

目录

A. 整体统计.....	1
1、 项目缺陷数统计.....	1
B. 项目统计.....	1
1、 版本缺陷数统计.....	2
2、 模块缺陷数统计.....	2
3、 缺陷严重程度统计.....	2
4、 缺陷状态统计.....	2
5、 缺陷激活次数统计.....	2
6、 缺陷类型统计.....	2
7、 每人提交的缺陷数统计.....	2
8、 每人关闭的缺陷数统计.....	2
9、 指派给每人的缺陷数统计.....	2
10、 每人解决的缺陷数统计.....	3
11、 缺陷解决方案统计.....	3
12、 是否确认统计.....	3
C. 回归统计.....	3
1、 模块缺陷数统计.....	3
2、 缺陷严重程度统计.....	3
3、 缺陷增减, 重激活等状态变化统计.....	3
4、 每人新增缺陷数统计.....	3
5、 指派给每人的缺陷数统计.....	3
6、 是否确认统计.....	3
D. 个人统计.....	3
1、 模块缺陷数统计.....	4
2、 缺陷严重程度统计.....	4
3、 缺陷状态统计.....	4
4、 缺陷激活次数统计.....	4
5、 缺陷解决方案统计.....	4
特别说明.....	4

A. 整体统计

站在整体的视角对所有项目进行简单的统计

1、 项目缺陷数统计

统计每个项目的缺陷数量, 每个项目的缺陷占比

统计价值: 理论上, 项目缺陷数越少, 说明项目质量越高, 工作效率越高, 反之越低。

B. 项目统计

对单个项目进行统计与分析

1、 版本缺陷数统计

统计每个版本的缺陷数量，每个版本的缺陷占比

统计价值：理论上，随着版本的不断迭代，缺陷数应该越来越少。当然不排除需求变更，导致版本缺陷数突然上升。通过统计数据，可以看到版本缺陷数占比，大致的变化趋势，进而分析产品质量变化趋势，同时也可能获得其它信息，比如产品需求把控能力。

2、 模块缺陷数统计

统计每个模块的缺陷数量，每个模块的缺陷占比

统计价值：了解缺陷的分布情况，模块代码质量，对模块质量风险有个比较好的把握；某种程度上也体现了测试覆盖度，测试广度。

3、 缺陷严重程度统计

统计不同严重级别的缺陷数量，每种严重级别的缺陷占比

统计价值：缺陷的严重级别，某种程度可以体现开发的代码质量，工作质量；同时也体现了测试人员的测试深度，测试价值，对产品质量的重视程度。

严重级别：致命，严重，一般，轻微，建议

4、 缺陷状态统计

统计不同状态的缺陷数量，每种状态的缺陷占比

统计价值：统计项目残留缺陷数，结合缺陷严重程度，可为产品风险分析提供参考数据。

状态：激活，已关闭，已解决

5、 缺陷激活次数统计

统计重新激活的缺陷数量，不同激活次数的缺陷占比

统计价值：理论上，缺陷激活次数越多，代码质量越低，工作效率越低，进而体现了开发人员的工作态度，代码质量，效率。

6、 缺陷类型统计

统计不同类型的缺陷数量，不同类型的缺陷占比

统计价值：挖掘缺陷的来源，理论上，缺陷一直都会有，我们要不能只找缺陷，还要找缺陷的源头，找到后对症下药。

7、 每人提交的缺陷数统计

统计每人提交的缺陷数量，每人提交的缺陷占比

统计价值：缺陷数某种程度也提现了测试人员的付出

8、 每人关闭的缺陷数统计

统计每人关闭的缺陷数量，每人关闭的缺陷占比

统计价值：关闭数量越多，回归缺陷数越多，投入也越多，结合提交的缺陷数，上容易分析测试人员对缺陷的跟踪情况。

9、 指派给每人的缺陷数统计

统计指派给每人的缺陷数量，指派给每人的缺陷数占比

统计价值: 缺陷数某种程度也是工作量的一种体现。

10、每人解决的缺陷数统计

统计每人解决的缺陷数量, 每人解决的缺陷数占比

统计价值: 结合上面 指派给每人的缺陷数统计, 直观的提现开发人员的工作态度, 效率等。

11、缺陷解决方案统计

统计不同解决方案的缺陷数量, 每种解决方案的缺陷数占比

统计价值: 某种程度体现了开发人员对缺陷的态度, 对工作的态度, 工作质量等
解决方案: 延迟处理, 拒绝处理等

12、是否确认统计

统计是否确认的缺陷数量

统计价值: 理论上, 未确认缺陷数越多, 说明开发对缺陷越不重视, 处理越不及时。

C. 回归统计

统计本次测试完成后的缺陷情况

1、模块缺陷数统计

统计未关闭状态的缺陷中, 每个模块的缺陷数量, 每个模块的缺陷占比

2、缺陷严重程度统计

统计未关闭状态的缺陷中, 不同严重级别的缺陷数量, 每种严重级别的缺陷占比

严重级别: 致命, 严重, 一般, 轻微, 建议

未关闭缺陷总数: xxx 个, 详情如下:

致命: xx 个, 严重: xx 个, 一般: xx 个, 轻微: xx 个, 建议: xx 个

3、缺陷增减, 重激活等状态变化统计

统计本次测试回归的缺陷数量, 缺陷关闭、新增、重新激活的缺陷数

缺陷回归总数: xx 个 关闭缺陷数: xx 个 重新激活缺陷数: xx 个

新增缺陷数: xx 个 转需求缺陷数: xx 个

说明: 如果时间允许, 建议细化: 针对新增缺陷, 按严重级别统计缺陷数

4、每人新增缺陷数统计

统计本次测试, 每个人新提交的缺陷数量, 每人提交的缺陷占比

5、指派给每人的缺陷数统计

统计未关闭状态的缺陷中, 指派给每人的缺陷数量, 指派给每人的缺陷数占比

6、是否确认统计

统计未关闭状态的缺陷中, 是否确认的缺陷数量

D. 个人统计

统计每个开发人员的缺陷情况

建议:

- 1、短期回归测试统计中, 建议仅统计未关闭缺陷; 长期阶段性统计中, 建议统计所有状态的缺陷
- 2、开发人员多的情况下, 统计可能比较耗时, 可能不是那么容易做到, 所以, 一般建议在仅阶段性统计中进行分析

1、 模块缺陷数统计

#统计某开发人员负责模块的缺陷数量, 缺陷占比等

2、 缺陷严重程度统计

#统计某开发人员的不同严重级别的缺陷数量, 缺陷占比等

3、 缺陷状态统计

#统计某开发人员的不同状态的缺陷数量, 缺陷占比等

4、 缺陷激活次数统计

#统计某开发人员的重新激活的缺陷数量, 缺陷占比等

5、 缺陷解决方案统计

统计某开发人员的不同解决方案的缺陷数量, 缺陷占比等

#短期回归测试统计中, 建议不做该统计(如果只统计未关闭缺陷)

特别说明

- 1、 缺陷数据统计是需要花时间的, 特别是纯手工统计的情况下, 需要花费更多的时间, 所以效率起见, 建议**认真选择一款好工具**, 笔者在这里推荐大家用禅道
- 2、 不是所有工具都会提供所有你想要的统计功能, 有的工具可能不提供、或者仅提供上述中的部分, 或者提供的统计功能存在功能缺陷, 不能用, 所以针对选择的工具, 适当的对上述统计进行裁剪, 当然, 也可以考虑手工统计
- 3、 不是每次统计都要包含上述所有统计, 可以根据测试阶段进行适当的裁剪, 比如, 回归测试统计, 建议【回归统计】+【个人统计】或者仅进行【回归统计】; 类似里程碑这样的阶段性测试统计, 建议【整体统计】+【项目统计】+【个人统计】
- 4、 数据来源缺陷, 所以, 要想正确的度量, 请务必对缺陷相关字段进行规范性设置, 同时提交缺陷时, 规范的填写缺陷, 相关人员按规范操作。
- 5、 短期回归统计中, 有人可能会想, 要是统计完成后又发现缺陷咋办?
解答: 可以增加每日统计, 即对每天新增缺陷数做个反馈(如果有进行测试的话), 简单的, 反馈每日新增缺陷总数, 当然如果时间允许的话, 也可以继续细化, 按严重级别分类统计。

个人很推荐每日对测试情况做个简单的反馈(不用另外出个文档报告)