### 需求规格说明书(ISO标准版)

编者说明：

当需求调查、分析工作告一段落时，你就需要将这些需求进行规格化描述，整理成文，即软件需求规格说明书，也就是SRS。这是在软件项目过程中最有价值的一个文档。ISO所提供的标准虽然已经时间久远，但还是颇具参考价值的。

1．引言

1.1编写的目的

[说明编写这份需求说明书的目的,指出预期的读者。]

1.2背景

a. 待开发的系统的名称;

b. 本项目的任务提出者、开发者、用户；

c. 该系统同其他系统或其他机构的基本的相互来往关系。

1.3定义

[列出本文件中用到的专门术语的定义和外文首字母组词的原词组。]

1.4参考资料

[列出用得着的参考资料。]

2．任务概述

2.1目标

[叙述该系统开发的意图、应用目标、作用范围以及其他应向读者说明的有关该系统开发的背景材料。解释被开发系统与其他有关系统之间的关系。]

2.2用户的特点

[列出本系统的最终用户的特点，充分说明操作人员、维护人员的教育水平和技术专长，以及本系统的预期使用频度。]

2.3假定和约束

[列出进行本系统开发工作的假定和约束。]

3．需求规定

3.1对功能的规定

[用列表的方式，逐项定量和定性地叙述对系统所提出的功能要求，说明输入什么量、经怎么样的处理、得到什么输出，说明系统的容量,包括系统应支持的终端数和应支持的并行操作的用户数等指标。]

3.2 对性能的规定

3.2.1精度

[说明对该系统的输入、输出数据精度的要求，可能包括传输过程中的精度。]

3.2.2时间特性要求

[说明对于该系统的时间特性要求。]

3.2.3灵活性

[说明对该系统的灵活性的要求，即当需求发生某些变化时，该系统对这些变化的适应能力。]

3.3输入输出要求

[解释各输入输出数据类型，并逐项说明其媒体、格式、数值范围、精度等。对系统的数据输出及必须标明的控制输出量进行解释并举例。]

3.4数据管理能力要求（针对软件系统）

[说明需要管理的文卷和记录的个数、表和文卷的大小规模，要按可预见的增长对数据及其分量的存储要求作出估算。]

3.5故障处理要求

[列出可能的软件、硬件故障以及对各项性能而言所产生的后果和对故障处理的要求。]

3.6其他专门要求

[如用户单位对安全保密的要求，对使用方便的要求，对可维护性、可补充性、易读性、可靠性、运行环境可转换性的特殊要求等。]

4．运行环境规定

4.1设备

[列出运行该软件所需要的硬设备。说明其中的新型设备及其专门功能，包括：

a. 处理器型号及内存容量

b. 外存容量、联机或脱机、媒体及其存储格式，设备的型号及数量

c. 输入及输出设备的型号和数量，联机或脱机；

d. 数据通信设备的型号和数量

e. 功能键及其他专用硬件]

4.2支持软件

[列出支持软件，包括要用到的操作系统、编译程序、测试支持软件等。]

4.3接口

[说明该系统同其他系统之间的接口、数据通信协议等。]

4.4控制

[说明控制该系统的运行的方法和控制信号，并说明这些控制信号的来源。]

### 需求规格说明书(Volere版)

编者说明：

Atlantic System Guild（[www.atlsysguild.com](http://www.atlsysguild.com)）公司所提供的Volere需求过程与软件需求规格说明书模板则充分利用了现代软件工程思想与技术，是一个十分实用、完善的SRS模板。其所提供的Volere需求记录卡也十分实用，强烈推荐。

*注：从Atlantic System Guild公司网站www.atlsysguild.com上获得，并稍做修改*

1.产品的目标

1.1 该项目工作的用户问题或背景

[对引发开发任务的工作和情况的描述。同时也应描述用户希望用将要交付的软件来完成的工作。]

*[该节内容为该项目提供了合法的理由，你应该考虑用户的问题是否严重，是否应该解决和为什么应该解决。]*

1.2 产品的目标

[用一句话或很少的几句话来说明“我们希望该产品做什么？”换言之，即开发该产品的真正原因。

*[项目如果没有一个表述清晰、易于理解的目标，就会迷失在产品开发的沙漠中。产品必须带来某种优势。典型的优势是产品会增加组织在市场上的价值，减少运作成本，或提供更好的客户服务。这个优势应该是可度量的，这样才能够让您确定交付的产品是否达到目标。]*

2.客户、顾客和其它风险承担者

2.1 客户是为开发付费的人，并将成为所交付产品的拥有者

[这一项必须给出客户的姓名，三个以内是合理的。]

*[客户最终将接受该产品，因此必须对交付的产品满意。如果你无法找到一个客户的姓名，那么也许你就不应该构建该产品。]*

2.2 顾客是将花钱购买该产品的人

[也给出姓名和相关的信息]

2.3 其它风险承担者

[其他的一些人或组织的名称，他们或者受到产品的影响，或影响产品。]

1. 经理或项目负责人；
2. 业务领域专家；
3. 技术人员；
4. 系统开发者；
5. 市场人员；
6. 产品经理；
7. 测试和质量保证人员；
8. 审查员，诸如安全审查员或审计人员；
9. 律师；

10）易用性专家；

11）你所处行业的专业人员。

3.产品的用户

3.1 产品的用户

[产品的潜在用户或操作员的列表。针对每种类型的用户，提供以下信息：]

1. 用户分类
2. 用户工作的任务；
3. 主要相关的经验；
4. 技术经验；
5. 其他用户特征：包括身体、智力、工作态度、对技术的态度、教育程度、语言技能、年龄、性别等。

*[用户是为了完成工作而与产品交互的人，你了解用户，就越可能提交适合用户工作方式的产品。]*

3.2 对用户设的优先级

[在每类用户后面附上一个优先级，这区别了用户的重要性和优先地位：]

1. 关键用户：对产品的后续成功至关重要；
2. 次要用户：他们使用产品，但对产品的长期成功并无影响；
3. 不重要的用户：不常用、未授权和没有技能的用户。

*[如果认为某些用户对产品或组织更重要，那么应该写明，因为它会影响你设计产品的方式。]*

4.需求限制条件

4.1 解决方案限制条件

[此处明确了限制条件，它们规定了解决问题必须采取的方式。您可以认为它们是指令式的解决方案。仔细描述该解决方案，以及测试是否符合的度量标准。如果可能，您应该解释使用该解决方案的原因。]

*[换一句话说，就是要求软件解决方案满足哪些限制条件！]*

4.2 实现环境

[此处描述产品将被实施的技术环境和物理环境。]

*[该环境也将成为设计解决方案时的限制条件之一。]*

4.3 伙伴应用

[此处描述那些不属于产品的一部分，但产品却又必须与其协作的应用程序。]

4.4 COTS

[此处描述实现产品需求所必须使用的COTS(商业组件)。]

4.5 预期的工作场地环境

[此处描述用户工作和使用该产品的工作场地。此处应该描述任何可能对产品设计产生影响的工作场地特征。]

4.6 开发者构建该产品需要多少时间

[任何已知的最后期限，或商业机会的时限，应在此处说明。]

4.7 该产品的财务预算是多少

[该产品的预算，以金钱的形式或可得资源的形式说明。]

5.命名标准和定义

[定义项目中使用到的所有术语，包括同义词。这里的内容就是一个字典，包括在需求规格说明书中使用的所有名称的含义。这个字典应该使用你的组织或行业使用的标准名称。这些名称也应该反映出在工作领域中当前使用的术语。该字典包括项目中用到的所有名称。请仔细地选择名称，以避免传达不同的、不期望的含义。为每个名字写下简明扼要的定义，这些定义必须经过相应的风险承担者同意。]

6.相关事实

[可能对产品产生影响的外部因素，但不是命令式的需求限制条件。]

7.假定

[列出开发者所做的假设。]

*[将所有的假设列在此的目的是让每一个项目成员都意识到这个假设。]*

8.产品的范围

8.1 工作的上下文范围

[上下文范围图用来表示将要开发的系统、产品与其它系统之间的关系，以确定系统边界。]

8.2 工作切分

[一个事件清单，确定系统要响应的所有业务事件。清单包括：]

1. 事件名称
2. 输入和输出

8.3 产品边界

[你可以使用用例图(use-case)来确定了用户与产品之间的边界。]

9.功能性需求与数据需求

9.1 功能性需求

[对产品必须执行的动作的描述。]

*[每个功能性需求必须有一个验收标准。]*

9.2 数据需求

[与产品/系统有密切关系的主题域相关的业务对象、实体、类的说明书。]

*[进行问题域建模，生成相应的类图。]*

10.观感需求

[一些与产品的用户界面相关的需求描述。]

11.易用性需求

11.1 易于使用

[描述如何构建符合最终用户期望的产品。]

11.2 学习的容易程序

[学习使用该产品应该多容易的说明。通常是有学习时间来衡量。]

12.性能要求

12.1 速度需求

[明确完成特定任务需要的时间，这常常指响应时间。]

12.2 安全性的需求

[对可能造成人身伤害、财产损失和环境破坏所考虑到的风险进行量化描述。]

12.3 精度需求

[对产品产生的结果期望的精度进行量化描述。]

12.4 可靠性和可用性需求

[本节量化产品所需的可靠性。这常常表述为允许的两次失败之间无故障运行时间，或允许的总失败率。]

12.5 容量需求

[本节明确处理的吞吐量和产品存储数据的容量。]

13.操作需求

13.1 预期的物理环境

[本节明确产品将操作的物理环境，以及这种环境引起的任何特殊需求。]

13.2 预期的技术环境

[硬件和其它组成新产品操作环境的设备的规范。]

13.3 伙伴应用程序

[对产品必须与之交互的其它应用程序的描述。]

14.可维护性和可移植性需求

14.1 维护该产品需要多容易

[对产品作特定修改所需时间的量化描述。]

14.2 是否存在一些特殊情况适用于该产品的维护

[关于预期的产品发布周期和发布将采取的形式的规定。]

14.3 可移植性需求

[对产品必须支持的其他平台或环境的描述。]

15.安全性需求

15.1 该产品是保密的吗？

[关于该被授权使用该产品，以及在什么样的情况下授权等方面的描述。]

15.2 文件完整性需求

[关于需要的数据库和其他文件完整性方面的说明。]

15.3 审计需求

[关于需要的审计检查方面的说明。]

16.文件和政策需求

[本节包括针对社会和政策的因素的规格说明，这些因素会影响产品的可接受性。如果你开发的产品是针对外国市场的，可能要特别注意这些需求。]

*[问一下是否产品的目标是你所不熟悉的文化环境，是否其它国家的人或其他类型的组织的人会使用该产品。人们是否有与你的文化不同的习惯、节日、迷信、文化上的社会行为规范。]*

17.法律需求

17.1 该产品是否受到某些法律的管制

[明确该产品的法律需求的描述。]

17.2 是否有一些必须符合的标准

[明确适用的标准和参考的详细标准的描述。]

18.Opend问题

[对未确定但可能对产品产生重要影响的因素的问题描述。按照需求分析的术语还说，就是TBD（To Be Define）的问题。]

19.COTS解决方案

19.1 是否有一些制造好的产品可以购买

[应该调查现存产品清单，这些产品可以作为潜在的解决方案。]

19.2 该产品是否可使用制造好的组件

[描述可能用于该产品的候选组件，包括采购的和公司自己的产品。列出来源。]

19.3 是否有一些我们可以复制的东西

[其他相似产品的清单。]

20.新问题

20.1 新产品会在当前环境中带来什么问题

[关于新产品将怎样影响当前的实现环境的描述。]

20.2 新的开发是否将影响某些已实施的系统

[关于新产品将怎样与现存系统协同工作的描述。]

20.3 是否我们现有的用户会受到新开发的敌对性影响

[关于现有用户可能产生的敌对性反应的细节。]

20.4 预期的实现环境会存在什么限制新产品的因素

[关于新的自动化技术、新的组织结构方式的任何潜在问题的描述。]

20.5 是否新产品会带来其他问题

[确定我们可能不能处理的情况。]

21.任务

21.1 为提交该产品已经做了哪些事

[用来开发产品的生命周期和方法的细节。画一个高层的过程图展示各项任务和它们之间的接口，这可能是沟通这方面信息的最好办法。]

21.2 开发阶段

[关于每个开发阶段和操作环境中的组件的规格说明。]

22.移交

22.1 我们要让已有数据和过程配合新产品，有什么特殊要求

[一个移交活动的列表，一个实现的时间表。]

22.2 为了新产品，哪些数据必须修改/转换

[数据转换任务清单，同时确定新产品需要转换的数据。]

23. 风险

23.1 当你开发该产品时，要面对什么风险

23.2 你制定了怎样的偶然紧急情况计划

24.费用

[需求的其他费用是你必须投入到产品构建中去的钱或工作量。当需求规格说明书完成时，你可以使用一种估算方法来评估费用，然后以构建所需的资金或时间的形式表述出来。]

25.用户文档

[用户文档的清单，这些文档将作为产品的一部分交付。]

26.后续版本的需求

[这里记录下一些希望今后版本中实现的需求。]

**Volere需求记录卡**

编者说明：

正如前面所述，Atlantic System Guild还提供了一个配套的Volere需求记录卡，这个记录卡十分实用。建议大家在需求调查、分析过程中，将需求记录在一系列的Volere需求记录卡上，这个卡让你能够很好的理清需求之间的关系，需求提出的背景，用户对需求的期望，有了这些素材，整理SRS时将变得更加简单。

需求#： 需求类型： 事件/用例#：

描述：

理由：

来源：

验收标准：

顾客满意度： 顾客不满意度：

依赖关系： 冲突：

支持材料：

历史：

Copyright @ Atiantic system Guild

**Volere**

*注：顾客满意度是指完成该项功能顾客满意的程度，而顾客不满意度则是指未实现该功能顾客不满意的程度。*

### 软件需求规格说明书

编者说明：

如果在需求分析时采用了用例（Use case）技术，那么该需求规格说明书将更加符合你的需要。当然，你也可以结合Volere需求规格说明书对该模板进行必要的修改。

1. 文档概述

[该部分主要是对软件需求规格说明书文档进行基本的描述，包括该文档的目的、范围、术语定义、参考资料以及概要。]

[软件需求规格说明书用来系统、完整地记录系统的软件需求。该软件需求说明书的基础是用例分析技术。因此该文档中应包括用例模型、补充规约等内容。]

1.1目的

[在此小节中，主要对软件需求规格说明书的目的做一概要性说明，通常软件需求规格说明书应详细地说明应用程序、子系统的外部行为，还要说明非功能性需求、设计约束，以及其它的相关因素。]

1.2范围

[系统是有范围的，而不是无限扩展的，对于无限扩展的需求是无法进行描述的。因此，在本小节应该对该说明书所涉及的项目范围进行清晰的界定。指定该规格说明书适用的软件应用程序、特性或者其它子系统分组、其相关的用例模型。当然在此也需要列出会受到该文档影响的其它文档。]

1.3 定义、首字母缩写词和缩略语

[与其它文档一样，该文档也需要将本文档中所涉及的所有术语、缩略语进行详细的定义。还有一种可简明的做法，就是维护在一个项目词汇表中，这样就可以避免在每个文档中都重复很多内容。]

1.4参考资料

[在这一小节中，应完整地列出该文档引用的所有文档。对于每个引用的文档都应该给出标题、标识号、日期以及来源，为阅读者查找这些文档提供足够详细的信息。]

1.5 概述

[在本小节中，主要是说明软件需求规格说明书各个部分所包含的主要内容，就像一个文章摘要一样。同时也应该对文档的组织方式进行解释。]

2. 整体说明

[在本节中，将对整个软件需求进行总体性的描述，以期让读者对整个软件系统的需求有一个框架性的认识。也就是说，该节中主要包括影响产品及其需求的一般因素，而不列举 具体的需求。主要包括产品总体效果、产品功能、用户特征、约束、假设与依赖关系、需求子集等方面的内容。]

2.1用例模型

[在本小节中，将列出该软件需求的用例模型，该模型处于系统级，对系统的特性进行宏观的描述。在此应该列出所有的用例和Actor的名称列表，并且对其做出简要的说明，以及在图中的各种关系。]

2.2 假设与依赖关系

[在软件系统的开发过程中，存在许多假设和依赖关系。在本小节中应列举出所有的重要的技术可行性假设、子系统或构件可用性假设，以及一些可行性的假设。]

3. 具体需求

[如果说第二章节是框架，那么本节就是血肉。在本节中，应该详细列出所有的软件需求，其详细程序应使设计人员能够充分理解并且进行设计的要求，同时也应该给予测试人员足够的信息，以帮助他们来验证系统是否满足了这些需求。整个需求的组织可以采用用例描述进行。]

3.1用例描述

[如果你使用用例建模技术，那么你已经通过用例定义了系统的大部分功能性需求和一些非功能性需求。因此，在软件需求规格说明书只需将这些具体的用例描述，整理在一起，全部放在该小节之中。当然也可以将用例描述做为附件，在此列出引用，只是这样做并不利于阅读。建议在组织形式上采用以“软件需求”为线索，在每个需求中，填入对应的1个或几个用例描述。]

3.2补充需求

[由于用例毕竟主要针对功能性需求，因此还会有一些其它的补充需求遗漏，因此在本小节中就是将这些东西补充出来。这些补充需求大部分集中在非功能需求之上，包括以下几个方面的内容：]

* 1. 易用性：例如指出普通用户和高级用户要高效地执行某个特定操作所需的培训时间；指出典型任务的可评测任务次数；或者指出需要满足的可用性标准（如IBM的CUA标准、Microsoft的GUI标准。
  2. 可靠性：包括系统可用性（可用时间百分比、使用小时数、维护访问权、降纸模式操作等）；平均故障间隔时间（MTBF，通常表示为小时数，但也可表示为天数、月数或年数）；平均修复时间（MTTR，系统在发生故障后可以暂停运行的时间）；精确度（指出系统输出要求具备的精密度、分辨率和精确度）；最高错误或缺陷率（通常表示为bugs/KLOC，即每千行代码的错误数目或 bugs/function-point，即每个功能点的错误数目）；错误或缺陷率（按照小错误、大错误和严重错误来分类：需求中必须对“严重”错误进行界定，例如：数据完全丢失或完全不能使用系统的某部分功能）。
  3. 性能：包括对事务的响应时间（平均、最长）；吞吐量（例如每秒处理的事务数）；容量（例如系统可以容纳的客户或事务数）；降级模式（当系统以某种形式降级时可接受的运行模式）；资源利用情况：内存、磁盘、通信等。
  4. 其它：包括用户界面要求、联机帮助系统要求、法律许可、外购构件，以及操作系统、开发工具、数据库系统等设计约束。

4.支持信息

[支持信息用于使软件需求规格说明书更易于使用。它包括：目录、索引、附录等。]