



developerWorks 中国 技术主题 Rational 文档库

# 使用 RUP 管理小型项目和团队

来自 **Rational Edge**: 软件项目管理者常常认为 **Rational Unified Process** (即大家所熟知的RUP), 不适用于有限规模的软件项目。本文提供了在整个迭代开发阶段均遵循RUP, 从而受益匪浅的两个小项目的典型示例。

David Kohrell, Technology As Promised, LLC, ([www.aspromised.com](http://www.aspromised.com))总裁和 TAP University (<http://tapuniversity.aspromised.com>)。他是位于 Omaha, Nebraska的Bellevue大学项目管理方向的一位教授。作为IBM Scholars 网络的一员, 他曾在几个组织中负责产品开发、软件交付、网络基础设施和过程工程项目方面的领导、培训和咨询。他获得管理 M.A., MIS, 社区和地区计划 M.A., Nebraska 大学的 B.S.。他通过了项目管理协会的项目管理专家资格认证。此外, 他还通过 MS Project 2000 认证, 成为 Microsoft Solutions Framework Practitioner 和 Microsoft Certified Programmer。

Bill Wonch, Instructor, Technology As Promised, LLC([www.aspromised.com](http://www.aspromised.com)), Nebraska 州劳工开发部门的一名软件架构师。他从 Oklahoma 的 Rogers State University 获得 Associates 学位。他在软件工具的使用方面有丰富的经验, 包括 Cold Fusion、PHP、Rational Product Suite、UML、WebSphere、DB2、MS SQL、Crystal、ASP和XML。

2005 年 10 月 19 日

David Kohrell 在2005年2月的 Rational Edge 期刊上指出, *Rational Unified Process*, ® 或者称 *RUP*, ®为项目的推进提供了一个灵活的过程 -- 从先启阶段, 经过细化阶段、构建阶段, 以及产品化阶段 -- 给予指导和说明。本文特别关注 *RUP*如何同样能为小项目和团队提供指导。另外, 在用于敏捷开发环境的能力方面, 我们也观察了*RUP*和其它指导 (比如, 项目管理协会的项目管理知识体系, 或*PMBOK*®)。



在 IBM Bluemix 云平台上  
开发并部署您的下一个应用。

[开始您的试用](#)

## 小型项目和团队的背景

通常看来, 如果被安排来管理一个小项目, 也就意味你是新人或者你已经落伍了。大家都认为“一流的资源”应该被分配给大型的、企业级的、全特性的发布项目。这种认识是错误的, 让我们来看一下市场, 特别是2001年 .com 破碎之后, 小型项目和敏捷团队的时机成熟了。公司在一个月、一个季度、或者一年之内完成的项目越小, 那么, 产生收益、减少成本、或者拓展品牌和价值的机会就越多。

明确以下一些定义之后, 我们继续这个话题的讨论:

大型项目: 预算超过\$500,000, 团队规模为十三人或者更大, 项目进行时间超过一年。

中等项目: 预算\$100,000-\$500,000, 团队规模为六到十二人, 项目进行时间为六个月到一年。

小型项目: 预算低于\$100,000, 团队规模少于六人 (包括在该项目和其他项目之间共用的团队成员, 以及每日必须的人员)。项目进行时间少于六个月。

变更请求: 预算低于\$50,000的所有任务都是被一个人在几周之内来完成。

## RUP同样适用于小型项目

在 Michael Jordan、Greg LeMond、Tiger Woods之前, Bo Jackson统治着整个体育世界。19世纪80年代后期流行着这样一句话: “Bo 懂得篮球、足球、投资”。

过去的三个多月里, 在研讨会或课堂上, 我引用Bo Jackson的例子来反驳RUP“不适”小型项目的错误观点。我认为RUP“适合”于所有类型的项目, 这让很多人都感到惊诧。就我在过去几年使用RUP的经历而言, 它能够用在所有大型、企业级项目, 并且组织变更请求。它不仅仅是一个可有可无的方法论。

下面是人们经常提起的用来说明“RUP不适用于小型项目”的两个方面，我将逐一解答这些问题，来证明他们的看法是错误的：

1. 敏捷方法考虑到迅速和紧密的增加或者阶段；减少开销；并且确保开发人员与客户之间的紧密联系。

我的回答：敏捷方法以及类似的方法（**SCRUM**, **Paired Programming**）在软件构建中是革新的、有用的。然而，在**RUP**中也可以使用敏捷方法。那些轻量级的方法可以很好地在新系统的构建阶段、解决方案，或者程序中得到运用；但是仍然需要管理其它三个阶段的上游和下游活动，比如决定需要做什么（需求）以及操作环境将受到什么影响（发布管理）。**RUP**并不关注先启阶段、细化阶段、构建阶段和产品化阶段所有业务原则的使用，事实上，它是为这些活动提供了一个最佳框架。

2. **RUP**以及类似的指导，比如**PMBOK**, 软件工程协会(**SEI**)的集成的能力成熟度模型 (**CMMI**)，或 **UK** 的 **IT Infrastructure Library (ITIL)**标准给小型项目强加了一些不必要的过程。他们其实仅适用于一千万以上的大型项目。

我的回答：方法、知识体系，或者成熟模型不会强加过程。他们只为估算需要做什么，以及如何做得更好而提供一定的基础。“如何做”这部分是由实施组织来决定的。

**PMBOK**并没有规定2000版本中的39个过程或者2004版本中的44个过程在项目中都必须得到使用。它是一个知识体系，为项目管理者可能遇到的各种情况提供了一个起点。例如，它有助于定义组织的变更控制过程应该包括哪些内容。现在，项目管理专业人员（**PMP®**）在项目管理协会（**PMI**）监督之下，当然必须遵循**PMBOK**。**PMI**提供**PMP**资格认证，这样，聘用专业人员的组织机构就能够放心该专业人员懂得**PMBOK**。但是这并不意味着专业人员必须在每个项目中都使用到**PMBOK**的每一项知识。

**SEI**的能力成熟度模型（**CMM**）和**CMMI**从五个级别来评估并验证某组织的成熟度。按照**SEI**的规定，很清楚地评估和验证一个组织做什么，以及在某种程度上，他们如何完成。然而，这并不是规定一个“可重复过程”（二级）必须利用过程、工具和组织角色来完成。

相似地，“**RUP**的精髓”-- 以及已开发的许多实施**RUP**的工具 -- 培养逐渐细化的理念，即增量开发的本质。**RUP**的观点是组织应当设计并构建部分而不是全部解决方案，需求是已知的。现实中，验证某特色或者系统是“受人欢迎的应用程序”（比如，想法），还是“失败”（比如，**Coca-Cola's New Coke**，自1984）的一个最有效办法就是将产品交付给用户。

应用**RUP**，探寻**SEI CMM/CMMI**评估，或者使用**PMI PMBOK**时，最佳实践是成体系地使用这些向导。例如，你应该首先懂得业务需要（**a.k.a** 需求），从本质的用例开始，基于那些用例和**UML**的强大功能进行建模。在2004年《*The Rational Unified Process Made Easy*》一书中，**Per Kroll**和**Philippe Krutchen**很好地描述了这个方法：

... 也许，人们采用**RUP**时最常出现的错误是使用太多工件或者做太多活动。过量使用**RUP**将会降低你的开发效率；**RUP**过程框架类似于自助餐，如果你还想保持健康和快乐，那么就不能吃光所有的饭菜。<sup>1</sup>

## **RUP**应用在小型项目环境中

现在，让我们举两个例子，来验证**RUP**在小型项目环境中的使用。首先是公共部分项目 – 更新一个使用了十五年的打印工作过程。第二个项目涉及将**RUP**用于创建一个学习管理系统入口，称为“**TAP University**”。两个项目预算均低于\$100,000，由小型团队在90到120天以内完成。

### 打印服务更新项目

**Bill Wonch**，本文作者之一，是 **Nebraska** 州劳工部的兼职讲师和软件架构师。他最近负责更新一套已使用20年的程序，合计并打印出成千上万份报表和帐单，以下是他的故事。

这只是一个项目。但是，它却是系统的核心，称为 *Mix*，而且，必须支持部门内其它系统的更新。这个大框架说明了RUP中可交付的软件体系架构文档 -- 理解每个项目、变更命令，或者任务都影响着工作的进展，如同高尔夫球的每个线都与其它相关联一样。

当即有系统需要更新，以便与公司现代化的失业保险利益支付系统一起运作的时候，“*Mix*更新项目”开始了。原先的系统*Mix*是用COBOL构建的，运行于一个主机系统上。“*Mix*”并不是一个简称；1987年起名为“*Mix*”是因为它混合了进行大量打印工作的主框架数据和窗体。

新系统将在Java中使用成熟的商业化（commercial-off-the-shelf，COTS）应用和组件来构建，生成必要的XML文件。

项目的先启阶段，我们为系统定义三个参与者：

- 抽象应用类，表示使用即有*Mix*应用程序的所有系统。
- 操作类，表示员工管理打印的操作。
- 业务使用者，即使用该文档存储库的人员。

如图1所示，每个参与者均与相应的用例关联。记住这些参与者和用例，我们可以为更新系统选择最佳的商业应用。通过这个信息，我们可以精确地计算出更新所需的成本。那些是项目合同与计划中有限的内容。基于此，我们可以估算出项目的资金。

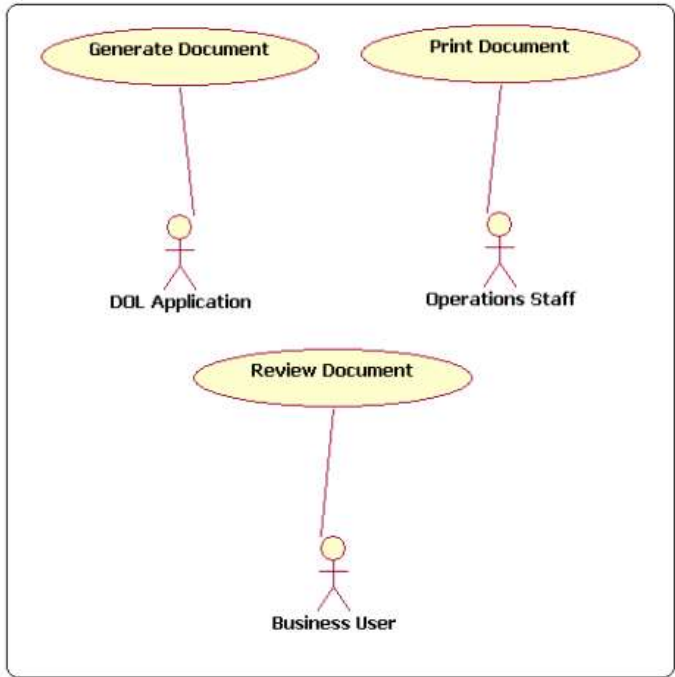


图1：在项目先启阶段为系统定义三个参与者

根据先启阶段确定的计划和捕获的用例，RUP指引着项目的进行。RUP精髓的一部分就是可以将需求划分成不同的组，并根据需要将各组归入先启、细化、构建和产品化阶段。*Mix*系统中包括106个打印程序，从先启阶段到产品化阶段，将这些程序分成几个组，然后再单独迭代地处理，经过四个阶段的每次进展都是低风险的（验证方法），然后再将大大小小打印程序集成。以上做法是有意义的。

TAP University

TAP (Technology As Promised) University是一个在线学习管理系统项目。TAP University的目标是延伸这种由TAP伙伴提供给公司客户的面对面培训，并为公司、公共用户及学生提供在线服务。<sup>2</sup>

这是一个小型的项目。改进一个开源的学习管理系统。该项目的可视化文档草案于2005年2月22日提出，项目计划完成于2005年5月3日，包括需要的资源、成本和范围。表1描述了每个迭代和用例。

表1：TAP University项目的迭代和用例

Iteration	Target release
1. LMS functional and ready for course loading and configuration	May 23, 2005
a. Use Cases	
1. Administer LMS	
2. Ingest Content	
3. Manage Users -- load instructors and students	
b. Actors	
1. TAP University LMS	
2. Instructors	
3. Course Developers	
4. System administrators	
5. Students	
2. Student registration and e-Commerce	June 20, 2005
a. Use Cases	
1. Register students	
2. Process payments	
3. Enable courses	
b. Actors	
1. Same as in #1 plus	
2. ACH systems (check)	
3. Credit card validation systems	
4. Accreditation systems	
3. Course conduct and extensions	August 15, 2005
a. Use Cases	
1. Modify Courses	
2. Interface with institutions	
b. Actors	
1. Same as #1 and #2 plus	
2. Institutional systems	

从设想到实施，这个项目只有不到六个月的时候；从正式的项目工作开始到功能的完成，从项目计划到支持这个产品仅花费了90天。

这里涉及到了8项资源；估计完成该项目所需的小时数为652。成本主要是“人力资本” -- 低于 \$15,000。

RUP在本项目中的应用主要包括以下两方面：

1. 在迭代和用例的组织方面，RUP已经提供了一个框架。表1所示的用例与包含MS project 进度表输出的两页项目计划共同构成了文档文件。CVS 1.12 和 LMS 充当共享库的作用。
2. RUP指导我们如何构建和产品化，甚至在仅仅已知80%需求的情况下。例如，有三个可选的电子商务解决方案有待评估。决定使用哪个电子商务工具并不排除在迭代1众的产出。这意味着公司客户能够立即地使用迭代1。

## 结论：RUP的确也适合于小型项目

文章中提到的两个小型项目呈现了不同类型的组织的需要：大型公共部门办事处与新近发展起来的小公司。项目的关注点也不同：更新使用15年之久的打印集合工具和在线学习管理系统。两个项目共同之处是，他们的规模都很小，并且RUP都可以提供一套严格而灵活的方法。

Gary Pollice 等几位作者在《小型团队的软件开发》一书中为小型项目的管理者提出了一些有价值的建议：面对不断的变化，项目团队如何掌握怎样应对变化才能获得最大的生产率？我们认为，关键在于尽可能多地学习不同技术，学习如何有效地使用工具来支持不同的技术，以及决定联合起什么作用和什么时候起作用。<sup>3</sup>

RUP以及各种支持RUP的工具，确确实实也“适用于小型项目”，另外，项目管理者应懂得如何最好地发挥RUP的优势。

## 注释

<sup>1</sup> Per Kroll and Philippe Kruchten, *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*, Addison-Wesley: 2004, pp. 244-245.

<sup>2</sup> <http://www.tapuniversity.com/>

<sup>3</sup> Pollice, Augustine, Lowe, and Madhu, *Software Development for Small Teams: A RUP-Centric Approach*, Addison-Wesley, 2004. p. xix.

---

## 参考资料

您可以参阅本文在 developerWorks 全球站点上的 [英文原文](#)。



### IBM Bluemix 资源中心

文章、教程、演示，帮助您构建、部署和管理云应用。



### developerWorks 中文社区

立即加入来自 IBM 的专业 IT 社交网络。



### IBM 软件资源中心

免费下载、试用软件产品，构建应用并提升技能。