

# 《中南大学 软件测试工程》

## 课后作业一

徐鸣飞

2023 年 10 月 20 日

# 目录

第一章 基础概念	1
1.1 软件功能需求 . . . . .	1
1.2 软件非功能需求 . . . . .	2
第二章 软件非功能需求反例	4

# 第一章 基础概念

## 1.1 软件功能需求

**定义 1.1.1.** 功能需求是软件系统所必须具备的功能或服务，涉及系统能够执行的任务、操作或功能。这些需求描述了系统应该如何响应特定的输入，以及在特定的情况下应该产生怎样的输出。

下面是一些常见的软件功能需求的案例：

### 1. 电子商务网站：

- 用户注册和登录
- 商品搜索和浏览
- 购物车管理和结算
- 订单跟踪和配送管理
- 用户评价和反馈

### 2. 医疗保健软件：

- 病历管理和患者档案
- 预约挂号和排队管理
- 医生处方和药品发放
- 疾病诊断和治疗方案管理
- 医疗报告和检测结果管理

### 3. 学校管理系统：

- 学生信息管理和档案管理
- 课程安排和教师安排

- 成绩管理和考试安排
- 学生选课和课程评价
- 教职工通知和沟通平台

## 1.2 软件非功能需求

**定义 1.2.1.** 非功能需求是指软件系统除了功能之外的特定要求，包括性能、安全性、可靠性、可用性、可维护性等方面的要求。这些需求通常涉及到系统运行的环境、用户体验以及系统本身的特定属性。

下面是一些常见的软件非功能需求的案例：

### 1. 性能需求

- 响应时间：系统必须在用户发出请求后的 2 秒内响应。
- 吞吐量：系统应该能够同时处理 1000 个并发用户。
- 资源利用率：系统在运行时不应占用超过 50% 的服务器资源。

### 2. 安全性需求

- 数据加密：系统必须采用 128 位 AES 加密算法对用户的敏感数据进行加密。
- 用户身份验证：系统应该要求用户使用双因素身份验证登录。
- 访问控制：系统管理员应该有权限控制对系统的访问权限。

### 3. 可靠性需求：

- 故障恢复：系统应该能够在发生故障后 30 分钟内恢复正常运行。
- 数据完整性：系统中的数据应该具有完整性，不能出现数据丢失或损坏的情况。
- 容错能力：系统应该能够自动处理系统错误，并避免导致系统崩溃。

### 4. 可维护性需求：

- 可扩展性：系统应该能够轻松扩展以适应未来的增长需求。
- 可测试性：系统的各个模块应该易于单独测试和调试。
- 文档完整性：系统的相关文档应该详尽完整，以便未来维护和更新。

5. 可用性需求:

- 用户界面友好性: 用户界面应该简洁易用, 符合用户的直觉习惯。
- 多语言支持: 系统应该支持多种语言, 以满足不同国家和地区的用户需求。

## 第二章 软件非功能需求反例

接下来将介绍一些由于未实现软件非功能需求而导致事故的反例：

1. 可靠性故障反例：2017 年，亚马逊云服务发生故障（AWS 云存储服务 S3 出现大宕机，原因是一名程序员运行了一个错误脚本，结果输错了一个字母，导致大量服务器被删），导致包括网站、应用程序和物联网设备在内的大量服务受到影响。这一事件严重影响了许多企业的运营，暴露了未能满足可用性需求可能带来的风险。  
1
2. 安全性需求反例：Equifax 是一家信用评级机构，2017 年因安全漏洞而遭受了严重的数据泄露事件。攻击者利用漏洞获取了超过 1 亿用户的敏感信息，包括社会安全号码和信用卡信息。黑客于当年 5 月到 7 月间利用网络安全漏洞入侵 Equifax 系统，导致 1.47 亿人信息泄露，其中包括姓名、地址、出生日期、身份证号以及护照、驾照、信用卡信息等，美国、英国、加拿大等多国公民受到影响。
3. 性能需求反例：2012 年，美国纳斯达克证券交易所发生了一次严重的交易系统故障，很多证券交易员的交易员未能及时对订单进行确认，导致了市场参与者在数小时、甚至几天的时间内都无法获知他们持有 Facebook 股票的风险，并导致数十亿美元的交易被取消或延迟。
4. 可维护性需求：1998 年的火星探测器失联事件，部分原因是由于软件中的错误导致探测器未能正确执行指令，但由于代码的复杂性和缺乏适当的文档，NASA 难以及时发现和解决问题。

---

<sup>1</sup>可靠性是指系统保持正常运行和可访问的能力，是非功能性需求中的一个重要方面。在这种情况下，由于 S3 服务中断，许多企业和服务无法正常访问或提供服务，导致了严重的业务中断和影响。这突显了云服务的可靠性对企业和消费者的重要性，同时也强调了保持高可靠性的挑战和复杂性。

5. 可用性需求反例：一些科学研究领域的软件由于复杂的算法和模型，导致了软件的可维护性和可理解性较差。尽管这些软件具有强大的分析功能，但由于缺乏良好的文档和模块化设计，使得其他研究人员难以理解和修改这些软件，从而限制了其进一步的应用和发展。