



(12)发明专利申请

(10)申请公布号 CN 107688534 A

(43)申请公布日 2018.02.13

(21)申请号 201710618312.2

(22)申请日 2017.07.26

(71)申请人 平安科技(深圳)有限公司

地址 518052 广东省深圳市福田区八卦岭
八卦三路平安大厦六楼

(72)发明人 程先燕 董嘉君

(74)专利代理机构 广州华进联合专利商标代理
有限公司 44224

代理人 王宁

(51)Int.Cl.

G06F 11/36(2006.01)

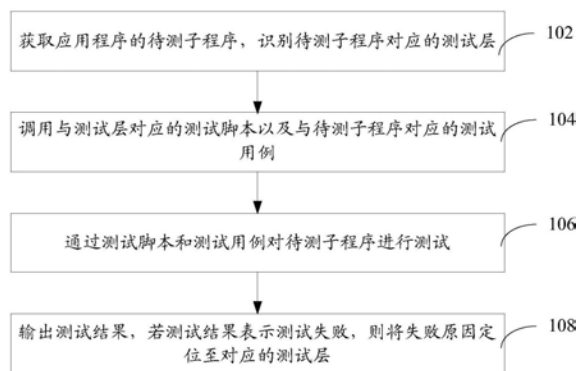
权利要求书2页 说明书8页 附图2页

(54)发明名称

分层测试方法、装置、计算机设备和存储介质

(57)摘要

本发明涉及一种分层测试方法、装置、计算机设备和存储介质,所述方法包括:获取应用程序的待测子程序,识别待测子程序对应的测试层;调用与所述测试层对应的测试脚本以及与待测子程序对应的测试用例;通过所述测试脚本和测试用例对所述待测子程序进行测试;输出测试结果,若所述测试结果显示测试失败,则将失败原因定位至对应的测试层。通过本方法能够有效提高测试失败原因的定位效率。



1. 一种分层测试方法,包括:

获取应用程序的待测子程序,识别待测子程序对应的测试层;

调用与所述测试层对应的测试脚本以及与所述待测子程序对应的测试用例;

通过所述测试脚本和测试用例对所述待测子程序进行测试;

输出测试结果,若所述测试结果表示测试失败,则将失败原因定位至对应的测试层。

2. 根据权利要求1所述的方法,其特征在于,所述测试层具有对应的测试层配置文件,所述识别待测子程序对应的测试层的步骤,包括:

获取所述待测子程序对应的版本信息;所述版本信息中包括待测子程序对应的功能点;

获取多个测试层配置文件;所述测试层配置文件中包括测试层对应的功能点;

根据所述待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到所述待测子程序所需的测试层。

3. 根据权利要求1所述的方法,其特征在于,所述识别待测子程序对应的测试层的步骤,包括:

运行所述待测子程序,检测与所述子程序发生交互的测试层;

将与所述待测子程序发生交互的测试层记录为所述待测子程序所需的测试层。

4. 根据权利要求1所述的方法,其特征在于,所述待测子程序具有对应的版本信息,所述版本信息包括版本号,所述方法还包括:

获取测试用例配置文件,所述测试用例配置文件中记录了待测子程序的版本号与对应的测试用例名称;

根据所述待测子程序的版本号获取对应的测试用例名称;

根据所述测试用例名称调用对应的测试用例。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

获取所述待测子程序的当前版本的映射文件;所述当前版本的映射文件包括当前版本的功能点、存储路径和对应的测试用例;

获取所述待测子程序的上一版本的映射文件;所述上一版本的映射文件包括上一版本的功能点、存储路径和对应的测试用例;

将所述当前版本的映射文件与所述上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径;

利用所述发生变化的功能点和/或所述发生变化的存储路径识别待测功能点;

利用所述当前版本的映射文件获取所述待测子程序中待测功能点对应的测试用例。

6. 一种分层测试装置,其特征在于,所述装置包括:

识别模块,用于获取应用程序的待测子程序,识别待测子程序对应的测试层;

测试模块,用于调用与所述测试层对应的测试脚本以及与所述待测子程序对应的测试用例;通过所述测试脚本和测试用例对所述待测子程序进行测试;

定位模块,用于输出测试结果,若所述测试结果表示测试失败,则将失败原因定位至对应的测试层。

7. 根据权利要求6所述的装置,其特征在于,所述测试层具有对应的测试层配置文件,所述识别模块还用于获取所述待测子程序对应的版本信息;所述版本信息中包括待测子程

序对应的功能点;获取多个测试层配置文件;所述测试层配置文件中包括测试层对应的功能点;根据所述待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到所述待测子程序所需的测试层。

8.根据权利要求6所述的装置,其特征在于,所述识别模块还用于运行所述待测子程序,检测与所述子程序发生交互的测试层;将与所述待测子程序发生交互的测试层记录为所述待测子程序所需的测试层。

9.一种计算机设备,包括存储器及处理器,所述存储器中储存有计算机可读指令,所述指令被所述处理器执行时,使得所述处理器执行权利要求1-5中任意一项所述方法的步骤。

10.一个或多个存储有计算机可读指令的计算机可读非易失性存储介质,计算机可读指令被一个或多个处理器执行时,使得一个或多个处理器执行权利要求1-5中任意一项所述方法的步骤。

分层测试方法、装置、计算机设备和存储介质

技术领域

[0001] 本发明涉及计算机技术领域,特别是涉及一种分层测试方法、装置、计算机设备和存储介质。

背景技术

[0002] 随着APP(Application,应用程序)的普及,各种APP开始涌现,并且APP的版本也在不断更新。对APP进行版本测试时,如果出现测试失败的情况,需要定位失败原因。在传统的自动测试中,是通过自动化测试脚本对整个APP版本进行测试。如果出现测试失败,需要在整体上对该版本排除失败原因,由此导致失败原因定位的耗时较长,效率较低。

发明内容

[0003] 基于此,有必要针对上述技术问题,提供一种能够有效提高测试失败原因定位效率的分层测试方法、装置、计算机设备和存储介质。

[0004] 一种分层测试方法,包括:

[0005] 获取应用程序的待测子程序,识别待测子程序对应的测试层;

[0006] 调用与所述测试层对应的测试脚本以及与所述待测子程序对应的测试用例;

[0007] 通过所述测试脚本和测试用例对所述待测子程序进行测试;

[0008] 输出测试结果,若所述测试结果表示测试失败,则将失败原因定位至对应的测试层。

[0009] 在其中一个实施例中,所述测试层具有对应的测试层配置文件,所述识别待测子程序对应的测试层的步骤,包括:

[0010] 获取所述待测子程序对应的版本信息;所述版本信息中包括待测子程序对应的功能点;

[0011] 获取多个测试层配置文件;所述测试层配置文件中包括测试层对应的功能点;

[0012] 根据所述待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到所述待测子程序所需的测试层。

[0013] 在其中一个实施例中,所述识别待测子程序对应的测试层的步骤,包括:

[0014] 运行所述待测子程序,检测与所述子程序发生交互的测试层;

[0015] 将与所述待测子程序发生交互的测试层记录为所述待测子程序所需的测试层。

[0016] 在其中一个实施例中,所述待测子程序具有对应的版本信息,所述版本信息包括版本号,所述方法还包括:

[0017] 获取测试用例配置文件,所述测试用例配置文件中记录了待测子程序的版本号与对应的测试用例名称;

[0018] 根据所述待测子程序的版本号获取对应的测试用例名称;

[0019] 根据所述测试用例名称调用对应的测试用例。

[0020] 在其中一个实施例中,所述方法还包括:

[0021] 获取所述待测子程序的当前版本的映射文件;所述当前版本的映射文件包括当前版本的功能点、存储路径和对应的测试用例;

[0022] 获取所述待测子程序的上一版本的映射文件;所述上一版本的映射文件包括上一版本的功能点、存储路径和对应的测试用例;

[0023] 将所述当前版本的映射文件与所述上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径;

[0024] 利用所述发生变化的功能点和/或所述发生变化的存储路径识别待测功能点;

[0025] 利用所述当前版本的映射文件获取所述待测子程序中待测功能点对应的测试用例。

[0026] 一种分层测试装置,所述装置包括:

[0027] 识别模块,用于获取应用程序的待测子程序,识别待测子程序对应的测试层;

[0028] 测试模块,用于调用与所述测试层对应的测试脚本以及与所述待测子程序对应的测试用例;通过所述测试脚本和测试用例对所述待测子程序进行测试;

[0029] 定位模块,用于输出测试结果,若所述测试结果表示测试失败,则将失败原因定位至对应的测试层。

[0030] 在其中一个实施例中,所述测试层具有对应的测试层配置文件,所述识别模块还用于获取所述待测子程序对应的版本信息;所述版本信息中包括待测子程序对应的功能点;获取多个测试层配置文件;所述测试层配置文件中包括测试层对应的功能点;根据所述待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到所述待测子程序所需的测试层。

[0031] 在其中一个实施例中,所述识别模块还用于运行所述待测子程序,检测与所述子程序发生交互的测试层;将与所述待测子程序发生交互的测试层记录为所述待测子程序所需的测试层。

[0032] 一种计算机设备,包括存储器及处理器,所述存储器中储存有计算机可读指令,所述指令被所述处理器执行时,使得所述处理器执行上述方法的步骤。

[0033] 一个或多个存储有计算机可读指令的计算机可读非易失性存储介质,计算机可读指令被一个或多个处理器执行时,使得一个或多个处理器执行上述所述方法的步骤。

[0034] 上述分层测试方法、装置、计算机设备和存储介质,根据待测子程序可以识别所需的测试层。通过调用与测试层对应的测试脚本和待测子程序对应的测试用例对待测子程序进行自动化测试。如果测试失败,则能够将失败原因准确定位至对应的测试层。在这个过程中,由于应用程序的多个子程序是分别独立进行测试的,如果出现测试失败的情况,可以只针对待测子程序对失败原因进行精准定位,不需要对整个应用程序从整体上来排除失败原因,能够有效提高测试失败原因的定位效率。

附图说明

[0035] 图1为一个实施例中分层测试方法的流程图;

[0036] 图2为一个实施例中分层测试装置的结构示意图;

[0037] 图3为一个实施例中计算机设备的结构示意图。

具体实施方式

[0038] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0039] 在一个实施例中,如图1所示,提供了一种分层测试方法,以该方法应用于测试终端为例进行说明,包括:

[0040] 步骤102,获取应用程序的待测子程序,识别待测子程序对应的测试层。

[0041] 步骤104,调用与测试层对应的测试脚本以及有待测子程序对应的测试用例。

[0042] 步骤106,通过测试脚本和测试用例对待测子程序进行测试。

[0043] 步骤108,输出测试结果,若测试结果表示测试失败,则将失败原因定位至对应的测试层。

[0044] 应用程序(Application,简称APP)在开发过程中可以按照其软件架构被拆分为多个子程序。每个子程序可以具有独立的功能。在应用程序的开发过程中,测试人员可以通过测试终端对已完成的子程序(即待测子程序)进行独立的测试,不需要等到应用程序全部开发完成才进行测试。

[0045] 测试终端上运行了自动化测试平台。自动化测试平台可以调用自动化测试脚本对待测子程序进行测试。自动化测试平台可以按照其软件架构被划分为多个层,包括控制层、界面层、数据库层和服务层等。自动化测试脚本也可以按照自动化测试平台的软件架构被相应地划分为多个层,包括控制层、界面层、数据库层和服务层等。划分后的层也可以称为测试层。通过不同的测试层可以调用不同的测试脚本进行不同的功能测试。

[0046] 在其中一个实施例中,测试层具有对应的测试层配置文件,识别待测子程序对应的测试层的步骤,包括:获取待测子程序对应的版本信息;版本信息中包括待测子程序对应的功能点;获取多个测试层配置文件;测试层配置文件中包括测试层对应的功能点;根据待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到待测子程序所需的测试层。

[0047] 待测子程序可以具有对应的标识和版本信息。测试终端可以获取待测子程序的标识,根据该标识获取对应的版本信息。版本信息中可以记录了待测子程序的一个或多个功能点。每个测试层都可以具有对应的测试层配置文件。测试层配置文件中记录了多个配置项和对应的功能点。测试终端可以根据待测子程序的版本信息识别所需的测试层。

[0048] 具体的,测试终端可以获取多个测试层对应的测试层配置文件,利用待测子程序对应的功能点对多个配置文件中查询。若在测试层配置文件中查询到待测子程序对应的一个或多个功能点,则测试终端将该测试层配置文件对应的测试层记录为待测子程序所需的测试层。由此可以准确得到每个待测子程序所需的所有测试层。

[0049] 在另一个实施例中,识别待测子程序对应的测试层的步骤,包括:运行待测子程序,检测与子程序发生交互的测试层;将与待测子程序发生交互的测试层记录为待测子程序所需的测试层。

[0050] 待测子程序运行时,可能会与一个或多个测试层发生交互。测试终端可以通过自动化测试平台调用线程检测与待测子程序发生交互的测试层,记录发生交互的测试层的名

称,将其记录为待测子程序所需的测试层。由此可以准确得到每个待测子程序所需的所有测试层。

[0051] 在识别出待测子程序所需的测试层之后,测试终端通过自动化测试平台调用与测试层对应的测试脚本。不同的测试层所对应的测试脚本不同。测试终端通过自动化测试平台还可以根据待子程序获取对应的测试用例。不同的待测子程序其功能点不同,所需的测试用例可以不同。测试用例可以存在测试终端本地,也可以存在服务器。

[0052] 测试终端通过自动化测试平台运行与测试层对应的测试脚本,调用相应的测试用例对待测子程序进行测试。在测试过程中,不同测试层的测试可以相互隔离。测试终端可以对不同待测子程序进行独立测试。待测子程序的测试结束之后,可以通过自动化测试平台输出测试结果。

[0053] 测试终端对测试结果进行断言。具体的,测试结果中包括了多个接口参数,多个接口参数可以采用正则表达式进行描述,对采用正则表达式描述的测试结果与预期结果进行比对。若一致,则表示测试通过,若不一致,则表示测试失败。若测试失败,则测试终端可以根据测试层对待测子程序的失败原因进行定位。即测试终端可以确定待测子程序的失败原因发生在哪一个测试层,由此能够实现精准定位。

[0054] 以待测子程序为“登录”进行说明。当游客登录时,测试终端可以识别出所需的测试层可以只有界面层。界面层脚本调用相应的测试用例进行测试。如果测试失败,则可以指将失败原因定位在界面层测试失败。当已注册的用户登录时,测试终端可以识别出所需的测试层可以包括界面层、数据库层和服务层。每个测试层分别调用不同的测试用例进行测试。具体的,通过待测子程序向服务器发送登录请求,登录请求中携带了用户名、密码以及验证码等信息。界面层测试脚本根据登录请求进行界面层测试。数据库层测试脚本根据用户名、密码以及验证码,调用数据库的校验逻辑进行校验测试。服务层测试脚本调用接口进行页面跳转测试。其中,界面层、数据库层以及服务层的测试可以完全隔离。如果某一层的测试结果表示该层测试失败,则可以将其失败原因定位至该层,测试失败的测试层可以是一个也可以是多个。由此实现精准定位。

[0055] 本实施例中,根据待测子程序可以识别所需的测试层。通过调用与测试层对应的测试脚本和待测子程序对应的测试用例对待测子程序进行自动化测试。如果测试失败,则能够将失败原因准确定位至对应的测试层。在这个过程中,由于应用程序的多个子程序是分别独立进行测试的,如果出现测试失败的情况,可以只针对待测子程序对失败原因进行精准定位,不需要对整个应用程序从整体上来排除失败原因,能够有效提高测试失败原因的定位效率。

[0056] 在一个实施例中,待测子程序具有对应的版本信息,版本信息包括版本号,该方法还包括:获取测试用例配置文件,测试用例配置文件中记录了待测子程序的版本号与对应的测试用例名称;根据待测子程序的版本号获取对应的测试用例名称;根据测试用例名称调用对应的测试用例。

[0057] 本实施例中,在对待测子程序进行测试之前,测试终端还需要获取相应的测试用例。测试用例可以存储在测试终端本地,也可以存储在服务器。测试用例具有对应的测试用例配置文件。测试用例配置文件可以存储在测试终端本地,也可以存储在服务器。

[0058] 不同的待测子程序可以共用一个测试用例配置文件。共用的测试用例配置文件中

可以记录了应用程序所有待测子程序进行测试所需的测试用例,包括待测子程序的标识、版本号与对应的测试用例名称。每个待测子程序也可以具有单独的测试用例配置文件。单独的测试用例配置文件中可以只记录每个待测子程序测试所需的测试用例,包括待测子程序的标识、版本号以及对应的测试用例。

[0059] 测试终端可以获取测共用的测试用例配置文件,根据待测子程序的标识和版本号在共用的测试用例配置文件中检索对应的测试用例名称。测试终端根据检索到的测试用例名称调用对应的测试用例,以便对待测子程序进行测试。

[0060] 测试终端还可以根据待测子程序的标识获取独立的测试用例配置文件,在独立的测试用例配置文件中利用待测子程序的版本号检索对应的测试用例名称。测试终端根据检索到的测试用例名称调用对应的测试用例,以便对待测子程序进行测试。通过测试用例配置文件,可以快速获取到与待测子程序版本号对应的测试用例,有利于提高测试效率。

[0061] 在一个实施例中,该方法还包括:获取待测子程序的当前版本的映射文件;当前版本的映射文件包括当前版本的功能点、存储路径和对应的测试用例;获取待测子程序的上一版本的映射文件;上一版本的映射文件包括上一版本的功能点、存储路径和对应的测试用例;将当前版本的映射文件与上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径;利用发生变化的功能点和/或发生变化的存储路径识别待测功能点;利用当前版本的映射文件获取待测子程序中待测功能点对应的测试用例。

[0062] 为了进一步提高测试效率,在待测子程序的回归测试中可以考虑缩小测试范围,只对修复后的功能点或新增的功能点进行测试。待测子程序可以包括一个或多个功能点。功能点可以通过代码文件来实现其对应的功能。测试人员通过测试终端可以根据待测子程序的功能点、代码文件对应的存储路径以及功能点对应的测试用例生成映射文件。不同版本的待测子程序可以具有不同的映射文件。

[0063] 测试终端获取待测子程序当前版本的映射文件与上一版本的映射文件,将两个映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径。当前版本可以对上一版本中的一个或多个功能点进行修复。当前版本中修复后的功能点也可以称为发生变化的功能点。当前版本还可以新增功能点,新增的功能点与上一版本的功能点相比,也可以称为发生了变化的功能点。

[0064] 待测子程序当前版本的各个功能点的代码文件与上一个版本的多个代码文件相比,存储路径也可能会发生变化。除了新增代码文件的存储路径之外,修复后的功能点所对应的代码文件可能会存在与之前不同的路径下。也可能在之前的路径下替换掉上一版本的代码文件。

[0065] 测试终端根据发生变化的功能点和/或发生变化的存储路径分别获取当前版本的代码文件以及上一版本的代码文件,将两个版本的代码文件进行比对,若不同,则将当前版本的代码文件所对应的功能点记录为待测功能点。测试终端根据待测功能点在测试用例配置文件中检索对应的测试用例名称。根据测试用例名称调用对应的测试用例,以便对待测子程序进行测试。

[0066] 本实施例中,通过对待测子程序当前版本的映射文件与上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径。进而利用发生变化的功能点和/或发生变化的存储路径可以准确识别出待测功能点,通过待测功能点获取对应的测试用

例。由此在待测子程序的回归测试中能够精准的确定测试范围,从而能够有效提高测试效率。

[0067] 在一个实施例中,如图2所示,提供了一种分层测试装置,包括:识别模块202、测试模块204和定位模块206,其中:

[0068] 识别模块202,用于获取应用程序的待测子程序,识别待测子程序对应的测试层。

[0069] 测试模块204,用于调用与测试层对应的测试脚本以及有待测子程序对应的测试用例;通过测试脚本和测试用例对待测子程序进行测试。

[0070] 定位模块206,用于输出测试结果,若测试结果表示测试失败,则将失败原因定位至对应的测试层。

[0071] 在一个实施例中,测试层具有对应的测试层配置文件,识别模块202还用于获取待测子程序对应的版本信息;版本信息中包括待测子程序对应的功能点;获取多个测试层配置文件;测试层配置文件中包括测试层对应的功能点;根据待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到待测子程序所需的测试层。

[0072] 在一个实施例中,识别模块202还用于运行待测子程序,检测与子程序发生交互的测试层;将与待测子程序发生交互的测试层记录为待测子程序所需的测试层。

[0073] 在一个实施例中,待测子程序具有对应的版本信息,版本信息包括版本号,测试模块204还用于获取测试用例配置文件,测试用例配置文件中记录了待测子程序的版本号与对应的测试用例名称;根据待测子程序的版本号获取对应的测试用例名称;根据测试用例名称调用对应的测试用例。

[0074] 在一个实施例中,测试模块204还用于获取待测子程序的当前版本的映射文件;当前版本的映射文件包括当前版本的功能点、存储路径和对应的测试用例;获取待测子程序的上一版本的映射文件;上一版本的映射文件包括上一版本的功能点、存储路径和对应的测试用例;将当前版本的映射文件与上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径;利用发生变化的功能点和/或发生变化的存储路径识别待测功能点;利用当前版本的映射文件获取待测子程序中待测功能点对应的测试用例。

[0075] 在一个实施例中,提供了一种计算机设备,包括上述实施例中提及的测试终端。如图3所示,该计算机设备包括通过系统总线连接的处理器、存储器和网络接口。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该计算机设备的非易失性存储介质存储有操作系统和计算机可读指令,该计算机可读指令被处理器执行时以实现一种分层测试方法。该计算机设备的内存储器为非易失性存储介质中的操作系统和计算机可读指令的运行提供环境。该计算机设备的网络接口用于与外部的服务器通过网络连接通信,比如,向服务器发送登录请求等。处理器执行计算机可读指令时可以执行以下步骤:

[0076] 获取应用程序的待测子程序,识别待测子程序对应的测试层;调用与测试层对应的测试脚本以及有待测子程序对应的测试用例;通过测试脚本和测试用例对待测子程序进行测试;输出测试结果,若测试结果表示测试失败,则将失败原因定位至对应的测试层。

[0077] 本领域技术人员可以理解,图3中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0078] 在一个实施例中,测试层具有对应的测试层配置文件,处理器执行计算机可读指令时还执行以下步骤:

[0079] 获取待测子程序对应的版本信息;版本信息中包括待测子程序对应的功能点;

[0080] 获取多个测试层配置文件;测试层配置文件中包括测试层对应的功能点;

[0081] 根据待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到待测子程序所需的测试层。

[0082] 在一个实施例中,处理器执行计算机可读指令时还执行以下步骤:

[0083] 运行待测子程序,检测与子程序发生交互的测试层;

[0084] 将与待测子程序发生交互的测试层记录为待测子程序所需的测试层。

[0085] 在一个实施例中,待测子程序具有对应的版本信息,版本信息包括版本号,处理器执行计算机可读指令时还执行以下步骤:

[0086] 获取测试用例配置文件,测试用例配置文件中记录了待测子程序的版本号与对应的测试用例名称;

[0087] 根据待测子程序的版本号获取对应的测试用例名称;

[0088] 根据测试用例名称调用对应的测试用例。

[0089] 在一个实施例中,处理器执行计算机可读指令时还执行以下步骤:

[0090] 获取待测子程序的当前版本的映射文件;当前版本的映射文件包括当前版本的功能点、存储路径和对应的测试用例;

[0091] 获取待测子程序的上一版本的映射文件;上一版本的映射文件包括上一版本的功能点、存储路径和对应的测试用例;

[0092] 将当前版本的映射文件与上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径;

[0093] 利用发生变化的功能点和/或发生变化的存储路径识别待测功能点;

[0094] 利用当前版本的映射文件获取待测子程序中待测功能点对应的测试用例。

[0095] 在一个实施例中,提供了一个或多个存储有计算机可读指令的计算机可读非易失性存储介质,计算机可读指令被一个或多个处理器执行时,使得一个或多个处理器执行以下步骤:

[0096] 获取应用程序的待测子程序,识别待测子程序对应的测试层;

[0097] 调用与测试层对应的测试脚本以及待测子程序对应的测试用例;

[0098] 通过测试脚本和测试用例对待测子程序进行测试;

[0099] 输出测试结果,若测试结果表示测试失败,则将失败原因定位至对应的测试层。

[0100] 在一个实施例中,测试层具有对应的测试层配置文件,计算机可读指令被处理器执行时还执行以下步骤:

[0101] 获取待测子程序对应的版本信息;版本信息中包括待测子程序对应的功能点;

[0102] 获取多个测试层配置文件;测试层配置文件中包括测试层对应的功能点;

[0103] 根据待测子程序对应的功能点对多个测试层配置文件中的测试层对应的功能点进行检索,得到待测子程序所需的测试层。

[0104] 在一个实施例中,计算机可读指令被处理器执行时还执行以下步骤:

[0105] 运行待测子程序,检测与子程序发生交互的测试层;

[0106] 将与待测子程序发生交互的测试层记录为待测子程序所需的测试层。

[0107] 在一个实施例中,待测子程序具有对应的版本信息,版本信息包括版本号,计算机可读指令被处理器执行时还执行以下步骤:

[0108] 获取测试用例配置文件,测试用例配置文件中记录了待测子程序的版本号与对应的测试用例名称;

[0109] 根据待测子程序的版本号获取对应的测试用例名称;

[0110] 根据测试用例名称调用对应的测试用例。

[0111] 在一个实施例中,计算机可读指令被处理器执行时还执行以下步骤:

[0112] 获取待测子程序的当前版本的映射文件;当前版本的映射文件包括当前版本的功能点、存储路径和对应的测试用例;

[0113] 获取待测子程序的上一版本的映射文件;上一版本的映射文件包括上一版本的功能点、存储路径和对应的测试用例;

[0114] 将当前版本的映射文件与上一版本的映射文件进行比较,得到发生变化的功能点和/或发生变化的存储路径;

[0115] 利用发生变化的功能点和/或发生变化的存储路径识别待测功能点;

[0116] 利用当前版本的映射文件获取待测子程序中待测功能点对应的测试用例。

[0117] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机可读指令来指令相关的硬件来完成,所述的计算机可读指令可存储于一非易失性计算机可读取存储介质中,该计算机可读指令在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)等。

[0118] 以上所述实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0119] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,发明专利的保护范围应以所附权利要求为准。

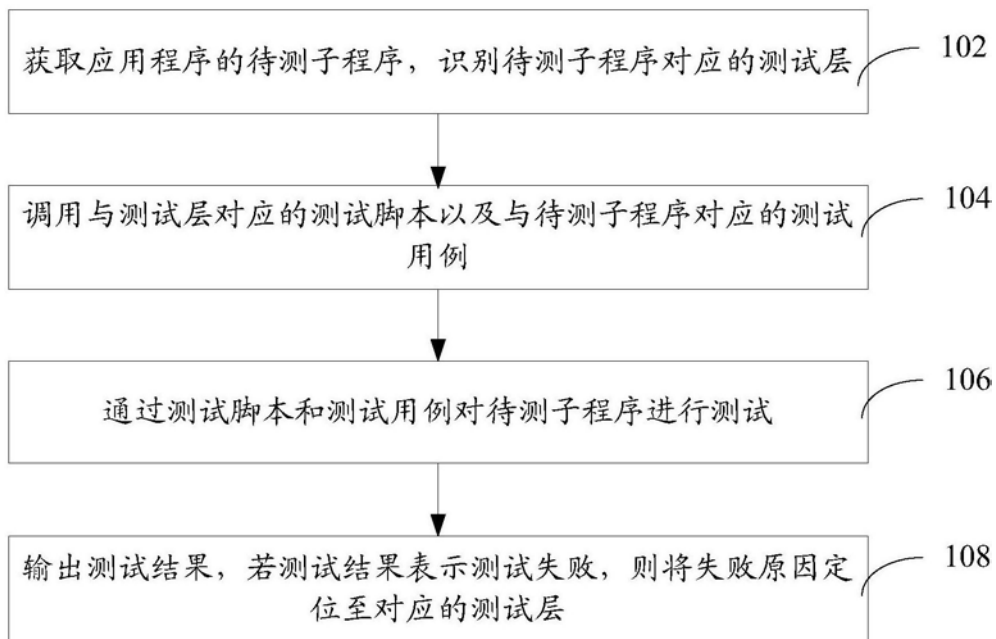


图1

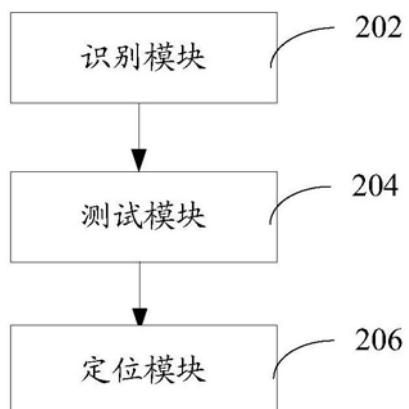


图2

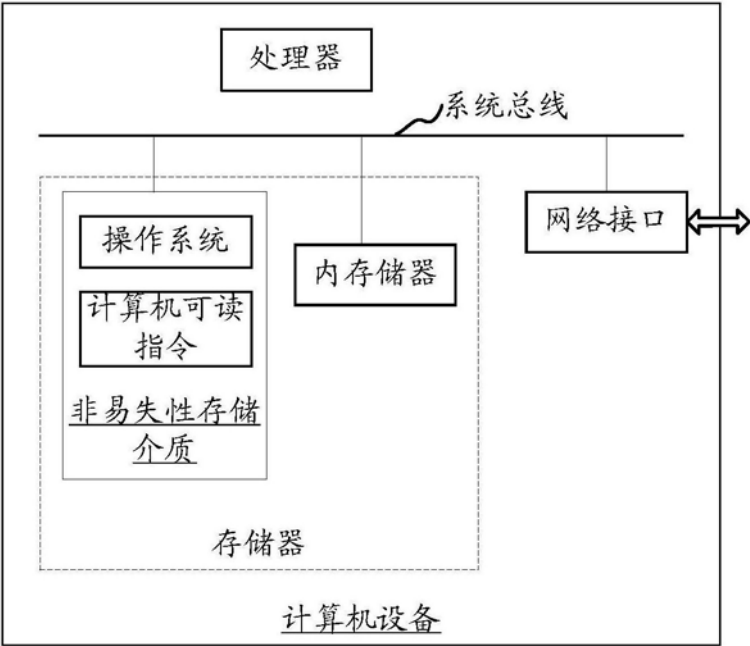


图3