

什么是需求管理

作者: Richard Stevens & James Martin, Telelogic

包含对 Ricka Harwell, Pradip Kar, Bob Smith, Michele Bailey 等人观点的引用。本文旨在探究需求管理的本质，需求管理所要涉及的任务在文中将适时提及，以阐释“对需求管理的需求 (requirements for requirements)”的涵义。

1. 概要

需求管理恰如裁缝的量体裁衣，它直接关系到最终产品的成型。仅从字面出发，如果一个产品满足了客户需求，那它无疑就是成功的。需求管理的过程，从需求分析开始贯穿整个项目始终，力图实现最终产品同需求的最佳结合（参见图 1）。通过对需求管理在项目进程中实施的不同任务进行分析，我们可以看出需求管理所起的作用。

需求管理可以被定义为确证：

- 我们确知客户的需求是什么（质量）；
- 满足客户需求的最佳解决办法（一致性）；

著名学者 Crosby 对于质量的定义是“同需求保持一致”。从定义系统的功能开始，需求管理是一种保证我们生产出合格产品的方式。通过需求管理对问题的描述，使得每个人都始终明白他们所做的具体任务。然而，在一个产品的生命周期里，需求是能动的，是处于变化之中的。

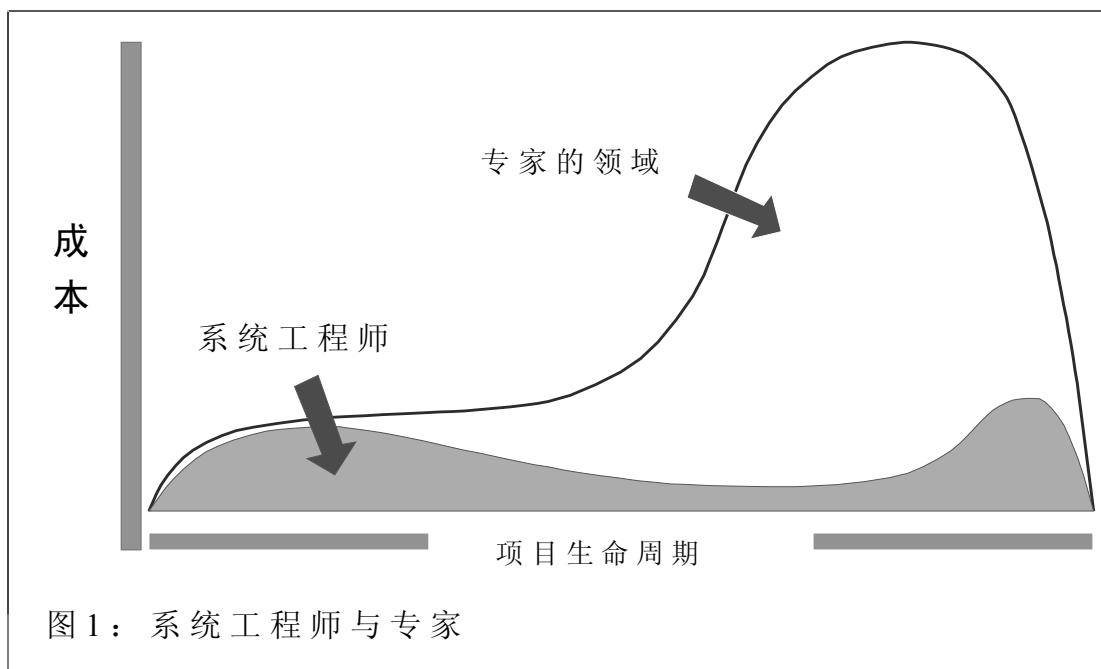


图 1：系统工程师与专家

对于系统工程没有严格统一的定义，因此很难找到足够的证据以说明系统工程所起的作用。有些致力于研究需求分析的组织认为，一项开发计划应当至少将 8-15% 的资源投入到系统工程（通常以客户的角色来看）方面。如果低于这一标准，将很可能导致无法对客户群做出准确把握。如果该项开发计划含有许多创新或实验的成分，那么这一百分比还应当适度提高。

2. 需求管理所要完成的任务

可以说需求是一种模型，是系统的早期雏形，通过进行需求分析，我们可以对最终产品做出优化。需要始终保持注意的是，需求是始终处于变化的并存在于整个生命周期之中。需求管理需要完成的任务包括：

- 定义与沟通需要做什么
- 对外部文档进行跟踪
- 根据不同需求设计相应解决办法
- 进行系统优化
- 驱动设计与实施；
- 管理变更，报告问题，管理建议
- 把任务分解到不同的专业领域
- 对最终产品做出评测
- 控制迭代开发过程
- 提出项目实施时间表
- 确定最终用户界面

有时候我们所进行的需求分析只停留于分析本身，而没有进一步去思索我们怎样对需求进行扩展。需求是项目开发的源头，只有进行认真的需求分析，我们才能做到对症下药、量体裁衣，才能在设计开发中去伪存真，不断改进。“对需求管理的需求”正是强调了贯穿生命周期的需求分析的重要。离开了系统流程而空谈需求管理，无异于纸上谈兵。需求管理是前期的工作，它的效益在系统流程的后期才能得到体现。如果没有可控的系统工程流程，需求管理的工作就不能见效。以下篇幅分别介绍需求管理在系统工程流程中的不同应用。

需求管理是对所需要功能的沟通：

用户需求模型以非术语的形式定所需要的功能。这个模型使每一位开发者明确自己的职责所在，并且清楚知道不同开发工作之间的关联。这里的“用户”泛指在实际应用环境中每一位可能使用最终产品的人。如果一个产品不能满足客户所需，那么设计方案再出色也无济于事，

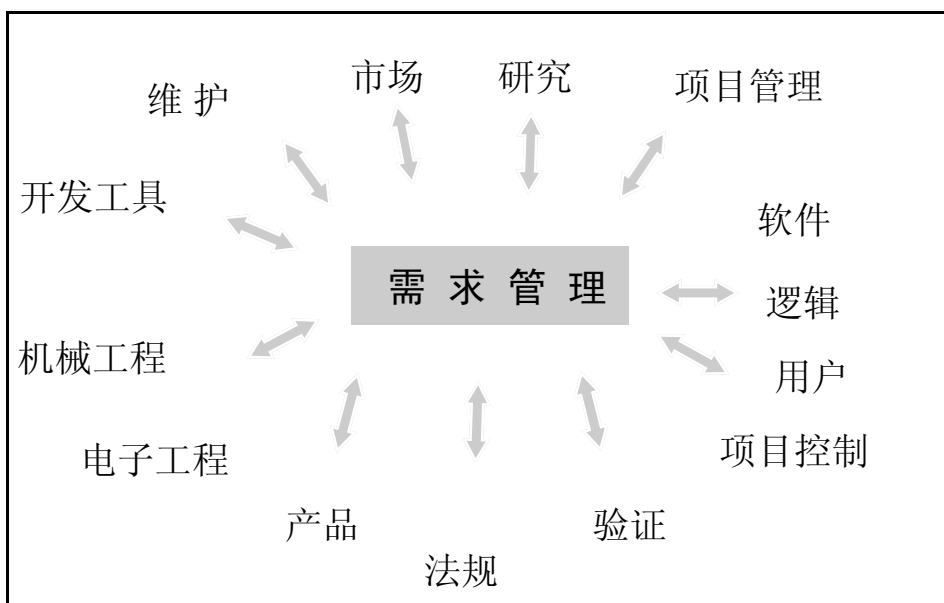


图3：以需求管理作为沟通方式

许多方案有很高的技术设计水准却最终不能获得成功，其原因正在于此。可以把产品功能说得天花乱坠，但却无法改变用户需求决定最终产品基本模式的事实。

需求管理的首要任务在于使开发人员和用户双方对于需求都有一个明确的认识。因此用来进行需求分析的语言应当使所有相关人员，包括用户，都能够理解，进而都能够对整个项目有一个整体把握，并明确每一个人在项目中所起的作用。因而需求管理需要解决的第一位也是最基本的任务就是明确需求，并使所有相关人员达成共识。

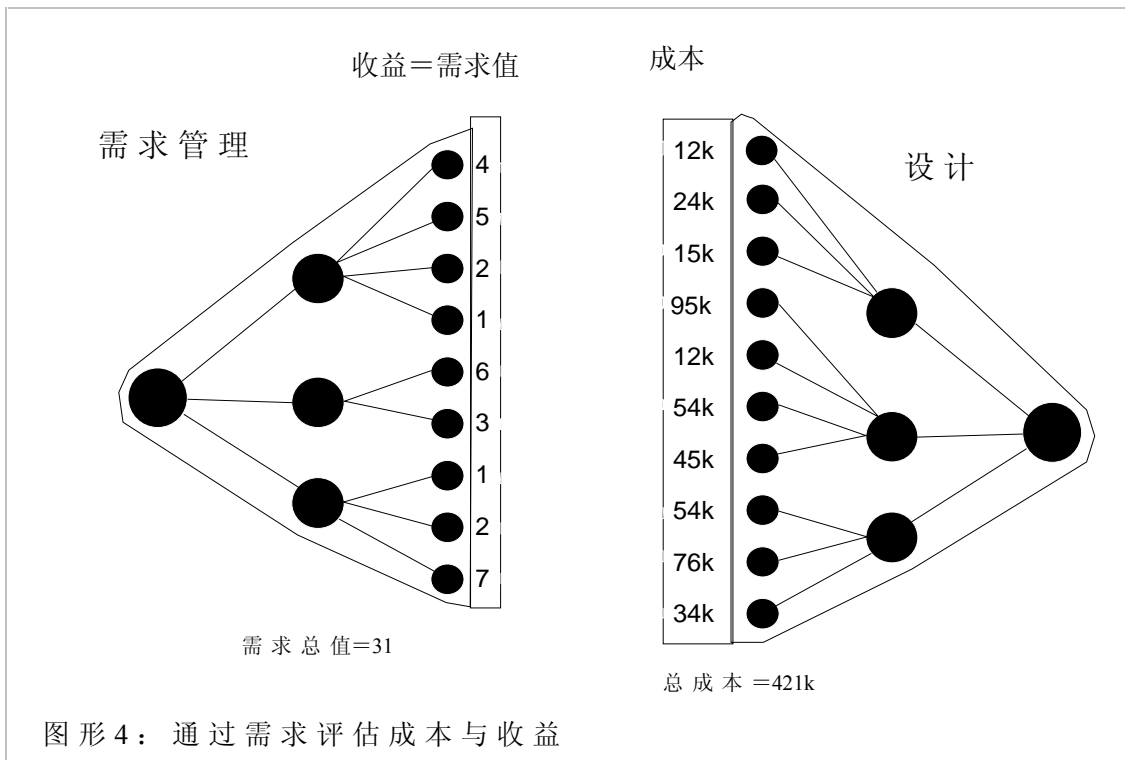
根据需求设计解决办法：

我们在进行系统设计时，应当首先建立一个需求模型，但不能是为了建模而建模，需求模型实际是最终产品的抽象化表现。需求模型的建立使我们在明确需求的基础上更进一步，使我们知道我们将要生产何种产品，该产品都具有那些功能。同时，需求模型也使开发者明确自己的工作如何同整个项目有机地结合在一起。抽象的需求模型应当允许不同的设计方案选择，而不能被限制在某一特定的架构建模方式中。

根据需求优化系统：

任何设计都应以考虑用户需求为优先，用户需求的满足程度即成为衡量设计优劣的标准。在一个项目设计周期中，开发人员经常会面临选择，以提炼需求，决定开发的优先次序，并在不同的实施方案中作出选择。这些选择必须考虑到收益与成本的平衡比例，这种选择的重要性尤其在建立需求模型的后期凸现出来。基本需求在这时都已明确，而实施方案还未敲定，为了使用户的基本需求得到落实，一定程度的开销用于搭建不同构架的需求模式是合理的。假使我们已经有了有一套翔实的需求分析，我们甚至不必将每一套方案都付诸实行，就可以成功地对系统设计进行优化。

面对不同的可行性而需要作出选择时，我们也必须参照收益与付出的比例关系。例如，在



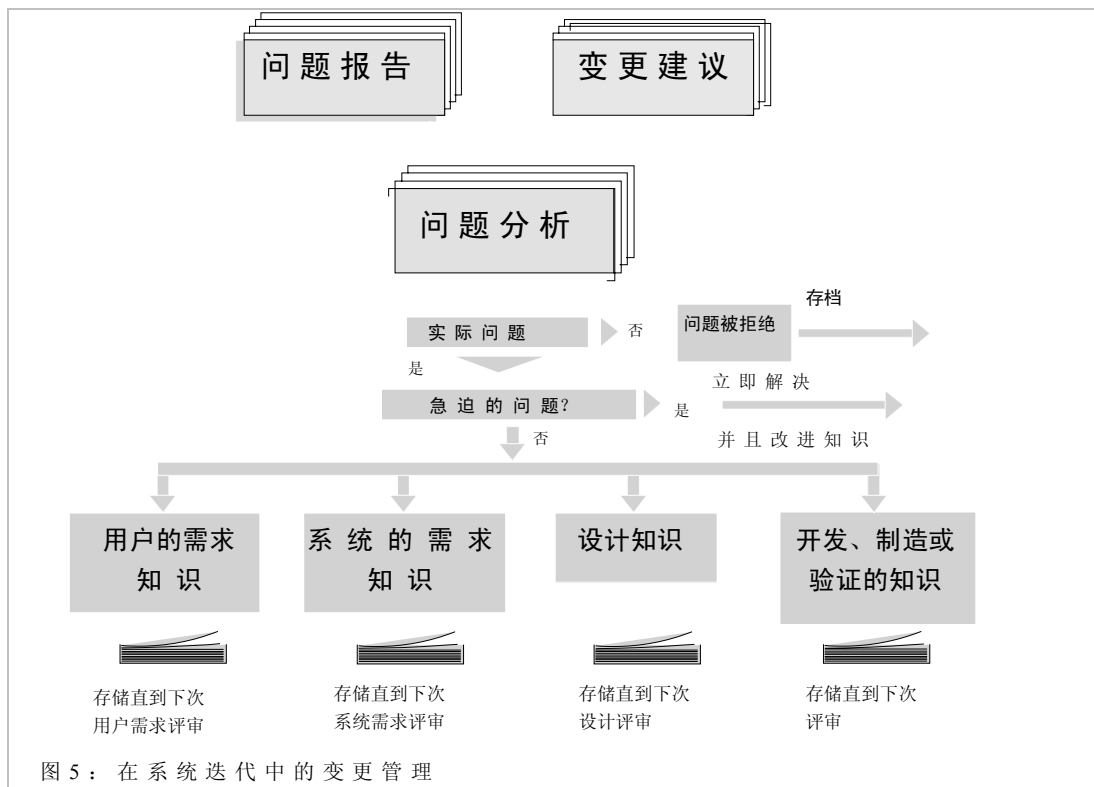
被要求提供建议书时（Request for Proposal），我们应当尽量做到每一份建议书的提供都物有所值。

需求驱动设计与实施：

明确需求后，开发人员就可以进行方案设计。通过对用户需求和设计方案之间所存在关联性进行分析比较，我们就能够对设计方案进行评估。

需求用于变更管理：

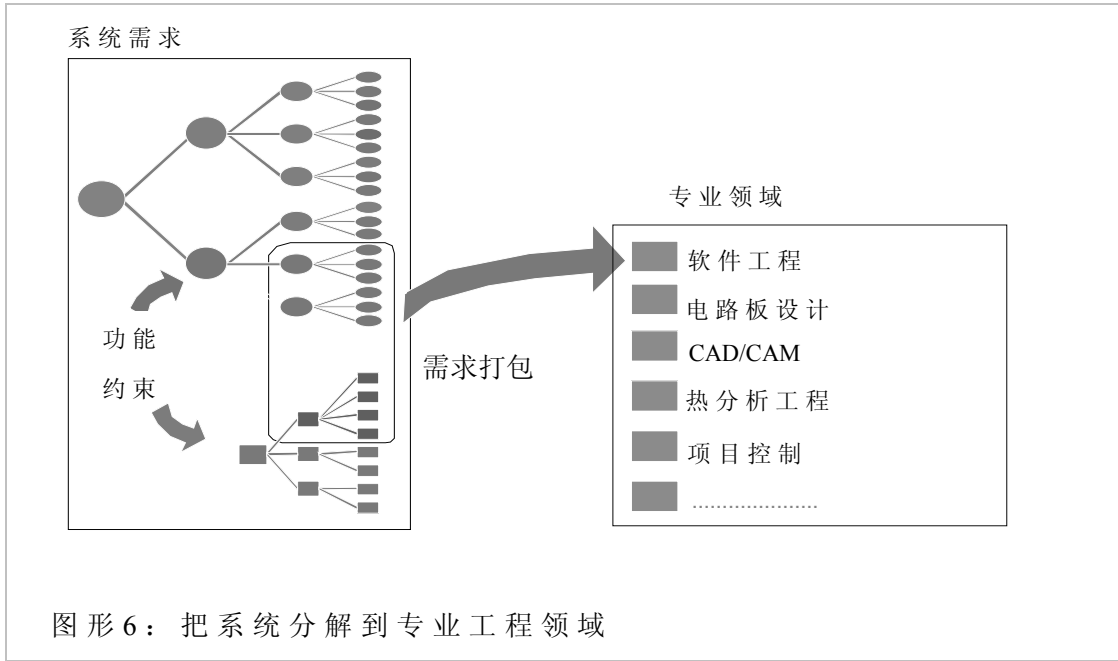
方案的设计不可能是一成不变的，经常会有方案设计同需求相悖的情况。如果我们无法准确把握用户需求同方案设计之间的关系，我们就无法在需要对方案进行必要修改时正确判断。优秀的需求分析应当非常精确细致地对用户需求作出描述，同时也应该最大程度地给予方案设计者充分发挥的余地。



需求用于系统的分解：

一个大的开发项目可能涉及 20-30 组不同的开发队伍，人员包括技术工程师、软件工程师以及具体项目主管等等，而每一个模块都有它自己的开发工具和开发语言。主持一个大项目的开发并不是件容易的事，总体项目主管的首要任务是对开发项目进行任务划分，将整体开发任务细化为多个子模块，从而使这些子模块能够并行开发而无需太多在系统层次上的干预。总体项目主管可以将细化的不同模块的需求分析交给不同的开发队伍，对于开发进程的监控只需参照需求的解决情况，对于具体的开发细节则不必过问太多。

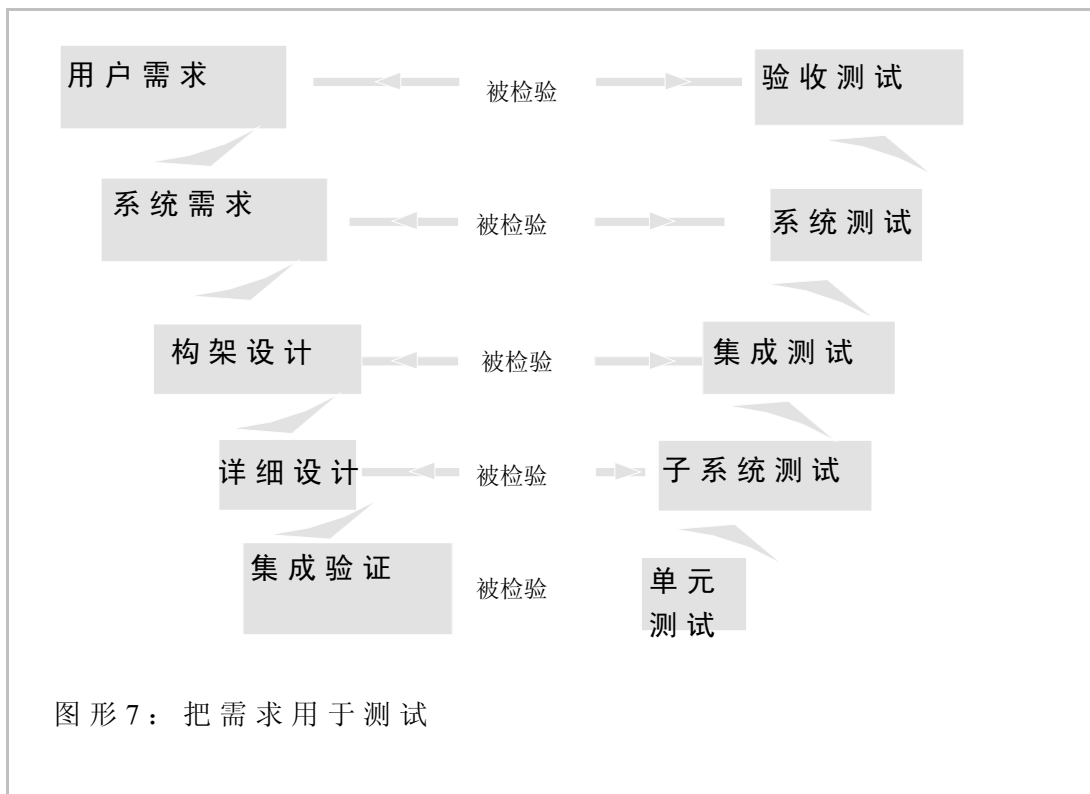
不同的开发队伍会使用不同的开发语言和开发工具，会使用不同的符号和标记。相反，作为总体项目主管所使用的语言、符号和标记等则必须简单易懂，以使所有的开发人员都等理解。换言之，总体项目主管应当使用自然语言，或只涉及少量的，简单的术语和专用词汇。



需求用于产品测试：

需求的满足情况是决定最终产品成败的判定基础，对最终产品的测试评估必须以产品所试图解决的需求为标准。V模式的验证（图 7）标示了不同的开发阶段所对应的测试需求。

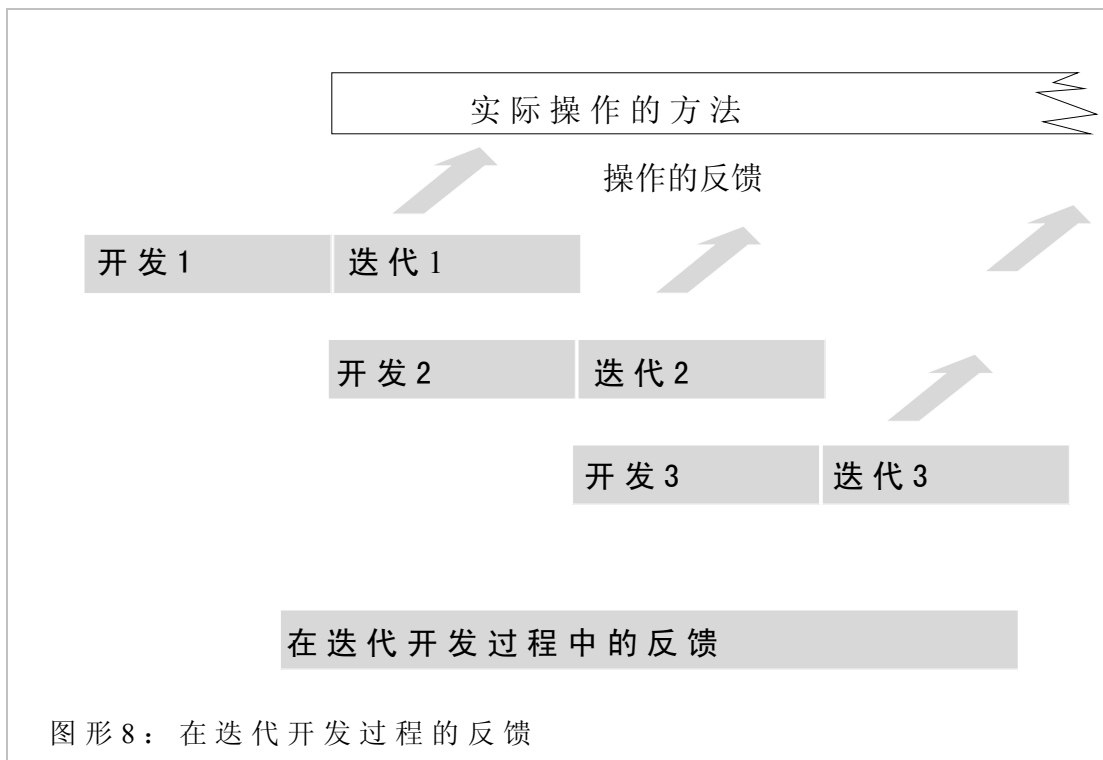
这里有一个需求、产品和测试系统之间的关系问题（图 7），确定需要进行的测试属于总体开发主管的工作范畴，虽然具体工作也许是由分包商来承担。



需求用于管理迭代开发中的变化：

对于总体开发主管而言，针对方案设计的修改是一项经常性的工作（因为修改而造成的影响则应当尽可能减小）。在进行项目开发时，随着开发进程的深入，各种修改的建议和问题的报告是屡见不鲜的，每解决一个问题，就是将产品同其需求的结合又完善了一步。存在问题正是需求尚未满足的表现。

在迭代式的开发过程中，方案设计的完善和需求的满足是同步的。实际用户对于产品的评价和建议尤其具有重要意义。在那些一步到位的产品设计中，实际用户无法左右开发进程，但在迭代设计、迭代开发的过程中，开发人员应当及时搜集用户对于产品的反馈信息，并将这些信息结合到下一步的开发工作中去。如下图所示，用户反馈同产品开发是统一的。

**需求用于辅助项目管理：**

在有些地方，需求管理被作为一个技术问题来处理，需求管理所针对的对象只是产品，而同项目管理所涉及的问题例如进程安排或资源分配等无关。实际上，项目管理涉及三方面问题：进程安排、资源分配和质量管理（=与需求保持一致）。

试想以下三种情况：

- “一场高水准的音乐会，预算合理，演出时间却晚了两天”
- “质量优良的小轿车，交货及时，然而造价是市价的两倍”
- “一套系统，完全满足了用户需求，但在开发过程中使用非法劳工”

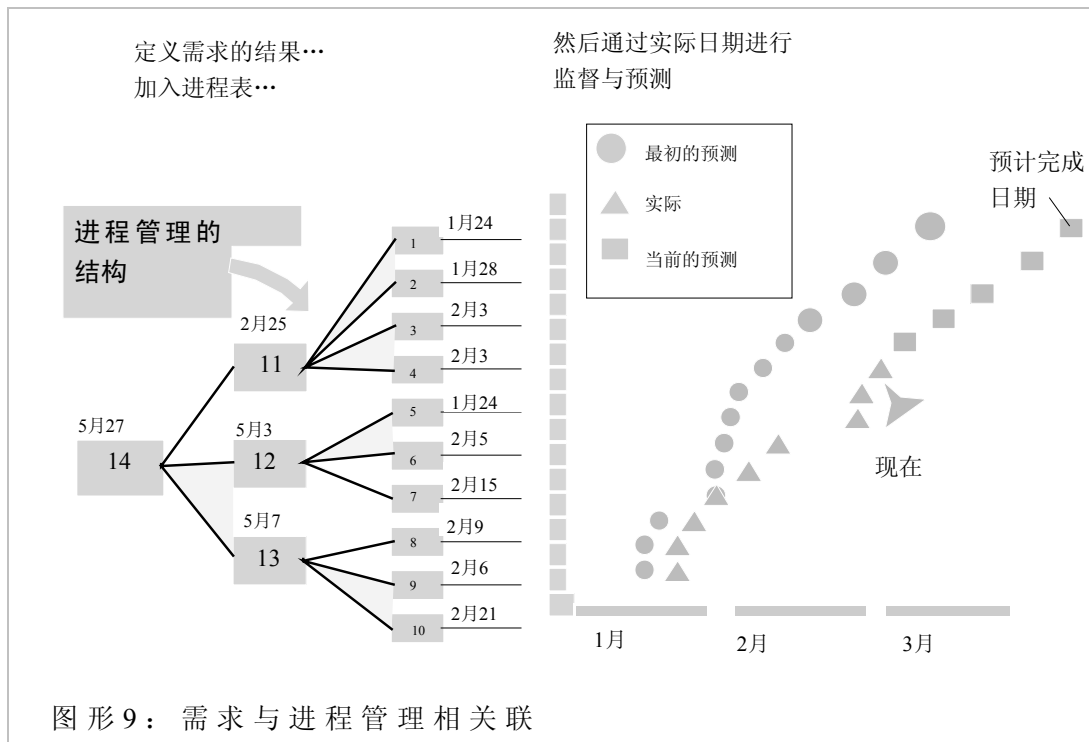
这三种情况虽然都满足了用户所需，然而缺乏实际意义，因此都以失败告终。

“我付了钱，但这不是我想要的”，没有用户愿意这么说。要避免出现这种情况，在进行

项目管理和财务预算时，也必须以需求管理为基础。例如，仅仅完成了一个设计并不意味着工作的结束，只有这个设计充分解决了需求，它才具有里程碑般的意义。同样的，一件产品只有在测试和实际操作中完全满足了需求，已经完全准备好了投入到下一阶段的运营，才意味着这件产品在本阶段工作的结束。

开发进程中的每一块里程碑都意味着需求的解决又前进了一步，这样的每一块里程碑也都是委托商付款的重要参照，产品开发的整个进程都可以通过需求管理进行监控。

里程碑构造机制的基本方法之一就是进程管理，一项需求的满足就意味着一块里程碑的确立。我们应当对用户需求、针对需求而进行的模块设计以及每个子模块的开发进程之间的关联做到心中有数。



对“需求管理的需求”的总结

通过我们对需求管理实际应用的分析，几个关键因素凸现出来。首先，需求管理在开发周期中是自始至终存在的。注意：不要把它简单理解为“需求周期”，需求管理必须始终保持更新，它构成了技术管理的基础。

其次，需求管理同项目管理是密不可分的。如果我们把每一个需求的解决看作一个里程碑，并以此出发对整个开发进程进行监控。

3. 需求的抽象解释

需求模型应当具有以下几个特征：能够在开发周期的初期就建立需求模型、建模的成本很低、易于可视化和优化、本身能体现最终解决方案的特征。也许某些细节是抽象的，但需求管理模型本身必须是完整的。需求模型不应当具有诱导性或倾向性，必须为开发工作留有充分发挥和优化的空间。同时，我们能够通过需求模型对最终产品作出评估。但不幸的是，这些特征本身也不是彼此完全兼容的，很难在一个模型中做到面面俱到。在开发初期针对需求而搭建初期模型是容易的，成本也不会太高，但是这样的模型是很抽象的，绝非等同于最终产品。随后

的产品“原型”或经确证后的模型将更接近于最终产品，但搭建这样的模型会要求更高的成本，同时可供修改的余地也更少。

需求管理的多种模式：

需求管理所要搭建的不同模式是由系统工程所采用的标准决定的。传统上需求管理有两种模式（图 10）—客户模式和系统需求模式。从这两种模式出发的方案应该分别进行设计，不幸的是我们常常将此二者混为一谈。

用户模式着重描述用户面临的问题或希望得到的结果。用户模式的语言组织很象使用场景的实地描述，指明时间，侧重结果。无论谁搭建用户模式，都必须从用户的角度出发。

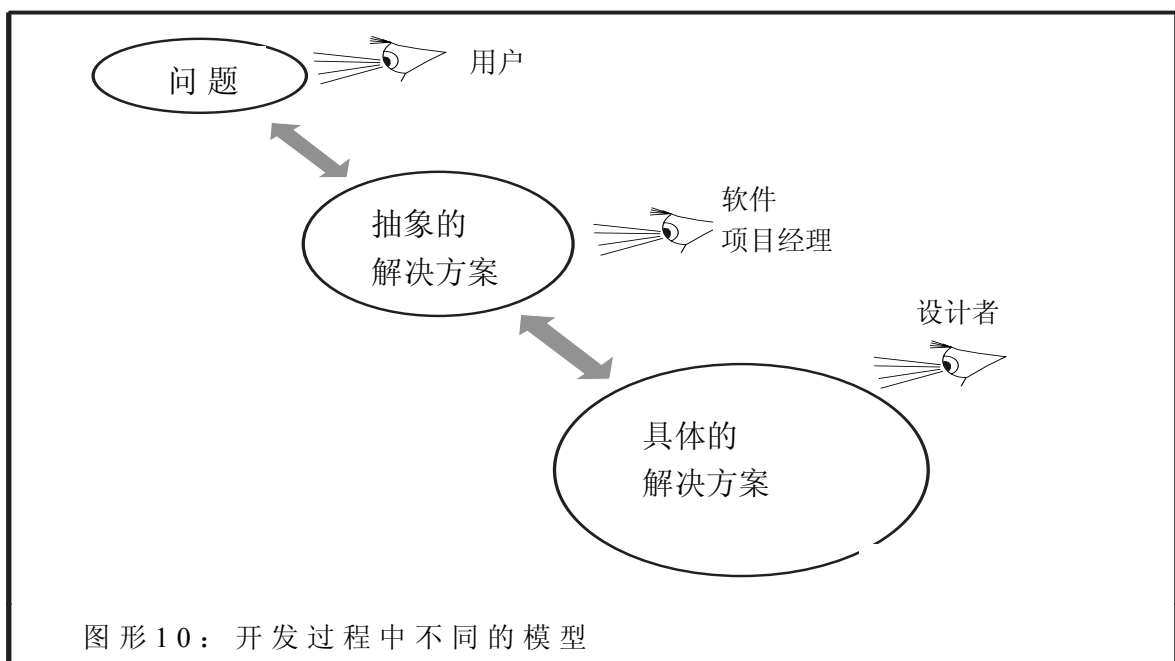
系统需求模式实际是抽象化的解决方案。系统需求模式的语言组织经常运用功能描述或使用详解性的说明文字，事实上功能描述和使用详解正是系统需求模式语言组织的典型风格。

实际上设计方案应当是第三种（非需求）模式，即具体化的解决方案。很明显这种模式已经非常接近于最终解决方案。很多不同的设计方案都能解决用户需求，而在用户需求既定的同时对设计方案作出修改也是切实可行的。在硬件系统设计中，最终进行规模生产的产品体现的往往是第四种模式。

其他设计模式：

搭建多种系统设计模式需要付出相当的工作量，因为每种设计都做到条理清晰并不是件容易的事。如果设计构架和最终方案是一致的，那么工作量可能会减少一些。有些面向对象的设计建议用统一的结构建立系统需求与进行设计。在实际应用当中，设计模式必须采用不同构架，这是因为：

- 在一种模型中同功能无关的需求会引起另一个模型中需求的变化；
- 出于重复利用现存模块的考虑；
- 出于对机构效率的考虑；
- 不同设计方案涉及的步骤要求，我们并不是都要实现；



以上每种因素都会导致设计方案同最初模式不尽相同。如果这些因素并不重要，采用单一的模型也是可行的策略。

我们必须记住，一套完整的系统开发要求有不同侧重点的多种设计模式与之配合，例如：框架配置模式侧重于工作的结果，而工作细化模式则标明了需要完成的各种具体工作。各种模式之间并不是孤立的，在实际需求和各种模型之间存在着多种关系。这些关系表现在：

- 关联性：**不同模式下开发的产品应当具有一致性（系统需求和用户需求）。
- 适用性：**非功能需求同功能需求之间的关联。
- 与测试相关联：**需求管理同被测系统之间的关联（以及产品的关联）。
- 与设计开发相关联：**需求管理同设计模式或产品之间的关联，我们必须清楚每一部分工作同相应需求之间的对应关系。

什么是需求管理

以下段落将通过分析传统需求模型的特点，看看传统需求模型与“对需求管理的需求”是如何发生关联的。

需求模型的特点：

单一的需求本身并没有意义—它必须被放到整体模型中去。一些特性如完整性、一致性也必须从整个模型来理解。需求模型应当是已知系统需求的完整体现，每部分解决方案都是对总体需求一定部分的满足（甚至是充分满足），仅仅解决部分需求是没有意义的。对关键需求的疏忽很可能是灾难性的，试想一架飞机的安全设计不过关将会带来什么样的后果。不同的需求组合起来，构成了一套完整的需求模型，表达了整体系统的部分抽象。用户需求决定了系统设计所要解决的问题，所要带来的结果。可以说，需求管理指明了系统开发所要做和必须做的每一件事，指明了所有设计应该提供的功能和必然受到的制约。

需求的特点：

一个需求必然表明了系统的某一个方面。需求性的描述可以是抽象的，也可以是具体的；它针对的可以是产品本身，也可以是产品开发的方式。

需求性的提出是建立在可验证的基础上的，就是说，我们能够根据需求而通过设定某种检验标准对最终产品进行评估，并给出或是或非的唯一回答。验证的目的是尽可能地发现与需求不一致的地方。在测试中，我们永远不能说产品完全解决了需求，只能说它更加接近于满足需求。其它特点还包括来源，清晰性与优先级。

多种形式的关联：

需求管理的一项重要工作就是在整个项目的不同部分之间建立联系。这也许是在进行系统工程设计时自然而然得到的一种结果。如果我们对需求模式的阐释正确，并对需求与设计的一致性有了确证，那么我们就有了可跟踪性。在出色的系统设计中，系统各部分所存在的各种联系应当是清晰简明的，不能过多。系统的相关性、可追溯性保证了从不同侧重点出发的系统设计能取得一致的结果。举例来说：

- 系统需求满足于用户需求；
 - 设计方案满足于系统需求；
- 一般来说，联接可以被用来显示：

- 非功能性需求同功能性需求适用性之间的关系；
- 测试与相应的设计或需求的关联；
- 开发框架内部的关系（例如目标管理、进度安排、任务细分等）；
- 开发过程中各类信息的存档与交换；
- 对每一需求的验证；
- 对于核心需求的合理阐释。

需求管理的工具：

需求管理所用到的工具必须能够处理和应用于本文所提到的各种需求，应当有助于我们组织需求，开发支持工具以处理与需求相关信息，并使项目的不同部分得到关联。系统工程师始终致力于用简单的工具将需求管理自动化，常用的工具比如附有标注说明的系统发布工具或关系数据库等。

需求管理涉及到一系列复杂的对象，其涉及面向很广，关系到整个设计开发的方方面面。其使用的工具应当提供如下列举的一些功能：

- 在上下文中清楚地表达需求
- 可跟踪性、可适用性、可关联性
- 能够在属性与需求间建立关联
- 能够与描述性信息如研究报告建立关联
- 排序与选取
- 生成用于展示的文本、图形与报告
- 允许加入用户自己领域的专业知识
- 自动建立基线、历史与标识号
- 能够查看不同层次的信息

4.需求管理的本质是什么？

本文论述围绕于需求管理工程。需求管理是开发工作有效进行的保证。很明显需求管理是一种很高层次的系统行为，涉及整个开发过程和产品本身（产品被开发的方式）。

需求管理首先要针对需求做出分析，随后应用于产品并提出方案。需求分析的模型正是产品的原型样本，优秀的需求管理提高了这样的可能性：它使最终产品更接近于解决需求，提高了用户对产品的满意度，从而使产品成为真正优质合格的产品。从这层意义上说，需求管理是产品质量的基础。