义了五条指导性原则和十二条须遵循的核心准则

2.2 极限编程的5条指导原则

□ 反馈速快

- ✓ 从用户处迅速得到有关软件的反馈，确认开发是否满足用户需求，通过自动化测试迅速了解软件运行状况

□ 设假性单简

- ✓ 开发人员只考虑当前迭代所面临问题，无需考虑下一次迭代的问题，用简单方法和技术来解决问题。

□ 改更步逐

- ✓ 通过一系列修改来逐步解决问题和完善系统，不要期望一次迭代就开发出完整的软件系统。

□ 支持化变

- ✓ 欢迎用户改变需求，支持用户需求动态变化。

□ 作工的量高

- ✓ 采用诸如测试驱动开发等技术高质量地开展工作，确保软件质量。

2.3 极限编程的12条核心准则(1/3)

□计划游戏

- ✓ 软件开发团队快速制定下一次迭代的软件开发计划

□隐喻(Metaphor)

- ✓ 使用业务相关术语来描述需求，促使开发人员和业务人员对系统达成共同和一致的理解

□小型发布

- ✓ 经常性发布可运行软件系统，每次发布的软件系统仅提供少量功能

□简单设计

- ✓ 只为当前的需求做设计，程序能运行所有测试、没有重复逻辑、包含尽可能少的类和方法

极限编程的12条核心准则(2/3)

□测试

- ✓测试应在编写代码之前进行

□重构

- ✓在不改变程序代码功能的前提下，改进程序代码的设计，使程序代码更加简单，更易于扩展

□结对编程

- ✓两名程序员同时在一台计算机上共同开展编程工作

□代码集体拥有

- ✓开发小组的任何成员都可以查看并修改任何部分的代码

极限编程的12条核心准则(3/3)

□持续集成

- ✓经常性地集成

□每周工作40小时

- ✓倡导质量优先

□现场用户

- ✓用户代表在现场办公，参与开发全过程，确保能及时得到反馈

□编码标准

- ✓遵循统一编码标准，以提高软件系统的可理解性和可维护性

2.4 传统软件开发的局限

□程序员先编写程序代码，然后再对程序代码进行测试

- ✓基于已有的程序代码进行测试

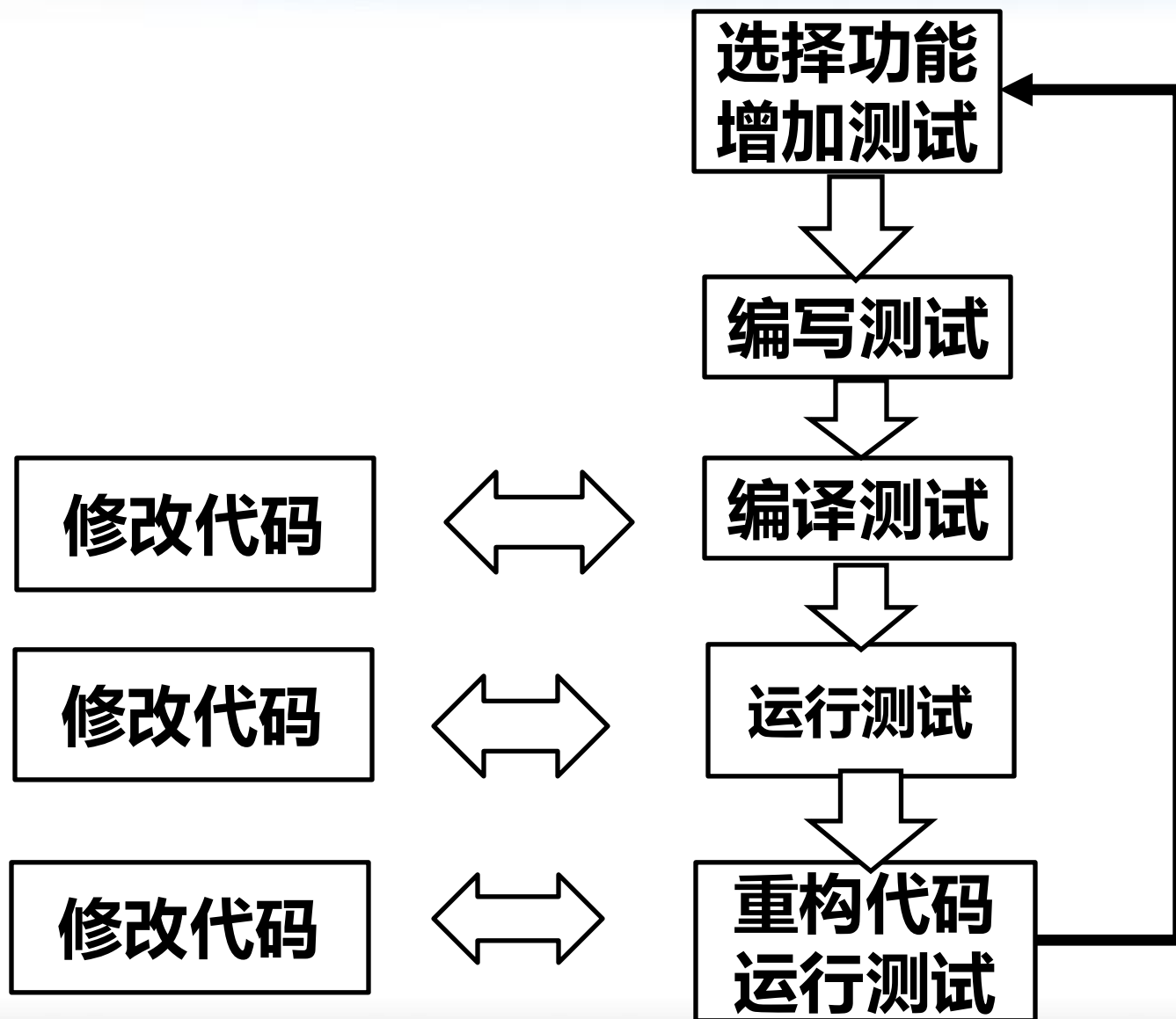
□局限性

- ✓测试常被视为是附加工作，由于进度压力，常被忽视，导致没有足够时间对代码进行详尽和充分的测试
- ✓当文档与程序代码不一致时，对程序代码进行测试就会存在许多问题
- ✓测试通常是在程序代码编写完成之后才进行的，因而**无法保证编写程序和测试同步**
- ✓测试被视为是乏味的工作，人员缺乏积极性和成就感

2.5 测试驱动开发的思想

- 在开发程序代码之前，先确定和编写测试
- 程序员首先要思考如何对某个功能进行测试，设计好相应的测试用例，编写好相关的测试代码，然后编写相应的程序代码以通过软件测试

测试驱动开发的过程



测试驱动开发的特点

- 根据测试来编写代码
- 编写测试的目的不仅是为了测试程序代码能够正常工作，而且被用于定义程序代码的内涵
- 确保任何程序代码都是可测试的
- 编码完成后即完工
- 易于维护、质量保证

2.6 Scrum方法

□旨在通过增量或迭代的方式加强软件项目的管理



Scrum方法的大致流程

□首先，产品拥有者需创建**软件产品订单库**即 “Backblog”

- ✓ 描述软件产品需提供的功能需求以及它们的优先级排序

□其次，筛选出最应该实现的软件需求，

- ✓ Scrum主人基于 “Backblog” 中各项软件需求及其优先级，形成待实现的软件产品冲刺订单库，即 “SprintLog”

□然后，软件开发将进入冲刺 “Sprint” 周期

- ✓ 以实现选定软件订单，每个冲刺就是一次增量开发，一般持续1到4周

□最后，共同开展Scrum评审

- ✓ 一次冲刺完成后，每个团队成员演示自己的开发成果，大家共同审查成果是否高质量地实现了既定功能，并就其中的问题进行反思，以指导和改进下一次冲刺

2.7 敏捷方法的特点

□小

- ✓生成少量软件文档，每个文档规模要小
- ✓每次迭代要实现软件功能的数量和规模要小，迭代周期要小

□简

- ✓技术、工具以及每次迭代要解决的问题尽可能简单
- ✓只关注当前欲实现的功能需求，而不要考虑将来的问题

□快

- ✓快速响应变化、从用户处获得反馈，给用户提交有价值软件，对软件产品进行迭代和更新

□变

- ✓允许需求动态变化，要以变应变，开发团队应是自组织的

问题思考和讨论

- 敏捷开发方法适合于哪些类别的软件开发？
- 敏捷开发方法不适合于哪些类别的软件开发？



小结

□敏捷开发方法的本质

- ✓应对软件**需求变化**，解决传统过程模型的不足

□敏捷开发方法的特点

- ✓小、简、快、变、体，**轻量级方法**，以**代码为中心**的方法

□敏捷开发方法的构成

- ✓由许多具体的方法组成，如Scrum方法、极限编程等

问题和讨论

