



(12) 发明专利申请

(10) 申请公布号 CN 115543435 A

(43) 申请公布日 2022. 12. 30

(21) 申请号 202211375660.9

(22) 申请日 2022.11.04

(71) 申请人 创优数字科技(广东)有限公司

地址 510000 广东省广州市海珠区新港东路2429号首层自编011房

(72) 发明人 刘钰霖 马识佳

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

专利代理师 周伟

(51) Int. Cl.

G06F 8/72 (2018.01)

G06F 8/36 (2018.01)

G06F 8/30 (2018.01)

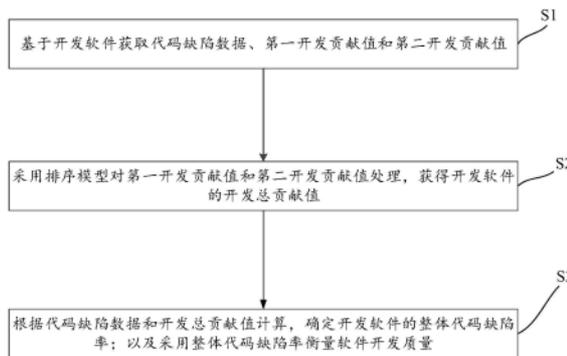
权利要求书2页 说明书7页 附图1页

(54) 发明名称

基于开发贡献值的软件开发质量衡量方法、装置及设备

(57) 摘要

本申请涉及一种基于开发贡献值的软件开发质量衡量方法、装置及设备,该方法包括基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;采用排序模型对第一开发贡献值和第二开发贡献值处理,获得开发软件的开发总贡献值;根据代码缺陷数据和开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用整体代码缺陷率衡量软件开发质量。该基于开发贡献值的软件开发质量衡量方法通过排序模型将开发软件代码的第一开发贡献值和第二开发贡献值组合获得开发总贡献值,再采用代码缺陷数据与开发总贡献值的比例作为整体代码缺陷率,从而更准确地衡量软件开发的质量,解决了现有采用千行代码缺陷率衡量软件开发的质量是不准确的技术问题。



1. 一种基于开发贡献值的软件开发质量衡量方法,其特征在于,包括以下步骤:

基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;

采用排序模型对所述第一开发贡献值和所述第二开发贡献值处理,获得开发软件的开发总贡献值;

根据所述代码缺陷数据和所述开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用所述整体代码缺陷率衡量软件开发质量。

2. 根据权利要求1所述的基于开发贡献值的软件开发质量衡量方法,其特征在于,采用排序模型对所述第一开发贡献值和所述第二开发贡献值处理,获得开发软件的开发总贡献值包括:根据所述第一开发贡献值和所述第二开发贡献值采用排序模型的组合函数计算,得到开发总贡献值。

3. 根据权利要求2所述的基于开发贡献值的软件开发质量衡量方法,其特征在于,所述组合函数为:

$$V = w^T \begin{bmatrix} d \\ t \end{bmatrix}$$

式中, V 为开发总贡献值, w 为排序模型的权重向量, d 为第一开发贡献值, t 为第二开发贡献值。

4. 根据权利要求1所述的基于开发贡献值的软件开发质量衡量方法,其特征在于,基于开发软件获取第一开发贡献值包括:将所述开发软件中函数被调用次数的量化数值作为第一开发贡献值。

5. 根据权利要求1所述的基于开发贡献值的软件开发质量衡量方法,其特征在于,基于开发软件获取第二开发贡献值包括:

获取所述开发软件的提交代码,并对所述提交代码进行分类,得到分类代码;

获取每个所述分类代码基于对开发软件提交的影响价值,根据与所述分类代码对应的所有影响价值计算,得到第二开发贡献值。

6. 一种基于开发贡献值的软件开发质量衡量装置,其特征在于,包括数据获取模块、数据处理模块和计算衡量模块;

所述数据获取模块,用于基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;

所述数据处理模块,用于采用排序模型对所述第一开发贡献值和所述第二开发贡献值处理,获得开发软件的开发总贡献值;

所述计算衡量模块,用于根据所述代码缺陷数据和所述开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用所述整体代码缺陷率衡量软件开发质量。

7. 根据权利要求6所述的基于开发贡献值的软件开发质量衡量装置,其特征在于,所述数据处理模块还用于根据所述第一开发贡献值和所述第二开发贡献值采用排序模型的组合函数计算,得到开发总贡献值。

8. 根据权利要求7所述的基于开发贡献值的软件开发质量衡量装置,其特征在于,所述组合函数为:

$$V = w^T \begin{bmatrix} d \\ t \end{bmatrix}$$

式中, V 为开发总贡献值, w 为排序模型的权重向量, d 为第一开发贡献值, t 为第二开发贡献值。

9. 一种存储装置, 其中存储有多条程序代码, 其特征在于, 所述程序代码适于由处理器加载并运行以执行如权利要求1-5任意一项所述的基于开发贡献值的软件开发质量衡量方法。

10. 一种终端设备, 其特征在于, 包括处理器以及存储器;

所述存储器, 用于存储程序代码, 并将所述程序代码传输给所述处理器;

所述处理器, 用于根据所述程序代码中的指令执行如权利要求1-5任意一项所述的基于开发贡献值的软件开发质量衡量方法。

基于开发贡献值的软件开发质量衡量方法、装置及设备

技术领域

[0001] 本申请涉及软件开发技术领域,尤其涉及一种基于开发贡献值的软件开发质量衡量方法、装置及设备。

背景技术

[0002] 软件开发的过程或者技术团队的管理存在着如何去合理的度量效率的问题,软件开发最直观的产出就是一行行代码,目前统计软件研发质量的常用评估指标为千行代码缺陷率指标,用代码行数反映代码工作量。而千行代码缺陷率的计算公式为千行代码缺陷率=缺陷行数数量/(代码行数/1000),目前统计软件研发质量的度量的标准是千行代码缺陷率数值越小质量越好。

[0003] 但是在实际上,软件代码行数的多少并不能准确地代表开发人员的代码贡献价值,代码行数容易受到编程习惯或特定代码行为的干扰(如换行、空行、注释),可能导致出现大量的复制和冗余,不能反映代码开发所涉及的实际贡献量,因此仅仅采用千行代码缺陷率衡量软件开发的质量是不准确的。

发明内容

[0004] 本申请实施例提供了一种基于开发贡献值的软件开发质量衡量方法、装置及设备,用于解决现有采用千行代码缺陷率衡量软件开发的质量是不准确的技术问题。

[0005] 为了实现上述目的,本申请实施例提供如下技术方案:

[0006] 一种基于开发贡献值的软件开发质量衡量方法,包括以下步骤:

[0007] 基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;

[0008] 采用排序模型对所述第一开发贡献值和所述第二开发贡献值处理,获得开发软件的开发总贡献值;

[0009] 根据所述代码缺陷数据和所述开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用所述整体代码缺陷率衡量软件开发质量。

[0010] 优选地,采用排序模型对所述第一开发贡献值和所述第二开发贡献值处理,获得开发软件的开发总贡献值包括:根据所述第一开发贡献值和所述第二开发贡献值采用排序模型的组合函数计算,得到开发总贡献值。

[0011] 优选地,所述组合函数为:

$$[0012] \quad V = w^T \begin{bmatrix} d \\ t \end{bmatrix}$$

[0013] 式中, V 为开发总贡献值, w 为排序模型的权重向量, d 为第一开发贡献值, t 为第二开发贡献值。

[0014] 优选地,基于开发软件获取第一开发贡献值包括:将所述开发软件中函数被调用次数的量化数值作为第一开发贡献值。

[0015] 优选地,基于开发软件获取第二开发贡献值包括:

[0016] 获取所述开发软件的提交代码,并对所述提交代码进行分类,得到分类代码;
 [0017] 获取每个所述分类代码基于对开发软件提交的影响价值,根据与所述分类代码对应的所有影响价值计算,得到第二开发贡献值。

[0018] 本申请还提供一种基于开发贡献值的软件开发质量衡量装置,包括数据获取模块、数据处理模块和计算衡量模块;

[0019] 所述数据获取模块,用于基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;

[0020] 所述数据处理模块,用于采用排序模型对所述第一开发贡献值和所述第二开发贡献值处理,获得开发软件的开发总贡献值;

[0021] 所述计算衡量模块,用于根据所述代码缺陷数据和所述开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用所述整体代码缺陷率衡量软件开发质量。

[0022] 优选地,所述数据处理模块还用于根据所述第一开发贡献值和所述第二开发贡献值采用排序模型的组合函数计算,得到开发总贡献值。

[0023] 优选地,所述组合函数为:

$$[0024] \quad V = w^T \begin{bmatrix} d \\ t \end{bmatrix}$$

[0025] 式中, V 为开发总贡献值, w 为排序模型的权重向量, d 为第一开发贡献值, t 为第二开发贡献值。

[0026] 本申请还提供一种存储装置,其中存储有多条程序代码,所述程序代码适于由处理器加载并运行以执行上述所述的基于开发贡献值的软件开发质量衡量方法。

[0027] 本申请还提供一种终端设备,包括处理器以及存储器;

[0028] 所述存储器,用于存储程序代码,并将所述程序代码传输给所述处理器;

[0029] 所述处理器,用于根据所述程序代码中的指令执行上述所述的基于开发贡献值的软件开发质量衡量方法。

[0030] 从以上技术方案可以看出,本申请实施例具有以下优点:该基于开发贡献值的软件开发质量衡量方法、装置及设备,该方法包括基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;采用排序模型对第一开发贡献值和第二开发贡献值处理,获得开发软件的开发总贡献值;根据代码缺陷数据和开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用整体代码缺陷率衡量软件开发质量。该基于开发贡献值的软件开发质量衡量方法通过排序模型将开发软件代码的第一开发贡献值和第二开发贡献值组合获得开发总贡献值,再采用代码缺陷数据与开发总贡献值的比例作为整体代码缺陷率,从而更准确地衡量软件开发的质量,解决了现有采用千行代码缺陷率衡量软件开发的质量是不准确的技术问题。

附图说明

[0031] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术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它的附图。

[0032] 图1为本申请实施例所述的基于开发贡献值的软件开发质量衡量方法的步骤流程图；

[0033] 图2为本申请实施例所述的基于开发贡献值的软件开发质量衡量装置的框架图。

具体实施方式

[0034] 为使得本申请的发明目的、特征、优点能够更加的明显和易懂，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，下面所描述的实施例仅仅是本申请一部分实施例，而非全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例，都属于本申请保护的范围。

[0035] 本申请实施例提供了一种基于开发贡献值的软件开发质量衡量方法、装置及设备，用于解决了现有采用千行代码缺陷率衡量软件开发的质量是不准确的技术问题。

[0036] 实施例一：

[0037] 图1为本申请实施例所述的基于开发贡献值的软件开发质量衡量方法的步骤流程图。

[0038] 如图1所示，本申请实施例提供了一种基于开发贡献值的软件开发质量衡量方法，包括以下步骤：

[0039] S10. 基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值。

[0040] 需要说明的是，在步骤S10主要是获取开发软件的代码缺陷数据、第一开发贡献值和第二开发贡献值。

[0041] 进一步地，基于开发软件获取第一开发贡献值包括：将开发软件中函数被调用次数的量化数值作为第一开发贡献值。

[0042] 需要说明的是，获取第一开发贡献值指的是获取代码的复用价值，代码的复用价值通过在开发软件中函数的被调用次数量化作为第一开发贡献值，函数被调用的次数越多，节省的调用者的开发工作越多，因而具有较高的价值。

[0043] 进一步地，基于开发软件获取第二开发贡献值包括：

[0044] 获取开发软件的提交代码，并对提交代码进行分类，得到分类代码；

[0045] 获取每个分类代码基于对开发软件提交的影响价值，根据与分类代码对应的所有影响价值计算，得到第二开发贡献值。

[0046] 需要说明的是，根据与分类代码对应的所有影响价值相加计算，得到第二开发贡献值。获取第二开发贡献值指的是代码提交的影响价值，通过对代码提交对开发软件的影响进行分类来判断代码的价值，代码的价值规则为：修复构建错误的价值>修复严重的非构建错误的价值>重要的新功能的价值>修复严重的风险性错误的价值>修复轻微错误的价值>常规新功能的价值>体验优化的价值>外观错误的价值>源代码卫生的价值。在本实施例中，并不是所有的开发价值都体现在代码结构中，代码提交的贡献对整个开发软件也有非结构性的影响，通过对提交代码的影响进行分类来判断代码的价值。例如，修复bug、新增功能或维护文档。因此，提交代码的影响类型是确定开发贡献值的一个重要特性。每个提交代码的价值是根据实际项目建立提交代码的类别与代码价值的对应关系，例如bug修复这个

类别的价值得分为0.8分,文档更新这个类别为0.2分。而提交代码的类别与代码价值的对应关系也能够根据实际项目进行修改。

[0047] 在本申请实施例中,可以按照约定式提交规范提交代码,提交代码的分类包括:新增功能feat、bug修复fix、文档更新docs、不影响程序逻辑的代码修改style、重构代码refactor、新增测试用例或是更新现有测试test、性能和体验优化perf、修改项目构建系统build、修改项目继续集成流程ci和某个更早之前的提交revert。

[0048] 需要说明的是,代码修改包括修改空白字符、格式缩进、补全缺失的分号等,没有改变代码逻辑。重构代码指的是既没有新增功能,也没有修复bug。修改项目构建系统build包括glup、webpack、rollup的配置等提交。修改项目继续集成流程ci包括Travis CI、Jenkins、GitLab CI、Circle等提交。提交代码的分类包括不属于以上类型的其他类chore,比如构建流程,依赖管理。gulp是前端开发过程中一种基于流的代码构建工具,是自动化项目的构建利器。webpack是一个现代JavaScript应用程序的静态模块打包工具。Rollup是一款ES Modules打包器,它也可以将项目中散落的细小模块打包为整块代码,从而使得这些划分的模块可以更好地运行在浏览器环境或者Node.js环境。Travis CI是在软件开发领域中的一个在线的,分布式的持续集成服务,用来构建及测试的代码。Jenkins是一款由Java编写的开源的持续集成工具。Gitlab CI是一个功能非常强大的持续集成系统,有很多不同的功能,而且每次发布都会增加新的功能。Circle指的是功能。

[0049] S20.采用排序模型对第一开发贡献值和第二开发贡献值处理,获得开发软件的开发总贡献值。

[0050] 需要说明的是,在步骤S20中根据第一开发贡献值和第二开发贡献值采用排序模型计算得到开发软件的开发总贡献值。

[0051] 进一步地,采用排序模型对第一开发贡献值和第二开发贡献值处理,获得开发软件的开发总贡献值包括:根据第一开发贡献值和第二开发贡献值采用排序模型的组合函数计算,得到开发总贡献值。组合函数为:

$$[0052] \quad V = w^T \begin{bmatrix} d \\ t \end{bmatrix}$$

[0053] 式中,V为开发总贡献值,w为排序模型的权重向量,d为第一开发贡献值,t为第二开发贡献值。

[0054] 需要说明的是,w是排序模型的权重向量,在排序模型L2R中,将第一开发贡献值和第二开发贡献值作为排序模型L2R的输入特征,通过排序模型L2R对第一开发贡献值和第二开发贡献值训练输出得到开发总贡献值,实现开发软件中复用价值和提交的影响价值的组合。在本实施例中,排序模型L2R是本领域成熟的技术,此处不再说明排序模型L2R对第一开发贡献值和第二开发贡献值训练处理输出开发总贡献值的具体内容。

[0055] S30.根据代码缺陷数据和开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用整体代码缺陷率衡量软件开发质量。

[0056] 需要说明的是,在步骤S30中将代码缺陷数据和开发总贡献值采用计算公式,得到开发软件的整体代码缺陷率。计算公式为整体代码缺陷率=(代码缺陷数据÷开发总贡献值)×100%。在本实施例中,整体代码缺陷率越低,开发软件的开发质量越高。

[0057] 本申请提供了一种基于开发贡献值的软件开发质量衡量方法,该方法包括基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;采用排序模型对第一开发贡献值和第二开发贡献值处理,获得开发软件的开发总贡献值;根据代码缺陷数据和开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用整体代码缺陷率衡量软件开发质量。该基于开发贡献值的软件开发质量衡量方法通过排序模型将开发软件代码的第一开发贡献值和第二开发贡献值组合获得开发总贡献值,再采用代码缺陷数据与开发总贡献值的比例作为整体代码缺陷率,从而更准确地衡量软件开发的质量,解决了现有采用千行代码缺陷率衡量软件开发的质量是不准确的技术问题。

[0058] 需要说明的是,该基于开发贡献值的软件开发质量衡量方法通过改进现有计算代码缺陷率计算方式,提高整体代码缺陷率计算的准确性,从而实现更准确地衡量开发质量。

[0059] 实施例二:

[0060] 图2为本申请实施例的基于开发贡献值的软件开发质量衡量装置的框架图。

[0061] 如图2所示,本申请实施例还提供一种基于开发贡献值的软件开发质量衡量装置,包括数据获取模块100、数据处理模块200和计算衡量模块300;

[0062] 数据获取模块100,用于基于开发软件获取代码缺陷数据、第一开发贡献值和第二开发贡献值;

[0063] 数据处理模块200,用于采用排序模型对第一开发贡献值和第二开发贡献值处理,获得开发软件的开发总贡献值;

[0064] 计算衡量模块300,用于根据代码缺陷数据和开发总贡献值计算,确定开发软件的整体代码缺陷率;以及采用整体代码缺陷率衡量软件开发质量。

[0065] 在本申请实施例中,数据处理模块200还用于根据第一开发贡献值和第二开发贡献值采用排序模型的组合函数计算,得到开发总贡献值。

[0066] 在本申请实施例中,组合函数为:

$$[0067] \quad V = w^T \begin{bmatrix} d \\ t \end{bmatrix}$$

[0068] 式中, V 为开发总贡献值, w 为排序模型的权重向量, d 为第一开发贡献值, t 为第二开发贡献值。

[0069] 需要说明的是,实施例二装置中模块对应于实施例一方法中的步骤,该基于开发贡献值的软件开发质量衡量方法的内容已在实施例一中详细阐述了,在此实施例二中不再对装置中模块的内容进行详细阐述。

[0070] 实施例三:

[0071] 本申请实施例提供了一种存储装置,其中存储有多条程序代码,程序代码适于由处理器加载并运行以执行上述的基于开发贡献值的软件开发质量衡量方法。

[0072] 实施例四:

[0073] 本申请实施例提供了一种终端设备,包括处理器以及存储器;

[0074] 存储器,用于存储程序代码,并将程序代码传输给处理器;

[0075] 处理器,用于根据程序代码中的指令执行上述的基于开发贡献值的软件开发质量衡量方法。

[0076] 需要说明的是,处理器用于根据所程序代码中的指令执行上述的一种基于开发贡献值的软件开发质量衡量方法实施例中的步骤。或者,处理器执行计算机程序时实现上述各系统/装置实施例各模块/单元的功能。

[0077] 示例性的,计算机程序可以被分割成一个或多个模块/单元,一个或者多个模块/单元被存储在存储器中,并由处理器执行,以完成本申请。一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述计算机程序在终端设备中的执行过程。

[0078] 终端设备可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。终端设备可包括,但不仅限于,处理器、存储器。本领域技术人员可以理解,并不构成对终端设备的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如终端设备还可以包括输入输出设备、网络接入设备、总线等。

[0079] 所称处理器可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0080] 存储器可以是终端设备的内部存储单元,例如终端设备的硬盘或内存。存储器也可以是终端设备的外部存储设备,例如终端设备上配备的插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)等。进一步地,存储器还可以既包括终端设备的内部存储单元也包括外部存储设备。存储器用于存储计算机程序以及终端设备所需的其他程序和数据。存储器还可以用于暂时地存储已经输出或者将要输出的数据。

[0081] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0082] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,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,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0083] 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。

[0084] 另外,在本申请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。

[0085] 所述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,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请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一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0086] 以上所述,以上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。

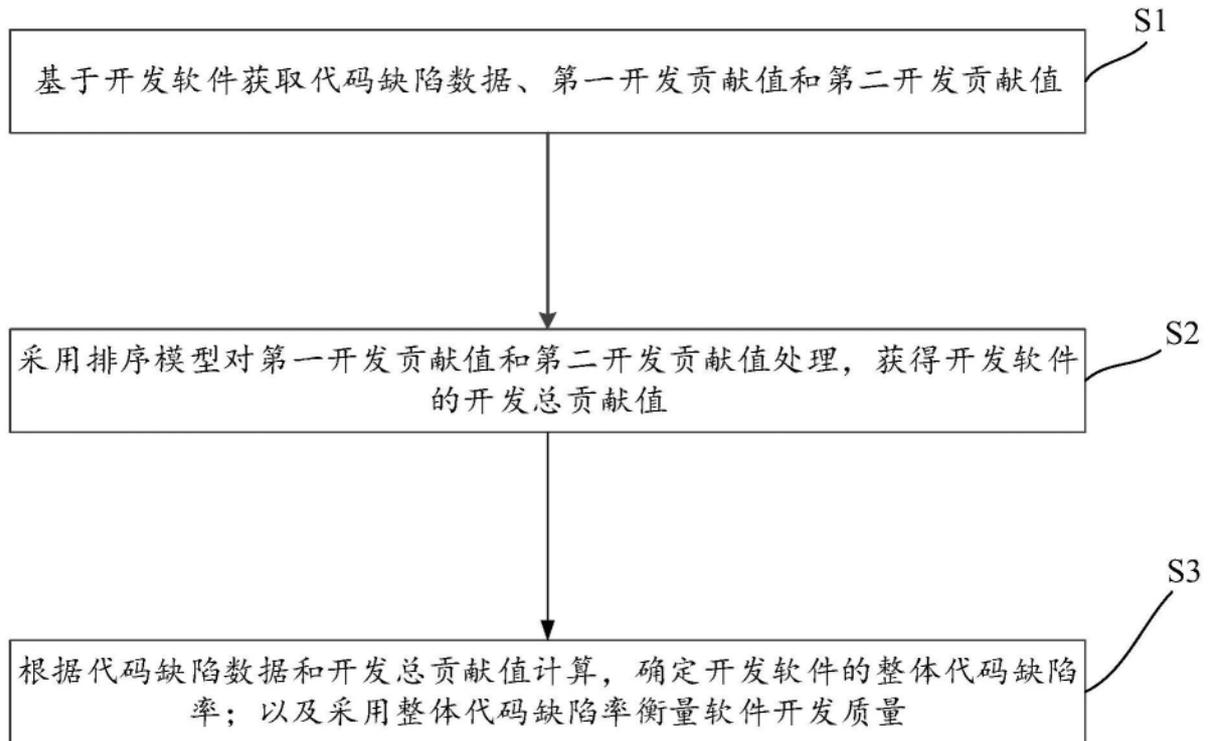


图1

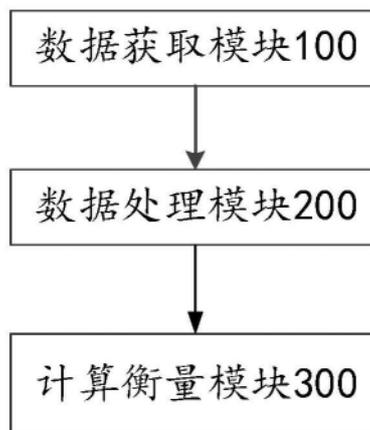


图2