

第四章 需求工程

4.1 软件需求

需求的关键在于文档化。客户所说的需求在开发人员看来是一个较高层级的概念，而开发人员所说的需求在用户看来又像是详细设计。

通常软件需求可以分为：业务需求、用户需求、系统需求、功能需求和非功能性需求。

4.1.1 业务需求

业务需求是组织或客户对于系统的高层次的目标需求，定义了项目的远景和范围，确定软件产品的发展方向、功能范围、目标客户和价值来源。

如：该系统使用计算机实现图书资料的日常管理，提高工作效率和服务质量。

应该以文档的形式描述出来，虽然一般较短（**模板**）。

4.1.2 用户需求

从用户的角度描述的系统功能需求和非功能需求，只涉及系统的外部行为，而不涉及系统的内部特性。

如：用户可以通过该系统随时查阅图书信息。

4.1.3 功能需求和非功能需求

功能需求描述系统应该提供的服务和功能，通常设计用户或外部系统与该系统之间的交互，一般不考虑系统的实现细节。

如：用户可以从图书资料库中查询或者选择其中的一个子集。

非功能需求是从各个角度对系统的约束和限制，反映了应用对软件系统质量的额外要求，如时间、精度等。

非功能性需求分为：过程需求、产品需求、外部需求。

过程需求：软件交付、实现方法、标准。

产品需求：可用性、软件性能、存储空间、可靠性、可移植性、安全性。

外部需求：互操作性、道德法规、成本。

4.1.4 系统需求

是更加详细的描述系统应该做什么，通常包括许多模型，比如对象模型、数据模型、状态模型等。

描述系统需求的三种方法有：结构化英语（PDL，介于自然语言和形式化语言之间）、可视化模型（如 UML）、形式化方法（如 Z 语言、Petri Net 等）。

4.2 需求工程过程

4.2.1 需求获取

应该集中在用户任务上，而不是用户接口上

主要工作包括：

1. 聆听用户需求
2. 分析和整理所获取的信息
3. 形成形式化文档

4.2.2 需求分析

需求分析的主要工作包括：

1. 定义系统边界。
2. 建立软件原型。
3. 分析需求可行性。
4. 确定需求优先级。
5. 建立需求分析模型。
6. 创建数据字典。

4.2.3 需求规格说明（SRS）

它精确的阐述一个系统必须提供的功能和性能以及它要考虑的限制条件。

通常采用 **IEEE 标准 830-1998 模板**。

4.2.4 需求验证

为了确保需求说明准确的、完整地表达必要的质量特点。

需求验证主要围绕着需求规格说明的质量特性展开，这些质量特性包括正确性、无二义性、完整性、可验证性、一致性、可修改性、可跟踪性。**每个方面都有自己的问题。**

4.2.5 需求管理

1.需求变更控制

通常包括：(1) 仔细评估已建议的变更；(2) 挑选合适的人选对变更做出决定；(3) 变更应及时通知所有涉及的人；(4) 项目要按一定的程序实施需求变更。

2.需求文档的版本控制

它保证在需求文档中记录和反应所有的需求变化。

3.需求跟踪

需求跟踪包括编制每个需求同系统元素之间的联系文档，这些元素包括别的需求、体系结构、其他设计部件、源代码模块、测试帮助文档等。从而建立了需求的跟踪联系链。当需求变化时，使用需求跟踪可以确保不忽略每个受影响的系统元素。

4.需求管理工具

如 Caliber-RM、DOORS 等。

4.3 需求获取技术

4.3.1 面谈：任何时候都可以使用的简单而直接的方法。

4.3.2 需求专题研讨会

是需求获取的一种最有力的方式。项目的主要风险承担人在短暂的时间段内集中在一起达成共识、对操作过程取得尽快统一的意见。

4.3.3 观察用户的工作流程

- 1.被动观察，在不直接干预的情况下观察用户企业的业务活动。
- 2.主动观察，开发人员直接参与到用户的业务活动中。

4.3.4 原型化方法：分为抛弃式原型和演化式模型。

4.3.5 基于用力的方法

只关心系统所实现的功能，而不关心内部的具体实现细节。

1. 确定参与者
2. 确定用例

3. 描述用例

4.4 案例：小型图书馆资料管理系统