

从瀑布型开发到迭代型开发的转变 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__E4_BB_8E_E7_80_91_E5_B8_83_E5_c29_141284.htm 本文来自 Rational Edge：一个理想的迭代开发方法模型在很多方面与理想的瀑布开发模型有着根本上的不同。但是，从实际来说，没有一个团队严格的应用了每一种开发方法模型。本文解释了为什么开发团队决定逐步的从类似瀑布型的开发方法转变成更加类似迭代开发的方法，同时也概述了能够帮助这种转变的步骤。多数的软件开发团队仍然在开发项目中使用瀑布型的开发过程。采用极端的瀑布型开发方法意味着你要以严格的顺序来完成一系列的项目阶段：需求分析、设计、实现/集成然后是测试。当项目中出现的问题解是困难的并且解决问题是昂贵时，你可能会推迟测试直到项目周期的末端；这些问题也能够严重的威胁软件发布的期限并且使主要的团队成员在某些开发环节上是空闲的。实际上，多数的开发团队使用了改进了的瀑布型开发方法，他们将项目分解成为两个或者更多的部分，有时这些部分被称为阶段或者是时期。这种改良可以帮助简化集成、使测试人员更早的进行测试工作和提供更早的项目状态的观测。这种方法也将代码分解成了易于管理的片断并最小化了以存根和驱动程序形式的、被测试需要的代码集成。此外这种方法允许你原型化你认为有风险的或者有难度的部分，并且使用来自每一个阶段的反馈修改你的设计。然而，使用瀑布型开发方法的执行与想象是相反的：很多设计团队把在阶段 1 之后的修改设计视为他们的最初设计或者需求过程的失败。虽然一个改进了的瀑布型开发方法

并不排除反馈的使用，但是它并没有促进、支持和鼓励反馈的使用。最后，想要最小化风险就不要典型的驱动一个瀑布型的开发项目。对于软件开发过程来说，本文探索了“迭代”开发方法超越传统的瀑布型开发方法的进步。迭代开发的好处相比较而言，迭代开发方法以IBM的Rational统一过程reg.为例包括了一系列的增量的步骤或者迭代。每一个迭代都包括一些或者很多的开发活动（需求、分析、设计、实现等等），就像你在图1中看到的那样。每一个迭代也有一系列良好定义的目标并生成最终系统的部分工作实现。每个后续的迭代都建立在前一个迭代的基础上以使系统得到发展和细化，直到最终产品被完成。早期的迭代着重于需求、分析和设计；后期的迭代着重于实现和测试。图1: RUP的迭代开发。每一个迭代都包括需求、分析、设计、实现和测试活动。同时，每个迭代都建立在前一个迭代工作的基础上，每一次迭代都会生成更加接近最终产品的可执行版本。迭代开发方法已经证明了自己优于瀑布型的开发方法，理由如下：它允许需求的变化需求的变化和“进一步的蔓延”技术和客户驱动的特性的累加一直是项目中导致麻烦、延期交付、令客户不满意和使开发人员泄气的主要原因。为了解决这些问题，使用迭代开发方法的团队应该在项目开发的几周里就关注生成和演示可执行的软件，这样就强制了需求的检查并可以帮助减少需求从而反映系统的本质。集成不是在项目的尾声进行的“大动作”将系统的集成留到项目的结尾几乎总是会导致耗时的返工有时这种返工会花费整个项目工作量的百分之四十的时间。为了避免这种返工，每一次迭代都以集成构建系统各部分结束；这样不断的积累将最小化日后的返工。早

期的迭代可以暴露风险 迭代的开发方法可以帮助团队在早期的迭代中减少风险，因为在这些迭代中包括了对所有过程组件的测试。当早期的迭代覆盖了项目的很多方面时 工具、购买的软件和团队成员的技能等等 团队能够很快的发现被预感的风险是否是真实的，并且能够在问题相对容易并花费很少成本解决时揭示没有被发现的新的风险。 对产品的管理能够采取战术性的变化 迭代开发能够快速的生成可执行的架构（虽然功能有限），这个架构能够为了应对竞争对手的快速版本发布容易的调整产品使之成为”轻量级的“或者“改进的”版本。 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