1. 传统软件开发过程模型：
2. 瀑布模型。

瀑布模型需严格遵循预先计划的需求分析、设计、编码、集成、测试、维护的步骤顺序进行，是最典型的预见性方法。步骤成果作为衡量进度的方法，例如需求规格，设计文档，测试计划和代码审阅等。瀑布式的主要问题是它的严格分级导致的自由度降低，项目早期即做出承诺导致对后期需求的变化难以调整。

1. 迭代式开发。

迭代式开发（也称迭代增量式开发或迭代进化式开发），是一种与传统的瀑布式开发相反的软件开发过程，它弥补了传统开发方式中的一些弱点，具有更高的成功率和生产率。

每次只设计和实现这个产品的一部分,逐步完成的方法叫迭代开发，每次设计和实现一个阶段叫做一个迭代。在迭代式开发方法中，整个开发工作被组织为一系列的短小的、固定长度的小项目，即一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试。

采用这种方法，开发工作可以在需求被完整地确定之前启动，并在一次迭代中完成系统的一部分功能或业务逻辑的开发工作，再通过客户的反馈来细化需求，并开始新一轮的迭代。优点：1、降低风险2、得到早期用户反馈3、持续的测试和集成4、使用变更5、提高复用性

1. 螺旋开发。

将瀑布模型和快速原型模型结合起来，强调了其他模型所忽视的风险分析，更适合于大型复杂的系统。“螺旋模型”刚开始规模很小，当项目被定义得更好、更稳定时，逐渐展开。其的核心在于开始时只需定义最重要的功能，实现后听取客户的意见，再进入到下一个阶段。不断重复，直到得到满意的最终产品。螺旋模型是一种风险驱动的方法体系，因为在每个阶段之前及经常发生的循环之前，都必须首先进行风险评估。

1. 敏捷开发方法

敏捷过程模型的实例包括：极限编程、水晶方法、动态系统开发方法、精益开发、scrum等。

极限编程（XP）以客户的需求变化为重点，同时强调团队合作。XP更为注重开发过程中，团队（包括客户、管理人员、开发人员）不约而同地聚在一起讨论方案，解决难题；

Scrum的开发模式称为“橄榄球”式方法：团队成员都对产品开发的整个过程保留自己的看法，实现自治。产品开发方式由线性方法转向集成方法，这种转变刺激团队内部各层次间的交叉学习、交流以及思维发散。

功能驱动开发模式（FDD）主要针对中小型软件开发项目，“是一个以Architecture为中心的，采用短迭代期、目期驱动的开发过程。它首先对整个项目建立起一个整体的模型，然后通过两周一次‘设计功能——实现功能’的迭代完成项目开发。”

1. 极限编程XP。

极限编程是一个轻量级的、灵巧且严谨的软件开发方法，也是一种近螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。极限编程过程分为策划、设计、编码和测试四个阶段。

基于敏捷的核心思想和价值目标，XP要求项目团队遵循13个核心实践：

1. 团队协作：通过客户、开发团队、项目经理三方共同参加的会议来确定开发计划。
2. 规划策略： 计划是持续的、循序渐进的。每2周，开发人员就为下2周估算候选特性的成本，而客户则根据成本和商务价值来选择要实现的特性。
3. 结对编程：系统的每一行代码都是两个人用一个键盘完成的。
4. 测试驱动开发：先写测试，后写代码。
5. 重构：不断优化系统设计，使之保持简单。
6. 简单设计：为明确的功能进行最优的设计，不考虑未来可能需要的功能。
7. 代码集体所有权：开发队伍中任何人可以修改任何其他人的代码，代码不属于某个个人。
8. 持续集成：至少每天将整个系统集成一次，保持一个能运转的系统。
9. 客户测试：客户自己也是软件开发队伍的重要一份子。
10. 小版本发布：尽快发布，尽早发布。
11. 每周40小时工作制：保证休息，保持体力。
12. 编码标准：必须有统一的编码规范，确保代码的可读性。
13. 系统隐喻：将整个系统联系在一起的全局视图；它是系统的未来影像，是它使得所有单独模块的位置和外观变得明显直观。如果模块的外观与整个隐喻不符，那么你就知道该模块是错误的。
14. 水晶方法

水晶方法按照项目重要程度以及参加人员规模划分为透明水晶方法论、黄色水晶方法论、橙色水晶方法论以及红色水晶方法论。水晶方法把开发看作是一系列的协作游戏，而写文档的目标是帮助团队在下一个游戏中取得胜利。水晶方法的工作产品包括用例、风险列表、迭代计划、核心领域模型，以及记录了一些选择结果的设计注释。这些文档没有模板，描述也不太规范，但目标清晰，能够满足下次游戏开始的条件。不管是哪种水晶，都会对团队的角色、团队的工作项和产出、核心实践、支持过程等进行定义。一般来讲，透明水晶方法，适用于一个小团队来进行敏捷开发，人数在6人以下为宜。相比于同样适用于小规模团队的XP，水晶方法的纪律性较弱，但其管理运作与团队产出相协调。

水晶方法有七大体系特征：

1. 经常交付。敏捷方法对交付成果要求很高，注重频繁小批次交付，水晶方法也不例外。通过经常交付以及时获得客户、产品经理的反馈，从而提升客户价值，使产品价值最大化。
2. 反思改进。对在迭代开发过程中和在交付成果中发现的问题，团队及时的反思。把握问题的关键，快速找到解决方案。
3. 渗透式交流。在两个或多个成员进行交流的时候，与他们同处于一个空间范围内的其他人员会或多或少地获取他们的对话信息。这种接收并非有意创造的信息来源的方式即渗透式交流，成员根据自己的当前工作可以选择忽略或接收。
4. 个人安全。当个人产生问题困惑的时候，选择指出而非隐瞒问题，且自己的人身安全受到保障。
5. 焦点。确定首先要做什么，然后安排时间。确保团队成员了解自己最重要的任务，确保他们能够有充分的时间去完成这些任务。
6. 与专家、用户建立方便的联系。建立方便的联系是保证专家、用户、团队能够形成一个短周期反应链。对于小批次交付成果、用户需求变动等建立一个快速反馈机制，可以提高团队工作效率。
7. 自动化测试、配置管理和经常集成的技术环境。自动化测试可以对代码进行自动测试，减少人工成本，使工作变得高效、快捷。配置管理就是可以返回上一步，通过撤销新操作出现的失误来解决问题；经常集成能使团队对系统快速集成，以及时发现错误、纠正错误。

水晶方法保证了产品开发的进度，较好地满足了客户的需求，使开发过程顺利实施。

1. 动态系统开发方法（DSDM）

倡导以业务为核心，快速而有效地进行系统开发。可以把DSDM看成一种控制框架，其重点在于快速交付并补充如何应用这些控制的指导原则。DSDM不仅仅包括软件开发内容和实践，也包括了组织结构、项目管理、估算、工具环境、测试、配置管理、风险管理、重用等各个方面的内容。

DSDM实施的思路是，在时间进度和可用资源预先固定的情况下，力争最大化地满足业务需求（传统方法一般是需求固定，时间和资源可变），交付所需要的系统。对于交付的系统，必须达到足够的稳定程度以在实际环境中运行；对于业务方面的某些紧急需求，也必须能够在短时间内得到满足，并在后续迭代阶段中对功能进行完善。

DSDM的基本原则是1.活动用户必须参与。2.必须授权DSDM团队进行决策。3.注重频繁交付产品。4.判断产品是否可接受的一个基本标准是符合业务目的。5.对准确的业务解决方案需要采用循环和增量开发。6.开发期间的所有更改都是可逆的。7.基本要求是高层次的并区分优先级（以在低优先级的项目上获得一定的灵活性）。8.在整个生命周期集成测试。9.在所有参与者之间采用协作和合作方法。

1. 精益开发。

通过改良流程不断地消除浪费，对于IT系统建设，精益开发的常用工具模型是价值流模型。精益思想为IT项目的开发提出一系列的流程，培养技术队伍，运用最有效的技术和工具。

在人的方面，精益思想强调将每个员工的能力发挥到极限，不只是管理人，而是去培训人。精益思想不是着眼于结果，而是强调过程。

精益思想开发的首先是确定高效的IT软件开发流程。建立一套完整的开发流程，然后建立一套测量流程的手段，持续改进。首先了解客户需求，对工作流程进行细化分割，把流程分成更细微的步骤，并保证每个步骤都能满足客户的需求，增加价值。其次是流程的标准化和可操作化，在软件开发过程中，精益思想提到如何关注研发流程，让管理流程“落地”，并要让流程规范起来。

其次是引入首席项目主管负责制，将合适的人员安排在合适的岗位上，建立一个有效的软件项目开发组织。每个项目的开发团队确立核心领头人，首席项目主管在接手任务后，思考设计思路，并与团队交流。团队是推进精益管理的关键，应调动起每一个员工的积极性。

最后是用有效技术和工具的支持流程和人的工作。在引进新技术方面，不是积极倡导新技术，而是使用可靠的、已经过充分测试的技术。工具和技术的意义在于支持而不是驱动流程，是加强人的工作，而不是替代人。

1. Scrum。

Scrum 是一个用于开发和维护复杂产品的框架 ，是一个增量的、迭代的开发过程。在这个框架中，整个开发过程由若干个短的迭代周期组成，一个短的迭代周期称为一个Sprint，每个Sprint的建议长度是2到4周。在Scrum中，使用产品Backlog来管理产品的需求，产品backlog是一个按照商业价值排序的需求列表，列表条目的体现形式通常为用户故事。Scrum团队总是先开发对客户具有较高价值的需求。在Sprint中，Scrum团队从产品Backlog中挑选最高优先级的需求进行开发。挑选的需求在Sprint计划会议上经过讨论、分析和估算得到相应的任务列表，我们称它为Sprint backlog。在每个迭代结束时，Scrum团队将递交潜在可交付的产品增量。Scrum起源于软件开发项目，但它适用于任何复杂的或是创新性的项目。

Scrum以经验性过程控制理论作为理论基础的过程，采用迭代、增量的方法来优化可预见性并控制风险。其过程框架的基石包括透明性、检验和适应。Scrum的四大支柱包括迭代开发、增量交付、自组织团队和高优先级的需求驱动。

1. 传统开发方法与敏捷开发比较

传统的瀑布式开发，也就是从需求到设计，从设计到编码，从编码到测试，从测试到提交大概这样的流程，要求每一个开发阶段都要做到最好。特别是前期阶段，设计的越完美，提交后的成本损失就越少。

迭代式开发，以先搭建主要功能为目的，以最短的时间，最少的损失先完成一个不完善的成果直至提交，然后再通过客户或用户的反馈信息，逐步进行完善。

螺旋开发，很大程度上是一种风险驱动的方法体系，因为在每个阶段之前及经常发生的循环之前，都必须首先进行风险评估。

敏捷开发，其周期更短，且更加强调队伍中的高度协作。敏捷开发的过程中，更加注重人的因素。敏捷开发过程中，整个项目是测试驱动的而不是文档驱动的。不仅每个模块有着自己的相应的测试单元，开发人员在开发自己的模块的过程中,必须保证自己所开发的模块可以通过这一单元的测试，并且集成测试贯穿了整个开发过程的始终。

1. 传统开发方式的缺点与敏捷开发的优点。
2. 传统开发方式依赖于预设性而敏捷开发的过程有着更强的适应性。因为软件开发过程的本身的不可预见性，用户在项目开始时不可能对于项目有完整而明确的预期，而是在后期的修改和完善过程中产生。因此传统开发方式可能导致后期的损失成本很高，而敏捷开发的高适应性显然更加符合软件工程开发的实际。
3. 敏捷开发有以人为本的特性。在传统软件工程中，个人因素很少的被考虑到分工中，每个人都是只是整个代码开发机器的一个部分，个人的意志和创造力很大程度上被抹去。而在敏捷开发过程中，每个人的潜力被充分的考虑，应用什么技术很大程度上直接由在第一线开发的技术人员决定。每个人的特点和创造力都可以充分地发挥，这样开发出来的软件更加的具有生命力，开发者不再是进行机械的乏味的堆砌，而是创造属于自己的艺术品，这样的条件下产生的代码必然在质量上更占优势。
4. 轻量型的开发方法即以测试为驱动取代了以文档为驱动。敏捷开发集成测试每天会进行十几次甚至几十次，而不是像传统方法一样只有当各个模块的编码都结束了之后再进行联合调试。在软件开发的进程中每一点改动所引起的问题都容易暴露出来，使得更加容易在错误刚刚产生的时候发现问题从而解决问题。避免了在最后整个系统完成时错误隐藏的太深给调试造成极大的困难。
5. 传统开发方式的优点与敏捷开发的缺点。

传统开发方式从系统整体出发，强调在整体优化的条件下自上而下地分析和设计，保证了系统的整体性和目标的一致性。遵循用户至上原则，严格区分系统开发的阶段性，每一阶段的工作成果是下一阶段的依据，便于系统开发的管理和控制。文档规范化，按工程标准建立标准化的文档资料。敏捷开发，最重要的因素是项目的规模。规模增长，面对面的沟通就愈加困难，因此敏捷方法更适用于较小的队伍。敏捷开发在每次新的需求产生时都可能引起整个系统的大幅度的修改，因为开发者在开发上一个版本的时候，完全没有考虑以后的优化将要如何进行。这样的开发方式实际的软件开发过程中，并不一定总是有效的。敏捷开发因为缺乏很多在敏捷开发中被认为“不重要”的文档，在一个大型项目比如一个操作系统开发的时候，由于项目周期很长，很难保证开发的人员不更换，没有文档会造成在交接的过程中出现很大的困难。

1. 细化风险管理

风险分级管控程序分为四个阶段：危险源识别、风险评价、风险控制、效果验证与更新。

1. 风险评估
2. 技术风险，技术导致无法完成

软件的构造需要对软件构造过程中使用的各种技术进行评估。导致技术风险的原因可能包括缺乏培训、对方法、工具和技术理解不够、应用领域的经验不足和缺乏新的技术和开发方法应用等。

1. 工期风险，未及时交工

很多原因都可能导致项目延期。测试用例设计不完整，忽视了边界条件、异常处理等情况，用例没有完全覆盖需求；测试用例没有得到全部执行，有些用例被有意或者无意的遗漏；某些缺陷偶发，难以重现，容易被遗漏；软件代码质量差，导致缺陷较多，容易出现测试的遗漏；有些情况下测试环境与生产环境不能完全一致，导致测试结果存在误差；某些项目存在技术难度，测试能力和水平导致测试进展缓慢，项目延期；回归测试一般不运行全部测试用例，可能存在测试不完全；测试过程中涉及的角色较多，存在不同人员、角色之间的沟通、协作，难免存在误解、沟通不畅的情况，导致项目延期。

1. 需求不一致，交付物有问题

导致需求风险的原因可能包括对产品缺少清晰的认识、对产品需求缺少认同、没有优先需求、在做需求中使用者参与不够、需求不断变化、缺少有效的需求变化管理过程、对需求的变化缺少相关的分析

1. 解决措施
2. 技术风险

组织组内的培训会、向老师和同学请教、从网上搜集资料。

1. 工期风险

准备必要的备用方案，制定好人员分工计划，确保各项工作有序进行。

1. 需求风险

确定敏捷开发的模式，小周期迭代，以适应不断地变化。