10.29 测试基础，测试过程

10.31 测试方法，软件质量，覆盖率

11.02 系统测试，测试用例，缺陷管理

**一、测试基础**

软件测试是属于研发流程中的一个环节

**软件生命周期**

计划：项目经理确定项目的目标、时间、预算等，输出《项目计划》；

需求：需求分析师与用户进行沟通，将用户需求转化为软件需求，输出《需求规格说明书》SRS；

设计：系统架构师设计软件的内部结构(框架)，输出《概要设计说明书》HLD，开发人员编写《详细设计说明书》LLD；

编码：开发工程师进行编码，用代码来实现软件的功能；

测试：测试工程师对研发的软件进行验证，确保软件实现了软件需求、满足用户需求；

维护：运维工程师把软件部署到正式的运行环境，处理软件在运行时出现的异常，收集问题进行二次开发。

一个软件的研发流程，从开始到结束的过程；对应软件的一个版本。

**软件测试的价值**

通过软件测试能发现软件存在的Bug；

通过软件测试能够提升软件的质量；

通过软件测试能够对软件的优缺点有更清楚的认识。

软件测试的工作需要有专业的工作流程，需要有独立的分析与设计能力，软件对质量的要求越高，对测试工程师的需求量就越大。

**软件测试的目的**

验证软件的基本功能和业务流程，证明软件是可以使用的；

尽可能的测试到用户能做的所有操作场景，证明软件是存在Bug的；

尽早的介入测试，能够降低Bug修复的成本，避免Bug的产生。

在不同的研发阶段，软件测试的目的并不相同，先证明软件可用 再尽可能多的发现Bug。

**什么是Bug？**

一切影响软件被正常使用的原因或现象都叫Bug

错误：人为主观判断

缺陷：软件的功能存在缺失或者漏洞，缺陷是引起故障的原因（登录代码的密码验证有问题）

故障：故障是软件存在问题的表现形式（错误密码也能进行登录）

失效：软件在运行时，表现的一个不可接受的现象（用户账号安全得不到保障）

软件因为存在缺陷所以会引起故障，故障如果没有合理解决会导致软件的功能失效

很多时候测试工程师发现的Bug仅是一个错误、故障或者失效，开发修复Bug需要定位到产生故障的原因上才行。

**研发的组织架构**

* 开发部门：主要负责产品的设计与开发

需求分析师：负责产品的需求分析与需求制定；

系统架构师：负责产品的架构设计；

UI前端工程师：负责产品的界面和外观实现；

开发工程师：负责产品的逻辑和功能实现；

* 测试部门：主要负责产品的验证与确认

测试工程师：负责需求的验证，产品的检测；

测试开发工程师：负责实现复杂的测试场景，或者提高测试效率；

QC：产品质量评测

* 质量部门：主要负责产品的研发进度与研发质量

管理岗位：保证研发进度

QA：监督软件研发过程中输出的文档的规范性与及时性

CMO：配置库管理员，管理研发的配置库(文件库)

大公司才会岗位职责明确，分工细致；小公司一般都是身兼多职。

**研发流程**

* 瀑布模型：按照软件生命周期的顺序从前往后进行

阶段明确，便于分析问题

缺点：需求响应速度慢

适用于传统的项目，一般1-3个月更新一次版本

* 螺旋模型：将需求分段实现，每段需求的实现都是一个瀑布模型，每段需求实现完成之后都会进行风险评估

需求分段实现能实时的发现项目风险，降低项目风险

适用于大型项目的研发过程，一般会经历半年以上的研发时间

* 敏捷迭代：

1、将需求进行细化与拆分（理论上不允许有13天工作量以上的需求）

2、将各部门拆散，需求、设计、开发、测试组成小的团队来认领需求

3、省去大部分的编写文档的工作，强调沟通与交流

适用于互联网类的软件研发，1-2周更新一次版本，能够快速的响应市场需求。

对员工的个人能力、沟通交流能力，对团队的默契要求较高。

**二、测试过程**

**软件测试的4个阶段**

* 单元测试 Unit Testing

测试软件的最小单元(函数)，验证软件是否满足LLD（详细设计说明书）的要求

例如：网购的衣柜，第一步需要检查零件是否完好和齐全

* 集成测试 Intergration Testing

测试软件组装之后的模块，以及模块与模块之间的接口，验证软件是否满足HLD（概要设计说明收）的要求

例如：衣柜的底座与四根支柱的组合，支架与衣柜罩子是否能够良好组合

* 系统测试 System Testing

测试软件的整体外部特性(功能、易用性、性能等)，验证软件是否满足SRS（需求规格说明书）的要求

例如：衣柜能否打开与关闭、衣柜长宽高、外观、负重等

* 验收测试 Acceptance Tesing

从用户的角度来验证软件是否满足用户的需求

α测试：在受控制的环境进行验收测试，例如游戏内测

β测试：在不受控制的环境进行验收测试，例如游戏公测

正式验收测试：在客户现场由客户代表进行确认结果的验收测试

UT --> IT --> ST --> AT

UT --> SIT --> UAT

当前IT行业，UT测试一般由开发自测，测试进行UT投入成本较高。

**回归测试**

验证Bug是否被正确修复的过程

所有的测试阶段都有可能发现Bug，都需要进行回归测试

* 回归测试需要测试什么?回归测试策略

完全回归测试很好，消耗时间和人力（考验项目组自动化测试的能力）

选择性回归测试：

1、已发现的Bug被正确修复，并且没有引入新的Bug

2、用户频繁使用的功能

3、软件的整体业务流程通畅

**软件测试的4个活动**

每个测试阶段都会有4个活动，下面以系统测试为例给大家讲解4个活动。

* 计划 测试经理确认当前版本的工作范围、时间安排、人员分工、风险预估、工作规范，输出《测试计划》

从管理的角度规划版本的软件测试工作，who what when

* 设计 高级测试工程师 根据需求和测试计划 从技术角度规划当前版本的测试工作如何实现，测试关注点、测试方法、测试用例设计、环境安排等，输出《测试方案》

从技术的角度设计软件测试工作如何实现，how

* 实现 测试工程师根据软件需求、测试计划、测试方案，把需求转化为一个可执行的文档《测试用例》
* 执行 搭建测试环境（操作系统、网络、数据库等技术）

冒烟测试（验证软件最基础的功能，确认当前版本的需求基本实现，保证测试用例可以正常执行）

写好测试用例后，从中抽取20%左右的测试用例进行冒烟测试

执行测试用例（按照测试用例中的操作步骤操作软件，对比软件的实际结果和用例的预期结果，不一致则有可能是Bug）

提交和跟踪Bug（测试人员发现Bug，开发修复Bug，测试人员验证。期间需要测试协助开发重现和定位Bug）

选择性回归测试

写日报，汇报每天的进度和问题（测试经理/测试组长，需要根据组员每天反馈的进度来把控项目进度，需要了解组员遇到的问题及时去协调解决，保证组员能够顺利完成任务）

写测试报告 （由测试经理/测试组长编写，汇总本版本的工作任务和成果，确认该版本测试是否通过，项目经理根据测试报告来判断软件是否可以发布）

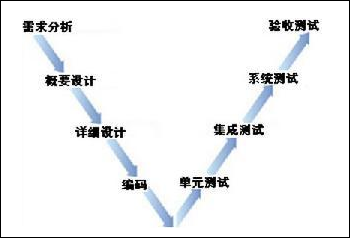
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **计划** | **设计** | **实现** | **执行** |
| **测试经理**  **测试组长** | 1.输出《测试计划》  人员分工、时间安排等  2.组织需求评审。  3.组织《测试计划》评审。 | 1.输出《测试方案》  制定测试方法、测试工具、测试规程等；  2.组织测试思路评审； | 组织测试用例评审； | 1.搭建测试环境；  2.组织每日例会；  3.汇总日报，编写测试组的测试日报；  4.跟踪小组的工作进度；  5.处理工作阻塞性问题；  6.测试执行完成后，进行测试分析，输出测试报告。 |
| **测试工程师** | 1.参与需求评审。  2.参与《测试计划》评审。 | 1.画流程图理解业务；  2.进行测试需求分析，梳理测试思路；  3.评审测试思路； | 编写测试用例；  用例评审； | 1.冒烟测试；  2.执行测试用例；  3.提交跟踪Bug；  4.选择性回归测试；  5.提交工作日报； |

**V模型与双V模型**

* V模型 描述了软件研发过程中，每个工作的开展顺序

会让人误以为测试工作是在编码完成之后才开始的

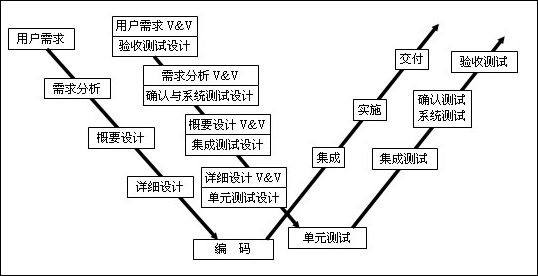
实际测试工作是从需求分析阶段就已经开始



* 双V模型 开发工作和测试工作是并行的

设计顺序和执行顺序是相反的

设计工作与执行工作是分离的，先设计，再执行



设计顺序：AT-->ST-->IT-->UT

执行顺序：UT-->IT-->ST-->AT

**在软件生命周期的过程中，开发和测试的工作内容**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 需求分析 | 概要设计 | 编码 | 测试 |
| 开发 | 评审需求 | 评审架构  详细设计 | 写代码 | 改Bug |
| 测试 | 评审需求  测试需求分析  编写测试计划 | 写测试方案  写测试用例 | | 搭建测试环境  冒烟测试  执行测试用例  回归测试  提交跟踪Bug  日报  测试报告 |