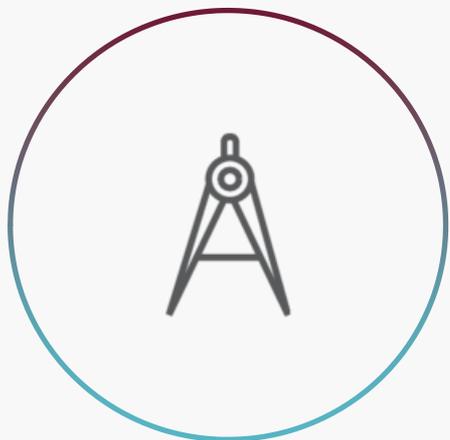


云智未来^{9th}

第九届中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2017



前端测试质量的度量

搜狗测试开发工程师 张杰



工作中的三个难题

如何评价测试质量的好坏

如何衡量测试效率的高低

如何寻找测试改进的方向

覆盖度

缺陷定位

执行效率

维护成本

方案推广

覆盖度

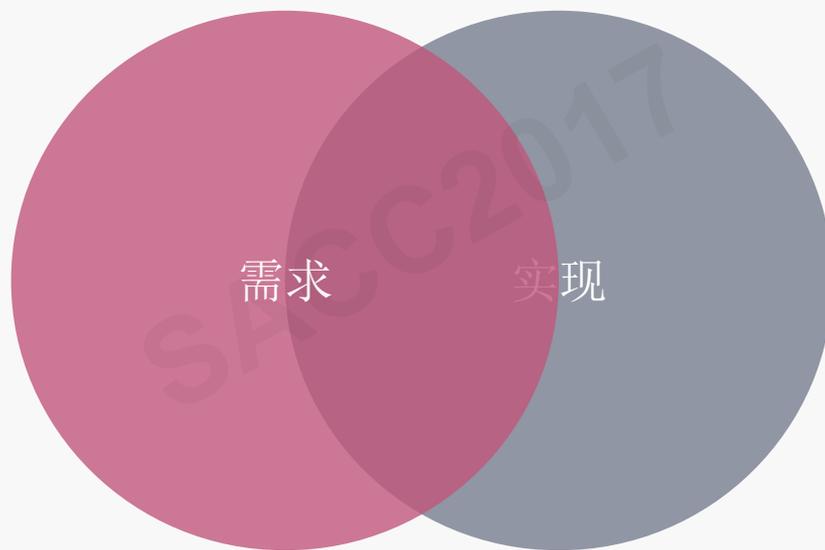
测试全面性的度量



覆盖度

测试全面性的度量

需求覆盖度



测试用例评审

覆盖度

测试全面性的度量

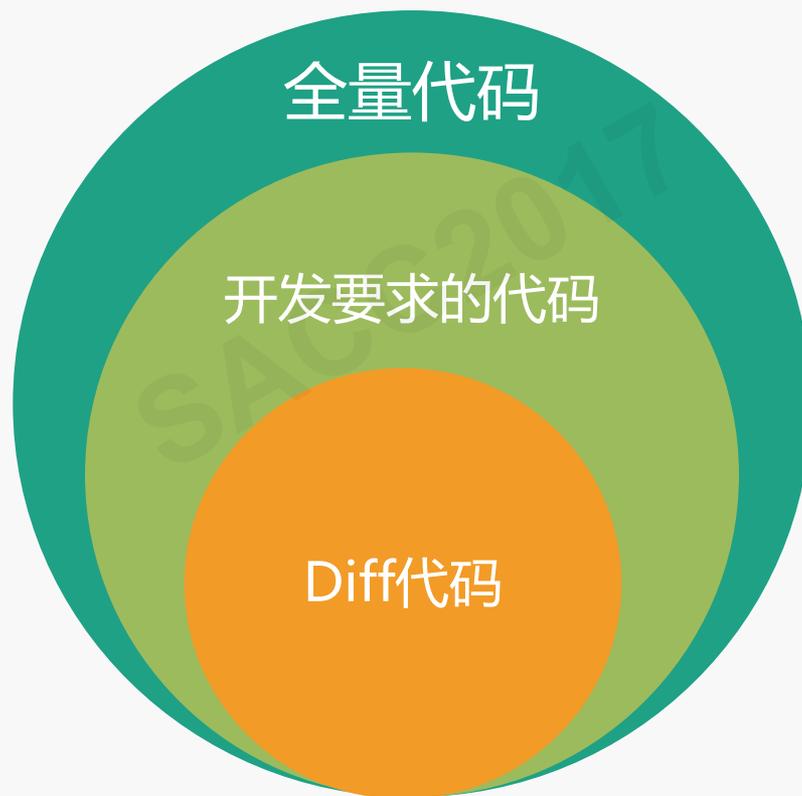
页面覆盖度



覆盖度

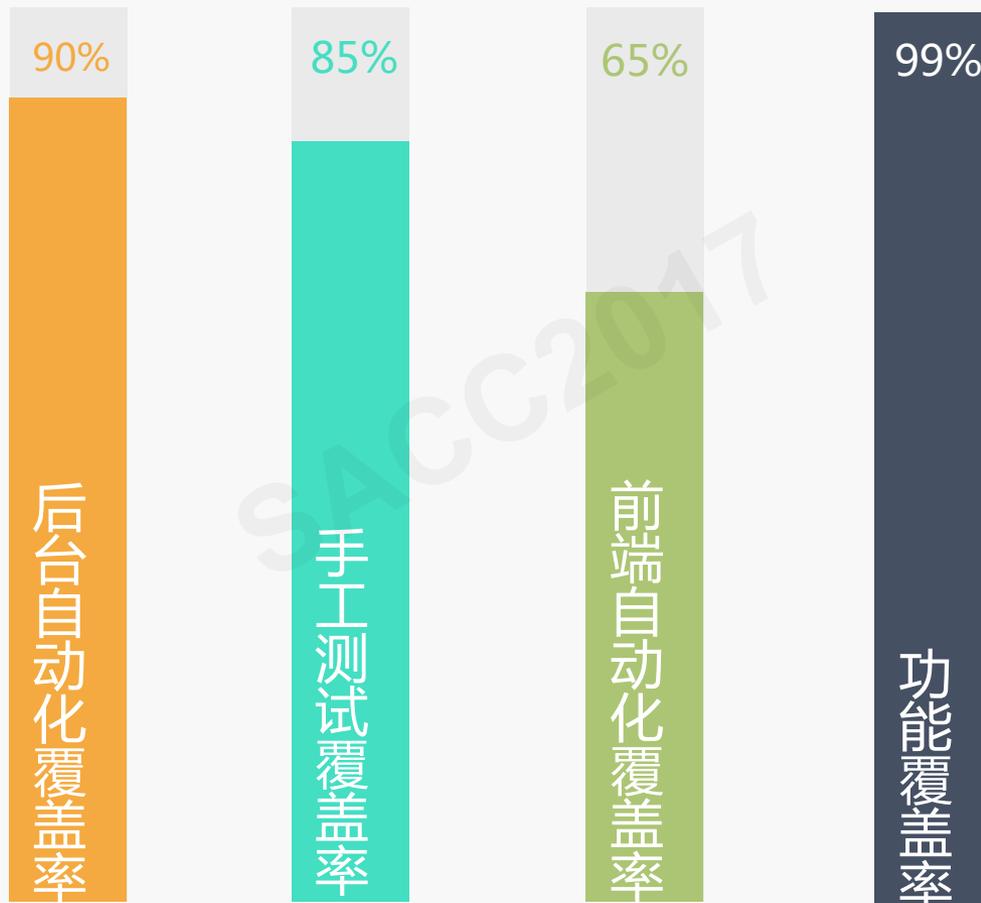
测试全面性的度量

代码覆盖度



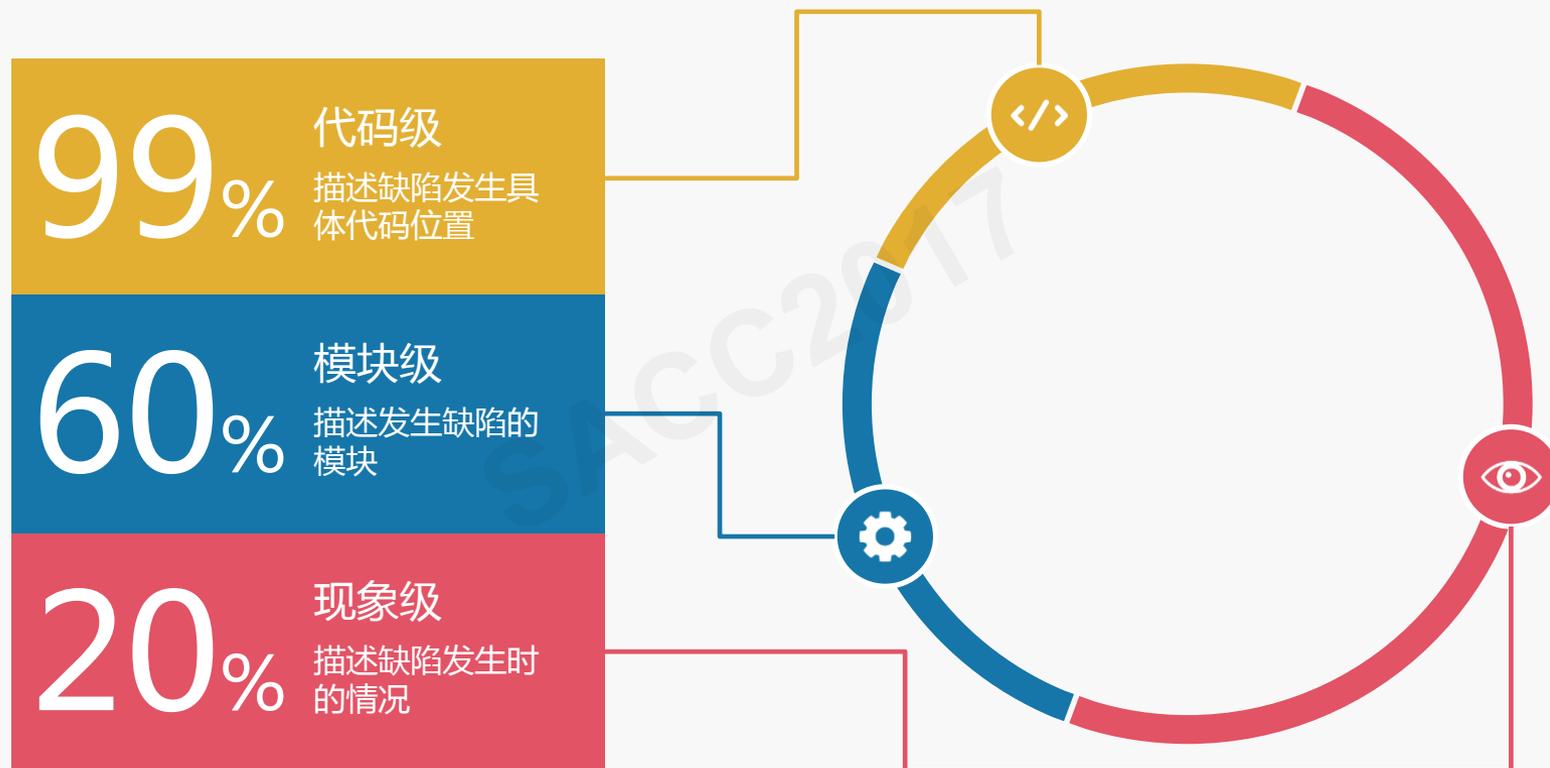
覆盖度

测试全面性的度量



缺陷定位

测试深入性的度量



缺陷定位

测试深入性的度量

【医案】【安卓】【2.0】删除书籍后，继续添加书籍然后返回书架页，新添加的书籍的选中状态被显示出来了

【医案】【2.7】【安卓】在发现页点击音频专辑信息，跳转到音频专辑详情页后不会加载收藏按钮

由于from参数=1的原因，导致没有进入initFav的流程

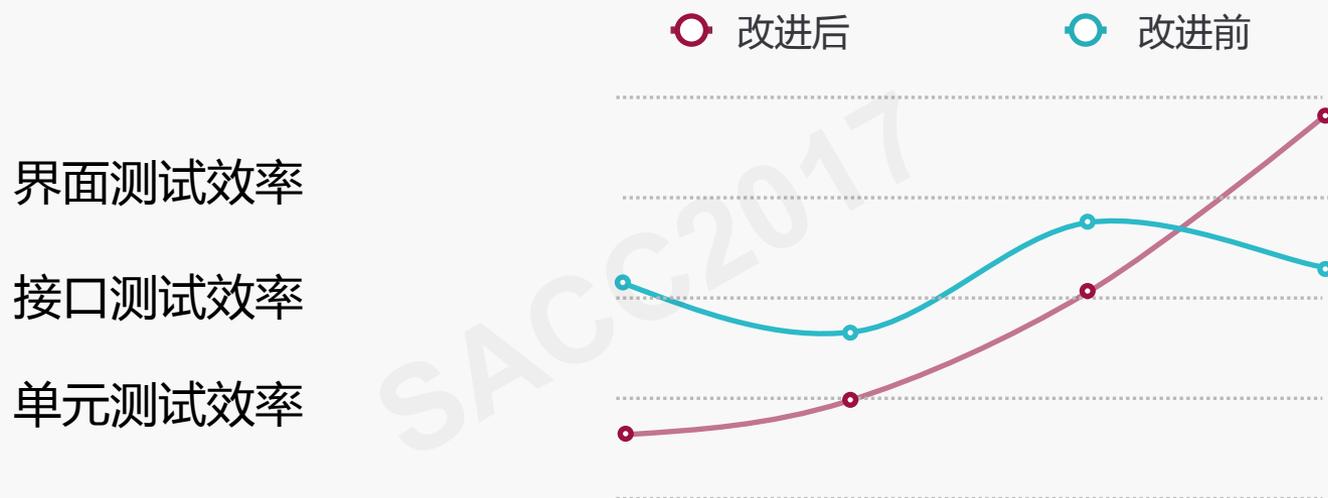
```
569 @Override
570 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
571     String albumId = (String) view.getTag();
572     gotoUrl("elephant://audio/album/" + albumId + "?from=1");
573     UigsUtil.pbUigsClickLog(AppConfig.NORMAL, PAGE, PAGE_NAME, "audio", "音频", "", position, "", "", null);
574 }
575 };
```

追踪一位工程师缺陷定位的轨迹



执行效率

测试时间性的度量



效率
原则

结果稳定 执行迅速 编写便捷 环境无关 减少依赖 合理分组

执行效率

测试时间性的度量

01

自动化测试由于需要经常重复执行，那么执行的时间也是重要的衡量指标

结果稳定

UI测试受到各种因素的影响最大，自动化case执行的稳定性要有保障

执行迅速

02

将UI自动化case进行合理的分组，可以有效的减少上线前带来的回归成本

合理分组

03

让我们的测试可以在不同的服务器环境上运行，让测试更加灵活

环境无关

04

公用逻辑进行封装，增加case的可读性，降低测试用例频繁改动的成本

编写便捷

05

使用mock工具，减少对于非目标代码的依赖，缩小测试范围

减少依赖

06

维护成本

测试持续性的度量

03.
模块复用
测试模块在类似项目复用比例如何

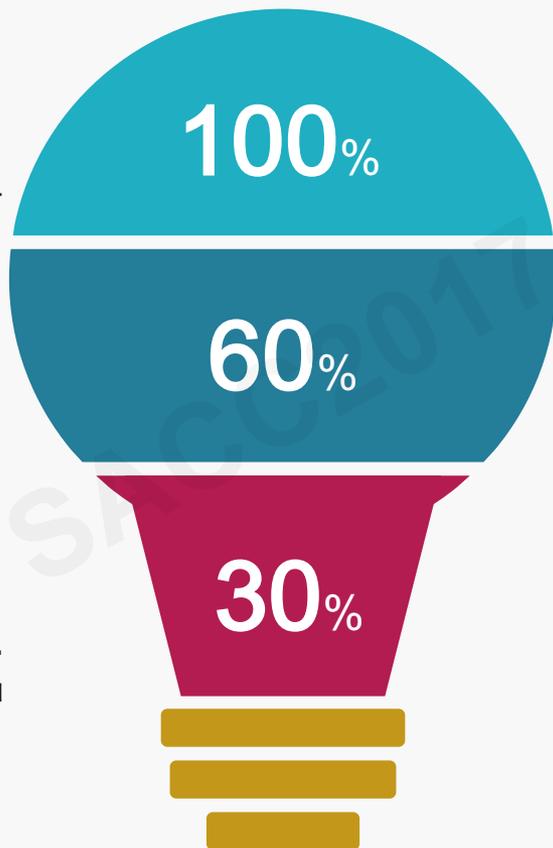
100%

60%

01.
测试管理
如何管理测试用例及缺陷

30%

02.
用例维护
如何评价测试用例维护的成本



维护成本

测试持续性的度量

测试
管理

.01

- 测试用例管理
- 缺陷追踪
- 项目追踪
- 执行记录

用例
维护

.02

- 编写成本
- 变更成本

模块
复用

.03

- 数据传输
- UI操作
- 覆盖率收集
- 测试框架

```

public void test_herb_search() throws ClassNotFoundException, InterruptedException {
    Intent intent = new Intent(mTargetContext, Class.forName(Constant.TOOLHERB_ACTIVITY_FULL_NAME));
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    mInstrumentation.startActivitySync(intent);

    // Start to input data
    Activity act = solo.getCurrentActivity();
    Log.i(debugTag, act.getClass().getName());
    EditText search_query = (EditText) solo.getView("searchquery");
    solo.clickOnView(search_query);

    solo.enterText(search_query, "ff");
    Thread.sleep(1000);
    assertTrue(search_query.getText().toString().equals("ff"));
    // solo.pressSoftKeyboardSearchButton();

    // sugg result
public void test_herb_search() throws Exception {
    ToolHerbAction.start_herb_activity(mInstrumentation, solo, mTargetContext);
    ToolHerbAction.enter_sugg_word(solo);
    ToolHerbAction.click_one_sugg(solo);
    BaseAction.checkCurrentActivityName(solo, Constant.WEBVIEW_ACTIVITY_FULL_NAME);
}
    TextView suggView = solo.getCurrentViews(TextView.class, view.getChildAt(i)).get(0);
    solo.clickOnView(suggView);
    Assert.assertTrue(
        solo.waitForActivity(Class.forName(Constant.WEBVIEW_ACTIVITY_FULL_NAME).getSimpleName(), 1000));

    Assert.assertEquals(suggView.getText().toString(), ((TextView) solo.getView("title")).getText().toString());

    ImageButton back = (ImageButton) solo.getView("back");
    solo.clickOnView(back);
    Assert.assertTrue(solo.waitForCondition(utils.activityWillClose(solo.getCurrentActivity()), 2000));
}
}

```

维护成本

测试持续性的度量



实现
通信接口封装



实现
UI通用操作封装



统一
测试用例调度机制



适配
不同项目的覆盖率方案

方案推广

测试适用性的度量



工具抽象

测试方案中是否有可以抽象为
公用工具模块



文档建设

针对测试方案建立详细的
说明文档和使用手册



技术支持

为应用该方案的其他项目成员
提供全面的技术支持

THANKS

The background features a dark, almost black, space filled with numerous bright blue particles. These particles are arranged in several distinct, curved paths that sweep across the frame from the bottom left towards the top right. A bright, white-to-blue gradient light source is positioned behind the word 'THANKS', creating a lens flare effect that illuminates the surrounding particles and the text itself.