



Python软件测试与质量保 证

汇报人：XX

2024-01-11



目录

- 引言
- Python软件测试基础
- Python软件测试流程
- Python软件质量保证方法
- Python软件质量评估指标
- Python软件测试实践案例



01 引言





目的和背景



提高软件质量

通过软件测试，可以确保软件的质量和稳定性，减少缺陷和错误，提高用户体验。



适应快速变化的需求

随着软件行业的快速发展，用户对软件质量的要求也越来越高，软件测试可以帮助开发团队适应这种快速变化的需求。

降低开发成本

在软件开发过程中尽早发现和修复缺陷，可以避免在后期修复缺陷所带来的高昂成本。

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 #include <sys/types.h>
6 #include <sys/socket.h>
7 #include <netinet/in.h>
8 #include <arpa/inet.h>
9 #include <netdb.h>
10 #include <sys/time.h>
11 #include <sys/resource.h>
12 #include <sys/wait.h>
13 #include <sys/stat.h>
14 #include <sys/mount.h>
15 #include <sys/utsname.h>
16 #include <sys/uio.h>
17 #include <sys/xattr.h>
18 #include <sys/epoll.h>
19 #include <sys/eventfd.h>
20 #include <sys/signalfd.h>
21 #include <sys/timerfd.h>
22 #include <sys/ptrace.h>
23 #include <sys/auxv.h>
24 #include <sys/procfs.h>
25 #include <sys/fsuid.h>
26 #include <sys/sem.h>
27 #include <sys/shm.h>
28 #include <sys/mman.h>
29 #include <sys/swap.h>
30 #include <sys/proc.h>
31 #include <sys/procfs.h>
32 #include <sys/procfs.h>
33 #include <sys/procfs.h>
34 #include <sys/procfs.h>
35 #include <sys/procfs.h>
36 #include <sys/procfs.h>
37 #include <sys/procfs.h>
38 #include <sys/procfs.h>
39 #include <sys/procfs.h>
40 #include <sys/procfs.h>
```

```
578 int programming (ucler channel)
579 {
580     // display ended script 0.7
581     programming = channel & 0x07;
582     // script ext., adc right adjust:resmit;
583     code |= (0 << R4FS1) | (0 << REAS0) | (0 << ADLAR);
584     // script start conversion
585     code |= (1 << AD2C);
586     while (script & (1 << 3556)) ;
587 }
588 return information;
589 }
590 void WEB(void)
591 {
592     uint t a, pcc, k;
593     uint t i, j;
594     k=0x80; a=0b01000000;
595     for (j=0; j<8; j++) {
596         for (i=0; i<8; i++) {
597             if ( ((k >> i) & 1) == 1) pcc = 31; else pcc = 0;
598             hackersendCommand(a++);
599             hackersendChar(active);
600             k = (k >> 1) | 0x80;
601         }
602     }
603 }
```

软件测试的重要性



缺陷预防

通过软件测试，可以尽早发现和修复软件中的缺陷，避免缺陷在后续开发过程中扩散和放大。



质量控制

软件测试是质量控制的重要手段之一，可以确保软件符合预定的质量标准和用户需求。



提高开发效率

通过自动化测试和持续集成等手段，可以提高软件开发的效率，缩短开发周期，加快软件发布速度。

增强用户信任

经过严格测试的软件可以赢得用户的信任 and 好感，提高软件的美誉和品牌价值。

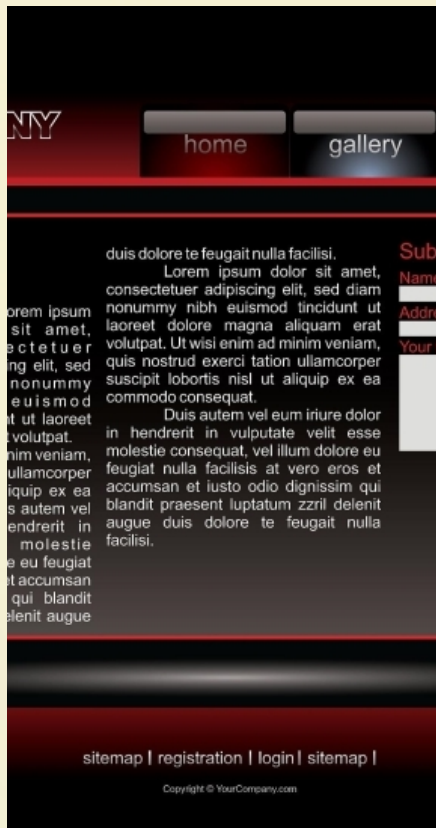
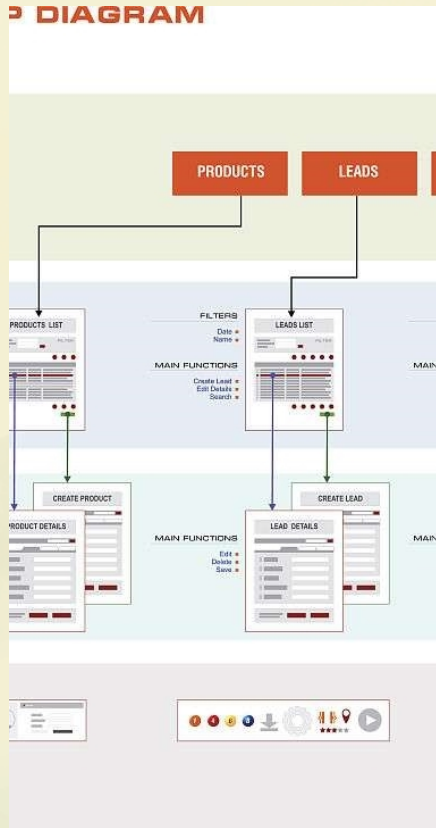


02

Python软件测试基础



软件测试的概念和分类



软件测试定义

通过运行程序或系统，发现其中的错误、缺陷或不符合预期的行为，从而评估软件质量的过程。



软件测试分类

根据测试目的、方法和阶段的不同，软件测试可分为单元测试、集成测试、系统测试、验收测试等多种类型。



Python软件测试工具介绍



unittest

Python标准库中的单元测试框架，提供测试用例组织、执行和结果输出等功能。

pytest

一个流行的Python测试框架，支持单元测试、集成测试和功能测试，具有灵活性和可扩展性。

Selenium

用于Web应用程序的自动化测试工具，支持多种浏览器和操作系统，可用于编写稳定可靠的测试用例。

Requests

一个用于发送HTTP请求的Python库，适用于API测试和性能测试等场景。



03

Python软件测试流程





测试计划制定



01

明确测试目标

确定测试的范围、重点和目标，明确要测试的功能点和性能指标。

02

制定测试策略

根据测试目标，选择合适的测试方法和技术，如单元测试、集成测试、系统测试等。

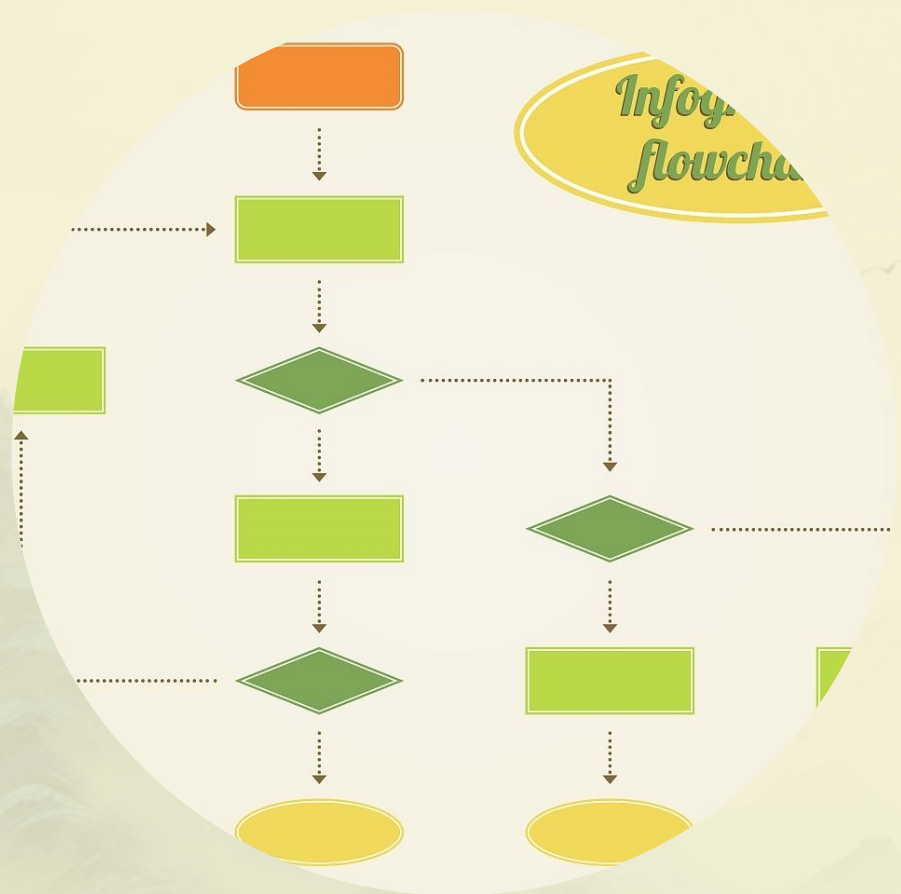
03

分配测试资源

确定测试人员、测试环境、测试工具等资源，确保测试的顺利进行。



测试用例设计



设计测试用例

根据测试目标和策略，设计覆盖所有功能点和性能指标的测试用例，包括正常场景和异常场景的测试用例。

编写测试脚本

使用Python编写自动化测试脚本，提高测试效率和准确性。

评审测试用例

组织相关人员对测试用例进行评审，确保测试用例的完整性和准确性。



测试执行和记录



No. of Weeks.	Initials of Society Officer.	Duration.
13		
13		
13		
13		6/6
25		
13		

01

执行测试用例

按照测试用例的设计，执行测试操作，记录测试结果和日志。

02

缺陷管理

对发现的问题进行记录、分类和跟踪，确保问题得到及时解决。

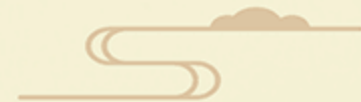
03

回归测试

在问题修复后，重新执行相关测试用例，确保问题不再出现。



缺陷管理和跟踪



● 缺陷记录

详细记录缺陷的信息，包括缺陷描述、重现步骤、严重程度等。

● 缺陷跟踪

对缺陷进行跟踪和管理，确保每个缺陷都得到妥善处理。

● 缺陷分析

对缺陷进行分析和归类，找出根本原因，提出改进措施，避免类似问题再次发生。

TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY



04

Python软件质量保证方法



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/307165152125006116>