

单元测试报告

刘林青 1354361
杨安琪 1353005

目录

1 引言

1.1 目的

1.2 测试策略

2 测试用例与分析

2.1 数据库模块

2.1.1 被测方法参考表：

2.1.2 不被测方法表：

2.1.3 用例分析与设计

1) openDataBase 测试分析与设计

2) modifyDataBase 测试分析与设计

2.2 收入支出模块

2.2.1 被测方法参考表

2.2.2 不被测方法参考表

2.2.3 用例设计与分析

1) setValue 测试分析与设计

2) removeValue 测试分析与设计

3) setType 测试分析与设计

4) getSum 测试分析与设计

3 单元测试缺陷汇总

4 单元测试缺陷分析

4.1 计划的用例数／执行的用例数

4.2 通过的用例数／失败的用例数

4.3 缺陷等级划分标准

4.4 识别的缺陷数—按测试类型

4.5 识别的缺陷数及其状态

4.6 缺陷分布—按模块划分

1 引言

1.1 目的

本文档对“随手记”记账软件的单元测试结果进行总结，对bug 进行汇总和分析，为后续集成和系统测试提供参考。本文档的主要读者是测试经理、测试人员和开发人员。

1.2 测试策略

以类为单元，采用自底向上的单元测试策略，通过设计相应的驱动和桩的方法来测试类中的方法，在选择类的被测方法时，根据类的功能进行判定。非空非注释代码行数LOC>10的方法进行单元测试，其他方法不进行单元测试。

决定单元测试次序的依据是《软件概要设计说明》中的软件层次架构图，按照被调用关系，从最底层的类开始测试，逐步向上，直至所有类测试完毕。主要采用白盒测试的测试方法进行用例设计，并以黑盒测试方法（边界值法等）进行补充。

测试方法：

- 保证所有的语句、分支被覆盖；
- 参考等价类划分方法；
- 参考边界值分析方法；
- 参考使用错误猜测方法；
- 参考程序插装的测试方法；
- 测试脚本使用Java语言实现，并且和驱动（Drive）代码、桩（Stub）代码构成一个可执行系统；
- 覆盖率分析使用Logicscope工具；
- 某函数的缺陷被修正后必须回归与该函数相关的所有单元测试用例。

2 测试用例与分析

根据《Imagine 图片分享网站单元测试计划书》中的内容，单元测试的测试项目如：

2.1 数据库模块

设计类标识：LLC_001

2.1.1 被测方法参考表：

方法标识符	方法名	代码行（LOC）
LLC_001_FUN_001	openDataBase(void)	11
LLC_001_FUN_002	modifyDataBase(const string & querySQL)	22
LLC_001_FUN_003	runSQL(const string & querySQL, QueryResult & res)	25
LLC_001_FUN_004	clearResult(QueryResult & res)	9

2.1.2 不被测方法表：

方法标识符	方法名	代码行（LOC）
LLC_001_FUN_005	DataBase()	4
LLC_001_FUN_006	~DataBase()	4
LLC_001_FUN_007	closeDataBase(void)	4
LLC_001_FUN_008	rmDataBase(void)	4
LLC_001_FUN_009	createDataBase(void)	5

2.1.3 用例分析与设计

1) openDataBase 测试分析与设计

略

2) modifyDataBase 测试分析与设计

(1) 标识符定义：UT_TD_001_002

(2) 被测特性：

(3) 测试方法：

a. 强健壮等价类法

采用等价类的测试方法，对插入收入项目这一功能进行测试，按照输入变量划分等价类。

输入变量有：（ID为自动设置），日期，项目名称，金额，项目类型代码，项目备注。

将等价类划分如下：（蓝色为无效情况）

日期D1:

D11={19800101~20991231内的合法日期}；
D12={19800101~20991231内的不合法日期}；
D13={小于19800101的日期}；
D13={大于20991231的日期}；

项目名称D2：

D21={中文字符串}；
D22={英文字符串}；
D23={中英文混合字符串}；
D24={空}；

金额D3：

D31={正整数，用+标识}；
D32={正数且两位小数，用+标识}；
D33={正数且两位以上小数，用+标识}；
D34={正整数，不用+标识}；
D35={正数且两位小数，不用+标识}；
D36={正数且两位以上小数，不用+标识}；
D37={负整数，用-标识}；
D38={负数且两位小数，用-标识}；
D39={负数且两位以上小数，用-标识}；

项目类型代码D4:

D41={整数}；
D41={小数}；

项目备注D5：

D51={中文字符串}；
D52={英文字符串}；
D53={中英文混合字符串}；
D54={空}；

b. 基路径测试法

采用基路径的测试方法，对插入收入类型这一功能进行测试，以源代码作为基础，导出程序图，如图1—1—2所示：

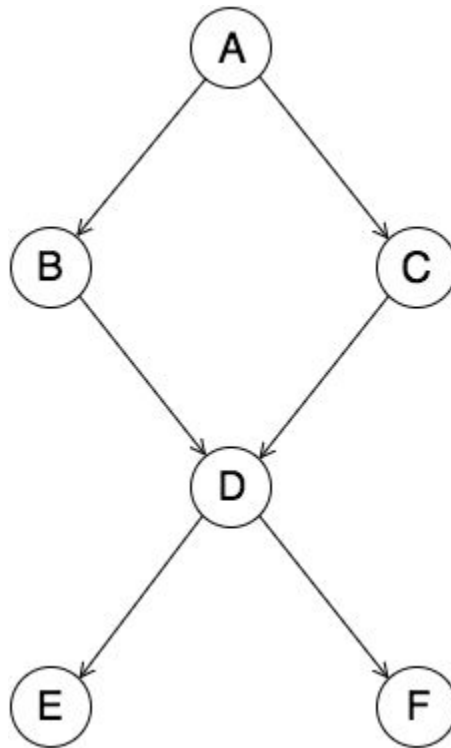


图1-1-1 Database::modifyDatabase 程序图

依据 $V_G = e - n + 2 = 7 - 6 + 2 = 3$ ，基路径中应该包含3条独立的路径，选择一条作为基线路径并由此构成一组基路径，如下：

- 路径1: A -> B -> D -> E
- 路径2: A -> C -> D -> E
- 路径3: A -> B -> D -> F

满足上述三条基路径的测试用例如下，详细测试用例参见UT_TD_001_002.xls

路径	输入		输出
	TypeID	TypeName	
1	不为零	不为空的字符串	插入成功
2	为零	不为空的字符串	插入失败
3	不为零	空	插入失败

(4) 测试用例：UT_TD_001_002.xls

(5) 测试通过标准：所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

2.2 收入支出模块

设计类标识：LLC_002

2.2.1 被测方法参考表

方法标识符	方法名	代码行 (LOC)
LLC_002_FUN_001	bool setValue(Table & IncomeTable)	38
LLC_002_FUN_002	bool removeValue(const int & ID)	12
LLC_002_FUN_003	bool setType(Type & IncomeType)	21
LLC_002_FUN_004	float getSum(const int & beginTime, const int endTime);	9
LLC_002_FUN_005	float getSumIncome(const int & beginTime, const int endTime)	10
LLC_002_FUN_006	float getSumExpense(const int & beginTime, const int endTime)	10
LLC_002_FUN_007	float getSumByType(const int & beginTime, const int endTime, const int & typeId);	13

2.2.2 不被测方法参考表

方法标识符	方法名	代码行 (LOC)
LLC_002_FUN_007	Income()	4
LLC_002_FUN_008	~Income()	4
LLC_002_FUN_009	string getErrorMessage()	4
LLC_002_FUN_010	float getSum()	4
LLC_002_FUN_011	int getTypeId(const string & typeName)	19
LLC_002_FUN_012	string getTypeName(const int & ID)	19

LLC_002_FUN_013	float Sum(const string & sql)	20
LLC_002_FUN_014	float getSumIncome()	4
LLC_002_FUN_015	float getSumExpense()	4
LLC_002_FUN_016	float getSumByType(const int & typeId)	7

2.2.3 用例设计与分析

1) setValue 测试分析与设计

(1) 标识符定义：UT_TD_002_001

(2) 被测特性：

- 输入日期不合法或不在指定范围内，数据库修改失败；
- 输入项目名称为空，数据库修改失败；
- 输入项目类型代码为小数时，数据库修改失败；
- 输入项目备注为空，数据库修改失败；
- 输入参数合法，数据库修改成功。

(3) 测试方法：

a. 基路径测试法：

以源代码作为基础，导出程序图，如图2-1-1所示：

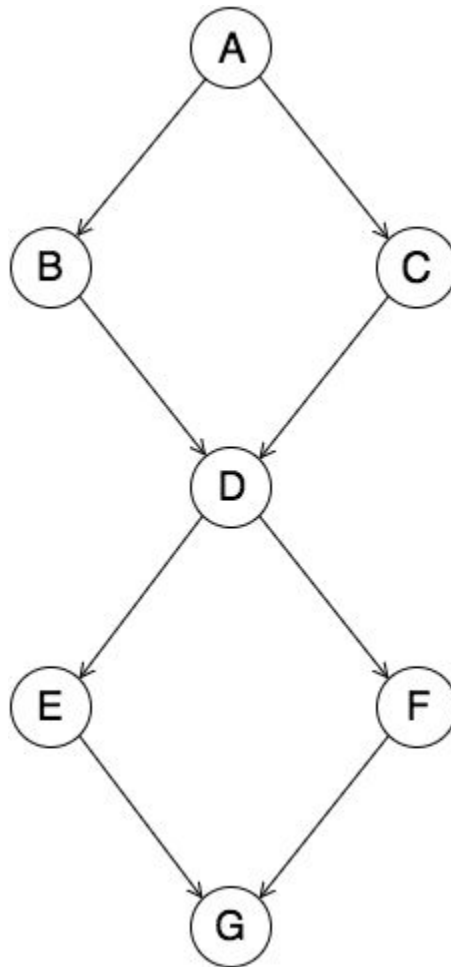


图2-1-1 Income::setValue 程序图

依据 $V_G = e - n + 2 = 8 - 7 + 2 = 3$ ，基路径中应该包含3条独立的路径，选择一条作为基线路径并由此构成一组基路径，如下：

- 路径1: A -> B -> D -> E -> G
- 路径2: A -> C -> D -> E -> G
- 路径3: A -> B -> D -> F -> G

满足上述三条基路径的测试用例如下，详细测试用例参见UT_TD_002_001.xls

路径	输入		输出
	ID	remark备注	
1	不为零	不为空的字符串	插入成功
2	为零	不为空的字符串	插入失败
3	不为零	空	插入失败

b. 强健壮等价类法：

采用强健壮等价类的方法进行测试，根据输入变量划分等价类。

输入变量有：（ID为自动设置），日期，项目名称，金额，项目类型代码，项目备注。

将等价类划分如下：（蓝色为无效情况）

日期D1:

- D11={19800101~20991231内的合法日期}；
- D12={19800101~20991231内的不合法日期}；
- D13={小于19800101的日期}；
- D13={大于20991231的日期}；

项目名称D2：

- D21={中文字符串}；
- D22={英文字符串}；
- D23={中英文混合字符串}；
- D24={空}；

金额D3：

- D31={正整数，用+标识}；
- D32={正数且两位小数，用+标识}；
- D33={正数且两位以上小数，用+标识}；
- D34={正整数，不用+标识}；
- D35={正数且两位小数，不用+标识}；
- D36={正数且两位以上小数，不用+标识}；
- D37={负整数，用-标识}；
- D38={负数且两位小数，用-标识}；
- D39={负数且两位以上小数，用-标识}；

项目类型代码D4:

- D41={整数}；
- D41={小数}；

项目备注D5：

- D51={中文字符串}；
- D52={英文字符串}；
- D53={中英文混合字符串}；
- D54={空}；

(4) 测试用例：参见 UT_TD_002_001.xls

(5) 测试通过／失败标准：

所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

2) removeValue 测试分析与设计

(1) 标识符定义：UT_TD_002_002

(2) 被测特性：

- 要删除的ID超过已存在记录范围，删除失败；
- 要删除的ID为小数，删除失败；
- 要删除的ID为负数，删除失败；
- 要删除的ID合法，删除成功。

(3) 测试方法：

a. 条件覆盖法

以源代码作为基础，分析每个条件可能的取值：

- ID = 0, 取真值, 记为T ;
- ID <> 0, 取假值, 记为F。

给出两个测试用例, 详细测试用例参见UT_TD_002_002.xls

路径	输入ID	输出
1	合法	删除成功
2	不合法	删除失败

b. 边界值法

采用边界值的测试方法, 执行UT_TD_002_001.xls之后, Income表中共有81条收入支出记录, 因此ID为 [1, 81] 范围内的整数,

由此取得5个有效值 :

$D_{min}=1, D_{min+}=3, D_{nom}=49, D_{max-}=77, D_{max}=81;$

无效值为 :

$D=\{0, 100, -1, 1.34\}$

(4) 测试用例 : 参见UT_TD_002_002.xls

(5) 测试通过/失败标准 :

所有的用例都被执行, 且测试结果与预期结果相同为通过, 否则失败。

3) setType 测试分析与设计

(1) 标识符定义 : UT_TD_002_003

(2) 被测特性 :

- 输入项目类型ID为负数时, 新建项目类型失败 ;
- 输入项目类型ID为小数时, 新建项目类型失败 ;
- 输入项目类型名称为空时, 新建项目类型失败 ;
- 输入参数合法时, 新建项目类型成功。

(3) 测试方法 :

a. 基路径测试法

以源代码作为基础, 导出程序图, 如图2-1-2所示 :

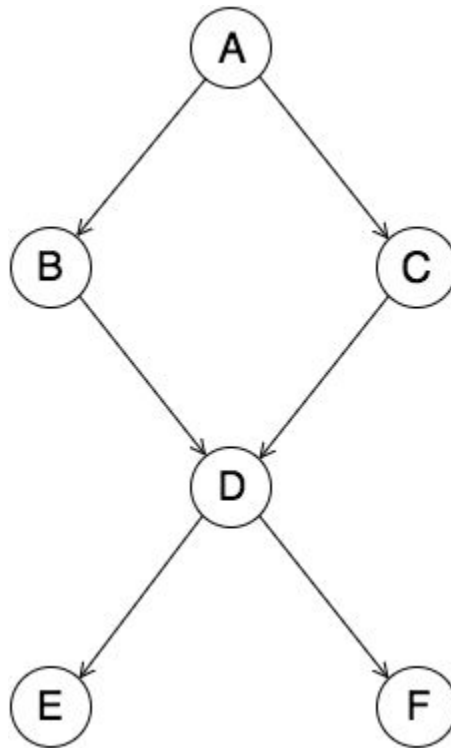


图2-1-1 Income::setValue 程序图

依据 $V_G = e - n + 2 = 7 - 6 + 2 = 3$ ，基路径中应该包含3条独立的路径，选择一条作为基线路径并由此构成一组基路径，如下：

- 路径1: A -> B -> D -> E
- 路径2: A -> C -> D -> E
- 路径3: A -> B -> D -> F

满足上述三条基路径的测试用例如下，详细测试用例参见UT_TD_002_003.xls

路径	输入		输出
	TypeID	TypeName	
1	不为零	不为空的字符串	插入成功
2	为零	不为空的字符串	插入失败
3	不为零	空	插入失败

b. 强健壮等价类法：

采用强健壮等价类的方法进行测试，根据输入变量划分等价类。

输入变量有：项目类型ID，项目类型名称

将等价类划分如下：（蓝色为无效情况）

项目类型代码D1:

D11={整数}；

D12={小数}；

D13={负数} ;

项目名称D2 :

D21={中文字符串} ;

D22={英文字符串} ;

D23={中英文混合字符串} ;

D24={空} ;

(4) 测试用例：参见UT_TD_002_003.xls

(5) 测试通过/失败标准：

所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

4) getSum 测试分析与设计

(1) 标识符定义：UT_TD_002_004

(2) 被测特性：

- 开始时间不合法，查询失败；
- 开始时间不在1980/01/01~2099/12/31范围内，查询失败；
- 结束时间不合法，查询失败；
- 结束时间不在1980/01/01~2099/12/31范围内，查询失败；
- 开始时间晚于结束时间，查询失败；
- 输入参数合法，查询成功。

(3) 测试方法：

采用等价类的测试方法，按照输入变量划分等价类。

输入变量有：开始时间begin，结束时间end。划分等价类如下：（蓝色为无效情况）

D1= {begin<=end} ；

D2= {begin>end} ；

D1这个等价类包含合法查询与不合法查询两种情况，进一步划分等价类如下：

D11= {begin<=end, begin和end均为1980/01/01~2099/12/31内合法日期}

D12= {begin<=end, begin为不合法日期, end符合要求}

D13= {begin<=end, begin为小于1980/01/01的日期, end符合要求}

D14= {begin<=end, begin为大于2099/12/31的日期, end符合要求}

D15= {begin<=end, end为不合法日期, begin符合要求}

D16= {begin<=end, end为小于1980/01/01的日期, begin符合要求}

D17= {begin<=end, end为大于2099/12/31的日期, begin符合要求}

(4) 测试用例：参见UT_TD_002_004.xls

(5) 测试通过/失败标准：

所有测试用例都执行，且测试结果与预期相同为通过，否则失败。

3 单元测试缺陷汇总

测试项编号	UT_TD_001_001_047
优先级	中

测试项描述	将消费条目插入数据库			
采用测试方法	等价类方法			
用例序号	输入	期望结果	实际结果	判定
001	sql="INSERT * FROM";	正常插入	项目名称中间有空格，导致条目错位	true

测试项编号	UT_TD_001_003_008			
优先级	中			
测试项描述	在数据库中查询不存在条目			
采用测试方法	强健壮边界值			
用例序号	输入	期望结果	实际结果	判定
001	sql=" SELECT* FROM";	正常插入	直接输出0	true

测试项编号	UT_TD_002_001_082			
优先级	中			
测试项描述	修改数据库，项目备注为空			
采用测试方法	基路径			
用例序号	输入	期望结果	实际结果	判定
001	sql=" INSERT* FROM";	正常修改	原程序报错，但备注为空在本项目中不算错误	true

测试项编号	UT_TD_002_001_083			
优先级	中			
测试项描述	修改数据库，TypeID为小数			
采用测试方法	基路径+边界值			
用例序号	输入	期望结果	实际结果	判定
001	sql=" INSERT* FROM";	正常修改	原代码中没有判断TypeID为小数的部分，导致后续部分出错	true

测试项编号	UT_TD_002_002_007			
优先级	中			
测试项描述	根据ID删除数据库中条目			
采用测试方法	条件覆盖法			
用例序号	输入	期望结果	实际结果	判定
001	sql=" DELETE* FROM";	正常修改	原代码中把0当作正常值0	true

测试项编号	UT_TD_002_002_008			
优先级	中			
测试项描述	根据ID删除数据库中条目			
采用测试方法	边界值			
用例序号	输入	期望结果	实际结果	判定

001	sql=" DELETE* FROM";	正常修改	原代码中对ID没有最大值的限制	true
-----	----------------------	------	-----------------	------

测试项编号	UT_TD_002_002_010			
优先级	中			
测试项描述	根据ID删除数据库中条目			
采用测试方法	边界值			
用例序号	输入	期望结果	实际结果	判定
001	sql=" DELETE* FROM";	正常修改	原代码中没有限制ID为小数	true

测试项编号	UT_TD_002_003_016			
优先级	中			
测试项描述	输入ID，新建消费类型			
采用测试方法	基路径			
用例序号	输入	期望结果	实际结果	判定
001	sql=" INSERT* FROM";	正常修改	原代码中直接把小数取整	true

测试项编号	UT_TD_002_00_017			
优先级	中			
测试项描述	输入ID，新建消费类型			
采用测试方法	基路径			

用例序号	输入	期望结果	实际结果	判定
001	sql=" INSERT* FROM";	正常修改	原代码没有限制 TypeID是否为负数	true

测试项编号	UT_TD_002_004_006			
优先级	中			
测试项描述	查询给定时间段内的总消费			
采用测试方法	等价类+边界值			
用例序号	输入	期望结果	实际结果	判定
001	sql=" SELETE* FROM";	正常修改	原代码中没有对最小时间的限制	true

测试项编号	UT_TD_002_004_011			
优先级	中			
测试项描述	查询给定时间段内的总消费			
采用测试方法	等价类+边界值			
用例序号	输入	期望结果	实际结果	判定
001	sql=" SELETE* FROM";	正常修改	原代码没有判断起止时间大小的模块	true

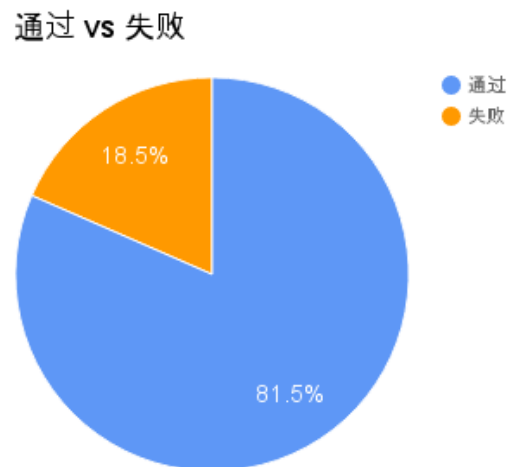
4 单元测试缺陷分析

4.1 计划的用例数／执行的用例数

4.2 通过的用例数／失败的用例数

计划的用例数	执行的用例数	通过	失败
276	276	221	55

图1 通过用例数 VS 失败用例数



4.3 缺陷等级划分标准

Critical 严重缺陷 应用系统崩溃或系统资源使用严重不足：

- 1、系统停机(含软件、硬件)或非法退出，且无法通过重启恢复；
- 2、系统死循环；
- 3、数据库发生死锁或程序原因导致数据库断连；
- 4、系统关键性能不达标。
- 5、数据通讯错误或接口不通
- 6、错误操作导致程序中断

Major 较严重缺陷 系统因软件严重缺陷导致下列问题：

- 1、重要交易无法正常使用、功能不符合用户需求；
- 2、重要计算错误；
- 3、业务流程错误或不完整；
- 4、使用某交易导致业务数据紊乱或丢失；
- 5、业务数据保存不完整或无法保存到数据库。
- 6、周边接口出现故障(需考虑接口时效/数量等综合情况)；
- 7、服务程序频繁需要重启(每天2次或以上)；
- 8、批处理报错中断导致业务无法正常开展。
- 9、前端未合理控制并发或连续点击动作，导致后台服务无法及时响应。
- 10、在产品声明支持的不同平台下，出现部分重要交易无法使用或错误。

Medium 一般性缺陷 系统因软件一般缺陷导致下列问题：

- 1、部分交易使用存在问题，不影响业务继续开展，但造成使用障碍。
- 2、初始化未满足客户要求或初始化错误
- 3、功能点能实现，但结果错误；
- 4、数据长度不一致；
- 5、无数据有效性检查或检查不合理；
- 6、数据来源不正确；
- 7、显示/打印的内容或格式错误；
- 8、删除操作不给提示；
- 9、个别交易系统反应时间超出正常合理时间范围
- 10、日志记录信息不正确或应记录而未记录
- 11、在产品声明支持的不同平台下，出现部分一般交易无法使用或错误。

Cosmetic 较小缺陷 系统因软件操作不便方面缺陷：

- 1、系统某些查询、打印等实时性要求不高的辅助功能无法正常使用；
- 2、界面错误
- 3、菜单布局错误或不合理
- 4、焦点控制不合理或不全面；
- 5、光标,滚动条定位错误；
- 6、辅助说明描述不准确或不清楚；
- 7、提示窗口描述不准确或不清楚；
- 8、日志信息不够完整或不清晰，影响问题诊断或分析的；
- 9、缺少产品使用、帮助文档、系统安装或配置方面需要信息；

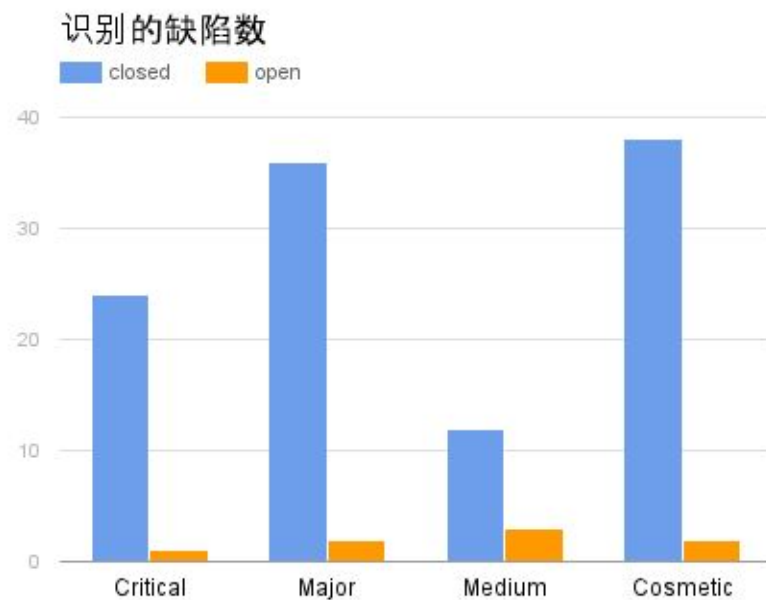
- 10、联机帮助、脱机手册与实际系统不匹配
- 11、系统版本说明不正确;
- 12、长时间操作未给用户进度提示;
- 13、提示说明未采用行业规范语言;
- 14、显示格式不规范
- 15、界面不整齐
- 16、软件界面、菜单位置、工具条位置、相应提示不美观，但不影响使用

4.4 识别的缺陷数—按测试类型

	单元测试	集成测试	系统测试	总计
Closed	270	153	10	433
Open	6	0	0	6

4.5 识别的缺陷数及其状态

	Critical	Major	Medium	Cosmetic	Total
Closed	3	0	8	7	18
Open	0	0	0	2	2



4.6 缺陷分布—按模块划分

	数据库模块	收入支出模块	投资负债模块	界面模块
Critical	0	1（类型名输入小数）	1（类型名输入小数）	1(7)
Major	0	0	0	0
Medium	2(1,2)	2(4: 数据长度不一致 5)	2(4: 数据长度不一致 5)	2(2,4)
Cosmetic	3(1:存储失败不提示 7:*2)	1(12)	1(12)	4(1,7,12,16)
Total	5	4	4	7

缺陷分布

