

测试用例的定义

从软件测试的定义看测试

- ▶ 根本目的：验证需求
- ▶ 核心关键：测试设计
- ▶ 整体思路：比较预期输出和实际输出
- ▶ 存在不足：没有说明具体如何做测试
- ▶ 对应等级：初始阶段

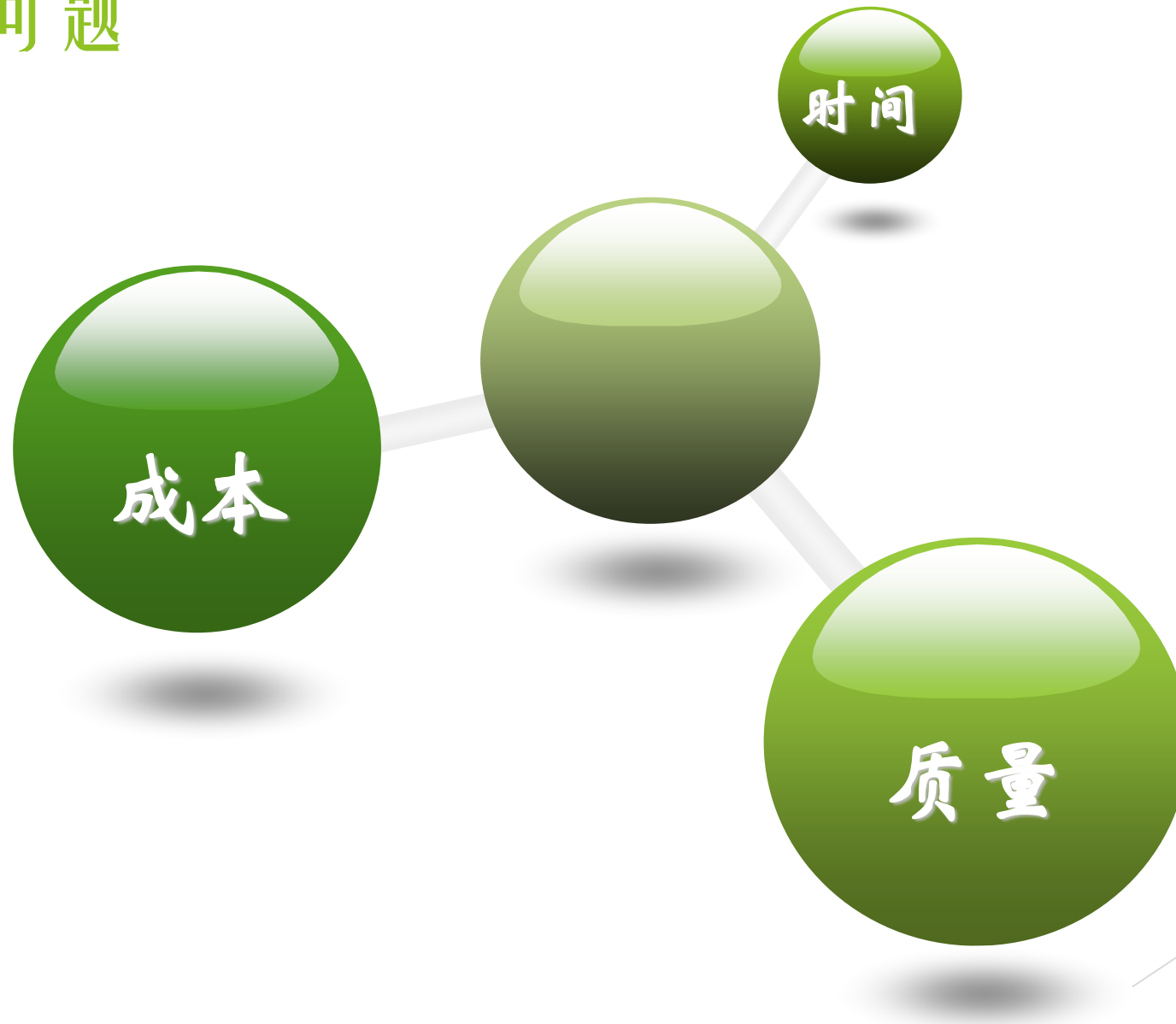


从软件缺陷的定义看测试

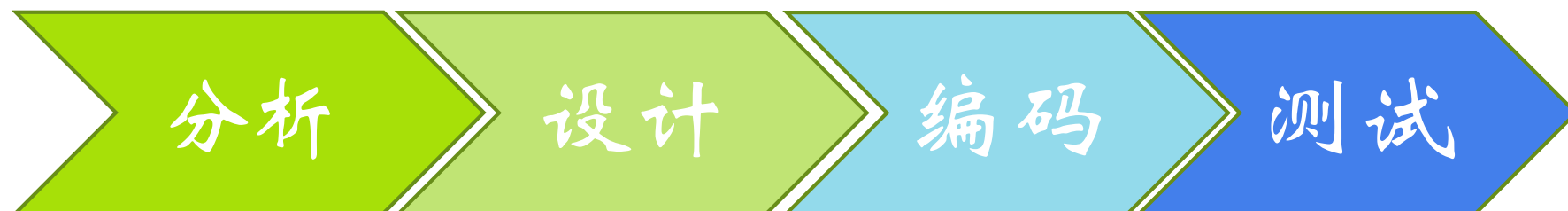
- ▶ 细化了测试的整体思路
 - ▶ 描述缺陷的典型表现
 - ▶ 体现如何比较系统预期与实际输出
- ▶ 问题：测试就是读读需求规格说明书，穷尽检查需求中每一项内容吗？
- ▶ 对应等级：定义阶段



现实问题



时间压力

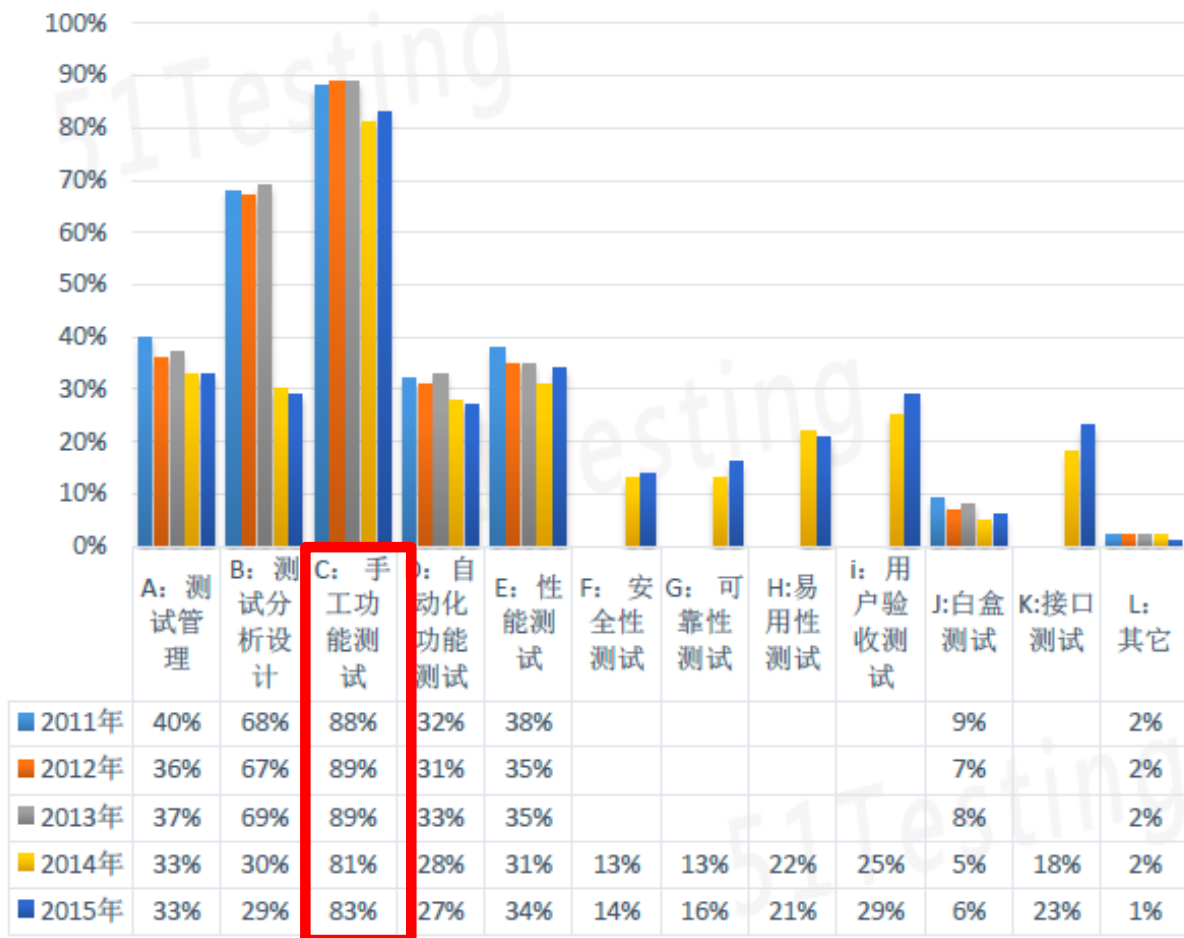


计划的进度



实际的进度

成本控制

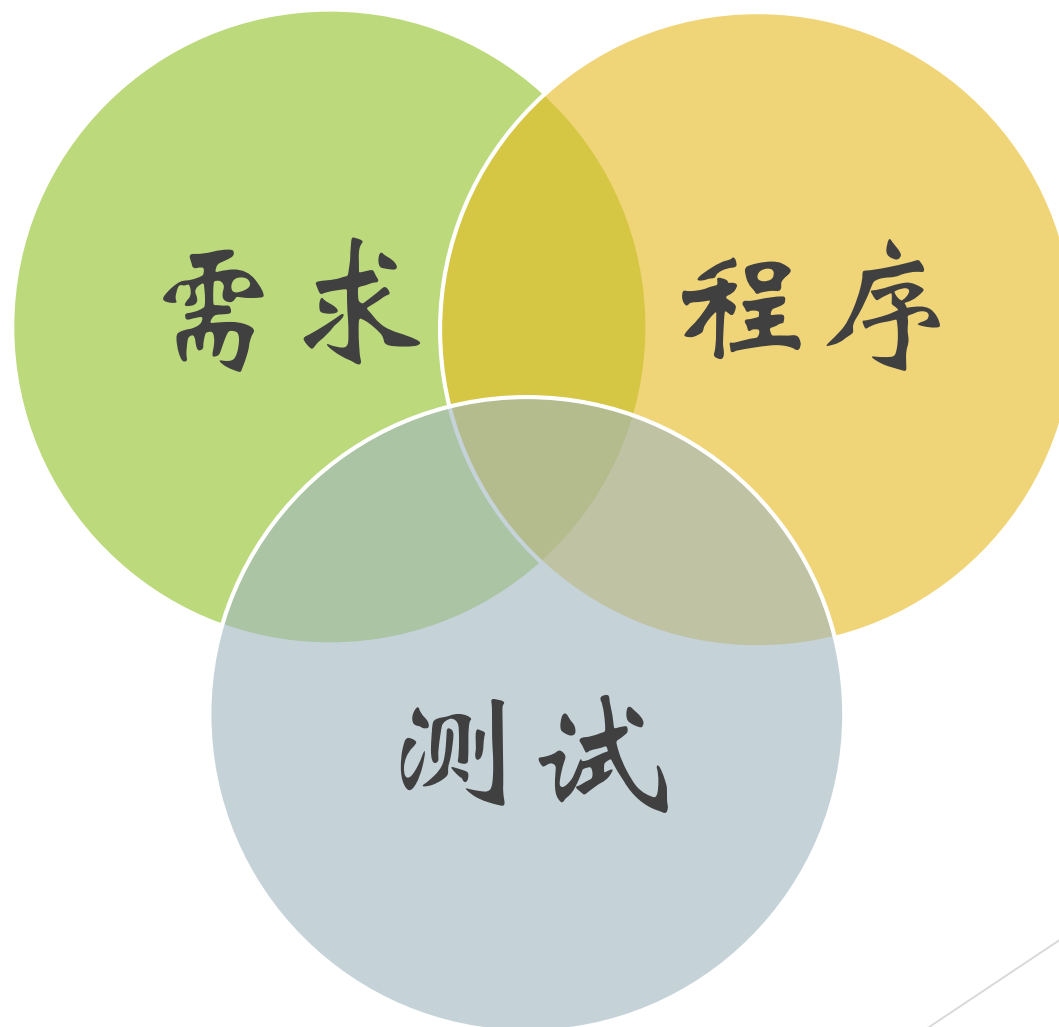


统计规则：基于 51Testing 2015 年第九届软件测试现状调查数据统计分析

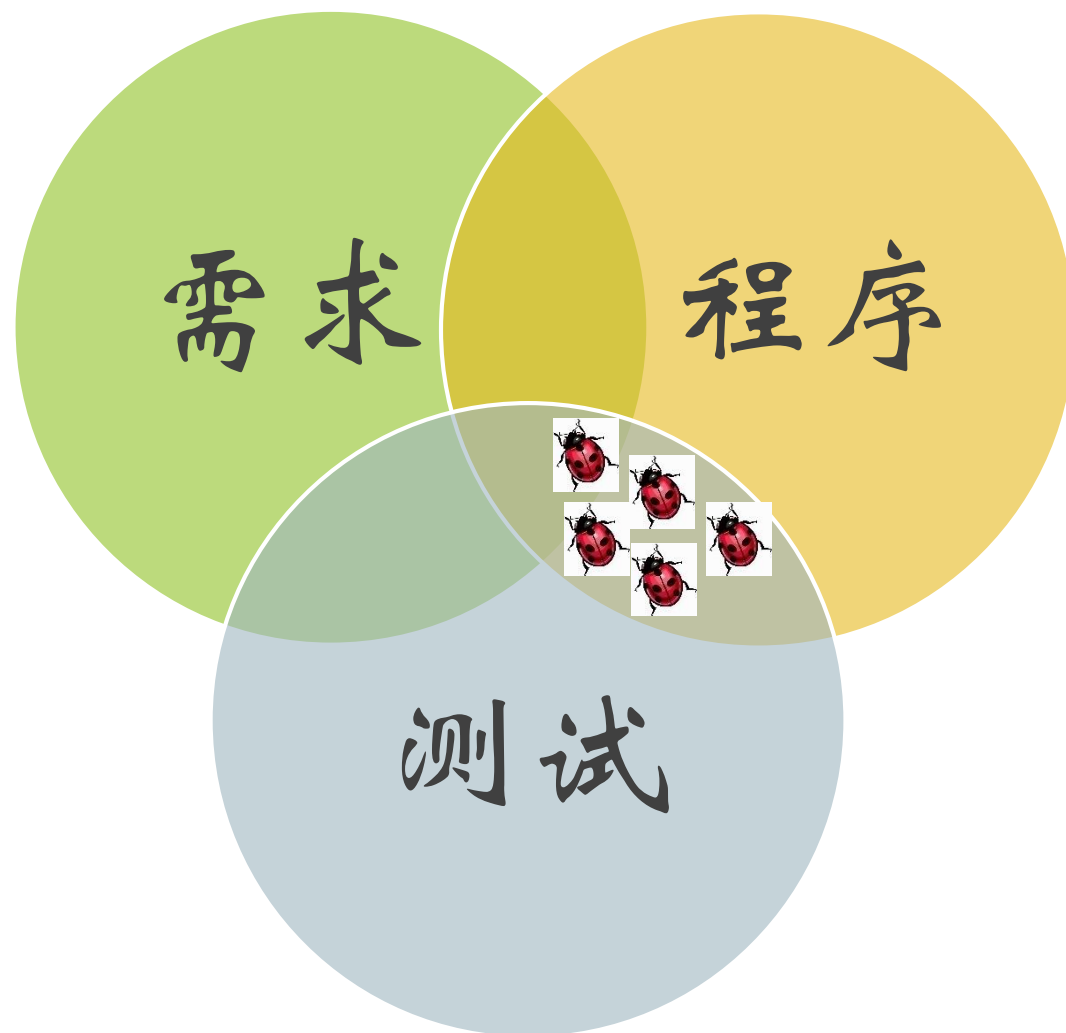
数据来源：51Testing (www.51testing.com)

质量要求

- ▶ 如何用测试验证产品质量？
- ▶ 根据需求设计测试，然后用测试去执行程序

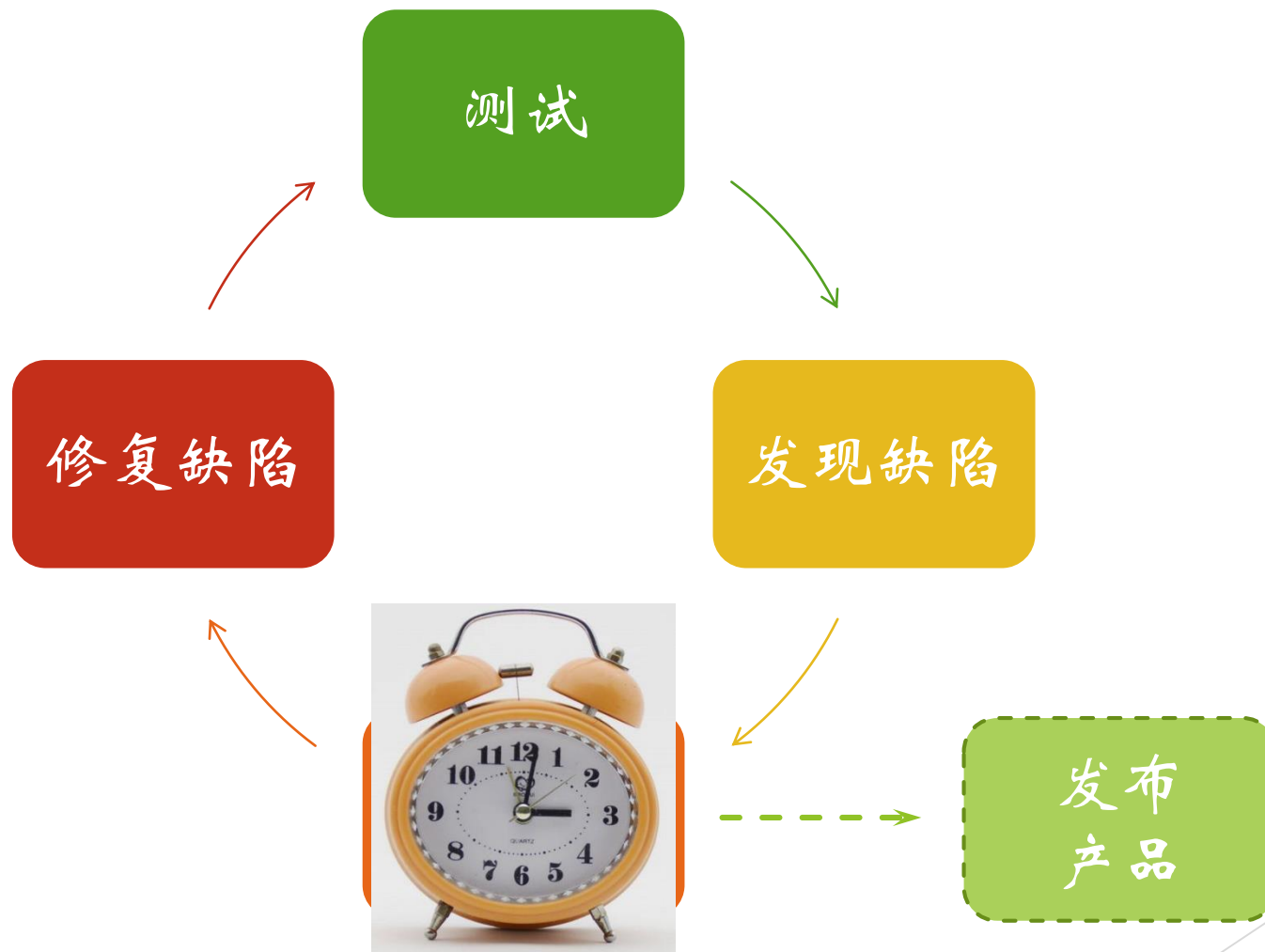


质量要求

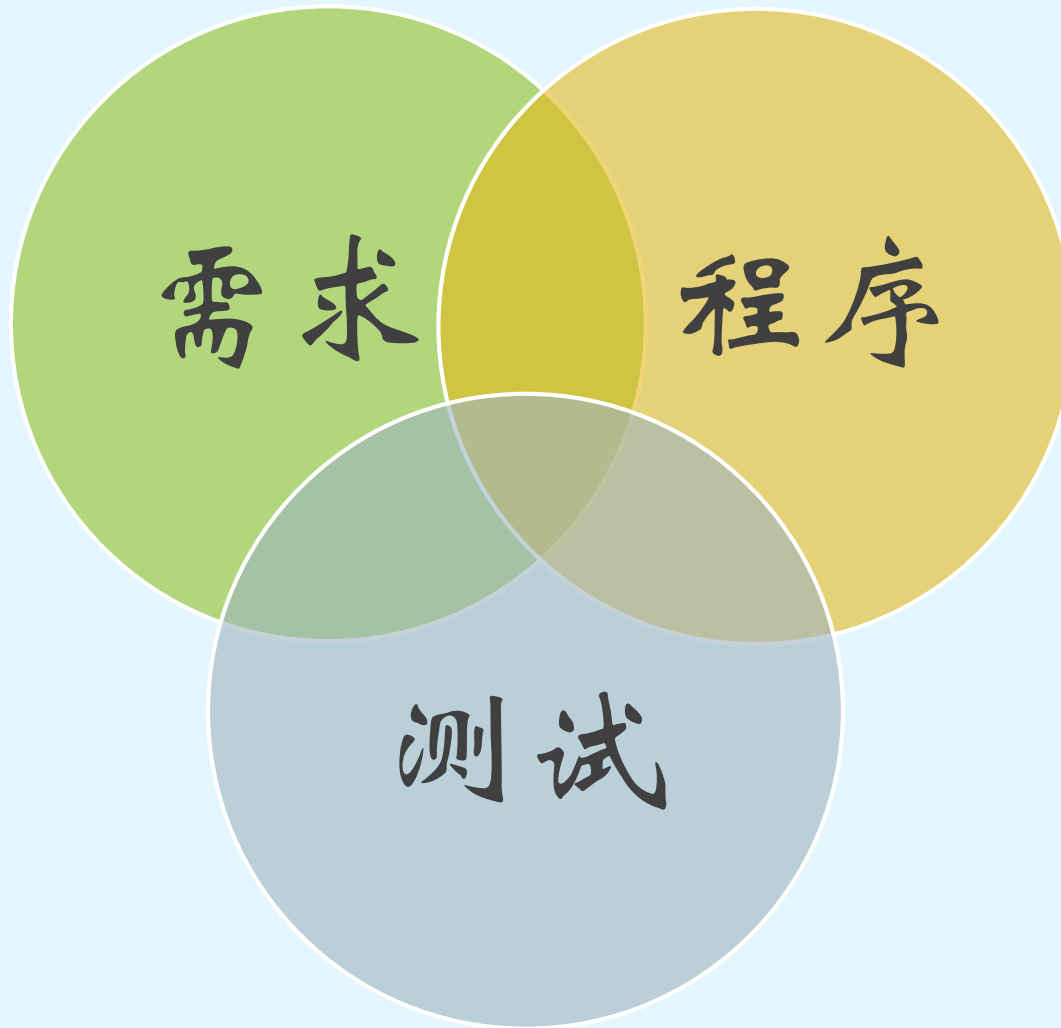


Glendord J Myers

质量要求：测试的怪圈



用户需求



测试用例 (Test Case)

- ▶ 基于风险最低、效率最高、分而治之的测试设计原则
- ▶ 测试用例就是
 - ▶ 能代表需求的小的测试单元
 - ▶ 描述用户预期输出
 - ▶ 反映系统实际执行结果



测试用例的组成



- ▶ 输入：测试数据和操作步骤
- ▶ 输出：系统预期执行结果
- ▶ 测试环境：是系统环境设置，即进行软件测试所必需的工作平台和前提条件

测试用例设计要关注的问题



- ▶ 哪里是系统输入输出中最可能潜伏缺陷的地方？
- ▶ 以怎样的流程操控系统更有利于发现缺陷？
- ▶ 如何保证测试用例对系统的全覆盖？

测试用例的基本属性

- ▶ **典型性**：能揭示最有可能存在缺陷的地方，能代表和覆盖合理与不合理、合法或不合法的情况。
- ▶ **可测试性**：一个测试用例的预期输出必须是可以检验的，可以根据相关开发文档得到明确的、可判定的结论。
- ▶ **可重现性**：对于相同的测试用例，系统的预期执行结果应该完全相同，否则，如果系统预期输出存在不确定性，一旦实际运行该测试用例，也无法进行校验。
- ▶ **独立性**：测试用例应尽量独立。

测试用例的设计

- ▶ 输入数据
 - ▶ 正常数据
 - ▶ 错误数据
 - ▶ 满足数据类型，不在有效取值范围内
 - ▶ 不完全满足数据类型
 - ▶ 输入条件缺失
 - ▶ 边界数据
- ▶ 操作步骤



时间、成本、质量

