

RUP 十大要素 - 有效开发流程的精髓

Leslee Probasco

Rational Software 白皮书

TP177, 9/00

目录

摘要 ...	1
请求帮助 ...	1
大量细节 ...	1
侧重于要素 ...	1
以“短列表”开始 ...	2
您的列表上应该有什么？“要看情况...” ...	2
首先构建框架 ...	2
RUP 十大要素 ...	3
远景 - 开发一个远景 ...	4
计划 - 计划 ...	4
风险 - 识别和降低风险 ...	5
问题 - 分配和跟踪问题 ...	5
业务案例 - 检验业务案例 ...	5
体系结构 - 设计组件体系结构 ...	5
产品 - 增量构建和测试产品 ...	5
评估 - 定期评估结果 ...	6
变更请求 - 管理和控制变更 ...	6
用户支持 - 为用户提供帮助 ...	6
关于需求？...	6
关于测试？...	7
总结：应用十大要素 ...	7
对于非常小的项目 ...	7
对于不断变大的项目 ...	7
对于成熟的项目团队 ...	7

摘要

要有效应用诸如 Rational Unified Process 那样的软件开发流程¹（简称“RUP”），重要的一点是首先理解其关键目标、每一个目标为何重要以及它们是如何协作的，这能帮助您的开发团队开发出符合项目干系人实际需要的高质量产品。

请求帮助

有一天晚上，我的邻居 Randy 来向我寻求帮助：他正在准备和他教堂的小组成员周末一起去野营和徒步旅行，他要决定外出时需要带上哪些装备¹。他已经有一些衣服和装备，并在犹豫是否要再买一些。他可以问我借装备列表吗？

我的装备列表是什么？是的。好像有一次我在准备登山旅行的背包时他也来找过我。他知道我在领导和参与野外旅行上很有经验，而且对我参考一张列有装备和衣服的详细列表来快速决定把什么放进我空间有限的背包里的这种做法，印象非常深。他能借那张列表吗？

是的，当然，很欢迎他来借我的装备列表。 - 但我担心它帮不了什么忙。为什么不能呢？是我认为他需要的东西并不都在那张列表上吗？

实际上，我相当确信他要的所有东西或多或少会出现在我的列表上，但是，很可能针对特定旅行的必需品可能未出现在我的列表上。例如，不仅是他的鞋码比我的大很多，而且他喜欢的食物（还有量）也和我携带的有区别。

大量细节

事实是在我的户外装备列表上写下了很多几百件物品。包括许多不同的外出物品类型：有登山背包、滑雪、雪地鞋、爬冰和皮船，还考虑到了旅行的时间长度，从一天到多天。比起我和密友一起在野外游玩，如果我是带领一个组织去旅行的话，我会带一些不同的物品（比如旅行签到与退出表）。我主要担心的问题是他无法看完列表上的多数物品，并且不能挑出什么是他这次相对简单的旅行所真正需要的。

Randy 恼怒地甩了甩手！他从没有及时正确地准备过他的旅行。他怎么知道包里需要装什么？他从上周就开始打包了 - 他在互联网上做了大量研究，买了新款时髦的靴子、夹克和衣服 - 他的包已经满了！

但他还没装任何食物！哦，对了，他还需要带水吧？但是如果再出现什么问题呢？如果他们迷路了该怎么办呢？或是有人受伤了呢？

侧重于要素

当我在给 Randy 已经放到包里的物品进行摆放时，我很容易就看出他不知道如何摆平那些野外游玩所必需的物品。

您有“十大要素”吗？我问道。十大要素？它们是什么？

Randy，我这里有您需要的列表。我抽出一张白纸，在顶端写上“十大要素”。在纸上，我写了 10 件物品：

1. 地图
2. 指南针
3. 太阳眼镜和遮光剂

¹ “装备”，许多登山者和其他户外运动爱好者的行话，用来表示他们在各种运动中所使用到的一些装备、工具、衣物、鞋类或者其他物品。如果您认为 RUP 有很多工件，请参阅：www.mgear.com。

4. 额外的衣服
5. 额外食物和水
6. 头帽灯
7. 急救箱
8. 生火器
9. 火柴
10. 小刀²

以“短列表”开始

就这些？是啊！就这些！如果您以这 10 件物品开始，请确认它们满足了各个方面的需要，然后其他一些东西会根据需求加进来。当然，每个“要素”的用处或大或小，这取决于旅行，但是以“短列表”开始然后再根据需要扩充，比起用一张长长的列表来决定不带什么，要轻松许多。

随着您经验的积累，您会不断改进这张“长列表”。它会在以后对您有用。到那时，您来看看我的或是其他人的列表会有助于您如何来正确扩充自己的列表。但是没有任何两个人的列表会完全一样（除非他们说谎）。

我想起了许多年前我刚开始登山的一张列表，无论我准备的是什么样的旅行或是外出多少时间，我仍然会参照这张列表。如果您问任何有经验的登山者“您有十大要素吗？”他们大多数很清楚您在问什么。实际上，不同的组或个人可能有不同的“十大要素”列表，但本质上它们是相同的。对于每次旅行，我实际带的物品可能完全不同，但我仍要说我有“十大要素”。

您的列表上应该有些什么？“要看情况...”

因此，这和 RUP 有什么关系？通常我在帮项目团队筛选 RUP 中的许多元素时（我最后发现了 4 个阶段、9 个核心工作流、31 名工人、103 个工件、136 项活动，再加上更多指导、核对表和工具向导，这比我想象的要多！），我会听到以下问题：“我该如何给所有这些物品进行筛选然后决定我的项目需要什么？”“我需要这个吗？”“RUP 是不是仅适用于大项目？”

大多数回答是（一种令人满意的回答）：“这要看情况了...”

我确定了我真正想要的是“RUP 十大要素”列表，我可以将它给正在寻找流程向导的朋友们 - 就如我为我的朋友 Randy 所做的那样。在决定项目包含的正确元素时，该列表可以作为一个合理的起点，同时它也可应用于短期旅行（较小的项目）、过夜背包，滑雪或皮船旅行（不同领域的中等项目）、珠穆朗玛峰探险（非常大而且重要的项目）时进行应用。这一想法就是要注重于我们所谓的 RUP 的“精髓” - 或者实际是其他有效的软件流程。这也回到了我们在软件开发中的“最佳实践”³概念。

首先构建框架

许多项目的常见问题在于：通常过分注重某个特定区域，以致他们陷入该特定区域的细节中，而无法确保他们能够很好地了解在生产高质量产品的整个流程生命周期内所涉及的“关键”元素。

然后，当项目进度落后的时候，他们会说：“看吧，我告诉过您需求管理会拖我们后腿！”您可以使用上文所述的“用例”、“收集项目度量”

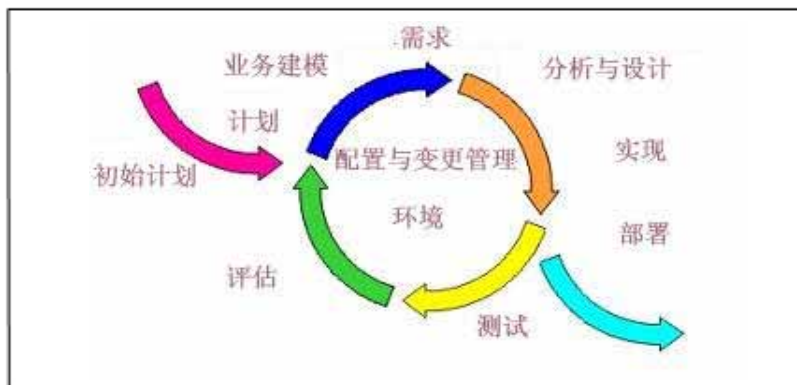
² 要获得此“十大要素”列表的完整分析，我推荐一本好书：Mountaineering - The Freedom of the Hills，第 6 版，The Mountaineers，西雅图，华盛顿，1997 年，第 35-41 页。

³ Rational 过去几年颇受青睐的“最佳实践”包括：（1）迭代开发、（2）管理需求、（3）使用组件体系结构、（4）可视化建模、（5）验证质量以及（6）控制变化。

“使用配置管理”、“使用检测跟踪工具”、“召开状态会议”或者其他我认为您熟悉的短语来替换“需求管理”。

我相信更有效的办法是采用更系统和整体的方法，在决定专注于某一特定问题领域之前，确认流程的关键因素（比如体系结构）正确。

一旦某个质量软件流程的框架适当，项目就可有效地注重于已确定为造成问题的主要因素的某个特定区域（通常我认为需求管理常会出现在列表的顶部）。记住，尽管 RUP 方法通过确定项目风险和设定这些风险的优先级，进行这种选择，从而为这些已确定的风险确定早期减轻策略。



RUP 的十大要素

因此，什么是“RUP 的十大要素”？以下列表描述了我认为一个项目所需项的最小集合，如果他们真正想要遵循 Rational Unified Process“精髓”的话：

1. 远景
2. 计划
3. 风险
4. 问题
5. 业务案例
6. 体系结构
7. 产品
8. 评估
9. 变更请求
10. 用户支持

让我们单独看看这些项中的每个，看看它们适合 RUP 的什么地方然后尝试理解为什么它们在我的要素项“短列表”上。

⁴ 由于需求管理是 Capability Maturity Model (CMM) 的初始“关键流程区域”之一，这根本不是那些决定提高他们软件开发流程的场景。

远景 - 开发一个远景

拥有清晰的远景是开发满足您项目干系人真正需求的产品。的关键。⁵

远景捕获 RUP 中需求流的“精髓”：分析问题、理解项目干系人的需求、定义系统和管理需求变化。

远景为更详细的技术需求提供高级（有时是约定的）依据。远景捕获极高级别的需求和设计约束，以便读者了解要开发的系统。它向项目批准流程提供输入信息，从而与业务案例紧密相关。它传达了有关项目的基本信息，包括为什么要开发系统以及具体要开发什么，同时它还是验证未来所有决定的标尺。

远景的内容应该回答以下问题（该工件可以按需要细分为独立的、更详细的工件）：

- ☐ 主要术语是什么？（词汇表）
- ☐ 我们要尝试解决什么问题？（问题描述）
- ☐ 谁是项目干系人？谁的用户？他们的需要是什么？
- ☐ 产品的功能是什么？
- ☐ 功能性需求是什么？（用例）
- ☐ 非功能性需求是什么？
- ☐ 设计约束是什么？

计划 - 进行计划

“有好的计划才会开发出好的产品。”⁶

在 RUP 中，软件开发计划（SDP）收集管理项目所需的所有信息。它包括了在先启阶段开发的许多独立工件，并在整个项目中得到维护。

它用于计划项目时间表和资源需要，以及按照时间表跟踪进度。它用于以下领域：项目组织，调度（项目计划、迭代计划、资源、工具）、需求管理计划、配置管理计划、问题解决计划、QA 计划、测试计划、测试用例、评估计划以及产品验收计划。

在简单的项目中，它们每个可能只包括一个或者两个句子。例如，CM 计划可以简单地声明：“在每天结束时，项目目录结构的内容将被压缩成 zip 文件，并复制到标有日期和标签的 zip 磁盘，然后标记一个版本号并放在集中的文件柜中。”

软件开发计划的格式本身不如用来生产它的活动和想法来的重要。因此，我不是很在乎它看起来如何 - 或者您用什么工具来构建它。就如 Dwight D. Eisenhower 所说：“计划什么都不是；计划又是一切。”

第 2、3、4、5 和 8 项要素捕获了 RUP 中项目管理工作流的“精髓”：构思一个新的项目、评估范围和风险、监视和控制项目、计划和评估每次迭代和阶段。

⁵ “目标是开发一个在时间和预算上满足项目干系人真正需要的合格产品。” - *Managing Software Requirements*, Dean Leffingwell 和 Don Widrig, Addison-Wesley Longman, 2000 年 1 月。

⁶ Johnson Space Center Shuttle Software Group, “They Write the Right Stuff”, Charles Fishman, *Fastcompany*, Issue 6, 第 95 页, 1996 年 12 月。

风险 - 识别和降低风险

RUP 的一个必要规则是在项目早期标识和攻击最高风险项。风险列表用来捕获已发现的风险，以使项目成功。它按优先级的降序来标识可能导致重大负面结果的事件。

对于每个风险，都应制定一个用来降低该风险的计划。它充当计划项目活动的焦点，并且是组织迭代的基础。

问题 - 分配和跟踪问题

使用直接从现行活动中推断出的客观数据进行持续的开放式沟通，以及不断演进的产品配置，这在任何项目中都很重要。在 RUP 中，这能通过常规状态评估来完成，而该评估提供处理、沟通以及解决管理问题、技术问题和项目风险的机制。除了确定这些问题外，每次评估都应指定一个到期日和一个负责解决该问题的责任人。这应按需要定期跟踪并更新。

这些项目快照提供了让管理层关注的缓冲时间。尽管阶段可能有所不同，但强制功能需要捕获项目历史记录并进行解析才能除去限制进度的任何障碍或瓶颈

业务案例 - 检验业务案例

业务用例从业务的观点提供了确定该项目是否值得投资的必要信息。

业务案例主要用于为实现项目远景而制定经济计划。一旦制定之后，就使用业务案例对项目提供的投资收益率（ROI）进行精确的评估。它提供对项目的调整并确定其经济约束。它向经济决策者提供关于项目价值的信息，并用于确定该项目是否应继续前进。

描述不应深入问题的细节，而是应就为什么需要该产品提出引人注目的看法。但是它必须简短，这样就容易让所有项目团队成员理解并牢记。在关键里程碑处，将重新检验业务案例，以查看预期收益和成本的估计值是否仍然准确，以及是否应继续该项目。

体系结构 - 设计组件体系结构

在 Rational Unified Process 中，软件系统（在给定点处）的体系结构是系统的重要组件与由依次减小的组件和接口组成的组件通过接口进行交互的组织或结构。主要组成部分有哪些？它们如何组合在一起？

RUP 提供了一种讲究方法的、系统的方式来设计、开发和验证体系结构。这就是 RUP 中“分析”和“设计”工作流的“精髓”：定义候选体系结构、优化体系结构、分析行为和设计系统组件。

要说明并理解软件体系结构，您必须先定义一种体系结构表示法，一种描述体系结构重要方面的方式。在 RUP 中，该描述可以在软件体系结构文档中找到，它以多个视图来表现体系结构。

每个体系结构视图均处理特定于开发流程中项目干系人（最终用户、设计人员、经理、系统工程师、维护人员等）的某一组具体问题。这充当架构设计师和其他项目团队成员就对项目做出的重大结构决策进行沟通的介质。

产品 - 增量构建和测试产品

RUP 中“实现”和“测试”工作流的“精髓”是使用开始后每次迭代结束时发行的可执行版本来增量地编码、构建和测试系统组件。

在精化阶段结束时，一个体系结构原型可用于评估，如果必要，这可能还包括一个用户界面原型。在构造阶段的每个迭代期间，都将组件集成到放入最终产品的，可执行的并且测试过的构件中。

这一要素的关键是伴随产品构建集成的测试活动、正在进行的配置管理和复审活动的集合。

评估 - 定期的评估报告

迭代评估捕获迭代的结果、符合评估条件的程度、得到的教训和要实现的流程变化。

迭代评估是迭代法的一个必要工件。根据项目的范围和风险以及迭代的性质，它的范围可能包括从简单的演示和结果记录到完整的正式测试复审记录。

关键集中在流程问题以及产品问题上：“您越早地落后，就需要越多的时间追赶。”

更改请求 - 管理和控制变化

“配置”和“更改”管理工作流的“精髓”是管理和控制项目的范围，因为变化会出现在整个项目的生命周期，同时还要有考虑所有项目干系人的需要和最大程度上满足他们需求的目标。

在向用户展示第一个原型时（通常比这还要早），就将产生变更请求。（生活的一种必然！）为了控制那些更改并有效地管理项目的范围和项目干系人的期望，重要的是通过变更请求来提出对任何开发工件所作的所有变更并始终使用一个流程来管理。

变更请求用于记录和跟踪缺陷、增强请求和任何其他类型的产品变更请求。变更请求的好处在于它们提供决策的记录，而且鉴于它们的评估流程，可确保所有项目团队成员都理解潜在变更的影响。变更请求对于管理项目的范围以及评估建议变更的影响非常必要。

用户支持 - 为用户提供帮助

至少应该有一个用户指南（可能通过联机帮助来实现），同时还可以包括安装指南和发行说明。根据产品的复杂性，还可能

需要培训资料以及伴随任何产品打包的材料清单。

在 RUP 中，“部署”工作流的“精髓”是包装和交付产品，同时使用一切必要的资料来帮助最终用户学习、使用、操作和维护产品。

关于需求？

有些人在看了我列出的要素后可能会对我的选择表示强烈反对。例如，“需求”在何处适合这一情况？会不会有比“远景”更好的选择呢？需求当然是必要的，是吗？好问题。我想说基本上那就是第一个要素 - 这是针对那些想要获得列表上“需求”的人来说的，您可以将它放到您自己列表上的这一位置！（记住，每个项目团队应该有他们自己的列表。我的十大要素列表只是作为更多讨论的起点。）

从我自己的经验来说，尽管有时我会问项目团队（尤其是对于内部项目来说）：“你们有什么需求？”并且得到回答：“我们实际上没有什么需求。”起初这让我很惊讶（回答来自航天军事背景）！他们怎么可能没有任何需求？然后我明白了当他们听到“需求”这个词时，他们脑子中所想的东西，那是指外部强制要求做的事情他们必须遵循这些事情或者拒绝这个项目 - 但是他们真的没有那些东西！他们可能不得不投入到研发之中，此时产品的需求会贯穿于整个项目。

现在我如果这样回答：好的，那么产品的远景是什么？那么，他们的眼睛一亮，轻松地回答以上 A 到 G 的问题而没有任何困难，而后紧接着便是需求了。这在协作环境中常是如此（那里可以发现需求,而不是被命令强加需求），而在合同性项目中则不是。

关于测试？

毫无疑问，一些敏锐的人也注意到了我把“测试”作为 RUP 的要素之一。为什么呢？与业务建模不同，我确实认为它对于一个有效的软件开发流程来说是可选的，但我绝不认为测试是可有可无的。我认为测试是伴随产品设计和构建集成的活动集（这非常像配置管理和复审活动的进行方式），而且，巧合的是，它和符合 IEEE 1074 软件流程标准。

如果您非常敏锐，您注意到测试确实包含在产品的构建之中（在要素 7 中），而且测试结果对于验证和评估（要素 8）相当重要。对于软件开发，Rational Unified Process 迭代方法的精髓是不断构建，测试和评估产品的可执行版本，以尽早清除问题和解决风险与问题。

总结：应用十大要素

因此，“RUP 十大要素”的发现是如何改变我们生活的？

对于非常小的项目

首先，如果有人像我的朋友 Randy 一样来问我关于如何自己或是和一个缺乏经验的小团队（只了解流程）一起合作使用 RUP 来购建简单的产品，现在我有我自己“十大要素”的列表，我可以和他分享，而不是将所有 RUP 的功能和 Rational 的工具包丢给他！

事实上，这十个要素不用专门的支持工具就能完成。一本项目笔记本，其中每一要素有一小节来描述，那么实际上它是管理项目的好的起点。我还发现记事贴说明在刚开始管理、划分优先级和为我个人 RUP 应用跟踪变更请求的时候非常有用！

对于不断变大的项目

当然，这十大要素的简单应用方法随着项目大小和复杂性的增加，也很快变得不可管理。所以那些在寻找 Rational 工具潜在用户的人会在那些应用“RUP 十大要素”的人们中发现成熟前景。

当然，最后，一旦他们决定深入研究而不仅仅是每个领域的最浅显级别时，他们希望得到所有 RUP 的功能，还有 Rational 工具的支持（如果您的工作是正确的！）。但是，我鼓励您以“RUP 十大要素”，还有我们的“最佳实践”为指导，而不是以工具为指导。

对于成熟的项目团队

对于更多的成熟项目团队，“十大要素”能有助于提供一个快速的方法来估计流程中关键元素的平衡，并且标识能使改进最有收益的领域。

在所有的项目中，研究如果忽略了这些要素会发生什么，或是如果您没有这些要素，则什么会失败这类问题，是非常重要的。例如：

- ☐ 没有远景？您可能失去方向并可能容易因走弯路而烦恼。
- ☐ 没有计划？您将无法跟踪进度。
- ☐ 没有风险列表？您现在可能正专注于处理错误问题，并可能在今后 5 个月内都在勘探一口未被怀疑的矿井。
- ☐ 没有列出问题？如果没有责任，成功的路上会有绊脚石。
- ☐ 没有业务案例？您会冒着在项目上浪费时间和金钱的风险，项目可能被取消或者崩溃。

- 没有体系结构？您可能无法在出现通信、同步和数据访问问题时处理它们；可能存在规模和性能方面的问题。
- 没有产品（原型）？开始写代码，仅积累书面工作不会让您走的很远，去获得些客户有用的东西吧。
- 没有评估？不要回避现实。重要的是面对真相。您究竟离截止日期有多近？距离您的质量目标或预算目标又如何呢？
- 没有变更请求？您如何跟踪它们？
- 没有用户支持？当用户有问题或不确定如何使用产品时怎么办？获得帮助的难易程度如何？

所以，您得到它了。

为什么是十大要素？原因只是因为我不想超过十个（数不出更多了！）。我倒是希望只有“九大要素”或更小的数字，但好象不能再减了。想想那个数它们变得都必不可少，是吗？至少对我来说是的！我鼓励您以下建议作为项目组的开始点：您的“十大要素”是什么”？

如果您像我一样，您可能需要使用一些记忆方法或首字母缩写来帮助您记住“十大要素”列表。例如，将它降为四音节，我使用 V-PRI-BAPE-CU，当然，由于我把要素的数量限定为 10 以后，我可以随意地用我的手指来数它们，而不用参考我 PalmPilot 上的列表。（现在您知道了我的秘密！）



两家总部：

Rational Software
18880 Homestead Road
Cupertino, CA 95014
电话：(408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
电话：(781) 676-2400

免费电话：(800) 728-1212
电子邮件：info@rational.com
Web：www.rational.com
全球网址：www.rational.com/worldwide

Rational、Rational 徽标和 Rational Unified Process 是 Rational Software Corporation 在美国和 / 或其他国家或地区的注册商标。Microsoft、Microsoft Windows、Microsoft Visual Studio、Microsoft Word、Microsoft Project、Visual C++ 和 Visual Basic 是 Microsoft Corporation 的商标或注册商标。其他所有名称均仅用于标识目的，它们是其相应公司的商标或注册商标。ALL RIGHTS RESERVED.

Copyright 2006 Rational Software Corporation.
如有更改，恕不另行通知。