

# **Standard glossary of terms used in Software Engineering**

## **软件工程中使用的标准术语表**



**国际业务分析认证委员会**

**Version 1.0**

**1.0版本**

版权声明

允许对本文档全文复制或摘录，请注明来源。

## （修订记录）

Version	Date	Remarks
1.0	10.09.2011	First version.初版

## 致谢

IQBBA术语翻译参与者（按姓氏拼音排序）

陈智迪、任亮、商超博、石婷、王翊宸、吴剑铭

IQBBA术语QA评审参与者（按姓氏拼音排序）

刘海英、熊晓虹

致谢企业：上海均瑜管理咨询有限公司



# （目录）

## 1. 目录

1. （修订记录） .....	2
1. （引言） .....	5
2. （范围） .....	5
3. （排序） .....	5
4. （参考标准） .....	5
5. （商标） .....	6
6. （定义） .....	7
7. A.....	7
8. B.....	8
9. C.....	11
10. D .....	15
11. E.....	17
12. F18	
13. G 20	
14. H 20	
15. I 20	
16. L 22	
17. M.....	22
18. N 24	
19. O 24	
20. P 25	
21. Q 27	
22. R 28	
23. S 31	
24. T 34	
25. U 35	
26. V 36	
27. W .....	37

2. Annex A(Informative)附件 A（说明） .....	37
---------------------------------------	----

## 1. （引言）

本文档的目的是提供标准化的术语表，供涉及业务分析和需求工程中的IT专业人员使用，以确保他们对基本术语和活动有共同的理解。

## 2. （范围）

本文档介绍了与业务和系统分析、通用软件工程以及相关学科有关的概念、术语和定义。

## 3. （排序）

术语表按字母顺序排列在“定义”这一章节中。有些术语相较于其他同义术语更好，在这种情况下，给出首选术语，同时指出参见的同义术语。

## 4. （参考标准）

截至发布时，所标注的版本是有效的。所有标准都有可能被修订，鼓励各方调研下列标准的最新版的术语以及适用的可能性，IEC和ISO成员维护现行有效的国际标准的登记。

- BS 7925-2:1998. 软件组件测试
- DO-178B:1992.航空机载系统和设备认证，要求和技术方案中软件的注意事项
- IEEE 610.12:1990. 软件工程专业术语标准术语表
- IEEE 829:1998. 软件测试文档标准。
- IEEE 830: 1998: 软件需求规格说明的推荐实践。
- IEEE 1008:1993. 软件单元测试标准。
- IEEE 1012:2004 验证和确认计划标准。
- IEEE 1028:1997. 软件评审和审计标准。
- IEEE 1044:1993. 软件异常标准分类。
- IEEE 1219:1998. SoftwareMaintenance.
- IEEE 1219:1998. 软件维护
- IEEE 1233:1998: 开发系统需求规格说明指南
- IEEE 1362:1998: 信息技术指南 –系统定义
- ISO/IEC 2382-1:1993. 数据处理 –词汇 –第1部分：基本术语。
- ISO 9000:2005. 质量管理体系 –基础和词汇.
- ISO/IEC 12207:1995. 信息技术 –软件生命周期过程。
- ISO/IEC 14598-1:1999. 信息技术 –软件产品评估 –第1部分：总体概述
- ISO 15504-9: 1998. 信息技术 –软件过程评估 –第9部分：词汇
- ISO/IEC 25000:2005 软件工程 –软件产品质量要求与评价（SQuaRE） –SQuaRE指南
- ISO 31000: 风险管理 –实施原则和指南
- IEC 31010: 风险管理 – 风险评估技术
- ISO/IEC 73: 风险管理 – 词汇
- ISTQB测试术语表2.1版本

## 5. （商标）

本文档中使用了下列商标：

- CMM，CMMI和IDEAL是卡内基梅隆大学的注册商标
- BABOK是IIBA商业分析知识体系的注册商标
- BPMN是业务过程管理组织（BPMI）的注册商标，目前与对象管理组织合并
- RUP是Rational 软件公司的注册商标
- SysML是对象管理组织注册商标
- TMMi是TMMi基金会的注册商标
- UML是对象管理组织注册商标

## 6. (定义)

5

5 x 为什么：参见“五个为什么”

### A

#### Acceptance

验收：参见“验收测试”。

#### Acceptance criteria

**验收准则：**组件或系统为了能被用户、客户或其他授权实体接受而必须满足的准出条件。[IEEE 610]

#### Acceptance testing

**验收测试：**针对用户需要、需求和业务过程的正式测试，用以确定系统是否满足验收准则，使用户、客户或其他授权实体能够决定是否接受系统。[IEEE 610]

#### Accuracy

**准确性：**软件产品提供具有所需精度的正确的或相符的结果或效果的能力。[ISO/IEC 25000]

#### Accuracy testing

**准确性测试：**确定软件产品准确性的测试过程。

#### Activity diagram

**活动图：**支持选择、迭代和并发的活动和行为步骤的一种图形化工作流。

#### Ad hoc review

**临时评审：**参见“非正式评审”。

#### Adaptability

**适应性：**软件产品适应不同特定环境，而无需软件本身为这个目的提供其他操作或手段的能力。[ISO/IEC 25000]另请参见“可移植性”

#### Agile manifesto

**敏捷宣言：**支持敏捷软件开发的价值观声明。价值观如下：

- 个体和互动高于流程和工具
- 工作的软件高于详尽的文档
- 客户协作高于合同谈判
- 响应变化高于遵循计划

## **Agile software development**

**敏捷软件开发：**一组基于迭代增量开发的软件开发方法，需求和解决方案通过自组织的跨功能团队间的协作而逐步进化。

## **Agreeing on requirements**

**需求达成一致：**参见“需求验收”。

## **Apprenticing**

**学徒制：**跟从客户学习他的工作的过程。客户教授需求工程师——就像老师和学生一样。

## **Artefact**

**工件：**软件开发过程中产生的结果之一。一些工件(例如，用例、类图和其他UML模型、需求和设计文档)帮助描述软件的功能、架构和设计。其他工件与开发过程本身有关—例如，项目计划、业务案例和风险评估。

## **Assessment**

**评估：**根据给定的质量或验收准则，定性或定量判定产品、服务、活动、过程的价值值的活动。

## **Attractiveness**

**吸引力：**软件产品吸引用户关注的能力。[ISO/IEC 25000] 另请参见“易用性”。

## **Audit**

**审计：**对软件产品或过程的独立评估，以确定符合标准、指导方针、规格说明和/或基于客观标准(包括指定的文档)的过程。[IEEE 1028]:

- (1) 拟生产产品的形式或内容
- (2) 拟生产产品的生产过程
- (3) 衡量标准或指南遵从的程度

## **availability**

**可用性：**组件或系统在需要使用时可操作和可访问的程度。常用百分数表示。[IEEE 610]

# **B**

**BA:** see *Business Analysis, Business Analyst*.

**BA:** 缩写，参见“业务分析”，“业务分析师”。

## **Baseline**

**基线：**一个规范或软件产品已经被正式评审或同意，之后作为进一步开发的基础，并且只能通过正式的变更控制过程进行变更。[IEEE 610]



## Behavioral diagram

**行为视图：**在统一建模语言中，描述系统或业务过程的行为特征的一种图。这包括活动图、状态机图和用例图以及四个交互图。参见“交互图”。

## Benefit

**收益：**交付给利益相关方的价值。[TGilb]

## Best practice

**最佳实践：**在给定的上下文，有助于提高组织绩效的一种更优的方法或创新实践，通常被其他同行组织评为“最佳”。

**bug:** See *defect*.

**缺陷：**参见“缺陷”。

## Business Analysis

**业务分析：**识别业务要求和决策业务问题解决方案所需的一组任务、知识、工具和技术。[BABOK] 另请参见“系统分析”。

## Business Analyst

**业务分析师：**负责识别客户和利益相关方的业务需求，以决策业务问题解决方案的人。[BABOK] 另请参见“系统分析师”。

## Business Case

**业务案例：**业务案例记录启动项目或任务的理由。通过项目的产出与开发新解决方案所需成本的对比，论述项目的业务增值。

## Business domain

**业务领域：**

- (1) 一组类的集合，代表业务模型中正在实现的对象。
- (2) 一般而言，某一业务区域会成为解决方案的主题，或会影响到解决方案。

## Business Goal

**业务目标：**组织的短期或长期目标。

## Business Need

**业务要求：**定义业务分析师必须理解的业务问题或机会，以便提出适当的解决方案。

**Business Process业务过程：**为特定客户或市场设计生产特定输出的一系列活动。

**BPM:** see *Business Process Management*.

**BPM：**缩写，参见“业务过程管理”。

**BPMN:** see *Business Process Modeling Notation*.

**BPMN:** 缩写，参见“业务过程建模标注”。

**BPS:** see *Business Process Simulation*.

**BPS:** 缩写，参见“业务过程模拟”。

### **Business Process Management (BPM)**

**业务过程管理 (BPM)：** 管理方法集中于使组织的各个方面与客户的需要和要求保持一致。业务过程管理试图不断改进流程，因此可以将其描述为“流程优化的过程”。业务过程管理活动可以分为五个类别：设计、建模、执行、监控和优化。

### **Business Process Modeling Notation (BPMN)**

**业务过程建模标注 (BPMN)：** 描述业务过程中步骤的图形符号。业务过程建模标注描述了一个业务过程的端到端的流。该符号是专门设计来协调流程的顺序和在一组相关活动中不同流程参与者之间流转的消息。【BPMN.ORG】

### **Business Process Simulation (BPS)**

**业务过程模拟 (BPS)：** 一种模拟业务过程执行的技术，流程模型的参数允许随时间而变化。

### **Business Sponsor**

**业务发起人：** 向管理决策层提出建议的新项目，让他们可以为企业选择并确定项目组合优先级的人。  
[BABOK]

### **Business Strategy**

**业务策略：** 描述组织方向和为实现其目标将采取的行动的文档或正式声明。业务策略可能源于为支持组织的既定使命而建立的目标。一个典型的业务策略是由三个步骤来制定的：

- (1) 分析，
- (2) 整合，
- (3) 实现。

## C

### Capability Maturity Model (CMM)

**能力成熟度模型（CMM）：**描述有效软件过程关键要素的五层阶段式框架。能力成熟度模型涵盖了计划、工程和管理软件开发和维护的最佳实践。[CMM]另请参见能力成熟度模型集成（CMMI）。

### Capability Maturity Model Integration (CMMI)

**能力成熟度模型集成（CMMI）：**描述有效产品开发和维护过程关键要素的框架。能力成熟度模型集成涵盖了计划、工程和管理产品开发和维护的最佳实践。能力成熟度模型集成（CMMI）是能力成熟度模型（CMM）指定的后续产物。[CMMI]另请参见能力成熟度模型（CMM）。

### Certification

**认证：**确认某个组件、系统或人员符合特定的要求的过程，例如，通过考试(确认)。

### Change Control

**变更控制：**参见“配置控制”。

### Change Control Board

**变更控制委员会：**参见“配置控制委员会”。

### Change List

**变更列表：**参见“变更日志”。

### Change Log

**变更日志：**一份正式文档，包含提交给变更控制委员会以供分析的所有变更请求的清单。变更日志包含所有变更的集合(例如，一个项目)；包括变更请求识别号和/或名称、变更技术可行性、变更成本和收益、变更影响分析、变更计划、测试报告和变更验证。如果流程较早终止(例如，如果没有实现更改)，则不需要包括所有这些内容。

### Change Management

**变更管理：**(1) 将个人、团队和组织从当前状态过渡到理想的未来状态的结构化方法。(2) 对产品或服务实施变更或建议变更的可控方式。另请参见“配置管理”。

### Change Request

**变更请求：**要求新增或修改现有特征、需求、功能正式文档。变更请求应该包含当前解决方案的描述，变更的理由和建议的(期望的)解决方案。

### Change Control Board (CCB): See Configuration Control Board

**变更控制委员会（CCB）：**参见“配置控制委员会”。

## **Changeability**

**易改变性：**软件产品能够实现指定修改的能力。[ISO/IEC 25000] 参见“维护性”。

*译者著：英文原文是“Changeability”，是 ISO9126 中的术语；在新版的 ISO25000 中已经用“Modifiability”替代，这里翻译成易修改性，以保持和 ISO25000 术语的一致性。*

## **Class**

**类：**类描述一组具有相同规范的特征、约束和语义的对象。类是一种分类器，其特征是属性和操作。

## **Class diagram**

**类图：**一种静态结构图，通过显示系统的类、它们的属性、操作（或方法）以及类之间的关系来描述系统的结构。

## **Client**

**客户端：**参见“客户”。

## **Commitment**

**承诺：**满足需求的义务程度。

## **Completeness of a requirement**

**需求完整性：**需求包含所有必要信息的程度。

## **Commercial off-the-shelf software**

**商业现货软件：**参见“现货软件”。

## **Communication diagram**

**协作图：**在统一建模语言中，显示类的实例、实例间相互关系以及实例间消息流的图。

## **Complexity**

**复杂性：**组件或系统的设计和/或内部结构难以理解、维护和验证的程度。

## **Compliance**

**依从性：**软件产品必须遵守标准、约定或法律和类似规定的的能力。[ISO/IEC 25000]

## **Component**

**组件：**（1）一个可以单独测试的最小软件项。（2）组件是某个较大系统或规范中潜在或实际存在的一部分。[TGilb]

## **Component specification**

**组件规格说明：**描述在特定条件下，针对特定输入值（产出）输出值的组件功能以及所需的非功能行为（例如，资源利用）。

## **Component diagram**

**组件图：**在统一建模语言中，描述组成应用程序、系统或重大规划的组件的图形。

### **Composite Structure diagram**

**组合结构图：**在统一建模语言中，描述分类器内部结构的图（例如类、组件或用例），包括分类器到系统其他部分的交互点。

### **Conceptual model**

**概念模型：**描述软件/硬件工艺技术的规格说明的模型。

### **Configuration**

**配置：**由组成部分的数量、性质和互连关系来定义的组件或系统的组成。

### **Configuration Auditing**

**配置审计：**检查配置库中配置项内容的功能，例如是否符合标准【IEEE610】。

### **Configuration Control**

**配置控制：**配置管理的一个要素，包括在正式建立配置标识后对配置项的变更进行评估、协调、批准或否决，以及实施变更。[IEEE 610]

### **Configuration Control Board (CCB)**

**配置控制委员会（CCB）：**一组人员，负责评估、批准或否决对配置项提出的变更，并确保已批准的变更的实现。[IEEE 610]

### **Configuration Identification**

**配置标识：**配置管理的一个组成部分，包括为系统选择配置项并在技术文档中记录它们的功能和物理特性。[IEEE 610]

### **Configuration Item**

**配置项：**参考TMMi术语表，一种硬件、软件或两者的集合，用于配置管理，并在配置管理过程中作为单个实体进行处理。[IEEE 610]

### **Configuration Management**

**配置管理：**用于技术、行政指导和监督的一系列措施：识别和记录配置项的功能和物理特征，控制对这些特征的更改，记录和报告变更处理和实施状态，并验证是否符合指定要求。[IEEE 610]

### **Configuration Management tool**

**配置管理工具：**一种工具，该工具提供对配置项的识别和控制、配置项所有的变更和版本状态，以及由配置项组成的基线的发布。

### **Consistency**

**一致性：**一个组件或系统的文档或部件之间的一致性、标准化和不矛盾的程度。[IEEE 610]

**Constraint**

**约束：**通过限制可接受解决方案的范围来限制一个或一组需求修改的声明。

**Context**

**上下文：**从任何有用的角度来看的系统视图。[TGilb]

**Contractor**

**承包商：**参加“供应商”。

**COTS**

**COTS：**商用现货软件的首字母缩写，参见“现货软件”

**Coverage**

**覆盖：**用百分比表示的程度，一个指定的覆盖项被测试套件执行到的程度。

**Context diagram**

**环境图：**表示系统外部可以与该系统交互的参与者的示意图。

**Criticality of requirements**

**需求的关键性**通过评估未满足需求时的损害来评估需求的风险。

**Customer**

**客户：**作为个人或组织的供给方、卖方或供应商用来称呼产品或服务的当前或潜在购买者或使用者的称谓。

## D

**Data definition**

**数据定义：**给一个可执行语句里的变量赋值。

**Data flow**

**数据流：**数据对象的顺序的和可能的状态变换的抽象表示，对象的状态可以是：已创建、已使用和已销毁。[Beizer]

**Data flow diagram**

**数据流图：**数据对象顺序的和可能的状态变化的图形化表示，其中对象的状态可以是：已创建、已使用或已销毁。

**Decision table**

**决策表：**可用来设计测试用例的表格，表现了输入和/或激励（因）以及相关的输出/行为（果）的组合。

**Data flow analysis:** A form of static analysis based on the definition and usage of variables.

**数据流分析：**基于变量定义和使用的静态分析表格。

## **Defect**

**缺陷：**组件或系统中会导致组件或系统无法执行其必需功能的瑕疵，例如：错误的语句或变量定义。如果在组件或系统运行中遇到缺陷，可能会导致失效。

## **Defect Management**

**缺陷管理：**发现、研究、处置、去除缺陷的过程。包括记录缺陷、分类缺陷和识别其造成的影响。[IEEE1044]

## **Defect Management tool**

**缺陷管理工具：**方便记录和跟踪缺陷和修改状态的工具，这种工具常常具有面向工作流的特性，以跟踪和控制事件的资源分配、更正和再测试，并提供报表。参见“事件管理工具”。

## **Defect Tracking tool**

**缺陷追踪工具：**参见“缺陷管理工具”。

## **Deliverable**

**交付物：**必须交付给作者之外的人的（工作）产品。

## **Delphi method**

**德尔菲法：**一种用于进行交互式预测的结构化协作技术。它包括一个专家小组。[Linstone75]

## **Deming cycle**

**戴明环：**包含四个步骤的解决问题的迭代过程，（计划、实施、检查、行动），通常用于过程改进。[Deming]

## **Deviation**

**偏差：**参见“事件”。

## **Dependency**

**依赖关系：**某种依赖，一组组件对另一组组件的依赖，或者一组需求或其他工件对另一组的依赖。[TGilb]

## **Deployment diagram**

**部署图：**在统一建模语言中用来描述系统运行时的结构的图。

## **Design Thinking**

**设计思维：**这是一个协作的过程，设计师运用他的感性和方法，将人们的需求与可行的技术和商业策略相匹配。简而言之，设计思维将要求转化为需求。这个过程被描述为三个主要阶段：灵感、构思、实现。



## **Documentation testing**

**文档测试：**测试文档的质量，例如用户指南或安装指南。

## **Domain**

**域：**可供有效输入和/或输出值选择的集合。

# **E**

## **Efficiency**

**效率：**在给定条件下根据资源的使用情况，软件产品能够提供适当性能的能力。[ISO/IEC 25000]

## **Elicitation**

**获取：**从他人获得信息的行为。在需求工程的环境中，获取是从利益相关方那里收集需求的过程。

## **Emotional intelligence**

**情商：**识别、评估和管理个人、他人和群体情绪的才能、能力和技能。

## **End user**

**最终用户：**参见“用户”。

## **Enterprise Analysis**

**企业分析：**企业分析由项目前期活动的集合组成，这些活动允许捕获业务的未来展望，从而为给定的计划或长期战略计划的项目需求识别和解决方案设计提供上下文。

## **Entity:**

**实体：**（1）一种要素或一组要素，具有独特的、独立的存在，尽管它不必是物质存在。（2）对某个复杂领域的抽象。

## **Entity-relationship diagram**

**实体关系图：**参见“实体关系模型”。

## **ERD**

**ERD：**缩写，参见“实体关系图”。

## **Entity-relationship model**

**实体关系模型：**数据的抽象和概念性表示。实体关系模型由一组实体组成，实体以属性为特征，通过关系进行链接。

## **ERM**

**ERM：**缩写，参见“实体关系模型”。

## **Error**

**错误：**一种产生错误结果的人类行为。[After IEEE 610]

## **Estimate**

**估算：**关于系统属性的过去、现在或未来的量化等级的数字判定。这包括所有效益和成本属性。估算通常是在直接度量不可能（未来），或不切实际（过去），或不经济（当前水平）的情况下进行的。[TGilb]

## **Evaluation**

**评估：**参见“测试”。

## **Exception handling**

**异常处理：**组件或系统对错误输入的行为反应。该错误输入包括来自人为、来自其他组件或系统以及内部失效等。

## **Exit criteria**

**出口准则：**与利益相关方达成一致的一组通用的和具体的条件，用于允许一个过程正式完成。出口准则的目的是避免一项任务在仍有未解决的部分待完成时被视为已完成。出口准则用于报告和计划何时停止测试。[Gilb and Graham]

## **Extreme Programming**

**极限编程：**在敏捷软件开发中使用的一种软件工程方法，其核心实践是结对编程，进行广泛的代码评审，对所有代码进行单元测试，以及使代码具有简单和清晰性。另请参加“敏捷软件开发”。

# **F**

## **Facilitator**

**推动者：**协助他人完成工作过程（如质量控制或设定目标）的人或小组；凭借他们在这方面受过特别的训练，有资质并且有见识。[TGilb]

## **Failure**

**失效：**组件或系统与其预期交付、服务或结果的偏差。[Fenton]

## **Failure mode**

**失效模式：**失效在物理上或功能上的表现形式。例如，一个处于失效模式的系统的表现可能是运行缓慢、输出错误或执行的彻底中断。[IEEE 610]

## **Failure Mode and Effect Analysis (FMEA)**

**失效模式及影响分析（FMEA）：**一种系统地进行风险识别和分析可能的失效模式的方法，目的在于预防失效的发生。另请参见“故障模式、影响和危害性分析（FMECA）”。

## **Failure Mode, Effects, and Criticality Analysis (FMECA)**

**失效模式、影响及危害性分析：**FMEA 的扩展，除了基本的失效模式和影响分析(FMEA)，还包括危急程度分析，该技术基于失效结果的严重程度将失效模式的可能性制成图表相对而言具有高可能性和严重性的失效模式优先采取补救措施以获得最好的效果。另请参加“失效模式及影响分析(FMEA)”。

## **Fault**

**故障：**参见“缺陷”。

## **Feature**

**特征：**需求文档中指定或隐含的组件或系统的属性（例如可靠性、易用性或设计约束）。[IEEE 1008]

## **Feasibility Analysis**

**可行性分析：**参见“可行性研究”。

## **Feasibility Study**

**可行性研究：**对拟建项目进行分析和评估，以确定该项目（1）在技术上是否可行，（2）在估计成本范围内是否可行，（3）是否有利润。可行性研究几乎总是在涉及大笔资金的情况下进行。也被称为“可行性分析”。

## **Five Why's**

**5个为什么：**一种技术，用于了解单个实例中真正发生了什么。每给出一个答案，进一步提问“为什么”。这有助于识别额外的需求并更深入地分析需求。

## **Free revealing**

**无偿发布：**公司向公众公布其创新过程的结果，而不是申请专利或保守秘密的做法。

## **Formal review**

**正式评审：**以记录程序和要求为特征的评审，如审查。

## **Function**

**功能：**对系统“做什么”的描述。功能具有相应的隐含目的，是系统描述的基本部分：系统由功能属性、性能属性、资源（成本）属性和设计属性组成。所有属性都与定义的指定条件相关。一个功能通常可以分解为一组分层级的子功能。[TGilb]

## **Function point**

**功能点：**一种度量单位，用于表示信息系统向用户提供的业务功能的数量。

## **Function Point Analysis (FPA)**

**功能点分析：**旨在测量信息系统功能大小的方法。测量与技术无关。这种测量可以作为生产力度量、所需资源的估算和项目控制的基础。

## **Functional integration**

**功能整合：**一种集成方法，将组件或系统组合在一起，以便尽早实现基本功能。

### **Functional requirement**

**功能需求：**组件或系统必须实现的特定功能的需求。[IEEE 610]

### **Functionality**

**功能性：**在规定条件下使用软件时，软件产品提供满足显性和隐性需求的能力。[ISO/IEC 25000]

### **Fuzzy Front End**

**模糊的前端：**新产品开发过程的启动阶段。它是组织制定要开发的产品概念并对可能的投资做出决定的关键。[Koen]

## **G**

### **Goal**

**目标：**期望的状态或结果。目标应该是可度量的，并及时确定，以便能够监测进展。

## **H**

### **High-level**

**高阶：**在已定义的系统组件的层次结构中，相对于这些组件的全部已定义集，它更接近顶部而不是底部的位置。[TGilb]

### **Horizontal traceability**

**横向可追溯性：**通过测试文档(例如测试计划、测试设计规格说明、测试用例规范和测试过程规范或测试脚本)对测试级别的需求进行跟踪。

## **I**

### **Idea Generation**

**创意产生：**参见“模糊的前端”。

### **IE table**

**IE表：**缩写，参见“影响估算表”。

### **Impact**

**影响：**在给定条件下，设计思想对需求属性的估算数值或实际数值效应。

### **Impact Estimation Table**

**影响估算表：****影响估算**过程的主要输出规格说明。影响估算表显示了任何一组设计(架构、策略、解决方案)对任何一组需求影响的影响估算度量或实际度量。重点是设计对性能和资源目标的影响。

### **Incremental development model**

**增量开发模型：**在开发生命周期中，项目被分解为一系列增量，每个增量交付了整个项目需求中的一部分功能。需求被划分优先级，并在适当的增量中按照优先级顺序交付。在这个生命周期模型的一些（但不是所有）版本中，每个子项目都遵循一个带有自己设计、编码和测试阶段的“迷你V模型”。

#### **Indicator**

**指标：**一种度量，可用于评价或预测其他度量。[ISO 14598]

#### **Informal review**

**非正式评审：**一种不按照预定义流程的评审，并且没有正式文档化输出。[ISO 20246]

#### **Innovation**

**创新：**更新现存事物的过程。创新改变了系统所依据的价值观。

#### **Input**

**输入：**可被组件读取的变量(存储在组件内部或外部)。

#### **Input value**

**输入值：**输入的实例。参见“输入”。

#### **Inspection**

**审查：**一种基于文档的可视化检查来检测缺陷的同行评审，例如违反开发标准和不符合更高级别的文档。审查是最正式的评审技术，因此总是基于一个有记录的程序。[IEEE 610, IEEE 1028]。另请参见“同行评审”。

#### **Inspection leader**

**审查负责人：**参见“仲裁人”。

#### **Inspector**

**审查员**参见“评审人”。

#### **Installability**

**易安装性：**将软件产品安装在指定环境中的能力。[ISO/IEC 25000] 另请参见“可移植性”。

#### **Integration**

**集成：**将组件或系统组合成更大的组件的过程。

#### **Interaction diagram**

**交互图：**强调对象交互的行为图的子集。这包括通讯、交互概述、序列和时序图。

#### **Interaction overview diagram**

**交互概述图：**活动图的一种变体，概述了系统或业务过程中的控制流。

### **Interoperability**

**互操作性：**软件产品与一个或多个指定组件或系统交互的能力。[ISO/IEC 25000]. 参见功能性.

### **Interview**

**访谈：**一种采访者通过询问回答者以获得特定主题信息的谈话技巧。

### **Iterative development model**

**迭代开发模型：**一个开发生命周期，其中项目通常被分解为大量的迭代。一个迭代是一个完整的开发循环，每次完成一个可执行产品的发布（内部或外部）。每个迭代都是最终开发产品的子集，从一个迭代到另一个迭代，不断发展成为最终产品。

## **L**

### **Lead assessor**

**主任评估师：**主导一场评估的人。在某些情况下，例如CMMI和TMMi，当进行正式评估时，主任评估师必须经过认证和正式培训。

### **Learnability**

**易学性：**用户能够学会软件产品应用的能力。[ISO/IEC 25000] 另请参见“易用性”。

### **Lifecycle model**

**生命周期模型：**将产品或项目的生命周期划分为几个阶段。[CMMI]

## **M**

### **Maintainability**

**维护性：**对软件产品进行修改以修复缺陷、修改以满足新的需求、修改以使未来的维护更容易，或适应变化的环境的难易程度。[ISO/IEC25000]

### **Maintenance**

**维护：**在交付后对软件产品进行修改，以修复缺陷，改进性能或其他属性，或使产品适应修改后的环境。[IEEE 1219]

### **Management review**

**管理评审：**一种对软件获取、供应、开发、运行或维护过程的系统评估，由管理层或管理层代表执行，监控软件的进展，确定计划和日程表的状态，确认需求及其系统分配，或评估管理方法的有效性以达到适用目的。[IEEE 610, IEEE 1028]

## **Maturity**

**成熟度：**（1）一个组织在其过程和工作实践的有效性和效率方面的能力。另请参见“能力成熟度模型”（2）软件产品避免由于软件缺陷而导致失效的能力。[ISO 9126] 另请参见“可靠性”。

## **Maturity level**

**成熟度等级：**过程改进的程度，事先定义过程域，实现全部设定的目标。[TMMi]

## **Maturity model**

**成熟度模型：**元素的结构化集合，这些元素描述了组织中某些方面的成熟度，并且有助于定义和理解组织的过程。成熟度模型经常向优先的改进行动提供一种通用语言、共同愿景和框架。

## **Measure**

**测量：**测量时赋予实体某个属性的数值或类别。[ISO 14598]

## **Measurement**

**测度：**给一个实体分配一个数量或类别来描述该实体的属性的过程。[ISO 14598]

## **Measurement scale**

**测度标度：**约束被执行的数据分析类型的标度。[ISO14598]

## **Metric**

**度量：**测量所使用的方法或者度量标准。[ISO 14598]

## **Milestone**

**里程碑：**项目中定义的（中间）可交付成果和结果应该准备就绪的时间点。

## **Mind-map**

**思维导图：**一种用于表示文字、思想、任务或者其他关联项的图。其内容链接并围绕着中心关键字或思想而展开。思维导图可用于生成、可视化、结构化和归类思想，并且有助于研究、组织、解决问题、做决策和写作。

## **Modeling tool**

**建模工具：**支持软件或系统模型的创建、修改和验证的工具。[Graham]

## **Moderator**

**仲裁人：**负责审查或其他评审过程的领导和主要人员。

## **Module**

**模块：**参见“组件”。

## MoSCoW

**MoSCoW:** MoSCoW优先级排序方法，MoSCoW技术允许通过分配以下术语所表达的适当优先级来对需求进行优先级排序：必须拥有、应该拥有、可以拥有和不会拥有（希望在未来拥有）。

## MOST:

**MOST:** MOST分析模型，一种通过定义大多数项目的属性来执行内部环境分析的技术，以确保项目符合以下四个属性中的每一个：使命、目标、战略、战术。

# N

## Need

**需要/要求:** 特定利益相关方所期望的东西。满足这种需要对某些利益相关方来说是有价值的。一个需要可能不会作为一个正式的需求而被同意，也可能没有排进优先级，以至没有被实际操作（设计和实现）。需要是一个术语，通常用于描述在需求规范执行之前，**利益相关方**对问题的看法。  
[TGilb]

## New Product Development

**新产品开发:** 将一种新产品或服务推向市场的完整过程。

## Non-conformity

**不符合:** 未满足规定的需求。[ISO 9000]

## Non-functional requirement

**非功能需求:** 与功能无关的需求，而是与诸如可靠性、效率、易用性、维护性和可移植性等属性相关的需求。

## NPD

**NPD:** 缩写，参见“新产品开发”。

# O

## Object

**对象:** 在OOAD中，类的实例。

## Object diagram

**对象图:** 在统一建模语言中，在某一时刻描述对象及其关系的图，通常是类图或通讯图的特殊情况。

## Object-oriented analysis and design

**面向对象分析与设计:** 一种将系统建模为一组交互对象的软件工程方法。每个对象表示被建模的系统中的感兴趣的实体，并以它的类、它的状态（数据元素）和它的行为特征。面向对象分析与设计（OOAD）包括面向对象分析（OOA）和面向对象设计（OOD）。OOA应用对象建模技术来分析系统的功能需求。OOD详细说明了分析模型以生成实现规范。



### **Off-the-shelf software**

**现成软件：**一种为通用市场(如为了大量客户)开发的软件产品，并以相同的形式交付给许多客户。

**OOA:** see *Object-oriented analysis and design*.

**OOA:** 缩写，参见“面向对象的分析与设计”。

**OOAD:** see *Object-oriented analysis and design*.

**OOAD:** 缩写，参见“面向对象的分析与设计”。

**OOD:** see *Object-oriented analysis and design*.

**OOD:** 缩写，参见“面向对象的分析与设计”。

### **Operability**

**易操作性：**软件产品使用户能够操作和控制它的能力。[ISO/IEC 25000] 另请参见“易用性”。

**Output:** A variable (whether stored within a component or outside) that is written by a component.

**输出：**由组件写入的变量(存储在组件内部或外部)。

## **P**

### **Pair Programming**

**结对编程：**一种软件开发方法，由两个程序员坐在一台计算机前编写组件的代码行（生产和/或测试）。这意味着正在进行的实时代码审查将被执行。

### **Pareto analysis**

**帕累托分析：**决策中的一种统计技术，用于选择能产生显著整体影响的有限数量的因素。在质量改进方面，大部分问题（80%）是由几个关键原因造成的（20%）。

### **Path:**

**路径：**组件或系统从入口点到出口的一系列事件，如可执行语句。

**PBS:** see *Product Breakdown Structure*

**PBS:** 缩写，参见“产品分解结构”。

### **Peer review**

**同行评审：**由产品生产者的同事对软件工作产品的评审，目的是识别缺陷和进行改进。例如审查、技术评审和走查。

### **Performance**

**性能：**一个系统或组件在给定有关处理时间和吞吐量的约束条件下完成其指定功能的程度。[IEEE 610]参见“效率”。

## **Persona**

**用户画像：**一个虚构的角色，一个原型描述，表示使用最终产品或解决方案的不同类型的用户。用户画像应该代表一群对产品有相同需求、态度、行为或期望的人。

## **PESTLE**

**PESTLE (Political、Economic、Social、Technological、Legal、Environmental)：**通过检查对组织有影响的外部因素来进行外部环境分析的技术。PESTLE分析了以下六个因素：政治、经济、社会、技术、法律、环境。

## **Point of view**

**视角：**对系统或需求的特定看法。

## **Portability**

**可移植性：**软件产品从一种硬件或软件环境转移到另一种环境的难易程度[ISO/IEC 25000]。

## **Priority**

**优先级：**分配给一个条目的（业务）重要性级别，例如缺陷。

## **Process**

**过程：**将输入转换为输出的一组相互关联的活动。[ISO 12207]

## **Process Assessment**

**过程评估：**根据参考模型对组织的软件过程进行严格的评估。[ISO 15504]

## **Process Improvement:**

**过程改进：**旨在提高组织过程的性能和成熟度的活动规划，以及这样规划的结果。【CMMI】

**过程模型：**（1）将相同性质的过程划分为整体模型的框架，例如测试改进模型。（2）与方法无关的开发过程描述。

**过程需求：**与开发过程相关的需求。

**Product:** An output of a process. 【与下面的Product术语重复】

**产品：**过程的输出。

## **Product requirement:**

**产品需求：**与开发过程的产品相关的需求。它们影响产品的质量。

## **Product risk:**

**产品风险：**与产品质量直接相关的风险。参见“风险”。

**Project:**

**项目：**项目是一套独特的协调和可控的活动，在项目的开始和完成日期内，实现符合特定需求的目标，受到时间、成本和资源的限制。[ISO 9000]

**Project risk:**

**项目风险：**与项目管理和控制相关的风险。另请参见“风险”。

**Prototype:**

**原型：**为测试某个概念或过程或作为可复制或学习的东西而建立的早期样本或模型。在需求工程中，原型可以用于需求获取和确认。

**Product:** A product is an output of a process.

**产品：**产品是过程的输出。

**Product Breakdown Structure:**

**产品分解结构：**产品组件的分解。

## Q

**Quality:**

**质量：**一个组件、系统或过程满足规定需求和/或用户/客户的需要和期望的程度[IEEE 610]。

**Quality Assurance:**

**质量保证：**质量管理的一部分，专注于为满足质量需求提供信心。【ISO 9000】

**Quality attribute:**

**质量属性：**一种影响产品质量的功能或特征。[IEEE 610]

**Quality characteristic:**

**质量特性：**参见“质量属性”。

**Quality Management:**

**质量管理：**在质量方面指导和控制一个组织的协同活动。通常包括建立质量方针和质量目标、质量计划、质量控制、质量保证和质量改进。[ISO 9000]

## R

### **Rational Unified Process:**

**统一软件开发过程：**一个专有的可适应迭代软件开发过程框架，由四个项目生命周期阶段组成：初始、细化、构造和交付。

### **Recoverability:**

**易恢复性：**软件产品在出现失效时重新建立指定性能水平和恢复直接受到影响的数据的能力。  
[ISO/IEC 25000] 参见“可靠性”。

### **Redundancy:**

**冗余：**相同的信息在不同的地方多次出现。

### **Release:**

**发布：**发布供客户/最终用户安装和使用的解决方案版本。

### **Reliability:**

**可靠性：**软件产品在规定的条件下，在规定的时间内，或在规定的操作次数下，执行所需功能的能力。  
[ISO/IEC 25000]

### **Replaceability:**

**可替换性：**在相同的环境中，为相同的目的而使用一种软件产品来代替其他指定的软件产品的能力。  
[ISO/IEC 25000] 另请参见“可移植性”。

### **Requirement:**

**需求：**（1）用户解决问题或实现目标所需要的条件或能力。（2）一个系统或系统组件为满足合同、标准、规格说明或其他正式规定的文件而必须满足或拥有的条件或能力。（3）对第（1）或（2）条所述条件或能力的文档表述。  
[IEEE 610]

**RD:** see *Requirements Development*.

**RD:** 缩写，参见“需求开发”。

**RE:** see *Requirements Engineering*.

**RE:** 缩写，参见“需求工程”。

**RM:** see *Requirements Management*

**RM:** 缩写，参见“需求管理”。

### **Requirements acceptance:**

**需求验收：**在所有相关的利益相关方之间，使需求内容和范围是准确和完整的正式协议的过程。  
[BABOK]

**Requirements analysis:**

**需求分析：**一组任务、活动和工具，用于确定所述（获取的）需求是否不清楚、不完整、不明确或矛盾，然后以统一样式记录需求。

**Requirement attribute:**

**需求属性：**关于需求的描述信息，丰富了需求的定义，超出了预期功能的陈述。例如包括来源、基本原则、优先级、所有者、发布号和版本号。[Wiegers]

**Requirements Development:**

**需求开发：**识别、分析和确认需求的活动、任务、技术和工具的集合。包括将需要转换为需求的过程。在CMMI模型中，需求开发是成熟度级别3的工程过程域。

**Requirements elicitation:**

**需求获取：**参见“获取”。

**Requirements Engineering:**

**需求工程：**系统工程和软件工程的一个子学科，与确定硬件和软件系统的目标、功能和约束相关[Laplante]。需求工程学科包括以下子过程：需求获取、分析和协商、规格说明、系统建模、需求确认和需求管理。

**Requirements Management:**

**需求管理：**记录、分析、跟踪、确定优先级、沟通、需求达成共识以及管理需求变更的连续过程。在CMMI模型中，需求管理是成熟度级别2的项目管理过程域。

**Requirements Management tool:**

**需求管理工具：**一个支持记录需求、需求属性（例如优先级、知识责任）和注释的工具，并通过需求层次化和需求变更管理促进可追溯性。一些需求管理工具还提供了静态分析的工具，例如一致性检查和违反预定义的需求规则。

**Requirements model:**

**需求模型：**使用文本和图表表现用户需求。需求模型也可以被称为用户需求模型或分析模型，并且可以补充文本需求规格说明。

**Requirements phase:**

**需求阶段：**软件生命周期中定义和记录软件产品需求的时间周期。[IEEE610]

**Requirements source:**

**需求来源：**衍生某些需求的源头。需求来源可以是利益相关方、文档、业务过程、现有系统、市场等。

**Requirements specification (customer):**

**需求规格说明（客户）：**描述问题领域的规格说明。（客户需求规格说明通常由客户提供，包含了从客户角度对解决方案所需功能的描述。）

**Requirements traceability:**

**需求可追溯性：**定义、捕获和跟踪需求在软件开发环境的其他元素上留下的痕迹，以及这些元素在需求上留下的痕迹的能力。[Pinheiro F.A.C. and Goguen J.A]

**Requirements Traceability Matrix:**

**需求跟踪矩阵：**是一个文档，通常以表格的形式出现，它将任何两个已基线化的文档关联起来，这些文档需要多对多的关系来确定关系的完整性。

**Review:**

**评审：**对产品或项目状态的评估，以确定与计划的差异，并提出改进建议。例子包括管理评审、非正式评审、技术评审、审查和走查。[IEEE 1028]

**Review tool:**

**评审工具：**为评审过程提供支持的工具。典型特征包括评审计划和跟踪支持、沟通支持、协作评审以及收集和报告评审度量的存储库。

**Reviewer:**

**评审员：**识别和描述被评审产品或项目中的异常情况的相关人员。在评审过程中可以选择代表不同的观点和角色的人作为评审员。

**Risk:**

**风险：**（1）不确定性对目标的影响，无论是积极的还是消极的[ISO 31000]。（2）可能会导致负面结果的因素。通常表达成可能性和（负面）影响性[ISTQB]。

**Risk analysis:**

**风险分析：**评估已识别风险以估计其影响和发生概率（可能性）的过程。

**Risk category:**

**风险种类：**参见“风险类型”。

**Risk control:**

**风险控制：**为降低风险到规定的水平或将风险维持在规定的水平而作出决定和实施保护措施的过程。

**Risk identification:**

**风险识别：**使用头脑风暴、检查表和失效的历史记录等技术来识别风险的过程。

**Risk level:**

**风险级别：**风险的重要性，由其特征、影响和可能性所定义。风险级别可用于确定要执行的测试的强度。风险级别可以定性地（例如高、中、低）或定量地表示。

**Risk Management:**

**风险管理：**对识别、分析、确定优先级和控制风险的程序和实践的系统性应用。

**Risk mitigation:**

**风险缓解：**参见“风险控制”。

**Risk type:**

**风险类型：**通过一个或多个通用因素（如风险的质量属性、原因、位置或潜在影响）对风险进行分类。

**RTM:**

**RTM：**缩写，参见“需求跟踪矩阵”

**RUP:**

**RUP：**参见“统一软件开发过程”。

## S

**Safety:**

**安全性：**在特定的使用范围内，软件产品具有对人、业务、软件、财产或环境造成损害的风险水平控制在可接受范围内的能力。[ISO/IEC 25000]

**Scalability:**

**可扩展性：**软件产品可升级以容纳更多负载的能力。[Gerrard]

**Scenario:**

**场景：**（1）导致特定结果的预期行动、事件或情况。（2）指定实体（如系统与参与者）之间有序的交互的序列。（3）在UML中：用例的执行跟踪。

**Scope:**

**范围：**事物的影响程度。范围可以应用于任何东西，如规格说明，或指定的系统或项目。[TGilb]

**Scribe:**

**记录员：**在评审会议中将每一个提到的缺陷和过程改进建议记录在日志表上的人。记录员应确保日志表的可读性和可理解性。

**Scrum:**

**Scrum：**用于管理敏捷软件开发中通用的项目的迭代增量框架。另请参见“敏捷软件开发”。

**Security:**

**信息安全性：**软件产品的一种属性，它能够防止未经授权的对程序 and 数据的访问，无论是偶然的还是故意的。[ISO/IEC 25000]。另请参见“功能性”。

**Sequence diagram:**

**序列图：**在UML中，它是一种行为的结构化表示，表现为一段时间内一系列连续的步骤。序列图是一种交互图，它显示了流程之间是如何操作的，以及顺序是什么。

**Signoff:**

**签收：**参见“需求验收”。

**SME:**

**SME：**缩写，参见“领域专家”。

**Software:**

**软件：**计算机程序、过程和可能与计算机系统运行相关的文档和数据[IEEE 610]。

**Software Failure Mode and Effect Analysis (SFMEA):**

**软件失效模式及影响分析（SFMEA）：**参见“失效模式及影响分析（FMEA）”。

**Software Failure Mode, Effects, and Criticality Analysis (SFMECA):**

**软件失效模式、影响及危害性分析（SFMECA）：**参见“失效模式，影响及危害性分析（FMECA）”。

**Software lifecycle:**

**软件生命周期：**从构思软件产品开始到软件终止使用时为止的一段时间。软件生命周期通常包括概念阶段、需求阶段、设计阶段、实现阶段、测试阶段、安装和验收阶段、运行和维护阶段，有时还包括退役阶段。注意，这些阶段可以重叠，也可以迭代地执行。

**Software quality:**

**软件质量：**一个软件产品的功能和特征的总和，它决定了软件产品满足明示或隐含需求的能力[ISO/IEC 25000]。

**Software quality characteristic:**

**软件质量特性：**参见“质量属性”。

**Software Process Improvement:**

**软件过程改进：**用于改进组织的软件过程和结果的表现和成熟度活动的程序[CMMI]。

**Solution:**

**解决方案：**（1）解决方案是需求的实现。（2）一种设计思想，如果得到实施，将有望部分或完全满足一组属性的需求；解决一个（确定的）问题[TGilb]。

**Solution model:**

**解决方案模型：**从系统的不同视图描述解决方案区域的模型。



**Solution validation:**

**解决方案的确认：**解决方案确认是向利益相关方和发起人解释解决方案适用性的活动。

**Subject Matter Expert:**

**主题专家：**在履行一项专门的工作、任务或技能方面表现出最高水平的专家[Pyzdek, Thomas and Paul A. Keller]。

**Solution Specification:**

**解决方案规格说明：**也称为功能规格说明、系统需求规格说明或软件需求规格说明。描述解决方案领域。

**Specification:**

**规格说明：**在理想情况下，以完整、精确和可验证的方式规定了部件或系统的要求、设计、行为或其他特性的文档，通常还规定了确定这些规定是否得到满足的规程[IEEE 610]。

**Stability:**

**稳定性：**软件产品避免软件修改带来的意外影响的能力[ISO/IEC 25000]。另请参见“维护性”。

**Stakeholder:**

**利益相关方（干系人方）：**任何对某个IT项目感兴趣的人。项目利益相关方是积极参与项目的个人和组织，或者他们的利益可能会因为项目的执行或项目的完成而受到影响。利益相关方可以对直接的系统运行特性以及长期的系统生命周期考虑因素(例如可移植性、生命周期成本、环境考虑因素和系统退役)进行控制。[TGilb]

**Standard:**

**标准：**正式的，可能是强制性的，开发和使用的的一组需求，规定了一致的工作方法或提供指南（例如ISO/IEC标准，IEEE标准和组织的标准）[CMMI]。

**State machine:**

**状态机：**由有限数量的状态、这些状态之间的转换和动作组成的行为模型，类似于流程图。

**State transition:**

**状态转换：**组件或系统的两种状态之间的转换。

**State machine diagram:**

**状态机图：**参见“状态机”。

**Storytelling:**

**讲故事：**一种说服技巧，用来说服另一方相信叙述人的论点。故事是基于假设或在研究阶段所经历的真实情况。故事围绕着产品、用户和用户体验展开。

**Structure diagram:**

**结构视图：**描述与时间无关的规范元素的一种图表。这包括类图、组合结构图、组件图、部署图、对象图和套件图。

**Suitability:**

**适合性：**软件产品为特定任务和用户目标提供一套适合功能的能力[ISO/IEC 25000]。另请参见“功能性”。

**SWOT:**

**SWOT：**一种分析方法，即从组织绩效一面确定组织的优劣和劣势，同时识别机遇和针对弱点的内、外部威胁。。SWOT的四个属性是：优势，劣势，机会，威胁。

**SysML:**

**系统建模语言：**一种用于系统工程应用的通用建模语言。它支持广范系统以及复杂系统的规格说明，分析、设计、验证和确认。

**System:**

**系统：**为完成特定功能或一组功能而组织起来的组件集合[IEEE 610]。

**System Analysis:**

**系统分析：**专注于将业务需求转化为系统需求的一组活动、方法、技术和工具。它描述了一个系统及其对环境的限制，并提供了对环境和系统需求的良好理解。

**System Analyst:**

**系统分析师：**以技术为导向的人，研究给定的业务问题，计划软件解决方案，推荐软件和系统，以及协调开发以满足业务或其他需求。系统分析师的任务是将业务需求转化成技术规格说明。

**System boundary:**

**系统边界：**系统与其上下文之间的边界。

**System Interface Analysis:**

**系统接口分析：**将新解决方案集成到业务和技术环境中所需的工作范围的表达。

**Systems Modeling Language:**

**系统建模语言：**参见“SysML”。

## T

**Technical review:**

**技术评审：**一种同行小组讨论活动，其重点是就应采取的技术方法达成共识[Gilb and Graham, IEEE 1028]。另请参见“同行评审”。

**Testability:**

**可测试性：**软件产品的一种能力，表示经过修改的软件可被测试的能力[ISO/IEC 25000]。另请参见“维护性”。

**Testable requirements:**

**可测的需求：**对需求可测试程度的一种说明，表示是可依据需求进行测试设计（以及后续的测试用例）和执行测试，以及判断是否满足需求[IEEE 610]。

**Timing diagram:**

**时序图：**在UML中，描述分类器实例或角色随时间变化的状态或条件的图。

**Traceability:**

**可追溯性：**识别文档和软件中关联项的能力，例如相关测试的需求。另请参见水平可追溯性、垂直可追溯性。

**Trend:**

**趋势：**市场（或具体产品或服务）随时间向特定方向移动的势头[G. Fontanills and T. Gentile]。

**Trial and Error:**

**试错法：**通过生成/原型化解决方案、测试它们并从自己的错误中学习来获得知识的过程。使用原型来执行解决方案的测试—因此原型的创建应该是容易并且便宜的。

## U

**UML:**

**UML：**缩写，参见“统一建模语言”

**Unified Modeling Language:**

**统一建模语言：**软件工程领域的一种标准化的通用建模语言。统一建模语言包括一组图形符号技术，用于创建软件密集型系统的可视化模型，如用例图、活动图、类图等。

**Understandability:**

**易理解性：**软件产品使用户能够了解软件是否适合，以及如何在特定的任务和使用条件下使用的能力。[ISO/IEC 25000]另请参见“易用性”。

**Usability:**

**易用性：**在特定条件下使用时，软件能够被理解、学习、使用和吸引用户的能力。[ISO/IEC 25000]

**Use case:**

**用例：**施动者与组件或系统之间的对话中的一系列事务，施动者可以是用户或任何可以与系统交换信息的事物。

**Use Case diagram:**

**用例图:** 在统一建模语言中，显示用例、参与者及其关系的图。

**User:**

**用户:** 使用软件产品的人。

**User Innovation:**

**用户创新:** 由消费者（个人最终用户或组）进行的创新。创新的创造者是最终用户，他们在使用现场开发或改进已获得的产品和服务。

## V

**Value:**

**价值:** 可感知的收益，对于一个或多个利益相关方来说，它是系统属性潜在的收益。价值与系统改进不是线性关系的：例如，属性级别上的一个小变化可以为一组利益相关方以相对低的成本增加巨大的能感知到的价值。价值是定义的系统组件或系统状态在特定条件下对特定利益相关方感知到的有用性、价值、效用或重要性。价值是相对于利益相关方而言的：它不是绝对的。[TGilb]

**Validation:**

**确认:** 通过检查和提供客观证据来证明需求已满足特定使用或应用。[ISO 9000]。

**Vendor:**

**供应商:** 提供解决方案的个人、团体或组织。

**Verification:**

**验证:** 通过检查和提供客观证据来证明指定的需求已经实现[ISO 9000]。

**Version:**

**版本:** 某个事物的特定形式或变体。

**Version Control:**

**版本控制:** 参见“配置控制”。

**Vertical traceability:**

**纵向可追溯性:** 从不同层级的开发文档到组件的需求跟踪。

**Vision:**

**愿景:** 意象化（形象化）的项目可交付成果，作为所述需求或问题的解决方案。

**V-model:**

**V模型:** 描述从需求规格说明到维护的软件开发生命周期活动的框架。V模型举例说明了如何将测试

活动集成到软件开发生命周期的每个阶段中。

## W

### **Walkthrough:**

**走查:** 文档作者为收集信息和建立对其内容的共同理解一步步进行演示[Freedman and Weinberg, IEEE 1028]。另请参见“同行评审”。

### **WBS:**

**WBS:** 参见“工作分解结构”。

### **Workshop:**

**工作坊:** 一种专注于特定（预先定义并向与会者宣布的）主题的会议，通常涉及代表不同区域或/和领域的利益相关方，周期短而密集。

### **Work Breakdown Structure:**

**工作分解结构:** 工作要素及其彼此之间的关系，以及与最终产品之间的关系的安排[CMMI]。

## 7. 附件 A（说明）

来源的索引；在构建本术语表时使用了下列非规范性的来源：

[BABOK]国际商业分析研究所（2006），业务分析知识体系的指南，版本1.6

[Beizer] B. Beizer（1990），软件测试技术，出版社，国际标准书号0-442-20672-0

[BPMN.ORG]对象管理组织/业务过程管理组织

[CMM]M. Paulk, C.Weber, B. Curtis and M.B. Chrissis（1995），能力成熟度模型，改进软件过程的指导方针，出版社，国际标准书号0-201-54664-7

[CMMI]M.B. Chrissis, M. Konrad and S. Shrum（2004），能力成熟度模型集成，过程集成和产品改进指南，出版社，国际标准书号0-321-15496-7

[Deming]D. W. Edwards（1986），转危为安，麻省理工学院高级工程研究中心，国际标准书号0-911379-01-0

[Fenton] N. Fenton（1991），软件度量：一种严格的方法，查普曼和霍尔公司，国际标准书号0-53249-425-1

[Fewster and Graham] M. Fewster and D. Graham（1999），软件测试自动化，有效使用测试执行工具，出版社，国际标准书号0-201-33140-3

[Freedman and Weinberg] D. Freedman and G. Weinberg (1990), 走查, 审查, 以及技术评审, 多家出版, 国际标准书号0-932633-19-6

[G. Fontanills and T. Gentile] G. Fontanills and T. Gentile (2001), 市场开始课程, George Fontanills, Tom Gentile, John Wiley and Sons Inc.

[Gerrard] P. Gerrard and N. Thompson (2002), 基于风险的电子商务测试, 多家出版, 国际标准书号0-932633-19-6

[Gilb and Graham] T. Gilb and D. Graham (1993), 软件审查, 出版社, 国际标准书号0-201-63181-4

[Gilb and Brodie RQNG] T. Gilb and L. Brodie (2010), 根本性错误是什么? 改进我们在需求规格说明中获取价值的方法

[Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), 软件测试基础, 汤姆森学习, 国际标准书号978-1-84480-355-2

[Koen] Koen et al. (2001), 为“模糊前端”提供清晰和通用的语言。研究技术管理, 44 (2), pp. 46-55

[Laplante] Laplante, Phil (2009) 软件和系统的需求工程 (第1版), 华盛顿州雷德蒙德: CRC出版社, 国际标准书号1-42006-467-3

[Pinheiro F.A.C. and Goguen J.A.] Pinheiro F.A.C. and Goguen J.A., An object-oriented tool for tracing requirements, in: *IEEE Software* 1996, 13(2), pp. 52-64

[Pinheiro F.A.C. and Goguen J.A.] Pinheiro F.A.C. and Goguen J.A., 用于跟踪需求的面向对象工具, 摘自IEEE 软件 1996, 13 (2), pp. 52-64

[Pyzdek, Thomas and Paul A. Keller] Pyzdek, Thomas and Paul A. Keller (2009). The Six Sigma Handbook, Third Edition. New York, NY: McGraw-Hill. ISBN 0071623388.

[Pyzdek, Thomas and Paul A. Keller] Pyzdek, Thomas and Paul A. Keller (2009) 六西格玛手册, 第3版。纽约, 纽约州: 麦格劳-希尔。国际标准书号0071623388

[TMap] M. Pol, R. Teunissen, E. van Veenendaal (2002), 软件测试, TMap方法指南, 出版社, 国际标准书号0-201-745712

[TGilb] 参见: <http://gilb.com>, Planguage概念术语表