个人工作报告

201705130112 张烜

本周调研了常见的软件开发模型，以及与发布方沟通了软件细节以及需求的变化。

在开发模型方面，我调研了常见的十大模型，分别是

1.边做边改模型（Build-and-Fix Model）；

2.瀑布模型（Waterfall Model）；

3.快速原型模型（Rapid Prototype Model）；

4.增量模型(演化模型)（Incremental Model）；

5.螺旋模型（Spiral Model）；

6.喷泉模型(fountain model)；

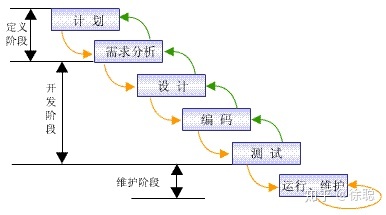
7.智能模型(四代技术（4GL）)；

8.混合模型（hybrid model）；

9.RUP模型；

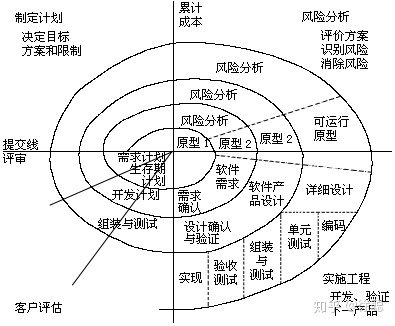
10.IPD模型

在这里我详细的介绍一下瀑布模型。1970年Winston Royce提出了著名的"瀑布模型"，直到80年代早期，它一直是唯一被广泛采用的软件开发模型。瀑布模型的流程图大致如下：



瀑布模型中，如图所示，将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。在瀑布模型中，软件开发的各项活动严格按照线性方式进行，当前活动接受上一项活动的工作结果，实施完成所需的工作内容。当前活动的工作结果需要进行验证，如果验证通过，则该结果作为下一项活动的输入，继续进行下一项活动，否则返回修改。瀑布模型强调文档的作用，并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，几乎被业界抛弃，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。 我们应该认识到，"线性"是人们最容易掌握并能熟练应用的思想方法。当人们碰到一个复杂的"非 线性"问题时，总是千方百计地将其分解或转化为一系列简单的线性问题，然后逐个解决。一个软件系统的整体可能是复杂的，而单个子程序总是简单的，可以用线 性的方式来实现，否则干活就太累了。线性是一种简洁，简洁就是美。当我们领会了线性的精神，就不要再呆板地套用线性模型的外表，而应该用活它。例如增量模 型实质就是分段的线性模型，螺旋模型则是接连的弯曲了的线性模型，在其它模型中也能够找到线性模型的影子。

而另一个比较常见的是螺旋模型，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，特别适合于大型复杂的系统。



瀑布模型适合需求明确或很少变更的项目，如二次开发或升级型的项目；新项目与过去成功开发过的一个项目类似，因为已经有了以前开发的经验和积累的软件模块，这些都可以应用到新项目中，因此，应该使用瀑布模型进行项目开发。

螺旋模型强调的是风险分析，适合大型系统的开发，而如果开发过类型项目，说明风险并不是很大的，而且这个的规模更大并不代表项目就是大型项目。

而在软件开发时候，总会遇到需求方变更需求，变更在项目中不可避免，但是频繁的变更在市场的情况下，总是让我们这些乙方苦不堪言。

首先在项目启动的时候，会有一个需求变更机制，当需求变更时，是需要相关干系人决策的；另外，用户需求发生变更时，首先考虑是否合理，变更成本，是否紧急，不紧急排在后面；如果需求变更不合理，就需和用户协商，告诉用户哪里不合理。当用户不断发生需求变更时，想一下是用户自己根本不知道自己的真正需求是什么，还是我们理解不到位给用户的东西不是他想要的？所以现在需要挖掘用户变更需求的原因，为什么变更，引导用户挖掘自己真正的需求。

不管怎样，还是要理清导致变频繁更的根本因素是什么？

1、投标中为了拿下项目，相关人员向客户许下很高的系统功能承诺；

2、应标中，为了符合客户的要求，在技术协议中承诺一切都可以做到；

3、中标后，技术协议没有明确描述乙方能够做到的功能，含糊不清，没有界定；

4、实施中的需求调研分析，没有摸清客户的需求；

5、实施中，对客户变更需求的引导不理想，甚至失败；

6、因为商务的因素，导致项目的需求不断变化，乙方也不断修改满足客户；

7、最可怕的就是所谓市场的占有因素，导致项目的被动，频繁的被动接受变更。