

# 软件测试Assign4

## 要求

选择一个测试要素，以你实现的或者熟悉的一个软件作品为例，分别讨论该要素在软件生命周期的需求、设计、编程、测试、安装、验收和维护各阶段的测试目标和内容。

## 软件生命周期

| SLDC阶段 | 验证活动  |
|--------|---|
| 需求     | 确定验证的方法<br>确定需求的充分程度<br>生成功能测试数据<br>确定与需求符合的设计    |
| 设计     | 确定设计的充分程度<br>生成结构和功能测试数据<br>确定设计与需求的一致性           |
| 编程     | 确定编程实现的充分程度<br>生成各种程序/单元的结构和功能测试数据<br>确定编码与设计的一致性 |
| 测试     | 确定测试计划的充分性<br>测试应用系统                              |
| 安装     | 把已经测试的系统部署运行                                      |
| 维护     | 修改和重新测试   |

## 测试要素

- 可靠性：系统在规定的时间内可以正常运行。
- 授权：特殊的授权可以执行一个特殊的操作。
- 文件完整性：文件被正确使用，恢复和存储的数据正确。
- 进程追踪：能够证实运行进程处在正常工作状态。
- 系统运行的连续性：发生非致命性问题后，系统仍然有能力继续运行关键任务。
- 服务级别：系统有紧急情况发生时，程序的输出结果不经过处理或进行简单的处理后可以直接使用。
- 存取控制：防止系统被误用 (意外或者有意的)。
- 方法论：采用选择的方法论实现系统
- 正确性：数据输入、过程处理和输出的正确性。
- 一致性：确保最终设计和用户需求完全一致。
- 易用性：多数人认为易于使用。
- 可维护性：出现问题时易于定位，并且进行修改。
- 可移植性：数据或者程序易于移植到其它系统上。
- 耦合性：系统中的组件易于联接。
- 性能：系统资源的占用率、响应时间、并发处理等能力。
- 易操作性：系统容易操作。

# 分析软件

---

## 测试软件：

微博

## 测试要素：

系统运行的连续性

## 软件生命周期：

- 需求：
  - 目标：
    - 分析系统需求，定义系统失效影响。
  - 内容：
    - 定义了客户端运行过程中，关于网络请求，后台运行，息屏重启记忆恢复，服务端API等需求。
    - 分析上述系统运行连续性需求中会出现的困难以及解决方法。
- 设计：
  - 目标：
    - 设计系统中断计划。
  - 内容：
    - 定义了客户端运行过程中，客户端与网络断开连接情况下运行的适配性响应，实现网络断开连接情况下的用户交互提醒界面。
    - 定义了客户端运行过程中，服务端API报错的响应，实现报错响应时的用户交互提醒界面。
    - 定义了客户端后台运行，息屏重启等操作时，客户端恢复连续运行的界面，恢复用户操作之前仍在浏览的界面。
- 编程：
  - 目标：
    - 编写中断计划和过程。
  - 内容：
    - 网络断开错误处理，依照不同的网络连接情况，断网，无wifi，无数据，实现不同的用户交互界面跳转。
    - 服务端API报错，依照服务端返回的报错文件，对API返回的数据进行处理，将处理后得到的正确的数据，提交给用户。
    - 后台运行，息屏重启客户端，需要保留用户当前正在浏览的界面和数据。
- 测试：
  - 目标：
    - 系统恢复性测试。
  - 内容：
    - 不停地改变客户端当前的网络环境，测试在不同网络环境下的不同用户交互界面的响应。
    - 测试不同的API查询，是否能够正确查询API，阅读信息。
    - 客户端运行过程中，尝试后台挂机，息屏重启等操作，观察客户端是否能够恢复用户之前的界面。

- 安装：
  - 目标：
    - 保证以前测试的完整性。
  - 内容：
    - 根据不同的安装环境，手机型号，系统版本的不同，进行不同的安装优化。
    - 软件安装成功后，对系统运行的连续性进行再次的测试。
- 维护：
  - 目标：
    - 修改中断计划。
  - 内容：
    - 根据用户的体验反馈，进行软件的更新维护，制定新的测试计划，实现更完善的用户交互体验。