



# 敏捷软件开发

- 敏捷开发的由来
- 敏捷开发方法
- 敏捷开发流程
- 敏捷宣言
- 敏捷开发之原则
- 适用性分析



# 敏捷软件开发（Agile software development）

- 敏捷开发是一种从1990年代开始逐渐引起广泛关注的一些新型软件开发方法，是一种应对快速变化的需求的一种软件开发能力。
- 敏捷一词来源于2001年初美国犹他州雪鸟滑雪圣地的一次敏捷方法发起者和实践者聚会，17位业内志士发起组成敏捷联盟，发布《敏捷宣言》。
- 敏捷开发以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。在敏捷开发中，软件项目在构建初期被切分成多个子项目，各个子项目的成果都经过测试，具备可视、可集成和可运行使用的特征。
- 相对于“非敏捷”，更强调人的作用发挥：程序员团队与业务专家之间的紧密协作、面对面的沟通（认为比书面的文档更有效）、频繁交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法。
- 相对于“传统”方法，更强调适应性而非预见性：当项目的需求起了变化，团队应该高度协作，迅速适应。迭代周期更短、功能交付更频繁，并在整个项目周期中持续改善和增强。

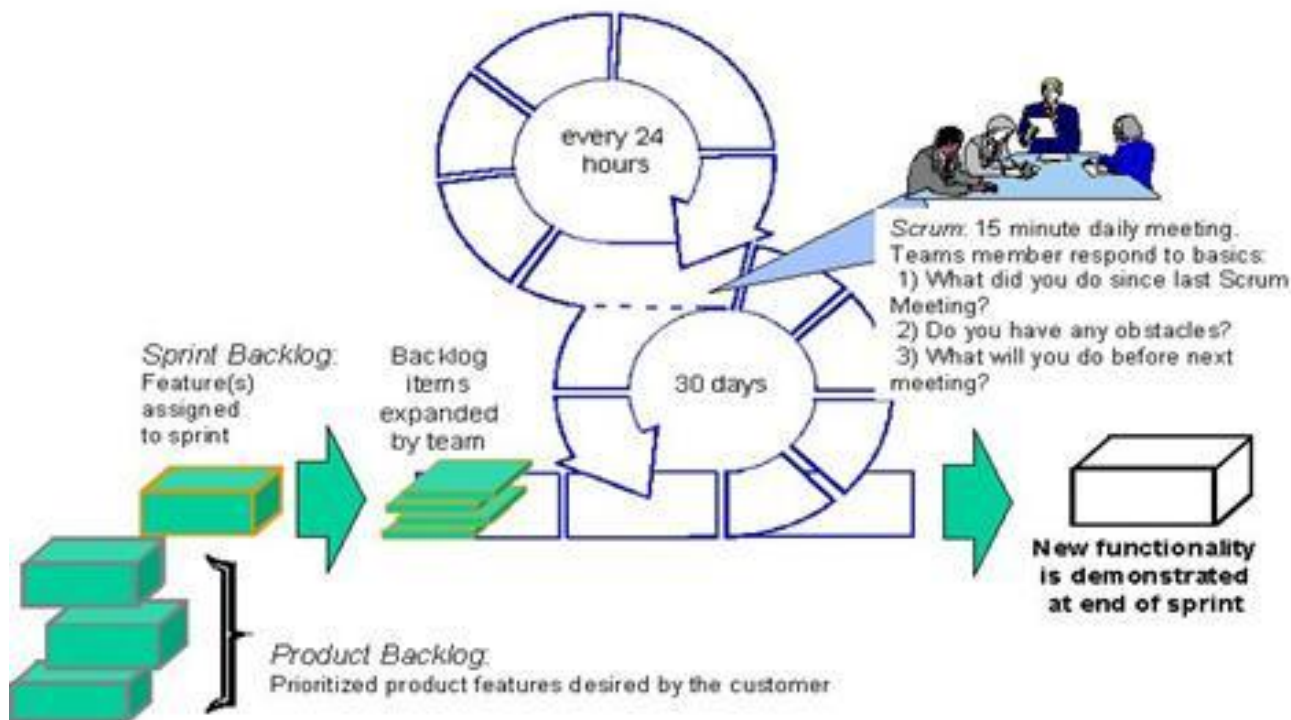


# 敏捷方法列表

- 软件开发之韵，Software Development Rhythms
- 敏捷数据库技术，AD/Agile Database Techniques
- 敏捷建模，AM/Agile Modeling
- 自适应软件开发，ASD/Adaptive Software Development
- 水晶方法，Crystal Development
- 动态系统开发方法，DSDM/Dynamic Systems Development Method
- 精益软件开发，Lean Software Development
- 敏捷统一过程，AUP（Agile Unified Process）
- Scrum敏捷软件开发
- XBreed敏捷软件开发
- 极限编程，XP Extreme Programming
- 测试驱动开发，TDD/Test-Driven Development
- 行为驱动开发，BDD/Behavior-Driven Development
- 特征驱动开发，FDD/Feature-Driven Development
- 验收测试驱动开发，ATDD（Acceptance Test Driven Development）



# 敏捷开发示意图



- 整个过程中夹杂了很多在敏捷开发前已经出现的软件开发方法，包括XP、Scrum、FDD、TDD等。这些方法在敏捷软件开发流程的各个阶段都有充分的体现和应用。
- **Scrum** 主要着重于项目管理，团队中的项目经理（**Scrum master**）需要在每个客户需求到来的时候制定 **Sprint** 的周期，定义每个 **Sprint** 的目标、分派任务、进行监督、最后总结得失并开始计划新的 **Sprint**。
- 测试人员需要在每天的站立会议（**Daily Standup Meeting**）上报告前一个工作日发现的新缺陷情况，项目经理根据项目进度和缺陷严重性来决定是否修复这些问题。需要及时修复的缺陷是目前 **Sprint** 中的一个新任务，将由项目经理添加到 **Sprint Backlog** 上并通知开发人员去修复漏洞。



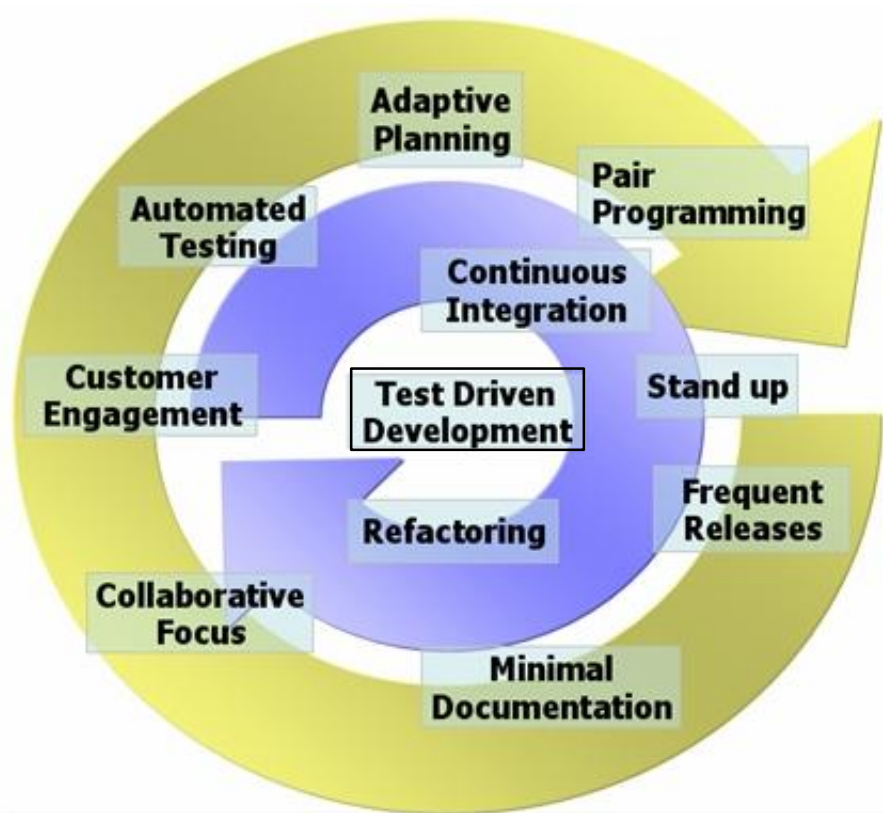
# 敏捷软件开发流程





# Test-Driven Development, 测试驱动开发

结对开发、测试为先：一个人依照系列Story，从业务需求的角度来编写测试用例，另一个人看着他并且进行思考，如果有不同的意见就会提出来进行讨论，直到达成共识，接着由另一个人编写该测试代码。如果没有测试代码，就不能编写功能的实现代码。

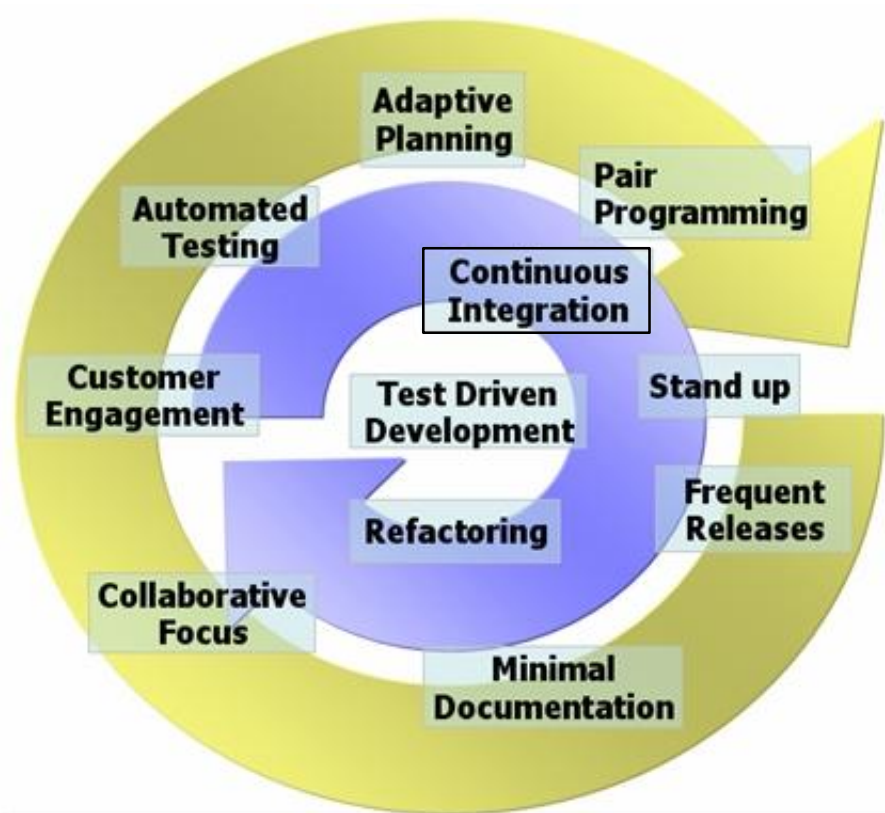






# Continuous Integration, 持续集成

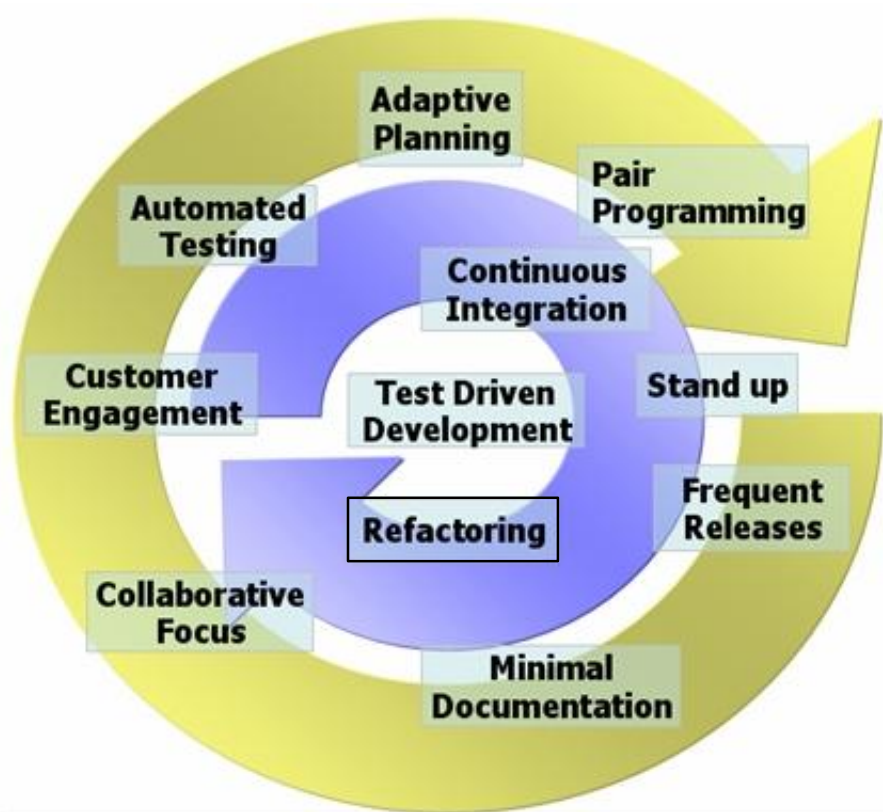
为减少冲突，一天之内做十几次甚至几十次集成，包括获得所有源代码、静态分析、编译源代码、运行所有测试、形成测试报告。





# Refactoring, 重构

重构是在不改变系统外部行为下，对内部结构进行整理优化，使得代码尽量简单、优美、可扩展。重构贯穿于整个开发流程，每一次开发者check in代码之前，都要对所写代码进行重构，让代码达到clean code that works。用单元测试来保证重构不引起冲突。

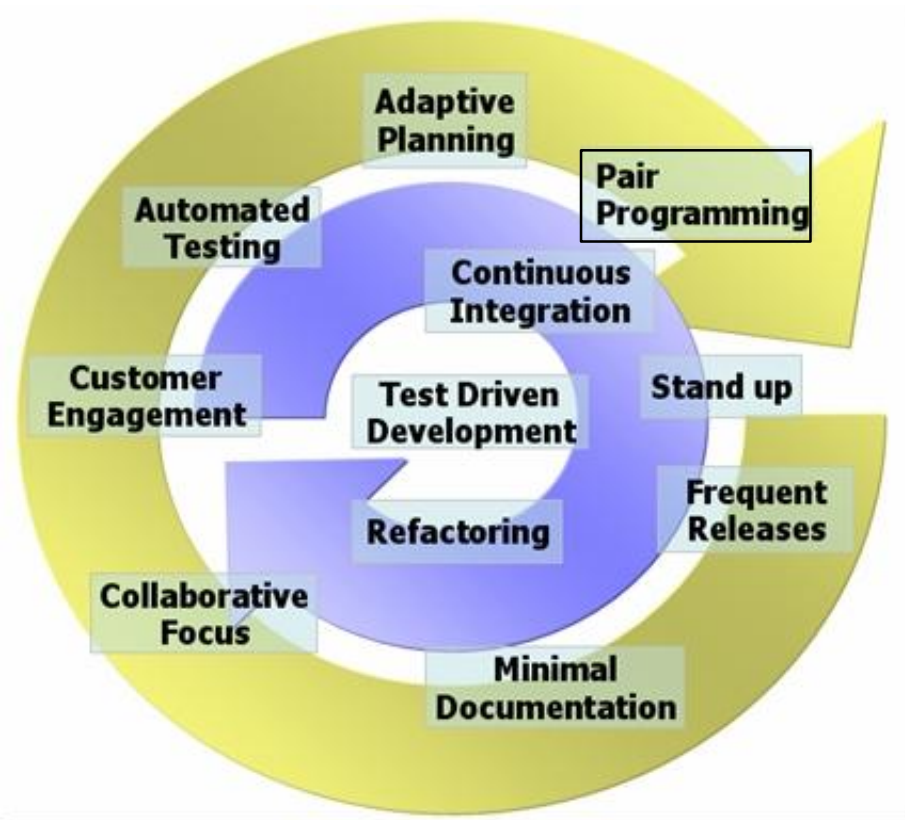






# Pair-Programming, 结对编程

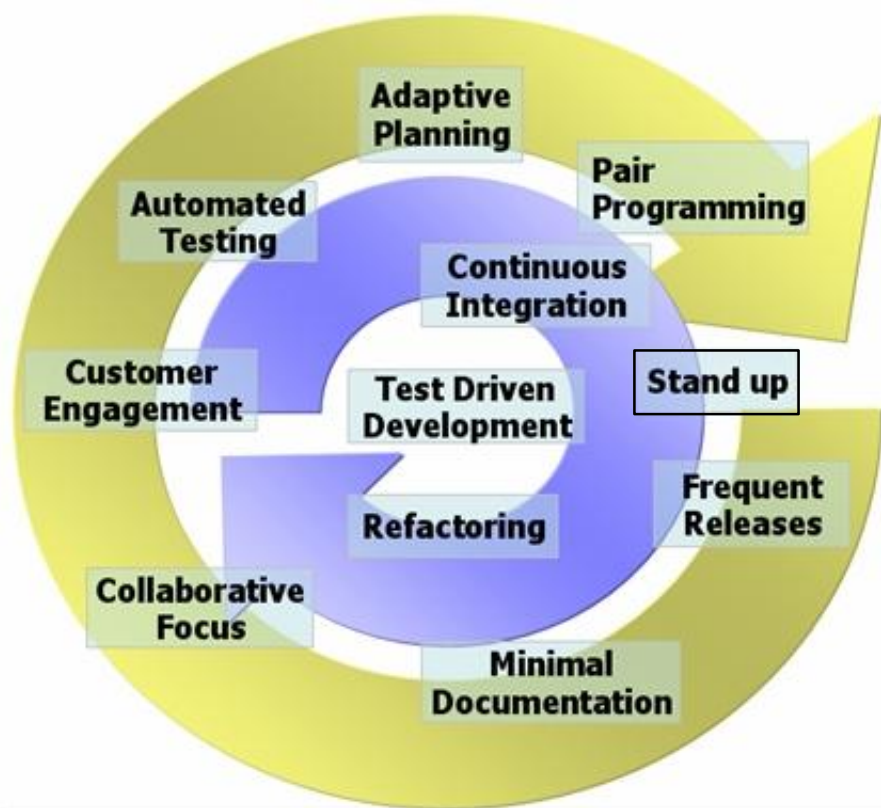
在敏捷开发中，做任何事情都是Pair的，包括分析、写测试、写实现代码或者重构。





## Stand up, 站立会议

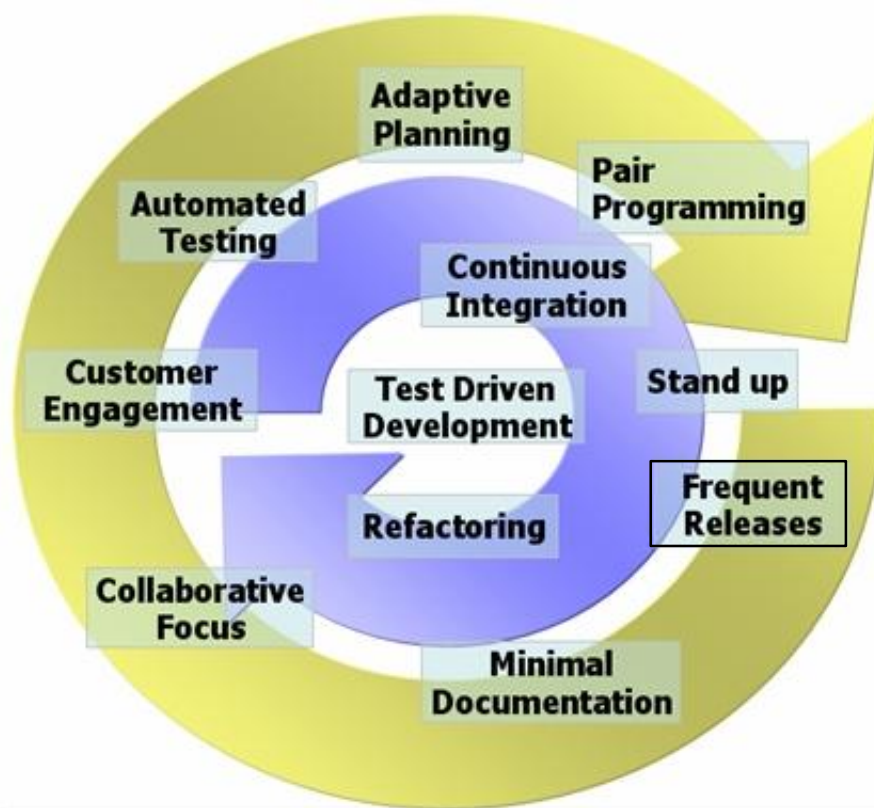
每天早上，项目组的所有成员都会站立进行一次会议，由于是站立的，所以时间不会很长，一般来说是**15-20分钟**。会议的内容并不是需求分析、任务分配等，而是每个人都回答三个问题：**1. 你昨天做了什么？ 2. 你今天要做什么？ 3. 你遇到了哪些困难？**站立会议让团队进行交流，相互熟悉分享讨论工作内容。





## Frequent Releases, 小版本发布

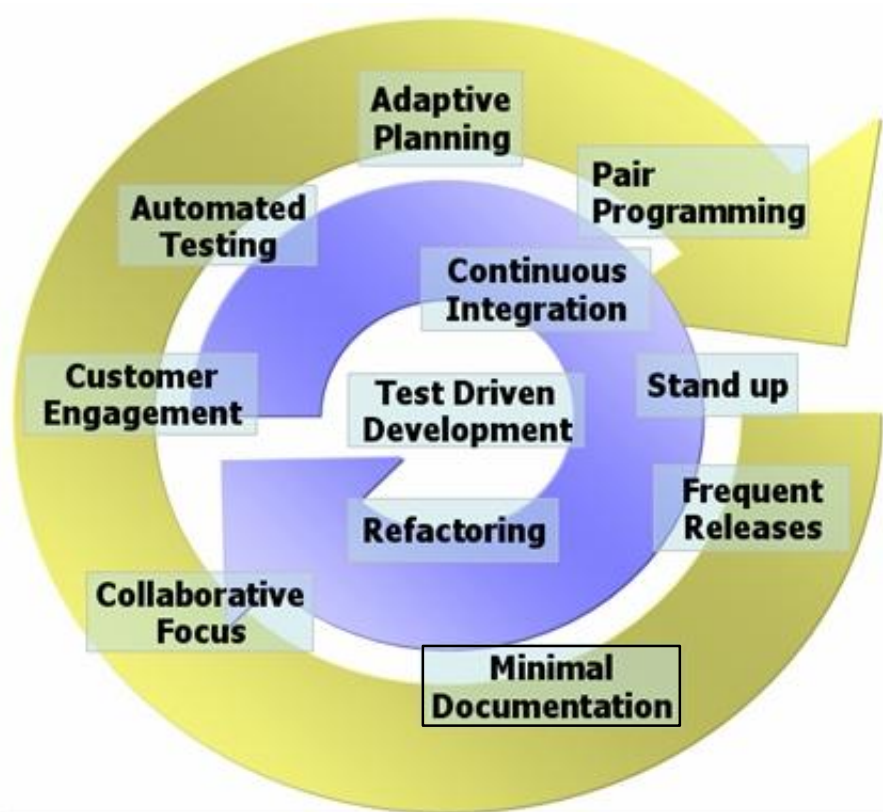
会有尽量多的产品发布，一般以周、月为单位。这样，客户每隔一段时间就会拿到发布的产品进行试用。正因为发布频繁，每一个版本新增的功能简单，不需要复杂的设计，这样文档和设计就在很大程度上简化了。又因为简单设计，没有复杂的架构，所以客户有新的需求或者需求进行变动，也能很快的适应。





## Minimal Documentation, 较少的文档

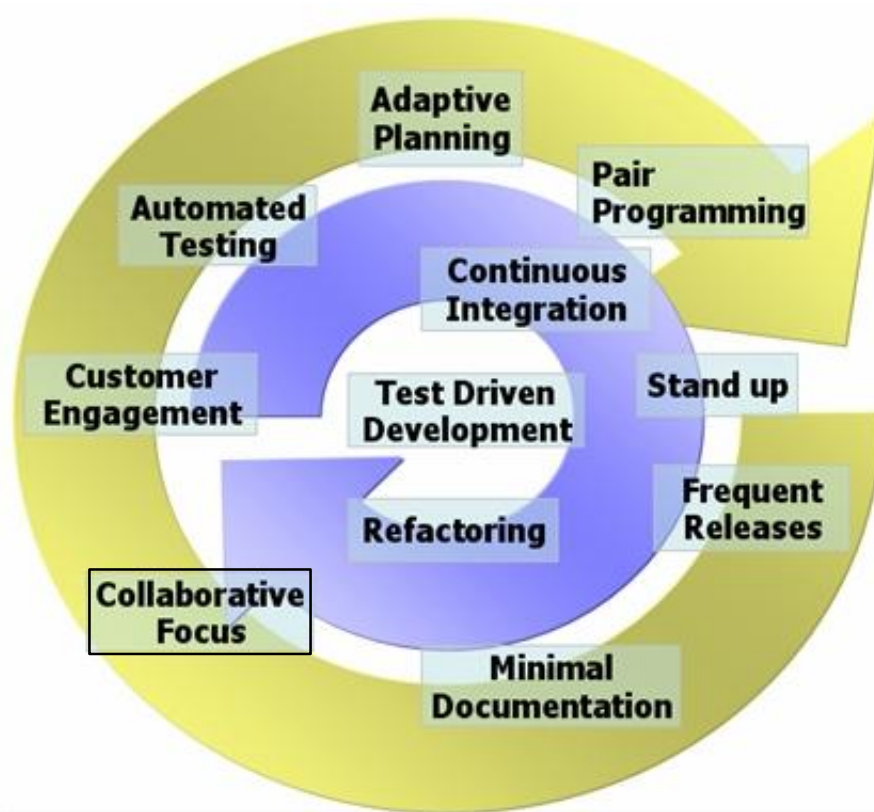
其实敏捷开发中并不是没有文档，而是有大量的文档，即测试。这些测试代码真实的反应了客户的需求以及系统API 的用法，如果有新人加入团队，最快的熟悉项目的方法就是给他看测试代码，测试是真实反应代码的。





## Collaborative Focus, 以合作为中心, 表现为代码共享

代码是归团队所有而不是哪些模块的代码属于哪些人, 每个人都有权利获得系统任何一部分的代码然后修改它并分享给其他人。这样每个人都能熟悉系统的代码, 即使团队的人员变动, 也没有风险。



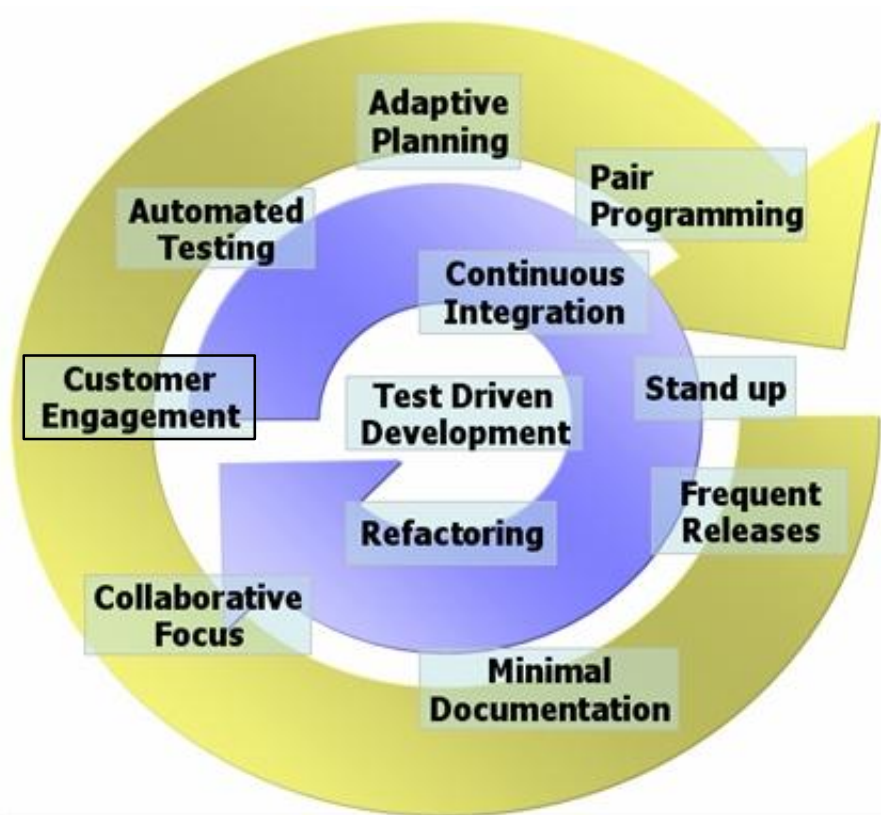




# Customer Engagement ， 现场客户

敏捷开发中，客户是与开发团队一起工作的，团队到客户现场进行开发或者邀请客户到团队公司里来开发。如果开发过程中有什么问题或者产品经过一个迭代后，能够以最快速度得到客户的反馈。

。

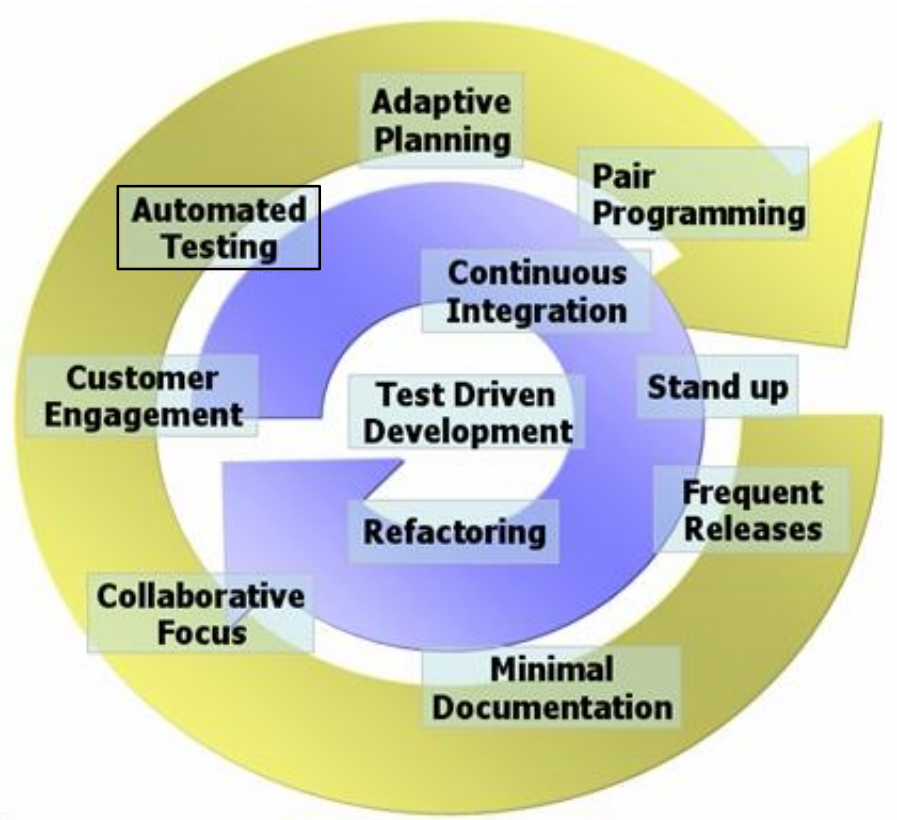






# Automated Testing ， 自动化测试

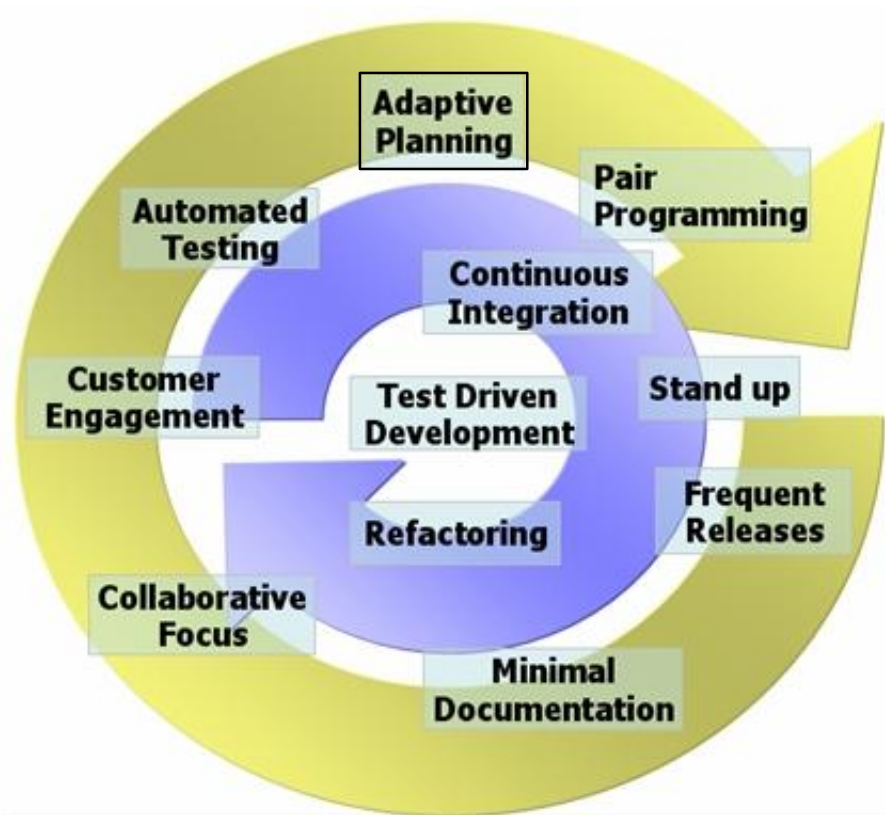
为了减小人力、提高效率，所有的测试包括单元测试、功能测试或集成测试等都是自动化的，这对QA人员提出了更高的要求。他们要熟悉开发语言、自动化测试工具，能够编写自动化测试脚本或者用工具录制。





# Adaptive Planning, 可调整计划

计划是可调整的，敏捷开发中只有一次一次的迭代，小版本的发布，根据客户反馈随时作出相应的调整 and 变化，所以计划往往是针对每次迭代制定的。





# 敏捷宣言

- 最重要的是通过尽早和不断交付有价值的软件满足客户需要。
- 为保持客户的竞争优势，能够驾驭变化、适应变化，即使在开发后期。
- 经常交付可以工作的软件，从几星期到几个月，时间尺度越短越好。
- 业务人员和开发者应该在整个项目过程中始终朝夕在一起工作。
- 给开发者提供适宜环境，满足他们的需要并相信他们能够完成任务。
- 在开发小组中最有效率也最有效果的信息传达方式是面对面交谈沟通。
- 可以工作的软件是进度的主要度量标准。
- 提倡可持续开发。出资人、开发人员和用户应该维持不变的节奏。
- 对卓越技术与良好设计的不断追求将有助于提高敏捷性。
- 简单——尽可能减少工作量的艺术至关重要。
- 最好的架构、需求和设计都源自自我组织的团队。
- 每隔一定时间，团队都要总结如何更有效率，并相应地调整自身行为。



# 敏捷开发的核心原则

- 主张简单
- 拥抱变化
- 可持续性
- 递增开发
- 投资最大化
- 高质量工作
- 快速反馈
- 正确使用模型
- 软件终极目标
- 轻装上阵
- 个体和交互 胜过 过程和工具
- 可以工作的软件 胜过 面面俱到的文档
- 客户合作 胜过 合同谈判
- 响应变化 胜过 遵循计划



# 敏捷方法的适用性

通常可以在以下方面衡量敏捷方法的适用性：

- 从产品角度看，敏捷方法适用于需求萌动并且快速改变的情况，如系统有比较高的关键性、可靠性、安全性方面的要求，则可能不完全适合。
- 从组织结构的角度看，组织结构、环境设施要支持沟通谈判，人员彼此信任、高效能干；更适用于较小的队伍，如10人以下的小团队，大规模的敏捷软件开发不太适合。
- 有系列管理工具支持：版本控制整合、进度跟踪、工作分配、集成发布和迭代规划、论坛和软件缺陷的报告和跟踪等。
- 可与其他软件开发过程模型融合使用，如在目标确定情况下在一次迭代中使用瀑布模型开发实现可用软件交付物。



# 案例分析

## 从在线 **B2B** 公司角度思考

**Q:** 这个搜索框对公司的业务有什么价值？

**A:** 搜索框可以为用户方便得提供商户的目录信息，增加访问量。

## 从用户角度思考

**Q:** 作为查询信息、寻找商业合作伙伴的网站用户，搜索框好处？

**A:** 模糊查询功能

## 从程序员角度思考

**Q:** 一个搜索框的最简单实现方法是什么？

**A:** 一个有 **text input** 和 **search button** 组成的 **form**;

后台通过 **server** 程序将符合类型和地址的商户名从数据库中取出，返回给用户；每个返回项包括商户的名称、地址和评价意见。





# 用户故事

## 挖掘

**Q:** 搜索框如何在用户忘记具体名字的时候提醒用户？

**A:** 在第一版本中实现比较困难。可以让用户输入至少一个类型来提高模糊查找的效果。

## 思考

- 系统应该能够在高峰时候处理 **200** 条搜索请求和 **1000** 个鼠标点击事件；
- 用户可以在已经查找到的内容中继续查找；
- 系统提供一个商户类别清单，如果用户选择商户类别而忘记具体名字，系统提供模糊查询。



# 设计验收测试用例

动作	数据	期待的结果
搜索	一组能成功搜索到的 (类别, 位置) 数据	在该类别和位置条件下 的一组商户信息
搜索	一组不能成功搜索到的 (类别, 位置) 数据	空列表

当一个 **Sprint** 周期正式开始时，项目经理将制定该周期的具体开发和测试任务。当开发团队开始编码和单元测试时，测试人员的工作重点包括：估算验收测试的时间、估算测试框架的搭建、详细设计验收测试和编写验收测试代码。前两个主要活动一般在项目初期的 **Sprint** 周期中完成。其他的三个主要活动将在接下来的多个 **Sprint** 周期中视情况迭代进行。