

## (12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织  
国际局

(43) 国际公布日  
2020 年 9 月 24 日 (24.09.2020)



(10) 国际公布号  
**WO 2020/186810 A1**

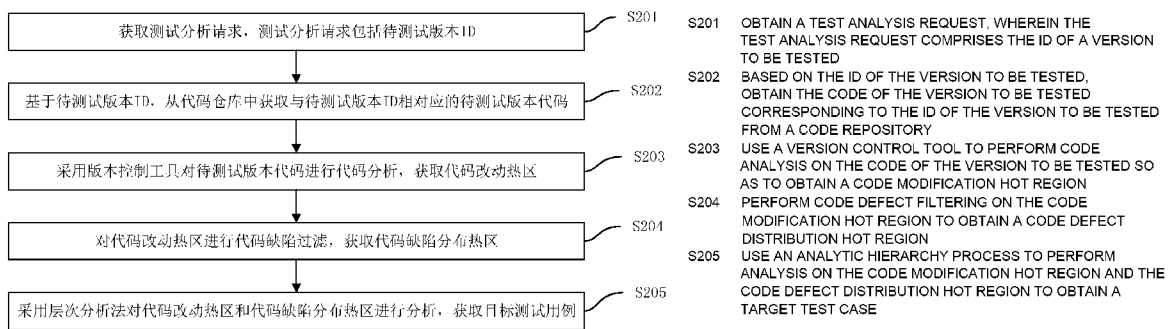
- (51) 国际专利分类号:  
**G06F 11/36** (2006.01)
- (21) 国际申请号: PCT/CN2019/120596
- (22) 国际申请日: 2019 年 11 月 25 日 (25.11.2019)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:  
201910207525.5 2019年3月19日 (19.03.2019) CN
- (71) 申请人: 深圳壹账通智能科技有限公司(ONE CONNECT SMART TECHNOLOGY CO., LTD. (SHENZHEN)) [CN/CN]; 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518000 (CN)。
- (72) 发明人: 张美苑(ZHANG, Meiyuan); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518000 (CN)。
- (74) 代理人: 深圳众鼎专利商标代理事务所(普通合伙)(SHENZHEN ZHONGDING INTELLECTUAL

PROPERTY AGENCY); 中国广东省深圳市龙岗区龙城街道中心城清林路546号城投商务中心4层/B, Guangdong 518172 (CN)。

- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,

(54) **Title:** REGRESSION TEST CASE DETERMINING METHOD AND APPARATUS, COMPUTER DEVICE, AND STORAGE MEDIUM

(54) 发明名称: 回归测试用例确定方法、装置、计算机设备及存储介质



(57) **Abstract:** A regression test case determining method and apparatus, a computer device, and a storage medium. The method comprises: obtaining a test analysis request, wherein the test analysis request comprises the ID of a version to be tested (S201); based on the ID of the version to be tested, obtaining the code of the version to be tested corresponding to the ID of the version to be tested from a code repository (S202); using a version control tool to perform code analysis on the code of the version to be tested so as to obtain a code modification hot region (S203); performing code defect filtering on the code modification hot region to obtain a code defect distribution hot region (S204); and using an analytic hierarchy process to perform analysis on the code modification hot region and the code defect distribution hot region to obtain a target test case (S205). By means of the method, when a regression test is subsequently performed according to the target test case, the repeated redundancy of a workload can be effectively reduced, and unnecessary labor waste is also reduced, and the software quality of the regression test is guaranteed.

RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,  
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

— 包括国际检索报告(条约第21条(3))。

---

**(57) 摘要：**一种回归测试用例确定方法、装置、计算机设备及存储介质。该方法包括：获取测试分析请求，所述测试分析请求包括待测试版本ID (S201)；基于所述待测试版本ID，从代码仓库中获取与所述待测试版本ID相对应的待测试版本代码 (S202)；采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区 (S203)；对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区 (S204)；采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例 (S205)。该方法可使后续基于目标测试用例进行回归测试时，可有效减少工作量的重复冗余，也减少不必要的人力浪费，并且保障回归测试的软件质量。

## 回归测试用例确定方法、装置、计算机设备及存储介质

本申请以 2019 年 3 月 19 日提交的申请号为 201910207525.5，名称为“回归测试用例确定方法、装置、计算机设备及存储介质”的中国发明申请为基础，并要求其优先权。

### 技术领域

本申请涉及软件测试技术领域，尤其涉及一种回归测试用例确定方法、装置、计算机设备及存储介质。

### 背景技术

软件测试是指在特定条件下对软件程序进行操作，以发现软件程序错误，衡量软件质量是否能满足设计要求的评估过程。当前软件测试过程中，会对不同测试阶段采用一套标准的测试策略和测试计划，以保证软件测试的顺利进行。为了保障软件程序的质量，需在软件开发过程中进行多次回归测试，该回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误的过程。当前软件程序的回归测试过程中，主要进行全功能的回归测试或者主观模糊判断回归测试范围，导致工作量重复冗余且造成不必要的人力浪费，或者导致遗漏重要的测试用例使得软件程序中特定的代码模块的质量无法得到保障。

### 发明内容

本申请实施例提供一种回归测试用例确定方法、装置、计算机设备及存储介质，以解决回归测试工作量冗余或者质量无法保障的问题。

一种回归测试用例确定方法，包括：

获取测试分析请求，所述测试分析请求包括待测试版本 ID；

基于所述待测试版本 ID，从代码仓库中获取与所述待测试版本 ID 相对应的待测试版本代码；

采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区；

对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区；

采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例。

一种回归测试用例确定装置，包括：

测试分析请求获取模块，用于获取测试分析请求，所述测试分析请求包括待测试版本 ID；

待测试版本代码获取模块，用于基于所述待测试版本 ID，从代码仓库中获取与所述待测试版本 ID 相对应的待测试版本代码；

代码改动热区获取模块，用于采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区；

代码缺陷分布热区获取模块，用于对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区；

目标测试用例获取模块，用于采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例。

一种计算机设备，包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机可读指令，所述处理器执行所述计算机可读指令时实现如下步骤：

获取测试分析请求，所述测试分析请求包括待测试版本 ID；

基于所述待测试版本 ID，从代码仓库中获取与所述待测试版本 ID 相对应的待测试版本代码；

采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区；

对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区；

采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例。

一个或多个存储有计算机可读指令的可读存储介质，所述计算机可读存储介质存储有计算机可读指

令，所述计算机可读指令被一个或多个处理器执行时，使得所述一个或多个处理器执行如下步骤：

获取测试分析请求，所述测试分析请求包括待测试版本 ID；

基于所述待测试版本 ID，从代码仓库中获取与所述待测试版本 ID 相对应的待测试版本代码；

采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区；

对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区；

采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例。

本申请的一个或多个实施例的细节在下面的附图及描述中提出。本申请的其他特征和优点将从说明书、附图以及权利要求书变得明显。

## 附图说明

为了更清楚地说明本申请实施例的技术方案，下面将对本申请实施例的描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

图 1 是本申请一实施例中回归测试用例确定方法的一应用环境示意图；

图 2 是本申请一实施例中回归测试用例确定方法的一流程图；

图 3 是本申请一实施例中回归测试用例确定方法的另一流程图；

图 4 是本申请一实施例中回归测试用例确定方法的另一流程图；

图 5 是本申请一实施例中回归测试用例确定方法的另一流程图；

图 6 是本申请一实施例中回归测试用例确定方法的另一流程图；

图 7 是本申请一实施例中回归测试用例确定方法的另一流程图；

图 8 是本申请一实施例中回归测试用例确定装置的一示意图；

图 9 是本申请一实施例中计算机设备的一示意图。

## 具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

本申请实施例提供的回归测试用例确定方法，该回归测试用例确定方法可应用如图 1 所示的应用环境中。具体地，该回归测试用例确定方法应用在软件测试系统中，该软件测试系统包括如图 1 所示的客户端和服务端，客户端与服务端通过网络进行通信，用于进行回归测试分析，以确定目标测试用例，以便基于该目标测试用例进行回归测试过程中，既可保障回归测试质量又可保障回归测试效率。其中，客户端又称为用户端，是指与服务端相对应，为客户提供本地服务的程序。客户端可安装在但不限于各种个人计算机、笔记本电脑、智能手机、平板电脑和便携式可穿戴设备上。服务端可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

在一实施例中，如图 2 所示，提供一种回归测试用例确定方法，以该方法应用在图 1 中的服务器为例进行说明，包括如下步骤：

S201：获取测试分析请求，测试分析请求包括待测试版本 ID。

其中，测试分析请求是用于触发服务器进行回归测试分析，以确定本次回归测试所需采用的测试用例的请求。待测试版本 ID 是用于唯一识别所要进行测试缺陷分析的特定版本的软件程序的标识。每一待测试版本 ID 对应一个需要进行测试缺陷分析的特定版本的软件程序，例如微信 V6.6.7。

S202：基于待测试版本 ID，从代码仓库中获取与待测试版本 ID 相对应的待测试版本代码。

其中，代码仓库是用于存储特定软件程序所有版本的代码的数据库。待测试版本代码是指最后一次代码修改之后，需要进行回归测试的版本代码。每一待测试版本代码 ID 对应一待测试版本代码。可以理解地，代码仓库中存储有与待测试版本 ID 相对应的待测试版本代码，以便基于该待测试版本代码进行后续分析。具体地，服务器与测试结果数据库相连，该测试结果数据库中存储有与待测试版本代码同

源的至少一个历史版本测试数据,该历史版本测试数据是指在最后一次代码修改之前所形成的版本代码在测试过程中形成的数据。

S203: 采用版本控制工具对待测试版本代码进行代码分析,获取代码改动热区。

其中,版本控制工具是用于提供完备的版本管理功能,用于存储、追踪目录(文件夹)和文件的修改历史的工具,是软件开发者的必备工具,是软件公司的基础设施。版本控制软件的最高目标,是支持软件公司的配置管理活动,追踪多个版本的开发和维护活动,及时发布软件。本实施例中,服务器所采用的版本控制工具具体为分布式版本控制系统(Distributed Version Control System,简称 DVCS),在 DVCS 中客户端并不只提取最新版本的文件快照,而是把代码仓库完整地镜像下来。这么一来,任何一处协同工作作用的服务器发生故障,事后都可以用任何一个镜像出来的本地仓库恢复,其原因在于每一次的提取操作,实际上都是一次对代码仓库的完整备份。当前应用 DVCS 的版本控制工具包括但不限于 Git, Mercurial, Bazaar 以及 Darcs 等工具。进一步地,服务器采用 Git 这一版本控制工具对待测试版本代码以及其对应的历史版本测试数据进行代码分析,以确定代码改动热区。该版本控制工具 Git 具有如下技术优点:1) 适合分布式开发,强调个体;2) 公共服务器压力和数据量都不会太大;3) 速度快、灵活;4) 任意两个开发者之间可以很容易的解决冲突;5) 离线工作。

代码改动热区是指采用版本控制工具对待测试版本代码和对应的历史版本测试数据进行对比分析,获取到的被频繁改动的代码区域。可以理解地,该代码改动热区是指被频繁改动的代码区域,此处的频繁改动通过代码改动行数和代码改动次数这两个指标来确定,而代码改动的原因有可能是因为代码复杂度过高或者代码耦合性太强而造成新增功能需要改动这些代码,也有可能仅仅是因为代码实现过程中有问题需要频繁改动来修复代码缺陷。具体地,服务器采用版本控制工具 Git 对待测试版本代码以及与该待测试版本代码同源的所有历史版本测试数据进行代码分析,以获取相应的代码改动行数和代码改动次数,并根据该代码改动行数和代码改动次数进行分析,获取相应的代码改动热区。

S204: 对代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区。

由于代码改动热区是基于代码改动次数和代码改动行数这两个维度进行综合评估后确定的被频繁改动的代码区域,而频繁改动的原因可能是基于代码缺陷也可能不是基于代码缺陷,在代码改动的原因是基于代码缺陷时,则该代码改动热区可能对应于高风险代码模块,即存在较多代码缺陷的功能模块。因此,在回归测试阶段,除了关注代码改动次数和代码改动行数,以确定改动代码热区之外,还需对代码改动热区中是否存在高风险代码模块进行识别判断,以过滤出高风险代码模块,从而确定代码缺陷分布热区。即通过在代码改动热区中,加入代码缺陷过滤来确定由于代码缺陷改动所形成的代码缺陷分布热区。

具体地,服务器中预先设置有助于匹配代码缺陷的正则表达式,在确定代码改动热区之后,采用该正则表达式对代码改动热区的代码文件进行匹配,获取匹配成功的代码文件对应的代码区域为代码缺陷分布热区。例如,服务器对代码改动热区进行代码缺陷过滤过程所采用的过滤方式:通过代码文件的名称(基于代码缺陷修复后形成的代码文件一般都会标记修复 XXX. bug)对应的正则表达式来判断过滤,以达到确定代码缺陷分布热区的目的。

S205: 采用层次分析法对代码改动热区和代码缺陷分布热区进行分析,获取目标测试用例。

层次分析法(Analytic Hierarchy Process,简称 AHP),是指将与决策总是有关的元素分解成目标、准则、方案等层次,在此基础上进行定性和定量分析的决策方法。在层次分析法中,从上到下配置有目标层、准则层和方案层。其中,目标层是决策的目的、要解决的问题,于本实施例中,目标层输出最终需要进行回归测试的目标测试用例。方案层是指决策时的备选方案,于本实施例中,方案层输入软件版本测试过程中预先配置的至少一个原始测试用例,例如,原始测试用例 1、原始测试用例 2……原始测试用例 Z,其中,Z 指代最后一个原始测试用例。准则层是中间层,是指决策过程中考虑的因素、决策的准则,即为实现决策的目的所采用的措施和方案,于本实施例中,将步骤 S203 所获取的代码改动热区和步骤 S204 所获取的代码缺陷分析热区作为准则层因素。

本实施例中,将软件版本测试过程中预先配置的至少一个原始测试用例作为 AHP 中方案层的输入,基于代码改动热区和代码缺陷分析热区形成准则层,通过执行 AHP 的具体算法,以获取目标层输出的目

标测试用例。该目标测试用例是采用 AHP 算法对代码改动热区和代码缺陷分析热区进行分析，所确定的在下次回归测试过程中可采用的测试用例。可以理解地，该目标测试用例综合考虑该待测试版本代码预先配置的至少一个原始测试用例中，其对应的代码改动热区和代码缺陷分布热区等信息进行综合分析，以挑选出可作为下次回归测试的测试用例（即目标测试用例）。可以理解地，目标测试用例的确定，以使后续基于目标测试用例进行回归测试时，可有效减少工作量的重复冗余，也减少不必要的人力浪费，并且保障回归测试的软件质量。

本实施例所提供的回归测试用例确定方法中，通过采用版本控制工具对待测试版本代码进行分析，可快速获取对应的代码改动热区，该代码改动热区反映软件代码被改动的频率；在对代码改动热区进行代码缺陷过滤，以获取代码缺陷分布热区，以确定高风险代码模块对应的代码区域；再采用层次分析法对代码改动热区和代码缺陷分布热区进行分析，获取目标测试用例，以使后续基于目标测试用例进行回归测试时，可有效减少工作量的重复冗余，也减少不必要的人力浪费，并且保障回归测试的软件质量。

在一实施例中，如图 3 所示，采用版本控制工具对待测试版本代码进行代码分析，获取代码改动热区，具体包括如下步骤：

S301：采用版本控制工具执行日志查询命令，获取与待测试版本代码相对应的历史版本测试数据。

其中，日志查询命令是用于查询软件程序测试过程中所形成的日志的命令，该日志查询命令具体用于查询特定存储目录中测试时间在预设时间段内的测试日志，基于该测试日志获取历史版本测试数据。具体地，用户可通过客户端向版本控制工具 Git 输入相应的日志查询命令，以获取与待测试版本代码相对应的历史版本测试数据，获取过程简单快捷。例如，版本控制工具可执行“`git log -prettyformat: '[%h] %an %ad %s' -date=short -numstat -before=yy-mm-dd -after=yy-mm-dd &gt; react_evo.log`”这一日志查询命令，获取相应的测试日志，基于该测试日志中记录的改动信息和改动文件获取历史版本测试数据。其中，改动信息具体反映代码改动情况的信息，如最近一次修改代码时，删除代码行数和新增代码行数等信息。改动文件是指基于改动信息对应的改动代码所形成的文件。

进一步地，为了保证时间的一致性，还可在日志查询命令中将时间切换到系统当前时间，例如，系统当前时间为 2018 年 8 月 27 日，则可在日志查询命令中加入“`git checkout `git rev-list -n 1 --before="2018-08-27" master``”，以获取在 2018 年 8 月 27 日之前，待测试版本代码对应的改动信息和改动文件，从而获取历史版本测试数据。

S302：采用版本控制工具对待测试版本代码和历史版本测试数据进行分析，获取代码改动热区。

由于本实施例中采用的版本控制工具为 Git，Git 中设置有可实现检测代码改动次数和代码改动行数的功能模块，通过执行相应的功能模块对待测试版本代码和历史版本测试数据进行分析，以获取待测试版本代码中每一代码模块对应的代码改动次数和代码改动行数，再对代码改动次数和代码改动行数进行综合分析，以确定代码改动热区。其中，代码模块是待测试版本代码中可实现某一功能的代码的组合。

本实施例所提供的回归测试用例确定方法中，先执行日志查询命令，以获取历史版本测试数据，再采用版本控制工具中的功能模块对待测试版本代码和历史版本测试数据进行分析，可快速获取代码改动热区，通过该代码改动热区了解待测试版本代码中每一代码模块的改动情况，以便后续综合分析确定下次回归测试所需采集的目标测试用例，可对代码改动热区对应的代码模块进行测试，以避免回归测试过程中遗漏对代码改动热区对应的代码模块进行测试。

在一实施例中，如图 4 所示，采用版本控制工具对待测试版本代码和历史版本测试数据进行分析，获取代码改动热区，具体包括如下步骤：

S401：采用 Code Maat 工具对待测试版本代码和历史版本测试数据进行分析，获取待测试版本代码中每一代码模块对应的代码改动次数。

具体地，Code Maat 工具是采用 Clojure 编写，用于挖掘和分析待测试版本代码中的代码改动次数的功能模块，该 Code Maat 工具设置在版本控制工具 Git 中。具体地，服务器在用户自己重定向的 Git 的日志文件 `react_evo.log` 中，采用 Code Maat 去获取每一代码模块对应的代码改动次数。该代码改动次数反映某一代码模块在最后一次测试之前被改动过的次数。

本实施例中，步骤 S401 具体包括如下步骤：（1）可先通过在版本控制工具中执行“java -jar code-maat/target/code-maat-1.1-SNAPSHOT-standalone.jar -l react\_evo.log -c git -a summary”这一命令行，获取待测试版本代码与其之前的版本代码对应的汇总数据，该汇总数据可反馈待测试版本代码与其之前的版本代码的提交次数、涉及代码文件的具体情况和开发人员的信息等。（2）再执行“java -jar ../code-maat/target/code-maat-1.1-SNAPSHOT-standalone.jar -l react\_evo.log -c git -a revisions &gt; react\_freqs.csv”这一命令行，获取待测试版本代码中每一代码模块对应的代码改动行数。

S402：采用 Cloc 工具对待测试版本代码和历史版本测试数据进行分析，获取待测试版本代码中每一代码模块对应的代码改动行数。

具体地，Cloc 工具是在版控制工具 Git 中配置的代码统计工具 Cloc，Cloc 是一款使用 Perl 语言开发的开源代码统计工具，支持多平台使用、多语言识别，能够计算指定目标文件或文件夹中的文件数（files）、空白行数（blank）、注释行数（comment）和代码行数（code）。其中，代码改动行数是判断代码改动热区的另一维度指标，具体是指该待测试版本代码相对于上一版本版本代码的改动行数，该代码改动行数包括删除代码行数和新增代码行数。代码改动行数这个维度简单粗暴，并且有两个好处，其一是能够查找方便快捷；其二是对于不同编程语言，它是中立的指标。本实施例中，采用 Cloc 工具作为对代码改动行数进行分析的工具，它使用 Perl 编写，并且能够得到针对编程语言、文件、空格、注释和代码本身的很直观的输出。

S403：对每一代码模块对应的代码改动次数和代码改动行数进行标准化和加权处理，获取每一代码模块对应的综合评估指标。

具体地，对每一代码模块的代码改动次数和代码改动行数进行标准化处理，是指将每一代码模块的代码改动次数和代码改动行数进行无量纲化处理，使其具有可比性，可进行后续的加权处理。

本实施例中，服务器预先存储有标准化数据转换表，该标准化数据转换表中存储有代码改动次数对应的次数标准化分值，并存储有代码改动行数对应的行数标准化分值。服务器在获取到每一代码模块的代码改动次数和代码改动行数之后，查询该标准化数据转换表，获取相对应的次数标准化分值和行数标准化分值。并且，服务器预先存储有代码改动次数和代码改动行数对应的权重。服务器对每一代码模块的代码改动次数和代码改动行数进行加权处理，是指将每一代码模块的次数标准化分值和行数标准化分值与其相应的权重进行加权计算，获取到相应的综合评估指标。该加权计算公式为  $X = \sum x_i h_i$ ，其中，

$X$  为综合评估指标， $x_i$  分别是指次数标准化分值或行数标准化分值， $h_i$  分别是指代码改动次数对应的权重或代码改动行数对应的权重。依据测试经验可知，代码改动次数比代码改动行数对代码改动热区的认定更为重要，因此，可设置代码改动次数要比代码改动行数拥有更多的权重，以获取每一代码模块的综合评估指标。

本实施例中，若代码改动次数越多，其次数标准化分值越大；若代码改动行数越多，其行数标准化分值越大，则若计算出的综合评估指标越大，反映了该代码模块对应的代码改动次数较多和/或代码改动行数较多，因此，可根据该综合评估指标确定其代码改动热区。

S404：基于每一代码模块对应的综合评估指标，获取代码改动热区。

具体地，基于每一代码模块对应的综合评估指标获取代码改动热区具体包括如下步骤：将每一代码模块的综合评估指标与预设指标阈值进行比较，若该综合评估指标大于预设指标阈值，则认定该综合评估指标对应的代码模块所形成的区域为代码改动热区；反之，若综合评估指标不大于预设指标阈值，则认定该综合评估指标对应的代码模块所形成的区域不为代码改动热区。其中，预设指标阈值是指预先设置的用于评估代码改动热区对应的指标的阈值。

由于代码改动次数和代码改动行数这两个单一维度不足以说明某个代码模块为代码改动热区，通过综合评估指标将每一代码模块的代码改动次数和代码改动行数进行整合，形成新的综合维度，以使基于综合评估指标确定的代码改动热区更客观和合理。

进一步地,服务器还可以采用 Circle Packing 或者其他图形转换工具将代码修改热区展示出来,以使用户可直观地查询代码改动热区的信息。例如,在采用 Circle Packing 所展示的图像中,每个圆的直径越大,代表这个圆对应的代码改动行数越多;而每一圆的颜色越深,代表这个圆的代码改动次数越多,改动越频繁;可基于每一代码模块对应的代码改动次数和代码改动行数确定其综合评估指标,将大于预设指标阈值的综合评估指标对应的代码模块进行突出显示,以使用户了解该代码模块所在区域为代码改动热区。

本实施例所提供的回归测试用例确定方法中,采用 Code Maat 工具和 Cloc 工具分别对待测试版本代码和历史版本测试数据进行分析,分别获取代码改动次数和代码改动行数,获取过程简单方便;再对每一代码模块的代码改动次数和代码改动行数进行标准化和加权处理,可获取可综合反映代码改动情况的综合评估指标;最后,基于综合评估指标和预设指标阈值的比较结果,可快速获取相应的代码改动热区,以使所获取的代码改动热区更具有客观性。

在一实施例中,如图 5 中,在对代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区的步骤之前,回归测试用例确定方法还包括如下步骤:

S501: 基于待测试版本 ID 查询测试结果数据库,获取与待测试版本 ID 相对应的至少一个代码缺陷数据。

其中,测试结果数据库是用于存储在该待测试版本 ID 对应的特定版本的软件程序的测试结果的数据库。该测试结果包括测试成功和测试失败两种结果,其中,基于测试结果为测试失败的软件代码所形成的数据为代码缺陷数据。即代码缺陷数据是在特定版本的软件程序在测试过程中代码存在缺陷的数据。本实施例中,代码缺陷数据是指在该待测试版本 ID 对应的软件程序在测试过程中出现的所有存在缺陷的代码对应的数据。

S502: 对代码缺陷数据进行缺陷度量分析,获取代码缺陷数据的缺陷度量值。

其中,缺陷度量值是依据预先设置的度量值评估规则对代码缺陷数据进行综合评估之后确定度量值。该缺陷度量值可客观反映每一代码缺陷数据的缺陷严重程度,一般来说,缺陷度量值越大,其缺陷越严重;反之,缺陷度量值越小,其缺陷越不严重。具体地,对代码缺陷数据进行缺陷度量分析是指依据预先设置的多个评估维度对代码缺陷数据进行综合评估,以使获取到的代码缺陷数据对应的缺陷度量值可综合反映该代码缺陷数据的缺陷严重程度。本实施例中,预先设置的度量值评估规则是依据影响代码缺陷严重程度的多个评估维度进行综合评估的规则。该评估维度具体包括但不限于功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数。

S503: 依据代码缺陷数据的缺陷度量值,获取代码缺陷数据对应的缺陷优先级,并配置每一代码缺陷数据对应的正则表达式。

由于缺陷度量值是依据多个评估维度对代码缺陷数据进行综合评估后确定的度量值,每一缺陷度量值可反映该代码缺陷数据的严重程度,这个严重程度可体现为缺陷优先级。一般来说,缺陷度量值越大,其对应的缺陷优先级越高。

本实施例中,依据代码缺陷数据的缺陷度量值,获取代码缺陷数据对应的缺陷优先级具体包括如下步骤:服务器可基于代码缺陷数据的缺陷度量值查询风险等级转换表,以确定其对应的缺陷风险等级,基于该缺陷风险等级确定相应的缺陷优先级。风险等级转换表中可将缺陷度量值的取值范围划分成若干个风险等级范围,每一风险等级范围对应的一缺陷风险等级,则服务器可根据该代码缺陷数据对应的缺陷度量值确定其所属的风险等级范围,将其所属的风险等级范围对应的缺陷风险等级确定为该缺陷度量值对应的缺陷风险等级。该缺陷风险等级可以包括紧急风险等级、高级风险等级、一般风险等级和低级风险等级等类型,也可以设置成其他类型。一般来说,缺陷度量值的取值越高,其缺陷风险等级越高,相应地,缺陷优先级越高。具体地,配置每一代码缺陷数据对应的正则表达式是指配置可匹配出相应的代码缺陷的正则表达式。

相应地,步骤 S204 中,对代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区,具体包括如下步骤:依据代码缺陷数据对应的缺陷优先级,采用代码缺陷数据对应的正则表达式对代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区。具体地,服务器会依据每一代码缺陷数据对应的缺陷优先级,



依次采用紧急风险等级、高级风险等级、一般风险等级和低级风险等级对应的代码缺陷数据对应的正则表达式，对代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区，以便快速获取到高风险代码模块对应的代码缺陷分布热区，提高代码缺陷分布热区的获取效率。一般来说，缺陷优先级在前的代码缺陷数据对应的代码模块不为代码缺陷分布热区，则缺陷优先级在后的代码缺陷数据对应的代码模块同样不为代码缺陷分布热区，无需进行后续处理，以节省处理时间，提高处理效率。

本实施例所提供的回归测试用例确定方法中，通过获取待测试版本代码的每一代码缺陷数据的缺陷度量值，以确定其缺陷优先级，由于缺陷度量值的确定参考多个评估维度确定，使得最终确定的缺陷优先级具有较高的客观性。然后，依据缺陷优先级的顺序，采用代码缺陷数据对应的正则表达式对代码改动热区进行代码缺陷过滤，以便快速获取代码缺陷分布热区，提高代码缺陷分布热区的获取效率。

在一实施例中，如图6所示，步骤S502中，对代码缺陷数据进行缺陷度量分析，获取代码缺陷数据的缺陷度量值，具体包括如下步骤：

S601：基于与待测试版本ID相对应的抽象语法树对代码缺陷数据进行语法树分析，获取与代码缺陷数据相对应的功能核心程度。

其中，抽象语法树（abstract syntax code, AST）是源代码的抽象语法结构的树状表示，树上的每个节点都表示源代码中的一种结构，之所以说是抽象的，是因为抽象语法树并不会表示出真实语法出现的每一个细节（如嵌套括号被隐含在树的结构中，并没有以节点的形式呈现）。具体地，服务器可预先采用 javascript Parser 这一源码解析器，对采用 JAVA 语言编辑的与待测试版本ID相对应的软件程序进行转化处理，以转化成与待测试版本ID相对应的抽象语法树，再进一步转化为字节码或直接生成机器码，以便后续进行程序分析。

本实施例中，服务器采用 emma 代码分析工具分析代码缺陷数据在与待测试版本ID相对应的抽象语法树上的运行流向，确定代码缺陷数据在抽象语法树中的位置，并根据其所在的位置确定其对应的功能核心程度，可采用R来表示。例如，若该代码缺陷数据处于抽象语法树中的主干位置，则其功能核心程度最高，确定为R0级别；若该代码缺陷数据处于抽象语法树中的次主干位置，则其功能核心程度次之，确定为R1级别；若该代码缺陷数据处于抽象语法树中的次干位置，则其功能核心程度更低，确定为R2级别……依此类推，确定代码缺陷数据在抽象语法树中的位置，从而确定其对应的功能核心程度。

可以理解地，通过预先创建的与待测试版本ID相对应的抽象语法树对代码缺陷数据进行语法树分析，可快速获取该代码缺陷数据对应的功能核心程度，可反映该代码缺陷数据在整个待测试版本ID对应的软件程序中的重要程度，以使后续可将该功能核心程度作为评估其对应的缺陷度量值的一个评估因素，综合评估该代码缺陷数据对应的缺陷度量值，以便对待测试版本ID对应的软件程序进行综合分析。

S602：基于代码缺陷数据查询与待测试版本ID相对应的埋点数据库，获取与代码缺陷数据相对应的页面使用频率和页面访问步长。

其中，埋点是指在软件程序的关键代码植入用于实现数据统计功能的代码。埋点的设置，主要用于追踪用户行为，从而统计关键代码的使用程度，以便进行数据分析。埋点数据库是用于存储埋点上传的埋点数据的数据库，每一埋点数据可反映该埋点对应的关键代码被触发或操作的情况。相应地，与待测试版本ID相对应的埋点数据库，用于存储与待测试版本ID相对应的软件程序在测试过程中埋点被触发所形成的所有埋点数据，以便基于该埋点数据进行数据分析。

与代码缺陷数据相对应的页面使用频率是指通过埋点统计确定代码缺陷数据对应的功能页面的点击量，并根据点击量确定其对应的页面使用频率。与代码缺陷数据相对应的页面使用频率的确定过程包括如下步骤：基于代码缺陷数据查询与待测试版本ID相对应的埋点数据库，获取与代码缺陷数据相对应的软件代码在测试周期内所触发的埋点数据；根据埋点数据确定代码缺陷数据相对应的功能页面的点击量，基于点击量查询预先设置的频率转换信息表，获取与点击量相对应的页面使用频率。其中，测试周期是指对该待测试版本ID对应的软件程序进行测试的周期。频率转换信息表是用于限定点击量与对应的频率之间的转换关系的信息表。在待测试版本ID对应的软件程序的测试周期内，每次访问该代码缺陷数据对应的功能页面，则该代码缺陷数据的软件代码中内置的埋点会触发一埋点数据，使得其功能页面的点击量被记录1次；在对同一功能页面的多次访问，其点击量累计，因此，可通过该代码缺陷数

据对应的软件代码在测试周期内所触发的埋点数据，确定其对应的功能页面的点击量。然后，基于该点击量查询频率转换信息表，以获取与待测试版本 ID 相对应的页面访问频率。本实施例中，代码缺陷数据对应的页面使用频率包括 V0、V1 和 V2 三种，分别对应高频、一般和低频三种情况。一种来说，代码缺陷数据对应的页面使用频率越高，说明其对应的软件代码被触发的频率越大，在待测试版本 ID 相对应的软件程序中越重要。

与代码缺陷数据对应的页面访问步长是指通过埋点统计确定该代码缺陷数据对应的功能页面的操作步骤长度。与代码缺陷数据相对应的页面访问步长的确定过程包括如下步骤：基于代码缺陷数据查询与待测试版本 ID 相对应的埋点数据库，获取代码缺陷数据所触发的埋点数据，根据埋点数据确定其对应的页面访问步长。具体地，在待分析版本对应的软件程序的埋点设置过程中，通过页面信息表预先记录每一埋点对应的功能页面的位置。服务器根据该埋点所触发的埋点数据可确定其对应的功能页面，根据软件程序开发过程中记录的页面信息表，可确定该功能页面的操作步骤，以获取与代码缺陷数据相对应的页面访问步长。例如，在微信这一软件程序开发过程中记录有微信刷朋友圈功能对应的页面信息表，其操作步骤包括：Step1：点击发现，进入发现页面；Step2：点击朋友圈，进入朋友圈页面，可查看朋友圈信息；若代码缺陷数据对应的软件代码所触发的埋点数据中确定出现缺陷的位置在查看朋友圈这个页面，则查看朋友圈这个功能在微信中的页面访问步长 D 就是 2。本实施例中，代码缺陷数据对应的页面访问步长包括 D0、D1 和 D2 三种，分别对应 1-2 步、3-4 步和 5 步以上这三种情况。一种来说，与代码缺陷数据相对应的页面访问步长越短，说明其对应的功能页面越经常触发，在待测试版本 ID 对应的软件程序中也重要。

S603：对代码缺陷数据进行重复测试，获取与代码缺陷数据相对应的复现概率。

其中，与代码缺陷数据相对应的复现概率是指在软件测试过程中，该代码缺陷数据对应的软件代码重复出现测试失败的概率。本实施例中，可设置代码缺陷数据对应的复现概率分别必然事件、高概率事件和低概率事件这三种，分别对于 100% 的概率会出现缺陷、50%-100% 的概率会出现缺陷和 50% 以下的概率会出现缺陷这三种情况，分别用 G0、G1 和 G2 来表示。具体地，服务器在获取代码缺陷数据之后，可对该代码缺陷数据对应的软件代码进行重复测试，以确定重复出现该代码缺陷数据的概率，从而确定其代码缺陷数据对应的复现概率。一般来说，与代码缺陷数据相对应的复现概率越大，则说明该代码缺陷数据对应的软件代码出现缺陷的概率越大，为了保证待测试版本 ID 对应的软件程序的质量，越需要尽早修复。

具体地，步骤 S603 具体包括如下步骤：（1）对代码缺陷数据对应的软件代码进行重复测试，获取每次重复测试对应的测试结果和重复测试次数。该测试结果包括测试成功和测试失败两种结果。该重复测试次数是服务器在代码缺陷分析过程中确定的重复测试的次数。（2）统计测试结果为测试失败的测试缺陷数量。（3）采用概率计算公式对重复测试次数和测试缺陷数量进行计算，获取缺陷概率，概率计算公式为  $L = U / Y$ ，其中， $L$  为缺陷概率， $U$  为测试缺陷数量， $Y$  为重复测试次数；（4）基于缺陷概率和预先设置的概率阈值，获取与代码缺陷数据相对应的复现概率。其中，概率阈值是服务器预先设置的用于评估不同复现概率的阈值。本实施例中，概率阈值包括 100% 和 50% 两个，也可以设置为其他数值。具体地，若代码缺陷数据的缺陷概率为 100%，则与代码缺陷数据相对应的复现概率为必然事件，可用 G0 来表示；若代码缺陷数据的缺陷概率在 50%-100% 之间，则与代码缺陷数据相对应的复现概率为高概率事件，可用 G1 来表示；若代码缺陷数据的缺陷概率在 50% 以下，则与代码缺陷数据相对应的复现概率为低概率事件，可用 G2 来表示。

S604：采用代码扫描工具对代码缺陷数据进行扫描分析，获取与代码缺陷数据相对应的可维护性。

其中，与代码缺陷数据相对应的可维护性是指基于代码缺陷数据对应的可维护指数与服务器预先设置的指数阈值进行分析所获取的评估指标。其中，可维护指数是指可维护指标 (Maintainability Index)，其范围是 0 到 100 之间的值，用来指示所有类、成员、命名空间或项目的可维护性。一般来说，代码缺陷数据对应的可维护性越小，其对应的软件代码越重复，越需要优先进行修复，以避免因代码缺陷数据对应的软件代码存在缺陷而影响整个待测试版本 ID 对应的软件程序的开发进度。

本实施例中，服务器预先配置有可维护指数公式，该可维护指数公式具体为可维护指数 = MAX (0,

$(171 - 5.2 * \ln(\text{霍尔斯特德量}) - 0.23 * (\text{圈复杂度}) - 16.2 * \ln(\text{代码行数})) * 100 / 171$ )。步骤 S304 具体包括如下内容：（1）采用代码扫描工具对代码缺陷数据进行扫描分析，获取与代码缺陷数据相对应的评估参数，评估参数包括霍尔斯特德量、圈复杂度和代码行数。霍尔斯特德量 (Halstead) 是测量代码计算复杂度。具体上，如果一个程序有  $N$  个操作数和运算符， $N$  个不同的操作数和运算符，那么  $\text{halstead} = N * \log_2(n)$ ，总之程序中的运算符和操作数越少越利于提高其霍尔斯特德量。圈复杂度 (Cyclomatic Complexity) 是代码的逻辑复杂度，程序的每个可能的执行分支 (if, while, for 等) 都为该指标贡献 1 个点。一般来说，圈复杂度的建议范围  $<10$ ，最多不超过 20。代码行数 (Lines of code) 是指待测试版本 ID 对应的软件程序的代码行数。例如，采用 Code Metrics 这一代码扫描工具对代码缺陷数据进行扫描分析，可快速获取与代码缺陷数据相对应的评估参数，如在 Code Metrics 的操作界面上选择 “Calculate Code Metrics”，即可在其结果窗口中获取对应一评估参数，这些评估参数包括霍尔斯特德量、圈复杂度和代码行数。（2）采用可维护指数公式对霍尔斯特德量、圈复杂度和代码行数进行计算，获取与代码缺陷数据相对应的可维护指数。（3）基于与代码缺陷数据相对应的可维护指数和预先设置的指数阈值，获取与代码缺陷数据相对应的可维护性。其中，其中，指数阈值是服务器预先设置的用于评估不同可维护性的阈值。具体地，服务器基于与代码缺陷数据相对应的可维护指数，根据该可维护指数确定其落入由至少两个指数阈值所界定的取值范围，根据该取值范围确定对应的可维护性。本实施例中，步骤 S402 计算出的可维护指数的取值范围为 0-100，若指数阈值为 10 和 20，则这两个指数阈值可将可维护指数划分成  $[0, 10]$ 、 $(10, 20]$  和  $(20, 100]$  三个取值范围，每个取值范围对应一可维护性，分别采用 F0、F1 和 F2 来表示。

可以理解地，通过采用代码扫描工具对代码缺陷数据进行扫描，以快速获取评估参数，再采用可维护指数公式对评估参数进行计算，可快速获取其对应的可维护指数，使得可维护指数的获取过程简单快速；然后，基于可维护指数和预设的指数阈值，可快速确定代码缺陷数据对应的可维护性，以便依据该可维护性对代码缺陷数据进行质量评估。

S605：基于待测试版本 ID 查询代码版本信息表，获取与代码缺陷数据相对应的稳定系数。

其中，代码版本信息表是存储在服务器中的用于记录待测试版本 ID 对应的软件程序对应的所有版本信息的数据表。稳定系数是用于评估待测试版本 ID 对应的软件程序是否稳定的指标。一般来说，全新的软件程序或者软件程序中的全新功能模块，其代码缺陷越容易发现，反之，软件程序的软件版本越高，其代码性能越稳定，其代码缺陷越难发现。具体地，服务器可通过待测试版本 ID 查询代码版本信息表，以获取与待测试版本 ID 对应的推出版本数量，根据该推出版本数量确定代码缺陷数据对应的稳定系数，以便基于该稳定系数对代码缺陷数据进行综合评估。

具体地，步骤 S604 具体包括如下步骤：（1）基于待测试版本 ID 查询代码版本信息表，获取与待测试版本 ID 对应的历史版本，根据历史版本确定推出版本数量。（2）基于推出版本数量查询稳定系数转换表，获取与代码缺陷数据相对应的稳定系数。服务器基于待测试版本 ID 查询代码版本信息表，可从该代码版本信息表中获取待测试版本 ID 对应的待分析版本的相关信息，也可以获取与待分析版本同源的至少一个历史版本的相关信息。然后，根据该待测试版本 ID 对应的至少一个历史版本的数量，确定推出版本数量。该推出版本数量可以具体限定为代码缺陷数据所在的历史版本对应的推出版本数量。其中，稳定系数转换表是用于存储推出版本数量和稳定系数之间的转换关系的数据表。具体地，服务器可基于推出版本数量查询稳定系数转换表，获取与该推出版本数量相对应的稳定系数，将该稳定系数确定为代码缺陷数据对应的稳定系数。可以理解地，待测试版本 ID 对应的软件程序中，全新功能模块中出现的代码缺陷比较多，其发现代码缺陷比较容易，因此，其稳定系数越低；相应地，若某一功能模块的版本越高，越稳定，其稳定系数越高。本实施例中，代码缺陷数据对应的稳定系数包括 W0、W1 和 W2 这三种，其对应的推出版本次数分别是 10 个以上、4-10 之间和 4 个以下。

S606：对同一代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数进行加权分析，获取与代码缺陷数据相对应的缺陷度量值。

其中，代码版本信息表是存储在服务器中的用于记录待测试版本 ID 对应的软件程序对应的所有版本信息的数据表。稳定系数是用于评估待测试版本 ID 对应的软件程序是否稳定的指标。一般来说，

全新的软件程序或者软件程序中的全新功能模块，其代码缺陷越容易发现，反之，软件程序的软件版本越高，其代码性能越稳定，其代码缺陷越难发现。具体地，服务器可通过待测试版本 ID 查询代码版本信息表，以获取与待测试版本 ID 对应的推出版本数量，根据该推出版本数量确定代码缺陷数据对应的稳定系数，以便基于该稳定系数对代码缺陷数据进行综合评估。

本实施例中，步骤 S606 中，即对同一代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数进行加权分析，获取与代码缺陷数据相对应的缺陷度量值，具体包括如下步骤：

(1) 基于同一代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数查询预先设置的权重分值数据表（如下表一所示），获取对应的评估分值和评估权重。其中，权重分值数据表是服务器中预先设置的用于记录每一评估因素对应的评估分值和评估权重的数据表。评估分值是预先设置每一种评估因素对应的分值，评估权重是预先设置的每一评估因素对应的权重。本实施例中，服务器基于同一代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数这六个评估因素查询预先设置的权重分值数据表，获取每一评估因素对应的评估分值和评估权重，以便后续基于该评估分值和评估权重进行综合计算其对应的缺陷度量值。

表一 权重分值数据表

| 评估因素   | 设置参数 | 说明                       | 评估分值 | 评估权重 |
|--------|------|--------------------------|------|------|
| 功能核心程度 | R    | 采用代码扫描工具分析代码缺陷数据的运行流行分支  |      | 30%  |
|        | R0   | 代码主干位置                   | 3    |      |
|        | R1   | 代码次主干位置                  | 2.1  |      |
|        | R2   | 代码次干位置                   | 1.5  |      |
| 页面使用频率 | V    | 埋点统计代码缺陷数据的点击量并计算页面使用频率  |      | 20%  |
|        | V0   | 高频                       | 2    |      |
|        | V1   | 一般                       | 1.4  |      |
|        | V2   | 低频                       | 1    |      |
| 页面访问步长 | D    | 埋点统计代码缺陷数据对应的页面功能的操作步骤长度 |      | 20%  |
|        | D0   | 1-2 步                    | 2    |      |
|        | D1   | 3-4 步                    | 1.4  |      |
|        | D2   | ≥5 步                     | 1    |      |
| 复现概率   | G    | 代码缺陷数据对应的软件代码重复测试出现缺陷的概率 |      | 10%  |
|        | G0   | 100% (必然事件)              | 1    |      |
|        | G1   | 50%-100% (高概率事件)         | 0.7  |      |
|        | G2   | 50%以下 (低概率事件)            | 0.5  |      |
| 可维护性   | F    | 采用代码扫描工具进行扫描并计算可维护性      |      | 10%  |
|        | F0   | 介于 0-10 之间               | 1    |      |
|        | F1   | 介于 10-20 之间              | 0.7  |      |
|        | F2   | 介于 20-100 之间             | 0.5  |      |
| 稳定系数   | W    | 根据推出版本数量确定稳定系数           |      | 10%  |
|        | W0   | 推出版本达到 10 个以上            | 1    |      |
|        | W1   | 推出版本介于 4-10 之间           | 0.7  |      |
|        | W2   | 推出版本小于 4 个               | 0.5  |      |

(2) 采用缺陷度量值公式对评估分值和评估权重进行加权计算, 获取代码缺陷数据对应的缺陷度量值; 其中, 缺陷度量值公式为  $Q = \sum S_i K_i$ ,  $Q$  为缺陷度量值,  $i$  为功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数或稳定系数,  $S_i$  为功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数或稳定系数对应的评估分值,  $K_i$  为功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数或稳定系数对应的评估权重。

具体地, 不同评估因素的评估权重可根据评估因素对软件程序的重复程度确定, 如表一所示, 可设置功能核心程度的评估权重为 30%, 页面使用频率和页面访问步长为 20%, 复现概率、可维护性和稳定系数为 10%, 则上述缺陷度量值公式具体为  $Q = R*30\% + V*20\% + D*20\% + G*10\% + F*10\% + W*10\%$ , 依据该缺陷度量值公式计算出的缺陷度量值的取值范围为 [1, 2], 可直观且客观反映代码缺陷数据的重要程度, 即

其取值越大，则代码缺陷数据越重复，越需要优先修复，以保证软件程序的开发进度。

本实施例所提供的回归测试用例确定方法中，依次获取该代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数等评估因素，通过对该评估因素进行加权计算，可快速获取对应的缺陷度量值，该缺陷度量值综合代码缺陷数据对应的软件代码的重要程度的多个评估因素，可客观地反映该软件代码的重要程度，以便基于该缺陷度量值合理安排缺陷修复工作，提高缺陷修复的效率。

在一实施例中，如图 7 所示，步骤 S205 中，采用层次分析法对代码改动热区和代码缺陷分布热区进行分析，获取目标测试用例，具体包括如下步骤：

S701：基于代码改动热区和代码缺陷分布热区构建层次结构模型，层次结构模型包括从上到下配置的目标层、准则层和方案层，方案层包含至少两个原始测试用例，准则层包括代码改动热区和代码缺陷分布热区。

具体地，基于代码改动热区和代码缺陷分布热区构建层次结构模型，该层次结构模型包括目标层 A、准则层 C 和方案层 P。该准则层 C 以代码改动热区和代码缺陷分布热区为准则层因素  $C_1$  和  $C_2$ ，进一步地，该代码改动热区是依据代码改动次数和代码改动行数进行综合获取的，该准则层 C 还可将代码改动次数和代码改动行数作为新的准则层因素  $C_3$  和  $C_4$ 。方案层 P 以软件版本测试过程中预先配置的至少一个原始测试用例为方案因素  $P_1$ 、 $P_2$ …… $P_n$ ，分别对应原始测试用例 1、原始测试用例 2……原始测试用例 Z。

S702：基于层次结构模型中的准则层因素的优先级，构建判断矩阵。

其中，判断矩阵是针对上一层次某一准则层因素而言，本层次中各准则层因素之间的相对重要性的数量表示，这是将定性判断转变为定量表示的一个过程。设准则层的因素  $C_k$  与方案层 P 中的因素  $P_1$ 、 $P_2$ …… $P_n$  有关，其构建判断矩阵 B 如下所示：

|       |           |           |    |           |
|-------|-----------|-----------|----|-----------|
| $C_k$ | $P_1$     | $P_2$     | …… | $P_n$     |
| $P_1$ | $B_{1,1}$ | $B_{1,2}$ | …… | $B_{1,n}$ |
| $P_2$ | $B_{2,1}$ | $B_{2,2}$ | …… | $B_{2,n}$ |
| ……    | ……        | ……        | …… | ……        |
| $P_n$ | $B_{n,1}$ | $B_{n,2}$ | …… | $B_{n,n}$ |

上述判断矩阵 B 中， $B_{i,j}$  通常取为 1, 3, 5, 7, 9 及其倒数，其含义为： $B_{i,j}=1$ ，表示  $P_i$  和  $P_j$  一样重要； $B_{i,j}=3$ ，表示  $P_i$  比  $P_j$  重要一点（稍微重要）； $B_{i,j}=5$ ，表示  $P_i$  比  $P_j$  明显重要； $B_{i,j}=7$ ，表示  $P_i$  比  $P_j$  重要得多； $B_{i,j}=9$ ，表示  $P_i$  比  $P_j$  极端重要。

S703：基于判断矩阵，获取下层因素相对上一层次的层次单排序；

其中，基于判断矩阵，获取下层因素相对上一层次的层次单排序是指根据判断矩阵，计算相对上一层次某因素而言，本层次与之有关的因素的重要性次序的权值。该层次单排序包括准则层因素相对于目标层的层次单排序，也包括方案层因素相对于准则层的层次单排序。层次单排序可归结为计算判断矩阵的特征根和特征向量的过程。具体地，对于判断矩阵 B，计算满足  $BW = \lambda_{\max} W$  的特征根和特征向量，W 的各个分量  $W_i$  为相应准则层因素的层次单排序的权值。

步骤 S703 具体包括如下步骤：（1）对判断矩阵中每一列进行标准化处理，获取列标准化矩阵具体地，服务器将判断矩阵 B 中的每一项除以该项所在列中的每一项的和，得到标准化矩阵，即

$\bar{B}_{i,j} = \frac{B_{i,j}}{\sum_{k=1}^n B_{k,j}}, i, j = 1, 2, \dots, n$ , 标准化处理的过程可保持矩阵的一致性。(2) 对列标准化矩阵的每一

行进行求和, 获取行权重, 即  $\bar{W}_i = \sum_{j=1}^n \bar{B}_{i,j}, j = 1, 2, \dots, n$ 。(3) 对列标准化矩阵的行权重进行标准化处

理, 获取判断矩阵的特征向量。具体地, 采用  $W_i = \frac{\bar{W}_i}{\sum_{j=1}^n \bar{W}_j}, i = 1, 2, \dots, n$  对列标准化矩阵中的行权重进

行标准化处理, 获取判断矩阵的特征值  $W_i$ 。(4) 判断矩阵的特征值  $W_i$ , 获取判断矩阵的最大矩阵特

征根  $\lambda_{\max}$ , 其中,  $\lambda_{\max} = \frac{1}{n} \sum_{i=1}^n \frac{(BW)_i}{W_i}$ 。

S704: 若层次单排序进行一致性校验通过, 则基于层次单排序获取层次总排序。

具体地, 采用一致性指标公式对判断矩阵的最大矩阵特征根  $\lambda_{\max}$  进行计算, 获取目标一致性指标, 基于目标一致性指标与平均随机一致性指标计算随机一致性比例, 若随机一致性比例小于或等于 0.1,

则该矩阵具有满意的一致性。一致性指标公式为  $CI = \frac{\lambda_{\max} - n}{n - 1}$ , 其中,  $\lambda_{\max}$  为判断矩阵的最大矩阵特

征根,  $RI$  是平均随机一致性指标, 可根据判断矩阵的阶数查表获知。根据  $CR = \frac{CI}{RI}$  计算随机一致性

比例, 其中,  $CR$  为随机一致性比例。具体地, 层次单排序进行一致性校验通过是指  $CR$  的取值小于或等于 0.1。可以理解地, 若层次单排序进行一致性校验通过, 则判断矩阵的层次单排序的结果为

$W[W_1, W_2, \dots, W_n]^T$ 。

具体地, 基于层次单排序获取层次总排序, 是指利用层次单排序, 综合计算方案层相对目标层重要性顺序的结合权值, 层次总排序从上到下进行。例如, 准则层因素  $C_1, C_2, \dots, C_m$  对目标层的层次单排序为  $c_1, c_2, \dots, c_m$ , 构建方案层元素  $P_1, P_2, \dots, P_n$  对  $c_1, c_2, \dots, c_m$  所形成的单列矩阵构成的矩阵, 即准则方案矩阵, 该准则方案矩阵具体为每一准则层因素  $C_1, C_2, \dots, C_m$  对方案层元素  $P_1, P_2, \dots, P_n$  所形成的矩阵, 即  $C_1$ - $P$  矩阵,  $C_2$ - $P$  矩阵  $\dots$   $C_m$ - $P$  矩阵。再根据准则方案矩阵, 综合计算方案层元素  $P_1, P_2, \dots, P_n$  对目标

层的层次总排序, 该层次总排序中每一方案层因素的方案权重为  $P_n = \sum_{i=1}^m c_i b_{n,i}$ 。

S705: 若层次总排序进行一致性校验通过, 则根据层次总排序确定方案层中的原始测试用例的方案权重, 确定目标测试用例。

具体地, 为了评价层次总排序的计算结果的一致性, 需要计算与层次单排序类似的一致性校验量,

在层次总排序中, 一致性指标公式为  $CI = \sum_{i=1}^m c_i CI_i$ ,  $RI = \sum_{i=1}^m c_i RI_i$ ,  $CR = \frac{CI}{RI}$ , 当  $CR \leq 0.1$  时,

认定层次总排序具有满意的一致性, 其结果可供决策者参考。此时, 可根据层次总排序确定方案层中的原始测试用例的方案权重, 确定目标测试用例。

本实施例所提供的回归测试用例确定方法中,根据回归测试决策过程中,无法兼顾测试效率和测试质量的问题,采用层次分析法对代码改动热区和代码缺陷分布热区进行判断矩阵构建,并通过分析计算获取层次单排序和层次总排序,在一致性校验通过时,基于层次总排序确定目标测试用例,以使其确定的目标测试用例可有效保障测试质量,可有效避免遗漏代码改动热区和代码缺陷分布热区对应的代码进行测试的测试用例;并且,根据目标测试用例进行测试,可避免采用所有原始测试用例进行全面测试所存在的效率不足的问题。

应理解,上述实施例中各步骤的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不应对本申请实施例的实施过程构成任何限定。

在一实施例中,提供一种回归测试用例确定装置,该回归测试用例确定装置与上述实施例中回归测试用例确定方法一一对应。如图8所示,该回归测试用例确定装置包括测试分析请求获取模块801、待测试版本代码获取模块802、代码改动热区获取模块803、代码缺陷分布热区获取模块804和目标测试用例获取模块805。各功能模块详细说明如下:

测试分析请求获取模块801,用于获取测试分析请求,测试分析请求包括待测试版本ID。

待测试版本代码获取模块802,用于基于待测试版本ID,从代码仓库中获取与待测试版本ID相对应的待测试版本代码。

代码改动热区获取模块803,用于采用版本控制工具对待测试版本代码进行代码分析,获取代码改动热区。

代码缺陷分布热区获取模块804,用于对代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区。

目标测试用例获取模块805,用于采用层次分析法对代码改动热区和代码缺陷分布热区进行分析,获取目标测试用例。

优选地,代码改动热区获取模块803包括历史版本数据获取子模块和代码改动热区获取子模块。

历史版本数据获取子模块,用于采用版本控制工具执行日志查询命令,获取与待测试版本代码相对应的历史版本测试数据。

代码改动热区获取子模块,用于采用版本控制工具对待测试版本代码和历史版本测试数据进行分析,获取代码改动热区。

优选地,代码改动热区获取子模块包括代码改动次数获取单元、代码改动行数获取单元、综合评估指标获取单元和代码改动热区获取单元。

代码改动次数获取单元,用于采用Code Maat工具对待测试版本代码和历史版本测试数据进行分析,获取待测试版本代码中每一代码模块对应的代码改动次数。

代码改动行数获取单元,用于采用Cloc工具对待测试版本代码和历史版本测试数据进行分析,获取待测试版本代码中每一代码模块对应的代码改动行数。

综合评估指标获取单元,用于对每一代码模块对应的代码改动次数和代码改动行数进行标准化和加权处理,获取每一代码模块对应的综合评估指标。

代码改动热区获取单元,用于基于每一代码模块对应的综合评估指标,获取代码改动热区。

优选地,在代码缺陷分布热区获取模块804之前,回归测试用例确定装置还包括:

代码缺陷数据获取模块,用于基于待测试版本ID查询测试结果数据库,获取与待测试版本ID相对应的至少一个代码缺陷数据。

缺陷度量值获取模块,用于对代码缺陷数据进行缺陷度量分析,获取代码缺陷数据的缺陷度量值。

缺陷优先级获取模块,用于依据代码缺陷数据的缺陷度量值,获取代码缺陷数据对应的缺陷优先级,并配置每一代码缺陷数据对应的正则表达式。

代码缺陷分布热区获取模块804,还用于依据代码缺陷数据对应的缺陷优先级,采用代码缺陷数据对应的正则表达式对代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区。

优选地,对缺陷度量值获取模块包括功能核心程度获取子模块、页面频率步长获取子模块、复现概率获取子模块、可维护性获取子模块、稳定系数获取子模块和缺陷度量值获取子模块

功能核心程度获取子模块,用于基于与待测试版本ID相对应的抽象语法树对代码缺陷数据进行语



法树分析, 获取与代码缺陷数据相对应的功能核心程度。

页面频率步长获取子模块, 用于基于代码缺陷数据查询与待测试版本 ID 相对应的埋点数据库, 获取与代码缺陷数据相对应的页面使用频率和页面访问步长。

复现概率获取子模块, 用于对代码缺陷数据进行重复测试, 获取与代码缺陷数据相对应的复现概率。

可维护性获取子模块, 用于采用代码扫描工具对代码缺陷数据进行扫描分析, 获取与代码缺陷数据相对应的可维护性。

稳定系数获取子模块, 用于基于待测试版本 ID 查询代码版本信息表, 获取与代码缺陷数据相对应的稳定系数。

缺陷度量值获取子模块, 用于对同一代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数进行加权分析, 获取与代码缺陷数据相对应的缺陷度量值。

优选地, 缺陷度量值获取子模块包括评估分值权重获取单元和缺陷度量值获取单元。

评估分值权重获取单元, 用于基于同一代码缺陷数据对应的功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数和稳定系数查询预先设置的权重分值数据表, 获取对应的评估分值和评估权重。

缺陷度量值获取单元, 用于采用缺陷度量值公式对评估分值和评估权重进行加权计算, 获取代码缺陷数据对应的缺陷度量值。

其中, 缺陷度量值公式为  $Q = \sum S_i K_i$ ,  $Q$  为缺陷度量值,  $i$  为功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数或稳定系数,  $S_i$  为功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数或稳定系数对应的评估分值,  $K_i$  为功能核心程度、页面使用频率、页面访问步长、复现概率、可维护指数或稳定系数对应的评估权重。

优选地, 目标测试用例获取模块 805 包括层次结构模型构建单元、判断矩阵构建单元、层次单排序获取单元和目标测试用例获取单元。

层次结构模型构建单元, 用于基于代码改动热区和代码缺陷分布热区构建层次结构模型, 层次结构模型包括从上到下配置的目标层、准则层和方案层, 方案层包含至少两个原始测试用例, 准则层包括代码改动热区和代码缺陷分布热区。

判断矩阵构建单元, 用于基于层次结构模型中的准则层因素的优先级, 构建判断矩阵。

层次单排序获取单元, 用于基于判断矩阵, 获取下层因素相对上一层次的层次单排序。

层次总排序获取单元, 用于若层次单排序进行一致性校验通过, 则基于层次单排序获取层次总排序。

目标测试用例获取单元, 用于若层次总排序进行一致性校验通过, 则根据层次总排序确定方案层中的原始测试用例的方案权重, 确定目标测试用例。

关于回归测试用例确定装置的具体限定可以参见上文中对于回归测试用例确定方法的限定, 在此不再赘述。上述回归测试用例确定装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中, 也可以以软件形式存储于计算机设备中的存储器中, 以便于处理器调用执行以上各个模块对应的操作。

在一个实施例中, 提供了一种计算机设备, 该计算机设备可以是服务器, 其内部结构图可以如图 9 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、网络接口和数据库。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。该非易失性存储介质存储有操作系统、计算机可读指令和数据库。该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。该计算机设备的数据库用于存储执行回归测试用例确定方法过程中采用或者获取的数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机可读指令被处理器执行时以实现一种回归测试用例确定方法。

在一个实施例中, 提供了一种计算机设备, 包括存储器、处理器及存储在存储器上并可在处理器上

运行的计算机可读指令，处理器执行计算机可读指令时实现上述实施例中回归测试用例确定方法的步骤，例如图 2 所示的步骤 S201-S205，或者图 3 至图 7 中所示的步骤，为避免重复，这里不再赘述。或者，处理器执行计算机可读指令时实现回归测试用例确定装置这一实施例中的各模块/单元的功能，例如图 8 所示的测试分析请求获取模块 801、待测试版本代码获取模块 802、代码改动热区获取模块 803、代码缺陷分布热区获取模块 804 和目标测试用例获取模块 805 的功能，为避免重复，这里不再赘述。

在一实施例中，提供一个或多个存储有计算机可读指令的可读存储介质，所述计算机可读存储介质存储有计算机可读指令，所述计算机可读指令被一个或多个处理器执行时，使得所述一个或多个处理器执行时实现上述实施例中回归测试用例确定方法的步骤，例如图 2 所示的步骤 S201-S205，或者图 3 至图 7 中所示的步骤，为避免重复，这里不再赘述。或者，该计算机可读指令被处理器执行时实现上述回归测试用例确定装置这一实施例中的各模块/单元的功能，例如图 8 所示的测试分析请求获取模块 801、待测试版本代码获取模块 802、代码改动热区获取模块 803、代码缺陷分布热区获取模块 804 和目标测试用例获取模块 805 的功能，为避免重复，这里不再赘述。本实施例中的可读存储介质包括非易失性可读存储介质和易失性可读存储介质。

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，该计算机可读指令可存储于一非易失性可读存储介质也可以存储在易失性可读存储介质中，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，本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器 (ROM)、可编程 ROM (PROM)、电可编程 ROM (EPROM)、电可擦除可编程 ROM (EEPROM) 或闪存。易失性存储器可包括随机存取存储器 (RAM) 或者外部高速缓冲存储器。作为说明而非局限，RAM 以多种形式可得，诸如静态 RAM (SRAM)、动态 RAM (DRAM)、同步 DRAM (SDRAM)、双数据率 SDRAM (DDRSDRAM)、增强型 SDRAM (ESDRAM)、同步链路 (Synchlink) DRAM (SLDRAM)、存储器总线 (Rambus) 直接 RAM (RDRAM)、直接存储器总线动态 RAM (DRDRAM)、以及存储器总线动态 RAM (RDRAM) 等。

所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，仅以上述各功能单元、模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能单元、模块完成，即将所述装置的内部结构划分成不同的功能单元或模块，以完成以上描述的全部或者部分功能。

以上所述实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围，均应包含在本申请的保护范围之内。

## 权 利 要 求 书

1. 一种回归测试用例确定方法，其特征在于，包括：

获取测试分析请求，所述测试分析请求包括待测试版本 ID；

基于所述待测试版本 ID，从代码仓库中获取与所述待测试版本 ID 相对应的待测试版本代码；

采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区；

对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区；

采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例。

2. 如权利要求 1 所述的回归测试用例确定方法，其特征在于，所述采用版本控制工具对所述待测试版本代码进行代码分析，获取代码改动热区，包括：

采用版本控制工具执行日志查询命令，获取与所述待测试版本代码相对应的历史版本测试数据；

采用版本控制工具对所述待测试版本代码和所述历史版本测试数据进行分析，获取代码改动热区。

3. 如权利要求 2 所述的回归测试用例确定方法，其特征在于，所述采用版本控制工具对所述待测试版本代码和历史版本测试数据进行分析，获取代码改动热区，包括：

采用 Code Maat 工具对所述待测试版本代码和历史版本测试数据进行分析，获取所述待测试版本代码中每一代码模块对应的代码改动次数；

采用 Cloc 工具对所述待测试版本代码和历史版本测试数据进行分析，获取所述待测试版本代码中每一代码模块对应的代码改动行数；

对每一代码模块对应的代码改动次数和代码改动行数进行标准化和加权处理，获取每一所述代码模块对应的综合评估指标；

基于每一所述代码模块对应的综合评估指标，获取代码改动热区。

4. 如权利要求 1 所述的回归测试用例确定方法，其特征在于，在所述对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区的步骤之前，所述回归测试用例确定方法还包括：

基于所述待测试版本 ID 查询测试结果数据库，获取与所述待测试版本 ID 相对应的至少一个代码缺陷数据；

对所述代码缺陷数据进行缺陷度量分析，获取所述代码缺陷数据的缺陷度量值；

依据所述代码缺陷数据的缺陷度量值，获取所述代码缺陷数据对应的缺陷优先级，并配置每一所述代码缺陷数据对应的正则表达式；

所述对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区，包括：依据所述代码缺陷数据对应的缺陷优先级，采用所述代码缺陷数据对应的正则表达式对所述代码改动热区进行代码缺陷过滤，获取代码缺陷分布热区。

5. 如权利要求 4 所述的回归测试用例确定方法，其特征在于，对所述代码缺陷数据进行缺陷度量分析，获取所述代码缺陷数据的缺陷度量值，包括：

基于与所述待测试版本 ID 相对应的抽象语法树对所述代码缺陷数据进行语法树分析，获取与所述代码缺陷数据相对应的功能核心程度；

基于所述代码缺陷数据查询与所述待测试版本 ID 相对应的埋点数据库，获取与所述代码缺陷数据相对应的页面使用频率和页面访问步长；

对所述代码缺陷数据进行重复测试，获取与所述代码缺陷数据相对应的复现概率；

采用代码扫描工具对所述代码缺陷数据进行扫描分析，获取与所述代码缺陷数据相对应的可维护性；

基于所述待测试版本 ID 查询代码版本信息表，获取与所述代码缺陷数据相对应的稳定系数；

对同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数进行加权分析，获取与所述代码缺陷数据相对应的缺陷度量值。

6. 如权利要求 5 所述的回归测试用例确定方法，其特征在于，所述对同一所述代码缺陷数据对应

的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数进行加权分析,获取与所述代码缺陷数据相对应的缺陷度量值,包括:

基于同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数查询预先设置的权重分值数据表,获取对应的评估分值和评估权重;

采用缺陷度量值公式对所述评估分值和所述评估权重进行加权计算,获取所述代码缺陷数据对应的缺陷度量值;

其中,所述缺陷度量值公式为 $Q = \sum S_i K_i$ ,  $Q$ 为缺陷度量值, $i$ 为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数, $S_i$ 为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数对应的评估分值, $K_i$ 为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数对应的评估权重。

7. 如权利要求1所述的回归测试用例确定方法,其特征在于,所述采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析,获取目标测试用例,包括:

基于所述代码改动热区和代码缺陷分布热区构建层次结构模型,所述层次结构模型包括从上到下配置的目标层、准则层和方案层,所述方案层包含至少两个原始测试用例,所述准则层包括所述代码改动热区和所述代码缺陷分布热区;

基于所述层次结构模型中的准则层因素的优先级,构建判断矩阵;

基于所述判断矩阵,获取下层因素相对上一层次的层次单排序;

若所述层次单排序进行一致性校验通过,则基于所述层次单排序获取层次总排序;

若层次总排序进行一致性校验通过,则根据层次总排序确定方案层中的原始测试用例的方案权重,确定目标测试用例。

8. 一种回归测试用例确定装置,其特征在于,包括:

测试分析请求获取模块,用于获取测试分析请求,所述测试分析请求包括待测试版本ID;

待测试版本代码获取模块,用于基于所述待测试版本ID,从代码仓库中获取与所述待测试版本ID相对应的待测试版本代码;

代码改动热区获取模块,用于采用版本控制工具对所述待测试版本代码进行代码分析,获取代码改动热区;

代码缺陷分布热区获取模块,用于对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区;

目标测试用例获取模块,用于采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析,获取目标测试用例。

9. 一种计算机设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机可读指令,其特征在于,所述处理器执行所述计算机可读指令时实现如下步骤:

获取测试分析请求,所述测试分析请求包括待测试版本ID;

基于所述待测试版本ID,从代码仓库中获取与所述待测试版本ID相对应的待测试版本代码;

采用版本控制工具对所述待测试版本代码进行代码分析,获取代码改动热区;

对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区;

采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析,获取目标测试用例。

10. 如权利要求9所述的计算机设备,其特征在于,所述采用版本控制工具对所述待测试版本代码进行代码分析,获取代码改动热区,包括:

采用版本控制工具执行日志查询命令,获取与所述待测试版本代码相对应的历史版本测试数据;

采用版本控制工具对所述待测试版本代码和所述历史版本测试数据进行分析,获取代码改动热区。

11. 如权利要求 9 所述的计算机设备,其特征在于,在所述对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区的步骤之前,所述处理器执行所述计算机可读指令时还实现如下步骤:

基于所述待测试版本 ID 查询测试结果数据库,获取与所述待测试版本 ID 相对应的至少一个代码缺陷数据;

对所述代码缺陷数据进行缺陷度量分析,获取所述代码缺陷数据的缺陷度量值;

依据所述代码缺陷数据的缺陷度量值,获取所述代码缺陷数据对应的缺陷优先级,并配置每一所述代码缺陷数据对应的正则表达式;

所述对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区,包括:依据所述代码缺陷数据对应的缺陷优先级,采用所述代码缺陷数据对应的正则表达式对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区。

12. 如权利要求 11 所述的计算机设备,其特征在于,对所述代码缺陷数据进行缺陷度量分析,获取所述代码缺陷数据的缺陷度量值,包括:

基于与所述待测试版本 ID 相对应的抽象语法树对所述代码缺陷数据进行语法树分析,获取与所述代码缺陷数据相对应的功能核心程度;

基于所述代码缺陷数据查询与所述待测试版本 ID 相对应的埋点数据库,获取与所述代码缺陷数据相对应的页面使用频率和页面访问步长;

对所述代码缺陷数据进行重复测试,获取与所述代码缺陷数据相对应的复现概率;

采用代码扫描工具对所述代码缺陷数据进行扫描分析,获取与所述代码缺陷数据相对应的可维护性;

基于所述待测试版本 ID 查询代码版本信息表,获取与所述代码缺陷数据相对应的稳定系数;

对同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数进行加权分析,获取与所述代码缺陷数据相对应的缺陷度量值。

13. 如权利要求 12 所述的计算机设备,其特征在于,所述对同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数进行加权分析,获取与所述代码缺陷数据相对应的缺陷度量值,包括:

基于同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数查询预先设置的权重分值数据表,获取对应的评估分值和评估权重;

采用缺陷度量值公式对所述评估分值和所述评估权重进行加权计算,获取所述代码缺陷数据对应的缺陷度量值;

其中,所述缺陷度量值公式为  $Q = \sum S_i K_i$ ,  $Q$  为缺陷度量值,  $i$  为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数,  $S_i$  为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数对应的评估分值,  $K_i$  为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数对应的评估权重。

14. 如权利要求 9 所述的计算机设备,其特征在于,所述采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析,获取目标测试用例,包括:

基于所述代码改动热区和代码缺陷分布热区构建层次结构模型,所述层次结构模型包括从上到下配置的目标层、准则层和方案层,所述方案层包含至少两个原始测试用例,所述准则层包括所述代码改动热区和所述代码缺陷分布热区;

基于所述层次结构模型中的准则层因素的优先级,构建判断矩阵;

基于所述判断矩阵,获取下层因素相对上一层次的层次单排序;

若所述层次单排序进行一致性校验通过,则基于所述层次单排序获取层次总排序;

若层次总排序进行一致性校验通过,则根据层次总排序确定方案层中的原始测试用例的方案权重,确定目标测试用例。

15. 一个或多个存储有计算机可读指令的可读存储介质,所述计算机可读存储介质存储有计算机可读指令,其特征在于,所述计算机可读指令被一个或多个处理器执行时,使得所述一个或多个处理器执行如下步骤:

获取测试分析请求,所述测试分析请求包括待测试版本 ID;

基于所述待测试版本 ID,从代码仓库中获取与所述待测试版本 ID 相对应的待测试版本代码;

采用版本控制工具对所述待测试版本代码进行代码分析,获取代码改动热区;

对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区;

采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析,获取目标测试用例。

16. 如权利要求 15 所述的可读存储介质,其特征在于,所述采用版本控制工具对所述待测试版本代码进行代码分析,获取代码改动热区,包括:

采用版本控制工具执行日志查询命令,获取与所述待测试版本代码相对应的历史版本测试数据;

采用版本控制工具对所述待测试版本代码和所述历史版本测试数据进行分析,获取代码改动热区。

17. 如权利要求 15 所述的可读存储介质,其特征在于,在所述对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区的步骤之前,所述计算机可读指令被一个或多个处理器执行时,使得所述一个或多个处理器执行如下步骤:

基于所述待测试版本 ID 查询测试结果数据库,获取与所述待测试版本 ID 相对应的至少一个代码缺陷数据;

对所述代码缺陷数据进行缺陷度量分析,获取所述代码缺陷数据的缺陷度量值;

依据所述代码缺陷数据的缺陷度量值,获取所述代码缺陷数据对应的缺陷优先级,并配置每一所述代码缺陷数据对应的正则表达式;

所述对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区,包括:依据所述代码缺陷数据对应的缺陷优先级,采用所述代码缺陷数据对应的正则表达式对所述代码改动热区进行代码缺陷过滤,获取代码缺陷分布热区。

18. 如权利要求 17 所述的可读存储介质,其特征在于,对所述代码缺陷数据进行缺陷度量分析,获取所述代码缺陷数据的缺陷度量值,包括:

基于与所述待测试版本 ID 相对应的抽象语法树对所述代码缺陷数据进行语法树分析,获取与所述代码缺陷数据相对应的功能核心程度;

基于所述代码缺陷数据查询与所述待测试版本 ID 相对应的埋点数据库,获取与所述代码缺陷数据相对应的页面使用频率和页面访问步长;

对所述代码缺陷数据进行重复测试,获取与所述代码缺陷数据相对应的复现概率;

采用代码扫描工具对所述代码缺陷数据进行扫描分析,获取与所述代码缺陷数据相对应的可维护性;

基于所述待测试版本 ID 查询代码版本信息表,获取与所述代码缺陷数据相对应的稳定系数;

对同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数进行加权分析,获取与所述代码缺陷数据相对应的缺陷度量值。

19. 如权利要求 18 所述的可读存储介质,其特征在于,所述对同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数进行加权分析,获取与所述代码缺陷数据相对应的缺陷度量值,包括:

基于同一所述代码缺陷数据对应的所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数和所述稳定系数查询预先设置的权重分值数据表,获取对应的评估分值和

评估权重；

采用缺陷度量值公式对所述评估分值和所述评估权重进行加权计算，获取所述代码缺陷数据对应的缺陷度量值；

其中，所述缺陷度量值公式为  $Q = \sum S_i K_i$ ， $Q$  为缺陷度量值， $i$  为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数， $S_i$  为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数对应的评估分值， $K_i$  为所述功能核心程度、所述页面使用频率、所述页面访问步长、所述复现概率、所述可维护指数或所述稳定系数对应的评估权重。

20. 如权利要求 15 所述的可读存储介质，其特征在于，所述采用层次分析法对所述代码改动热区和所述代码缺陷分布热区进行分析，获取目标测试用例，包括：

基于所述代码改动热区和代码缺陷分布热区构建层次结构模型，所述层次结构模型包括从上到下配置的目标层、准则层和方案层，所述方案层包含至少两个原始测试用例，所述准则层包括所述代码改动热区和所述代码缺陷分布热区；

基于所述层次结构模型中的准则层因素的优先级，构建判断矩阵；

基于所述判断矩阵，获取下层因素相对上一层次的层次单排序；

若所述层次单排序进行一致性校验通过，则基于所述层次单排序获取层次总排序；

若层次总排序进行一致性校验通过，则根据层次总排序确定方案层中的原始测试用例的方案权重，确定目标测试用例。

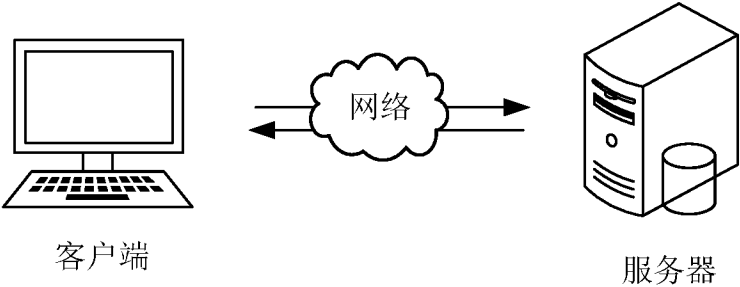


图 1

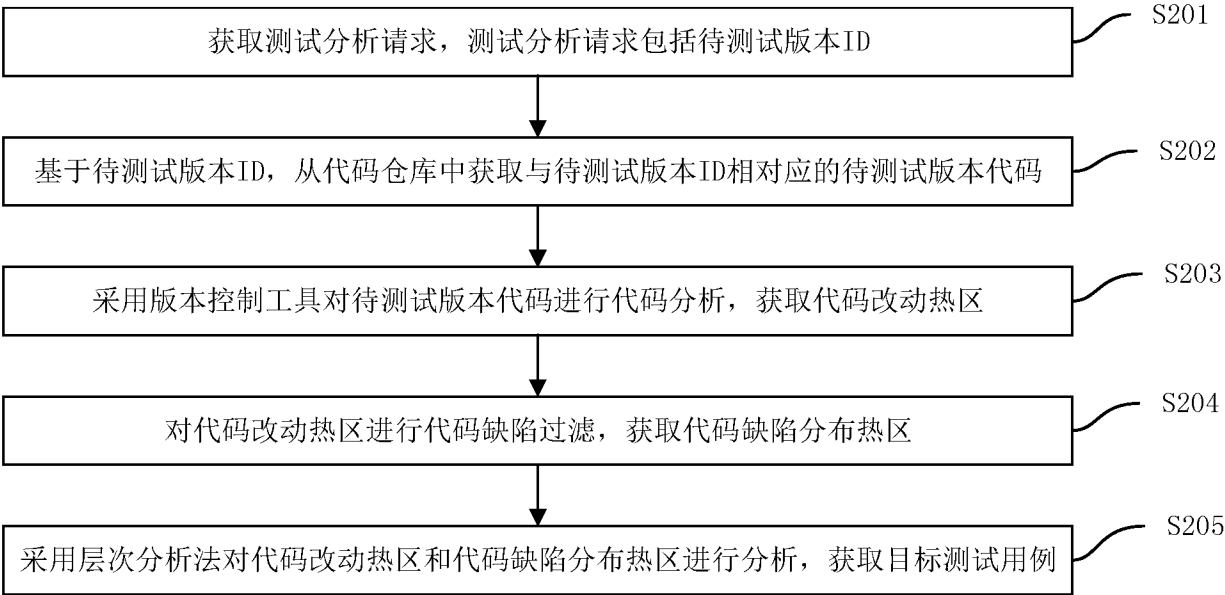


图 2

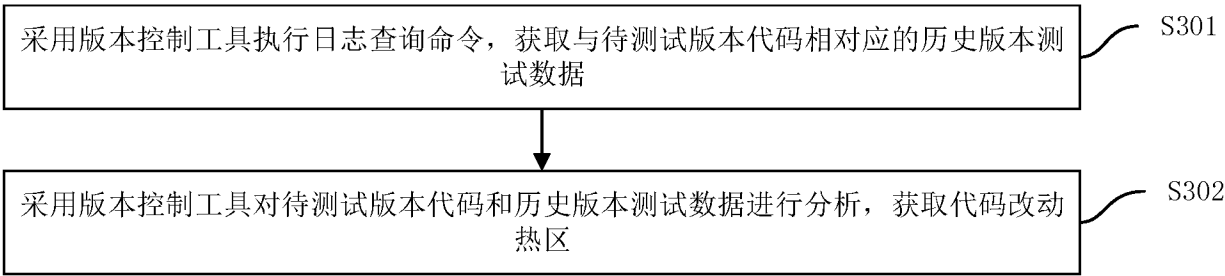


图 3



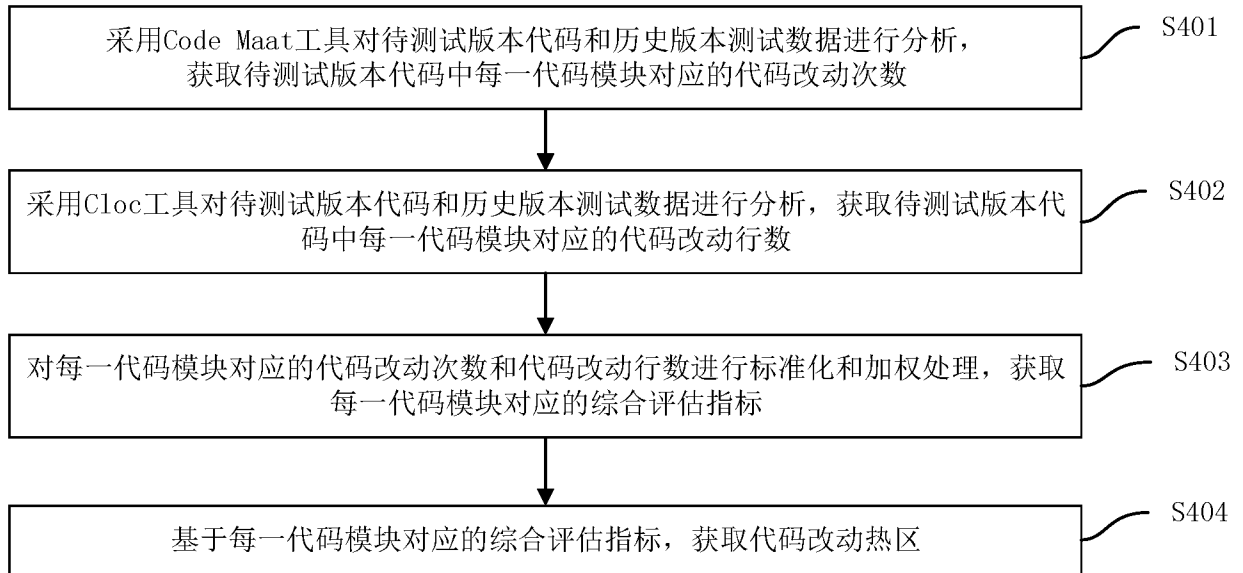


图 4

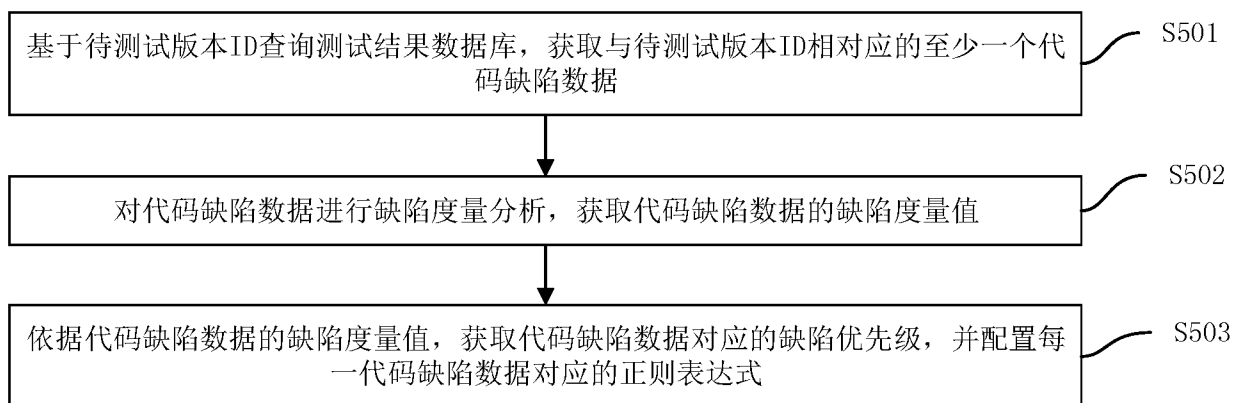


图 5

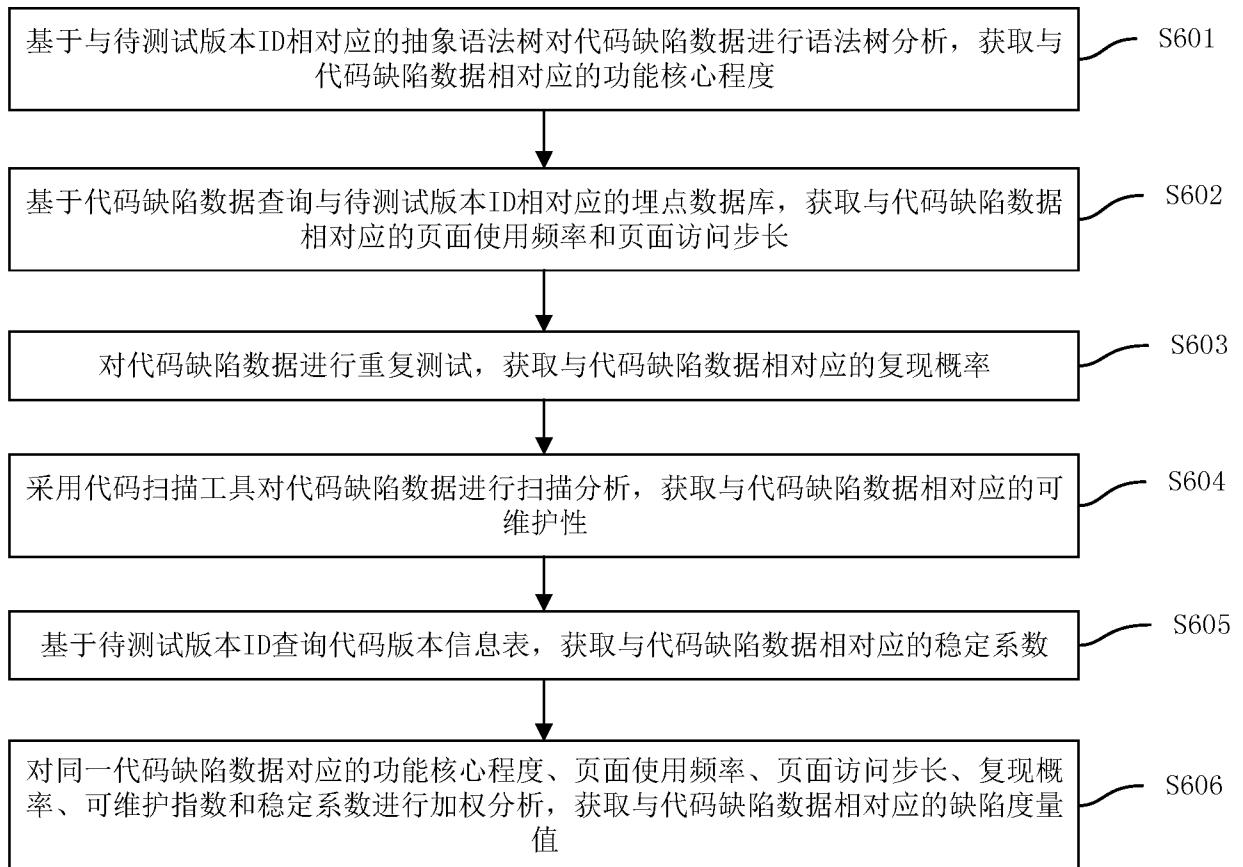


图 6

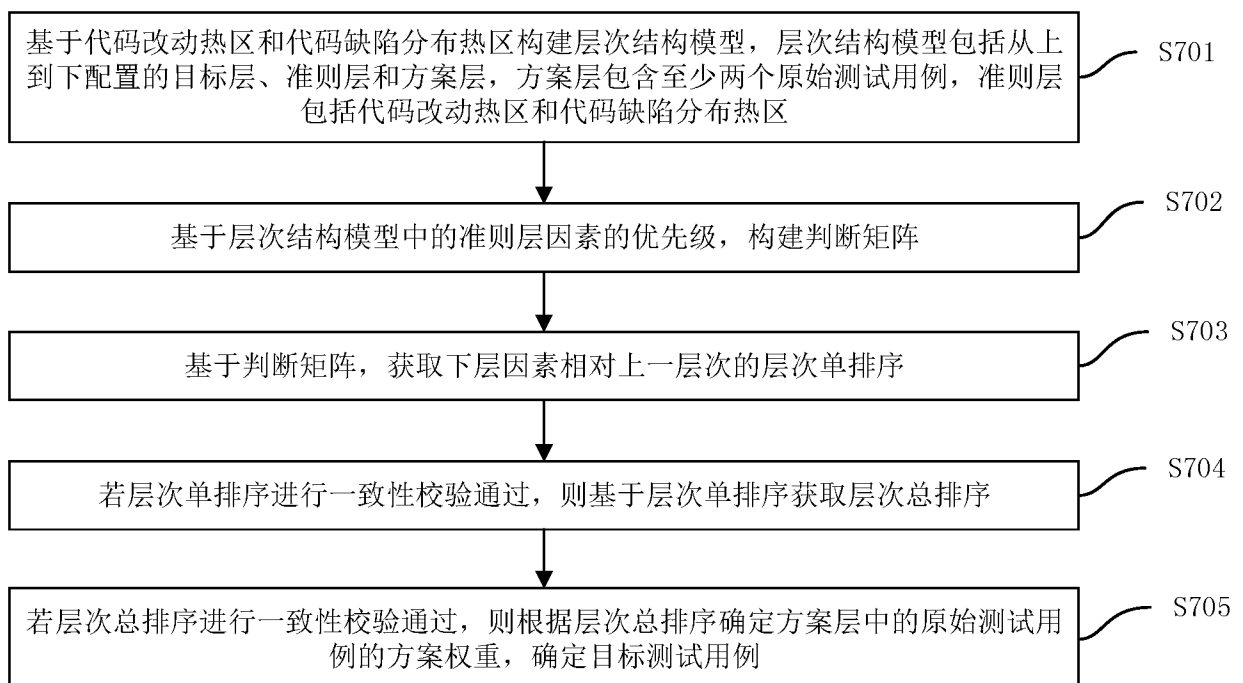


图 7

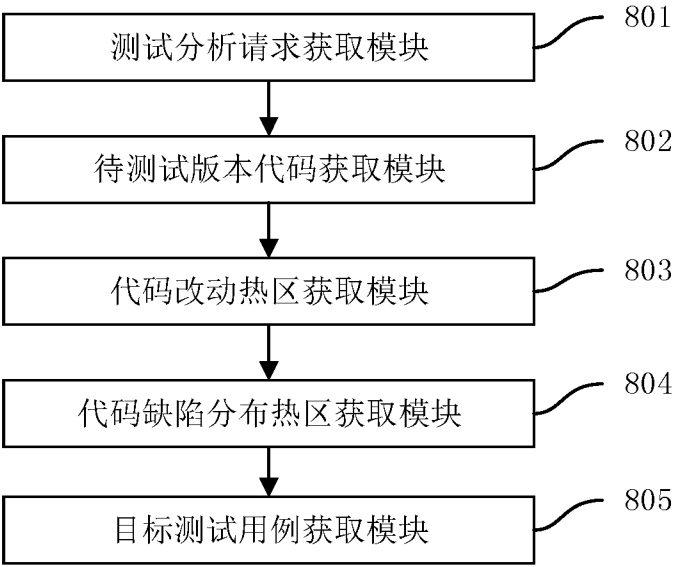


图 8

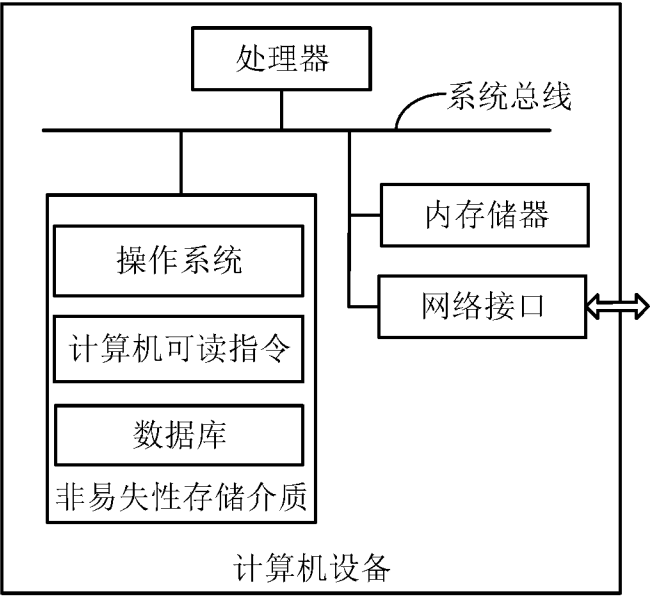


图 9

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/120596

**A. CLASSIFICATION OF SUBJECT MATTER**

G06F 11/36(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, CNKI, GOOGLE: 回归, 测试, 版本, 改动, 缺陷, 分析, ID, regression, test, version, change, defect, analysis

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                      | Relevant to claim No. |
|-----------|-------------------------------------------------------------------------------------------------------------------------|-----------------------|
| PX        | CN 110109820 A (SHENZHEN ONECONNECT TECHNOLOGY CO., LTD.) 09 August 2019 (2019-08-09)<br>description, paragraphs 30-163 | 1-20                  |
| X         | CN 101329652 B (ZTE CORPORATION) 08 December 2010 (2010-12-08)<br>description, paragraphs 7-17                          | 1-20                  |
| A         | CN 102184137 A (INSPUR GROUP SHANDONG GENERSOFT CO., LTD.) 14 September 2011 (2011-09-14)<br>entire document            | 1-20                  |
| A         | US 2017139685 A1 (CHENGDU PUZHONG SOFTWARE LIMITED COMPANY) 18 May 2017 (2017-05-18)<br>entire document                 | 1-20                  |



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

24 February 2020

Date of mailing of the international search report

27 February 2020

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/  
CN)  
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing  
100088  
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

**INTERNATIONAL SEARCH REPORT**  
**Information on patent family members**

International application No.

**PCT/CN2019/120596**

| Patent document<br>cited in search report |            |    | Publication date<br>(day/month/year) | Patent family member(s) |            |    | Publication date<br>(day/month/year) |
|-------------------------------------------|------------|----|--------------------------------------|-------------------------|------------|----|--------------------------------------|
| CN                                        | 110109820  | A  | 09 August 2019                       | None                    |            |    |                                      |
| CN                                        | 101329652  | B  | 08 December 2010                     | None                    |            |    |                                      |
| CN                                        | 102184137  | A  | 14 September 2011                    | None                    |            |    |                                      |
| US                                        | 2017139685 | A1 | 18 May 2017                          | CA                      | 2952093    | A1 | 13 December 2016                     |
|                                           |            |    |                                      | EP                      | 3163436    | A1 | 03 May 2017                          |
|                                           |            |    |                                      | CN                      | 105320504  | A  | 10 February 2016                     |
|                                           |            |    |                                      | WO                      | 2015196784 | A1 | 30 December 2015                     |

## 国际检索报告

国际申请号

PCT/CN2019/120596

|                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                    |                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|--------------------------------------|
| <b>A. 主题的分类</b><br>G06F 11/36 (2006.01) i<br><br>按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类                                                                                                                                                                                                                                                                                                              |                                                                                                    |                                      |
| <b>B. 检索领域</b><br>检索的最低限度文献 (标明分类系统和分类号)<br>G06F<br><br>包含在检索领域中的除最低限度文献以外的检索文献<br><br>在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))<br>CNPAT, WPI, EPDOC, CNKI, GOOGLE: 回归, 测试, 版本, 改动, 缺陷, 分析, ID, regression, test, version, change, defect, analysis                                                                                                                                                 |                                                                                                    |                                      |
| <b>C. 相关文件</b>                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                    |                                      |
| 类 型*                                                                                                                                                                                                                                                                                                                                                                                              | 引用文件, 必要时, 指明相关段落                                                                                  | 相关的权利要求                              |
| PX                                                                                                                                                                                                                                                                                                                                                                                                | CN 110109820 A (深圳壹账通智能科技有限公司) 2019年 8月 9日 (2019 - 08 - 09)<br>说明书30-163段                          | 1-20                                 |
| X                                                                                                                                                                                                                                                                                                                                                                                                 | CN 101329652 B (中兴通讯股份有限公司) 2010年 12月 8日 (2010 - 12 - 08)<br>说明书7-17段                              | 1-20                                 |
| A                                                                                                                                                                                                                                                                                                                                                                                                 | CN 102184137 A (浪潮集团山东通用软件有限公司) 2011年 9月 14日 (2011 - 09 - 14)<br>全文                                | 1-20                                 |
| A                                                                                                                                                                                                                                                                                                                                                                                                 | US 2017139685 A1 (CHENGDU PUZHONG SOFTWARE LIMITED COMPANY) 2017年 5月 18日<br>(2017 - 05 - 18)<br>全文 | 1-20                                 |
| <input type="checkbox"/> 其余文件在C栏的续页中列出。 <input checked="" type="checkbox"/> 见同族专利附件。                                                                                                                                                                                                                                                                                                              |                                                                                                    |                                      |
| * 引用文件的具体类型:<br>“A” 认为不特别相关的表示了现有技术一般状态的文件<br>“E” 在国际申请日的当天或之后公布的在先申请或专利<br>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)<br>“O” 涉及口头公开、使用、展览或其他方式公开的文件<br>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件<br>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件<br>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性<br>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性<br>“&” 同族专利的文件 |                                                                                                    |                                      |
| 国际检索实际完成的日期<br>2020年 2月 24日                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                    | 国际检索报告邮寄日期<br>2020年 2月 27日           |
| ISA/CN的名称和邮寄地址<br>中国国家知识产权局 (ISA/CN)<br>中国北京市海淀区蓟门桥西土城路6号 100088<br>传真号 (86-10)62019451                                                                                                                                                                                                                                                                                                           |                                                                                                    | 授权官员<br>赵天奇<br>电话号码 86-(10)-53961320 |

国际检索报告  
关于同族专利的信息

国际申请号

PCT/CN2019/120596

| 检索报告引用的专利文件 |            |    | 公布日<br>(年/月/日) | 同族专利 |            |    | 公布日<br>(年/月/日) |
|-------------|------------|----|----------------|------|------------|----|----------------|
| CN          | 110109820  | A  | 2019年 8月 9日    | 无    |            |    |                |
| CN          | 101329652  | B  | 2010年 12月 8日   | 无    |            |    |                |
| CN          | 102184137  | A  | 2011年 9月 14日   | 无    |            |    |                |
| US          | 2017139685 | A1 | 2017年 5月 18日   | CA   | 2952093    | A1 | 2016年 12月 13日  |
|             |            |    |                | EP   | 3163436    | A1 | 2017年 5月 3日    |
|             |            |    |                | CN   | 105320504  | A  | 2016年 2月 10日   |
|             |            |    |                | WO   | 2015196784 | A1 | 2015年 12月 30日  |