

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？并且分析自己项目中可能存在的风险，细化风险管理（做出风险分级及应对预案）。

我们小组在传统软件开发过程模型中决定采用增量模型这种传统软件开发过程模型，兼之以每日站立会议结对编程等多个敏捷管理实践。首先在传统软件开发过程中，分析各个传统软件开发过程模型的优缺点：

瀑布模型：

优点：1、软件生存周期的顺序性：只有前一阶段工作完成以后，后一阶段的工作才能开始；2、尽可能推迟软件的编码；3、保证质量：在每个阶段都要完成规定的文档，每个阶段都要对已完成的文档进行复审。

缺点：1、缺乏灵活性，不能反映实际的代码开发方式：必须在项目开始前说明全部需求；2、无法及时验证某一软件开发活动是否正确：最终产品直到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。

适用场合：1、当有一个稳定的产品定义和很容易被理解的技术解决方案时，纯瀑布模型特别合适；2、当你对一个定义得很好的版

本进行维护或将一个产品移植到一个新的平台上，瀑布模型也特别合适；3、对于那些容易理解但很复杂的项目，采用纯瀑布模型比较合适，因为可以用顺序方法处理问题；4、在质量需求高于成本需求和进度需求的时候，它尤为出色；5、当开发队伍的技术力量比较弱或者缺乏经验时，瀑布模型更为适合。

原型模型：

优点：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，因而更容易为人们所普遍接受；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。

缺点：1、对于一个大型系统，如果不经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。

适用场合：1、适合于那些不能预先确切定义需求的软件系统的开发；2、适合于那些项目组成员（包括分析员、设计院、程序员和用户）不能很好交流或通信有困难的情况。

增量模型：

优点：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。

缺点：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。

适用场合：1、进行已有产品升级或新版本开发；2、完成期限严格要求的产品；3、对所开发的领域比较熟悉而且已有原型系统。

螺旋模型：

优点：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。

缺点：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。

传统模型的缺点：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。

敏捷管理实践：

同时我们打算选用每日站立会议的敏捷管理实践，在每日工作前，我们的团队成员执行的例行沟通机制，在站立会议中选取一个 master 进行会议的组织和主持，其余 team 成员全员站立参加。就“我昨天为本项目做了什么？”“我计划今天为本项目做什么？”以及“我需要什么帮助以更高效的工作？”这三个主题，每人依次发言，进行会议讨论。优点是可以增加我们团队的凝聚力，产生积极的工作氛围，及时暴露风险和问题，促进团队内成员的沟通和协调。

除此之外我们也用到结对编程的敏捷工作实践，由两位程序员在一台电脑前工作，一位负责敲入代码，另一个实时监视每一行敲入的代码负责监视的程序员还需要考虑下一步的工作方向，比如可能出

现的问题以及改进等。同时两位程序员的工作经常性需要切换，确保平等协商和相互理解。其好处是有利于我们提升代码设计质量，同时提高总体效率并促进团队能力提升和知识传播。

采取敏捷开发的优点：我们采取简单计划策略，不需要长期计划和复杂模型，开发周期短。在全过程采用迭代增量开发、反馈修正和反复测试的方法，能适应我们软件用户经常变化的需求。

缺点就是注重人员的沟通，忽略文档的稳定性，若项目人员流动太多，给维护带来不少难度，对编码人员的经验要求高，新手比较多的时候，老手就很累。

风险评估：

风险评估，首先从案件场景模拟来分析软件项目风险。

（1）我们团队都在学校内部，不会出现离开项目组影响项目的进度不在这项风险。

（2）我们团队可能存在不科学不准确的计划制定，难以控制开发实践过程的进度

（3）同时也可能存在用户需求中的某项需求，我们实现不了，直接影响了软件项目的后续开发工作，影响到软件项目的成功完成。

(4) 因为不存在客户对接，也没有对应的风险危机。

然后对软件风险进行一项项列举对比，查看我们项目是否有对应风险。

(1) 我们不存在人员短缺问题

(2) 可能存在开发错误的软件功能的风险

(3) 可能存在开发错误的用户界面的风险

(4) 可能出现计划和实践差别明显

(5) 可能出现需求的持续变更风险。

(6) 可能出现外部执行的任务未达到要求的风险

(7) 构件未使用到，不存在其风险

(8) 实时性能可能达不到要求的风险

(9) 可能出现我们的计算机科学技术能力水平限制的风险。