**软件过程与管理实验 石家庄铁道大学信息学院**

**实验1：个人软件过程**

**本次实验内容是个人软件过程部分，通过本次实验，学生将掌握以下内容：**

1、掌握个人软件过程中的基本理念和各个台阶上对应的软件过程和活动。

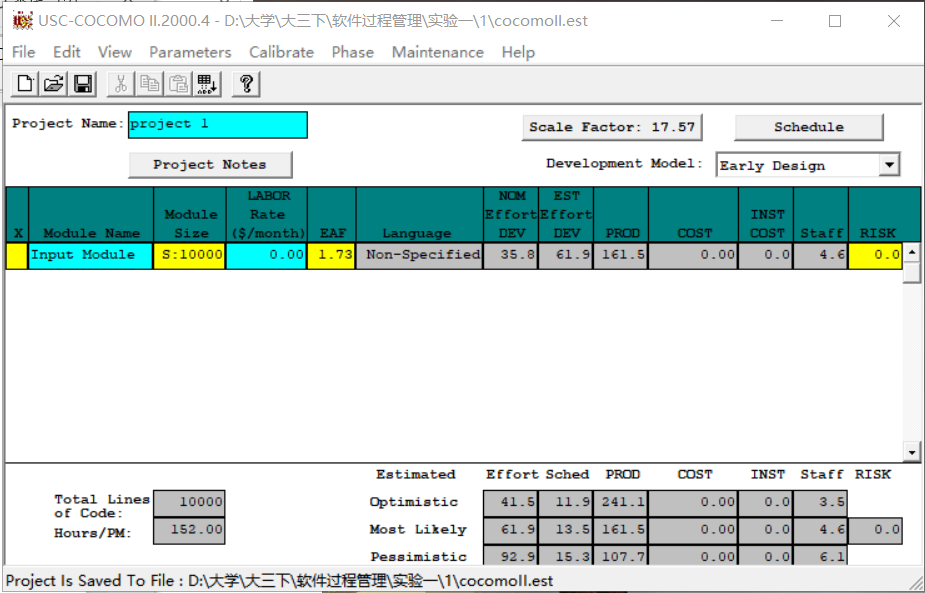
2、能够使用个人软件过程的辅助工具PSP Studio和Process Dashboard来提高自己的软件开发的管理水平。

3、会用COCOMO II模型对软件工作量进行估计。

# [实验任务一]：COCOMO II 软件的使用

**实验要求：**

1. 通过自己看Manual，熟悉COCOMO II软件的使用；
2. 验证**软件工作量估计\_例子.ppt**中的例子，结果如下所示：



1. 提交扩展名为est的COCOMO项目文件，并将结果进行截图。

# [实验任务二]：基于PSP的个人软件过程

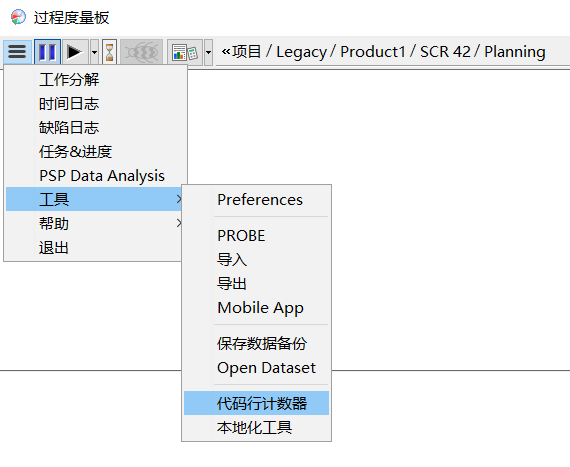
**实验要求：**

1. 通过阅读Manual，熟悉PSP Studio或Process Dashboard软件的使用；
2. 将PSP 2.1等级中的个人软件过程管理方法应用到其它课程中的项目(或者其他实际项目)，完成该等级上对应的表、模板、日志等文档；
3. 利用PSP提供的4个模板完成系统的设计和其中一些模块的设计(登录模块除外)，也可以利用UML中与PSP中4个模板等价的图；
4. 提交对应的程序和使用说明文档。

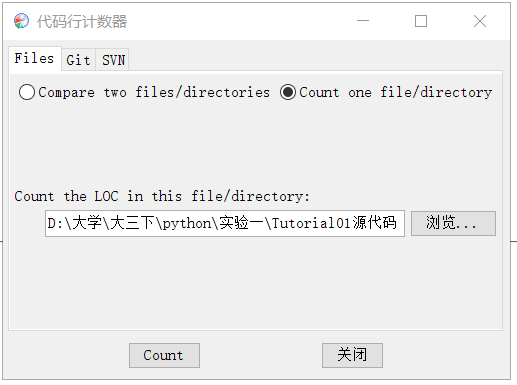
## 2.1 Process Dashboard软件的使用和PSP2.1的应用

**①软件使用（例）：使用代码行计数器**

通过下图操作调出代码行计数器：



然后导入项目，如下图：

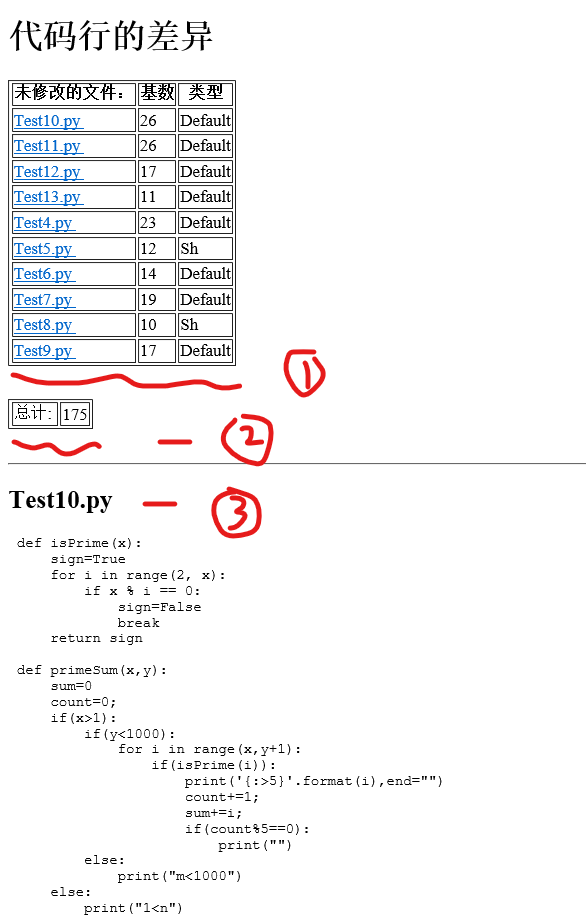


点击Count计数后，会在浏览器里出现以下计数结果：

1、为所选项目的程序文件信息

2、为总计代码行数

3、为某程序文件的具体内容。



表：脚本表，标准表

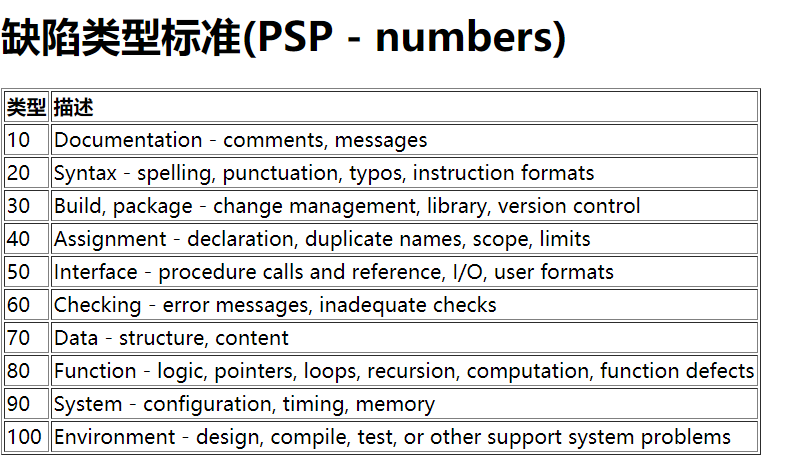
模板：逻辑规格模板，状态机模板，功能规格模板，操作场景模板。

日志：时间记录日志，缺陷日志

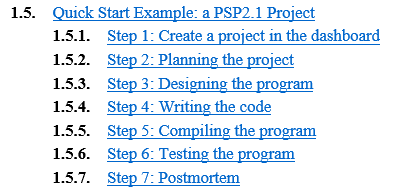
1. 表如图：
2. 脚本表：

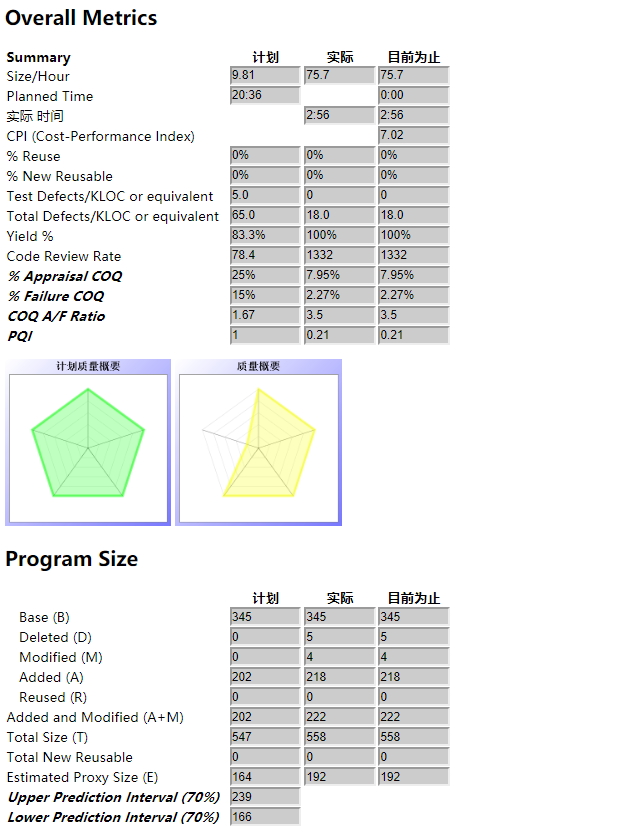


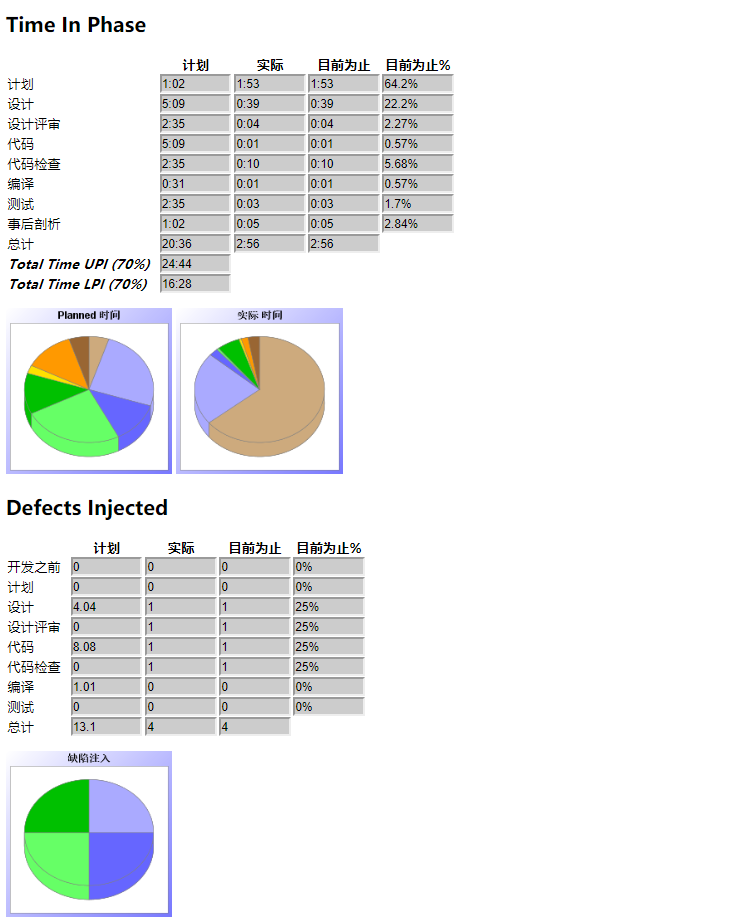
1. 标准表：

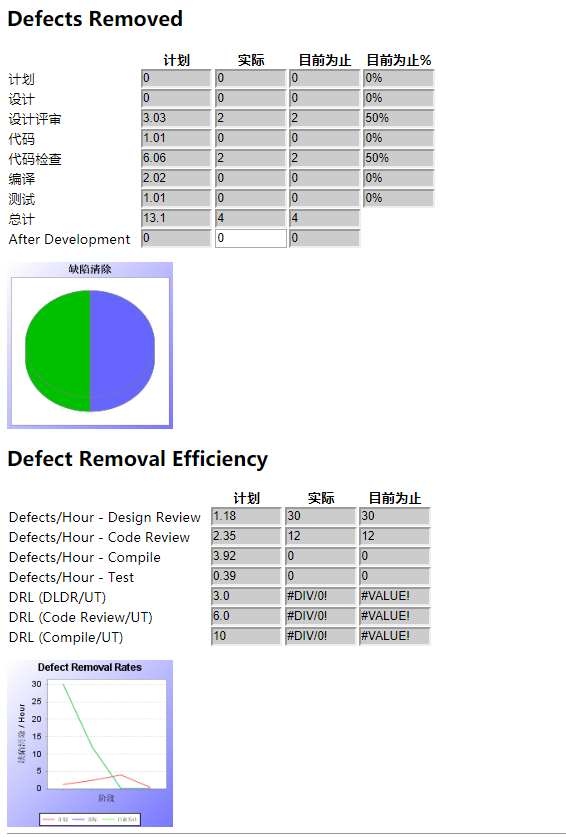


1. 项目计划表（对照用户手册步骤实现，详见步骤参见附属的用户手册中的快速入门示例，实现1.5.1-1.5.7，得出结果，后续会给出部分步骤）：

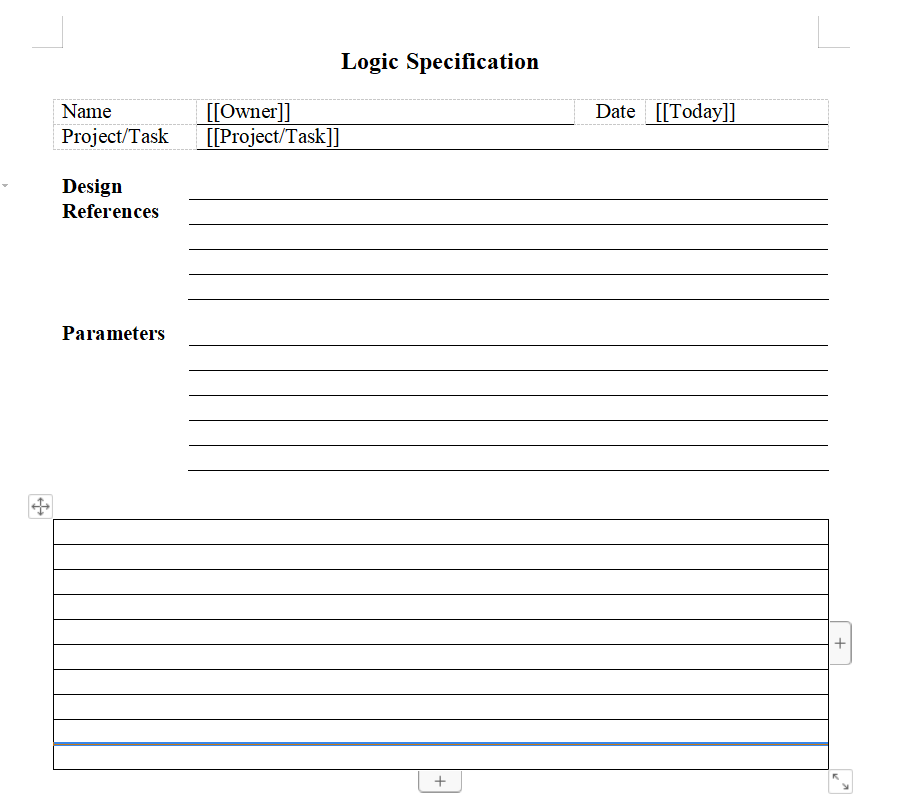




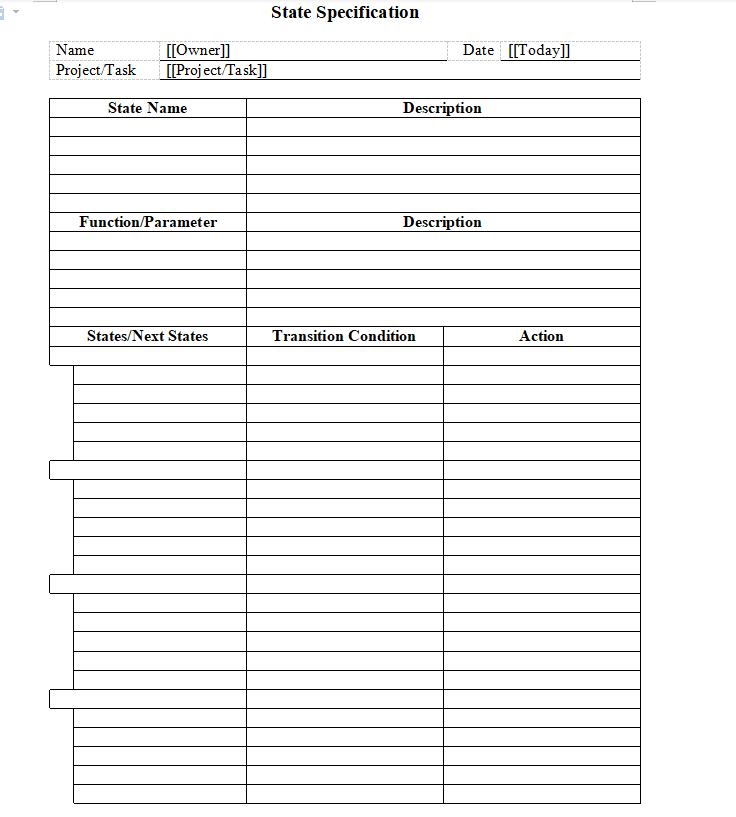




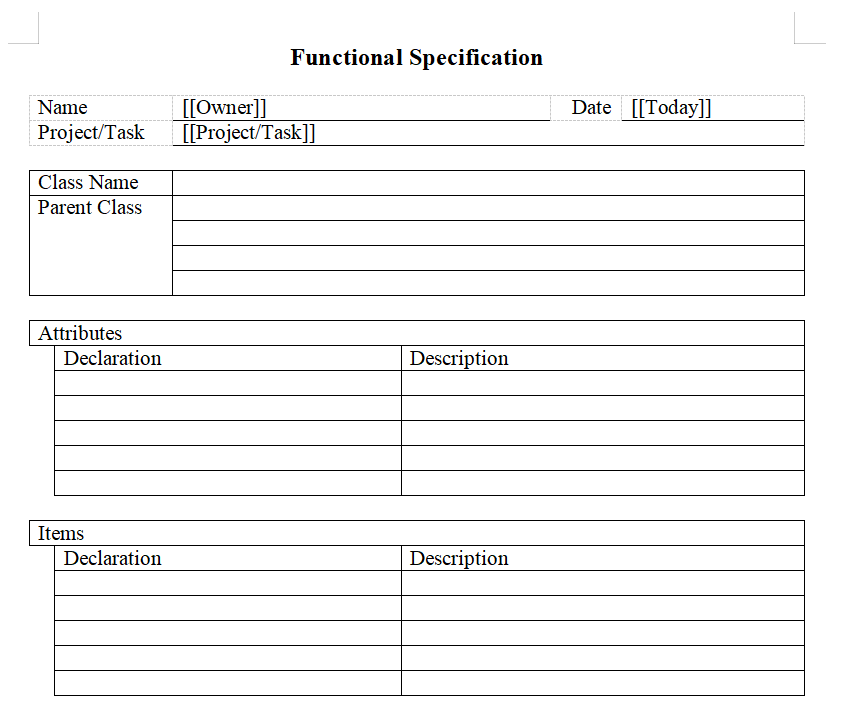
**逻辑规格模板：**



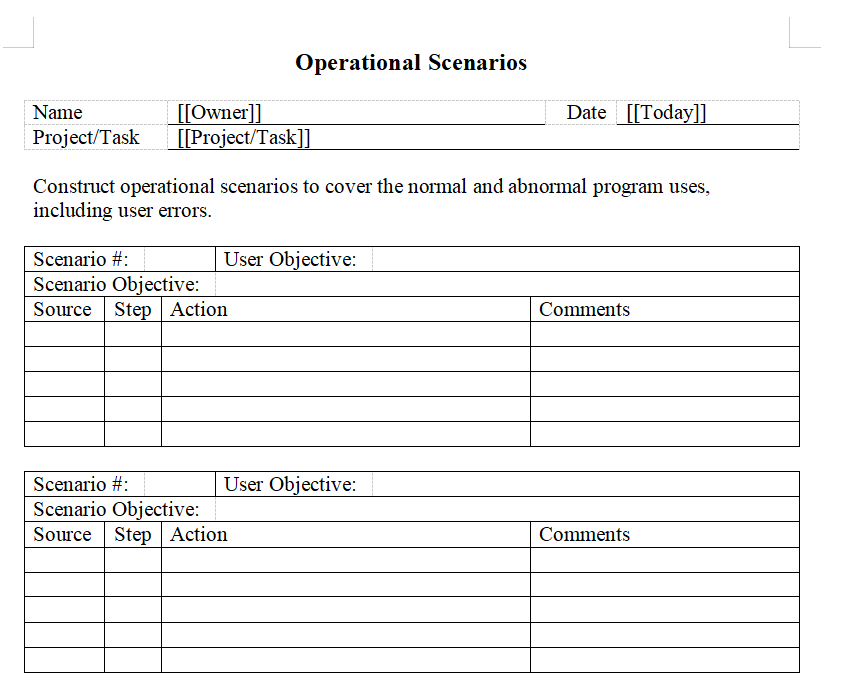
**状态机模板：**



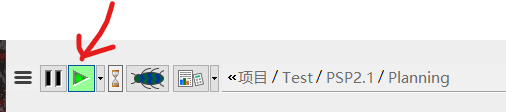
**功能规格模板：**



**操作场景模板：**



点击此处开启计时，



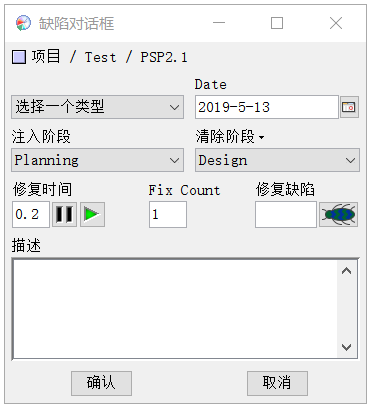
**查看时间、日志缺陷(后续会在规划项目步骤中体现具体应用)：**

**缺陷日志**（描述你的项目开发中所存在的某一些时间段所存在的缺陷，问题，你可以通过添加缺陷项来记录你的项目缺陷及更改完善情况。）提供缺陷列表和更改缺陷数据的方法。

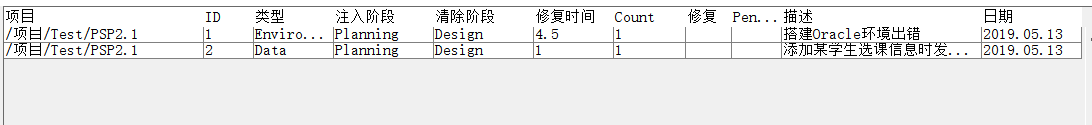
**时间日志：**提供了一个编辑器，允许编辑每个项目和任务中测量的时间数据。记录每一阶段项目的时间日志。

可以通过点击那个甲壳虫的图标来添加缺陷也可以通过点击下图中的缺陷日志中的添加选项来增加按钮来添加缺陷记录，点击时间日志中的增加按钮来添加时间日志。

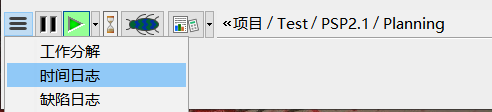
**①添加缺陷：**



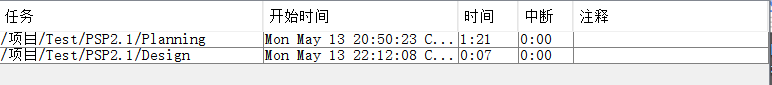
**②缺陷日志：**



**③点击时间日志：**



**④时间日志详情：**

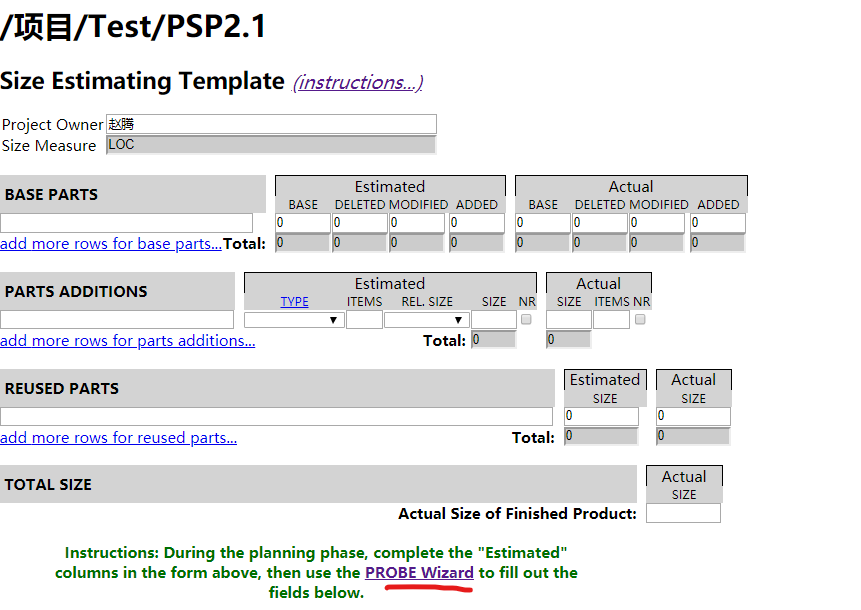


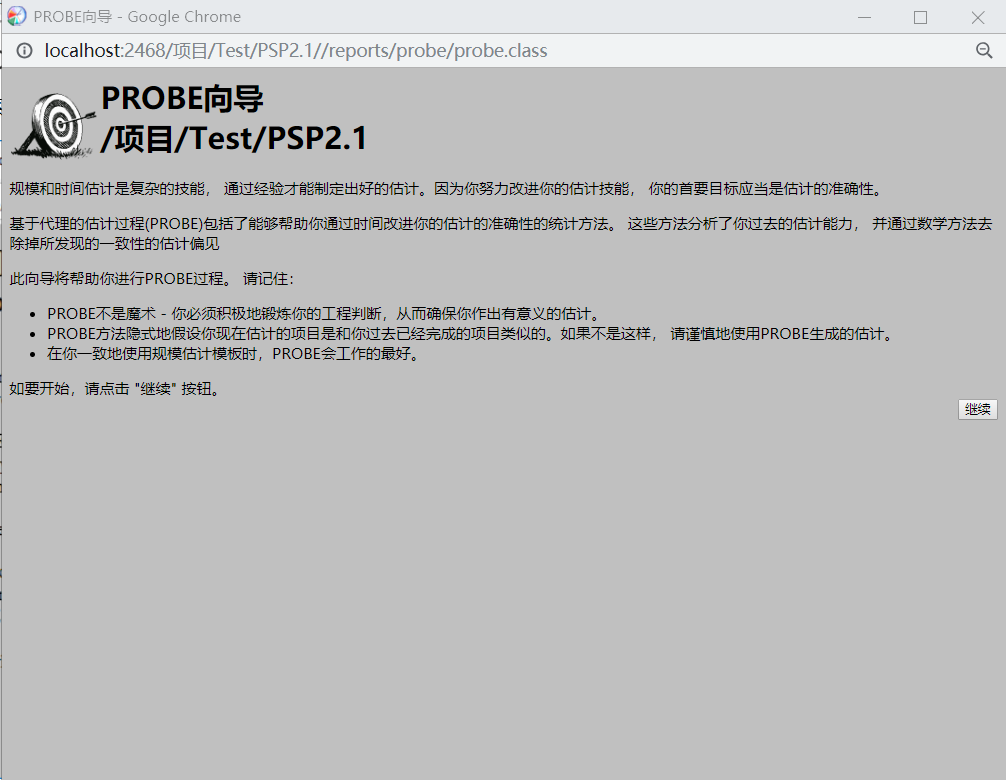
**⑤计时卡片展示：**

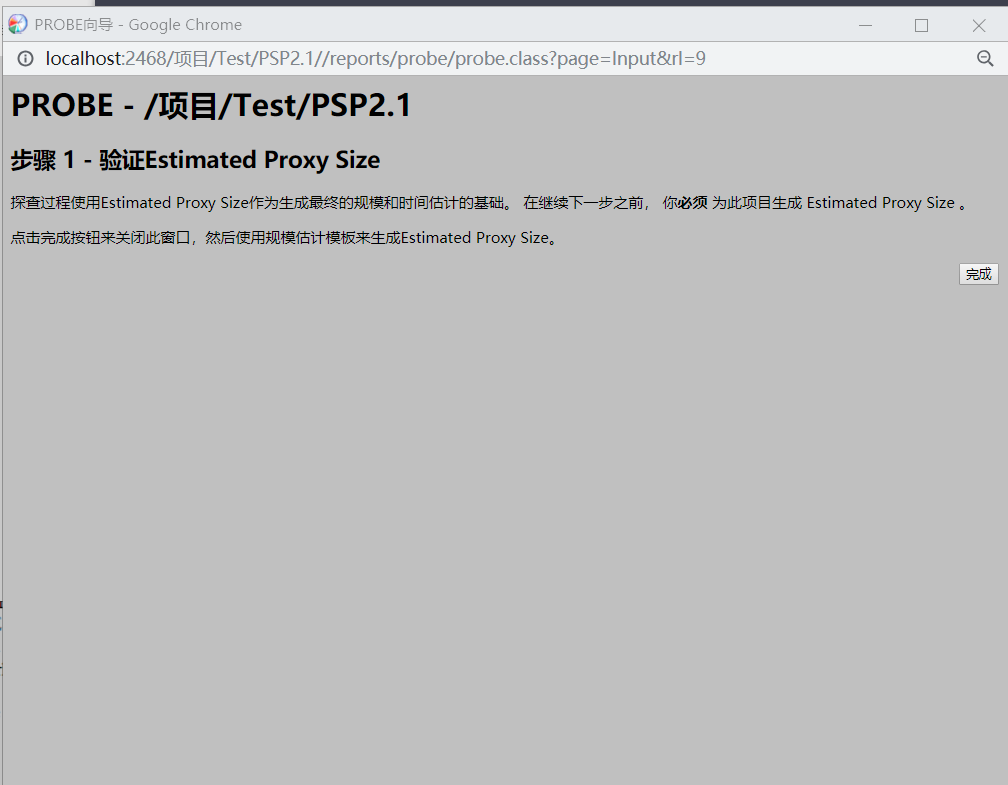


### 规划项目步骤（概述）：

**①理解了程序的要求，就需要进行大小估算。为了指导估算，我们可以使用PROBE方法。按照PROBE方法，计划脚本上的链接会显示PROBE Estimating Script。通过阅读这个脚本，看到一旦我们制作了概念设计，就需要填写尺寸估算模板。可以通过单击页面中间的超链接启动PROBE向导。**



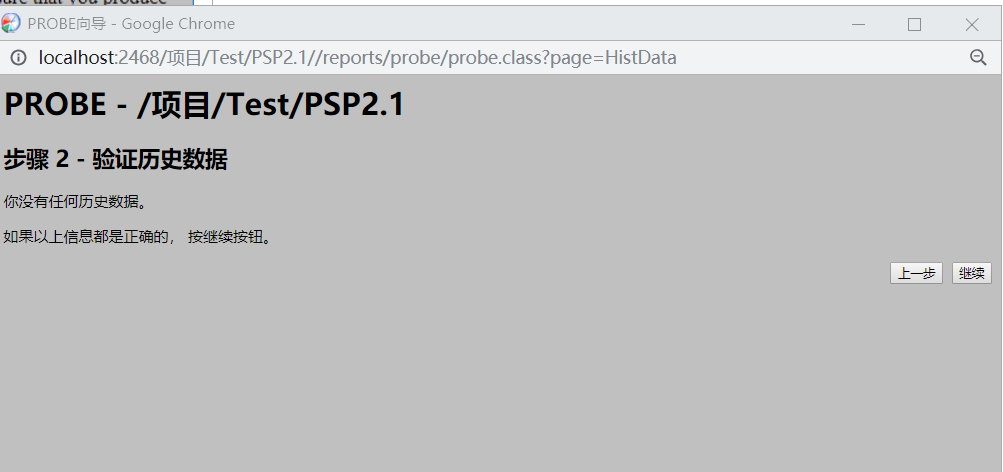




**②点击完成之后去更改Estimated Proxy Size的值，先点击尺寸估算模板的左下角的解除数据只读。**

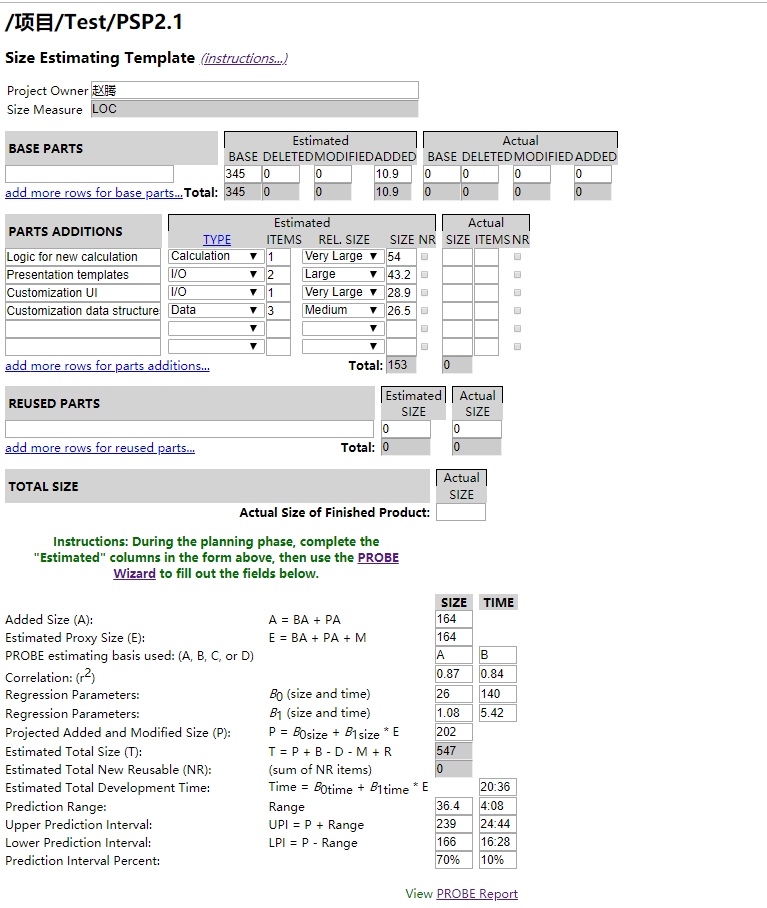




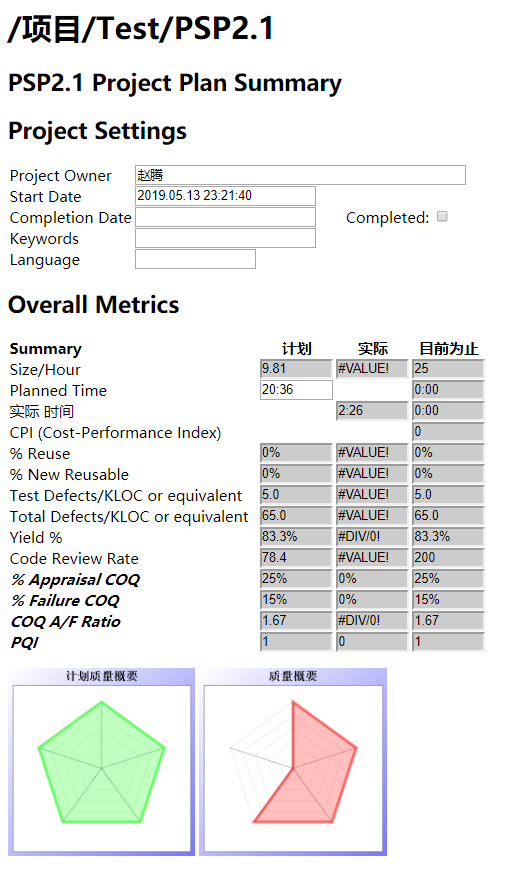


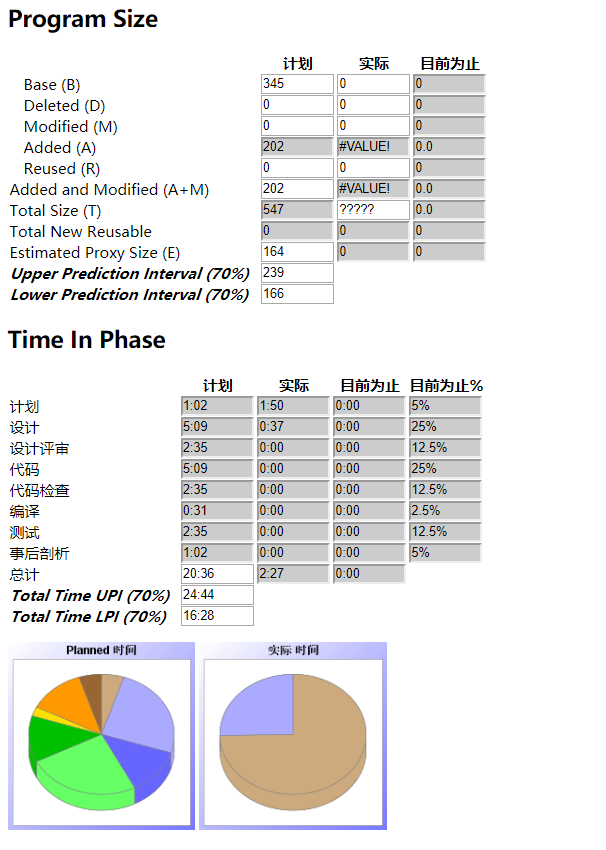


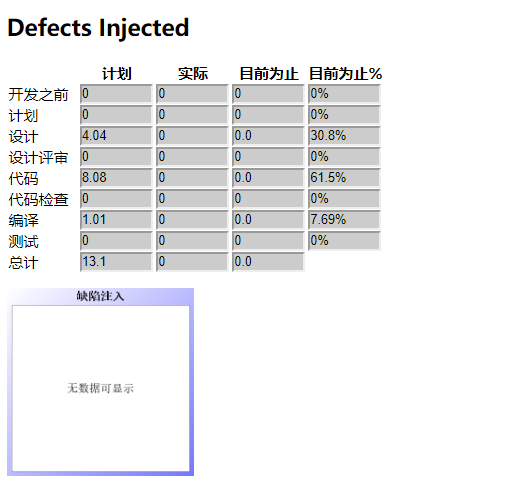
**③对此表单进行规划之后，如下图：**

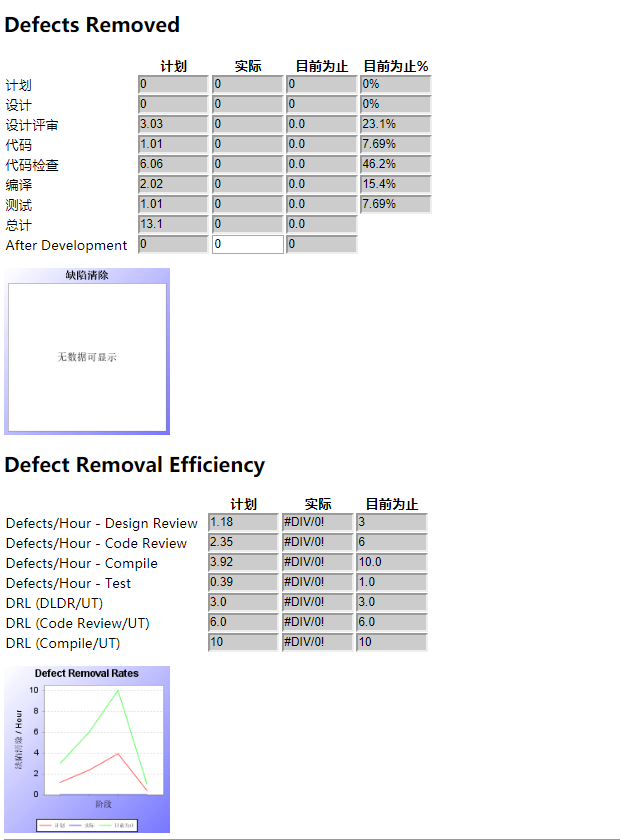


**④调出项目计划摘要，会看到计划数据全部自动输入（还有部分数据没有定义，会在后续操作进行补充）。**

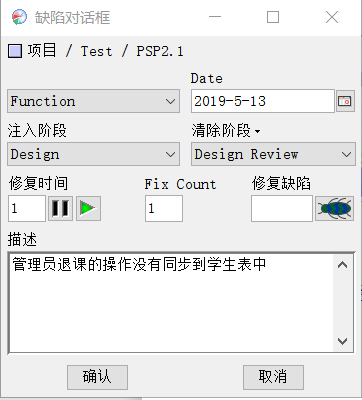




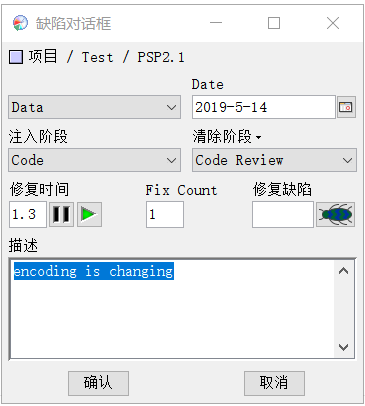




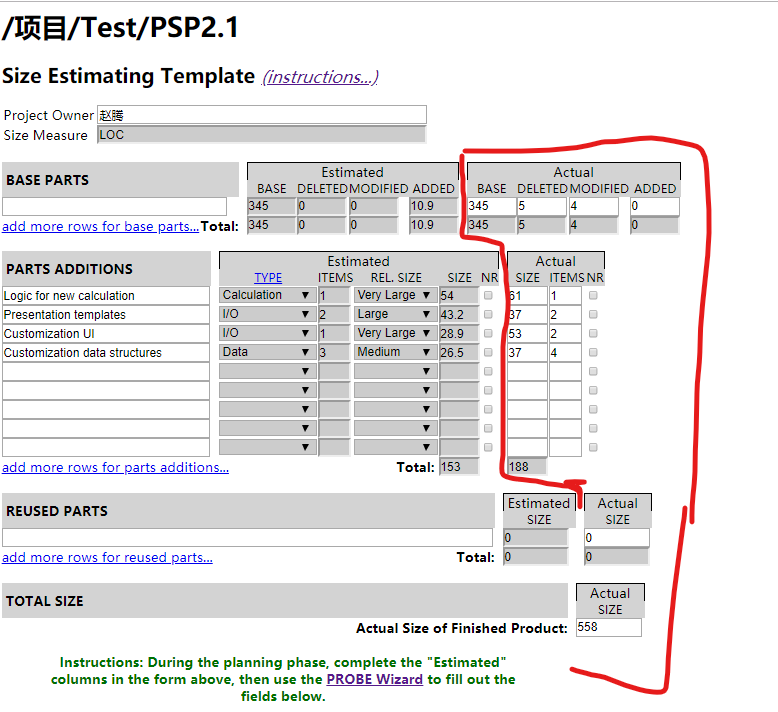
**⑤在Design review阶段，即设计审查，填写缺陷日志。**



**⑥在编码复查的阶段，添加以下缺陷。**

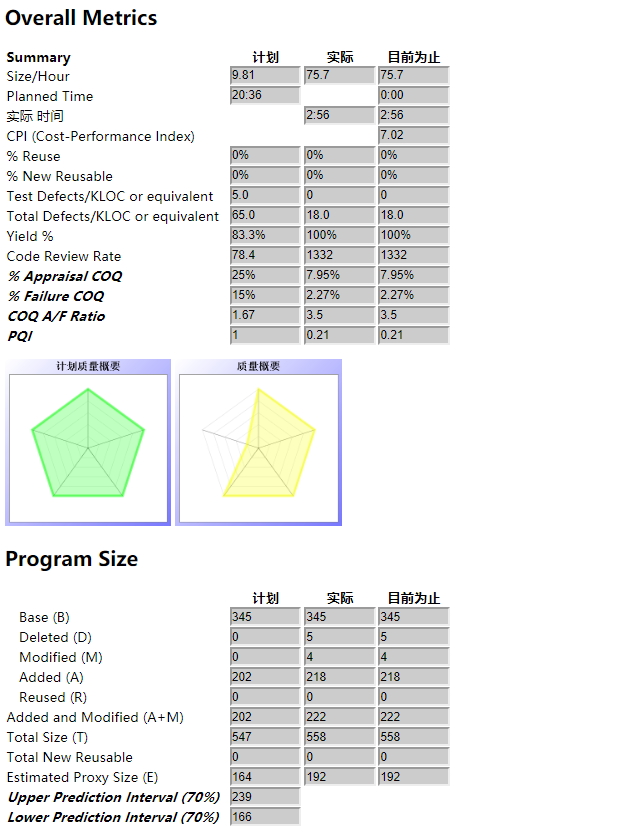


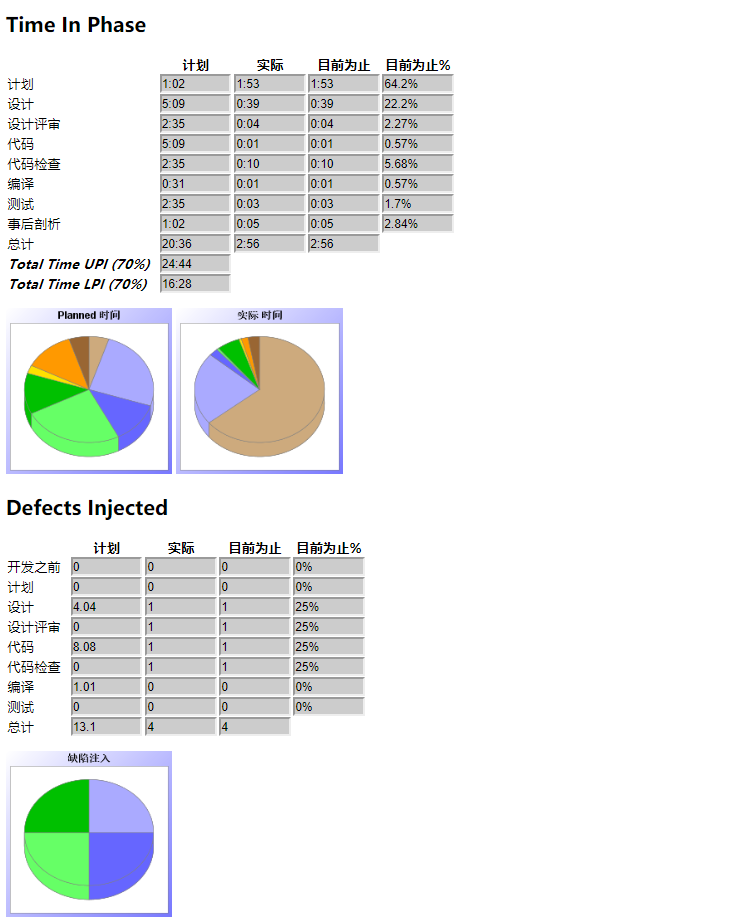
**⑦在Postmortem脚本阶段补充以下图内容。**

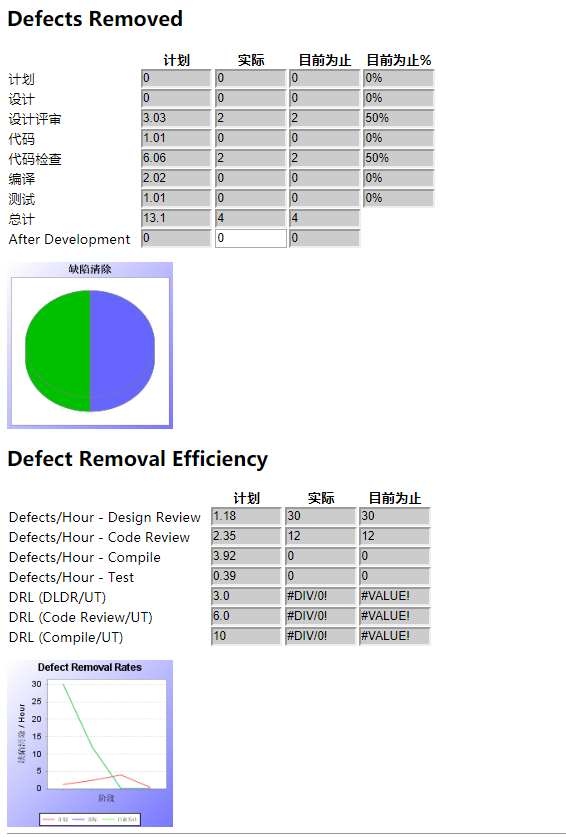


1. **在项目标记完成之后生成的最终的项目计划摘要的信息：**

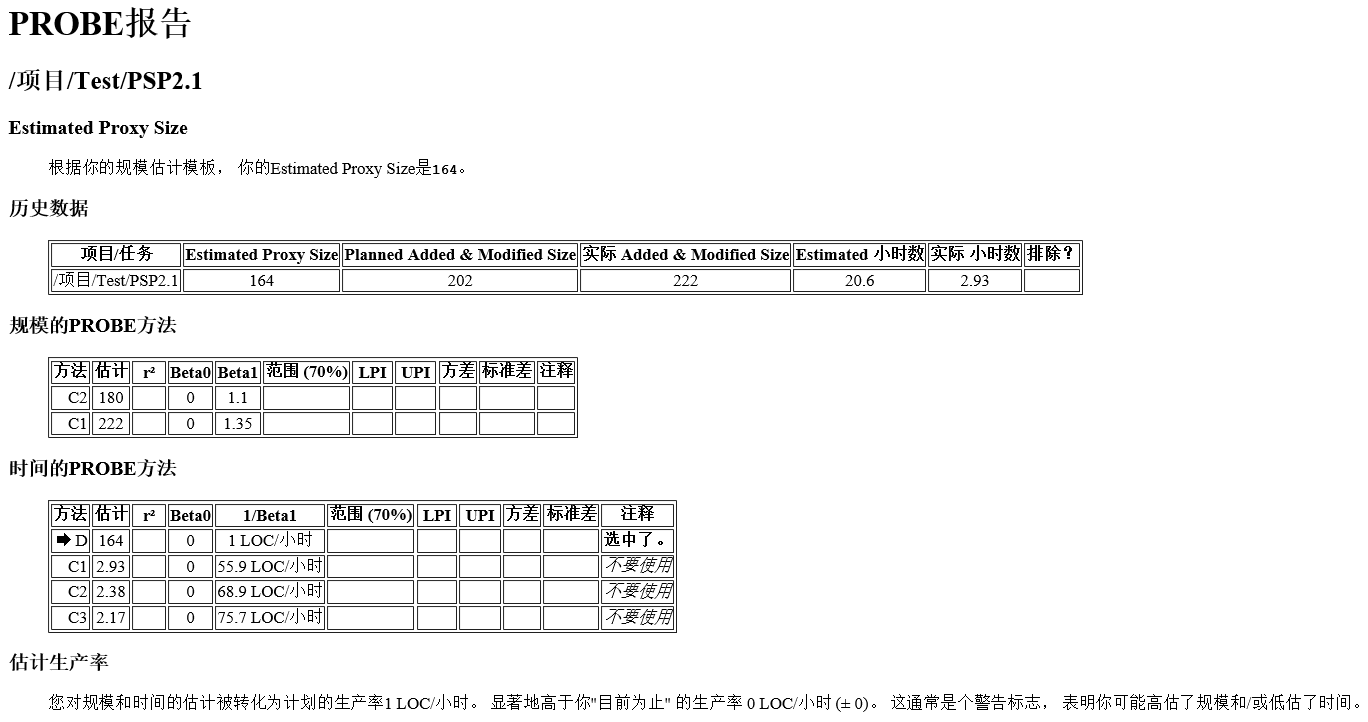
**将项目标记为完整非常重要。这使仪表板知道您已完成项目，并且它可以使用项目数据来计算报告和图表中的各个项目**



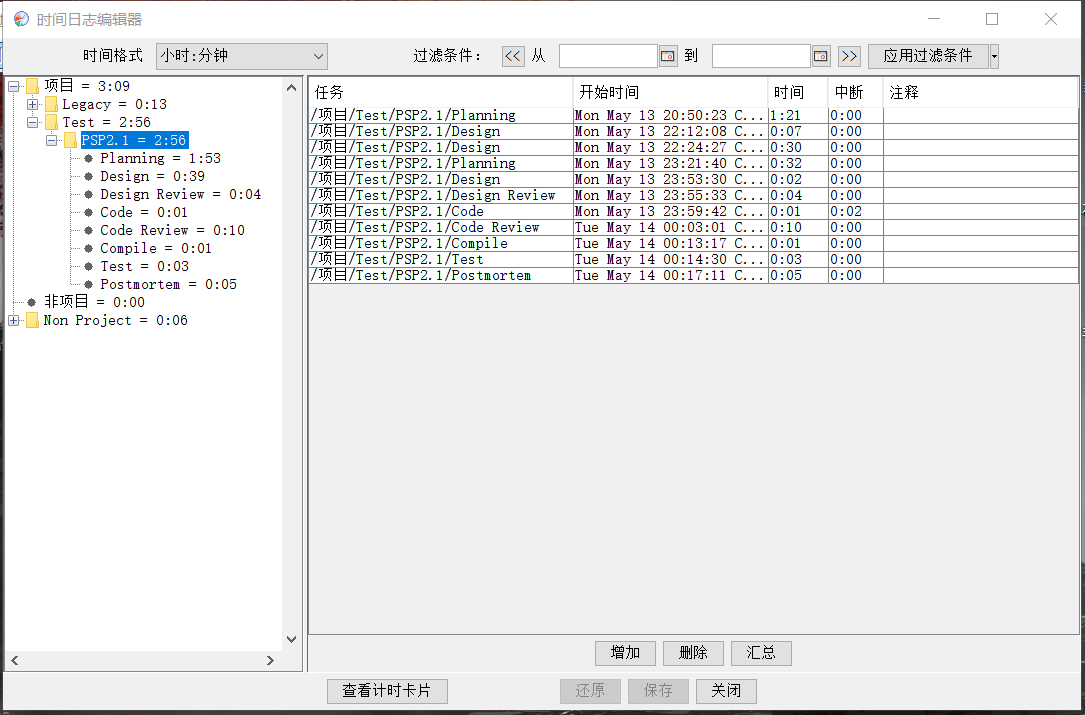




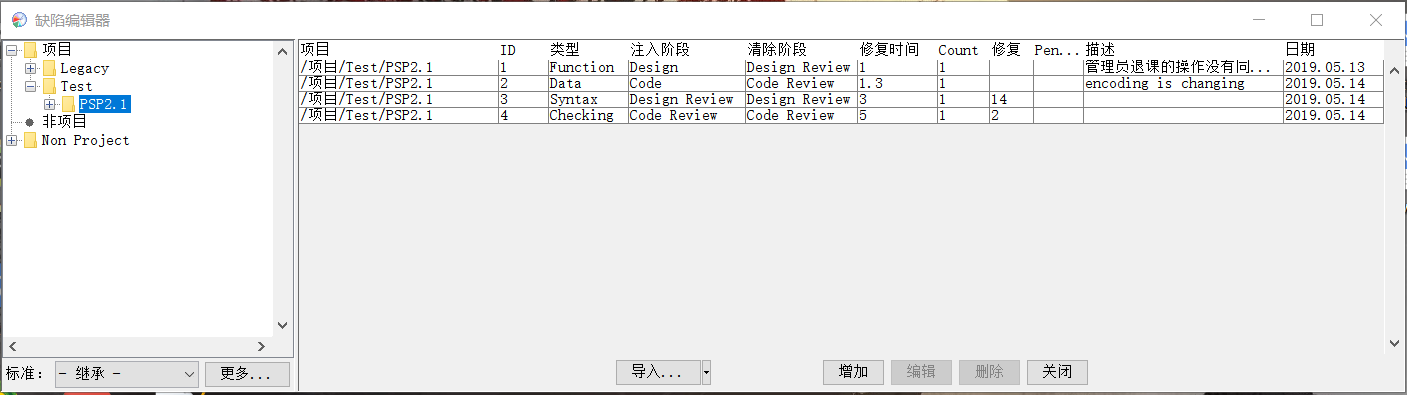
**B.生成的PROBE报告如下图：**



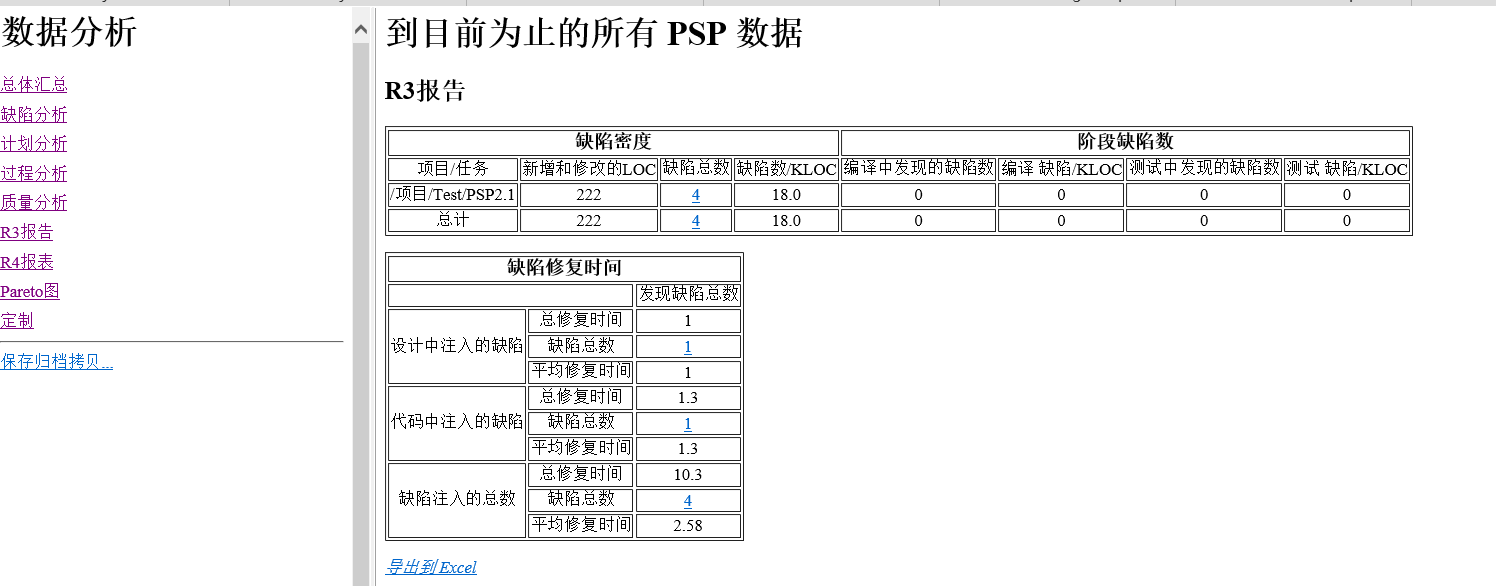
**C.时间日志：**



**D.缺陷日志：**



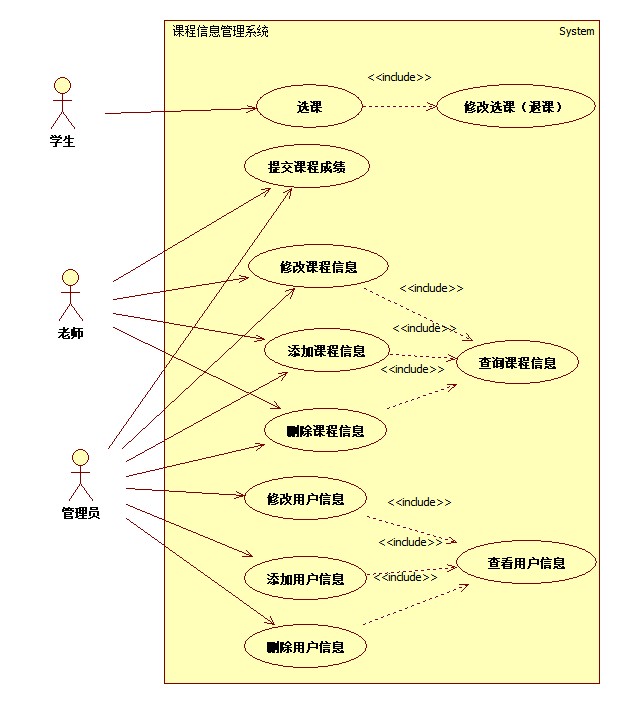
1. **数据分析**



## 2.2 PSP提供的4个模板设计与UML图的对应

**①项目名**：课程信息管理系统

**②用例图**：

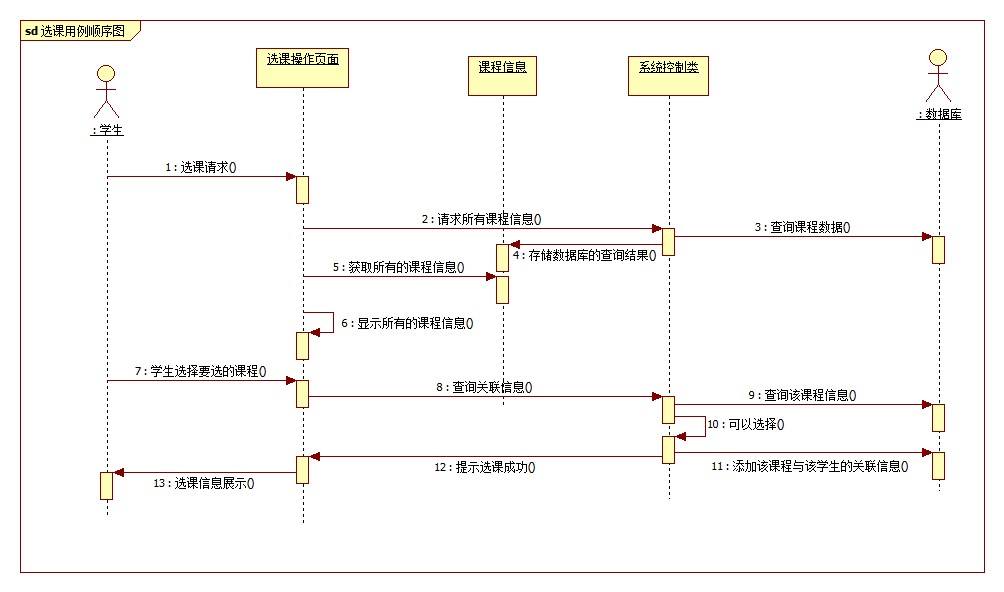


**③用例描述（仅根据选课、添加课程信息展开描述）：**

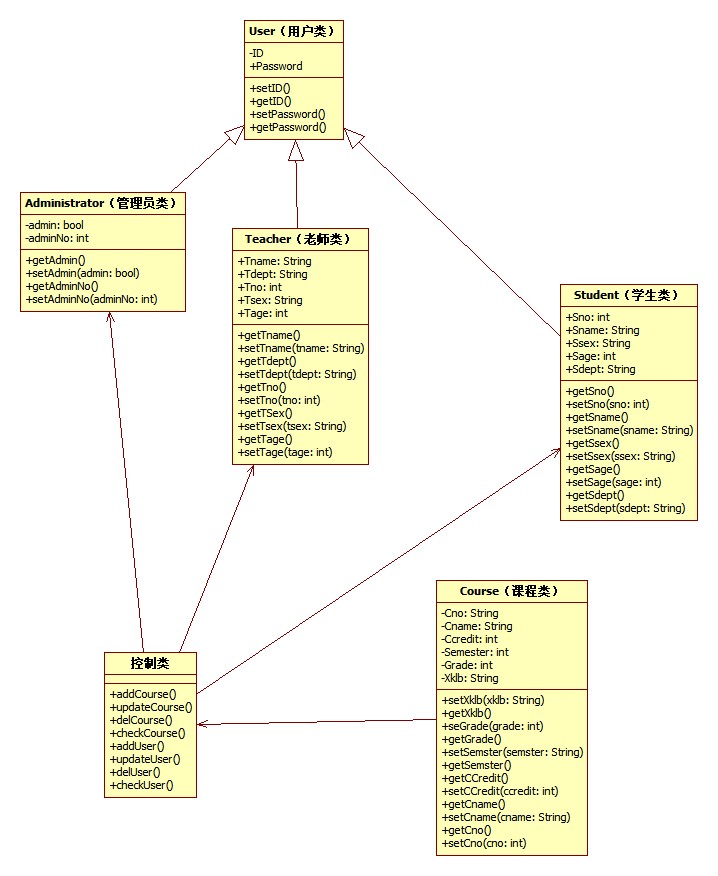
|  |  |
| --- | --- |
| 用例名称 | 添加课程信息 |
| 参与者 | 老师、管理员 |
| 过程描述 | 以下操作者以管理员为例：   1. 管理员选择进入课程信息页面（显示所有的课程信息） 2. 管理员选择添加课程信息按钮   3）管理员进入添加课程信息页面  4）管理员输入课程信息  5）系统验证该课程信息是否有效（或与已有的课程冲突）。  6）A1：无效  7）提示客户添加新课程添加成功，返回自动返回课程信息页面，  8）用例结束 |
| 异常事件流处理 | A1：无效，1）要添加的课程在数据库已存在,系统提醒该课程无法添加。2）学生重新添加课程信息3）选课成功进入第7步 |

|  |  |
| --- | --- |
| 用例名称 | 选课 |
| 参与者 | 学生 |
| 过程描述 | 1）进入课程信息主页面  2）学生点击选课  3）学生进入选课操作页面（显示所有该学生没有选的课程（课程号、课程名、课程学分、所在学期、课程老师工号、课程类型、操作）））  4）然后学生选择课程  5）系统验证该课程是否可选  6）A1：不可选  7）系统提示课程选择成功，返回自动返回课程页面  8）用例结束 |
| 异常事件流处理 | A1：不可选，1）系统提醒该课程不可选，选课人数已经到达上限，2）学生重新选课3）选课成功进入第7步 |

**④顺序图：（以选课用例为示例）**

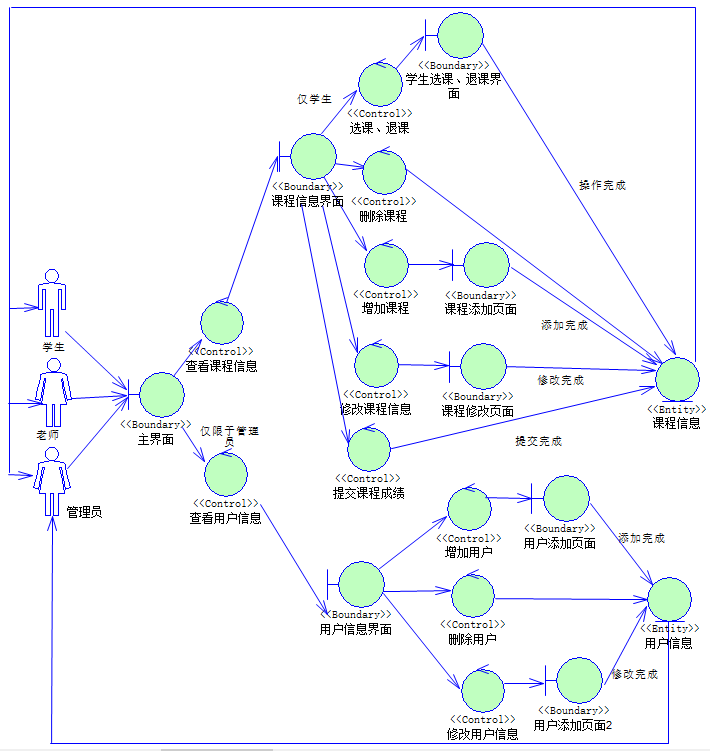


**⑤类图：**



**⑥边界类：**

主要由主页面、课程信息界面、学生选课、退课界面、添加课程页面、用户信息界面、用户添加界面、用户修改界面、课程修改界面构成。



# [实验任务三]：代码规范和评审模板的建立

**实验要求：**

1. 根据自己的编程习惯和缺陷，建立自己的**代码规范**和**代码评审标准**，并且今后务必加以使用；
2. 需要提交**代码规范**和**代码评审标准**两个表格，使用Word即可。

##### JAVA代码规范标准

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **基本对象** | **规则一** | **规则二** | | **规则三** | | **规则四** | | **规则五** | |
| 基本原则 | | | | | | | | | |
| 包的命名原则: | <com>+<组织名称>+<功能>+<具体功能说明> | 扩展包在“组织名称”后加”x” | | - | |  | |  | |
| 类命名: | <名称>+<功能(类别)> | 前缀单词首字母大写，前缀略写规则见后 | | 多个单词紧凑连接，且每个单词首字母大写 | |  | |  | |
| Tab 与 Space、Blank Space | 1 Tab = 4 Spaces | 方法与方法间留空行(Blank Space) | | 操作符号(=、+、|| 等)两边留一个空格 | |  | |  | |
| 大括号{}规则 | 大括号置于关键字或方法、类名下方同列处 | 空方法体，大括号与方法名同行并列 | |  | |  | |  | |
| 小括号()及逗号,规则 | <关键字或标识符>+<(左括号>+<参数>++<)右括号> | <标识符 1>+<,逗号>++<标识符 2>+… | |  | |  | |  | |
| 类文件大小: | 每个类文件不应大于 150 行（不包括注释） |  | |  | |  | |  | |
| main()方法 | 公共类(public)必须有 main()方法置于类文件底部，并有测试的简单代码 |  | |  | |  | |  | |
| 代码书写规范 | | | | | | | | | |
| 方法结构书写规范 | 在方法的头部必须有 Javadoc 注释或者类似的注释 | 如果多于 3 个以上的参数，分行书写 | 不要在方法名和开括号(“(“)间留空格 | | 在关括号(“)”)和开大括号间(“{“}保留空格 | |  | |
| 语句行书写规范 | 参数超过 3 个，分行书写 |  |  | |  | |  | |
| SQL 语句书写规范 | 查询字符串变量统一命名为: query  （SQL 关键字大写、字段名分行书写、表名首字母大写、查询条件分行书写） |  |  | |  | |  | |
| 类、方法、变量命名缩写规范 | | | | | | | | | |
| 命名 | 一般方法命名遵循：<动词缩写(首字母小写)>+<名词(首字母大写)> 变量 | 普通变量遵循：  <前缀缩写(首字母小写)>+< 名词(首字母大写)>**（参数超过 3 个，分行书写）** | final 变量遵循:<F>+<”\_”>+<变量名(全大写,各单词间用”\_”连接)> | | static 变量遵循:<S>+<”\_”>+<变量名(首字符大写,各单词间用”\_”连接)> | | final 且 static 变量遵循:  <FS>+<”\_”>+<变量名(全大写,各单词间用”\_”连接)> | |
| SQL 语句书写规范 | 查询字符串变量统一命名为: query  （SQL 关键字大写、字段名分行书写、表名首字母大写、查询条件分行书写） |  |  | |  | |  | |

|  |  |  |
| --- | --- | --- |
| 定义变量、类名时尽量的使用下列词汇表中的缩写: | | |
| **缩 写** | **全 拼** | **中 文 义** |
| Abv | Above | 在…之上 |
| Absol | Absolute | 绝对的 |
| Acpt | Accept | 接受、适合 |
| Acct | Account | 帐号 |
| Acc | Access | 访问 |
| Actn | Action | 行动、动作 |
| Actv | Active | 活动地 |
| Addr | Address | 地址 |
| Adv | Addvance | 高级地 |

|  |  |  |
| --- | --- | --- |
| Appl | Application | 应用 |
| Buf | Buffer | 缓冲 |
| Cb | Callback | 回叫、复查 |
| Cnf | Confirmation | 确认 |
| Comm | Communication | 通讯 |
| Comp | Component | 构件、组件 |
| Cond | Condition | 条件 |
| Conf | Configuration | 配置、结构 |
| Conn | Connection | 连接 |
| Db | Database | 数据库 |
| Desc | Description | 描述 |
| Deta | Detail | 详细描述 |
| Dest | Destination | 目的 |
| Dev | Device | 装置、设备 |
| Diag | Diagnostics | 诊断 |
| Dir | Directory | 目录 |
| Env | Environment | 环境 |
| Err | Error | 错误 |
| Hex | Hexadecimal | 16 进制 |
| Id | Identifier | 标识符 |
| Impl | Implementation | 执行、实现 |
| Inpt | Input | 输入 |
| Ind | Indication | 指示、迹象 |
| Indent | Indentation | 缩进 |
| Init | Initialize | 初始化 |
| Intro | Introduction | 介绍、初步 |
| It | Iterator | 迭代 |
| LineNo | Line Number | 行序号 |
| Max | Maximum | 最大 |

|  |  |  |
| --- | --- | --- |
| Mem | Memory | 记忆体 |
| Mgmt | Managment | 管理 |
| Mgr | Manager | 管理员、经理 |
| Min | Minimum | 最小 |
| Msg | Message | 信息、消息 |
| MsgQ | Message Queue | 消息队列 |
| Mux | Multiplexer | 多路、复路 |
| Ne | Network element | 网络元素 |
| NumOf | Number Of |  |
| Obj | Object | 对象、物件、目标 |
| Op | Operation | 操作、运算 |
| Outpt | Output | 输出 |
| Param | Parameter | 参数 |
| Pos | Position | 位置 |
| Prev | Previous | 在….之前 |
| Proc | Procedure | 过程 |
| Prog | Program | 程序 |
| Prj | Project | 项目 |
| Ptr | Pointer | 指针、指示器 |
| Recv | Receive | 收到、接收 |
| Ref | Reference | 涉及、参考、引用 |
| Reg | Register | 注册 |
| Req | Request | 请求 |
| Resp | Response | 回应 |
| Sec | Seconds | 秒 |
| Seq | Sequence | 序列 |
| Srv | Server | 服务端 |
| Tmp | Temporary | 临时、暂时 |
| Val | Value | 值 |
| Vec | Vector | 向量、矢量 |

|  |  |  |
| --- | --- | --- |
| 常用 AWT/SWING 类命名标准（组件名一般作为定语放在功能前） | | |
| **缩 写** | **类 名** | **中 文 义** |
| btn | Button | 按钮 |
| frm | Frame | 框架 |
| lab | Label | 标签 |
| mnu | Menu | 菜单 |
| mnuBar | MenuBar | 菜单条 |
| mnuItem | MenuItem | 菜单项 |
| pane | Panel | 面板 |
| scroBar | scrollBar | 滚动条 |
| scroPane | ScrollPanel | 滚动条面板 |
| txtArea | TextArea | 文本输入区域 |
| txt | TextField | 文本输入区框 |

|  |  |  |
| --- | --- | --- |
| 代码书写规范 | | |
| 注释书写规范 | Javadoc Tools 注释 | Javadoc 注释规范为以下格式：  第一段：本类的用途的简要说明  第二段：概要说明  第三段：算法的概要说明  第四段：作者，按年代顺序排列  第五段：版本号  第六段：开发时使用的 JDK 版本 |
| 段落块(Blocks)与语句(Statements)注释 | 尽量不使用/\*…\*/而使用//注释，并且注释要精简，一般在语句转折、算法提示、重要成员变量处注释 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 代码书写规范 | | | | | | | |
| **内容** | **序列一** | **序列二** | **序列三** | **序列四** | **序列五** | **序列六** | **序列七** |
| 资源列表 | Java 标准类（Java standard classes）：java.\*. | Java 扩展类（Java extension classes）：javax.\*. | 第三方厂商类(Third-party classes) | 应用类(Application classes.) |  |  |  |
| 类结构安排： | Javadoc 注释或者其他头部注释 | 类宣告  类宣告又按：  子序列一：extends  子序列二：implements  子序列三：throws  的顺序依次排列类的声明。 | 成员变量宣告  成员变量又按：  子序列一：Public contstants (final Æ static final).  子序列二：Public variables.  子序列三：Protected constants.  子序列四：Protected variables.  子序列五：Package constants.  子序列六：Package variables.  子序列七：Private constants.  子序列八：Private variables.  的顺序排列。  注：对 Public 和 protected 类型变量还要加上 Javadoc 注释 | 构造方法 | 方法(按逻辑归类排列) | 内部类 | main()方法 |

##### 代码评审标准

|  |  |  |
| --- | --- | --- |
| **常规** | **是/ 否** | **备注** |
| 程序逻辑是否符合详细设计 |  |  |
| 程序是否明确地注释了其实现功能 |  |  |
| 注释格式是否正确 |  |  |
| 程序单元是否承担单一职责 |  |  |
| 复杂程序是否合理地分解为子程序 |  |  |
| 程序中接口定义是否明晰 |  |  |
| 异常处理是否符合规范 |  |  |
| 包名、类名命名是否符合规范 |  |  |
| 方法名命名是否符合规范 |  |  |
| 代码中段落布局是否合理、是否用恰当的空行 分隔 |  |  |
| 代码的断行、对齐、缩进、空行是否恰当 |  |  |
| 代码中的变量、属性、参数、方法、类是否恰当命名，是否无相互混淆的命名 |  |  |
| 其他编码风格是否符合规范要求，如if、else永远加{}；for、while的格式；switch永远有default；case有break等等基本语句格式 |  |  |
| 记录错误日志使用架构统一方法 |  |  |
| 下标的值在范围之内且不存在非整数下标 |  |  |

|  |  |  |
| --- | --- | --- |
| **变量 /常量** | **是/ 否** | **备注** |
| 相关系列变量是否统一集中声明，自定义数据类型变量是否进行了必要而充分的描述 |  |  |
| 变量是否按相关规范进行了命名（有意义的命名、数据类型的区分） |  |  |
| 变量是否按其命名含义承担了单一的使用目 的，必要时是否清晰定义了额外变量 |  |  |
| 字段是否使用 getter/setter方法访问 |  |  |
| 是否应用良好定义的枚举类型替代了简单的字 符标识或布尔变量使用 |  |  |
| 处理可变 String时，是否使用 StringBuffer/ StringBuilder |  |  |
| 是否优先使用 Java标准库提供的容器 （ArrayList 、HashSet、HashMap、LinkList ）来 处理顺序结构、集合、关联数组合堆栈队列 |  |  |
| 成员变量和方法 public/private/protected 关 键字是否恰当 |  |  |
| 内部类、非公共类及其内部成 员、方法的访问权限是否合适 |  |  |
| 不同作用域变量名称尽量不要相同 |  |  |

|  |  |  |
| --- | --- | --- |
| **算法** | **是/ 否** | **备注** |
| 数据结构是否合理、精简 |  |  |
| 算法是否可以独立测试，是否与数据库和其他 算法隔离 |  |  |
| 是否尽量避免了嵌套的运用 |  |  |
| 复杂逻辑是否进行了必要而充分的注释 |  |  |

|  |  |  |
| --- | --- | --- |
| **控制** | **是/ 否** | **备注** |
| 代码执行路径是否清晰 |  |  |
| 控制逻辑复杂度是否合理，是否进行了必要而 充分的注释 |  |  |
| 每个循环体是否仅执行了单一而明确的功能 |  |  |
| 与常数比较需要将常数放在比较表达式的前面 |  |  |
| 没有使用任何实例类成员（包括方法和成员变 量）的方法是否被声明为静态的 |  |  |

|  |  |  |
| --- | --- | --- |
| **布局** | **是/ 否** | **备注** |
| 代码是否良好格式化并能体现其逻辑结构 |  |  |

|  |  |  |
| --- | --- | --- |
| **设计** | **是/ 否** | **备注** |
| 程序是否可读、可扩展并健壮 |  |  |
| 具体实现细节是否已尽可能的 “隐藏 ” |  |  |
| 是否优先使用接口而不是抽象类或具体类 |  |  |
| 方法参数是否在 5 个以内 |  |  |
| 方法慎用 public 修饰符 |  |  |
| 对于不需要子类来重载的类尽量使用 final |  |  |
| 对象作用域是否过大 |  |  |

|  |  |  |
| --- | --- | --- |
| **效率** | **是/ 否** | **备注** |
| 尽量不要在循环内出现远程调用 |  |  |
| 每个业务动作远程调用次数是否小于 3 次 |  |  |
| 远程调用数据传输是否有不必要的冗余数据 |  |  |
| 如果已知道数据长度， new一个 List 或Map时， 加上长度参数 |  |  |

|  |  |  |
| --- | --- | --- |
| **输入流 /输出流检查** | **是/ 否** | **备注** |
| 文件属性正确 |  |  |
| I/O语句符合格式规范 |  |  |
| 缓冲大小与记录大小相匹配 |  |  |
| 文件在使用后是否关闭 |  |  |
| 是否处理了 I/O错误 |  |  |

|  |  |  |
| --- | --- | --- |
| **SQL 检查** | **是/ 否** | **备注** |
| Sql语句小写 |  |  |
| 引用字符使用单引号 |  |  |
| 严禁使用 select \* 形式的语句，必须指出具体字段 |  |  |
| 严禁使用 insert into table values（？，？，？），必须指出具体要赋值的 字段 |  |  |
| 避免隐含的类型转换（不同数据类型字段相 加） |  |  |
| 子查询前后必须加上括号 |  |  |
| 禁止使用视图 |  |  |
| 禁止使用 XX in () or XX in () |  |  |
| 禁止使用 or超过 500个 |  |  |
| 禁止使用 not in，建议使用 not exist |  |  |
| 禁止在一条 sql语句中使用 3层以上的嵌套 |  |  |
| 如果有多表连接时，应该有主从之分，尽量从 一个表取数 |  |  |
| Where子句过滤条件，索引列或过滤记录最多 的条件应该放在前面 |  |  |
| 字符串连接必须使用 “|| ” |  |  |
| Case when语句中只能出现 =、>=、<=以及 is null 运算符 |  |  |
| Sql 中函数是否在指定范围内 |  |  |
| Join 与 on 必须严格匹配 |  |  |

|  |  |  |
| --- | --- | --- |
| **目录结构** | **是/ 否** | **备注** |
| Jsp,Action ，service ，dao , 逻辑分层是否明 确 |  |  |
| 配置文件（ struts ，spring ，sql ）是否条理清 晰，目录规范 |  |  |
| 页面资源文件（ JavaScript,CSS,images ）目录 是否规范 |  |  |
|  | | |
| 其他检查 |  |  |
| 不存在“警告”或“提示”信息 |  |  |
| 是否有输入信息的合法性检查 |  |  |
| 是否有功能遗漏 |  |  |