



需求和质量管理解决方案
白皮书
2009 年 10 月

Rational. software

通过集成方法来管理需求和质量

需求驱动测试原则

目录

- 2 概述
- 2 什么是需求管理?
- 3 什么是测试?
- 4 需求驱动测试原则
- 4 及早规划测试
- 5 及早开展测试
- 5 将测试与需求相关联
- 6 将缺陷与需求相关联
- 7 根据需求满足情况评估项目进度
- 7 W 模型
- 11 结论
- 11 更多信息

如想有效管理需求，公司必须能够设计明确的、可测试的需求，了解它们在项目中的重要性，并且在项目的整个生命周期中对需求进行跟踪。

概述

需求管理在项目开发的各个阶段均发挥至关重要的作用。开发团队若能使用精准的、可测试的需求来统一设计项目并且以能否满足需求来评估项目进度的话，将能够大幅度提高效率、灵活性和可销售性。

在这个白皮书中，我们将简要介绍对需求和质量管理应用集成方法的多个优势，并且提供最佳业务实践来帮助您有效地开展需求驱动的测试。

此外，我们还将回顾您在将需求驱动的测试集成到项目生命周期时需要谨记的原则，并且讨论典型测试模型的优点和发展史。

什么是需求管理？

需求是产品或流程必须满足的要求。

需求分许多类型，我们可以使用许多字眼来定义它们，包括：针对、渴望、条件、合同、局限性、目标、命令、需要、目的、义务、制度、前提条件、规则、规范和目的地等。在这篇文章中，我们将使用需求（requirement）来指代上述所有概念。我们还将使用“需求（requirement）”来指代单句说明、从句和项目，而不是指代整个需求文档。

需求管理由一系列原则和活动组成，与需求的捕获、表达、组织、版本确定、出版、跟踪、分析和变更有关。

许多人都误把需求管理理解成局限于项目定义阶段的活动。实际上，需求以各种形式存在于所有开发阶段。需求在许多测试阶段也发挥至关重要的作用，甚至包括开发过程中的最后几个阶段。

摘要

在本文中，我们将重点讨论三个重要的需求管理原则：

- 需求表达。需求不仅需要简单明了，而且还必须能够测试。这通常意味着您需要确保合理地量化需求并且明确定义测试标准。测试标准将为开展测试工作提供重要信息。
- 需求分类。需求多种多样，作用有轻有重。如想有效地管理项目开发，您必须为面向应用的每句需求陈述进行适当的分类，以便做出明智的决策并且合理分配资源。
- 需求可跟踪性。确保可跟踪性是需求管理的关键环节。开发人员和经理都必须能够在整个开发期间对需求进行跟踪。此外，他们还必须能够结合相关需求对测试进行跟踪，以便有效跟踪项目进度并且分析变更的影响。

在整个开发周期中始终坚持测试能够帮助您找出缺陷并且跟踪哪些需求已

什么是测试？

测试是指旨在解决问题的任何活动。

在此，我们故意较为宽泛地定义测试，让其覆盖大量的早期活动，如需求审核，设计检查，结构化预排，建模分析，以及传统的单元、系统和验收测试活动等。

对许多人来说，这个定义可能过于宽泛，将测试活动V模型右手边的传统位置上移除。但是，这个定义允许您在开始任何构建或编码工作之前，对需求和设计本身进行“测试”（审核、验证、确认、质量认证或评估）。

在这个定义中，我们使用单词*质量认证 (qualification)* 来覆盖所有的测试、验证、确认和评估工作。因此，“测试标准”变成了“质量认证标准”，“测试方案”变成了“质量认证方案”。

摘要

需求驱动的最佳业务实践要求我们同时创建需求及需求测试计划。

与需求的任何偏离都视为缺陷。

理论上说，缺陷是相对需求而言的。需求分为产品需求（源自客户）和开发流程需求（标准和程序）。后者包括最佳业务实践，本身设计用于减少产品开发期间产生和传播的缺陷数量。

需求驱动测试原则

我们必须始终遵守一些核心测试原则，这些原则体现了需求驱动测试的最佳业务实践：

及早规划测试

在编写需求之际为每个需求做好测试规划。想一想如何测试需求。这样，当您捕获需求时，便能够更加明确地表达需求，因为这个流程会鼓励您去考虑如何量化需求。对于每个需求，都应该自问“我们如何知道自己是否有能力满足这个需求，或者已经满足了这个需求？”

请注意，回答这个问题时既应该借鉴以往的经验，又应该放眼未来。设计检查等早期测试能够证实若根据设计来构建产品，产品能否满足需求。后期测试则能够检查您所构建的产品是否已经满足了需求。

要想回答上面这个问题，应该对每个需求进行质量认证。但是，由于几乎每个需求都需要接受多个测试，贯穿整个开发周期，因此，质量认证活动的工作量是巨大的。例如，质量认证的早期工作包括根据需求来检查设计。设计中是否包括必要的单元来确保满足需求？此外，您可能还需要开展一次或多次的系统测试来确保最终产品能够满足需求。质量认证方法由一系列精挑细选的测试组成。

摘要

在项目开始后尽早通过测试找到缺陷有助于最大限度地减少昂贵的重复工作。

每个需求至少必须开展一次测试，您才能了解哪些需求变更会对整个系统产生影响。

有效的测试不仅需要测试单个组件，而且还要测试这个组件与其他系统组件之间的互动。

这个问题还会使您为每个测试确立*质量认证标准*。这些标准能够评估每次测试是否取得了成功。质量认证标准由需求和计划内测试决定。

及早开展测试

在开发过程中及早开展测试。

古语云“防微杜渐”。缺陷会从一个开发阶段传播到另一个阶段，沿着开发流程前进，纠正缺陷的成本会成倍增长。若能及早发现并且纠正缺陷，您将能够在随后的工作中最大限度地减少昂贵的重复工作，既省钱又省时。

因此，面向每个需求的质量认证计划都应该规定尽早检测缺陷，包括审核、检查和预排等方法。

将测试与需求相关联

基于相关需求对测试进行跟踪检查。

跟踪是在开发期间记录工件相关性的方法。最常见的跟踪方法是跟踪不同需求层之间的*满意度*关系。然而，跟踪需求之间的质量认证关系同样非常重要，此外，您还需要对测试进行跟踪，以便查看需求是否得到了满足。

您可通过跟踪开展两类分析：

- *覆盖范围分析，能够帮助您确保至少为每个需求规划并且最终执行一次质量认证活动。还能帮助您确保每次质量认证活动都与相关的需求相挂钩，从而创造优势。*
- *影响分析，能够帮助您决定在需求发生变更时需要采取哪些质量认证活动。在接受需求变更之前，您必须评估重新定义和重新执行相关质量认证活动的成本。此外，您必须了解失败的测试对需求质量认证活动的影响，还必须了解失败的测试对需求的影响。*

摘要

将每个缺陷与一个或多个需求相挂钩。良好的需求管理系统允许您跟踪项目进度，随时查看项目对既定需求的满足情况。

影响分析结合使用*满意度*和*质量认证*关系来决定潜在的重复工作。例如，如果组件发生故障，不仅会影响到相关组件需求（通过*质量认证*关系），而且还会影响到旨在满足这些组件需求的子系统（通过*满意度*关系），进而影响到系统和客户需求的满意度。

测试人员知道各级测试都必不可少。例如，如果仅仅测试子系统组件的话，将无法全面测试子系统需求，因此，必须对整个子系统进行测试（即这些组件的集成体）。这与需求管理中的*加和特性*（*emergent properties*）概念相呼应：系统不是简单的组件和，系统行为有时也与组件间的互动有关。

将缺陷与需求相关联

将缺陷与需求相关联，存在缺陷即体现为产品未能满足需求。

当测试发现意外行为时，或者发现存在未能满足测试标准的行为时，即表示存在缺陷。导致出现缺陷的原因主要有三个：

1. *测试定义中存在错误。*
2. *需求表达或者质量认证标准中存在错误。*
3. *产品中存在真正的缺陷。*

当您对产品进行跟踪时，如果发现存在未能满足的需求，便能够找到缺陷。实际上，“缺陷”被定义为“与需求相偏离”。由于您需要基于多个需求对每次测试进行跟踪，因此，您必须分析每个缺陷以便准确判断它们会对哪些需求产生影响。

摘要

根据需求满足情况评估项目进度

设置目标并且根据需求满足情况来评估项目进度。

什么时候应该停止测试是开发过程中最难做出的决策。在资源有限的情况下，测试经理需要了解开展哪些测试活动最有效。如果测试与需求相脱离的话，经理将无法判断不同系统组件的相对重要性，因此，无法做出明智的资源分配决策，也无法准确分析故障影响。

如果拥有适当的跟踪工具，测试经理将能够轻松了解每次测试的相对重要性以及测试成败对各级需求的影响—客户需求和系统需求等。此外，可跟踪性还允许项目主管选择合理的目标并且准确评估项目进度，将资源用在刀刃上，生成准确的报告来显示与需求相对应的测试结果。

W 模型扩展了典型的 V 模型，以便显示需求与行动之间的关系。

W 模型

W 模型是流程和数据模型，是典型 V 模型的发展结果，能够为我们显示如何通过最佳方法来实施需求驱动测试。

在典型的 V 模型中，如图1 所示，方框代表项目在整个生命周期中的活动。（这个模型不应被解释为“瀑布”模型，因为所有这些活动都是并行发生的，并且支持不变的反馈）。

摘要

质量认证时间表显示了各开发阶段每个单元测试、审核、或其他质量认证活动的时间安排。

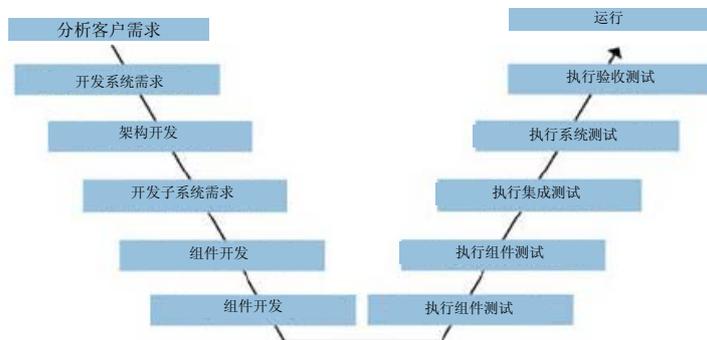


图1: 典型的V模型

图2 添加了活动生成的数据，是长方形里面的内容，还显示了数据之间的踪迹，以加粗的箭头表示。这个模型以 V 模型为轴心，在左手边并行显示质量认证方案和需求。质量认证方案中只包括既定的质量认证活动及标准，系统会在适当的开发阶段收集每次活动的执行结果。

您可沿着踪迹来分析两种关系：

1. 需求层之间的满意度关系。
2. 既定的质量认证活动与需求之间的质量认证关系。

您应根据质量认证方案，在适当的开发阶段安排执行既定的质量认证活动（审核、检查、单元测试、集成测试等）。这些活动汇聚在一起，构成了面向整个项目的“质量认证时间表”。

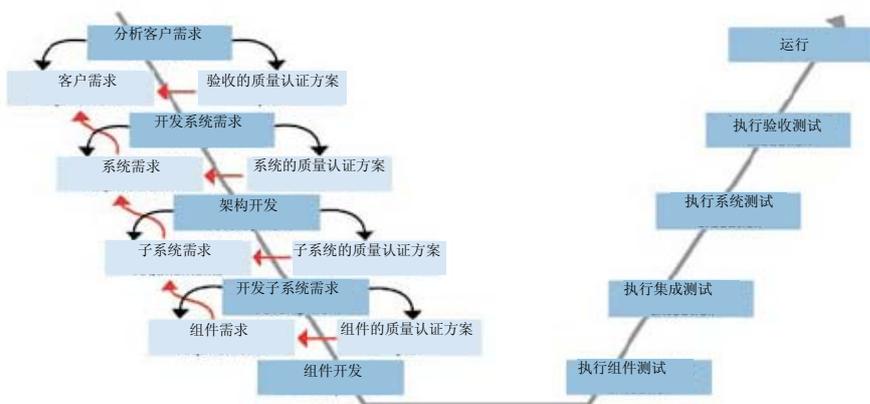


图2: 带质量认证方案的V模型

摘要

您可将缺陷信息保存在单一存储库中以便进一步分析。

开展最初的设计测试时，您经常没有足够的信息来规划测试的具体细节。因此，图 3 以单独的活动为基础来显示测试设计。这些活动所在的轴与开发轴并行。处于对基本形状的考虑，这个模型有时被称为 W 模型。

如想设计有效的测试，请使用质量认证方案来开发特定的测试方案。您可能需要为每个方案设计多次测试，或者通过一次测试来满足多个既定的测试需求。

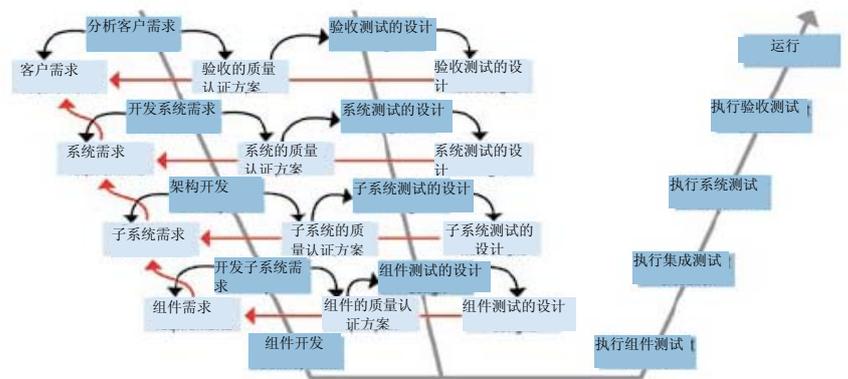


图3: 带测试设计的V模型 (或W模型)

W 模型会以单个活动为基础来显示测试的设计架构。

图4 显示了带测试结果数据的模型。在该图中，您可以看到状态和缺陷信息。

由于没有将缺陷具体归类到特定的开发层，因此，缺陷在一个统一框中显示。您在执行测试计划时要对缺陷进行分析，以便决定这个缺陷是不是真正的产品缺陷及其影响到哪些需求。图5 显示了这个分析的结果。

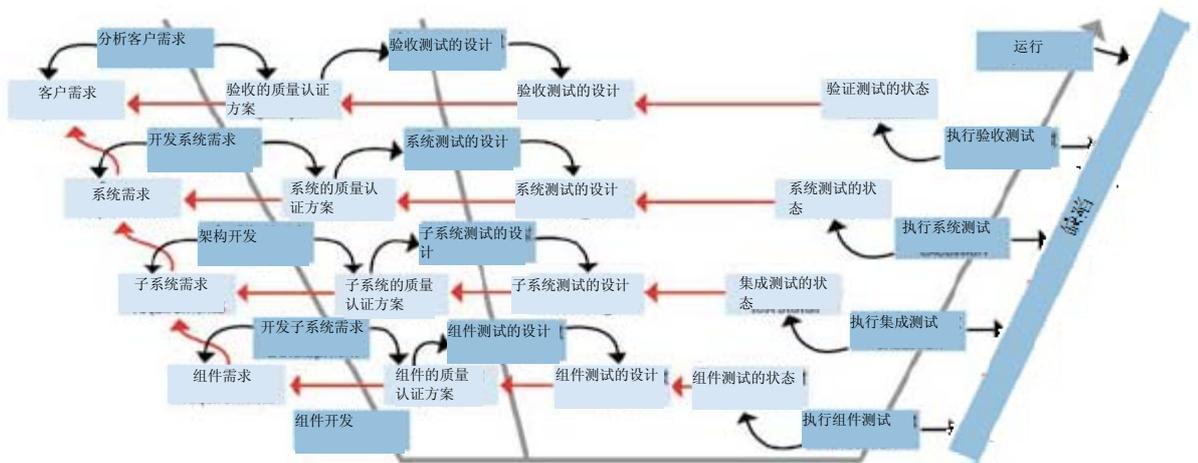


图4: 带测试结果的W模型

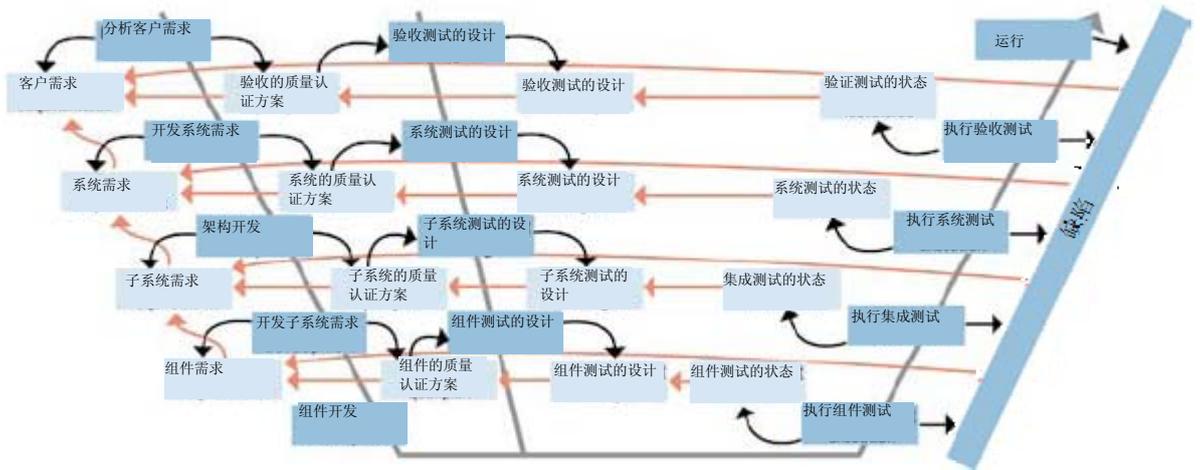


图5: 带未能满足需求的缺陷的W模型

摘要

分析缺陷可以决定哪些需求受到了影响以及必须采取哪些行动。

这种方法与上述的三个原则相一致:

- 及早规划测试。与需求同步编写质量认证计划, 确保在编写需求之际考虑到测试。
- 及早开展测试。将质量认证方案与质量认证时间表相挂钩, 最好能够创建质量认证活动时间表, 并且决定对每个需求进行测试的最早时间点。
- 跟踪测试结果, 找出与需求相关的缺陷。跟踪模型允许您维护并且分析这个关系。

请注意, 如想决定某个需求是否完全被满足, 需要同时使用满意度和质量认证关系来收集完整的测试结果集。

迭代测试可以帮助您确保根据需求识别、分类并且解决全部问题。

最后, 测试应该是高度重复的; 您应该反复开展测试, 发现缺陷并且纠正缺陷。图6 显示了采用统一标准的多次测试。通过顺序开展测试, 您可以判断测试结果有何改进、测试有何进步。

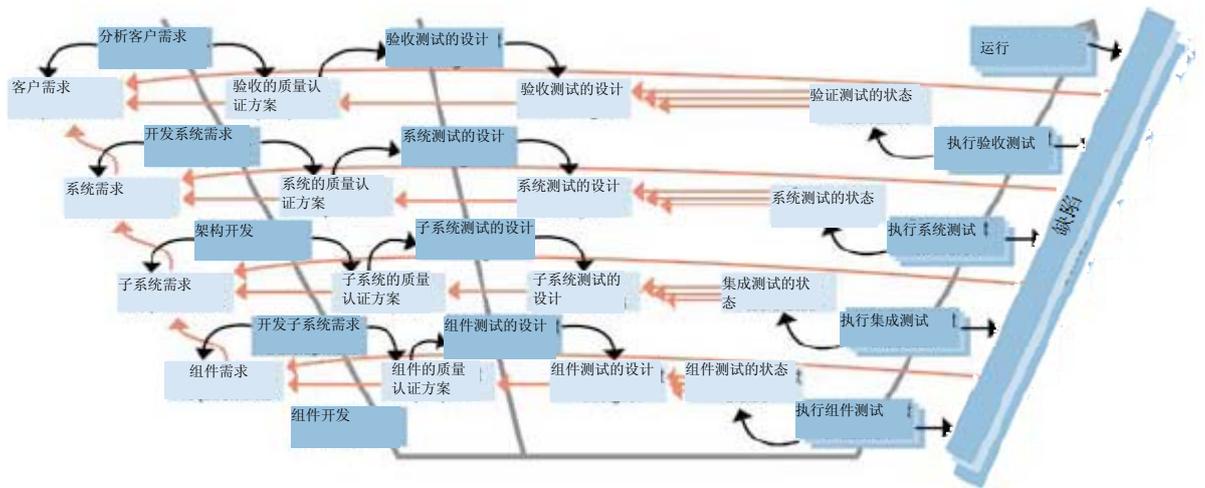


图6: 运行多个测试的W模型

结论

为了在市场中维护竞争优势，现在的产品、系统和软件开发公司必须杜绝使用单一用途的生产力增强工具和单点解决方案。

使用集成工具来开展需求驱动测试能够帮助您在分析人员、开发人员和测试人员之间同步测试和开发和数据，以便他们交流对测试结果和需求相关性的理解，帮助公司提高生产力和产品质量，在现在和未来更好地满足客户需求。

IBM® Rational® DOORS®和IBM Rational Change 提供集成工具集，能够帮助公司在项目的整个生命周期中创建测试计划、跟踪需求并且管理缺陷。

更多信息

如想详细了解面向集成需求管理和质量管理的 IBM 解决方案，请与当地的 IBM 业务代表或 IBM 业务伙伴联系，或者访问：

ibm.com/software/rational。



© IBM 公司 2009 年版权所有
保留所有权利

IBM Corporation
Software Group
Route 100
Somers, NY 10589, U.S.A.

本文 2009 年 10 月在中国制作

IBM, IBM 标识, ibm.com, DOORS 和 Rational 是国际商用机器公司在美国及/或其他国家的商标或注册商标。这些及其他因为在本文中第一次出现而标记出商标符号 (® 或™) 的 IBM 术语, 均代表在本文出版之际, 它们是 IBM 在美国注册的商标或约定俗成的商标。这些商标可能也是 IBM 在其他国家注册的商标或约定俗成的商标。关于 IBM 商标的最新列表, 请访问 ibm.com/legal/copytrade.shtml, 参见“Copyright and trademark information”。

其他公司、产品或服务名称是各自所有者的商标或服务标记。

本文提到的 IBM 产品或服务, 不代表 IBM 计划在其开展业务的所有国家都提供它们。

本文中包含的信息只用作信息目的, 是“按原样”提供的, 不包括任何明示或暗含的保证。此外, 这些信息基于 IBM 现在的产品计划和战略, 未来定将有所变动, 恕不另行通知。由于直接或间接使用本文或任何其他文件导致的任何损失, IBM 概不负责。本文绝非打算提供 IBM (或其供应商和许可颁发机构) 的保证或陈述, 也不应对他们的保证或陈述产生任何影响, 或者更改现行 IBM 软件许可协议的任何条件和条款。

IBM 客户负责确保自己对法律的遵从。客户自己全权负责就与其业务相关的任何法律的识别和解释向合格律师请求建议, 并全权负责为达到此类法律的要求而采取的行动。

RAW14142 -USEN-00