**一、角色**

软件开发工程师（SWE）

软件测试开发工程师（SET）

测试工程师（TE）

**二、测试类型**

|  |  |  |  |
| --- | --- | --- | --- |
| **测试类型** | **介绍** | **类似** | **优点** |
| **小型测试** | 用于验证一个单独函数或独立功能模块的代码是否按照预期工作；运行时间较短；着重于典型功能性问题、数据损坏、错误条件和大小差一错误；一般使用mock和fake | 一个代码单元的功能；  **单元测试** | 可带来优秀的代码质量、良好的异常处理、优雅的错误报告 |
| **中型测试** | 着重验证“功能邻近区”之间的交互以及彼此调用时的功能是否正确； | **集成测试** | 整体产品质量和数据验证 |
| **大型测试** | 涵盖三个或以上的功能模块，使用真实用户使用场景和实际用户数据；关注所有模块的集成，倾向于结果驱动，验证软件是否满足用户的需求 | **系统测试**或者**端到端的测试** |

**三、设计文档**

最早期的项目设计文档：主要包括项目的目标、背景、团队成员、系统设计。

**四、审计设计文档**

完整性、正确性、一致性、设计、接口与协议、测试

**五、测试计划的特性**

1. 及时更新
2. 描述了软件的目标和卖点
3. 描述了软件的结构、各种组件和功能特性的名称
4. 描述了软件的功能和操作简介
5. 可随时被修改
6. 描述必测点
7. 能在测试中提供有用的信息、从而帮助确定进展以及覆盖率上的不足

**六、ACC**

ACC ，**特质、组件、能力**，这是一种测试计划的替代方法

ACC指导原则：

1. 避免散漫的文字
2. 不必推销
3. 简洁
4. 不要把不重要的、无法执行的东西放进测试计划
5. 渐进式描述
6. 指导计划者的思路
7. 最终结果应该是测试用例

**七、风险分析**

影响风险的因素：**失败频率**和**影响**

风险缓解措施：

1. 确定低风险的使用场景，反馈给开发人员，针对性的增加约束
2. 编写回归测试用例
3. 编写和运行发生故障的测试用例，推动开发实现回复和回滚的特性
4. 插入监听代码
5. 插入代码监听软件，发现回归问题

**八、bug的生命周期**

发现bug、分类bug、修复bug、回归测试是软件质量的心跳和运作模式