实验3：小组分工讨论传统软件开发过程模型与敏捷开发（中几种主要方法）的比较，分析各自的优缺点，以及如何应用于自己的项目中？

**传统开发过程模型与敏捷开发的比较**

|  |  |  |
| --- | --- | --- |
| 软件过程模型 | 优点 | 缺点 |
| 瀑布模型 | 1. 为项目提供了按阶段划分的检查点 2. 当前活动完成后，只需要去关注后续活动。 3. 它提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。 | 1. 缺乏灵活性，不能反映实际代码的开发方式，必须在项目开始前说明全部需求。 2. 不能及时验证某一软件开发活动是否正确，最终产品知道最后一个步骤才出现，而软件客户无法在早期知道软件原型。 |
| 原型模型 | 1. 从认知论的角度看，原型方法遵循了人们认识事物的规律，因而更容易为人们普遍接受。 2. 原型方法将模拟的手段引入分析的初期阶段，沟通了人们的思想，缩短了用户和开发人员之间的距离。 | 1. 对于一个大型系统，如果不经过系统分析得到系统的整体划分，而直接用原型来模拟是很困难的 2. 对于原有应用的业务流程、信息流程混乱的情况，原型构造与使用有一定的困难 3. 对于一个批处理系统，由于大部分活动是内部处理的，因此应用原型方法会有一定的困难 4. 文档容易被忽略 5. 项目难以规划和管理 |
| 增量模型 | 1. 有利于增加客户对系统的信心 2. 降低系统失败风险 3. 提高系统可靠性，稳定性和可维护性 | 1. 增量粒度难以选择 2. 把每个新的增量构建集成到现有软件体系结构中时，必须不破坏原来已经开发出的产品 3. 容易退化为边做边改模型，从而是软件过程的控制失去整体性 |
| 螺旋模型 | 1. 支持用户需求的动态变化, 具有良好的可扩充性和可修改性。也支持软件系统的可维护性，每次维护过程只是沿螺旋模型继续多走一两个周期 2. 原型易于用户和开发人员共同理解需求，还可作为继续开发的基础，并为用户参与所有关键决策提供了方便。 3. 螺旋模型为项目管理人员及时调整管理决策提供了方便，进而可降低开发风险 | 1. 很难让用户确信这种演化方法的结果是可以控制的 2. 建设周期长，而软件技术发展比较快，所以经常出现软件开发完毕后，和当前的技术水平有了较大的差距，无法满足当前用户需求 |
| 传统模型 |  | 1. 过分强调了分阶段实施，使得开发过程各个阶段之间存在严重的顺序性和依赖性 2. 思维成果的可重用性很差 3. 忽视了人在软件开发过程中的地位和作用 |
| 敏捷开发模型 | 1. 采用简单计划策略，不需要长期计划和复杂模型，开发周期短 2. 在全过程采用迭代增量开发、反馈修正和反复测试的方法，能够适应用户经常变化的需求 3. 注重市场快速反应能力，客户前期满意度高 | 1. 注重人员的沟通，忽略文档的重要性，若项目人员流动大太，给维护带来不少难度 2. 对编码人员的经验要求高，若项目存在新手比较多时，老员工比较累 |

**敏捷开发模型应用到人才招聘系统中：**

我们在软件开发的过程中主要应用了敏捷开发模型的思想，它是一种灵活的软件开发方法，适用于快速变化的需求和项目。

1. 用户故事（User Stories）：将每个模块的功能需求转化为用户故事，明确描述用户的需求和期望。例如，“作为个人用户，我希望能够方便地修改简历信息。”

2. 迭代开发（Iterative Development）：将整个项目划分为多个迭代周期，每个周期都可以交付部分功能。可以先实现基本的功能，然后逐步完善和扩展。

3. 持续集成（Continuous Integration）：通过持续集成工具，确保团队成员的代码能够快速集成并自动化测试。这有助于发现和解决问题，并确保软件质量。

4. 原型开发（Prototyping）：在早期阶段制作简单的原型，让用户快速试用并提供反馈。根据反馈不断改进和调整设计。

5. 每日站会（Daily Stand-up）：团队成员每天定时举行站会，分享进展、遇到的问题和计划。这有助于团队协作、及时解决问题和调整计划。

6. 优先级排序（Prioritization）：根据用户需求和价值，对任务进行优先级排序。确保先实现最重要和最有价值的功能。