

谈软件生命周期模型及其选择 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/141/2021_2022__E8_B0_88_E8_BD_AF_E4_BB_B6_E7_c29_141285.htm 瀑布模型/改进的瀑布模型 虽然瀑布模型仍然存在很多的问题有待解决,但瀑布模型仍然是最基本的和最效的一种可供选择的软件开发生命周期模型.瀑布模型要求软件开发严格按照需求->分析->设计->编码->测试的阶段进行,每一个阶段都可以定义明确的产出物和验证准则.瀑布模型在每一个阶段完成后都可以组织相关的评审和验证,只有在评审通过后才能够进入到下一个阶段. 由于需要对每一个阶段进行验证,瀑布模型要求每一个阶段都有明确的文档产出,对于严格的瀑布模型每一个阶段都不应该重叠,而应该是在评审通过,相关的产出物都已经基线后才能够进入到下一个阶段. 瀑布模型的优点仍然是可以保证整个软件产品质量较高的质量,保证缺陷能够提前的被发现和解决.采用瀑布模型可以保证系统在整体上的充分把握,使系统具备良好的扩展性和可维护性.但对于前期需求不明确,而又很难短时间明确清楚的项目则很难很好的利用瀑布模型.另外对于中小型的项目,需求设计和开发人员往往在项目开始后就会全部投入到项目中,而不是分阶段投入,因此采用瀑布模型会导致项目人力资源过多的闲置的情况,这也是必须要考虑的问题. 很多人往往会以进度约束而不选择瀑布模型,这往往是一个错误的观点.导致这种情况的一个关键因素往往是概念需求阶段人力不足.因此在概念需求阶段人力能够得到充分保证的情况下,瀑布模型和迭代模型在开发周期上并不会存在太大的差别.反而是很多项目对于迭代或敏捷模型用不好,为了赶进度在前期需求不明确,没

有经过一个总体的架构设计情况下就开始编码,后期出现大量的返工而严重影响进度. 架构设计是软件开发中一个重要的关注点.因此在RUP中也提及到软件开发要以架构为核心.因此在架构设计完成后系统会被分为相关的子系统和功能模块.每个功能模块间的接口都可以定义清楚.在这种情况下,当模块B的详细设计做完成后往往就没有必要等到其它模块的详细设计都要完全作完才开始编码,因此在架构设计完成后可以将系统分为多个模块并行开发,每个模块仍然遵循先设计和编码测试的瀑布模型思路.这是瀑布模型的一种最重要的改进思路,也可以说这是一种增量开发的模型. 当一个新系统的开发存在多个完全不相关的独立需求的功能开发的时候,这个时候也可以选择将整个开发过程按独立的需求来分为多个小瀑布进行操作. 这种方式的最大问题就是没有一个完全总体的设计,架构设计人员无法在洞悉了所有需求后从系统的可扩展性,复用等方面总体规划. 在项目管理中有一种压缩进度的方法叫赶工,因此瀑布模型的另外改进处就在适当的重叠各个阶段过程,达到资源的有效利用.比如我们通过讨论,会议确定的实现方式就可以开始执导下一个阶段的工作而不一定完全等到相关的交付物文档化出来. 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 www.100test.com