

## 附录 当前需求实践的自我评估

本附录包括20个问题，您可以用它们来校准您当前的需求工程策略，以确定在哪些领域需要加强。从每个问题的四个可选项中，选出一个最能描述您当前处理软件需求问题方法的答案。如果您想确定自我评估的质量，为每个“a”答案给0分，“b”给1分，“c”给3分，“d”给5分。最高的可能得分是100分。但是，不要太在意分数的高低，用自我评估发现能应用新策略的机会，您的组织将会从中受益。每个问题都指出了与该问题主题相关的章节。

某些问题可能并不适合您组织所开发的软件。环境不尽相同，您的项目也许不需要最全面、最复杂的方法。例如，对于市场上没有先例的高度创新的产品，由普通产品概念得到的需求可能就不适合。但是，应该承认非正式的需求方法增加了大量返工的可能性。从总体上遵照以下的策略，大多数组织都将从中获益。

1. 项目范围是如何确定、交流和使用的？【第6章】
  - a. 设计产品的人通过心灵感应与开发组织进行交流。
  - b. 有书面的项目任务陈述。
  - c. 使用标准的任务和范围文档模板，所有项目成员都能访问这个任务和范围文档。
  - d. 评估所有建议的特性和需求变更，确定它们是否与文档中的任务和范围相符。
2. 客户如何与确定的和表征的产品进行交流？【第7章】
  - a. 不能确定谁是客户。
  - b. 销售也许知道谁是客户。
  - c. 通过管理，从销售调查和从现有客户基础中确定目标客户。
  - d. 销售、管理和关键客户代表不同的用户类别，软件需求规格说明概括了他们的特征。
3. 您如何得到用户需求的输入？【第7章】
  - a. 开发人员已经知道需要建设什么。
  - b. 销售能提供用户的观点。
  - c. 调查或访问典型用户的中心小组成员。
  - d. 明确代表不同用户类别的个人参与项目，并约定其责任与权利。
4. 如何培训您的需求分析员，他们是否富有经验？【第2章】
  - a. 他们丝毫没有经验，未接受过特殊的开发需求培训，他们宁愿编写代码。
  - b. 分析员是受过一两天软件需求训练的开发者，以前具有和用户交互的经验。
  - c. 他们经过短期培训，在采纳技术和主持小组会议方面具有丰富的经验。
  - d. 我们具有专业或系统分析员，他们在与用户合作方面具有广泛的经验。他们同时理解应用领域和软件开发过程。
5. 如何将系统需求分配到产品的软件部分？【第7章】
  - a. 软件是用来弥补硬件的不足。
  - b. 软件与硬件工程师讨论哪个子系统应该实现什么功能。
  - c. 系统工程师分析系统需求并将其中一些分配给软件。

- d. 系统需求的一部分分配给软件子系统并跟踪明确的软件需求。清楚地定义子系统界面并文档化。
- 6. 用什么方法来分析客户的问题？【第 8 章】
  - a. 我们的开发者很聪明，他们理解问题很深入。
  - b. 我们询问用户想要什么，记录下来，将其实现。
  - c. 我们与用户讨论他们的业务需求与当前的系统，然后写一个软件需求规格说明。
  - d. 我们观察用户如何完成他们当前的任务，将他们当前的工作流程模型化，了解他们希望新系统完成什么功能。这向我们表明他们的部分过程如何自动化，告诉我们什么软件特性是最有价值的。
- 7. 用什么方法来确定所有明确的软件需求？【第 8 章】
  - a. 我们从一个概要的理解开始编写代码，并修改代码直到完成。
  - b. 管理或销售提供了一个产品概念，开发者制定需求。由销售来确定开发是否有遗漏，有时销售也负责向开发者转告产品概念的变更。
  - c. 销售或客户代表告诉我们产品应该具有什么特性和功能。
  - d. 我们为产品与不同用户类别的代表进行有组织的会见或专题讨论。我们通过使用实例来理解用户任务，然后从使用实例中得到功能需求。
- 8. 如何将软件需求写成文档？【第 9、10 章】
  - a. 我们搜集口头历史、e-mail 信息、语音邮件信息、会见记录和会议记录。
  - b. 我们编写非结构化的叙述性的文本文档，或者画出结构化的或面向对象的分析模型。
  - c. 我们结合一些用标准概念表示的图形分析模型，在与标准软件需求规格说明模板一致的基础上，用结构化的自然语言书写需求。
  - d. 我们将需求存储在一个数据库或商业需求管理工具中，将分析模型存储在 CASE 工具中。每个需求将和一些属性共同存储。
- 9. 如何获取非功能性需求，如软件质量属性，并将他们编写成文档？【第 11 章】
  - a. 什么是“软件质量属性”？
  - b. 我们通过用户界面的 beta 测试来得到用户喜欢什么的反馈。
  - c. 将某些属性，如操作与安全需求，编写成文档。
  - d. 我们通过与用户交谈来确定每个产品的重要质量属性，然后将它们在软件规格说明中用准确的、可验证的方式记录下来。
- 10. 如何标记单个需求？【第 9 章】
  - a. 我们采用叙述性文本段落，明确的需求并未单独确认。
  - b. 我们采用加重号和数字的列表。
  - c. 我们采用层次数字方案，如“3.1.2.4”。
  - d. 每个单独的需求都有一个独立的、有意义的标记，它不会随其它需求的增加、移动或删除而遭到破坏。
- 11. 如何建立单个特征或需求的优先级？【第 13 章】
  - a. 它们都很重要，否则我们不会首先将它们记录下来。
  - b. 客户告诉我们什么需求对他们最重要。
  - c. 根据客户的意见所有的需求都标记为高、中或低优先级。

- d. 我们使用分析过程，来评估与每一个使用实例、特征或功能需求相联系的价值、代价和风险。
12. 采用什么技术来准备局部的解决方法，并验证对问题的相互理解是否一致？【第 12 章】
- a. 没有任何技术，我们只管开发系统。
  - b. 我们开发一些简单的原型并从询问用户获得反馈。有时，我们不得不分发原型的代码。
  - c. 我们开发原型不仅用来作为用户界面的模型，也在适当的时候作为概念的技术证据。
  - d. 我们计划开发抛弃型的报告和电子原型来帮助我们改进需求。有时我们也开发评估模型。我们在原型的基础上通过结构化的评估脚本来获得客户反馈。
13. 如何评估需求文档的质量？【第 14 章】
- a. 我们认为我们的需求相当好。
  - b. 我们巡回传递需求文档来获得反馈。
  - c. 分析员和一些开发者进行非正式的评审。
  - d. 在客户、开发者和测试者的参与下，我们对软件需求规格说明和分析模型进行正式的审查。我们针对需求编写测试用例，用它们来确认软件需求规格说明和模型。
14. 如何分辨不同版本的需求文档？【第 16 章】
- a. 自动生成文档的打印日期。
  - b. 我们为每一个文档版本采用一个序列号，如 1.0，1.1 等等。
  - c. 我们采用一种区分方案能将草稿版本和基线版本、主要修改和次要修改区别开来。
  - d. 需求文档通过版本控制存储在配置管理系统中，或者需求存储在保存每个需求修改历史的商业需求管理工具中。
15. 如何跟踪软件需求到它们的来源？【第 18 章】
- a. 不跟踪。
  - b. 我们知道许多需求的来源。
  - c. 所有需求都有一个确定的来源。
  - d. 我们在软件需求和一些客户需求陈述、系统需求、使用实例、标准、规则、结构要求或其它来源之间建立全面的双向跟踪。
16. 如何利用需求作为开发项目计划的基础？【第 15 章】
- a. 交付日期在我们搜集需求之前就已经确定了。我们不能修改进度或需求。
  - b. 在交付日期之前我们进行一个快速的缩小范围的过程来去掉一些特性。
  - c. 项目计划的第一次反复确定收集需求所需的计划，项目计划的余下部分在我们得到需求后完成。但是，在此之后不能修改计划。
  - d. 我们通过需求估算产品的大小，并在对实现要求功能所需工作的估算的基础上制定进度和计划。如果需求变更或者进度延期，通过协商来更新计划和协定。
17. 如何利用需求作为设计的基础？【第 15 章】
- a. 我们并不进行明确的设计。
  - b. 如果有所编制的需求文档，我们在编程时也许会参考它们。

- c. 需求文档包含用户界面设计和我们计划实现方案的其它方面。
- d. 设计者审查软件需求规格说明以确保它能作为设计的基础。我们在单个功能需求和设计元素之间具有全面的双向跟踪。

18. 如何利用需求作为测试的基础？【第 15 章】

- a. 需求和测试之间没有直接的联系。
- b. 测试者根据开发者的陈述来进行测试。
- c. 我们根据使用实例和功能需求来设计系统测试用例。
- d. 测试者检查软件需求规格说明书以确保需求是可验证的，并开始计划测试过程。我们将系统测试回溯到明确的功能需求。测试的进展部分地由需求覆盖来度量。

19. 如何确定和管理每个项目的软件需求基线？【第 16 章】

- a. 什么是“基线”？
- b. 虽然客户和经理不再提出要求，但我们仍然收到大量的变更和客户意见。
- c. 虽然我们定义了需求基线，但它不能总是与过去作出的变更保持一致。
- d. 当初始基线定义时，需求存储在一个数据库中。当批准需求变更时更新数据库和软件需求规格说明。一旦确定了基线则保存对每个需求的变更历史。

20. 如何管理需求的变更？【第 17 章】

- a. 常常有未经控制的变更蔓延到项目中。
- b. 当需求阶段完成后，通过冻结需求来拒绝变更。
- c. 我们采用标准表格将变更请求提交到一个建议中心。由项目经理决定采纳哪些变更。
- d. 变更通过一个确定的变更控制过程来进行，该过程使用一个工具来搜集、存储和交流变更请求。在变更控制委员会决定是否批准每个变更并评估其影响。