



第三章

需求规约



北京大学



软件需求及系统/产品(需求)规约

不论是自顶向下的软件开发,还是自底向上的软件开发,正确定义问题,是解决问题的前提.

--定义问题的基本要素是什么?

--定义问题的基本格式是什么?





1 如何认识需求工作

首先,了解现代系统中软件的作用

在许多现代技术产品中, 最重要的技术是软件技术。

软件通常为一些产品:

- 提供了控制功能

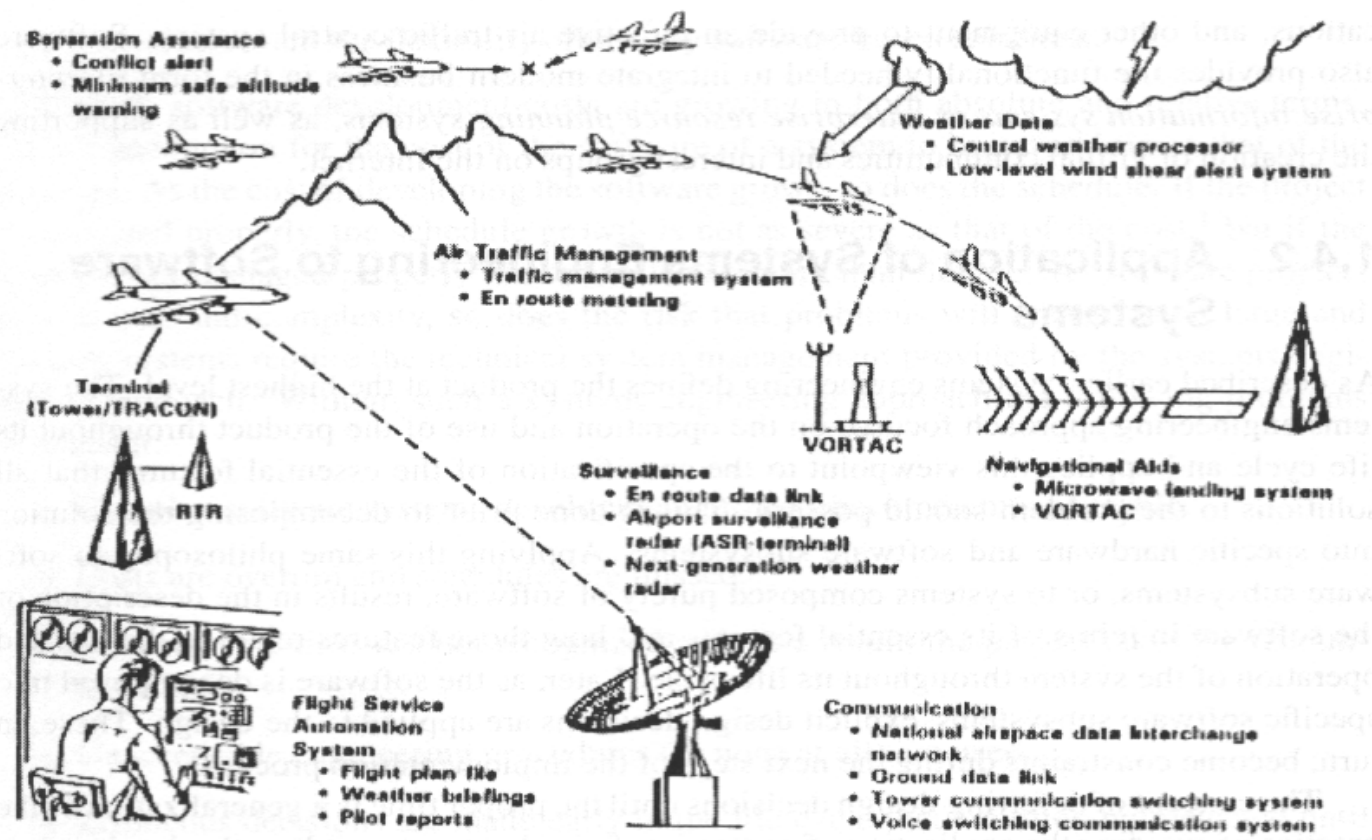
- 提供耦合功能,

- 提供一些由软件本身所实现功能。

即软件是现代系统中的重要元素, 使这些产品/系统成为用户的解决方案。



北京大学



软件集成系统示例--空中交通管制系统

其中软件集成了人员、雷达、飞机、通讯及其它设备等



北京大学



特别地:

- 为解决系统集成所遇到的问题，尤其是开发周期后期中所发现的问题，提供了的灵活性。

- 为软/硬件接口中所出现的问题，提供了低成本解决途径, 即修改软件与修改硬件相比可大大节省费用。

但要认识到:“软件是容易修改的，但修改正确是很难的”。

由此可见，软件通常是任何系统中最为复杂的部分。在系统创建时，软件的开发经常成为最大的技术挑战。



北京大学



其次,了解需求工作在系统工程中的位置

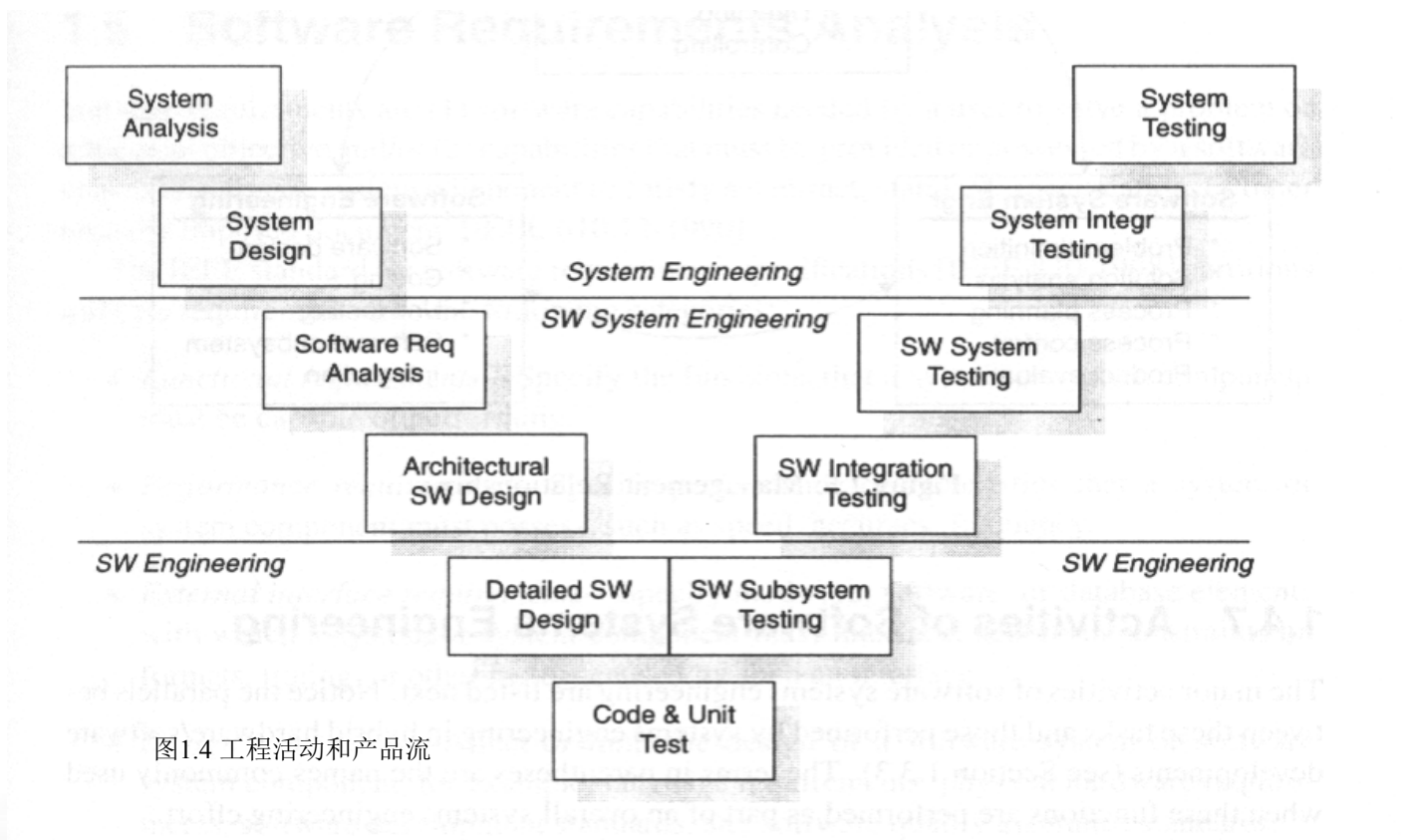


图1.4 工程活动和产品流





其中标注为软件系统工程的一些活动, 是作为系统工程师工作的一部分被实施的。

①需求分析(问题定义)：通过分析分配给软件的那些系统需求, 确定软件需求及约束

②软件体系结构设计(解决方案分析)：为软件需求及约束, 确定一组解决方案, 进行实例研究, 分析可能的方案, 并选择一个最佳的一个方案。

③验证、确认及测试(产品评估)：以需求为准则, 通过测试、演示、分析及审查等方式, 评估最终产品和文档。其中包括一些必要的软件系统集成活动。





按系統工程的觀點, 以下兩個活動也屬於系統工程活動的
範疇:

④過程規劃：確定要做的任務，這些任務的範圍，完成這些任務所需的工作，任務之間先後次序，以及這些任務對項目的潛在風險。

⑤過程控制：確定用於控制項目和過程、進度測量、中間產品複審的技術方法，並確定在必要時所採取的改正活動所使用的方法。





可见,在任何软件开发活动中, 第一步都是:

- 调查、确定在一个系统需求规约中的系统需求;
- 调查、确定在一个软件需求规约 (SRS) 中的软件需求。

即:不论是采用自顶向下的软件开发,还是采用自底向上的软件开发, 软件需求是软件开发的工作基础。



北京大学



2 需求与需求获取

1) 定义

一个需求是一个有关“要予构造”的陈述，描述了待开发产品／系统（或项）功能上的能力、性能参数或者其它性质。

A requirement is a statement that has been constructed to describe a necessary functional capability, performance parameter, or other property of the intended product(or item).





例如：

①系统必须有能力强支持100个以上的并发用户，每个用户可以处理附录A中操作任务的任选组合，平均响应时间应该小于1秒，最大响应时间应小于5秒。

其中：功能-可以处理附录A中操作任务的任选组合

性能-有能力强支持100个以上的并发用户

平均响应时间应小于1秒，最大响应时间应小于5秒。

②必须在对话框的中间显示错误警告，其中使用红色的、14点加粗Arial字体。

其中：功能-能显示错误警告

设计约束-在对话框的中间显示,并使用红色的、14点加粗Arial字体。





2)什么样的陈述可以作为需求

--需求的基本性质

IEEE标准830-1998要求单一需求必须具有5个基本性质:

- ①必要的(Necessary)。是要求的吗?
- ②无歧义的(Unambiguous)。只能用一种方式解释吗?
- ③可测的(testable)。可以对它进行测试吗?
- ④可跟踪的(Traceable)。可以从一个开发阶段到另一个阶段对它进行跟踪吗?
- ⑤可测量的(Measurable)。可以对它进行测量吗?

注:确定一个需求是否满足以上五个性质是复杂耗时的过程.



北京大学



3) 需求分类

功能； 性能； 外部接口； 设计约束； 质量属性。

① 功能需求

功能需求规约了系统或系统构件必须执行的功能。

例如

系统应对所有已销售的应纳税商品计算销售税。

系统应提供一种方法，使系统用户可根据本地利率调整销售税比例。

系统应能够产生月销售报表。



北京大学



除了对要执行的功能给出一个陈述外，还应规约如下内容：

- ①关于该功能输入的所有假定，或为了验证该功能输入，有关检测的假定。
- ②功能内的任一次序，这一次序是与外部有关的。
- ③对异常条件的响应，包括所有内外部所产生的错误。
- ④需求的时序或优先程度。
- ⑤功能之间的互斥规则。
- ⑥系统内部状态的假定。
- ⑦为了该功能的执行，所需要的输入和输出次序。
- ⑧用于转换或内部计算所需要的公式。



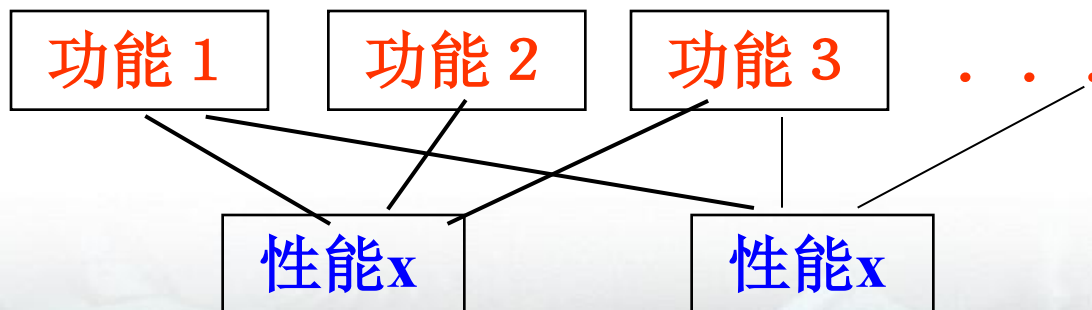


注 1 :

功能需求是整个需求的主体, 即没有功能需求, 就没有非功能需求, 即性能需求、外部接口需求、设计约束和质量属性。

注 2 :

非功能需求对功能需求而言, 可以是一对多的, 例如:



北京大学



② 性能需求

性能需求(Performance requirement)规约了一个系统或系统构件必须具有的性能特性。例如：

系统应该在5分钟内计算出给定季度的总销售税。

系统应该在1分钟内从100000条记录中检索出一个销售定单。

该应用必须支持100个Windows 95/NT工作站的并行访问。

注：性能需求隐含了一些满足功能需求的设计方案，经常对设计产生一些关键的影响。例如：排序，关于花费时间的规约将确定哪种算法是可行的。





③外部接口需求

外部接口需求(**External interface requirement**)规约了系统或系统构件必须与之交互的硬件、软件或数据库元素。它也可能规约其格式、时间或其他因素。

例如：

账户接收系统必须为月财务状况系统提供更新信息，如在“财务系统描述”第4修订版中所描述的。

引擎控制系统必须正确处理从飞行控制系统接收来的命令，符合接口控制文档**B2-10A4**，修订版C的1到8段的规定。



北京大学



--**用户接口(User interfaces)**：规约了软件产品和用户之间接口的逻辑特性。即规约 对给用户所显示的数据，对用户所要求的数据以及用户如何控制该用户接口。

--**硬件接口(Hardware interfaces)**：如果软件系统必须与硬件设备进行交互，那么就应说明所要求的支持和协议类型。

--**软件接口(Software interfaces)**：允许与其它软件产品进行交互，如，数据管理系统、操作系统或数学软件包。

--**通讯接口(Communications interfaces)**：规约待开发系统与通讯设施(如，局域网)之间的交互。如果通讯需求包含了系统必须使用的网络类型（TCP/IP, WindowsNT, Novell），那么有关类型的信息就应包含在SRS中。





--内存约束(**Memory constraints**)：描述易失性存储和永久性存储的特性和限制，特别应描述它们是否被用于与一个系统中其它处理的通讯。

--操作(**Operation**)：规约用户如何使系统进入正常和异常的运行以及在系统正常和异常运行下如何与系统进行交互。应该描述在用户组织中的操作模式，包括交互模式和非交互模式；描述每一模式的数据处理支持功能；描述有关系统备份、恢复和升级功能方面的需求。

--地点需求(**Site adaptation requirements**)：描述系统安装以及如何调整一个地点，以适应新的系统。





④设计约束

设计约束限制了系统或系统构件的设计方案。就约束的本身而言，对其进行权衡或调整是相当困难的，甚至是不可能的。它们必须予以满足。这一性质，是与其它需求的最主要差别。为了满足功能、性能和其它需求，许多设计约束将对软件项目规划、所需要的附加成本和工作产生直接影响。例如：

系统必须用C++或其他面向对象语言编写。 系统用户接口需要菜单。

任取10秒，一个特定应用所消耗的可用计算能力平均不超过50%。

必须在对话框口的中间显示错误警告，其中使用红色的、14点加粗
Arial字体。



北京大学



针对产品开发，为确定其相关的设计约束，一般需要考虑以下10个方面：

--法规政策(Regulatory policies)；

--硬件限制(Hardware limitations)，例如：处理速度、信号定序需求、存储容量、通讯速度以及可用性等；

--与其它应用接口(Interfaces to other applications)，如，当外部系统处于一个特定状态时，禁止新系统某些操作

--并发操作(Parallel operations)，例如，可能要求从/自一些不同的源，并发地产生或接收数据。对此，必须清晰地给出有关时间的描述。





--**审计功能(Audit functions)**, 规约软件系统必须满足的数据记录准则或事务记录准则。如, 如果用户察看或修改数据, 那么就可能要求该系统为了以后复审, 记录该系统的动作。

--**控制功能(Control functions)**: 可以对系统的管理能力进行远程控制、可以对其他外部软件以及内部过程进行控制。

--**高级语言需求(Higher order language requirements)**:

--**握手协议(Signal handshake protocols)**: 通常用于硬件和通讯控制软件, 特别当给出特定的时间约束时, 一般就要把“握手协议”作为一项约束。

--**应用的关键程度(Criticality of the application)**, 许多生物医学、航空、军事或财务软件属于这一类。

--**安全考虑(Safety and security considerations)**



北京大学



⑤ 质量属性

质量属性(Quality attribute)规约了软件产品必须具有的一个性质是否达到质量方面一个所期望的水平。例如：

属性	描述
----	----

可靠性	软件系统在指定环境中没有失败而正常运行的概率。
-----	-------------------------

存活性	当系统的某一部分系统不能运行时，该软件继续运行或支持关键功能的可能性。
-----	-------------------------------------

可维护性	发现和改正一个软件故障或对特定的范围进行修改所要求的平均工作。
------	---------------------------------

用户友好性	学习和使用一个软件系统的容易程度。
-------	-------------------

安全性	在一个预定的时间内，使软件系统安全的可能性。
-----	------------------------

可移植性	软件系统运行的平台类型。
------	--------------





4) 需求发现

常用的发现初始需求的技术，包括：

① 自悟(Introspection)

需求人员把自己作为系统的最终用户，审视该系统并提出问题：“如果是我使用这一系统，则我需要...”

适用条件：需求工程师不能直接与用户进行交流，自悟是乎是一种比较有吸引力的方法，可能确实是必须的。

成功条件：若使自悟是成功的，需求人员必须具有比最终用户还要多的应用领域和过程方面的知识，并具有良好的想象能力。





② 交谈(Individual interviews)

为了确定系统应该提供的功能，需求人员通过提出问题，用户回答，直接询问用户想要的是一个什么样的系统。

成功条件：交谈通常是一种比自悟更好的技术。这种途径成功与否依赖于：

- 需求人员是否具有“正确提出问题”的能力，
- 回答人员是否具有“揭示需求本意”的能力。

存在的风险：在交谈期间需求可能不断增长，或是以前没有认识到的合理需求的一种表现，说是“完美蠕行”(Creeping elegance)病症的体现，以至于很难予以控制，可能导致超出项目成本和进度的限制。





应对措施：项目管理人员和客户管理人员应该定期地对交谈过程的结果进行复审。其中具有挑战的问题是：
判断：
——什么时候对这一增长划界；
——什么时候将这一增长通知客户。





③观察(Observation)

通过观察用户执行其现行的任务和过程，或通过观察他们如何操作与所期望的新系统有关的现有系统，了解系统运行的环境，特别是了解要建的新系统与现存系统、过程以及工作方法之间必须进行的交互。尽管了解的这些信息可以通过交谈获取，但“第一手材料”一般总是能够比较好的“符合现实”的。

存在的风险：

- 客户可能抵触这一观察。其原因是他们认为开发者打扰了他们的正常业务。
- 客户还可能认为开发者在签约之前，就已经熟悉了他们的业务。





④小组会(Group session)

举行客户和开发人员的联席会议，与客户组织的一些代表共同开发需求。其中：

—通常是由开发组织的一个代表作为首席需求工程师或软件工程项目经理，主持这一会议。但还可以采用其它形式，这依赖于其应用领域和主持人的能力。主持人的作用主要是掌握会议的进程。

—必须仔细地选择该小组的成员，不仅要考虑他们对现存的和未来运行环境的理解程度，还要考虑他们的人品。





优点：

第一，如果会议组织得当，可很快地标识出一些需求。

第二，可使需求开发人员在一次会议中能够对一个给定的需求得到多种观点，从而不但可节省与个人交谈的时间，还可节省联系他们的时间。

最后，有关需求不同观点之间的冲突，可以揭示需求中存在的问题，也有助于客户在其内部达成一致。



北京大学



⑤提炼(Extraction)

复审技术文档（例如，有关需要的陈述，功能和性能目标的陈述，系统规约接口标准，硬件设计文档以及ConOps文档），并提出相关的信息。

适用条件：提炼方法是针对已经有了部分需求文档的情况。依据产品的本来情况，可能有很多文档需要复审，以确定其中是否包含相关联的信息。在有的情况，也可能只有少数文档需要复审。

在许多项目中，在任何交谈、观察、小组会或自悟之前，应该对该项目的背景文档进行复审，还应对系统规约进行复审，同时了解相关的标准和政策。





- 注 1：在任意特定的环境中，每项技术都有其自己的优点和不足。在实施上述任何一项技术时，都可以辅以其他方法，例如原型构造，在举行小组会时可以使用原型，方便人员之间的交流。
- 注 2：依据需求工程人员的技能和产品、合同的实际情况，往往需要“组合”地使用这些技术来开发初始需求。
- 注 3：执行需求发现这项活动的人，其技能水平将对这项活动的成功具有重大的影响。
- 注 4：大型复杂项目和一些有能力的组织，在开发需求文档时，往往使用系统化的需求获取、分析技术和工具。一些方法提供了系统化、自动化的功能，并可逐一验证单一需求所具有的五个性质，验证需求规约是否具有四个性质。



2 需求规约（SRS）及其格式

1) 概念

一个需求规约是一个软件项/产品/系统所有需求陈述的正式文档，是一个软件产品／系统的概念模型。

2) 基本性质

一般来说，SRS应必须具有以下4个性质：

- ①重要性和稳定性程度(Ranked for importance and stability)。
- ②可修改的(Modifiable)。在不过多地影响其它需求的前提下，可以容易地修改一个单一需求。
- ③完整的(Complete)。没有被遗漏的需求。
- ④一致的(Consistent)。不存在互斥的需求。





其中，就功能的需求规约来说，还应考虑以下问题：

- (1) 功能源。
- (2) 功能共享的数据。
- (3) 功能与外部界面的交互。
- (4) 功能所使用的计算资源。





3)需求规约（草案）格式

在获取以上初始需求的基础上，可采用IEEE标准830-1998所给出的格式，完成一个完整的需求文档草案的编制工作。

1. 引言

1.1 目的

1.2 范围

1.3 定义，缩略语

1.4 参考文献

1.5 概述





2. 总体描述

2.1 产品概述

2.2 产品功能

2.3 用户特性

2.4 约束

2.5 假设和依赖

3. 功能需求

附录

索引



北京大学



第三部分“特定需求”，是文档的技术核心。一般来说，应根据不同类型的系统来构造这一部分，其中可能会涉及一些模板：

模板 1：根据系统运行模式，把第三部分划分为一些小节，并在一个小节中给出系统性能的规约。

模板 2：通过一种可选的模式划分，把第三部分划分为一些小节，其中每种模式的性能包含在该模式的规约中。

模板 3：根据用户类，把第三部分划分为一些小节，其中每类用户执行的功能包含在该类用户的描述中。

模板 4：按对象，把第三部分划分为一些小节，在每一小节中给出该对象所关联的功能。



北京大学



模板 5：根据系统层的特征，把第三部分划分为一些小节，其中，对任意给定的功能需求，可以分布于若干个特征。

模板 6：根据激发（Stimulus），把第三部分划分为一些小节，其中给出响应每一激发所执行的功能的规约。

模板 7：按一个功能层次，把第三部分划分为一些小节，其中，功能的规约是根据它们在信息流上的活动、信息流上所执行的处理以及通过该信息流的数据。

模板 8：根据用户类、功能和特征，把第三部分划分为一些小节。

注：还可能给出其他组织方式。最终所选定的格式，应适合组织的经验、应用及环境、表达需求所使用的语言等。



4)表达需求规约(规格说明书)的三种风格

在获取SRS（草案）期，一般应使用非形式化语言来表达需求规约。

①非形式化的规约

即以一种自然语言来表达需求规约,如同使用一种自然语言写了一篇文章.

其中:可以不局限于那种语言通常所约定的任何符号或特殊限制(例如文法和词法),但要为那些在一个特定语境中所使用的术语提供语义定义,一般情况下,该语境与通常使用该术语的语境是有区别的。





在对需求进行技术分析期间，一般应采用半形式化语言来表达需求规约

② 半形式化的规约

即以半形式化符号体系(包括术语表、标准化的表达格式等)来表达需求规约。因此,半形式化规约的编制应遵循一个标准的表示模板(一些约定)。

其中:

- 术语表明确地标识了一些词，可以基于某一种自然语言
- 标准化的表达格式(例如例如数据流图、状态转换图、实体关系图、数据结构图以及过程结构图等)标识了一些元信息,支持以更清晰的方式系统化地来编制文档。

注：应用中,不论是词还是标准化的表达格式,在表达上均必须遵循一些约定,即应以一种准确和一致方式使用之。



北京大学



对于质量（特别是安全性）要求比较高的软件产品／系统，
一般应采用形式化语言来表达需求规约

③形式化规约

即以一种基于良构数学概念的符号体系来编制需求规约，一般往往伴有解释性注释的支持。其中：

- 以数学概念用于定义该符号体系的词法和语义；
- 定义了一组支持逻辑推理的证明规则,并支持这一符号体系的定义和引用。





注 1：软件系统本来就是复杂的，因此没有必要把系统的规约或实现“束缚”于某一技术上，即可以同时使用多种技术，分析用户需求，并建立相应的文档。例如，假定一个软件系统可能需要一个数据库、一些通讯构件和一个关键控制部分。其中，有关数据库的需求，可以使用一个实体关系图；对于那些通讯构件的需求，就可以使用一个状态变迁图；而对于关键控制部分，就可能需要使用形式化符号。适宜地使用多种可用的方法，就有可能实现高质量SRS的目标。

注 2：确定什么样的需求规约表达方式，这是组织或项目经理的责任，并负责监督需求开发过程的状态和进展，保证其结果符合项目规定的质量、预算和进度。



北京大学



5)需求规约的作用

其作用可概括为：

第一也是最重要的，作为软件开发组织和用户之间一份事实上的技术合同书；是产品功能及其环境的体现。

第二，对于项目的其余大多数工作，它是一个管理控制点。

第三，对于产品的设计，它是一个正式的、受控的起始点。

第四，是创建产品验收测试计划和用户指南的基础,即基于需求分析规规约一般还会产生另外两个文档——初始测试计划和用户系统操作描述。





①初始测试计划

主要内容:对未来系统中的哪些功能和性能指标进行测试, 以及达到何种要求。

作用:指导系统开发早期发现并修改一个错误, 减少测试代价。

注: 在以后阶段的软件开发中, 对这个测试计划要不断地修正和完善, 并成为相应阶段文档的一部分。

注: 大量的统计数字表明, 在系统开发早期发现并修改一个错误的代价往往很低, 越到系统开发的后期, 改正同样错误所花费的代价越高。例如, 假设在需求分析阶段检测并改正一个错误的代价为1个单位, 那么到了软件测试阶段检测并改正同样的错误所花费的代价, 一般需要10个单位, 而到软件发布后的代价就可能高达100个单位。





②用户系统操作描述

主要内容:从用户使用系统的角度,简要描述系统功能和性能,使用系统的主要步骤和方法, 以及系统用户的责任等。系统,

注:相当于一份初步的用户手册。

作用:①在软件开发的早期, 准备一份初步的用户手册可以使未来的系统用户能够从使用的角度检查、审核目标系统, 从而比较容易判断这个系统是否符合他们的需要。

②为了书写这样的文档, 也会迫使系统分析员从用户的角度来考虑软件系统。这样不论是审查还是复审时,就更容易发现不一致和误解的地方, 这对保证软件质量和项目成功是很重要的。



北京大学



SRS所不能实现的作用

第一，它不是一个设计文档。它是一个“为了”设计的文档

第二，它不是进度或规划文档，不应该包含更适宜包含在

工作陈述（SOW）、软件项目管理计划(SPMP)、软件生存周

期管理计划(SLCMP)、软件配置管理计划(SCMP)或软件质量保证计划(SQAP)等文档中的信息。因此，在SRS中不应给

出： 项目成本； 交付进度； 报告规程；

软件开发方法；质量保证规程；配置管理规程；

验证和确认规程；验收规程；安装规程。



北京大学



关于项目的需求及其需求规约

项目需求是客户和开发者之间有关技术合同-产品/系统需求的理解，应记录在工作陈述SOW中或其他某一项目文档(例如，项目管理计划)中。

即 SRS应只关注产品需求,即:

产品/系统需求-“交付给客户的产品是什么”

SOW应关注项目工作与管理,即:

项目需求-“开发组要做的是做什么”。



北京大学