

计算机软件水平考试：RUP是敏捷的吗？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_AE_A1_E7_AE_97_E6_9C_BA_E8_c29_141328.htm

Q：我的经理引入了RUP，告诉我们使用它，他说RUP是敏捷的，但是对我来说它看起来是瀑布型，能给我一些怎样继续下去的建议吗？

A：在回答这个问题之前，我想简单说明一下一个人为什么会建议使用RUP。简单说，RUP是由一组最佳实践组成的迭代的和适应性过程，例如迭代开发和配置管理。进一步说，它是灵活的和面向结果的。在它的主要结构中，采用了很多有用的实践，诸如DSDM，Evo，Scrum，和XP。同样的，它也是得到广泛验证的流行的迭代方法，上万个组织采用了它。最后，它提供了通用的软件开发词汇。例如，RUP定义了象版本和风险列表这样的制品。那些采用了RUP的组织在软件开发活动中使用相同的术语，这增进了交流。几年前，RUP的架构师Philippe Kruchten和我写了一篇文章《How to fail with RUP:7 step to pain and suffering》，其中总结了一些敏捷 - 毁灭的缺陷我们看到的RUP的使用者或顾问所陷入的一些缺陷，下面是一些陷阱：陷阱：将瀑布模型的概念或阶段划分和RUP相对应。瀑布模型或者顺序的生命周期意味着首先尝试去理解和记录尽可能多的需求，然后设计规格说明和模型，然后再构建系统。研究表明瀑布模型是一种高风险和极易失败的过程。它的作用在过去的10年里被误传了。RUP并没有沿用瀑布模型的顺序的开发周期，相反，它是迭代和不断改良的过程，就象DSDM，Evo，Scrum，XP一样。也就是说，我们在快速构建软件的过程中不断精化需求和设

计。我们快速开始编程，例如，当我们只理解了10%的需求的时候。事实上，RUP提出了6个最佳实践，第一项就是迭代开发。一个RUP项目应该以2-6周为一个迭代周期来持续改进系统。任何一个感觉象“首先要整理出大量的需求”或者“在开始编码前写大量的规格说明书”的RUP项目都是对RUP的滥用，例如：将RUP当作瀑布模型，或者被所谓的专家建议采用连他们自己都不理解的过程。另一种常见的以瀑布模型来看待RUP的想法是错误的将瀑布模型的各开发阶段（需求，设计等）和RUP中的开发阶段画上等号。RUP将每次迭代分为四个更小的阶段：初始，细化，构建，移交。对于任何将开始和需求，精化和设计，构建和编码画上等号的观点都是错误的。这里不打算对每个阶段都作出详细的说明，只给出一些简要的说明：1. 初始：这一阶段，我们开发系统的版本和商业用例，发现一些小的但是很重要的需求，以及关键的风险，确定哪些要留到细化阶段。在开始阶段我们只需要发现“Top-10”的高级别的需求。通常，这个阶段非常短，也许只有几个小时或几天，否则将会陷入到瀑布模型的“预先定义”的陷阱中。在初始阶段不包括对项目的整体评价，主要是为下面的细化阶段定义范围和目标。2. 细化：迭代地构建核心架构和解决项目的高级别的风险。这里所说的构建核心架构指编程，整合，测试，而不是指“纸上”的练习或者构建将会被抛弃的原型。当我们在并行开发系统的核心部分同时反复地发现需求的细节。在这个阶段需求会发生显著的变化，经过反馈 - 适应的过程，变化能够得到有规律的评估和实现。注意这和传统的瀑布模型的需求定义形成鲜明的对比，多数的需求在并行开发核心架构的时候会被细化，并

从实际的开发中得到大量反馈。例如，在细化阶段，每个为期2周的迭代周期内，都有可能1-2天的时间进行对需求的讨论。在一些较大型的项目中，这个迭代周期可能为2-4周。在细化阶段的最后，开发的成果物被提出，这时再决定是否进行构建。

3. 构建：迭代地构建在细化阶段没有完成的部分，迭代进行集成，质量评估，并准备部署。需求变更在这一阶段会比较少，因为需求在细化阶段已经被精练，细化了。

4. 移交：完成beta测试，发布，部署系统。

陷阱：创建过多的工件 RUP是一个过程的FrameWork，或者说是一组可选的实践和工件，可以根据项目的实际情况进行裁剪，也就是说“对症下药”。在这个前提下，RUP定义了一组大约40个可选工件，例如：版本，风险列表，发布说明。通常的“过程裁剪”的原则是“越少越好”，而且尽可能少得创建工件（两个或三个）。大多数项目都会从简短的版本和风险列表中收益，很多的其他工件都可以被忽略掉。而敏捷RUP更提倡尽早开始编码和创建软件的基础部分。

陷阱：像大规模制造业的过程那样使用RUP RUP不仅仅定义了一组工件，还包含一组可选的活动，例如整合子系统。很明显，一个团队不需要学习很多的细节就能够决定他们应该采用哪些活动。如果一个人学习，并尝试在项目中严格实行这些可选活动，就像生产一辆汽车的制造过程，就是对RUP的一种滥用。软件开发是“新产品的开发”，而不是重复的制造，需要灵活性，创造性来应对来自构筑软件的挑战，不是一张按部就班的处方。RUP是灵活的，并且鼓励采用来自于所有软件开发方法学中的技术，例如，你可以采用Scrum的实践中提倡的实行每天的“站立会议”。而不是遵循RUP中的“定义”式的过程

。 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