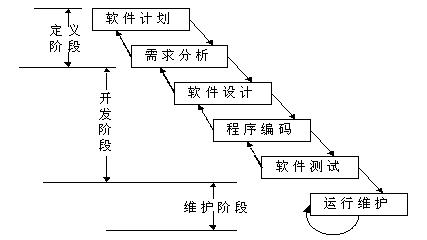
软件的测试工作和软件的开发模型是息息相关的，所以不同的软件开发模型中，测试的工作是不容得，因此我们要充分了解软件的开发模型，以便于找到自己的任务。所以我们小组对这一部分进行了讨论，得到以下结论。

软件开发模型在整个软件开发中是十分重要的，也是整个软件开发周期的导航图，有了它我们能够比较直观的表达软件开发过程中的全过程，以及每个阶段的活动及其任务。并且开发人员在选择开发模型的时候，要根据软件的特点，开发人员的参与方式选择比较稳定可靠的开发模型。到现在软件开发模型已经从最开始的“边做边改”发展出了很多模型。

**1. 瀑布模型**

瀑布模型是W.W.罗伊斯(W.W.Royce)于1970年提出的软件开发模型，由模型名称我们就可以得知该模型遵循从上至下一次性完成整个软件产品的开发方式瀑布模型将软件开发过程分为6个阶段:计划→需求分析→软件设计→编码→测试→运行维护，开发过程如下所示。



直到80年代的早期，瀑布模型还是唯一被广泛采用的开发模型。在瀑布模型中开发软件的各项活动要严格按照线性方式进行，当前活动要接受上一活动的工作结果，然后实施所需的工作内容，当前活动的结果也需要进行验证，若验证通过，则作为下一活动的输入，继续进行下一项的活动，否则需要返回进行修改。

**特点：**

瀑布模型的优点是严格遵循预先计划的步骤顺序进行，线性进行一切按部就班比较严谨。

瀑布模型还强调文档的作用，并且要求每个阶段都要仔细验证，但是这种模型的线性过程太理想化，已不再适合现代的软件开发模式，几乎被人们抛弃，其主要问题在于：

1)各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量;

2) 由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发的风险;

3)早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果。

**2、迭代式开发**

迭代式开发也被称为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。它将一个软件拆分为不同的部分，然后不同组件进行开发测试。在迭代式开发中，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，被称为一系列的迭代，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。在迭代模型中，第一个迭代通常是软件需求的核心部分，第一个组件完成后在进行下一组件的开发计划，这样重复迭代步骤直到整个软件的完成。

迭代式开发的优点如下：

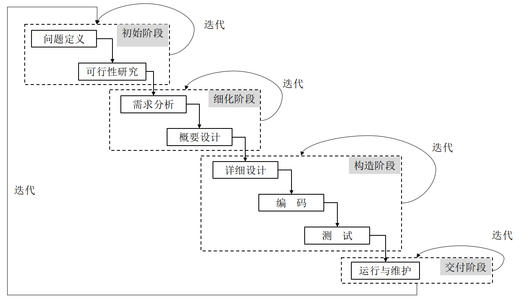
1)需求更改则可能只需要改变一个组件，降低风险

2)逐步交付，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

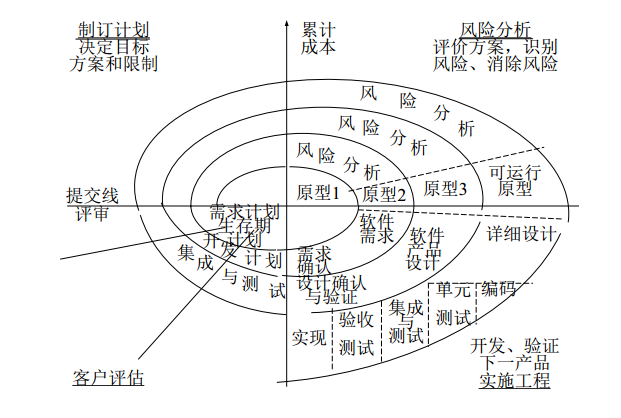


但是迭代开发需要将开发完成的组件集成到已完成的软件体系结构中，这样容易有集成失败的风险，所以要求开发的软件必须有开放的体系结构。而且，迭代模型是逐个组件的完成开发，容易造成“边做边改”的开发形式，这样有可能失去对软件开发过程的掌控。

**3、螺旋式开发**

1988年，巴利·玻姆(Barry Boehm)正式发表了软件系统开发的“螺旋模型”，它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，如果项目不能排除重大的风险，则可以把项目停止，从而减小项目的损失，所以更适合于比较大型且复杂的系统。其核心在于不需要在刚开始的时候就把所有事情都定义的清清楚楚，而是在定义最重要的功能时去实现它,然后听取客户的意见,之后再进入到下一个阶段.如此不断轮回重复,直到得到你满意的最终产品。

螺旋模型将项目开发的过程分为几个不同的阶段，每个阶段要按部就班的进行，这样的划分和瀑布模型是差不多的，借鉴了瀑布模型。并且每个阶段执行之前要先进行安全性分析也就是风险评估，若有重大风险则要停掉项目，若没有重大的风险，才可以执行这一阶段。



螺旋模型按照螺线进行若干次迭代，我们发现图中有四个象限。分别是代表了如下活动：

1) 制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件;

2) 风险分析：分析评估所选方案，考虑如何识别和消除风险;

3) 实施工程：实施软件开发并验证;

4) 客户评估：评价开发工作，提出修正建议，制定下一步计划。

在螺旋模型中，每一个阶段都是要经过这四个步骤，然后才能得到完善的产品，进行提交。因为螺旋模型进行了风险分析，这就说明可选方案和限制条件也进行了评估，这有助于提高软件质量提升使用感受。并且螺旋模型用小分段构建大型软件，这使得计算成本小很多，而且客户可以始终参与到软件的开发，保证了项目不会偏离原来的方向，也保证了整体项目的可控性。

**4、敏捷模型**

敏捷开发（敏捷软件开发），是从1990年代开始引起关注的新型软件开发方式，它是一种以人为核心、进行迭代循序渐渐的开发方法。相对于传统的软件开发方式，更强调程序员团队与业务专家之间的紧密协作、面对面的沟通、并且需要频繁交付软件版本、紧凑而自我组织型的团队。可以更好地适应需求变化的代码编写和团队的组织方式，也更注重于软件开发过程中热的作用。当客户需求变更的时候，敏捷模型可以迅速的对某个子项目进行修改来满足客户的需求，并且在这个过程中软件是一直可以使用的。

除此之外，敏捷模型还有一个重要的概念，那就是迭代。也就是对产品进行细微的改进，每次修改一部分，若可以则逐步扩大改进范围。敏捷模型中，软件开发不是线性的，开发的同时也可以进行测试工作，甚至可以提前写好测试代码。敏捷模型也描述了一套软件开发的价值和原则。

(1)个体和交互重于过程和工具。

(2)可用软件重于完备文档。

(3)客户协作重于合同谈判。

(4)响应变化重于遵循计划。

对于敏捷模型来说，人与人之间的交流比工具文档更加重要，当然，工具、文档也是很重要的一部分。

**传统开发与敏捷模型的对比**

**1、相对于瀑布模型**

敏捷模型和瀑布模型的特征都很明显。那就是敏捷开发是以一种迭代的方式推进的，而瀑布模型则是典型的预见性的方法，需要严格遵守计划的步骤顺序进行，步骤则成为了衡量进度的方法。敏捷开发中软件则是一直可以使用的，讲项目分成若干个相互联系、可以独立运行的子程序，所以每阶段软件都是可见的。

**2、相对于迭代模型**

相同之处就是都要在较短的开发周期内提交软件。不同的是敏捷开发的周期可能会更短，并且强调队伍中的高度协作，认为人的作用更加重要。需求是会变得，人们对需求的理解也是变化的，项目环境也在不停的变化，所以开发方法要反应这种现实，敏捷开发就是这种适应性的开发方法，而不是预设性的。

而且重视交互也是一种明显的区别，敏捷开发比较适合于小型的团队，这有助于团队之间的交互，也强调用户与团队一起工作，这有助于及时沟通交流。

**3、相对于螺旋模型**

不同于螺旋模型，敏捷开发更注重适应性而不是非预见性。螺旋模型不需要再开始的时候就把所有的事情全部定义清楚，而是知需要定义最重要的功能，并且付诸实现，之后再听取客户的意见，在进行下一个阶段。一直迭代得到最终的产品。螺旋模型更像是一个风险驱动的方法，因为在每个阶段之前都要进行风险评估，若没有重大风险才可以开始。而敏捷开发强调的则是适应性，而不是开发过程中不可预见性，适应性方法集中在快速适应现实的变化。当项目的需求发生变化的时候，整个团队都需要进行快速的适应。

**总结：**

敏捷开发是一个过程，而不是一个事件。在敏捷开发的过程中集合了很多软件开发的方法，也就是在原有的方法之上取其精华去其糟粕，所以敏捷开发继承了原有方法的很多优点，这也是敏捷开发延续至今的原因。

**如何应用于自己的项目中？**

**1.沟通很重要**

无论是在工作还是在日常生活中，沟通对我们来说，都是非常重要的交流与获取反馈的有效途径，在项目管理中，尤为重要。若沟通不良，对项目造成的影响是具大的，比如没有向客户询问该问的问题，而导致程序在关键性设计中出现失误。管理人员不向开发人员问该问的问题，而导致项目进度误报。

在项目管理中，沟通是获取客户需求的关键，是程序设计的依据，是制定项目计划的参考。所以我们要促进良性沟通，避免不良沟通。

**2. 简单设计**

有三条准备可以避免软件项目中常见的戏剧性效果和机能障碍。

（1）在项目初期不可能收集到所有需求。

（2）不管你收集到什么需求，最终它们肯定会发生变化。

（3）总会有任务超时、超支。

所以在设计的时候要尽量的简单，避免过度设计

**3. 每周交付一些有价值的东西**

大部分情况下，客户不能确切的告诉你他们需要什么，一旦客户见到第一个版本，他们就知道了在第二个版本中他们需要的东西……或者说他们在第一个版本中实际想要什么。所以我们有必要每周都交付一些可用的东西。

这样做的好处是：

（1）及时反馈：可以及时获取用户反馈，通过频繁的与用户沟通、反馈，形成良性沟通，明确用户真实需求。

（2）保障项目进度：短周期的交付，便于项目的调整，减少用户意途与开发之间的理解误差。便于项目进度的掌控。

（3）给用户和开发人员信心：由于每周都能看到可用或可演示的东西，让用户对项目有信心，也可以了解项目进度，这要比一个月交付的一次更有说服力。同时对于开发人员，每周的功能都能得到用户的确认，若需要调整，也是小幅度调整，若不需要调整，则是对工作内容的肯定与激励，可以提高工作激情。

**4. 控制范围**

一般在项目合同中，均会对时间、范围进行约束，质量就不用说了，如果质量不合格，用户是不会验收的。对于成本和时间基本是不可能有大的改变的，对于范围，合同中不可能对边界定义的十分明确，这就需要我们在整个项目进展中进行有效控制。客户在开发的过程中，会将范围无限的扩大，如果项目经理不对范围进行控制，这个项目会不断的扩展，造成成本增加，开发周期变长，无法在合同规定内完成项目的验收,所以说范围的控制最为有价值。

关于质量和时间，牺牲质量以换取时间的做法是不可取的，这种短暂的收益在项目后期将导致致命的伤害。所以要平衡这两者的关系，使得开发人员可以在有限的时间内开发出高效的产品。

我们团队主要从一下几个方面应用敏捷开发:

(1). 四个变量

成本、时间、质量和范围，其中范围的控制最有价值。

(2). 四个准则

沟通、简单、反馈、勇气

(3).基本原则

快速反馈、假设简单、递增更改、提倡更改、优质工作。

(4) 开发软件的四项基本工作

编码、测试、倾听、设计