

windows基础

软件

定义

计算机=软件+硬件

软件=程序+文档

注意：测试对象是程序和文档，不能只测程序，文档也要测。

开发阶段

阶段

第一阶段：需求分析阶段

由产品部门的需求分析人员完成

产出物：《需求规格说明书》

第二阶段：设计阶段

由系统架构师或分析师完成

产出物：《概要设计说明书》、《详细设计说明书》

第三阶段：编码阶段

由开发人员完成

产出物：程序

说明

1、需求分析阶段

由需求分析人员完成

《需求规格说明书》

2、概要设计阶段

由系统架构师或分析师完成

《概要设计说明书》

3、详细设计阶段

由系统架构师或分析师完成

《详细设计说明书》

4、编码阶段

由开发人员完成

程序

缺陷

1、缺陷定义方式1（5条）重点

说明：定义与顺序无关，只要满足其中任何一条就是bug。

- (1) 需求要求的功能没有实现。
- (2) 实现了需求没有要求的功能。（画蛇添足）
- (3) 软件中出现了指明不应该出现的错误。
- (4) 需求虽未明确说明，但是应该实现的功能没有实现。
- (5) 软件不易使用，难以理解，运行缓慢等，站在用户角度上，一切不好的地方。

2、软件缺陷的定义2（理解）--IEEE

- (1) 从软件产品的外部来看， --黑盒测试（功能）
- (2) 从软件产品的内部来看， --白盒测试

测试流程

软件项目的测试流程

1、熟悉、分析需求

2、制定测试计划

测试计划通常由测试组长或测试经理制定。

测试人员要阅读并执行测试计划。

3、设计测试（分析，编写测试用例）

4、执行测试，并记录测试结果（通过，失败）

5、记录缺陷

6、对缺陷跟踪、管理

7、测试总结（测试总结报告/测试报告）

测试总结主要总结测试中的数据。

提示：面试时经常会问到具体数据，参考总结报告，提前准备。

要求：必须熟练掌握，并且有项目经验后，要能够结合项目说明。

缺陷报告

1、什么是缺陷报告？

(1) 测试人员发现bug，通过缺陷报告将bug记录下来。

(2) 通过缺陷报告测试方将bug告知给开发方。

(3) 通过缺陷报告实现对bug的跟踪管理。

(4) 缺陷报告是测试方和开发方之间重要的沟通方式。

2、如何编写缺陷报告？

(1) 说明：在企业中缺陷是通过缺陷管理工具进行管理，常见的工具有：禅道、QC、bugfree、bugzilla、jira（鸡爪）、企业自制等

不同企业使用不同的bug管理工具，模板会有不同，但是缺陷管理的核心部分是大同小异的。

(2) 缺陷的核心组成部分：

案例：除法操作，除数是0时，程序异常退出。

1) 缺陷编号（defect ID）

记录发现bug的顺序号

在工具中缺陷编号是自动生成的，可以唯一标识每条缺陷。

说明：在企业中项目的缺陷是统一管理的，所以编号是统一生成，不是只生成自己的。

2) 缺陷标题（summary）

简明扼要的将缺陷描述出来（概括）

说明：没有标准答案，老师给的也只是参考。

3) 缺陷发现者/创建者（detected by）

测试人员发现bug，填写自己的工作账号

4) 提交缺陷的日期（detected on date）

注意：发现缺陷后应及时提交缺陷。

说明：1、发现缺陷应确认审核，尽量避免提交“假bug”2、缺陷“审核”后，不要人为延误应及时提交。

5) 指派给谁（assigned to）

指派过程：

首先：测试人员→开发方负责人

接下来：开发方负责人→具体负责的开发人员

6) 功能模块（subject）

定位在哪个模块中发现的该bug

说明：开发方负责人可以通过功能模块，明确解决该bug的对应开发人员。

7) 版本（detected in release）

说明：在软件研发过程中出现的若干临时版本，和软件最终发布的版本都是版本，都需要测试。

扩展：回归测试(重点)

回归测试就是在当前版本中，对上一个版本中测过的所有功能点，再重新测试一遍。

回归测试的必要性：通过回归测试验证当前版本中:新增加的功能对原有功能是否有影响，带来bug；修复的bug是否同时带来了新的问题。

回归测试存在重复，建议使用自动化的方式进行，可以大大提高回归测试效率。

8) 状态 (status)

表明该缺陷处于怎样的处理情况。

缺陷的状态：

新的 (new)

激活的 (open)

已解决的 (fixed)

关闭的 (closed)

被拒绝的 (rejected)

重新激活的 (reopen)

重点面试题：缺陷的跟踪管理过程（生命周期、一生、流程、步骤）？

步骤1：测试人员提交新的bug给开发方负责人。

步骤2：开发方负责人审核bug，

情况1：bug审核通过，开发方负责人将会激活bug，并指派给相应的开发人员。

情况2：bug审核未通过，开发方将拒绝bug。

步骤3：开发人员解决bug，解决后将缺陷设置为“已解决”状态。（待返测的bug）

步骤4：测试人员对已解决的bug进行返测，

情况1：返测通过，测试人员将bug关闭。

情况2：返测未通过，测试人员将缺陷重新激活，并指派回开发人员继续解决，直到返测通过，bug关闭为止。

9) 缺陷的严重程度 (severity)

表明缺陷有多糟糕，对程序的影响有多坏。

严重级别：(1,2,3,4)

致命的 (urgent)

严重的 (high)

中等的 (medium) --最多

建议性的小问题(low)

发现问题：严重级别的定义过于笼统，容易在实际工作中引起争议，所以公司通常会编写级别的详细说明，进一步规范级别的划分。

注意：严重级别定义应专业、准确，不应为了引起开发方重视而故意夸大。

不同公司、项目组严重级别详细说明可能不同，工作中要注意参考。

10) 缺陷的优先级 (priority)

表明的是希望或者建议开发方在什么时候，什么版本解决该bug。

优先级的级别：

立即解决 (urgent)

下一个版本解决 (high) --常用

软件产品发布之前解决 (medium)

尽量在软件发布之前解决 (low)

注意：对于优先级不同公司的详细说明可能不同，要注意参考。

补充：关于严重程度和优先级的面试题。

Q1:影响优先级的因素有哪些？

(1) 缺陷的严重程度，一般越严重，优先级越高。

(2) 开发人员的开发压力，开发压力越小，解决bug的优先级越高。

(3) 缺陷的影响范围，影响范围越大，bug的优先级越高。

(4) 解决bug的成本（时间），成本越小，优先级越高。

Q2：缺陷的严重程度和优先级一定是严格成正比关系吗？

缺陷的严重程度和优先级不是严格成正比的，例如：界面的错别字，就是严重级别低，但是优先级高的情况。

Q3：优先级和严重程度一旦确定后能改吗？

严重程度一旦确定不能修改；

优先级一般能改，而且经常延后处理。

Q4：是否存在发布的软件中，有bug是发现了但是没有解决的情况？

在发布的软件中可能存在发现但是没有解决的bug。此类bug应通过“bug讨论”，在权衡解决的成本，和不解决是否给用户带来严重后果，以及是否会造成法律纠纷后才能决定。

关于此类bug，软件企业通常通过后期升级版本或者打补丁的方式给予解决。

11) 缺陷描述 (description)

通过将发现的过程和数据记录下来，使开发人员能够重现该bug。（开发人员能看明白）

要求：逻辑要清晰，用语应专业、准确，描述应易读、易懂，在描述中不要出现任何评价性的语言。如实记录缺陷。