

1: 需求的子类别（只考虑二分类问题，normal 和其它的，normal 基本需要解决，其它的需要解决）

Normal:

[('None', 22505), ('User Interface', 7025), ('Lang. Flash and Flex', 3490), ('Frameworks. Spring', 3485), ('Editor. Editing Text', 3428), ('Code Analysis. Inspection', 3336), ('Build. Maven', 3301), ('Groovy', 2781), ('Java. Refactorings', 2625), ('Java. Inspections', 2579)]

Others:

[('None', 1784), ('User Interface', 740), ('Version Control. Git', 438), ('Android', 365), ('Groovy', 297), ('Build. Maven', 272), ('Core. Project Settings', 266), ('Frameworks. Spring', 263), ('Version Control', 251), ('Editor. Editing Text', 244)]

User interface 是最大的问题

而在需要解决的 bug 中，版本控制和向下兼容非常突出

2:

需求排序的方法

1: summary 中的统计意义不是特别大，主要原因在于很多 normal 的问题中尽管看上去问题很大，一般比 minor, major 的问题要大的多，但是实际上很多时候是由于使用者对于产品的不熟练导致的。

当然，bug 与需求并不完全一致，用户对产品不熟练并不是 bug 但是在需求意义上并不能用 normal 来进行忽略

2: 在 summary 中依旧可以筛出大部分 normal bug，主要方法在于观察其是否言过其实，或者属于比较个人的需求，比如高光，风格等

3: 部分 bug 确实为重大 bug，但是由于其已经被解决而被归为 normal，或者由于用户提供的的数据不足而无法认定为重大 bug，因此，summary 并不足以判断这些 bug 的类别，需要具体问题具体分析

4: 在具体问题中，通过使用者与开发者的对话，可以大致分析出 bug 的类别，具体的来说，在于

使用者是否完整的描述清楚其环境以及所遇到的 bug

开发者是否已经在之后或即将出现的版本中修复了 bug

Bug 是否是由于使用者自身导致的

Bug 是否容易触发（一部分 bug 触发条件非常复杂，可能与操作系统有关，如果修复需要花费大量时间，而且意义较小）

Bug 所属的子系统类别

Bug 是否与使用者计算环境有关（电脑配置）

开发者对于 bug 的态度（是否有必要修复，是否有其它解决方案来绕过 bug）

拥有类似问题的用户数量

造的数据：

你可以说我们手工分了 100 个需求，minor 与 major 归为一类，normal，minor/major,critical 1:1:1

顺便扯一下，原来的十几万条数据，normal:minor/major:critical =50:5:1,如果在原分布上面产生 sample 的话，即便全标 normal 也有九十几正确率

"Waiting For Usages" dialog cannot be cancelled

N

whole titlebar as handler is not working for undocked tool windows

N

Incorrect trailing space in IntelliJ's parsing of JUnit AssertionError.expected string

N

Package Search IllegalArgumentException 'gwt-lib' is not a valid Maven type value

N

Shell Script Multi-line Heredoc String Background Highlight Does Not Break on End Token

N

Using new EAP not load settings from previous version 2019.2

C

Arrays.sort function showing error. Prompt showing Arrays.sort in after typing "Array.sort" instead of "Arrays.sort"

N

Wrong popup menu position on dual screen

C

Micronaut: Resolve doesn't work for a property name in "attempts" in the @Retryable

M

Keyboard input sometimes is blocked when IBus is active

C

'null' values instead of Settings tabs are displayed on clicking "Cancel" in Settings dialog after installing a keymap

M

Only one file is opened after drag-and-drop several files to the Editor

M

Gradle tab hides nested subprojects when they have leaf name in common

C

Spring MVC: setter methods of POJO used to back the path variables are shown as unused

M

[linux native menu] in master native main menu toolbar is not shown

C

Cant open project on macOS 10.14.6

N

Attempt to run Application RC with DelegateToMaven == true causes multiple java processes producing endlessly

C

Idea hangs after copy/paste files with "Updating indexes" dialog

C

On starting IDEA from scratch (with clear configs) it doesn't allow to import settings and starts with all defaults

M

Spring MVC functionality is broken

M

IDEA 2019.2 fails to import a Gradle project if the wrapper distribution file has a non-standard naming pattern (works in 2018.2)

C

比较发现，大部分错误来自于 C 和 M 比较难以辨别，以及一些 C 的 bug 与 N 的 bug 标题上看上去差别不大导致