(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105302716 B
(45)授权公告日 2019.05.24

(21)申请号 201410370865.7

(22)申请日 2014.07.30

(65)同一申请的已公布的文献号
申请公布号 CN 105302716 A

(43)申请公布日 2016.02.03

(73)专利权人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市福田区振兴路赛格科技园2栋东403室

(72)发明人 潘金赤 胡冬胜 马艳

(74)专利代理机构 北京康信知识产权代理有限责任公司 11240
代理人 吴贵明 张永明

(51)Int.Cl.
G06F 11/36(2006.01)

(56)对比文件
CN 101932999 A,2010.12.29,
CN 102799515 A,2012.11.28,
CN 1568458 A,2005.01.19,
CN 101448178 A,2009.06.03,
JP 2013191003 A,2013.09.26,
US 2004073581 A1,2004.04.15,
李宇,李方.淘宝网最佳实践之ABS自动编译.《www.uml.org.cn/jchgj/201110213.asp》.2011,第1-6页.

审查员 高新琳

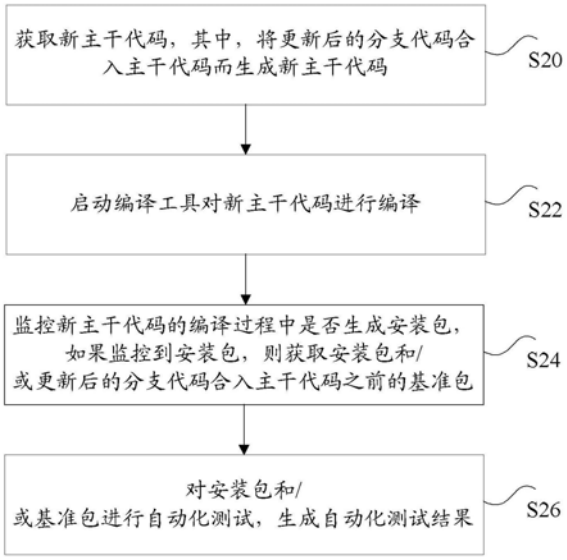
权利要求书2页 说明书13页 附图5页

(54)发明名称

合流开发模式下的测试方法、装置

(57)摘要

本发明公开了一种合流开发模式下的测试方法、装置。其中,该方法包括:获取新主干代码,其中,将更新后的分支代码合入主干代码而生成新主干代码;启动编译工具对新主干代码进行编译;监控新主干代码的编译过程中是否生成安装包,如果监控到安装包,则获取安装包和/或更新后的分支代码合入主干代码之前的基准包;对安装包和/或基准包进行自动化测试,生成自动化测试结果。本发明解决了现有技术中在分支代码合并主干代码的软件开发模式中,采用定时测试方法导致测试结果不准确、测试效率低的技术问题。



CN 105302716 B

1.一种合流开发模式下的测试方法,其特征在于,包括:

获取新主干代码,其中,将更新后的分支代码合入主干代码而生成所述新主干代码;

启动编译工具对所述新主干代码进行编译;

监控所述新主干代码的编译过程中是否生成安装包,如果监控到所述安装包,则获取所述安装包和/或更新后的分支代码合入所述主干代码之前的基准包;

对所述安装包和/或所述基准包进行自动化测试,生成自动化测试结果;

其中,所述基准包包括以下任意一种或多种数据包:所述主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将所述更新后的分支代码合入所述主干代码之前,对所述更新后的分支代码进行复位基底时所使用的主干代码。

2.根据权利要求1所述的方法,其特征在于,监控所述新主干代码的编译过程中生成安装包的步骤包括:

记录编译所述新主干代码的开始时间和结束时间;

如果扫描到所述新主干代码的编译结果的文件名称包含可执行文件名,且编译所述新主干代码的开始时间和结束时间的间隔时间在预定范围之内,则确定所述编译结果为所述新主干代码的安装包。

3.根据权利要求1或2所述的方法,其特征在于,对所述安装包和/或所述基准包进行自动化测试的步骤包括如下任意一个或多个自动化测试方案:

对所述安装包进行冒烟测试,得到冒烟测试结果;

测试所述安装包的文件大小是否小于等于预定文件大小,其中,如果所述安装包的文件大小小于等于所述预定文件大小,则测试通过;

对所述基准包进行性能测试,得到性能测试结果;

将所述新主干代码与所述基准包进行差异化对比,获取差异数据。

4.根据权利要求3所述的方法,其特征在于,在生成自动化测试结果之后,所述方法还包括:

判断所述自动化测试结果是否为自动化测试通过;

如果所述自动化测试结果为所述自动化测试通过,则结束合流,并对所述安装包进行黑盒测试;

如果所述自动化测试结果为所述自动化测试不通过,则重新进行合流或代码回滚。

5.根据权利要求4所述的方法,其特征在于,在获取新主干代码之前,所述方法还包括:

系统对保存所述更新前的分支代码的目录下发所述合流权限;

在所述更新前的分支代码获取到所述合流权限之后,将所述主干代码同步至所述更新前的分支代码的目录;

更新所述分支代码,并执行将所述更新后的分支代码合入所述主干代码的步骤。

6.根据权利要求5所述的方法,其特征在于,在满足以下任意一个或多个条件下,系统自动回收所述合流权限:

条件一:在自动化测试结果为所述自动化测试通过的情况下;

条件二:在对所述安装包进行黑盒测试,且黑盒测试结果通过的情况下;

条件三:在监控所述新主干代码的编译过程中生成所述安装包的情况下。

7.一种合流开发模式下的测试装置,其特征在于,包括:

获取模块，用于获取新主干代码，其中，将更新后的分支代码合入主干代码而生成所述新主干代码；

启动模块，用于启动编译工具对所述新主干代码进行编译；

监控模块，用于监控所述新主干代码的编译过程中是否生成安装包，如果监控到所述安装包，则获取所述安装包和/或更新后的分支代码合入所述主干代码之前的基准包；

自动化测试模块，用于对所述安装包和/或所述基准包进行自动化测试，生成自动化测试结果；其中，所述基准包包括以下任意一种或多种数据包：所述主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将所述更新后的分支代码合入所述主干代码之前，对所述更新后的分支代码进行复位基底时所使用的主干代码。

8.根据权利要求7所述的装置，其特征在于，所述监控模块包括：

记录模块，用于记录编译所述新主干代码的开始时间和结束时间；

确定模块，用于如果扫描到所述新主干代码的编译结果的文件名称包含可执行文件名，且编译所述新主干代码的开始时间和结束时间的间隔时间在预订范围之内，则确定所述编译结果为所述新主干代码的安装包。

9.根据权利要求7或8所述的装置，其特征在于，所述自动化测试模块包括如下任意一个或多个子自动化测试模块：

冒烟测试模块，用于对所述安装包进行冒烟测试，得到冒烟测试结果；

功能测试模块，用于测试所述安装包的文件大小是否小于等于预定文件大小，其中，如果所述安装包的文件大小小于等于所述预定文件大小，则测试通过；

性能测试模块，用于对所述基准包进行性能测试，得到性能测试结果；

差异化测试模块，用于将所述新主干代码与所述基准包进行差异化对比，获取差异数据。

10.根据权利要求9所述的装置，其特征在于，所述装置还包括：

判断模块，用于判断所述自动化测试结果是否为自动化测试通过；

第一处理模块，用于如果所述自动化测试结果为所述自动化测试通过，则结束合流，并对所述安装包进行黑盒测试；

第二处理模块，用于如果所述自动化测试结果为所述自动化测试不通过，则重新进行合流或代码回滚。

11.根据权利要求10所述的装置，其特征在于，所述装置还包括：

下发模块，用于系统对保存所述更新前的分支代码的目录下发所述合流权限；

同步模块，用于在所述更新前的分支代码获取到所述合流权限之后，将所述主干代码同步至所述更新前的分支代码的目录；

更新模块，用于更新所述分支代码，并执行将所述更新后的分支代码合入所述主干代码的步骤。

合流开发模式下的测试方法、装置

技术领域

[0001] 本发明涉及计算机测试领域，具体而言，涉及一种合流开发模式下的测试方法、装置。

背景技术

[0002] 现有技术的软件项目的开发过程中，开发人员都会对开发项目中的软件进行内测，对于众多软件项目尤其是大型软件团队开发的软件项目，现有技术都使用分支开发合并主干的软件开发模式。这种在多个分支上完成各自新功能的代码开发，然后，将分支上完成的新功能进行分支测试，在每个分支功能测试通过后，将分支代码再合入到主干代码所在的目录上。

[0003] 目前在分支代码合入主干代码的整个合流流程中，研发人员通常仅关注于代码合入前的分支代码的测试，在分支代码的测试结果不达标的情况下，则不允许分支代码合入主干代码，在分支代码的测试通过之后，会将分支代码合入主干代码。而对于分支代码合入主干代码后的校验，现有技术通常仅提供主干代码上人工的功能验收。

[0004] 具体的，在将分支代码进行合流之后，会对合入了分支代码的主干代码进行编译，由于无法获知安装包的准确获取时间，导致无法对合入后的主干代码做自动化工具检测，因此，在对编译得到的安装包进行测试的过程中，现有技术通常采用定时执行自动化工具的测试方法，以期望发现这段时间内代码变动锁引入的问题。

[0005] 由此可知，上述测试方案采用的定时测试方法，在多条分支代码合入主干代码的情况下，如果测试阶段才发现问题，则较难定位是哪支分支代码在合流中引入的问题，这种情况下只有通过二分法逐步定位，检查开销较大测试结果，导致了测试开销较大，测试效率低且测试结果不准确的问题。

[0006] 除了上述问题，现有技术上述测试方案还具有如下缺陷：无法准确获知开发人员何时完成分支代码的合流，则无法确定收回合流权限的准确时间，从而无法准确通知测试人员进行验收测试；开发人员对分支代码进行合流之后，在无法成功编译出安装包或安装包有严重问题的情况下，需要经开发测试工程师手工验证才能发现，浪费了人工操作开销；在开发人员合入前的分支代码检测达标时，但合入过程中漏合或者夹带了未经检验的代码，即实际合入的分支代码是不达标的，该类问题不易通过人工检测的方式发现，同样会导致编译失败或得到错误的安装包；无法对本次分支代码合流过程中的提交信息，如代码增量、性能影响做准确的度量，不利于统计开发工作量及效率评估。

[0007] 针对上述现有技术中在分支代码合并主干代码的软件开发模式中，采用定时测试方法导致测试结果不准确、测试效率低的问题，目前尚未提出有效的解决方案。

发明内容

[0008] 本发明实施例提供了一种合流开发模式下的测试方法、装置，以至少解决现有技术中在分支代码合并主干代码的软件开发模式中，采用定时测试方法导致测试结果不准

确、测试效率低的技术问题。

[0009] 根据本发明实施例的一个方面，提供了一种合流开发模式下的测试方法，该方法包括：获取新主干代码，其中，将更新后的分支代码合入主干代码而生成新主干代码；启动编译工具对新主干代码进行编译；监控新主干代码的编译过程中是否生成安装包，如果监控到安装包，则获取安装包和/或更新后的分支代码合入主干代码之前的基准包；对安装包和/或基准包进行自动化测试，生成自动化测试结果；其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新后的分支代码进行复位基底时所使用的主干代码。

[0010] 根据本发明实施例的另一方面，还提供了一种合流开发模式下的测试装置，该装置包括：获取模块，用于获取新主干代码，其中，将更新后的分支代码合入主干代码而生成新主干代码；启动模块，用于启动编译工具对新主干代码进行编译；监控模块，用于监控新主干代码的编译过程中是否生成安装包，如果监控到安装包，则获取安装包和/或更新后的分支代码合入主干代码之前的基准包；自动化测试模块，用于对安装包和/或基准包进行自动化测试，生成自动化测试结果；其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新后的分支代码进行复位基底时所使用的主干代码。

[0011] 在本发明实施例中，采用获取新主干代码，其中，将更新后的分支代码合入主干代码而生成新主干代码；启动编译工具对新主干代码进行编译；监控新主干代码的编译过程中是否生成安装包，如果监控到安装包，则获取安装包和/或更新后的分支代码合入主干代码之前的基准包；对安装包和/或基准包进行自动化测试，生成自动化测试结果的方式，通过将更新后的分支代码合入主干代码而生成新主干代码，并启动编译工具对新主干代码进行编译之后，通过监控编译过程来准确获知安装包的生成时间，在监控到安装包生成之后，可以获取安装包和/或更新后的分支代码合入主干代码之前的基准包进行后续的自动化测试，生成自动化测试结果，其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新后的分支代码进行复位基底时所使用的主干代码。上述实施方案，由于在编译生成安装包之后，就开始进行自动化测试，从而在多个分支进行合流的场景中，可以准确测试每个分支的合流结果，减少了测试开销，提高了测试效率和测试准确度。而且如果编译出的安装包有严重问题，也可以立刻检测出来。由此可知，针对上述现有技术中在分支代码合并主干代码的软件开发模式中，采用定时测试方法导致测试结果不准确、测试效率低的问题，目前尚未提出有效的解决方案。。

附图说明

[0012] 此处所说明的附图用来提供对本发明的进一步理解，构成本申请的一部分，本发明的示意性实施例及其说明用于解释本发明，并不构成对本发明的不当限定。在附图中：

[0013] 图1是本发明实施例的一种运行合流开发模式下的测试的计算机终端的硬件结构框图；

[0014] 图2是根据本发明实施例一的合流开发模式下的测试方法的流程图；

[0015] 图3是根据本发明实施例一的合流开发模式下的测试系统的业务流程图；

[0016] 图4是根据本发明实施例一的一种可选的合流开发模式下的测试方法的详细流程图;以及

[0017] 图5是根据本法实施例二的合流开发模式下的测试装置的结构示意图。

具体实施方式

[0018] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0019] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0020] 下面对本申请涉及到的名词进行解释:

[0021] 主干(trunk):在版本控制软件中(例如:SVN,git,clearcase等)稳定代码的保存路径,由此路径编译出软件的发布版本。通常情况下一个项目只会有一个主干路径存在。主干的目录下保存的是主干代码。

[0022] 分支(branch):在版本控制软件中开发/调试代码的保存路径,开发人员会在分支上进行新功能的开发和测试,当功能验证无误后再将代码合并至主干。通常情况下一个项目可以有多个分支存在。分支的目录下保存的是分支代码。

[0023] 复位基底(Rebase):将主干的主干代码同步至分支代码,以保证分支上的代码与主干差别不会太大,减少分支合并至主干时的代码冲突。

[0024] 配置管理服务器,用于运行配置管理系统(CMS),配置管理系统记录每个版本下,主干及分支的对应关系信息,例如主干分支路径,分支申请者,分支上实现的需求等。

[0025] 持续集成服务器,用于运行持续集成系统(CIS),持续集成系统用于主干和分支的代码编译,当主干和分支上的代码有发生变化,就可以自动或手工的启动构建,编译出指定版本的安装包。

[0026] 实施例1

[0027] 本发明实施例一提供了一种可以实施硬件终端上的方法实施例,需要说明的是,在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行,并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0028] 本申请实施例一所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。以运行在计算机终端上为例,图1是本发明实施例的一种运行合流开发模式下的测试的计算机终端的硬件结构框图。如图1所示,计算机终端10可以包括一个或多个

(图中仅示出一个)处理器102(处理器102可以包括但不限于微处理器MCU或可编程逻辑器件FPGA等的处理装置)、用于存储数据的存储器104、以及用于通信功能的传输装置106。本领域普通技术人员可以理解,图1所示的结构仅为示意,其并不对上述电子装置的结构造成限定。例如,计算机终端10还可包括比图1中所示更多或者更少的组件,或者具有与图1所示不同的配置。

[0029] 存储器104可用于存储应用软件的软件程序以及模块,如本发明实施例中的合流开发模式下的测试对应的程序指令/模块,处理器102通过运行存储在存储器104内的软件程序以及模块,从而执行各种功能应用以及数据处理,即实现上述的合流开发模式下的测试。存储器104可包括高速随机存储器,还可包括非易失性存储器,如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中,存储器104可进一步包括相对于处理器102远程设置的存储器,这些远程存储器可以通过网络连接至移动终端10。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0030] 传输装置106用于经由一个网络接收或者发送数据。上述的网络具体实例可包括移动终端10的通信供应商提供的无线网络。在一个实例中,传输装置106可以包括一个网络适配器(Network Interface Controller,NIC),其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中,传输装置106可以为射频(Radio Frequency,RF)模块,其用于通过无线方式与互联网进行通讯。

[0031] 在上述运行环境下,上述计算机终端10可以是一个监控服务器。本申请提供了如图2所示的合流开发模式下的测试。图2是根据本发明实施例一的合流开发模式下的测试方法的流程图。

[0032] 如图2所示,该合流开发模式下的测试可以包括如下步骤:

[0033] 步骤S20,获取新主干代码,其中,将更新后的分支代码合入主干代码而生成新主干代码。

[0034] 结合图3可知,本申请上述步骤S20使用分支开发合并主干的软件开发模式,分支代码和主干代码可以保存在代码服务器的不同目录下,也可以采用主干代码保存在代码服务器中,分支代码保存在开发人员的本地分支客户端中,在获取到合流权限之后,开发人员开始更新代码服务器或本地分支客户端上的分支代码,以开发新的功能,然后将更新后的分支代码合入主干代码中,生成新主干代码。

[0035] 步骤S22,启动编译工具对新主干代码进行编译。

[0036] 结合图3可知,在获取到新主干代码之后,本申请上述步骤S22可以通过启动持续集成服务器中的持续集成系统来调用新主干代码,进一步启动编译新主干代码。此处需要说明的是,本申请可以采用手动启动编译功能,也可以设置系统自动启动编译功能,例如,系统如果检测到一次合流便自动启动编译工具对新主干代码进行编译。

[0037] 步骤S24,监控新主干代码的编译过程中是否生成安装包,如果监控到安装包,则获取安装包和/或更新后的分支代码合入主干代码之前的基准包。其中,上述实施例中涉及到的基准包包括以下任意一种或多种数据包:主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前,对更新后的分支代码进行复位基底时所使用的主干代码。

[0038] 结合图3可知,本申请上述步骤S24中的安装包可以是开发人员启动编译并成功编

译出的安装包，可以通过监控服务器来实现监听持续集成服务器的编译过程中是否生成了安装包，例如，在启动编译新主干代码之后，监听服务器开始对编译结果进行扫描，一旦成功编译出安装包之后，便立即获取该安装包。上述监控过程可以准确获知生成该安装包的时刻，从而确定当前分支已经完成了分支代码的合流，可以通知系统进行下一步测试，比较现有的定时测试方案无法针对当前生成的安装包进行实时测试，使得测试结果无法准确定位到对应的分支的缺陷，上述方案由于对合流后编译得到的安装包进行监听，使得后续的测试过程如果测试出问题可以准确定位到是哪一只分支的问题。

[0039] 同时，在获取到安装包的同时，还可以获取到该分支对应的基准包，该基准包可以是：开发人员对合入前的分支代码进行复位基底rebase时所使用的主干代码，即在分支代码的开发者获得合流权限之后，在提交分支代码给主干代码进行合并之前，对当前的分支代码所做的复位基底rebase时所使用的主干代码；或者在给分支代码授予合流权限时，代码服务器中保存的当前主干的代码已经生成的安装包，作为分支代码合入前的基准包。

[0040] 例如，当主干代码为A、更新后的分支代码为B时，对分支代码所做的复位基底rebase是指将主干代码A同步至分支本地的分支代码B，此时，分支本地执行同步操作而得到的数据包为C，将同步过程中使用的主干代码A作为分支代码合并前的基准包。另外，在将更新后的分支代码合并入主干代码之后，会生成新主干代码D。

[0041] 本申请可以通过监控服务器来启动代码服务器进行合流，并启动持续集成服务器获取到代码服务器中合流后的主干代码。并监控持续集成服务器是否编译得到安装包。

[0042] 此处需要说明的是，本申请在监控到生成安装包之后，可以根据需求，仅获取安装包或基准包进入后续的测试工作，也可以同时将安装包和基准包都进行后续的测试工作。

[0043] 步骤S26，对安装包和/或基准包进行自动化测试，生成自动化测试结果。

[0044] 结合图3可知，本申请上述步骤S26可以实现在监控到当前编译新主干代码生成了一个安装包之后，便立即启动对上述安装包和/或基准包进行自动化测试，可以通过自动自动化测试启动工具来完成测试。进而可以利用生成的自动化测试结果进行后续的功能测试或者结束测试流程。

[0045] 由此，本申请上述实施例一提供的方案，在将更新后的分支代码合入主干代码而生成新主干代码，并启动编译工具对新主干代码进行编译之后，通过监控编译过程来准确获知安装包的生成时间，在监控到安装包生成之后，可以获取安装包和/或更新后的分支代码合入主干代码之前的基准包进行后续的自动化测试，生成自动化测试结果，其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新前的分支代码进行复位基底而生成的数据包。上述实施方案，由于在编译生成安装包之后，就开始进行自动化测试，从而在多个分支进行合流的场景中，可以准确测试每个分支的合流结果，减少了测试开销，提高了测试效率和测试准确度。而且如果编译出的安装包有严重问题，也可以立刻检测出来。由此可知，针对上述现有技术中在分支代码合并主干代码的软件开发模式中，采用定时测试方法导致测试结果不准确、测试效率低的问题，目前尚未提出有效的解决方案。

[0046] 此处还需要说明的是，由于本申请上述实施例中还提供了对基准包进行自动化测试的功能，提供了对开发人员合入前的分支代码进行检测的方案，因此，对于合入过程中漏合或者夹带了未经检验的代码，即实际合入的分支代码不达标的情况，都可以得到完整的

测试，避免了分支代码合流之前的各种失误所导致编译失败或得到错误的安装包的情况。同时还可以实现对本次分支代码合流过程中的提交信息，如代码增量、性能影响做出准确的度量。

[0047] 基于上述步骤S20至步骤S26提供的方案，在执行步骤S24中监控新主干代码的编译过程中生成安装包的功能时，可以通过如下优选的方案来实现上述步骤：

[0048] 步骤S241，记录编译新主干代码的开始时间和结束时间。

[0049] 步骤S243，如果扫描到新主干代码的编译结果的文件名称包含可执行文件名，且编译新主干代码的开始时间和结束时间的间隔时间在预订范围之内，则确定编译结果为新主干代码的安装包。

[0050] 本申请上述步骤S241至步骤S243实现了对编译得到的安装包的实时监控的方案。可以通过记录启动编译的开始时间和结束时间，以及如果扫描到一个包含了预定的文件名称和具有可执行文件特征的编译结果时，监控服务器就可以确定持续集成服务器完成了当前分支合流后的主干代码的编译。

[0051] 本申请上述实施例一的实现过程中，可以采用如下任意一个或多个自动化测试方案来实现上述对安装包和/或基准包进行自动化测试的步骤：

[0052] 方案一：对安装包进行冒烟测试，得到冒烟测试结果。

[0053] 上述方案提供了一种对安装包进行软件基本功能测试的方案，检查合入后的安装包是否可用。例如，是否可以成功运行、可以完成登陆等操作功能。

[0054] 此处需要说明的是，冒烟测试(smoke test)的对象是每一个新编译的需要进行后续功能测试的软件版本，目的是确认软件基本功能正常。冒烟测试的执行者一般是版本的编译人员，在编译得到安装包之后，可以通过自动化测试工具进行基本性能确认测试，例如是否可以正确安装/卸载，主要功能是否实现，是否存在严重死机或数据严重丢失等Bug。如果通过了该测试，则可以根据正式测试文档进行正式测试。否则，就需要重新编译版本，再次执行版本可接收确认测试，直到成功。

[0055] 方案二：测试安装包的文件大小是否小于等于预定文件大小，其中，如果安装包的文件大小小于等于预定文件大小，则测试通过。

[0056] 上述方案提供了对安装包的文件大小进行测试的功能，由于安装包的大小开发人员一般是可以预估的，涉及到用户体验等问题，因此，在安装包的大小超过预定文件大小的情况下，则可以确定安装包出现了错误，或者分支代码出现错误等问题，就需要进行进一步的验证或者直接返回重新更新分支代码，重新进行合流。

[0057] 方案三：对基准包进行性能测试，得到性能测试结果。

[0058] 上述方案中的性能测试可以包括应用在客户端性能的测试、应用在网络上性能的测试和应用在服务器端性能的测试。

[0059] 通常情况下，性能测试类型可以包括但不限于以下测试方法：负载测试，强度测试，容量测试等。其中，负载测试(Load Testing)是一种用于测试应用软件在一定时期内，最大支持多少并发的登陆用户数，软件请求出错率等问题的测试手段，目的是为了测试软件系统的性能；压力测试(Stress Testing)是一种测试硬件系统在一定时期内，针对系统的cpu利用率，内存使用率，磁盘I/O吞吐率，网络吞吐量等硬件资源提供消耗测试的测试手段；容量测试(Volume Testing)是一种用于确定系统最大承受量的测试手段，譬如系统最

大用户数，最大存储量，处理器最多并行处理的数据流量等。

[0060] 方案四：将更新后的分支代码合入主干代码之后生成的数据包与基准包进行差异化对比，获取差异数据。

[0061] 上述方案四提供的差异数据如果与合入测试前测试的数据有较大偏差的情况下，可以提醒研发人员进行进一步验证。

[0062] 例如，当主干代码为A、更新后的分支代码为B时，分支本地执行复位基底rebase操作而得到的数据包为C，上述复位基底rebase执行的同步过程中所使用的主干代码A作为分支代码合并前的基准包。另外，由于更新后的分支代码合并入主干代码之后得到的数据包就是新主干代码D，因此，将基准包A与数据包D进行比对之后，可以得到一个差异数据X1。

[0063] 此处分析可知，理论上，在生成新主干代码D之前，对分支代码B进行复位基底rebase时得到的合并数据包C应该与新主干代码D的内容相同，即数据包C与基准包A之间的比较结果应该与差异数据X1相同，因此，在将基准包A与数据包D进行比对得到差异数据X1之后，将数据包C与基准包A也进行比对得到差异数据X2，进一步利用这两个差异数据X1和X2进行差异化比较，如果比较结果有较大偏差，则可能存在合入过程中漏合或者夹带了未经检验的代码，可以提醒研发人员进行进一步验证。

[0064] 优选地，在上述步骤S26生成自动化测试结果之后，本申请还可以执行如下优选的实施步骤：

[0065] 步骤S271，判断自动化测试结果是否为自动化测试通过，其中，如果自动化测试结果为自动化测试通过，则执行步骤S273，如果自动化测试结果为自动化测试不通过，则执行步骤S275。

[0066] 步骤S273，结束合流，并对安装包进行黑盒测试。

[0067] 步骤S275，重新进行合流或代码回滚。

[0068] 本申请上述步骤S271至步骤S275实现了，对安装包和/或基准包进行自动化测试之后，进一步进行合入后的自动化检测电子流工具，可以通过验收服务器或者管理员及测试人工审核，来测试自动化测试结果，只有自动化测试通过的情况下，可以认定合流结束，进而进行进一步的黑盒测试并释放合流权限，否则开发需重新修改代码或回滚。

[0069] 本申请上述实施例一提供的方案中，在步骤S20执行获取新主干代码之前，还可以执行如下步骤实现的功能：

[0070] 步骤S201，系统对保存更新前的分支代码的目录下发合流权限。

[0071] 步骤S203，在更新前的分支代码获取到合流权限之后，将主干代码同步至更新前的分支代码的目录。

[0072] 步骤S205，更新分支代码，并执行将更新后的分支代码合入主干代码的步骤。

[0073] 优选地，在满足以下任意一个或多个条件下，系统自动回收合流权限：

[0074] 条件一：在自动化测试结果为自动化测试通过的情况下。

[0075] 条件二：在对安装包进行黑盒测试，且黑盒测试结果通过的情况下。

[0076] 条件三：在监控新主干代码的编译过程中生成安装包的情况下。

[0077] 本申请上述步骤实现了，在分支开发人员获得合流权限后，可以提供一种可以按照需求回收合流权限的方案，由于回流权限回收的时机至少包括上述三种条件下的时机，因此，除了可以提供在检测开发完成了所有的代码提交及编译时，就及时收回合流权限的

方案，还提供了其他灵活的回收方案。这种合流权限按照预定要求进行回收的方法，可以使得测试人员提供准确获取合入后的安装包来进行功能验收，不需要依赖开发人员通知测试人员使用哪个安装包，或者测试人员等待较长时间后使用最新的安装包进行验收，导致工作效率降低的问题。

[0078] 下面就结合图3和图4，对本申请的方案应用在上述完成一次合流及合流后的检测逻辑的应用场景所实现的功能进行详细描述。

[0079] 步骤A，开发人员获得分支的合流权限，分支代码获得合流权限后，分支代码合流到主干代码中。

[0080] 步骤B，持续集成监控系统启动编译该合入了分支代码的主干代码，得到一条编译任务。

[0081] 步骤C，通过监控服务器来监控持续集成监控系统上的编译任务。

[0082] 步骤D，判断是否编译成功该编译任务得到安装包，当有该开发启动的主干上的编译构建任务并成功编译出安装包时，则将该安装包视作代码合入完毕后启动的编译，并执行步骤E，如果没有构建任务或无法成功构建安装包，则返回步骤C继续监控扫描是否编译得到安装包。

[0083] 步骤E，在获取到分支代码合入主干代码后编译得到的安装包之后，获取分支代码合入主干代码之前的分支代码的二次rebase(即开发的分支代码获得合流权限后，且在提交分支代码到主干代码前所做的rebase)的数据包，或者给开发的分支代码授予合流权限时当前主干代码已经生成的安装包，作为分支代码合入主干代码之前，系统的基准包。

[0084] 步骤F，启动自动化测试工具对安装包和/或基准包进行自动化测试，自动化测试可以包括：合入后的安装包的冒烟测试、合入前和合入后的代码差异化对比、性能检测对比等，上述冒烟测试可以实现检查合入后安装包是否可用，而合入前和合入后的代码差异化对比可以获取合入的差异数据。

[0085] 步骤G，判断自动测试结果是否通过，在自动化测试工具测试通过之后，进入步骤H，当自动化测试工具检测不通过，例如性能下降明显，或者合入代码的差异化数据与合入前测试的数据有较大差异时，则提醒管理员进行检查确认。

[0086] 上述步骤的一种可选实施例中，如果管理员确认不通过，则此次合入未完成，开发需针对测试中发现的问题进行修复后重新合入，或进行代码回滚。如果管理员确认通过，则从查询该分支上开发了哪些需求，给这些需求的测试工程师发送测试通知，并提供合入后的安装包下载地址提示他们完成合入后的功能验收。

[0087] 步骤H，系统向测试工程师发出黑盒测试等功能性测试的测试通知。

[0088] 步骤I，判断上述黑盒测试等功能性测试的测试验收是否通过，如果测试验收不通过，则此次合入未完成，开发需针对测试中发现的问题进行修复后重新合入，或进行代码回滚，如果通过，则进入步骤J。

[0089] 步骤J，合流结束，系统自动收回合流权限。

[0090] 由此可知，本申请旨在提供一套完整的合流后安装包检测方案，测试方案可以至少包括自动化测试、人工确认测试、黑盒测试等，能将整个合流后校验逻辑串联起来，并通过自动化检测及催办的方式，提升验收的效率。管理员在收集本次合入的准确结果之后，可以方便检查违规操作和事后回溯，而且合流完毕后及时收回合流权限的功能，不影响其他

人进行分支合流的操作。

[0091] 需要说明的是，对于前述的各方法实施例，为了简单描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本发明并不受所描述的动作顺序的限制，因为依据本发明，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定是本发明所必须的。

[0092] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中，包括若干指令用以使得一台终端设备(可以是手机，计算机，服务器，或者网络设备等)执行本发明各个实施例所述的方法。

[0093] 实施例2

[0094] 根据本发明实施例，还提供了一种用于实施上述方法实施例的装置实施例，本申请上述实施例所提供的装置可以在计算机终端上运行。

[0095] 图5是根据本法实施例二的合流开发模式下的测试装置的结构示意图。如图5所示，该合流开发模式下的测试装置可以包括如下功能模块：获取模块50、启动模块52、监控模块54、自动化测试模块56。

[0096] 其中，获取模块50，用于获取新主干代码，其中，将更新后的分支代码合入主干代码而生成新主干代码；启动模块52，用于启动编译工具对新主干代码进行编译；监控模块54，用于监控新主干代码的编译过程中是否生成安装包，如果监控到安装包，则获取安装包和/或更新后的分支代码合入主干代码之前的基准包；自动化测试模块56，用于对安装包和/或基准包进行自动化测试，生成自动化测试结果；其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新后的分支代码进行复位基底时所使用的主干代码。

[0097] 本申请上述实施例一提供的方案，在将更新后的分支代码合入主干代码而生成新主干代码，并启动编译工具对新主干代码进行编译之后，通过监控编译过程来准确获知安装包的生成时间，在监控到安装包生成之后，可以获取安装包和/或更新后的分支代码合入主干代码之前的基准包进行后续的自动化测试，生成自动化测试结果，其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新前的分支代码进行复位基底而生成的数据包。上述实施方案，由于在编译生成安装包之后，就开始进行自动化测试，从而在多个分支进行合流的场景中，可以准确测试每个分支的合流结果，减少了测试开销，提高了测试效率和测试准确度。而且如果编译出的安装包有严重问题，也可以立刻检测出来。由此可知，针对上述现有技术中在分支代码合并主干代码的软件开发模式中，采用定时测试方法导致测试结果不准确、测试效率低的问题，目前尚未提出有效的解决方案。

[0098] 此处需要说明的是，由于本申请上述实施例中还提供了对基准包进行自动化测试的功能，提供了对开发人员合入前的分支代码进行检测的方案，因此，对于合入过程中漏合或者夹带了未经检验的代码，即实际合入的分支代码不达标的情况，都可以得到完整的测

试，避免了分支代码合流之前的各种失误所导致编译失败或得到错误的安装包的情况。同时还可以实现对本次分支代码合流过程中的提交信息，如代码增量、性能影响做出准确的度量。

[0099] 此处还需要说明的是，上述获取模块50、启动模块52、监控模块54、自动化测试模块56对应于实施例一中的步骤S20至步骤S26，四个模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例一所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例一提供的计算机终端10中。

[0100] 优选地，本申请上述监控模块54可以包括：记录模块541、确定模块543。

[0101] 其中，记录模块541，用于记录编译新主干代码的开始时间和结束时间；确定模块543，用于如果扫描到新主干代码的编译结果的文件名称包含可执行文件名，且编译新主干代码的开始时间和结束时间的间隔时间在预订范围之内，则确定编译结果为新主干代码的安装包。

[0102] 本申请上述记录模块541、确定模块543实现了对编译得到的安装包的实时监控的方案。可以通过记录启动编译的开始时间和结束时间，以及如果扫描到一个包含了预定的文件名称和具有可执行文件特征的编译结果时，监控服务器就可以确定持续集成服务器完成了当前分支合流后的主干代码的编译。

[0103] 此处还需要说明的是，上述记录模块541、确定模块543对应于实施例一中的步骤S241至步骤S243，二个模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例一所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例一提供的计算机终端10中。

[0104] 优选的，本申请上述实施例中的自动化测试模块56可以包括如下任意一个或多个子自动化测试模块：

[0105] 冒烟测试模块561，用于对安装包进行冒烟测试，得到冒烟测试结果。

[0106] 功能测试模块563，用于测试安装包的文件大小是否小于等于预定文件大小，其中，如果安装包的文件大小小于等于预定文件大小，则测试通过。

[0107] 性能测试模块565，用于对基准包进行性能测试，得到性能测试结果。

[0108] 差异化测试模块567，用于将分支代码合入主干代码之后生成的数据包与基准包进行差异化对比，获取差异数据。

[0109] 优选的，本申请上述实施例二中，在执行自动化测试模块56来生成自动化测试结果之后，上述装置还可以包括如下功能模块：判断模块571、第一处理模块573和第二处理模块575。

[0110] 其中，判断模块571，用于判断自动化测试结果是否为自动化测试通过；第一处理模块573，用于如果自动化测试结果为自动化测试通过，则结束合流，并对安装包进行黑盒测试；第二处理模块575，用于如果自动化测试结果为自动化测试不通过，则重新进行合流或代码回滚。

[0111] 本申请上述判断模块571、第一处理模块573和第二处理模块575实现了，对安装包和/或基准包进行自动化测试之后，进一步进行合入后的自动化检测电子流工具，可以通过验收服务器或者管理员及测试人工审核，来测试自动化测试结果，只有自动化测试通过的情况下，可以认定合流结束，进而进行进一步的黑盒测试并释放合流权限，否则开发需重新

修改代码或回滚。

[0112] 此处还需要说明的是，上述判断模块571、第一处理模块573和第二处理模块575对应于实施例一中的步骤S271至步骤S275，三个模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例一所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例一提供的计算机终端10中。

[0113] 优选的，本申请上述实施例二中，在执行获取模块20来获取新主干代码之前，上述装置还可以包括如下功能模块：下发模块201、同步模块203、更新模块205。

[0114] 其中，下发模块201，用于系统对保存更新前的分支代码的目录下发合流权限；同步模块203，用于在更新前的分支代码获取到合流权限之后，将主干代码同步至更新前的分支代码的目录；更新模块205，用于更新分支代码，并执行将更新后的分支代码合入主干代码的步骤。

[0115] 优选地，本申请上述实施例二提供了一种在满足以下任意一个或多个条件下，系统自动回收合流权限的方案：条件一：在自动化测试结果为自动化测试通过的情况下；条件二：在对安装包进行黑盒测试，且黑盒测试结果通过的情况下；条件三：在监控新主干代码的编译过程中生成安装包的情况下。

[0116] 本申请上述步骤实现了，在分支开发人员获得合流权限后，可以提供一种可以按照需求回收合流权限的方案，由于回流权限回收的时机至少包括上述三种条件下的时机，因此，除了可以提供在检测开发完成了所有的代码提交及编译时，就及时收回合流权限的方案，还提供了其他灵活的回收方案。这种合流权限按照预定要求进行回收的方法，可以使得测试人员提供准确获取合入后的安装包来进行功能验收，不需要依赖开发人员通知测试人员使用哪个安装包，或者测试人员等待较长时间后使用最新的安装包进行验收，导致工作效率降低的问题。

[0117] 此处还需要说明的是，上述下发模块201、同步模块203、更新模块205对应于实施例一中的步骤S201至步骤S205，三个模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例一所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例一提供的计算机终端10中。

[0118] 实施例3

[0119] 本发明的实施例还提供了一种存储介质。可选地，在本实施例中，上述存储介质可以用于保存上述实施例一所提供的合流开发模式下的测试方法所执行的程序代码。

[0120] 可选地，在本实施例中，上述存储介质可以位于计算机网络中计算机终端群中的任意一个计算机终端中，或者位于移动终端群中的任意一个移动终端中。

[0121] 可选地，在本实施例中，存储介质被设置为存储用于执行以下步骤的程序代码：获取新主干代码，其中，将更新后的分支代码合入主干代码而生成新主干代码；启动编译工具对新主干代码进行编译；监控新主干代码的编译过程中是否生成安装包，如果监控到安装包，则获取安装包和/或更新后的分支代码合入主干代码之前的基准包；对安装包和/或基准包进行自动化测试，生成自动化测试结果；其中，基准包包括以下任意一种或多种数据包：主干代码的安装包、在更新前的分支代码获取到合流权限之后且在将更新后的分支代码合入主干代码之前，对更新前的分支代码进行复位基底而生成的数据包。

[0122] 可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：记录编译新主

干代码的开始时间和结束时间;如果扫描到新主干代码的编译结果的文件名称包含可执行文件名,且编译新主干代码的开始时间和结束时间的间隔时间在预订范围之内,则确定编译结果为新主干代码的安装包。

[0123] 可选的,存储介质还被设置为存储用于执行以下步骤的程序代码:对安装包进行冒烟测试,得到冒烟测试结果;测试安装包的文件大小是否小于等于预定文件大小,其中,如果安装包的文件大小小于等于预定文件大小,则测试通过;对基准包进行性能测试,得到性能测试结果;将分支代码合入主干代码之后生成的数据包与基准包进行差异化对比,获取差异数据。

[0124] 可选的,存储介质还被设置为存储用于执行以下步骤的程序代码:判断自动化测试结果是否为自动化测试通过;如果自动化测试结果为自动化测试通过,则结束合流,并对安装包进行黑盒测试;如果自动化测试结果为自动化测试不通过,则重新进行合流或代码回滚。

[0125] 可选的,存储介质还被设置为存储用于执行以下步骤的程序代码:系统对保存更新前的分支代码的目录下发合流权限;在更新前的分支代码获取到合流权限之后,将主干代码同步至更新前的分支代码的目录;更新分支代码,并执行将更新后的分支代码合入主干代码的步骤。

[0126] 可选的,存储介质还被设置为存储用于执行以下步骤的程序代码:在满足以下任意一个或多个条件下,系统自动回收合流权限:条件一:在自动化测试结果为自动化测试通过的情况下;条件二:在对安装包进行黑盒测试,且黑盒测试结果通过的情况下;条件三:在监控新主干代码的编译过程中生成安装包的情况下。

[0127] 可选地,在本实施例中,上述存储介质可以包括但不限于:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0128] 可选地,本实施例中的具体示例可以参考上述实施例1和实施例2中所描述的示例,本实施例在此不再赘述。

[0129] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0130] 上述实施例中的集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在上述计算机可读取的存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在存储介质中,包括若干指令用以使得一台或多台计算机设备(可为个人计算机、服务器或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。

[0131] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中没有详述的部分,可以参见其他实施例的相关描述。

[0132] 在本申请所提供的几个实施例中,应该理解到,所揭露的客户端,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连

接,可以是电性或其它的形式。

[0133] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0134] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0135] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

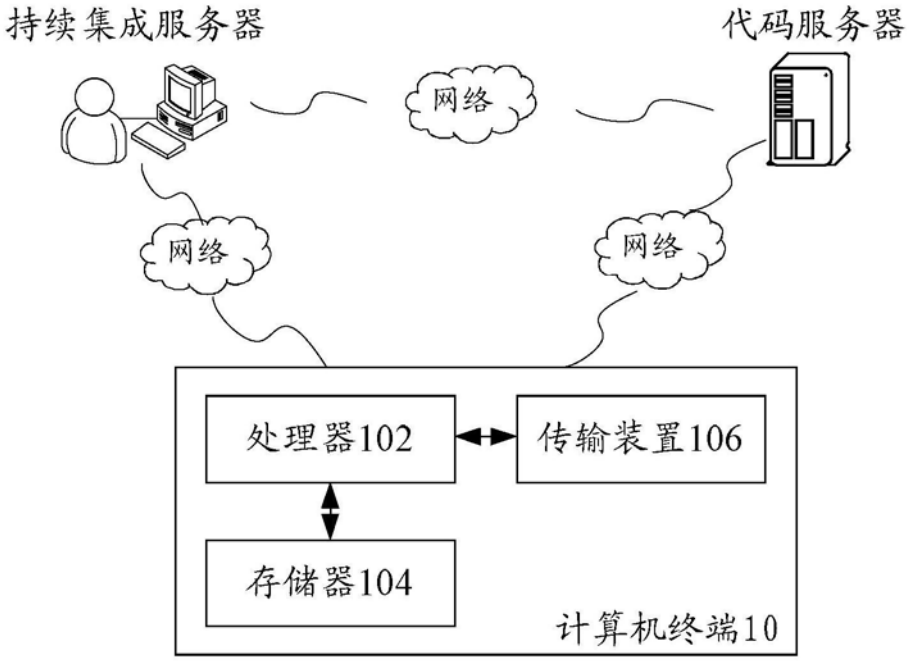


图1

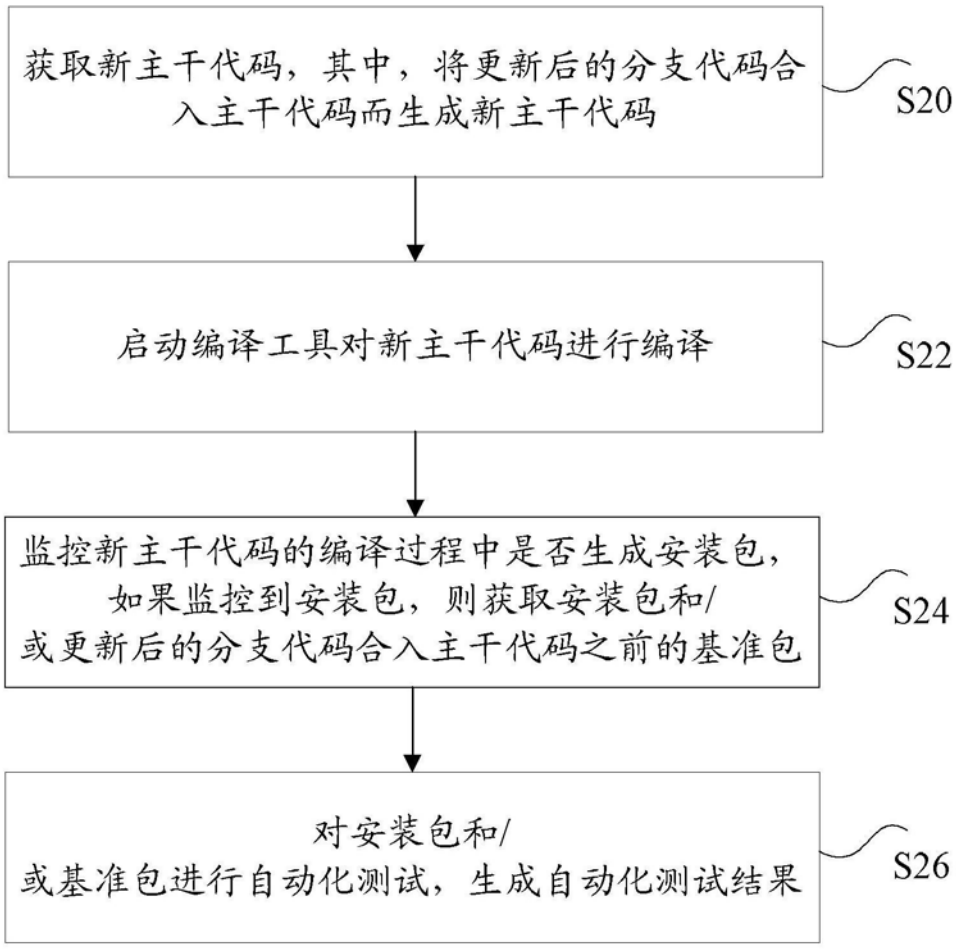

获取新主干代码，其中，将更新后的分支代码合入主干代码而生成新主干代码
S20
启动编译工具对新主干代码进行编译
S22
监控新主干代码的编译过程中是否生成安装包，如果监控到安装包，则获取安装包和/或更新后的分支代码合入主干代码之前的基准包
S24
对安装包和/或基准包进行自动化测试，生成自动化测试结果
S26

图2

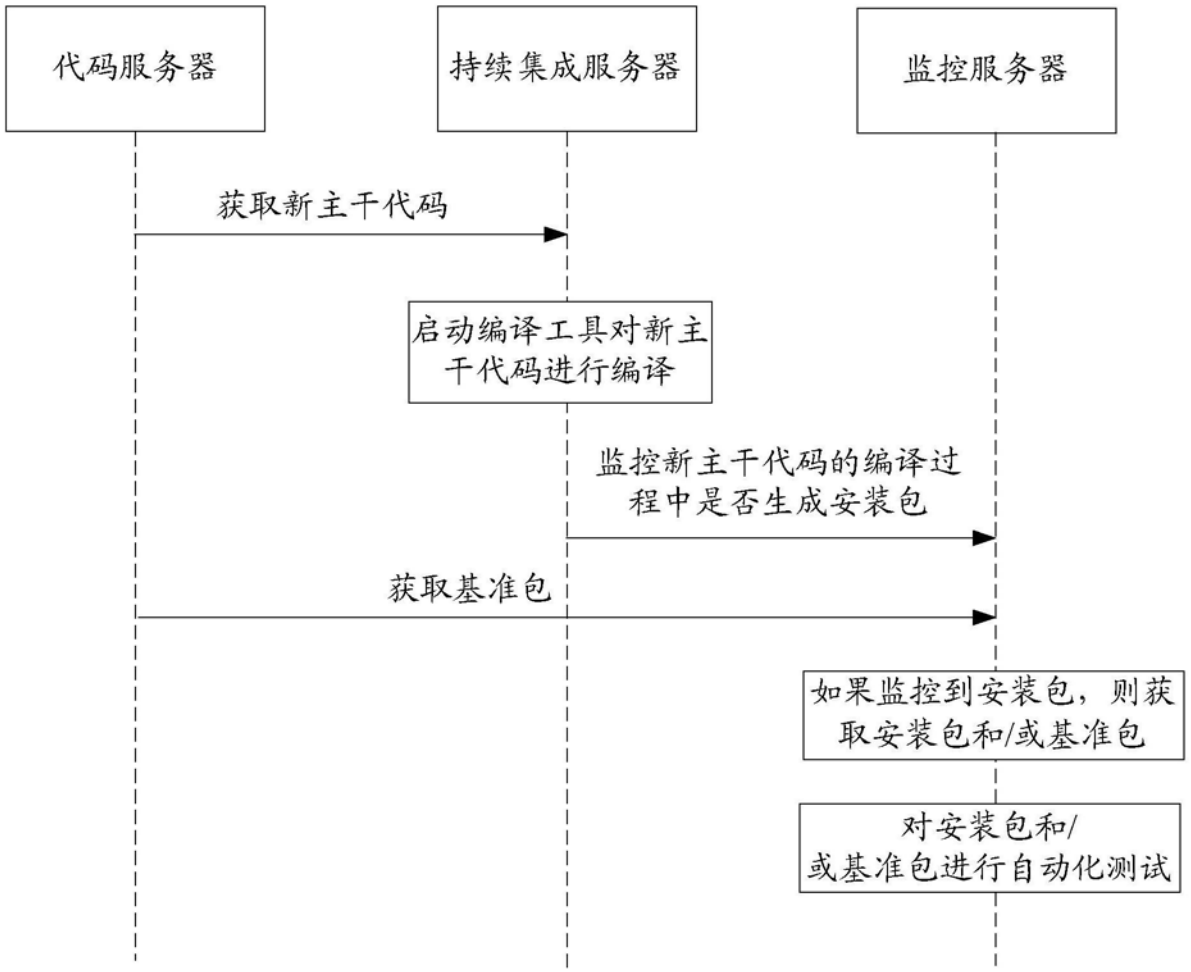

代码服务器
持续集成服务器
监控服务器
获取新主干代码
启动编译工具对新主干代码进行编译
监控新主干代码的编译过程中是否生成安装包
获取基准包
如果监控到安装包，则获取安装包和/或基准包
对安装包和/或基准包进行自动化测试

图3

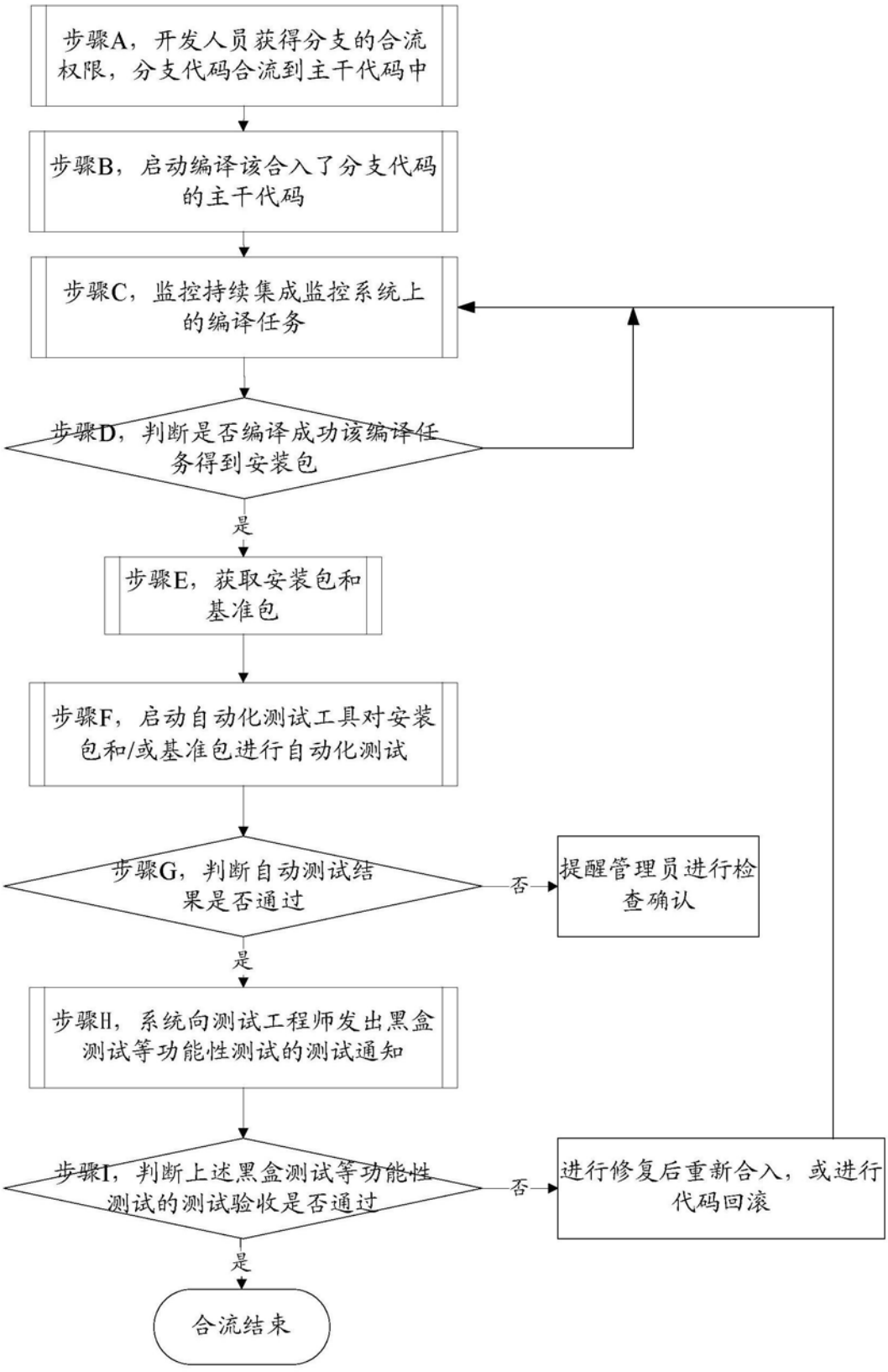

步骤A，开发人员获得分支的合流权限，分支代码合流到主干代码中
步骤B，启动编译该合入了分支代码的主干代码
步骤C，监控持续集成监控系统上的编译任务
步骤D，判断是否编译成功该编译任务得到安装包
是
步骤E，获取安装包和基准包
步骤F，启动自动化测试工具对安装包和/或基准包进行自动化测试
步骤G，判断自动测试结果是否通过
否
提醒管理员进行检查确认
是
步骤H，系统向测试工程师发出黑盒测试等功能性测试的测试通知
步骤I，判断上述黑盒测试等功能性测试的测试验收是否通过
否
进行修复后重新合入，或进行代码回滚
是
合流结束

图4

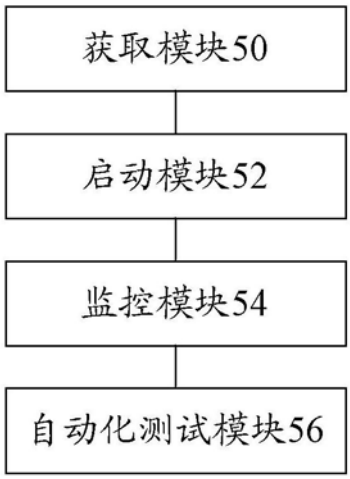


图5