

软件需求工程

陈林

lchen@nju.edu.cn

课程简介

- 221340 - 软件工程专业方向选修课
- 学分：2学分
- 周学时：2



课程简介

- 理论课程
 - 软件需求的基本概念
 - 软件需求开发
 - 软件需求管理
 - 软件需求的实施
- 实践课程
 - 开源项目软件需求分析
- 考核
 - 平时100% (实验+考勤)

课程参考书

- Karl E. Wiegiers 等著，李忠利、李淳等译，软件需求，清华大学出版社，2016
 - 金芝、刘璘、金英，软件需求工程：原理和方法，科学出版社
 -
-
- 课程部分讲义来源自北京大学软件与微电子学院《软件需求工程》课程组 (<http://www.ss.pku.edu.cn/SRE/index.html>)

尴尬癌都犯了：Win7 修复补丁错误的补丁，也坏了



达尔文 发布于2017年08月30日 收藏 1 评论 28

ite

HOT

Windows经常需要打各种补丁，但因为还赠个系统环境实在太复杂，而且微软的品控似乎越来越不靠谱，经常会出现各种错误，导致微软不得不随时修复。

但是你能想到吗？一个用来修复此前补丁错误的补丁，居然也倒下了！



作为8月份例行月度升级的一部分，Windows 7系统的**KB4034664**补丁此前惹了不小的麻烦，双屏用户更新后会导致第二台显示器出现画面错误。

微软曾提供了一个手动修复方法，近日又放出了修复补丁**KB4039884**，没想到又带来了更多麻烦。

安装此补丁后，Windows 7会将多个更新文件恢复到旧版本，已解决(回避?)此前存在的问题，但可能是文件版本检查机制不完善，随后用户就会在Windows Update里收到20多个重要、推荐补丁更新的通知。

但事实上，这些补丁早就都安装过了！

错误认识

- **A general statement of objectives is sufficient to begin writing programs — we can fill in the details later**

需求不清楚就进入编程阶段，期望以后修改。更多的情况下是边写边修改

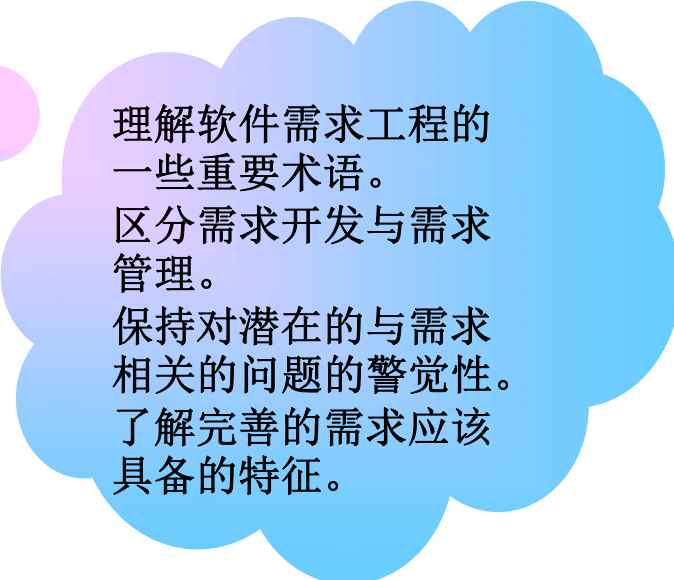
- **Project requirements continually change, but change can be easily accommodated because software is flexible**

软件调节和改变是很灵活的，任何需求的变更都可容易地在软件中反映出来

这些认识多来自极小项目的开发经验，当你面对一个中大型项目时必须彻底改变这些错误观念！

1 软件需求基础知识

- 软件或系统项目涉众(stake holder, 产品或项目相关人员)的利益之间的相互作用在需求过程中表现得最为强烈。项目涉众包括：
 - 客户：为达到其公司的业务目标而投资项目或购买产品。
 - 用户：直接或间接与产品打交道，是客户的一部分。
 - 需求分析员：负责编写需求并传达给开发团队。
 - 开发人员：设计、实现和维护产品。
 - 测试人员：确定产品的行为是否与预计的相一致。
 - 文档编制人员：负责编写用户手册、培训资料和系统帮助。
 - 项目经理：制定项目计划并带领开发人员获得成功。
 - 法律人员：确保产品符合所有相关法规。
 - 生产人员：制造包含该软件的产品。
 - 市场营销：技术支持及其他与产品和客户打交道的人员。



理解软件需求工程的一些重要术语。
区分需求开发与需求管理。
保持对潜在的与需求相关的问题的警觉性。
了解完善的需求应该具备的特征。

1.1 软件需求的定义

- 软件行业存在这样一个问题，用于描述需求工作的术语没有统一的定义。
- 对同一项需求，不同的人会有不同的描述，称其为用户需求、软件需求、功能需求、系统需求、技术需求、业务需求或产品需求。
- 客户对需求的定义，在开发人员看来可能只是高级别的产品概念；而开发人员的需求概念对用户来说也许就是详细的用户界面设计。
- 需求必须被记录成文档，这一点很重要。

1.1.1 对需求的不同解释

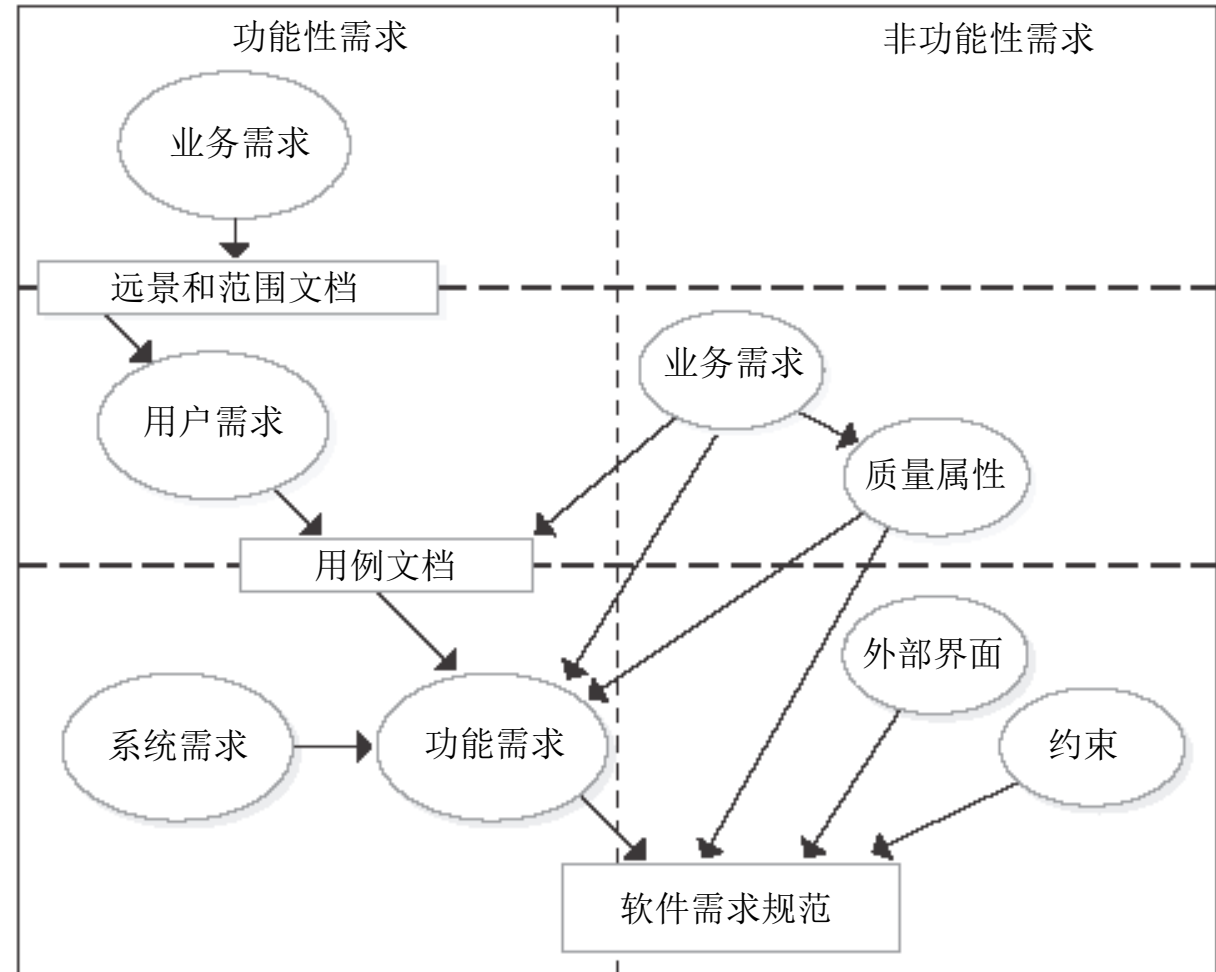
- IEEE的软件工程标准术语表(1990)则将需求定义为:
 - 用户为解决某个问题或达到某个目标而需具备的条件或能力。
 - 系统或系统组件为符合合同、标准、规范或其他正式文档而必须满足的条件或必须具备的能力。
 - 上述第一项或第二项中定义的条件和能力的文档表达。

注意：

不要一厢情愿地认为项目涉众对需求的理解是一致的。应该事先给出定义，才能保证大家谈论的是同一个问题。

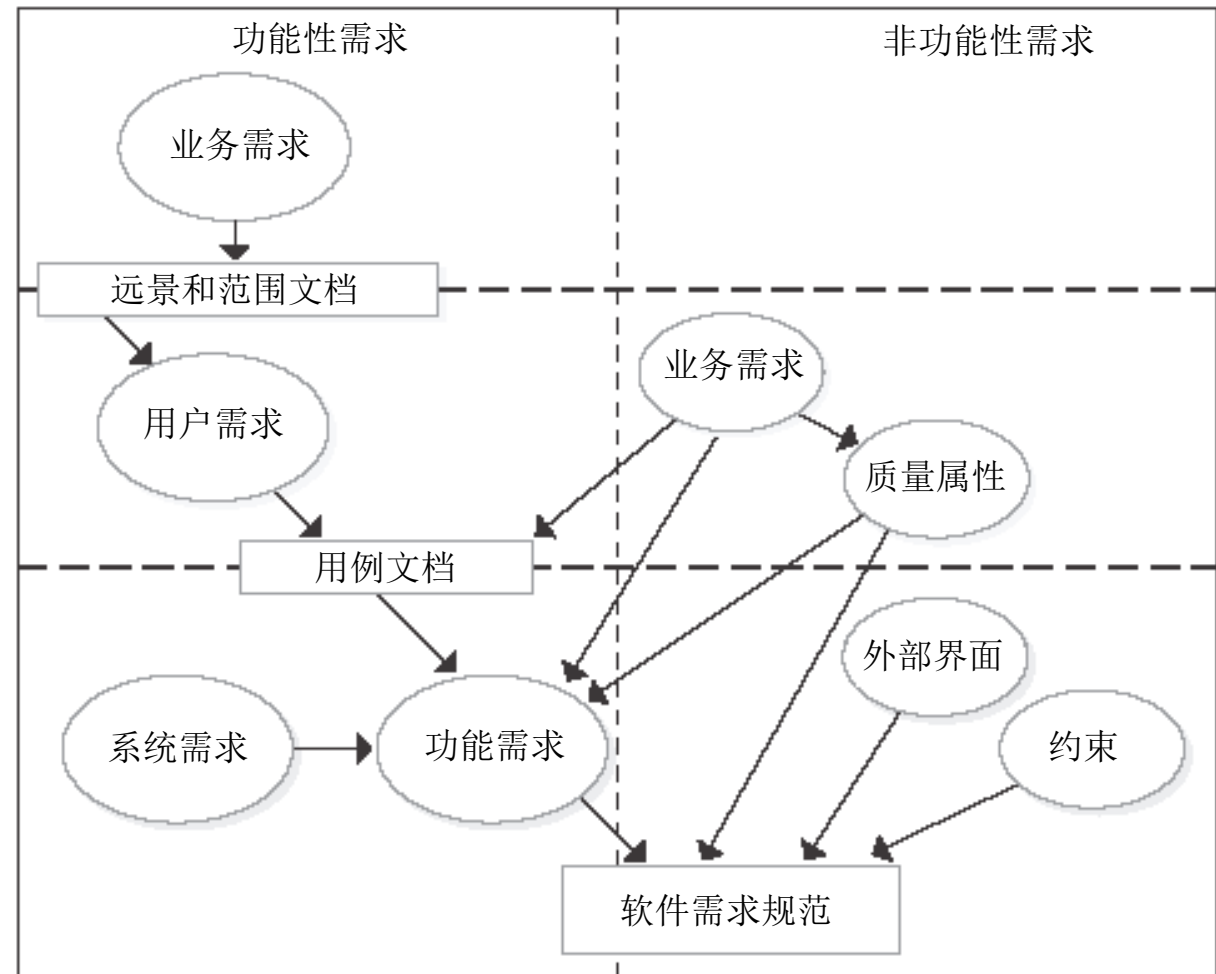
1.1.2 需求的层次

- 软件需求包括3个不同的层次
 - 业务需求
 - 用户需求
 - 功能需求
- 除此之外，每个系统还有各种非功能需求。图中的模型给出了各种需求关系的示意图。



1.1.2 需求的层次

- 业务需求(Business requirement)表示组织或客户高层次的目标。
- 用户需求(user requirement)描述的是用户的目标，或用户要求系统必须能完成的任务。
- 功能需求(functional requirement)规定开发人员必须在产品中实现的软件功能，用户利用这些功能来完成任务，满足业务需求。
- 术语系统需求(system requirement)用于描述包含多个子系统的产品(即系统)的顶级需求。



1.1.2 软件需求的分类

① 业务需求

业务需求(business requirement)反映了组织机构或客户对系统、产品高层次的目标要求，它们在项目视图和范围文档中予以说明

例如航空公司打算把机场柜台工作人员成本降低25%，
要建自助服务终端

1.1.2 软件需求的分类

② 用户需求

用户需求(user requirement)描述了用户使用产品必须要完成的任务，它们在使用实例(use case)和情景描述(scenario)文档中予以说明

例如通过机场自助服务终端 “办理登机手续”

1.1.2 软件需求的分类

③ 功能需求

“一般”意义的需求指的是功能或行为需求，这样的需求通常是和解系统的适当行为(用户需求)相关联，是开发人员必须实现的软件功能。功能需求可以在多种不同的抽象层次上来表达，这使得导出需求过程比较复杂和困难：

- a) Physical behavior
- b) Input-output relationship
- c) Observable states
- d) User interface

1.1.2 软件需求的分类

④ 非功能需求

非功能需求是功能需求的补充，它描述了系统完成功能实现的补充和约束条件。如产品必须遵从的标准、国际规范和合约；外部界面的规范；性能需求如：系统运行速度(Speed)，可靠性(Reliability)，容量(Capacity)，可用性(Availability)，可使用性(Usability)；其它质量属性如：快捷性、简易性、直觉性、健壮性等。在具体操作时，关于可靠性和可用性的规范最为困难，但又是客户最为关心的

1.1.2 软件需求的分类

④ 非功能需求

- a) Response
- b) Accuracy
- c) Frequency
- d) Capacity
- e) Throughput
- f) Defect rates
- g) Modifiability
- h) Supportability

1.1.2 软件需求的分类 – 项目需求

⑤ 设计约束

设计约束是真正意义上的非功能约束，它们约束系统怎样被构建而不是系统做什么。设计约束的一般内容为

- 解系统将在其上运行的目标机器
- 底层的体系结构 – 分布式的或本地的
- 系统运行的内存大小
- 应当采用的任何前端图形用户界面(GUI)程序包
- 系统运行的操作系统
- 应当使用的编程语言
- 其它应集成的软件包如数据库管理系统(DBMS)
- 必须应用的开发标准
- 应采用的设计方法等等

1.1.2 软件需求的分类 – 项目需求

⑤ 设计约束

- a) Language
- b) OS
- c) SW to HW interface
- d) Algorithm
- e) Power
- f) Timing
- g) Memory
- h) Processor utilization
- I) Weight etc

1.1.2 软件需求的分类 – 项目需求

⑥ 商业约束

- 商业约束通常关注的是软件产品完成的时间以及开发费用问题，是客户最为关心的问题。
- 解系统的开发时间和费用与系统功能性、可靠性、可用性等关键需求性能有着必然与复杂的关系。是项目需求与项目管理的结合点。
- 商业约束通常是和需求工程过程同步的，即商业约束是在调查其它需求的同时获得，而且易于识别，不存在技术上的困惑，但却是管理上的难点。
- 商业约束如果不存在一个独立文档，则会出现在业务需求描述中。

1.1.2 软件需求的分类

⑦ 系统需求

对于一个复杂的系统或产品，软件功能需求只是系统需求的一个子集，需要从系统需求中剥离出来。系统需求描述了系统中各个方面的需求，可能包含硬件、软件、其它关联系统，而且系统的功能及非功能描述并不依赖于物理层次，如软件和硬件的划分。系统和软件需求分析人员需要将软件需求部分独立出来。在CMM中这部分工作称为需求分配(requirement allocation)

1.1.2 软件需求的分类

⑧ 开发过程需求(Process Requirements)

A requirements that specifies a need or constraint about how a product will be designed, produced, delivered, or maintained; e.g., the specification of a methodology, manufacturing process, or delivery constraint.

Process requirements are distinguished from product requirements. Process requirements often appear in a Statement of Work rather than in a requirements document.

1.1.2 软件需求的分类

⑧ 开发过程需求(Process Requirements)

- a) Configuration Management
- b) Deployment
- c) Development Methods
- d) Documentation
- e) Manufacturing Methods
- f) Metrics
- g) Quality Assurance
- h) Review processes
- I) Verification Standards

1.1.3 不属于需求的内容

- 需求规格说明中不包括(除已知约束外的)设计和实现的细节、项目的计划信息, 以及测试信息(Leffingwell和Widrig 2000)。
- 项目中通常还包括其他类型的需求, 如开发环境需求, 进度或预算限制, 帮助新用户跟上进度的培训需求, 或者发布产品使其转入支持环境的需求。
- 这些都属于项目需求而不是产品需求。

意想不到的需求



Jessie Lawrence  @hitstreak · Sep 4

Hey @UPS, your driver left this package under our door knob like this and trapped us in our apartment. Had to call maintenance to get out.



客户/用户/开发者的需求观

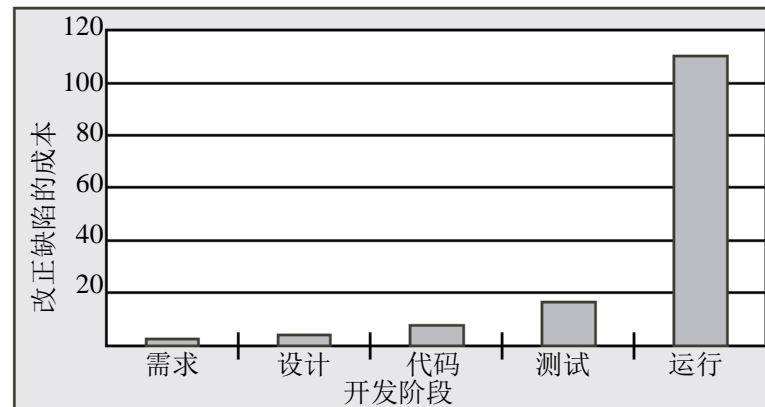
- 客户的需求观在一个比较高的层次上，为产品提供宏观的描述和指导性的框架，是项目的基础；
- 用户的需求观往往代表了产品应该完成的任务及其具有的特性，细节，真实性；
- 开发者的需求观则应是使产品最大限度的满足需求，并最大程度的理解用户的需求。

1.2 所有项目都有需求

- 需求在软件项目中的重要地位：
 - 软件系统开发过程中最难的部分是对要开发什么作出准确的判断。
 - 所有概念性工作中最难的是建立详细的技术需求，包括所有与用户、机器和其他软件系统的接口。
- 在开始开发软件之前，往往无法确定所有的需求。这种情况下，可以采用迭代和增量方法，每次实现一部分需求，得到用户反馈后再进入下一循环。

1.3 优秀的团队遇到糟糕的需求

- 需求问题导致的主要后果是返工——重复做您认为早已做好的事情。
- 返工的成本占了总开发成本的30% ~ 50%(Boehm和Papaccio 1988), 而对于返工的情况, 70% ~ 80%是因需求错误引起的(Leffingwell 1997)。
- 从图1.4可以看出, 在项目末期才发现缺陷, 对其进行改正的成本要比在缺陷刚产生不久时修改的成本高得多。



1.3.1 用户参与不足

- 开发人员往往也不重视用户的参与，原因是他们认为与用户打交道不像写代码那么有趣，或者自以为已经知道了用户想要什么。
- 用户参与不足将导致不能在项目早期及时发现需求中的缺陷，从而延误项目的完成。
- 在整个项目开发过程中，开发团队必须始终与实际用户直接合作。

1.3.2 用户需求扩展

- 由于开发过程中需求的不断发展与增加，项目往往会落后于计划的进度并超出预算。
- 出现这种情况是因为没有依据对需求的规模和复杂度的实际评估来制订计划，而不断修改需求又使情况变得更糟。
- 问题的责任部分在于用户不断提出修改需求的要求，部分在于开发人员处理这种要求的方式。
- 要控制项目范围的改变：
 - 首先应明确项目的业务目标、全局规划、范围、限制、成功标准以及产品的预计用途。
 - 然后参考这一框架对所有新特性和需求变更进行评估。

1.3.3 有歧义的需求

- 歧义表现为同一读者可以对一项需求声明作出多种解释，或者不同的读者对同一需求产生不同的理解。
- 歧义会导致以下几点：
 - 涉众对产品怀有不同的期望。因此最终交付的产品会让部分人感到意外。
 - 有歧义的需求使开发人员的时间浪费在解决无需解决的问题上。
 - 有歧义的需求导致测试人员与开发人员对产品功能的理解不同，从而使测试人员也要浪费时间来原因解决这些差异。
- 消除歧义的办法之一是让代表不同观点的人对需求作正式的检查。

1.3.4 镀金问题

- 开发人员为产品添加了一项需求说明中没有提到的功能，他认为“用户肯定会喜欢的”。这就是所谓的“镀金问题(gold plating)”。
- 开发人员和需求分析员不应擅自添加特性，应该把创意和备选方案提交给客户，让他们做决定。
- 要避免镀金问题，就应该追溯每项功能的来源，弄清楚为什么添加该功能。

1.3.5 过于抽象的需求

- 营销人员或经理经常喜欢只给出一个粗略的说明，他们希望开发人员在开发过程中充实它。
- 这种方式对研究性项目或需求特别灵活的项目或许管用，但是需要紧密合作的团队，而且仅限于开发小型系统(McConnell 1996)。
- 大多数情况下，这种做法的结果是使开发人员受挫，让客户失望。

1.3.6 忽略了某类用户

- 用户所使用的产品特性、产品的使用频率以及用户自身的经验水平不尽相同。
- 因此，多数产品都拥有不同的用户群。如果一开始没能找出产品的所有重要用户群，就会有某些用户需求得不到满足。确定所有用户群后，还要保证获得各类用户的需求。

1.3.7 不准确的计划

- 不能充分理解需求，就会作出过于乐观的估计，最终不可避免地陷入超支的泥潭。
- 造成软件成本估算失败的最主要原因包括频繁变更需求、遗漏需求、未与用户充分沟通、需求的说明不精确，以及对需求的分析不透彻等。

不适当的需求引起的一些风险



公安部打四黑除四害

12-30 12:51 来自Android

+ 关注

#治安提示#【警惕最新诈骗短信！】据网友，短信中HK代表的是换卡、后面的数字代表的是新的sim卡卡号！新卡在骗子手里，用户的卡就不能用了。这样一来，骗子可能轻松转走你绑定的银行卡、支付宝或者微信账户资金。扩散提醒！

< 信息 12520693318038 详细信息

短信/彩信
今天00:30

您好！您已成功订购中国移动无线TV业务，标准资费118元/月。如需退订请编辑短信“HK0D0011550629116”到10086。中国移动

www.10086.cn

腾讯大讲堂

1.4 优质需求过程的好处

- 实现有效的需求工程过程。减少开发后期以及整个维护过程中不必要的返工并可带来极大的回报。
- 下面列出的好处并不能完全量化，但确实存在：
 - 减少需求缺陷
 - 减少开发过程中的返工
 - 减少不必要的特性
 - 降低改进成本
 - 加快开发进度
 - 提高沟通效率
 - 控制需求范围的改变
 - 项目更有序
 - 对系统测试的评估更准确
 - 提高客户和开发人员的满意度

1.5 优秀需求的特点

- 如何才能将好的需求规格说明与那些有问题的区分开来？
- 这一节首先对说明中的单条需求(即需求声明)特点进行讨论，然后将介绍SRS作为整体应具备的特点。
- 如果想知道您的需求是否具备这些特点，最好的办法就是请几位项目相关人员仔细审阅您的SRS。不同的人会发现不同的问题。

1.5.1 需求陈述的特点

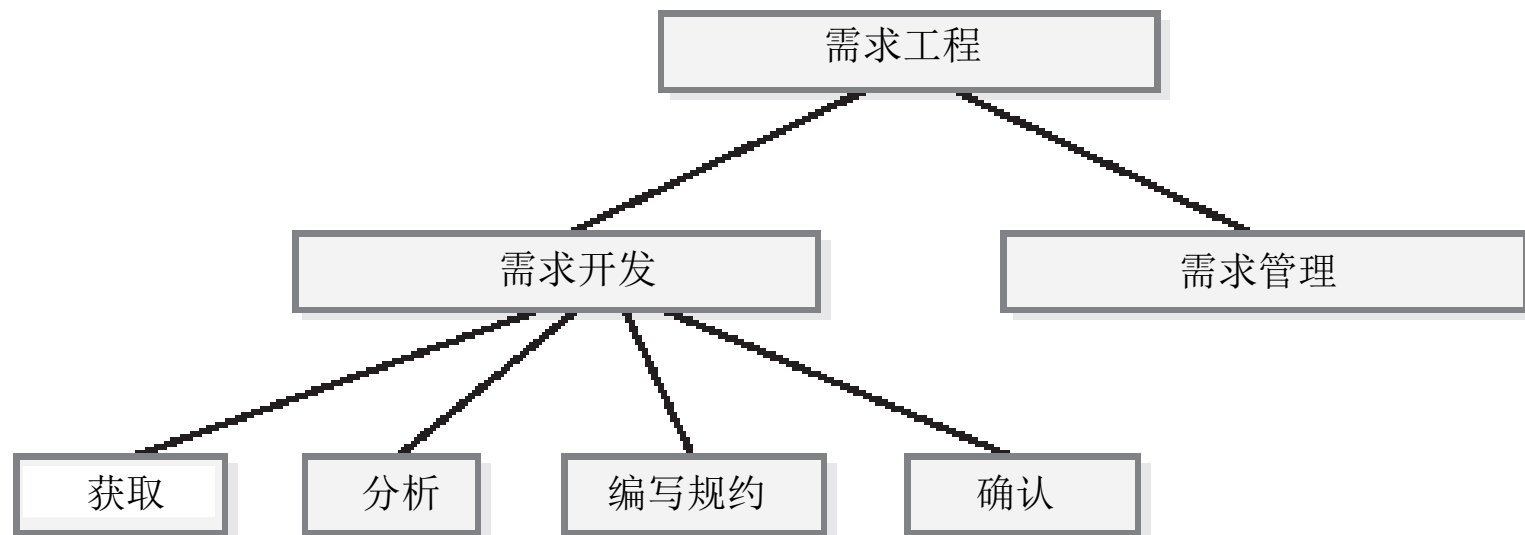
- 每一项用户需求、业务需求和功能需求都应具备下列性质。
 - 完整性
 - 每一项需求都必须完整地描述即将交付使用的功能。
 - 正确性
 - 每一项需求都必须准确地描述将要开发的功能。
 - 可行性
 - 需求必须能够在系统及其运行环境的已知能力和约束条件内实现。
 - 必要性
 - 每一项需求记录的功能都必须是用户的真正需要，或者是为符合外部系统需求或某一标准而必须具备的功能。
 - 有优先次序
 - 为每一项功能需求、特性或用例指定一个实现优先级，以表明它在产品的某一版本中的重要程度。
 - 无歧义
 - 一项需求声明对所有读者应该只有一种一致的解释，然而自然语言却极易产生歧义。
 - 可验证性
 - 看看您能否设计一些测试方法或使用其他验证方法，如检查或演示来判断产品是否正确实现了需求。

1.5.2 需求规格说明的特点

- 需求规格说明中所包含的整体需求集还必须具备下列特性。
 - 完整性
 - 不能遗漏任何需求或必要的信息。
 - 需求遗漏问题很难被发现，因为它们并没有列出来。
 - 一致性
 - 需求的一致性是指需求不会与同一类型的其他需求或更高层次的业务、系统或用户需求发生冲突。
 - 必须在开发前解决需求不一致的问题。
 - 可修改性
 - 必须能够对SRS作必要的修订，并可以为每项需求维护修改的历史记录。
 - 可跟踪性
 - 需求如果是可跟踪的，就能找到它的来源、它对应的设计单元、实现它的源代码以及用于验证其是否被正确实现的测试用例。

1.6 需求的开发与管理

- 需求领域的术语问题，有的作者称其为需求工程(Sommerville和Kotonya 1998)；也有人称之为需求管理(Leffingwell和Widrig 2000)。
- 把软件需求工程划分为需求开发和需求管理。



1.6.1 需求开发

- 需求开发可进一步细分为 (Abran和Moore 2001) :
 - 获取(Elicitation)
 - 分析(analysis)
 - 规格说明(specification)
 - 确认(Validation)

1.6.1 需求开发

- 涵盖了为软件 and 软件相关产品收集、评估和记录需求相关的所有活动，包括：
 - 确定产品将要面对的各类用户。
 - 从各类用户的代表处收集需求。
 - 了解用户的任务和目标，以及这些任务要实现的业务目标。
 - 分析从用户处得到的信息，将用户的任务目标与功能需求、非功能需求、业务规则、解决方案建议及其他无关信息区分开来。
 - 将顶层的需求分配到系统构架内定义好的软件组件中。
 - 了解各质量属性的相对重要性。
 - 协商需求的实现优先级。
 - 将收集的用户需求表述为书面的需求规格说明和模型。
 - 审阅需求文档，以确保在认识上与用户声明的需求相一致。应在开发小组接受需求之前解决所有分歧。

1.6.2 需求管理

- 需求管理的任务是“与客户就软件项目的需求达成并保持一致” (Paulk et al. 1995)。
- 需求管理包括下列活动：
 - 定义需求基线(某一时刻，对特定版本中已达成一致的需求内容的描述)。
 - 审查需求变更请求，评估其可能产生的影响以决定是否批准。
 - 以可控的方式将准的需求变更融入项目中。
 - 保持项目计划与需求的同步。
 - 估计需求变更的影响，在此基础上协商新的需求约定。
 - 跟踪每项需求，找到与其对应的设计、源代码和测试用例(test case)。
 - 在项目开发过程中，始终跟踪需求的状态和变更。

1.6.2 需求管理

- 如图从另一个角度反映了需求开发与需求管理间的区别

