

软件测试 Software Testing

25. 软件测试和质量分析报告

程适

cheng@snnu.edu.cn

计算机科学学院

2016 年 12 月 23 日



陕西师范大学
SHAANXI NORMAL UNIVERSITY

Outline

- 软件产品的质量度量
- 评估系统测试的覆盖程度
- 基于缺陷分析的产品质量评估
- 测试报告的具体内容

软件测试和质量分析报告

- 借助测试报告，公司管理人员和用户就能够了解该项目的测试是如何进行的，确定软件产品是否得到足够的测试以及对测试结果是否满意等。
- 测试报告需要对整个测试过程进行检查、评估和分析，并对当前的软件产品质量进行评估，从而针对产品是否达到发布的质量给出结论。

软件测试和质量分析报告

- 单元测试采用什么方法和工具？
- 代码行覆盖率是否达到所设定的目标？
- 集成测试是否全面验证了所有接口及其参数？
- 测试用例是否经过了开发人员、产品经理的严格评审？

软件测试和质量分析报告

- 系统测试是否包含了性能、兼容性、安全性、恢复性等各项测试？
如果执行了，又是怎么进行的？结果如何？
- 是否完成了测试计划所要求的各项测试内容？
- 需要执行的测试用例是否百分之百地完成了？
- 所有严重的 Bug 都修正了？

软件产品的质量度量

- 1 软件度量是根据一定的规则，将数字或符号赋予系统、构件、过程等实体的特定属性，从而能清晰地理解该软件实体及其属性的量化表示。
- 2 软件度量就是对软件所包含的各种属性的量化表示。
- 3 软件度量可以提供深入了解软件过程和产品的衡量指标，使组织能够更好地作出决策以达成目标。

软件度量作用

■ 软件度量有如下作用：

- 1 用数据指标表明验收标准；
- 2 监控项目进度和预见风险；
- 3 分配资源时进行量化均衡；
- 4 预计和控制产品的过程、成本和质量。

■ 软件度量是衡量软件过程质量和进行软件过程改进的重要手段。

■ 为了保持数据的可靠性、客观性和准确性，必须保证度量结果不用于评价数据提供者的个人工作绩效或素质。

软件产品的质量度量

- 1 软件度量及其过程
- 2 软件质量的度量
- 3 质量度量的统计方法

软件度量及其过程

- 1 规模度量：代码行数，以千行源代码（KLOC）为基准，它是工作量度量、进度度量的基础。
- 2 复杂度度量：分析和估算软件产品结构及其各部分的复杂度指标，以便选择最可靠的程序设计方法，确定测试策略。
- 3 缺陷度量：获取产品缺陷变化的状态，以指示修复缺陷活动所需的工作量；或分析产品缺陷分布的情况。

软件度量及其过程

■ 目标驱动的软件度量过程主要包括 5 个阶段：

- 1 识别目标。根据管理者的不同要求，分析出度量的工作目标，并根据其优先级和可行性，得到度量活动的工作目标列表，并由管理者审核确认。
- 2 定义度量过程。
- 3 搜集数据。
- 4 数据分析和反馈。
- 5 过程改进。根据度量的分析报告，管理者确定软件开发活动与计划之间时候有偏差，以便控制其执行；或者基于度量数据做出其他决策，这些决策可能包括滚动计划、纠正活动等。

软件度量及其过程

- “识别目标”和“定义度量过程”是保证成果搜索数据和分析数据的先决条件，是度量过程最重要的阶段；
- “过程改进”是度量的最终目的。

有效软件度量的属性

- 1 简单的和可计算的
- 2 经验和直觉上有说服力
- 3 一致的和客观的
- 4 在其单位和维度的使用上是一致的
- 5 编程语言是独立的
- 6 质量反馈的有效机制

软件质量的度量

- 定量的软件评估是通过数学模型来实现的，就是尺度度量（Metrics measurement）的方法，这种定量度量适用于一些能够直接度量的特性，包括软件可靠性度量、复杂度度量、缺陷度量和规模度量，如程序出错率可定义为每千行代码（Kilo Lines of Code, KLOC）所含有的 Bug 数。
- 为了进行质量度量，需要根据质量模型来准备足够的数据，然后进行产品质量的量化评估分析。

软件质量的度量

- 明确性、正确性、可理解性、完全性、可验证性、一致性、简洁性、可追踪性、可修改性、精确性和可复用性的数据可以用来评价分析模型和相应的质量表现特征。
- 公开的可能缺陷数与报告总缺陷数的对比则可以用来评价精确度和测试覆盖度，同时也可以预测项目发布时间。
- 产品发布前清楚的缺陷数在总缺陷数中所占的百分比，有助于评估产品的质量。

软件质量的度量

- 按严重缺陷、子系统缺陷来划分，分类统计出平均修复时间，这样有助于规划纠正缺陷工作。
- 利用测试的统计数据，评估可维护性、可靠性、可用性和原有故障总数等数据。这些数据将有助于评估应用软件的稳定程度和可能产生的失败。

软件质量的度量

$$M_i = c_1 \times f_1 + c_2 \times f_2 + \dots + c_n \times f_n \quad (1)$$

- 1 M_i 是一个软件质量因素；
- 2 $f_1 \sim f_n$ 是影响质量因素的度量值；
- 3 $c_1 \sim c_n$ 是加权因子。

质量度量的统计方法

- 质量度量的统计方法时质量评估量化的常用方法，主要包含以下步骤：
 - 1 收集和分类软件缺陷信息；
 - 2 找出导致每个缺陷的原因；
 - 3 使用 Pareto 规则（80% 缺陷主要是由 20% 的主要因素造成的，20% 缺陷是由另外 80% 的次要因素造成的），要将这 20% 的主要因素分离出来；
 - 4 一旦标出少数的主要因素，就比较容易纠正引起缺陷的问题。

评估系统测试的覆盖程度

- 为什么要做软件测试评估呢？如果没有测试评估，就没有测试覆盖率的结果，就没有报告测试进程的根据。

评估系统测试的覆盖程度

- 软件测试评估主要有两个目的：
 - 1 量化测试进程，判断测试进行的状态，决定什么时候测试可以结束；
 - 2 为编写测试报告或质量分析报告提供所需的数据，如缺陷清除率、测试覆盖率等
- 测试的覆盖率，可以用测试项目的数量和内容进行度量，除此之外，如果测试软件的数量较大，还要考虑数据量。

评估系统测试的覆盖程度

- 1 对软件需求的估算
- 2 基于需求的测试覆盖评估
- 3 基于代码的测试覆盖评估

基于需求的测试覆盖评估

- 在执行测试活动中，评估测试用例覆盖率又可以分为两类测试用例覆盖率估算：
 - 1 确定已经执行的测试用例覆盖率，即在所有测试用例中有多少测试用例已被执行。假定 T_x 已执行的测试过程数或测试用例数， R_{ft} 是测试需求的总数。
已执行的测试覆盖 = T_x/R_{ft}
 - 2 确定成功的测试覆盖，即执行时未出现失败的测试，如没有出现缺陷或意外结果的测试，假定 T_s 是已执行的完全成功、没有缺陷的测试过程数或测试用例数。
成功的测试覆盖 = T_s/R_{ft}

基于需求的测试覆盖评估

- 基于代码的测试覆盖通过以下公式计算：

$$\text{已执行的测试覆盖} = T_c / T_{nc}$$

- T_c 是用代码语句、条件分支、代码路径、数据状态判定点或数据元素名表示的已执行的项目数
- T_{nc} (Total number of items in the code) 是代码中的项目总数

基于缺陷分析的产品质量评估

- 1 缺陷评测的基线
- 2 经典的种子公式
- 3 基于缺陷清除率的估算方法
- 4 软件产品性能评估

缺陷评测的基线

- 1 软件评估首先要建立基线，为软件产品的质量设置起点，在基线的基础上再设置新的目标，作为对软件评估是否通过的标准

经典的种子公式

- 通过已知缺陷（称为种子 Bug）来估计程序当中潜在的、未知的缺陷数量。

已测试出的种子 $Bug(s)$ / 所有的种子 $Bug(S)$

= 已测试出的非种子 $Bug(n)$ / 全部的非种子 $Bug(N)$

- 程序的总 Bug 数为： $N = S * n/s$ ，其中， n 是进行实际测试时发现的 Bug 总数。如果 $n = N$ ，说明所有的 Bug 已找出来，说明做的测试足够充分。

基于缺陷清除率的估算方法

- F 为描述软件规模用的功能点， D_1 为在软件开发过程中所发现的所有缺陷数； D_2 为软件发布后所发现的缺陷数； D 为发现的总缺陷数。因此， $D = D_1 + D_2$
- 质量 = D_2/F
- 缺陷注入率 = D/F
- 整体缺陷清除率 = D_1/D

软件产品性能评估

- 软件产品性能评估其技术性相对比较强，方法的基础是获取与性能表现相关的数据，如响应时间、数据吞吐量、数据流速率、操作可靠性等。
- 主要的性能评测包括：
 - 1 动态监测
 - 2 响应时间/吞吐量
 - 3 百分比报告
 - 4 比较报告
 - 5 追踪报告

测试报告的具体内容

- 1 产品标识
- 2 用于测试的计算机系统
- 3 使用的文档及其标识
- 4 产品描述、用户文档、程序和数据的测试结果
- 5 与要求不符的清单
- 6 针对建议的要求不符的清单，产品未作符合性测试的说明
- 7 测试结束日期

小结

- 软件产品的质量度量
- 评估系统测试的覆盖程度
- 基于缺陷分析的产品质量评估
- 测试报告的具体内容

小结

- 代码和相应的文档是开发人员的主要工作成果，作为软件测试人员，其主要成果就是提交一份数据准确、信息充分、分析透彻的测试报告或质量报告。
- 报告需要对测试进行全面的评估，就是要对已做过的测试和测试结果等方面进行评估。概括起来，就是软件测试覆盖评估和测试结果的质量分析。

小结

- 测试覆盖是对测试完全程度的评测，它建立在测试覆盖基础上，测试覆盖是由测试需求和测试用例的覆盖或已执行代码的覆盖表示的。
- 质量分析是对测试对象（被测系统或软件产品）的可靠性、稳定性以及性能的评测。质量建立在对测试结果的评估和对测试过程中变更请求分析的基础上。

致谢

谢谢，欢迎提问！