

权 利 要 求 书

1、一种自动化测试构建方法，其特征在于，所述的自动化测试构建方法包括以下步骤：

(1) 得到最新的被测试对象；

(2) 得到针对该被测试对象的最新测试程序及其相应的标准输入信息和标准输出信息；

(3) 从缺陷管理系统中得到运行出错测试程序列表，并根据该运行出错测试程序列表，在步骤(2)得到的最新测试程序中剔除相应的测试程序，生成运行测试程序列表；

(4) 根据步骤(3)生成的运行测试程序列表运行相应的测试程序，得到对应测试程序的运行输出结果；

(5) 将步骤(4)中的测试程序的运行输出结果与相应的标准输出信息进行比较，如果运行输出结果和标准输出信息不一致，则记录测试程序的名称，生成原始出错测试程序列表；

(6) 将步骤(5)生成的原始出错测试程序列表导入缺陷管理系统；

(7) 缺陷管理系统处理步骤(6)导入的信息，并生成本次测试构建报告；

(8) 缺陷管理系统将本次测试构建结果输出；

(9) 根据缺陷修复的信息，对缺陷管理系统中相应的缺陷状态进行修改；

(10) 在下一一次测试构建过程开始时，缺陷管理系统生成新的运行出错测试程序列表。

2、根据权利要求1所述的自动化测试构建方法，其特征在于，所述的得到最新的被测试对象，包括以下步骤：

(1) 如果被测试对象是源代码，则先获取最新的源代码，然后编译该源代码，生成被测试对象；

(2) 如果被测试对象是可以执行的文件，则直接获取这些文件。

3、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的得到针对该被测试对象的最新测试程序，包括以下步骤：

(1) 如果测试程序为源代码，则先获取最新的测试代码，然后编译该测试代码，生成测试程序；

(2) 如果测试程序是可以直接执行的文件，则直接获取该测试程序。

4、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的标准输入信息为标准输入文件形式。

5、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的标准输出信息为标准输出文件形式。

6、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的测试程序的运行输出结果为运行输出结果文件形式。

7、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的缺陷管理系统处理导入的原始出错测试程序列表信息，包括以下步骤：

(1) 根据原始出错测试程序列表信息，缺陷管理系统在进行相应处理后给出提示信息，并将待处理的缺陷信息自动提交给用户；

(2) 用户根据该提示信息对待处理的缺陷信息进行处理。

8、根据权利要求 7 所述的自动化测试构建方法，其特征在于，所述的缺陷管理系统在进行相应处理后给出提示信息，包括以下步骤：

(1) 根据原始出错测试程序列表信息在缺陷管理系统中进行查询，如果该测试程序在缺陷管理系统中没有记录，则根据具体情况产生“该缺陷为一个新缺陷”的提示信息，或者产生“该缺陷为一种意外错误”的提示信息；

(2) 经过查询，如果该测试程序在缺陷管理系统中有记录，则查询其对应缺陷修复时间，如果该缺陷修复的时间是在上次测试构建之后，则根据具体情况产生“在上次测试构建以后修复的该缺陷可能没有真正修复，修复验证失败，该缺陷需要被再次设置为未修复状态”的提示信息，或者产生“该缺陷为该测

试程序发现的新缺陷”的提示信息；

(3) 查询其对应缺陷修复时间，如果该缺陷修复的时间是在上次测试构建之前，则根据具体情况产生“该缺陷是一个已经发现的缺陷再次复发”的提示信息，并将与这个测试程序相关的缺陷内容提供给用户，或者产生“该缺陷为该测试程序发现的新缺陷”的提示信息。

(4) 对于在上次测试构建之后修复的缺陷没有再次出现在原始出错测试程序列表中的测试程序，缺陷管理系统产生“该缺陷已经被成功修复”的提示信息。

9、根据权利要求 8 所述的自动化测试构建方法，其特征在于，所述的用户根据该提示信息对待处理的缺陷信息进行处理，包括以下步骤：

(1) 如果该提示信息表明该缺陷为一个新缺陷，则用户根据情况提交新的缺陷，并且缺陷管理系统自动将此缺陷和发现缺陷的测试程序名绑定；

(2) 如果该提示信息表明该缺陷未修复成功，则用户将该已经修复的缺陷状态设置为未修复状态；

(3) 如果该提示信息表明该缺陷为再次复发的缺陷，则用户将该已经修复的缺陷状态设置为复发状态；

(4) 如果该提示信息表明该缺陷已经被成功修复，则用户将保持该已经修复的缺陷状态不变。

10、根据权利要求 7、8 或 9 所述的自动化测试构建方法，其特征在于，所述的用户为构建负责人和缺陷管理系统的其他用户。

11、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的测试构建报告包括本次测试构建新发现缺陷数量、正常修复缺陷的数量和复发缺陷的数量报告，新发现的缺陷名称报告，正常修复的缺陷名称报告，复发缺陷的名称报告，本次测试构建中测试程序发现缺陷的数量和测试程序累计发现缺陷的数量的统计报告。

12、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的缺

陷管理系统将本次测试构建结果输出方式包括通过 Email 将测试构建结果信息发送给开发人员、提供一个固定的 URL 链接由开发人员访问并获取该测试构建结果。

13、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的根据缺陷修复的信息，对缺陷管理系统中相应的缺陷状态进行修改，包括以下步骤：

(1) 版本控制工具记录开发人员提交的进行缺陷修复的代码的文件名称信息；

(2) 项目管理系统同时也记录相应的开发人员提交的进行缺陷修复的代码的文件名称信息，并根据该信息建立缺陷修复记录；

(3) 项目管理系统将该缺陷修复记录与缺陷管理系统中的相应缺陷之间建立关联关系；

(4) 缺陷管理系统根据项目管理系统的缺陷修复记录的关联信息将相应缺陷的状态设置为已修复状态。

14、根据权利要求 13 所述的自动化测试构建方法，其特征在于，所述的将本次测试构建过程生成的原始出错测试程序列表导入缺陷管理系统的步骤后还包括以下步骤：

(1) 缺陷管理系统根据该原始出错测试程序列表中的测试程序对应缺陷和以往发现过的该缺陷的历史信息，在项目管理系统中通过所关联的缺陷修复记录定位与该缺陷相关的代码文件；

(2) 在项目管理系统中根据该缺陷相关的代码文件查询上次测试构建以后的代码提交记录，确定相应的开发人员和记录；

(3) 如果该缺陷是首次出现，则建立该缺陷与所关联的代码文件之间的耦合度因子；如果该缺陷是重复出现的，则增加该缺陷与所关联的代码文件之间的耦合度因子的值。

15、根据权利要求 1 所述的自动化测试构建方法，其特征在于，所述的缺

陷管理系统生成的新的运行出错测试程序列表中的测试程序为所有被上次测试构建过程发现的但还没有被修复的缺陷所对应的测试程序和新导入该缺陷管理系统的但还未被用户处理的缺陷所对应的测试程序。

16、根据权利要求 15 所述的自动化测试构建方法，其特征在于，所述的用户为构建负责人和缺陷管理系统的其他用户。

说明书

自动化测试构建方法

技术领域

本发明涉及计算机技术领域，特别涉及计算机环境中运行的软件管理过程领域，具体是指一种自动化测试构建方法。

背景技术

测试构建是通过自动化的方式进行软件的测试。如果这个构建过程是每日执行一次就是日构建（Daily Build）或者叫做每日集成，如果是每日多次就叫做持续集成。

一般的软件开发流程通常是，一个项目被分解后分配任务，由不同的人负责不同的软件部件，在开发完成之后，再把各人的部件整合起来，形成完整的软件，但是这种做法在实践中却有很多问题。

首先，这种方式适合开发人员之间工作彼此没有交集的情况，以前这种现象很常见，但是现在，随着软件规模的扩大、分工合作的加深，开发人员间的相互依赖程度越来越高，这种清晰的职责划分已经变得越来越难了。

其次，在软件集成时，往往会出现各种各样的问题，可是却很难发现到底问题在哪里。似乎每个人的代码都没有问题，但结合到一起就出现大量的问题。

所以日构建就将平时难得一见的集成工作转换成频繁进行的一件工作，从而使得复杂的集成变成了一件简单的工作。通过以天集成，排除缺陷（Bug）就变成一件很容易的事情了。

测试构建的目的是为了发现缺陷，并可以在以后的测试构建中验证修复的缺陷。在现有技术中，关于测试构建过程、缺陷管理等都有记载，但这几个部分在现有技术中都相对独立集成。测试构建技术是软件质量保证的一个重要部

分，是对产品的自动测试过程。在公知的技术中有关于测试构建的知识和有关于测试构建过程的流程，其中实现测试构建的一个基本前提是构建必需与版本控制系统结合；对于缺陷管理系统，在公知的技术中一般是一个相对独立的系统；对于项目管理系统，在公知的技术中一般也是一个相对独立的系统。

现有测试构建的过程，因为和缺陷管理系统是独立进行的过程，这样每当在测试构建中的测试程序发现缺陷时，这个缺陷需要另外记录和管理，同时为了保证每次的测试构建只发现新问题，在原有的测试构建过程中需要手动使得已发现缺陷并且尚未修复的测试程序停止运行。当缺陷被修复后，缺陷的管理者会通知测试构建人员，再手动将那个发现缺陷的测试程序加入到下一次的测试构建中，下一次测试构建过程会进行对这个缺陷修复情况的验证，如果还有问题就又需要重复如上的操作。

如上的流程都是手动的，对操作者来说，将会是非常繁重的工作。

同时，在现有的缺陷管理系统中，由于其是一个自封闭的系统，只能记录、跟踪和管理被用户手动填写到系统中的缺陷，当缺陷被修复后，只能通过用户来将这个缺陷修复的状况报告出来，因此不能实现与测试构建过程的互动，不能自动导入测试构建中发现的问题，不能将修复的缺陷自动加入到测试构建过程中去验证。

因此，现有技术中缺陷管理系统、项目管理系统与测试构建过程相对独立的问题，使得自动测试过程和缺陷管理系统之间的信息共享和传递无法实现，不能自动的解决软件过程中发现的缺陷和验证修复的缺陷，而且也不能自动实现测试程序和被测试对象的对应的问题。

发明内容

本发明的目的是克服了上述现有技术中的缺点，提供一种基于缺陷管理的自动化测试构建方法，该方法实现了测试构建和缺陷管理系统的集成，可以将测试构建过程中发现的问题自动导入到缺陷管理系统中，在下一次测试构建中，

又可以将缺陷系统中经过处理的数据导出，来自动验证在上一次测试构建过程后修复的缺陷，实现测试构建过程发现缺陷和验证修复缺陷的自动化。

本发明的另一个目的在于提供一种基于版本控制工具、结合缺陷管理系统和项目管理系统自动化测试构建方法，该方法实现了项目管理系统和缺陷管理系统的结合，使得在项目管理系统中通过代码的名字就可以自动访问代码在版本控制工具上的源文件，同时实现了项目管理系统和缺陷管理系统的结合，将测试程序和测试对象结合起来。

为了实现上述的目的，本发明的自动化测试构建方法如下：

该自动化测试构建方法，其主要特点是，该方法包括以下步骤：

- (1) 得到最新的被测试对象；
- (2) 得到针对该被测试对象的最新测试程序及其相应的标准输入信息和标准输出信息；
- (3) 从缺陷管理系统中得到运行出错测试程序列表，并根据该运行出错测试程序列表，在步骤(2)得到的最新测试程序中剔除相应的测试程序，生成运行测试程序列表；
- (4) 根据步骤(3)生成的运行测试程序列表运行相应的测试程序，得到对应测试程序的运行输出结果；
- (5) 将步骤(4)中的测试程序的运行输出结果与相应的标准输出信息进行比较，如果运行输出结果和标准输出信息不一致，则记录测试程序的名称，生成原始出错测试程序列表；
- (6) 将步骤(5)生成的原始出错测试程序列表导入缺陷管理系统；
- (7) 缺陷管理系统处理步骤(6)导入的信息，并生成本次测试构建报告；
- (8) 缺陷管理系统将本次测试构建结果输出；
- (9) 根据缺陷修复的信息，对缺陷管理系统中相应的缺陷状态进行修改；
- (10) 在下一次测试构建过程开始时，缺陷管理系统生成新的运行出错测试程序列表。

重复以上步骤（1）至步骤（10）。

该自动化测试构建方法中的得到最新的被测试对象，包括以下步骤：

（1）如果被测试对象是源代码，则先获取最新的源代码，然后编译该源代码，生成被测试对象；

（2）如果被测试对象是可以执行的文件，则直接获取这些文件。

该自动化测试构建方法中的得到针对该被测试对象的最新测试程序，包括以下步骤：

（1）如果测试程序为源代码，则先获取最新的测试代码，然后编译该测试代码，生成测试程序；

（2）如果测试程序是可以直接执行的文件，则直接获取该测试程序。

该自动化测试构建方法中的标准输入信息为标准输入文件形式。

该自动化测试构建方法中的标准输出信息为标准输出文件形式。

该自动化测试构建方法中的测试程序的运行输出结果为运行输出结果文件形式。

该自动化测试构建方法中的缺陷管理系统处理导入的原始出错测试程序列表信息，包括以下步骤：

（1）根据原始出错测试程序列表信息，缺陷管理系统在进行相应处理后给出提示信息，并将待处理的缺陷信息自动提交给用户；

（2）用户根据该提示信息对待处理的缺陷信息进行处理。

该自动化测试构建方法中的缺陷管理系统在进行相应处理后给出提示信息，包括以下步骤：

（1）根据原始出错测试程序列表信息在缺陷管理系统中进行查询，如果该测试程序在缺陷管理系统中没有记录，则根据具体情况产生“该缺陷为一个新缺陷”的提示信息，或者产生“该缺陷为一种意外错误”的提示信息；

（2）经过查询，如果该测试程序在缺陷管理系统中有记录，则查询其对应缺陷修复时间，如果该缺陷修复的时间是在上次测试构建之后，则根据具体情

况产生“在上次测试构建以后修复的该缺陷可能没有真正修复，修复验证失败，该缺陷需要被再次设置为未修复状态”的提示信息，或者产生“该缺陷为该测试程序发现的新缺陷”的提示信息；

(3) 查询其对应缺陷修复时间，如果该缺陷修复的时间是在上次测试构建之前，则根据具体情况产生“该缺陷是一个已经发现的缺陷再次复发”的提示信息，并将与这个测试程序相关的缺陷内容提供给用户，或者产生“该缺陷为该测试程序发现的新缺陷”的提示信息。

(4) 对于在上次测试构建之后修复的缺陷没有再次出现在原始出错测试程序列表中的测试程序，缺陷管理系统产生“该缺陷已经被成功修复”的提示信息。

该自动化测试构建方法中的用户根据该提示信息对待处理的缺陷信息进行处理，包括以下步骤：

(1) 如果该提示信息表明该缺陷为一个新缺陷，则用户根据情况提交新的缺陷，并且缺陷管理系统自动将此缺陷和发现缺陷的测试程序名绑定；

(2) 如果该提示信息表明该缺陷未修复成功，则用户将该已经修复的缺陷状态设置为未修复状态；

(3) 如果该提示信息表明该缺陷为再次复发的缺陷，则用户将该已经修复的缺陷状态设置为复发状态；

(4) 如果该提示信息表明该缺陷已经被成功修复，则用户将保持该已经修复的缺陷状态不变。

该自动化测试构建方法中的测试构建报告包括：本次测试构建新发现缺陷数量、正常修复缺陷的数量和复发缺陷的数量报告，新发现的缺陷名称报告，正常修复的缺陷名称报告，复发缺陷的名称报告，本次测试构建中测试程序发现缺陷的数量和测试程序累计发现缺陷的数量的统计报告。

该自动化测试构建方法中的缺陷管理系统将本次测试构建结果输出方式包括：通过 Email 将测试构建结果信息发送给开发人员、提供一个固定的 URL 链

接由开发人员访问并获取该测试构建结果。

该自动化测试构建方法中的根据缺陷修复的信息，对缺陷管理系统中相应的缺陷状态进行修改，包括以下步骤：

（1）版本控制工具记录开发人员提交的进行缺陷修复的代码的文件名称信息；

（2）项目管理系统同时也记录相应的开发人员提交的进行缺陷修复的代码的文件名称信息，并根据该信息建立缺陷修复记录；

（3）项目管理系统将该缺陷修复记录与缺陷管理系统中的相应缺陷之间建立关联关系；

（4）缺陷管理系统根据项目管理系统的缺陷修复记录的关联信息将相应缺陷的状态设置为已修复状态。

该自动化测试构建方法中的将本次测试构建过程生成的原始出错测试程序列表导入缺陷管理系统的步骤后还包括以下步骤：

（1）缺陷管理系统根据该原始出错测试程序列表中的测试程序对应缺陷和以往发现过的该缺陷的历史信息，在项目管理系统中通过所关联的缺陷修复记录定位与该缺陷相关的代码文件；

（2）在项目管理系统中根据该缺陷相关的代码文件查询上次测试构建以后的代码提交记录，确定相应的开发人员和记录；

（3）如果该缺陷是首次出现，则建立该缺陷与所关联的代码文件之间的耦合度因子；如果该缺陷是重复出现的，则增加该缺陷与所关联的代码文件之间的耦合度因子的值。

该自动化测试构建方法中的缺陷管理系统生成的新的运行出错测试程序列表中的测试程序为所有被上次测试构建过程发现的但还没有被修复的缺陷所对应的测试程序和新导入该缺陷管理系统的但还未被用户处理的缺陷所对应的测试程序。

该自动化测试构建方法中的用户为构建负责人和缺陷管理系统的其他用

户。

采用了该发明的自动化测试构建方法，由于基于缺陷管理的测试构建过程是将缺陷管理和构建过程有机结合起来，使得缺陷的发现和修复后的验证都能自动完成，提高了开发和测试的效率，节省了人员的工作投入；由于缺陷管理系统和项目管理系统也都是以源代码管理为基础，项目管理系统中的代码文件名和缺陷管理系统中的测试文件名都可以自动链接到版本控制工具上的对应文件，保证了一个开发团队共享统一的代码源；由于测试构建过程的结果会直接导入到缺陷管理系统中，对于测试结果的细节可以完整的记录下来，并对于后续的测试构建起到了指导作用，使得集成测试的效率大大提高；由于整个测试构建过程全部是自动进行的，没有人为干预，同时缺陷管理系统和项目管理系统结合，可以自动定位测试程序和测试对象，因此充分保证了执行过程的可靠性和缺陷定位的准确性；同时，由于通过测试构建系统与缺陷管理系统的结合，可以将每次测试构建中发现的问题进行分析和统计，并通过柱形图，坐标图等方式显示，为产品的质量提供可供参考的依据。

附图说明

图 1 为标准的测试构建过程模型流程图。

图 2 为本发明的基于缺陷管理的测试构建过程模型流程图。

图 3 为本发明的缺陷管理系统处理导入的原始出错测试程序列表信息的流程图。

图 4 为本发明的项目管理系统根据缺陷修复自动更改缺陷管理系统中的缺陷状态的流程图。

图 5 为本发明的版本控制工具、项目管理系统和缺陷管理系统结合共同建立测试程序与被测试对象的关联关系的流程图。

具体实施方式

为了能够更清楚地理解本发明的技术内容，特举以下实施例详细说明。

整个自动化测试构建方法全过程包含以下四个模块的结合：

- 版本控制工具
- 缺陷管理系统
- 项目管理系统
- 测试构建过程

1. 版本控制工具

源代码是开发过程的主要组成部分，通过版本控制工具来管理源代码可以统一整个开发团队的代码源。在公知的技术中有很多的版本控制工具，都可以用来进行源代码管理。这里所指的版本控制工具的必须要具备如下特征：

- 可以进行文件的版本控制（用户可以进行文件的检入和检出操作）
- 有基于命令行的工具可以自动进行文件的检入和检出

2. 缺陷管理系统

缺陷管理系统的作用是记录、跟踪和管理软件开发过程中发现的缺陷（Bug）。在公知的技术中有很多缺陷管理系统。想要实现自动化测试构建方法中的上述结合，缺陷管理系统必须增加如下特征：

- 可以记录和操作缺陷
- 缺陷必需具备状态（修复的缺陷还是没有修复的缺陷）、时间信息（修复时间）和缺陷修复负责人等信息
- 缺陷可以与相应的一个测试程序关联，缺陷可以由测试程序发现（但大多数情况下，缺陷并不是通过测试程序发现的）

3. 项目管理系统

项目管理系统可以将整个开发过程中的工作分解，同时开发过程中，开发

人员对版本控制工具上文件的更改，必须在项目管理系统中进行记录，通过访问这些记录中的文件名就可以自动访问在版本控制工具上的代码内容。在公知的技术中有很多项目管理工具。在本发明中实现的结合，项目管理系统必须要具备如下特征：

- 项目管理系统中有工作划分
- 项目管理系统中工作的实施过程有代码的提交和记录，提交的代码可以和与版本控制工具结合，所有提交到版本控制工具上的文件，在项目管理系统中有 **Checkin** 记录，一个记录包括本次更改的文件有哪些（也就是说可以有多个文件，这多个文件是一批的，完成某个特定功能），实现什么样的功能
- 缺陷修复也是对代码的操作，在项目管理系统中提交记录时可以自动更改缺陷状态

4. 测试构建过程

测试构建过程，也就是自动化测试过程，它是通过自动完成以下几个关键步骤而实现的：

- （1）获取（或者生成）最新的被测试对象（产品）
- （2）获取（或者生成）针对被测试对象的最新测试程序（该测试程序需要事先编写好）
- （3）运行所有的测试程序
- （4）比较测试程序运行输出的结果，如果输出结果和预先给的标准输出不一致，记录测试程序的名称
- （5）将发现问题的测试程序名输出

一个参加测试构建的测试程序除了具有一般应用程序的特点之外，还必须具备以下三个特征：

- 测试程序需要提前被编写

- 测试程序在执行时不需要用户干预。在执行过程中，不应要求用户输入信息后再继续运行，否则就做不到自动化。考虑到这个特殊性，我们将用户的输入信息用文件形式来替代，这个文件定义为标准的输入文件
- 测试程序在执行之前，需要预先给定一个标准的输出文件。这个文件内容就是在被测试对象没有缺陷的情况下测试程序正常运行后的输出结果，就类似一个标准答案，有了标准答案才能比较，才能知道每次测试程序运行后的输出结果是否正确。
- 每个测试程序的运行结果都必需输出到一个文件中。一般程序的运行结果是直接输出到屏幕的，这导致不能实现测试的自动化，我们定义这个记录输出结果信息的文件为测试程序运行的输出结果文件。

在本发明中，基于源代码版本控制工具结合项目管理系统和缺陷管理系统的测试构建过程的特点如下：

（1）测试构建过程可以通过脚本来控制，系统定时执行，整个测试构建过程不需要用户干预。

（2）每个参与测试构建的测试程序文件和标准输出文件需要提前被编写，有时可能还包括标准的输入文件。

（3）测试构建需要有一个对比过程，比较每个测试程序的实际输出和标准输出的区别。

（4）修复的缺陷可以在下一次测试构建过程中自动得到验证。

（5）每次测试构建发现的错误可以被及时记录到缺陷管理系统中，发现问题的测试程序不会继续参与下一次测试构建，保证每次测试构建都只会发现新错误。

（6）测试构建过程能够自动生成测试构建过程报告。

下面结合附图进一步说明对各应用范例进行详细说明：

1. 标准的测试构建过程模型

请参阅图 1 所示，测试构建的开始是通过一个可以定时执行的脚本来启动，也可以手动执行。在一些公知的操作系统中大部分都有系统定时器或者计划任务的功能，允许可以在某个特定时间激活一个可以执行的程序。

(1) 开始测试构建

(2) 获取最新的被测试对象（产品）

- 情况一：被测试对象是源代码

- 先获取最新的源代码

- 编译源代码，生成被测试对象

- 情况二：被测试对象是可以执行的文件

- 直接获取这些可执行文件

(3) 获取最新的测试程序及其相应的标准输入文件和标准输出文件

- 情况一：测试程序为源代码

- 获取最新的测试代码

- 编译测试代码，生成测试程序

- 情况二：测试程序可以直接执行

- 直接获取这些可执行的测试程序

(4) 运行测试程序，根据运行测试程序列表逐一运行测试程序，得到每个测试程序的输出结果

(5) 对比每个测试程序的输出和标准输出，如果不同，标识出错

(6) 测试构建过程完成，输出测试构建报告，报告记录了所有发现问题的原始出错测试程序列表。

2. 基于缺陷管理的测试构建过程模型

再请参阅图 2 所示，该测试构建过程包括以下步骤：

(1) 在生成测试程序清单之前，结合缺陷管理的测试构建过程和标准的构

建过程没有区别，都要执行清理获取最新的被测试对象、获取最新的测试程序的操作；

(2) 在获取了测试程序列表以后，基于缺陷管理的测试构建模型中增加了从缺陷管理系统中获取以前记录的运行出错测试程序列表（这个列表中的测试程序是每次测试构建过程自动累计加入到缺陷管理系统中的）的步骤。将前面的测试程序列表中剔除从缺陷管理系统中获取的运行出错测试程序列表中的测试程序，就生成了本次测试构建过程中必需的运行测试程序列表；

(3) 根据运行测试程序列表运行相应的测试程序，继续测试构建过程；

(4) 验证每个测试程序的输出，并将记录的输出结果文件与预先给的标准输出文件进行比较，将其中结果不一致的测试程序名称记录下来，生成原始出错测试程序列表；

(5) 将上述的原始出错测试程序列表导入到缺陷管理系统中；

(6) 在缺陷管理系统中，自动处理上述导入的信息，生成本次的测试构建报告。

进行了上述操作步骤，缺陷管理系统就记录了每次测试构建过程中运行出错的测试程序名称，这些测试程序将不会继续在下一次测试构建过程中参加运行，这样提高了测试构建的效率，也有利于测试构建过程发现新的缺陷。

通过将缺陷管理系统与测试构建过程的上述结合过程，每次发现缺陷的测试程序被缺陷管理系统记录下来，在下一次测试构建过程中，被记录的测试程序不参加运行，但缺陷管理系统还没有对缺陷进行有效管理。不过这个测试构建过程已经可以生成包括如下的内容的测试构建报告：

- 新发现问题的测试程序名称
- 出错测试程序数量
- 对比以前的数量，生成对比统计图

再请参阅图 3 所示，基于缺陷管理的测试构建过程模型中，在测试构建完成后，将原始出错测试程序列表导入到在缺陷管理系统中，而在缺陷管理系统

中，对于新导入的上述列表，统一进行如下处理：

(1) 将本次新发现的测试程序问题，缺陷管理系统在进行相应处理后给出提示信息，并将待处理的缺陷信息自动提交给构建负责人进行处理。这种提示信息包括如下几种情况：

- 如果这个测试程序在缺陷管理系统中没有记录，则提示信息为：
 - 可能是将其确认为一个新缺陷
 - 可能是一种意外错误
- 如果这个测试程序在缺陷管理系统中有记录
 - 如果缺陷修复的时间是在上次测试构建之后，则提示信息为：
 - ◆ 可能在上次测试构建以后修复的缺陷可能没有被真正修复，修复验证失败，该缺陷需要被重新设置为未修复状态
 - ◆ 可能此测试程序发现的缺陷为新的缺陷
 - 如果缺陷修复的时间是在上一次测试构建之前，则提示信息为：
 - ◆ 可能是一个已经发现的缺陷再次复发，并将与这个测试程序相关的缺陷内容提供给构建负责人
 - ◆ 可能此测试程序发现的缺陷为新的缺陷
- 对于在上一次测试构建之后修复的缺陷没有再次出现问题的测试程序（即没有再次出现在原始出错测试程序列表中的测试程序），系统将这些缺陷的名字提示给构建负责人，并提示说明这些缺陷已被成功修复。

(2) 构建负责人对如上的提示信息进行处理，可能会有以下几种操作：

- 提交新的缺陷（系统自动将此缺陷和发现缺陷的测试程序名称绑定）
- 将一个已经修复的缺陷状态设置为未修复状态。
- 将一个已经修复的缺陷状态设置为复发状态。

(3) 生成以下测试构建报告：

- 本次测试构建新发现缺陷数量、正常修复缺陷的数量和复发缺陷的数

量报告

- 新发现的缺陷名称报告
- 正常修复的缺陷名称报告
- 复发缺陷的名称报告
- 统计测试程序发现缺陷的数量，可以包括本次的和累计的报告

(4) 将测试构建结果输出，测试构建完成，这个输出可以同时采用如下两种方式：

- 发送的方式。它一般是通过 Email 将如上信息发送给开发人员，可以保证信息的及时传递
- 让开发人员取的方式，就是将一个固定的 URL 链接提供给开发人员访问，让他们来获取输出的结果信息（一般来说，这种方式是必需的，这次记录可以做为系统日志保留，也可以用于和以后的测试构建结果进行比较）

(5) 在下一次测试构建过程开始时，缺陷管理系统会产生新的运行出错测试程序列表，并将其输出至测试构建过程，该列表中的测试程序为：

- 所有发现缺陷但缺陷还没有被修复的测试程序
- 新导入的但还未得到用户处理的测试程序

该基于缺陷管理的测试构建过程模型中，其用户可以为构建负责人，也可以为缺陷管理系统中的其他用户。

3. 缺陷管理系统与项目管理系统结合的缺陷修复的自动验证构建过程模型

版本控制工具负责代码的管理，软件项目的重要工作内容是编写和更改产品的代码，代码之间的关系不是完全孤立的，一组（一个或者多个）源代码是用来实现或者完成一个功能，所以代码的添加或者更改往往是批量进行的，这样的更改对我们工作的记录是有意义的，所以我们在更改了版本控制工具上的代码的同时，将这些代码还记录在项目管理系统中，并要求提交者给出这批代

码提交的用途、完成什么样的功能等相关信息。如上的说明阐述了两件事情，一个记录可以是包括多个代码文件，另外一个代码的更改需要另外的系统记录。

在如上结合的基础上，进一步实现项目管理系统和缺陷管理系统的结合。首先，开发的最初阶段只是添加代码，这些代码是为了实现一个一个功能，但这些代码实现的功能可能存在不足，或者说已经添加的代码与预期的功能不相符，这就是所说的发现了缺陷，我们将这个缺陷记录到缺陷管理系统中，这时缺陷管理系统会针对该缺陷给定一个唯一的编号，并提示要实现这个功能的开发人员进行缺陷的修复。之后开发人员进行修复缺陷的过程其实还是对代码的操作，通过更改和删除原来代码，也可能是增加新的代码来完成。完成后的操作又是一组代码的提交（添加、更改和删除都是提交），这个提交又被记录在项目管理系统中，同时又要求提交人员记录这次更改，这次的记录方式和开始阶段的工作几乎相同，唯一的区别这次更改的目的是为了修复一个已经存在的缺陷，因为每个缺陷有唯一的编号，所以这次记录的提交会在提交的同时做另外一件事情，就是更改缺陷管理系统中这个编号所对应缺陷的状态，从未修复状态，转为修复状态。

再请参阅图 4 所示，根据缺陷修复的信息，对缺陷管理系统中相应的缺陷状态进行修改，需要包括以下步骤：

- （1）版本控制工具记录开发人员提交的进行缺陷修复的代码的文件名称信息；
- （2）项目管理系统同时也记录相应的开发人员提交的进行缺陷修复的代码的文件名称信息，并根据该信息建立缺陷修复记录；
- （3）项目管理系统将该缺陷修复记录与缺陷管理系统中的相应缺陷之间建立关联关系；
- （4）缺陷管理系统根据项目管理系统缺陷修复记录的关联信息将相应缺陷的状态设置为已修复状态。

按照上述步骤实现了项目管理系统和缺陷管理系统的结合后，记录提交就可以自动更改缺陷的状态，但这次更改是否有效，还需要对这此更改进行验证。而此时测试构建恰恰可以进行这个验证，测试构建在开始时先将修复的缺陷所对应的测试程序加入到本次测试构建中，在测试构建过程中检查这个测试程序的运行结果是否正确，如果正确，则提示信息为这次缺陷修复是正确的；如果不正确，提示信息为这此修复工作可能失败。这里说的提示信息只将这种情况作为最大的可能性提交给用户处理，而不是自动将上次修复的缺陷自动从修复状态转为未修复状态，因为我们要考虑到这个测试程序在这次测试构建中可能发现了新的错误，所以状态的更改是需要在用户确认后才能进行的。

4. 缺陷关联和项目管理系统结合的测试构建过程模型

测试一般分为两个部分，测试程序和被测试对象，对于来说测试程序一般是比较确定的，但测试对象大多数是不能确定的，通过如上某个缺陷的编号和某批提交的代码关联，加之测试程序与缺陷的关联，就实现了测试程序和某批提交的代码关联，也就实现了测试程序和被测试对象的关联。

再请参阅图 5 所示，在项目管理系统和缺陷管理系统之间建立测试程序与被测试对象的关联包括以下步骤：

(1) 缺陷管理系统根据该原始出错测试程序列表中的测试程序对应缺陷和以往发现过的该缺陷的历史信息，在项目管理系统中通过所关联的缺陷修复记录定位与该缺陷相关的代码文件；

(2) 在项目管理系统中根据该缺陷相关的代码文件查询上次测试构建以后的代码提交记录，确定相应的开发人员和记录；

(3) 如果该缺陷是首次出现，则建立该缺陷与所关联的代码文件之间的耦合度因子；如果该缺陷是重复出现的，则增加该缺陷与所关联的代码文件之间的耦合度因子的值。

按照上述步骤建立的关联，可能这个关联不是非常清晰的一对一关系，但

可以做为一种依据。当一个测试程序在新的测试构建中再次发现问题，并被记录到缺陷管理系统时，缺陷管理系统会根据这个缺陷以往发现过缺陷的历史，定位与之相关的代码文件，在项目管理系统中查询上次测试构建以后的代码提交记录，就可以初步确定缺陷引入的人员和记录，从而可以自动给出是谁在哪一次的代码提交时引入这个缺陷的提示。另外每次的确认过程可以用来更改耦合度因子，在某个缺陷重复出现时增加相应的耦合度因子的值，使得重新出现的缺陷能自动的在项目管理系统中被标识和定位。

此过程模型将项目管理和缺陷管理结合起来，使得项目在开发中的代码记录在版本控制工具中，并同时在项目系统中记录这些代码的名字，在项目管理系统中通过代码的名字就可以自动访问代码在版本控制工具上的源文件。项目管理和缺陷管理的结合，可以将项目中的缺陷记录在缺陷管理系统中，在项目管理系统中对与某个缺陷修复提交的记录就会自动转化为修复缺陷系统的特定缺陷记录，并且这个过程可以自动完成的。此外，由于这个缺陷是在测试构建过程中的某个测试程序发现的，所以这种结合就可以将测试程序和测试对象（修复缺陷的一组代码）结合起来。

实现如上的测试构建过程的方法，除了项目管理系统和缺陷管理系统结合，还需要一些脚本文件的支持：

- 从版本控制工具中获取最新的代码执行脚本
- 自动编译脚本
- 从缺陷系统中获取不参加本次构建的测试程序清单执行脚本
- 将构建后的测试程序清单和构建日志写入到缺陷管理系统执行脚本
- 自动处理缺陷系统中的信息并进行与前一次构建对比汇总脚本
- 自动 Email 发送脚本

采用本发明的基于缺陷管理的自动化测试构建过程方法具有如下的优点：

1. 统一代码源

保证整个开发团队共享统一的代码源，将所有人的代码都归总到一起，这是测试构建的基础，缺陷管理和项目管理也都是以源代码管理为基础，项目管理系统中的代码文件名和缺陷管理系统中的测试文件名都可以自动链接到版本控制工具的对应文件上。

2. 有利于集成测试

测试应该全部执行完毕，而不是遇到测试运行的错误就放弃测试过程。测试结果中包括有成功的测试信息、失败的测试信息和失败的测试的细节信息。最后的结果将通过某种方式通知给相应的开发人员，要求他们修改设计或测试（如果是测试本身的问题的话）。

3. 提高发现缺陷的效率和工作效率

如果测试构建是每日执行（日构建），可以在最快的时间发现当日提交的代码更改产生的缺陷，早发现早清除，可以大大提高产品的效率。另外修复的缺陷可以在下一次测试构建过程中自动得到验证，免去了验证过程的工作量。

4. 保证执行过程的准确性

整个测试构建过程全部是自动进行的，没有人为干预，充分保证了执行过程的可靠性和准确性。同时缺陷管理和项目管理结合，可以测试程序和测试对象进行自动定位。

5. 对比每次构建结果，保证产品质量

通过与缺陷管理系统的结合，可以自动记录每次测试构建中发现的问题，而缺陷管理系统可以对此进行分析和统计，通过柱形图，坐标图等方式将结果显示出来，为产品的质量提供可供参考的依据。

在此说明书中，本发明已参照其特定的实施例作了描述。但是，很显然仍可以作出各种修改和变换而不背离本发明的精神和范围。因此，说明书和附图应被认为是说明性的而非限制性的。

说明书附图

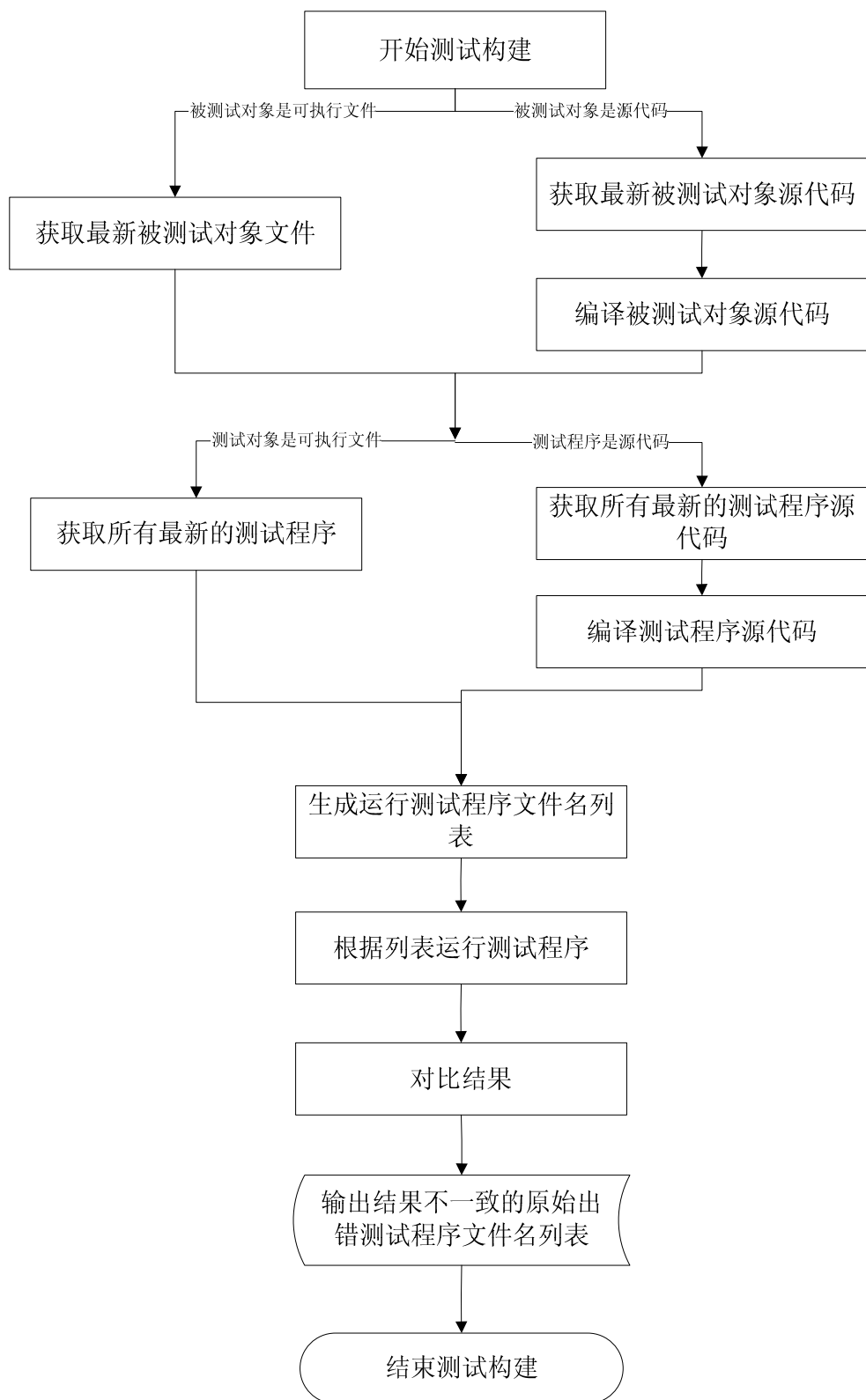


图 1

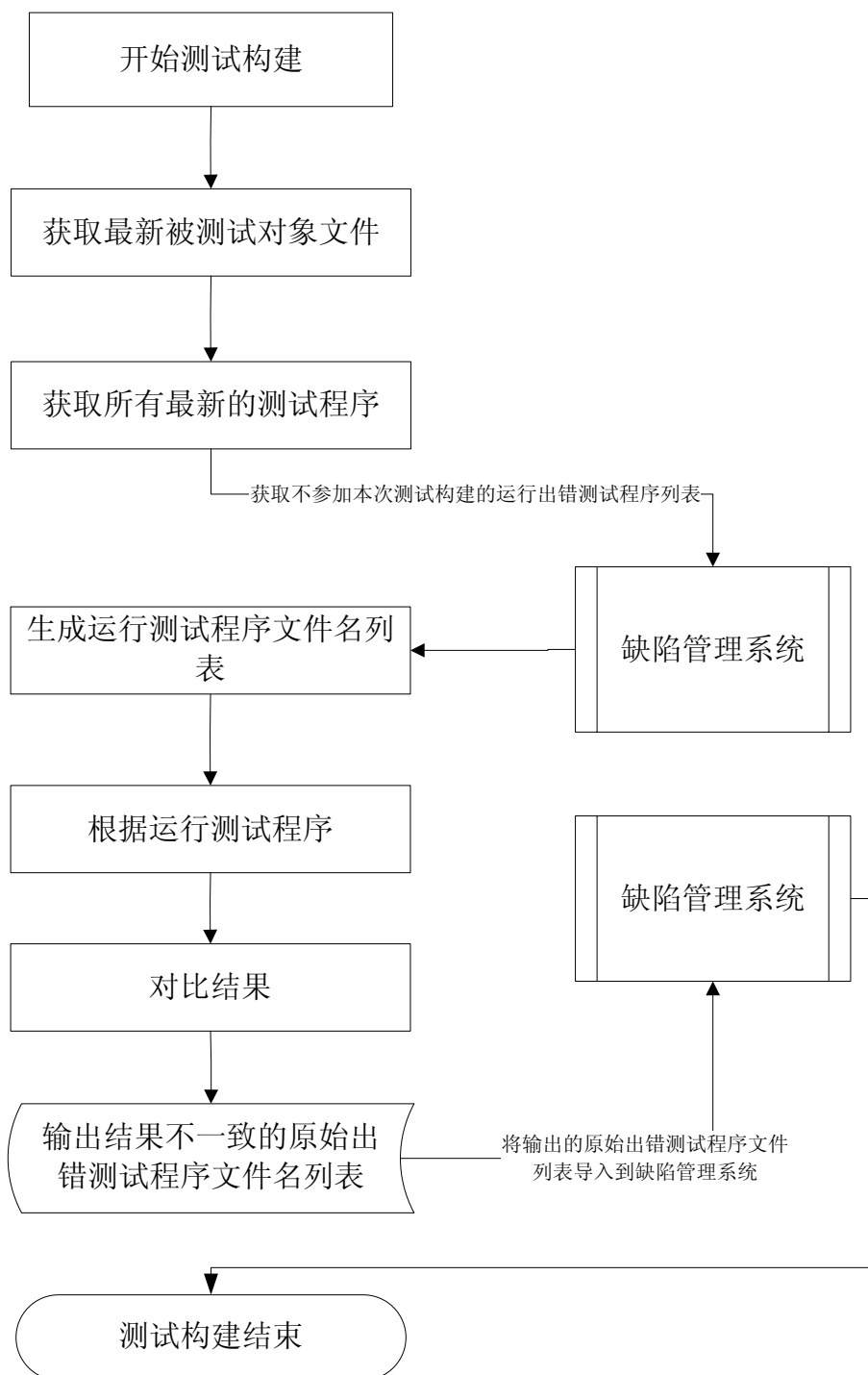


图 2

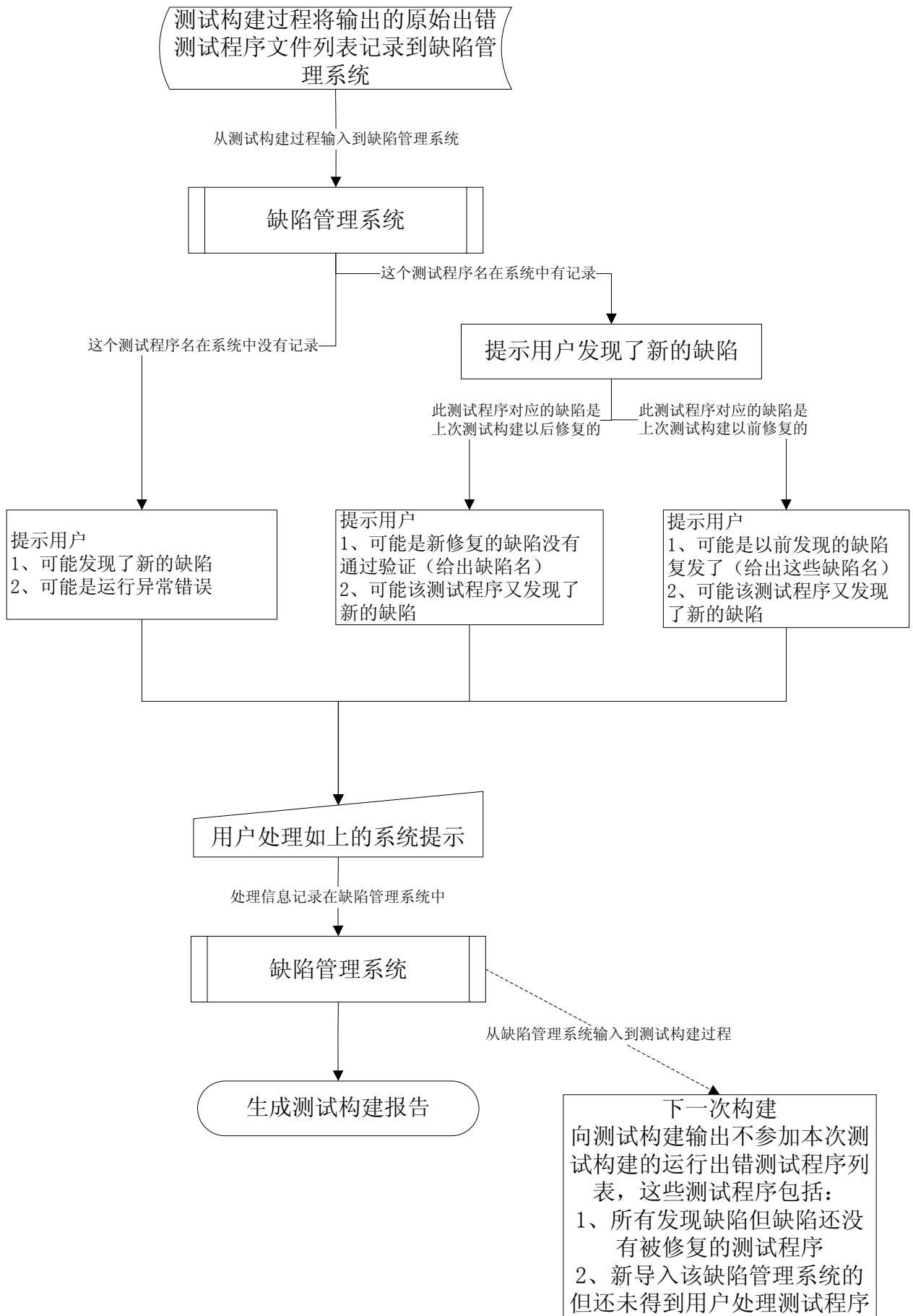


图 3

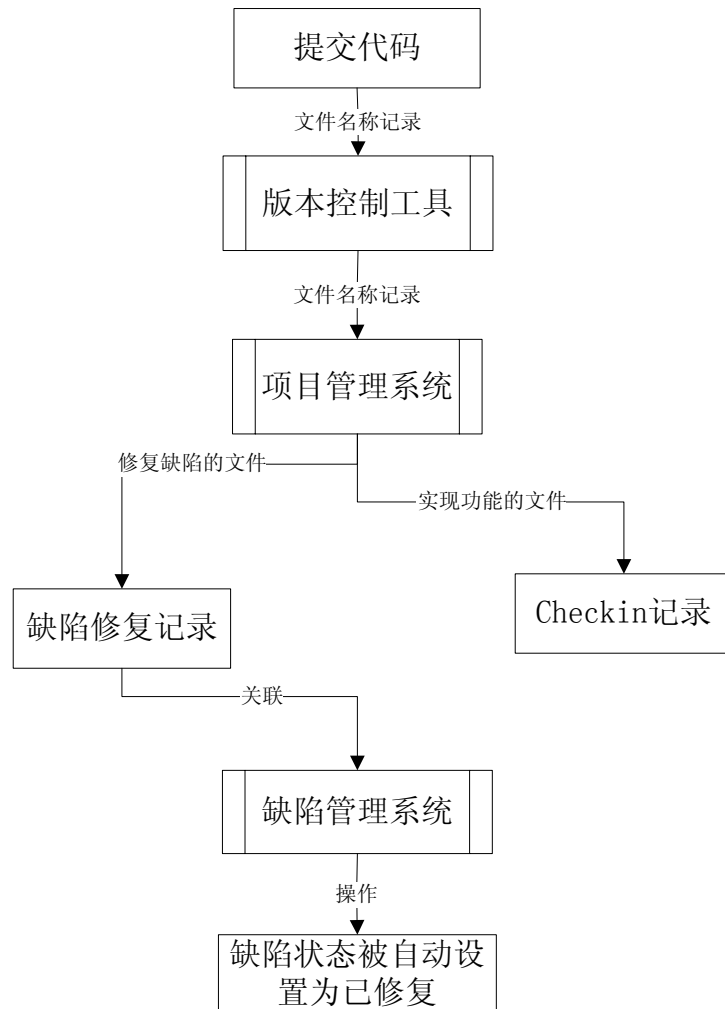


图 4

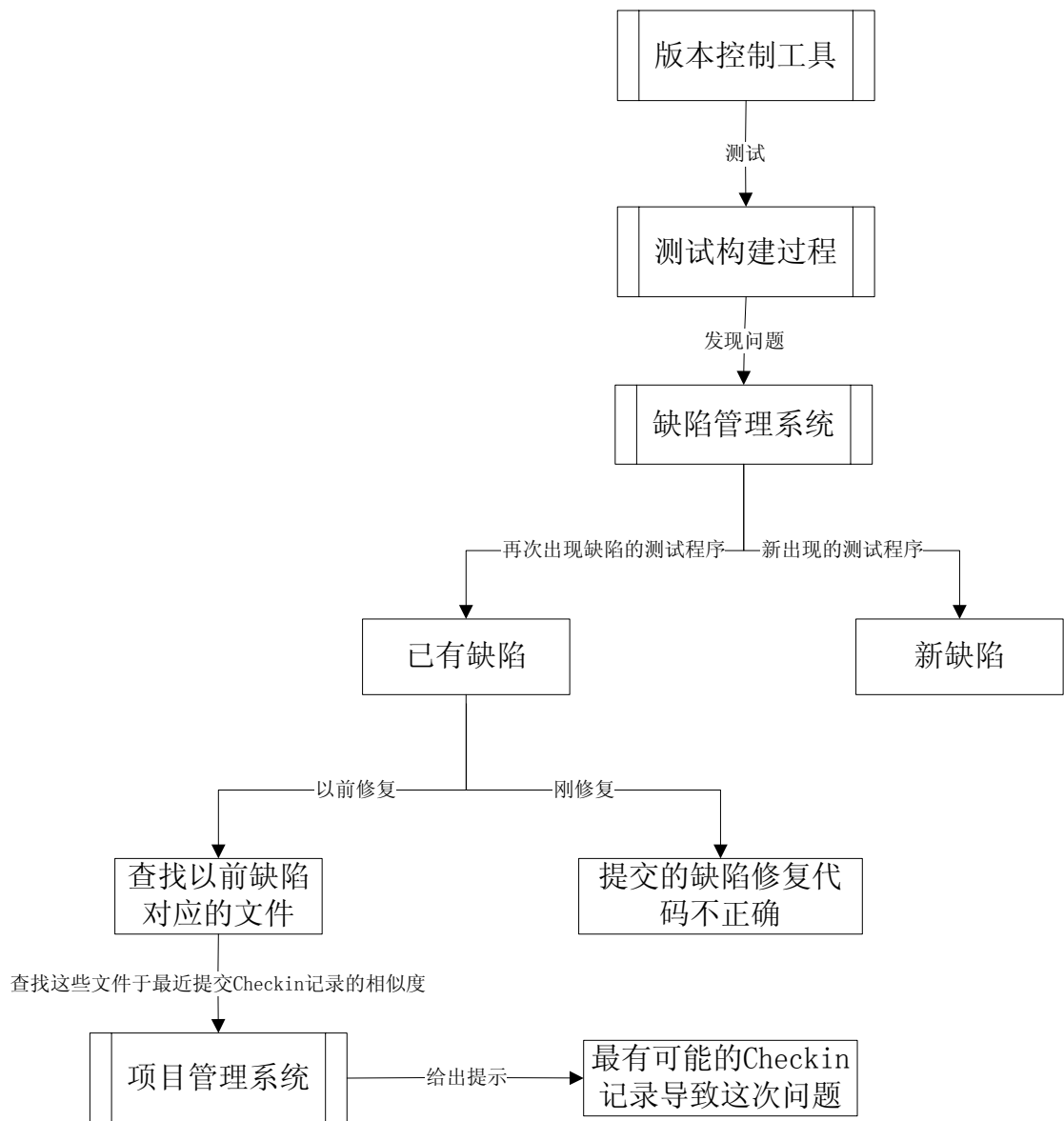


图 5

说明书摘要

本发明涉及一种自动化测试构建方法，其中，该方法包括：得到最新的被测测试对象、得到最新测试程序、生成运行测试程序列表、运行相应的测试程序得到运行输出结果、生成原始出错测试程序列表、将该出错测试程序列表导入缺陷管理系统、生成测试构建报告并将测试结果输出、对相应的缺陷状态进行修改、在下一次测试构建过程开始时生成新的运行出错测试程序列表的步骤。采用以上步骤的自动化测试构建方法，使得缺陷的发现和修复后的验证都能自动完成，提高了开发和测试的效率，保证了一个开发团队共享统一的代码源，使得集成测试的效率大大提高，保证了执行过程的可靠性和缺陷定位的准确性，同时为产品的质量提供可供参考的依据。

摘要附图

