软件质量保 证和管理

实验 1 质量工具实验

### 一 、实验目的

① 巩固所学的质量工具的相关知识和应用方法 。

② 掌握常见的质量管理工具数据分析方法 。

③ 学习如何对收集到的数据选择合适的质量工具进行分析 。

④ 质量管理工具的环境搭建和操作使用 。

### 二 、实验 前提与准备

① 理解各种质量工具的应用场景 。

② 质量工具选型 ：Mi n i Tab 工具安装包 、Excel 数据分析工具库 。

③ 提前准备好实验所需要的待分析的质量数据样例 。

### 三 、实 验内容

基于质量管理工具 ，对质量数据选择相应的质量工具 ，完成数据分析 。

·时序图 ：观察质量数据的变量随时间变化所呈现的趋势 。

·直方图 ：对质量数据进行分组 ，分析其分布情况 ，包括 ：分布区间 、平均值等。

． · 柏拉图 ：对汇总的质量数据按数据类型进行排序 ，确立主要质量因素 。

·散点阁 ：质量因素间的相关性分析 ，判断两个因素间是否存在某种关联 。

·回归分析：对有相关性的两种或两种以上变量的定量关系进行分析，得到回归方 程 ，以作出数据预测 。

### 四、实 验 坏 境

质量管理工具 ：Mi niTab 工具或 Excel 工具 。

Mi n iTab 工具的下载地址 ：h t t p :// www . mi n i tab. com/ zh-cn/ downloads ／，可以下载试 用版或早期版本网上的免费版 。

### 五 、实验过程

1) 质量工具环境构建和熟悉

* MiniTa b 工具

工具的安装配置 ，具体可以参见相关的工具安装指导 。实验中用到的工具主要位于图 形 、统计这两个菜单下。

* Excel 工具

时序图 、散点图等采用Excel 基本图表即可 ，直方图 、Pareto 图、回归分析采用Excel

的数据分析工具库 。数据分析库需要 手工加载 ，根据 Excel 版本不同 ，加载方法会 有所不同 ：

方法一 ：文件→选项→加载项→分析工具库 。

方法二 ：工具→加载宏→分析工具库→数据分析 。

加载完成后 ，在“数据”工具栏下出现相应的按钮 。

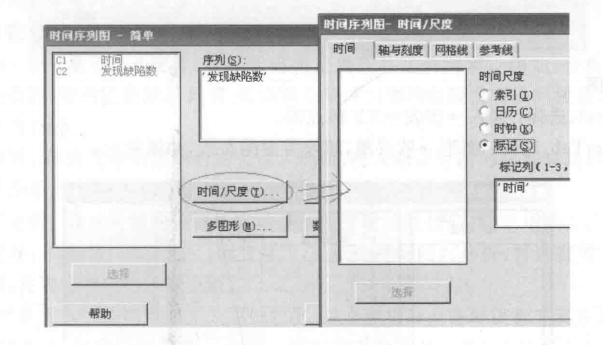
2 ） 质量工具应用

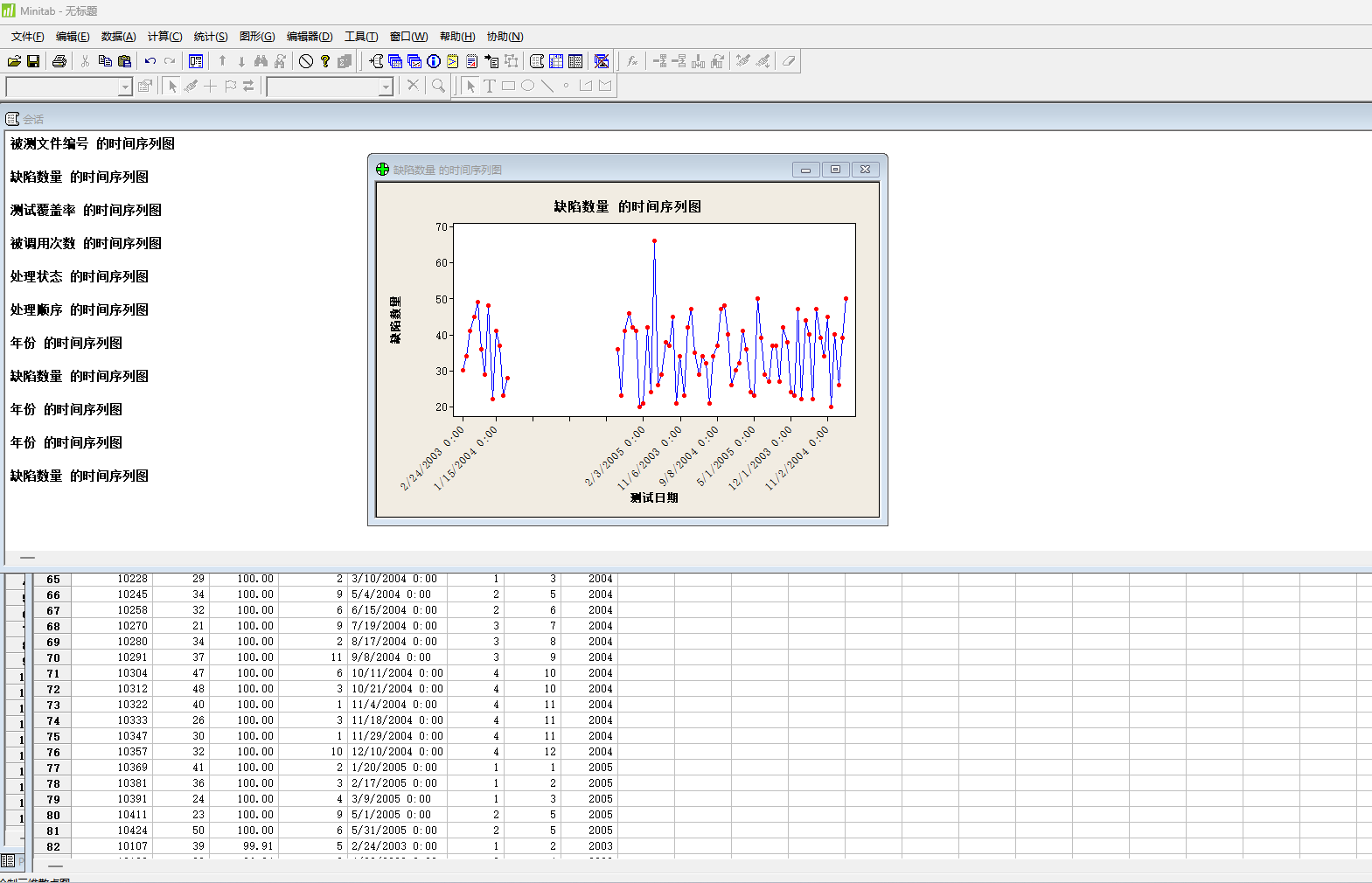
(1） 结合实验所准备的质量数据样例 ，熟悉质量工具的应用 。针对不同的质量工 具选取不同的质量数据 。

·时间序列图

对于 Excel ，选择 ：插入→图表→折线图 。

对于 Mini Tab ，选择 ：图形→时间序列图 ，填好需要的参数，如图所示 。





对时间序列图存在的缺陷的分析：

****数据点重复****：在2004年5月18日这一天，有多个记录显示被测文件编号为10251.0的缺陷数量为28.0。这种重复可能是由于数据导入错误或者在该日期进行了多次测试。

****高缺陷数量****：在2003年10月10日，被测文件编号为10159.0的缺陷数量达到49，这是迄今为止观察到的最高缺陷数量。这可能表明该版本软件的质量控制存在问题。

****测试覆盖率****：尽管在多个时间点测试覆盖率达到了100%，但仍然存在缺陷，这可能表明测试覆盖率高并不意味着软件无缺陷。

****缺陷数量波动****：从2003年到2005年，缺陷数量在不同时间点波动，这可能与软件发布周期、测试策略或测试资源分配有关。

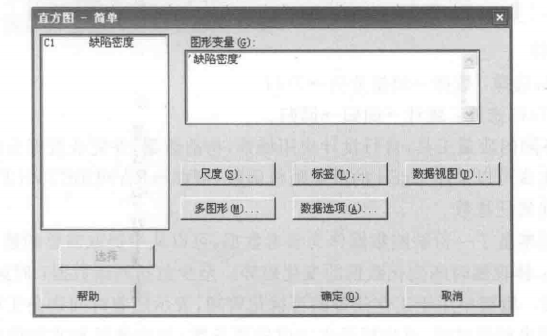
****时间序列的异常点****：例如，2005年3月3日的被测文件编号10388.0，测试覆盖率为76.36%，远低于其他时间点的100%，但缺陷数量却很高（50个）。这可能表明测试不充分或软件质量下降。

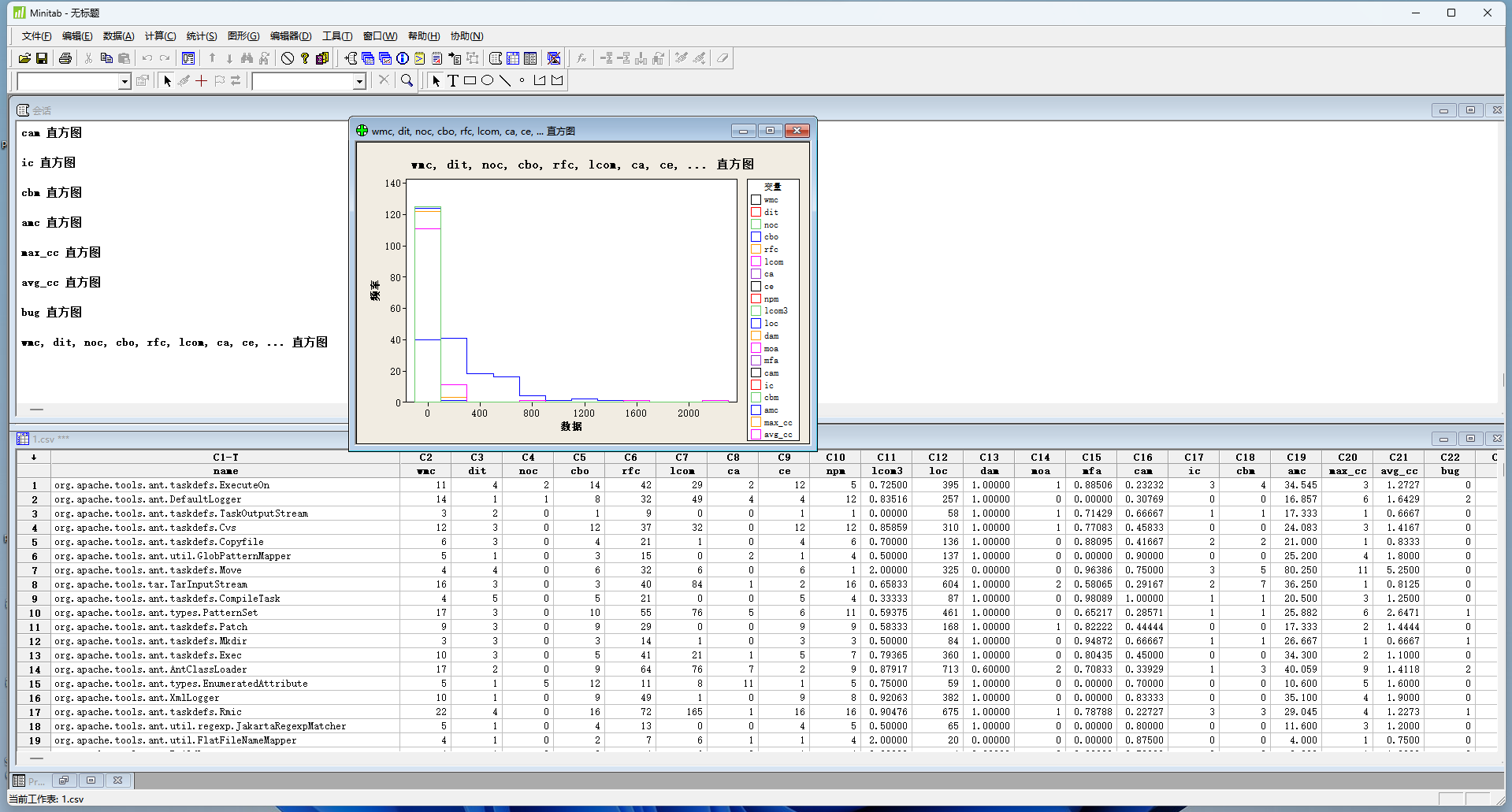
·直方图

对于 Excel ，选择 ：数据→数据分析→直方图 。

对于 Mi niTab，选择 ：图形→直方图，填好需要的参数 。通过带拟合的直方图 ，可以计 算出平均值 、分布区间等 。

…「

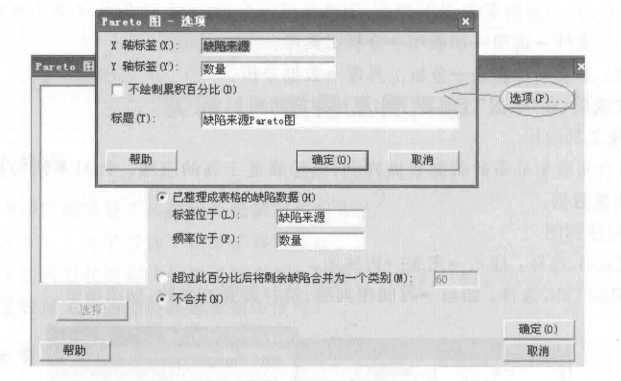


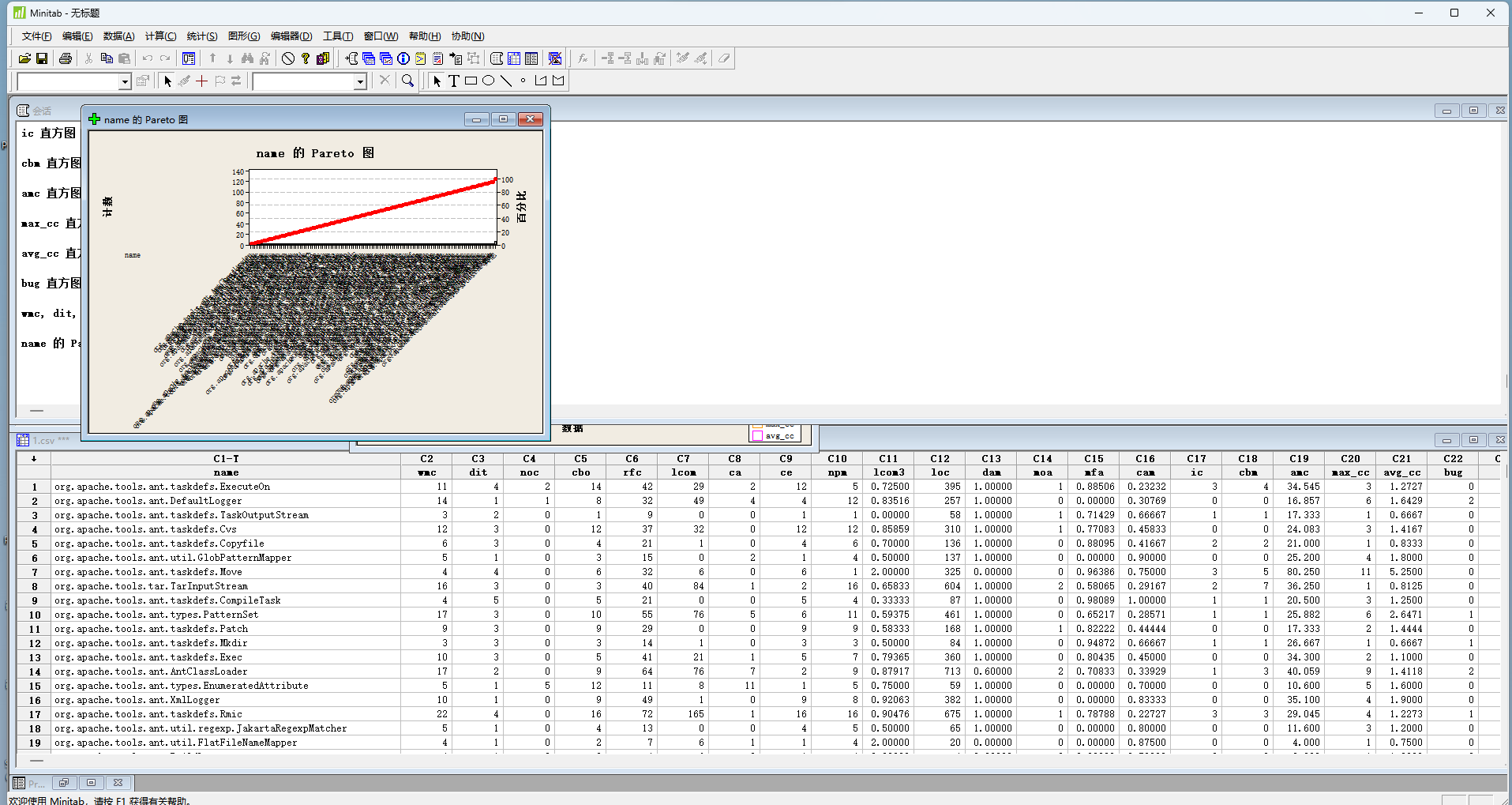


* Pareto 图

对于 Excel ，选择 ：数据→数据分析→直方图，确定后 ，句选“Pareto 图”。

对于 Mi niTa b ，选择 ：统计→质量工具→Pa reto 图，填好需要的参数 ，如图所示 。

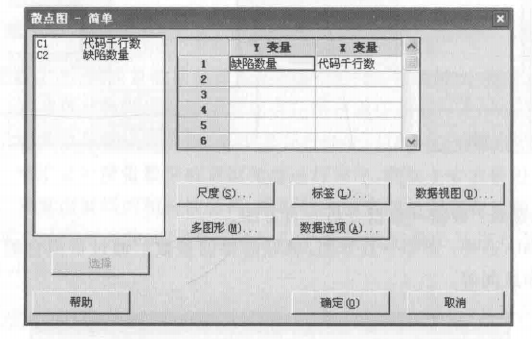


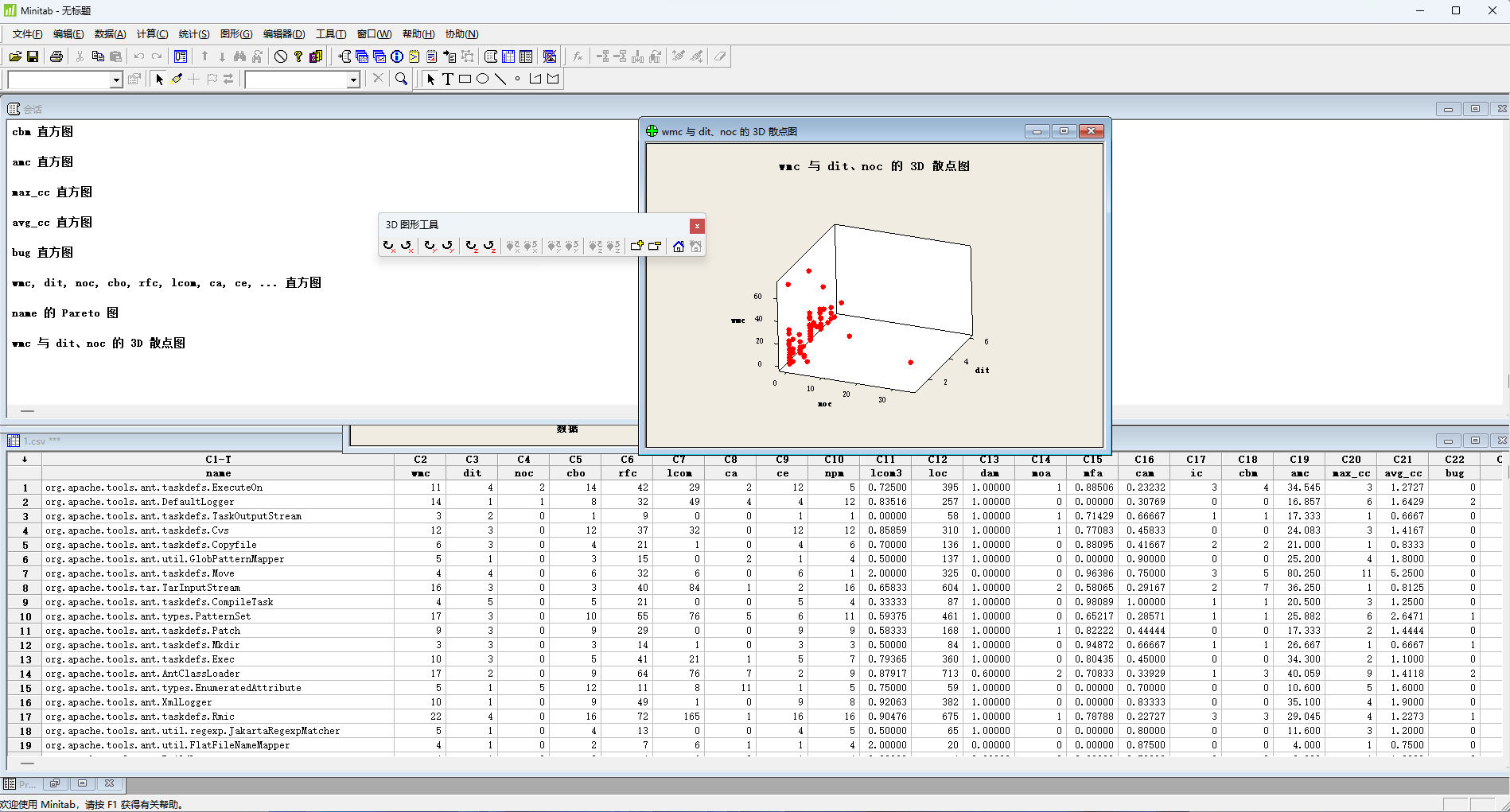


·散点图

对于 Excel ，选择 ：插入→图表→ XY 散点图 。

对于 Mini Tab ，选择 ：图形→散点图，填好需要的参数 ，如图所示 。

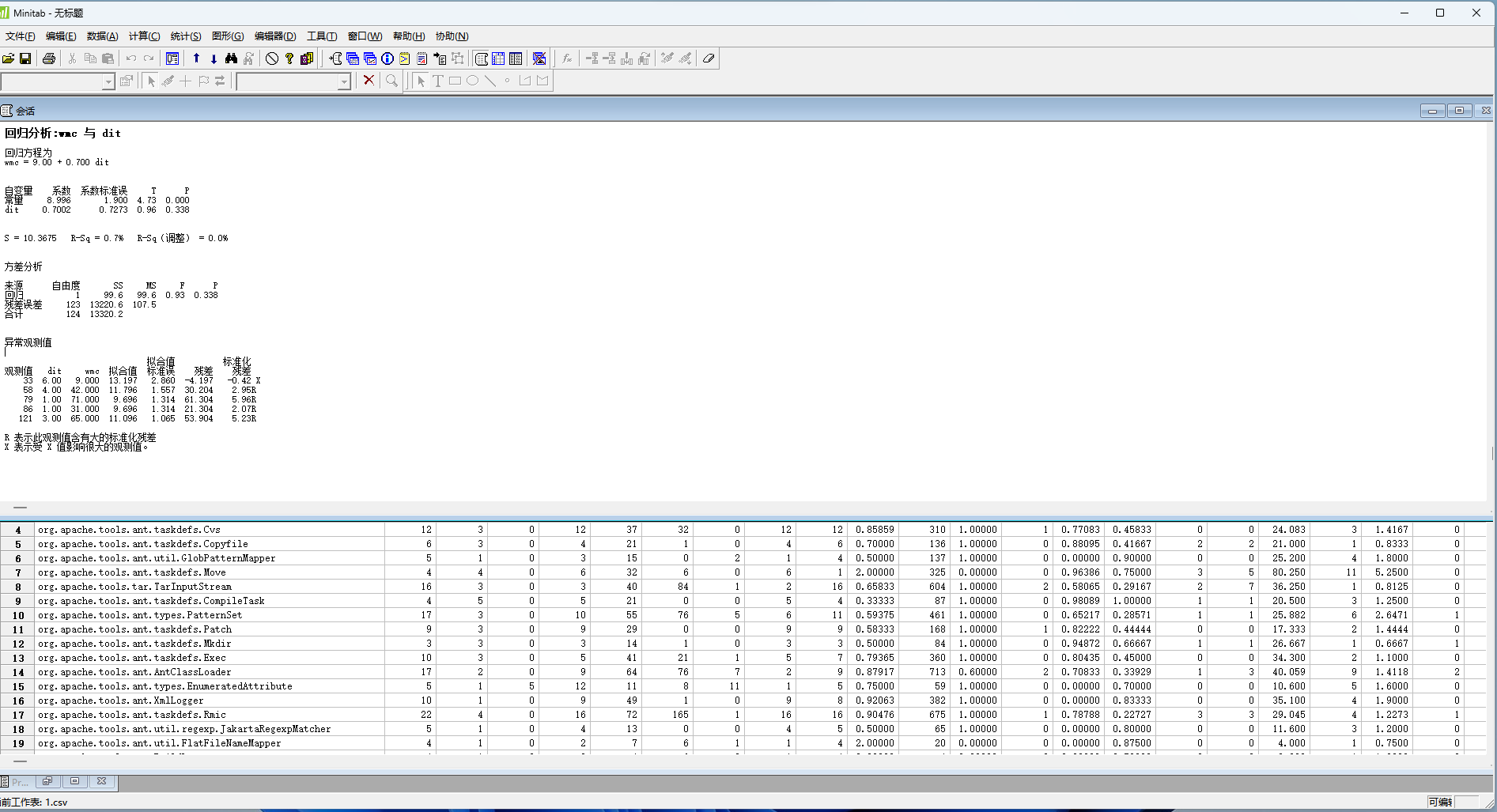




·回归分析

对于 Excel ，选择 ：数据→数据分析→回归 。

对于 Mini Tab ，选择 ：统计→回归→回归。



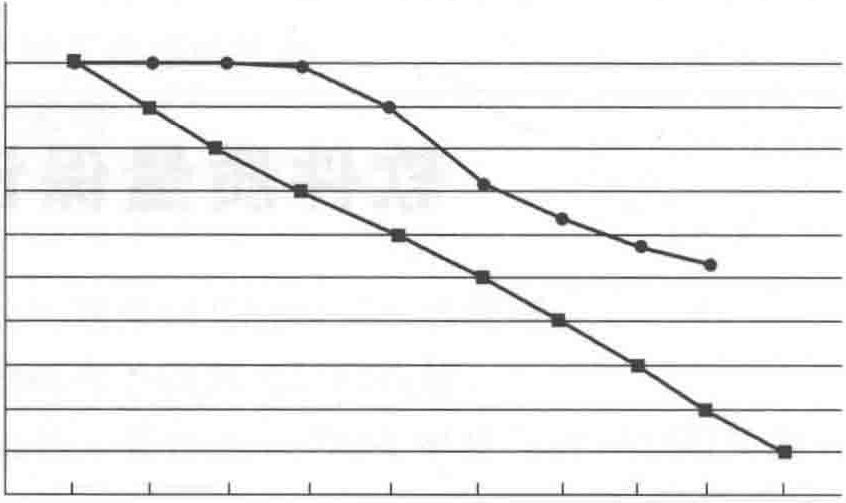
1. 针对不同的质量工具 ，自行设计应用场景 ，构造数据 ，并完成数据分析 。

构造数据生成可以采用 Excel 的数学随机函数 ，例如 =R ANDBETWEEN ( 30 , 100） ，生成 30~100 之间的任意数 。

实验材料里准备了一份缺陷数据作为参考数据 ，可以从中提取需要的数据 。

·时序图：体现随时间变化数据的变化趋势 。至少包括两组数据 ：时间、序列值 。

例 ：燃尽图。敏捷中用于工作任务的可视化管理 ，表示随着时间剩余工作的完成情况 ， “烧尽”至零 。横坐标是时间 ，纵坐标是未完成的任务数 ，分为预计和实际两组 ，图形序列值 应随着时间推移逐步下降 。

so

45

40

35

7音 30

言 25

20

I S

10

s

0 4.22 4.23 4.24 4.25 4.26 4.27 4.28 4.29 4.30 5. 1

日期

··预计 ··实际

·直方图 ：体现对质量数据的分组分布分析。例如 ，每个测试小组的每个人发现的

Bug 数分布 。

• Pareto 图：基于二八定律 ，找出影响质量的这些主要因素 。例如，分析每个版本的 缺陷数据，哪些模块缺陷较多 、需要重点测试 ？哪些阶段发现缺陷较多 ，缺陷发现阶 段是否合理？

·散点图 ：体现质量因素间的相关性 。例如 ，投入测试时间与发现缺陷数之间是否有 关联关系 。

·回归分析 ：设计一组包含两个有相关关系变量的数据（可以采用散点图的数据〉 ，井 确定好 ：自变量 Xl ,X2 ， ，因变量 Y，并依此进行回归分析 ，得到有关 X 、Y 的回归 方程，并进行相应的预测分析 。

(3） 选择某版本的缺陷数据作为原始数据 ，综合应用以上各种质量工具进行分析 ，并对 分析结果加以总结 ，提交分析报告 。体现 ：缺陷数据的趋势分析，缺陷分类数据的分布分 析．缺陷分类数据的相关性分析 ，主要质量因素的分析等 。

# 六 、交付 成果与总结

(1） 实现中所要求的过程数据 、分析结果和分析报告 。

(2 ） 实验过程数据遇到的问题及解决方案 。

实验 2 需求评审实验

### 一 、实验目的

① 加强评审的意识;

② 提高评审的能力;

③ 掌握不同的评审方法 。

### 二 、实验内容

① 选定一个相对简单的软件开发规范或流程规范 。

② 先完成个人单独的评审,再进行交互评审 。

③ 最后进行集体的会议评审 。

④ 整理评审中发现的问题,进行讨论分析 。

### 三 、实验过程

### 软件开发规范或流程规范 ；

### 每个人独立从头到尾看一遍 ，发现问题 ，记下来 ；

### 两个人相互交换材料 ，进行评审 ，然后讨论 ，发现自己评审的不足 ；

### 再分配角色 ，4 人分别担任主持人 、作者 、记录员 、评审人 ；

### 作者简单介绍对开发规 范或流程规范的个人理解 ；

### 开始评审会议 ，主持人协调每个人呈现自己的问题 、发表自己的意见 ；

### 记录员负责记录问题 ，作者或主持人促进大家思考 ，发现更多问题 ；

### 主持人判定是否结束会议 ；

### 主持人跟踪问题 ，并督促作者修改问题 ；

### 主持人召集大家开会 ，达成一致意见 。

四、交付成果

① 交付评审报告,包括评审过程、发现的问题清单。

② 个人总结报告,谈谈对评审作用的认识。

实验 3 软件配置管理实验

### 一 、实验目的

① 熟悉软件配置管理的基本概念和基本原理,确配置管理在团队开发中的重要性。

② 掌握软件配置管理服务的构建方法;

③ 掌握配置管理工具Git的基本使用命令

④ 理解并分析配置管理工具解决冲突的方法,学会为项目团队制定版本控制策略。

### 二 、实验前提与准备

① 准备一个项目的配置文件,包括文档、代码以及不同的版本等 。

② 注册自己的GitHub或者码云账号，或在 Windows 平台下使用 GitBlit 搭建一个 Git服务器（需要配置 Java 运行环境〉 。本教材使用GitHub 作为远程仓库 。

### 三 、实验内容

### ① 安装 Git。

### ② 本地仓库的创建与使用 。

### ③ 连接远程仓库 。

### ④ 本地仓库 、远程仓库协同工作 。

### 四、实验环境

### 本实验使用 Git。根据当前使用的平台，选择在 Linux 、UNIX 、Mac OS X 、Windows 上 安装Git ，所有的安装帮助都可以在Git官网( https://git-scm.com/downloads）获取 。

### 五 、实验过程（ 步 骤）

### ① 在本地安装 Git 。

### 在 Linux 上安装 Git 比较简单，输入命令 $ git ，即可根据提示查看系统是否安装 Git ,井按提示完成安装 。

### 在 Ubuntu 中，则通过命令 Sudo Apt→Get Install Git 可以直接完成 Git 的安装。

### 在 Mac OS 中 .直接从 AppStore 安装集成了 Git 的 Xcode 。

### 在 Windows 上使用 Git ，从 Git 官网下载安装程序，运行 Git Bash 即可。

### ② 本地仓库的创建与使用 。

### 明确工作区 、版本库和暂存区的概念 。练习在本地创建 Git 版本库，在工作区中的项目 配置文件上传到版本库 ，提交对文件的修改文件 、删除文件、下载文件 、版本回退 、管理修改 、撤销修改等基本操作命令 。

### ③ 连接远程仓库 。

### 有 3 种方式可以添加远程仓库 。

### 自行注册 GitHub 账号 ，完成本地仓库和 GitHub 仓库之间 SSH 加密传输设置 ，在

### GitHub 上托管项目代码 、配置文档等 。

### ·下载 GitBlit 等工具，搭建自己的远程配置管理服务 。

### ·在码云上注册自己的账号．创建一个私有项目作为远程仓库 。码云是开游、中国社区 团队推出的基于 Git 的快速的、免费的 、稳定的在线代码托管平台 ，相较于 GitHub 的国外服务器．码云在罔内的访问速度是很快的 ，而且私有项目免费。

### ④ 在本地仓库 、远程仓库间协同工作 。

### 熟悉本地版本库和远程版本库的区别 。练习从远程仓库克隆 、分支管理 、创建和合并分 支 、解决冲突等重要操作命令 。

### 六 、交付成果与总结

### (l）制订版本控制策略 ，为项目团队创建相应的配置管理库结构 ，包括权限分配 。

### (2） 思考 ：多人提交产生冲突如何解决？针对不同的产品线或者项目团队 ，版本控制 策略如何评价 ？

实验 4 基于用例分析技术的需求分析

### 一 、实验目的

① 加强需求质量的意识。

② 提高需求分析的能力;

③ 掌握不同的需求分析技术。

### 二 、实验背景

某公司在线教育事业部正在开发一个在线教育系统，涉及三个版本:Android App，iOS App，和Web 应用，由”课程发现””课程购买””课程学习””课程分享”和”账户管理”等五个模块组成，例如:

① 课程发现具有课程搜索、课程试读、课程分类等功能．

② 课程购买有不同的支付方式和购买方式（如拼团购买)。

③ 课程学习包含已购课程管理、课程留言、课程评分等功能 。

④ 课程分享包含生成海报 、收益管理等功能。

系统应用也有不同的几个角色。

### 三 、实验内容

① 基于上述业务 ／应用背景 ，进行用户角色分析 ；

② 基于已给出的功能提示 ，完善相关的系统功能 ；

③ 结合角色和系统功能 ，进行用例分析 ，画出用例图 ；

④ 针对一些常用的功能（ 如课程发现 、课程试读 、拼团购买等〉，进行应用 场景的挖掘。

### 四、实验过程

(1) 5 个学生组成一个小组 。

(2)小组成员认真阅读实验背景 ，每个成员独立列出用户角色，再同时呈现出 来，进行讨论，整合成一个“角色列表” 。

(3) 每个组员领取其中一个功能模块 ，每个组员先独立进行自己模块的功能完 善 ，如增加新功能、功能分解等 。

(4) 汇集大家所列功能 ，全组成员一起讨论 ，整合成一个完整的、按模块分类的 功能列表。

(5)小组成员各自扮演不同角色 ，设想要使用哪些功能？如何使用?了解每个角 色所使用的特定功能，挖掘应用场景 ，并记录下来。

(6)根据上述记录 ，大家一起绘制用例图，进一步分析功能，理清功能之间的 关系。

(7) 针对一些常用的功能列出应用场景 ，进一步开展头脑风暴活动，最后列出 常用功能的一些典型的应用场景。

(8 )如果可能，决定功能需求的优先级(对*客*户的价值，应用 的频繁性等)。

# 五 、交付 成果

① 用户角色列表。

② (按优先级，从高到低的)功能列表。

③系统用例图 ；

④ 针对关键功能，一些典型应用场景列表；

实验 5 基于代码的质量度量

#### 一 、实 验目的

##### ① 巩固所学的代码质量度量方法及度量指标计算方法 。

② 掌握常见的代码质量度量工具的使用（ 基于 Java 语言） 。

③ 学习如何对度量结果进行分析 、应用 。

### 二 、实 验前提与准备

① 理解各种代码度量指标的含义 、计算方法 、度量指标的建议值或建议范围。

② 度量工具的安装 ：基于Eclipse 平台的插件 ，包括 Metrics 、CheckStyle 等。

③ 提前准备好实验所需要的待分析 Java 代码 。

#### 三 、实 验内容

##### 此处的基于代码的度量包含以下内容 。

① 代码度量 ：对代码单元的度量 ，如规模 、复杂度 。

② 设计度量 ：对代码单元之间关系的度量 ，如内聚 、耦合等 。

在该实验中 ，基于代码的度量主要考虑了如下几方面的度量项 。

·规模 ：非空非注释的代码行(NBNC) 、函数参数个数 、return 语句数 。

·控制流 ：CyclomaticComplexity 、BooleanExpressionComplexit 、NPathComplexit y 、NestedBlockDepth ( NBD ） 。

·耦合 ：扇入 Ca ，扇出 Ce 。

·内聚 ：LCOM 。

．·面向对象的度量 ：类的规模 、类的方法数（NoM ) 、类的属性数（ NoA ) ,DIT、WMC 。

1. 实验环境

Ecli pse IDE 平台 ，已安装好 Metri cs 、CheckSt yle 等代码度量插件 。

Met rics 插件获取地址 ：h tt ps : / / sourcforge. net/ projects / metrics ／。

CheckSt y le 插件获取地址 ：h ttp :/ / checksty le. sourcef rge. net ／。

也可以通过 Eclipse IDE 平台在线安装插件 。

#### 五 、实 验过 程

##### 1） 度量工具安装与配置

(1) Metrics 插件 。

Metrics 插件以 Java 包为单位进行度量 ，属于设计级度量 。除了通用的代码行 LOC 、

McCabe 复杂度等之外 ，该款工具主要提供了面向对象的很多度量项 ，包括 ：C&. K 度量集

的 NOC 、NOM 、DIT 、WMC 等度量项 ；Ma rti n 度量集包括 Ca , Ce , I , A , D 在内全部度量项 ；

Henderson-Sellers 的 LCOM 内聚；LorenZ & Kidd 度量集的 SI 因子等。

有关该款工具的安装配置 ，具体可参见相关的工具安装指导 。

安装配置后．，需要根据度量需要进行调整参数 （ Window → Preference →Metric

Preference ），主要关注如下 。

·度量项的排列次序 。一般来说，产品中较为关心的度量指标放到前面 ，每个度量指标的含义参见相关说明 。

·度量项的统计范围配置 。例如 ，一些静态方法是否纳入 LCOM 的度量范围。

·部分度量项的安全范围 Safe Ranges 配置 。此值作为度量通过标准，超过此值工具会给予告瞥 ，例如，圈复杂度的最大值为 10。但不是所有的度量项工具给出了默认值，没有给出阂值的度量项可以参照教科书中所给出的建议范围进行设置 。

实验中用到的 Metrics 的度量项及默认阀值如下 。

Metrics 度量项

规模

指标解释

默 认 阂 值

TotalLines of Code Namer of Parameters 控制流

有效代码行数

参数个数，单个函数所包含的参数个数 <=5

McCabe Cyclomatic Complexity Nested Block Depth( NBD)

内聚与事Ill合

Afferent Coupling (Ca)

圈复杂度

块的嵌套层次

扇人，包外 class 数依赖该包内 class 数

《10

《5

Efferent Coupling (Ce) 扇出．包内 class 依赖该包外 class 数

Instability (l) 不稳定系数 ，愈高愈不稳定 Ce/ (Ca+ Ce) 《l Lack of C心hesion Of Methods(LCQM) 缺乏内聚

面向对象

umber of Classes

umber of Mcthods(NoM) umber of Attributes( NoA)

Depth of Inherita nce Tree(DlT) Weighted me1hods Per Class( WMC)

类的个数，指定包内的 class 数 单个 class 的方法数

单个 class 的属性数 继承深度

类的方法权重和，指定类内所有方法带权重

（ 般取圈复杂度）的总和，权重取 l [I等于 类的方法数

( 2) CheckStyle 插件。

CheckStyle 是开发人员桌面必备工具 ，目前很多基于Java 的 IDE 开发环境都集成了该款工具 。该款工具为人们所熟知的是代码规范检查，也提供了较多实用的代码度量项 。其度量主要集中在 Size Violations 和 Metrics 这两个模块，分别对应规模和控制流这两方面的度量项 。

工具的安装配置同样需要调参 ，配置必检项 、检查标准等。每个项目可以有自己的配置文件，在度量前选择修改（ Project→Properties→CheckStyle→Configure） 。

实验中用到的 Checkstyle 度量项及默认值如下表所示 。

规模

|  |  |  |
| --- | --- | --- |
| Pa rameter N umber | 参数个数 ，单个函数所 包含的参数个数 | 《7 |
| reru rn Cou n t | return 语句数 ，单个函数内 retu rn 语句个数 | 《2 |
| J ava NCSS | 类／方法中没有注释的语句行数  A method Maximum B c l assMaximum  C fi l eMaximum | A《50  B《1500 C《 2000 |
| Line Length | 单行语句包含字符数 | 《80 |
| FileLengt h | 源文件长度 | 《2000 |
| Method Length | 方法长度 | 《150 |

控制流

Checkstyle 度量项 指标解释 默认阀值

|  |  |  |
| --- | --- | --- |
| CyclomaticComplexity | 环形复杂度 | 主主 10 |
| BooleanExpressionComplexity | 布尔表达式复杂度 ，单行语句中 boolea n 表达式的  ＆＆、｜｜ 、＆、｜ 和A 出现的次数 | 《3 |
| NPathComplexity | N 路径复杂度，指定函数非循环的执行路径总和 | 《 200 |
| NestedBlockDepth | 块的嵌套层次 ，指定函数的块（ （ ｝ 指起）的嵌套层次 | 《《4 |

2 ） 质量工具应用

(1） 导人实验所准备的项目程序 ，选择相应的工具进行度量 ，度量数据会在相应的窗口显示 。

对于 Metrics ：选择 Window→Show View→Other→Metrics ，度量结果在专门 View 中

显示 。

对于 CheckStyle ：右击选择 CheckStyle，违反项以浮动窗口的形式提示 。

( 2 ） 度量结果数据分析 。

每个指标的度量值是一个通用统计值 ，在实际应用时需要结合实际情况具体分析 ，以确认是否有修改代码 。例如 ，框架代码 ，由于架构上职责分工 ，使得度量值具有一定的分布特点 ，如果某类负责分发 ，处于调用链的上层 ，则它的扇出必然很高 ，此时度量值超出范围也不必修改 。某段代码度量值过高 ，如果修改出错的风险较高 ，经过综合评估后 ，也可以暂时不处理。

六 、交付 成果占 总 结

① 实现中所要求的过程数据 、分析结果和分析报告 。

② 实验过程数据遇到的问题及解决方案 。