

何时好的群体将发生不合格的需求说明 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/66/2021_2022__E4_BD_95_E6_97_B6_E5_A5_BD_E7_c41_66034.htm 不重视需求过程的项目

队伍将自食其果。需求工程中的缺陷将给项目成功带来极大风险，这里的“成功”是指推出的产品能以合理的价格、及时限在功能、质量上完全满足用户的期望。下面将讨论一些需求风险。不适当的需求过程所引起的一些风险 #8226. 用户需求增加带来过度的耗费和降低产品的质量。 #8226. 用户增加一些不必要的特性和开发人员画蛇添足(gold-plating)。 #8226. 忽略某类用户的需求将导致众多客户的不满。

•. 不完善的需求说明使得项目计划和跟踪无法准确进行

。1. 无足够用户参与 客户经常不明白为什么收集需求和确保需求质量需花费那么多功夫，开发人员可能也不重视用户的参与。究其原因：一是因为与用户合作不如编写代码有意思；二是因为开发人员觉得已经明白用户的需求了。在某些情况下，与实际使用产品的用户直接接触很困难，而客户也不太明白自己的真正需求。但还是应让具有代表性的用户在项目早期直接参与到开发队伍中，并一同经历整个开发过程

。2. 用户需求的不断增加 在开发中若不断地补充需求，项目就越变越庞大以致超过其计划及预算范围。计划并不总是与项目需求规模与复杂性、风险、开发生产率及需求变更实际情况相一致，这使得问题更难解决。实际上，问题根源在于用户需求的改变和开发者对新需求所作的修改。要想把需求变更范围控制到最小，必须一开始就对项目视图、范围、目标、约束限制和成功标准给予明确说明，并将此说明作为评

价需求变更和新特性的参照框架。说明中包括了对每种变更进行变更影响因素分析的变更控制过程，有助于所有风险承担者明白业务决策的合理性，即为何进行某些变更，相应消耗的时间、资源或特性上的折中。产品开发中不断延续的变更会使其整体结构日渐紊乱，补丁代码也使得整个程序难以理解和维护。插入补丁代码使模块违背强内聚、松耦合的设计原则，特别是如果项目配置管理工作不完善的话，收回变更和删除特性会带来问题。如果你尽早地区别这些可能带来变更的特性，你就能开发一个更为健壮的结构，并能更好地适应它。这样设计阶段需求变更不会直接导致补丁代码，同时也有利于减少因变更导致质量的下降。

3. 模棱两可的需求

模棱两可是需求规格说明中最为可怕的问题(Lawrence 1996)。它的一层含义是指诸多读者对需求说明产生了不同的理解；另一层含义是指单个读者能用不止一个方式来解释某个需求说明。模棱两可的需求会使不同的风险承担者产生不同的期望，它会使开发人员为错误问题而浪费时间，并且使测试者与开发者所期望的不一致。一位系统测试人员曾告诉我，她所在的测试组经常对需求理解有误，以致不得不重写许多测试用例并重做许多测试。模棱两可的需求带来不可避免的后果便是返工重做一些你认为已做好的事情。返工会耗费开发总费用的40%，而70%~85%的重做是由于需求方面的错误所导致的(Leffingwell 1997)。想像一下如果你能减少一半的返工会是怎样的情况？你能更快地开发出产品，在同样的时间内开发更多、更好的产品，甚至能偶尔回家休息休息。处理模棱两可需求的一种方法是组织好负责从不同角度审查需求的队伍。仅仅简单浏览一下需求文档是不能解决模棱

两可问题的。如果不同的评审者从不同的角度对需求说明给予解释，但每个评审人员都真正了解需求文档，这样二义性就不会直到项目后期才被发现，那时再发现的话会使得更正代价很大。其它检测模棱两可需求的技术由Gause和Weinberg (1989) 给予介绍。

4. 不必要的特性 “画蛇添足” 是指开发人员力图增加一些 “用户欣赏” 但需求规格说明中并未涉及的新功能。经常发生的情况是用户并不认为这些功能性很有用，以致在其上耗费的努力 “白搭” 了。开发人员应当为客户构思方案并为他们提供一些具有创新意识的思路，具体提供哪些功能要在客户所需与开发人员在允许时限内的技术可行性之间求得平衡，开发人员应努力使功能简单易用，而不要未经客户同意，擅自脱离客户要求，自作主张。同样，客户有时也可能要求一些看上去很 “酷”，但缺乏实用价值的功能，而实现这些功能只能徒耗时间和成本。为了将 “画蛇添足” 的危害尽量减小，应确信：你明白为什么要包括这些功能，以及这些功能的 “来龙去脉”，这样使得需求分析过程始终是注重那些能使用户完成他们业务任务的核心功能。

5. 过于精简的规格说明 有时，客户并不明白需求分析有如此重要，于是只作一份简略之至的规格说明，仅涉及了产品概念上的内容，然后让开发人员在项目进展中去完善，结果很可能出现的是开发人员先建立产品的结构之后再完成需求说明。这种方法可能适合于尖端研究性的产品或需求本身就十分灵活的情况 (McConnell 1996)。但在大多数情况下，这会给开发人员带来挫折 (使他们在不正确的假设前提和极其有限的指导下工作)，也会给客户带来烦恼 (他们无法得到他们所设想的产品)。

6. 忽略了用户分类 大多数产品是由

不同的人使用其不同的特性，使用频繁程度也有所差异，使用者受教育程度和经验水平也不尽相同。如果你不能在项目早期就针对所有这些主要用户进行分类的话，必然导致有的用户对产品感到失望。例如，菜单驱动操作对高级用户太低效了，但含义不清的命令和快捷键又会使不熟练的用户感到困难。

7. 不准确的计划 “上述是我对新产品的看法，好，现在你能告诉我你什么时候能完成吗？”许多开发人员都遇到这种难题。对需求分析缺乏理解会导致过分乐观的估计，而当不可避免的超支发生时，会带来颇多麻烦。据报道，导致需求过程中软件成本估计极不准确的原因主要有以下五点：频繁的需求变更、遗漏的需求、与用户交流不够、质量低下的需求规格说明和不完善的需求分析（Davis 1995）。对不准确的要求所提问题的正确响应是“等我真正明白你的需求时，我就会来告诉你”。基于不充分信息和未经深思的对需求不成熟的估计很容易为一些因素左右。要作出估计时，最好还是给出一个范围（如最好的情况下，很可能的，最坏情况下）或一个可信赖的程度（我有90%的把握，我能在8周内完成）。未经准备的估计通常是作为一种猜测给出的，听者却认为是一种承诺。因此我们要尽力给出可达到的目标并坚持完成它。

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