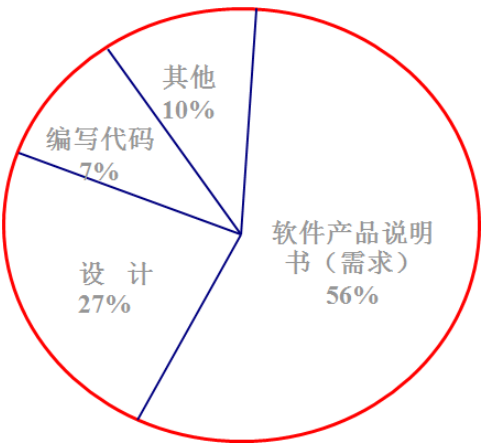


测试用例和测试报告管理

一，缺陷来源

为什么会出现软件缺陷？

- (1) 导致软件缺陷最大的原因是产品说明书。
- (2) 软件缺陷的第二大来源是设计方案。
- (3) 编写代码
- (4) 其他



缺陷类型

缺陷类型编号	缺陷类型	描述
10	F-功能	如逻辑，指针，循环，递归，功能等缺陷
20	G-语法	拼写、标点符号、打字
30	A-赋值	如声明、重复命名，作用域
40	I-接口	与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表相互影响的缺陷
50	B-联编打包	由于配置库、变更管理或版本控制引起的错误
60	D-文档	需求、设计类文档
70	U-用户接口	人机交互特性：屏幕格式，确认用户输入，功能有效性
80	P-性能	不满足系统可测量的属性值，如：执行时间，事务处理速率等
90	N-标准	不符合各种标准的要求，如编码标准、设计符号等
100	E-环境	设计、编译、其他支持系统问题

缺陷严重程度

编号	缺陷严重等级	描述
----	--------	----

1	严重缺陷 (Critical)	不能执行正常工作功能或重要功能。
2	较大缺陷 (Major)	严重地影响系统要求或基本功能的实现，且没有办法更正。（重新安装或不属于更正办法）
3	较小缺陷 (Minor)	严重地影响系统要求或基本功能的实现，但存在合理的更正办法。（重新安装该软件不属于更正办法）
4	轻微缺陷 (Cosmetic)	使操作者不方便或遇到麻烦，但它不影响执行工作功能或重要功能。
5	其他缺陷 (Other)	其它错误

二，测试用例管理

什么是黑盒测试

- （1）黑盒测试法把程序看成一个黑盒子，完全不考虑程序内部结构和处理过程。
- （2）黑盒测试是在程序接口进行测试，它只是检查程序功能是否按照规格说明书的规定正常使用。
- （3）黑盒测试又称功能测试。

什么是测试用例？

测试用例是为特定的目的而设计的一套测试方案，包括测试环境、操作步骤、测试输入数据和预期的结果，实际结果。

测试用例的特征：

- （1）最有可能抓住错误的；
- （2）不是重复的、多余的；
- （3）一组相似测试用例中最有效的；
- （4）既不是太简单，也不是太复杂。

为什么做测试用例

完全测试是不可能的

- （1）输入量太大；
- （2）输出结果太多；
- （3）软件实现途径太多；
- （4）软件说明书没有客观标准。从不同角度，软件缺陷的标准不同。

设计测试用例的基本准则

我们不可能进行穷举测试，为了节省时间和资源、提高测试效率，必须从数量极大的可用测试

数据中精心挑选出具有代表性或特殊性的测试数据来进行测试

测试用例的代表性

能够代表并覆盖各种合理的和不合理的、合法的和非法的、边界的和越界的以及极限的输入数

据、操作和环境设置等

测试结果的可判定性

即测试执行结果的正确性是可判定的，每一个测试用例都应有相应的期望结果。

测试结果的可再现性

即对同样的测试用例，系统的执行结果应当是相同的。

黑盒测试内容：

主要针对软件功能、软件界面、外部数据库访问、软件初始化等方面进行测试。

测试用例设计依据：

编写黑盒测试用例时，需要以《软件需求分析文档》为依据

黑盒测试用例主要设计方法：

- (1) 等价类划分
- (2) 边界值分析
- (3) 错误猜测法
- (4) 因果图
- (5) 判定决策表

三，缺陷情况分析

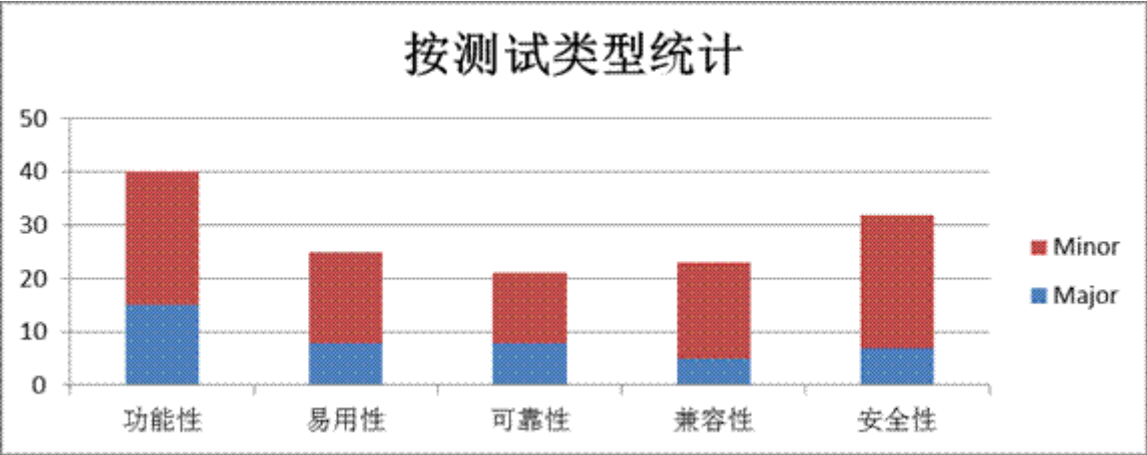
缺陷修复率标准

- (1) 一、二级错误修复率应达到100%
- (2) 三、四级错误修复率应达到80%以上
- (3) 五级错误修复率应达到60% 以上

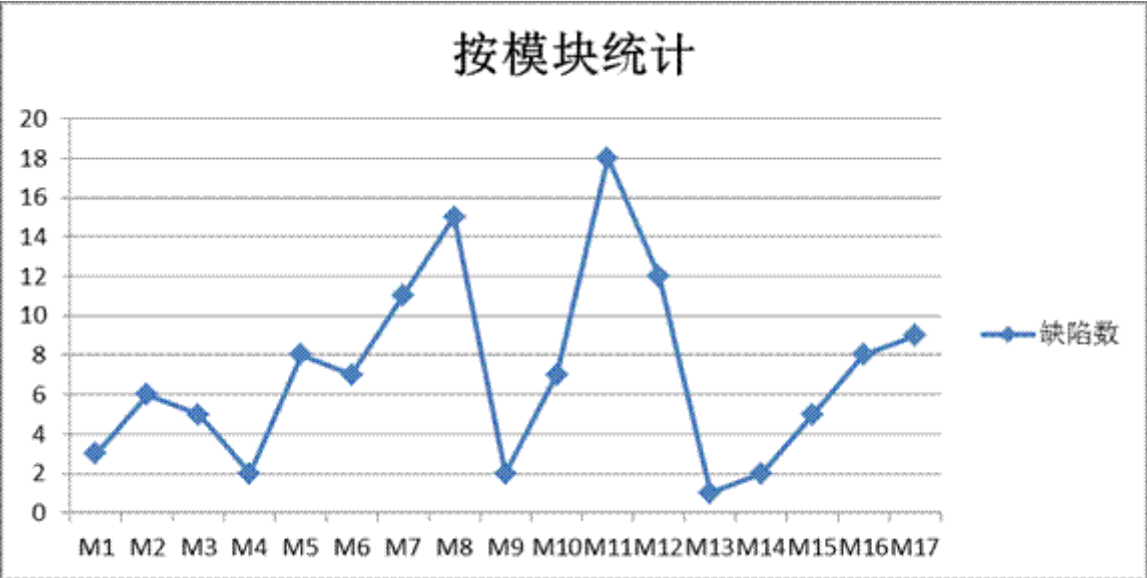
缺陷分析

(1) 缺陷分布情况分析

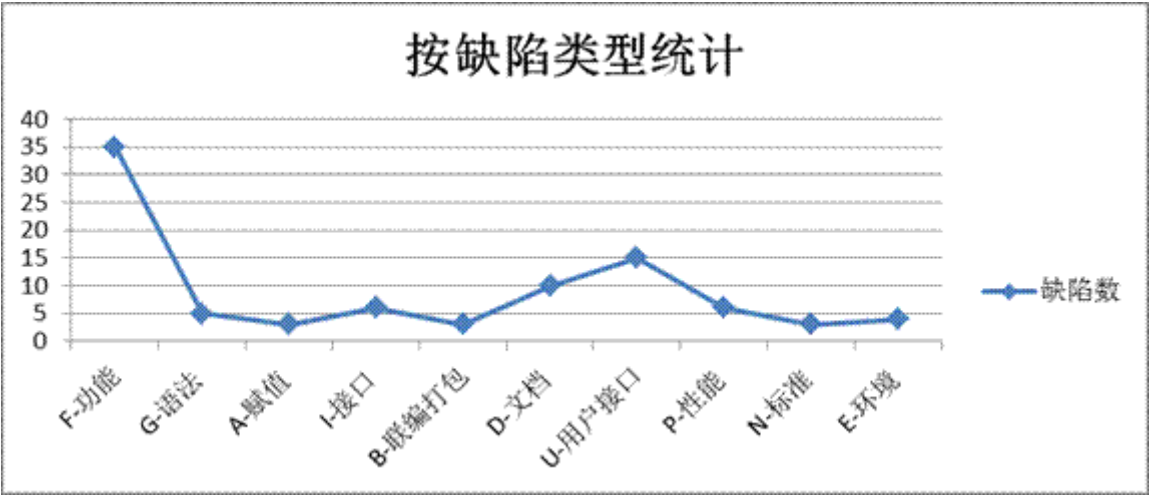
按照测试类型统计



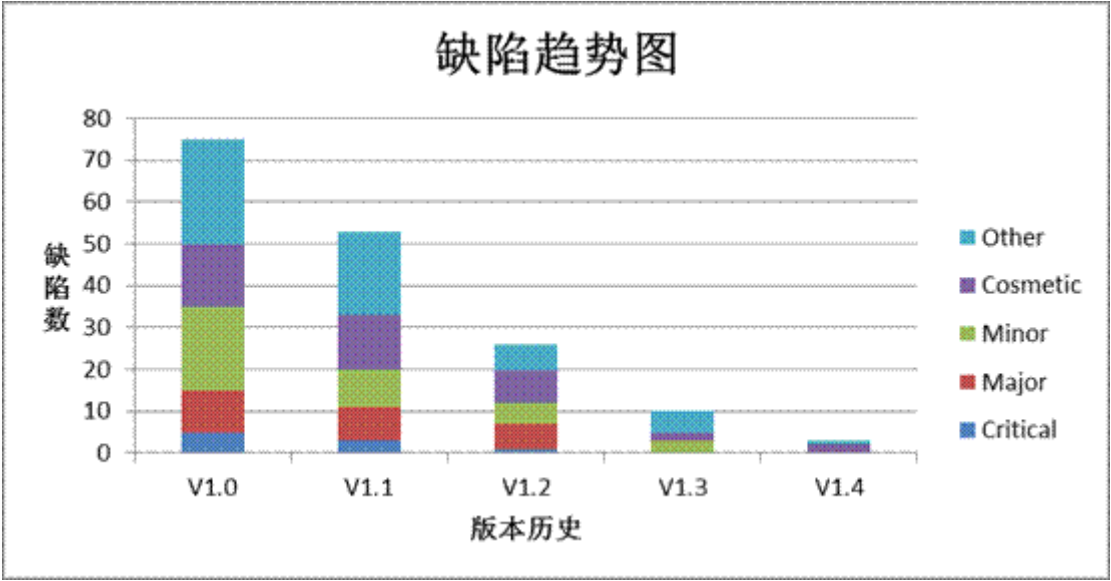
按照功能模块分布



按照缺陷类型分布



缺陷趋势图



四，测试报告模板