LOC / # of predicted bugs

The Effort per bug

indicates the effort required to uncover one bug and is defined as the ratio of

LOC and the number of bugs.

以下三个量是在论文中提到一句，用来头脑风暴不错

maximum changeset

number of revisions

number of authors

stepwise linear regression algorithm

以下蓝色均为你的昨天给我的论文中关于输入量的参数，分为了两个部分，我都稍微解释了一下，个别没看懂的还望一起讨论

Source code metrics 源代码中度量

WMC Weighted method count 权重？

DIT Depth of inheritance 继承深度

NOC Number of children 子（文件）数量？

CBO Coupling between object classes 对象类之间耦合

RFC Response for a class 对类的响应

LCOM Lack of cohesion in methods

CE Efferent coupling

传入耦合（Afferent Coupling），也称作Fan In

传出耦合（Efferent Coupling），也称作Fan Out

不稳定性 = 传出耦合/（传出耦合 + 传入耦合）。值为1表示不稳定，值为0表示稳定。

NPM Number of public methods 公共方法的数量

CC Cyclomatic complexity is a popular procedural software metric equal to the

number of decisions that can be taken in a procedure

（The formulas for the cyclomatic complexity proposed by McCable are:

V(G)= e - n + 2p

Where e = the number of edges in the graph

n = the number of nodes in the graph

P = the number of connected components in the graph.

但是下面还有一句话

The complexity of a procedure p definer

as:cp=(fan-in\*fan-out)^2）

NOF Number of fields in the class 类的字段数

NOI Number of interfaces implemented by the class 通过类执行的界面（接口）数

LOC Number of lines of code in the file

NOM Number of methods in the class 类中的方法数

Fan-in Number of other types this class is using

Fan-out Number of other types using this class

PC Percentage of the file commented 注释率

Change code metrics 代码更改中度量

noOfBugs Number of bugs found and fixed during development 开发中的已找到和已修复bug

noOfStories Number of stories this file is part of 不懂？？

noOfSprints Number of sprints this file is part of 不懂，但是结合上一个感觉是指小改和大改的区别

revisionCount Number of commits this file is involved in 文件指派次数？（我的理解是被修改操作中包括的次数）

noOfAuthors Number of authors who worked on this file during the release 发布期间文件操作者数

LocA Number of lines of code added in total to this file during development 开发期间增加总代码行数

MaxLocA Max number of lines of code added among all commits to this file during development

AvgLocA Average number of lines of code added among all commits to this file during development

LocD Total number of lines of code deleted among all commits to this file during development

MaxLocD Max number of lines of code deleted among all commits to this file during development

AvgLocD Average number of lines of code deleted among all commits to this file during development

LocAD Total number of lines of code added-deleted among all commits to this file during development

MaxLocAD Max number of lines of code added-deleted among all commits to this file during development

AvgLocAD Average number of lines of code added-deleted among all commits to this file during development

一下红色字体为之前的论文中，我找到关于在软件测试时，可以作为我们的输出量的度量方式，仅供参考

Defect Density during Testing

。Error discovery rate: number of total defects found / number of test procedures

execution.

。Defect acceptance: (Number of valid defects / total number of defects) \* 100

。Test case defect density: (Number of failed tests / Number of executed test cases)\*100

Defect Arrival / removal During Testing:

。Bad Fix defect: defect whose resolution give rise to new defects are bad fix defect. Bad Fix

defect = (Number of Bad Fix defects / Total number of valid defects)\*100

International Journal of Software Engineering and Its Applications

。Defect removal effectiveness (DRE): (Defects removed during a development phase /

Defects latent in the product)\* 100. The denominator of the metric can only be

approximated by defects removed during the phase + defects found later.