质量属性报告

**——项目需求工程**

版本号：[0.1.0.20240505]

拟制人：钱佳苗 32101114

钱丁瑜 32101113

胡欣阳 32101109

黄子涵 32101110

李玉炜 32101112

审核人： 杨 枨 老 师

|  |  |  |
| --- | --- | --- |
| 文件状态：  　[√]草稿  　[　]正式发布  　[ ]正在修改 | 文件标识： | SRA2024-G07-质量属性报告 |

文档修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修订人 | 参与者 | 修订日期 | 修订备注 |
| 0.0.1 | 钱佳苗 | 全组人员 | 2024-05-12 | 首次编写，完善初步模板与格式 |

目录

[1．非功能性需求 4](#_Toc167144786)

[2.1 软件质量度量方法学（IEEE标准） 4](#_Toc167144787)

[2.2 软件质量属性 5](#_Toc167144788)

[2.2.1用户角度 5](#_Toc167144789)

[2.2.2开发者角度 5](#_Toc167144790)

[2.2.3项目的质量保证人员 6](#_Toc167144791)

[2.2.4组织机构 6](#_Toc167144792)

[3．质量属性的优先级 7](#_Toc167144793)

[3.1 目的 7](#_Toc167144794)

[3.2 IEEE 1061-1998标准 7](#_Toc167144795)

# 1．非功能性需求

在在系统工程及需求工程中，非功能性需求（Non-functional requirement）指的是对系统操作状态或其特性的要求，而非针对系统具体行为的需求。与非功能性需求相对的是功能需求，后者定义了系统的具体行为或功能。非功能性需求可以看作是为了满足客户业务需求而需要达到的标准，但不属于功能性需求的范畴。

非功能性需求通常被称为系统的“质量属性”，有时也被称为“约束”、“质量目标”、“服务质量需求”或“非行为需求”。这些需求主要涉及系统的性能、可靠性、安全性等方面。

## 2.1 软件质量度量方法学（IEEE标准）

IEEE Std 1061软件质量度量方法学是一种系统的方法，以便建立某个软件系统的质量需求，标识、实现、分析并确认该软件的质量度量过程。该方法学跨越整个软件生存周期，并包括下列5个步骤。

1. **建立软件质量需求。**在系统开发或系统更改的一开始就选择一个质量要素的清单，给出这些质量要素的优先级，并量化这些质量要素。这些需求应当用于指导和控制系统的开发及交付，以便评估该系统是否满足了合同中规定的质量需求。
2. **标识软件质量度量。**在选择有关度量时使用该软件质量度量框架。
3. **实现软件质量度量。**在软件生存周期的每一阶段，采购或者开发必要工具，收集有关数据并使用度量。
4. **分析软件质量度量结果。**分析并报告度量结果，有助于控制软件开发和评估最终产品。
5. **确认软件质量度量。**把预测的度量结果与直接度量结果进行比较，以确定预测的度量是否准确地测定了它们的相关质量要素。

|  |  |
| --- | --- |
| **度量方法学步骤** | **输出** |
| 建立软件质量需求 | 质量需求 |
| 标识软件质量度量 | 批准的质量度量框架 |
| 度量集 |
| 费用效益分析 |
| 实现软件质量度量 | 数据项描述 |
| 度量/数据项 |
| 可跟踪性矩阵 |
| 培训计划和进度安排 |
| 分析软件质量度量结果 | 组织机构和开发过程变更 |
| 确认软件质量度量 | 确认结果 |

IEEE Std 1061为建立质量需求，标识实现、分析和确认软件质量度量的过程和产品提供了一种方法学，该方法学应用于各种软件在生存周期结构的所有阶段，适用于涉及软件采购、开发、使用、支持、维护和审计的人员，特别适用于度量或估计软件质量的人员。

## 2.2 软件质量属性

### 2.2.1用户角度

有效性（Availability）：系统使用计算机资源的难易程度。

高效性（Efficiency）：系统响应用户输入或者其他事件的快慢程度以及可预见性。

灵活性（Flexibility）：系统是否能够很容易的适应环境和需求的变化。

完整性（Integrity）：防止系统数据错误以及数据丢失的程度

互操作性（Interoperability）：系统与其他系统或者其他组件互联或者交互数据的难易程度。

可靠性（Reliability）：在故障发生之前，系统正常运行时间。

强壮性（Robustness）：系统如何应对非预期的操作。

可用性（Usability）：当在某时某地需要系统服务时，系统服务能够被有效访问的程度。

### 2.2.2开发者角度

可维护性（Maintainability）：系统的可维护性是衡量一个系统的可修复(恢复)性和可改进性的难易程度。

可移植性（Portability）：使系统能够在其他操作环境中运行的难易程度。

可重用性（Reusability）：在多大程度上能够把组件使用在其他系统中。

可测试性（Testability）：可测试性是指软件发现故障并隔离、定位其故障的能力特性，以及在一定的时间和成本前提下，进行测试设计、测试执行的能力。

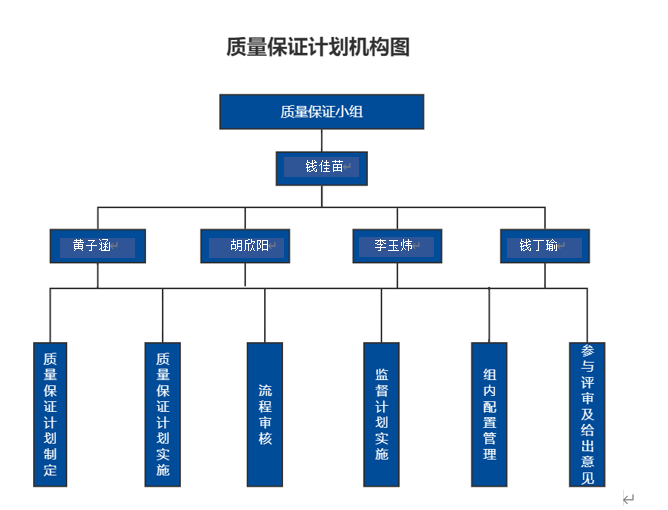
### 2.2.3项目的质量保证人员

质量保证人员的质量职责如下：

1. 负责项目实施过程中对项目实施情况进行监督，包括对项目实施过程和工作产品进行监督检查。
2. 实施项目组成员的质量保证培训。
3. 制定质量保证计划。
4. 按计划实施审计活动，依照质量保证计划执行评审/审计，并记录执行中发现的不符合项。
5. 对不符合问题提交不符合项报告，跟踪并验证纠正措施的执行情况。
6. 对项目内不能解决的不符合项问超；向高层管理提交报告。
7. 向项目经理报告项目质量工作状况和质量度量结果。
8. 定期向项目组报告质量活动的结果。

制定质量保证的过程改进计划，记录过程数据

### 2.2.4组织机构



钱佳苗

# 3．质量属性的优先级

## 3.1 目的

属性需求产生矛盾时，便于对需求属性的取舍。

某些特定属性组合不得不做出一些权衡。用户和开发人员必须决定哪些属性比其他属性更重要，并且在决策时还要尊重属性的优先级。

## 3.2 IEEE 1061-1998标准

“X”代表对应行列的质量属性之间有冲突

“⭕”代表对应行列的质量属性之间有促进作用

