

Web 系统测试

3.2敏捷测试象限

目录

- 敏捷测试方法
- 什么是敏捷测试象限
- 为什么有敏捷测试象限

- 测试驱动开发
- 验收测试驱动开发
- 行为驱动开发

➤概念：使用自动化测试用例来指导和验证开发代码

➤包含的活动：

- 新增一个测试，用于描述在代码内的某一小片段所期望的功能
- 此时因为相应的业务代码还没有编写，测试无法通过
- 编写业务代码并允许测试直到测试通过
- 测试通过后，如果代码又有变更，再次允许测试以确保变更后的代码仍然可以通过测试

➤编写测试用例

- 操作过程尽量模拟正常使用的过程。
- 全面的测试用例应该尽量做到分支覆盖，核心代码尽量做到路径覆盖。
- 测试数据尽量包括：真实数据、边界数据。
- 测试语句和测试数据应该尽量简单，容易理解。
- 为了避免对其他代码过多的依赖，可以实现简单的桩函数或桩类（Mock Object）。
- 如果内部状态非常复杂或者应该判断流程而不是状态，可以通过记录日志字符串的方式进行验证

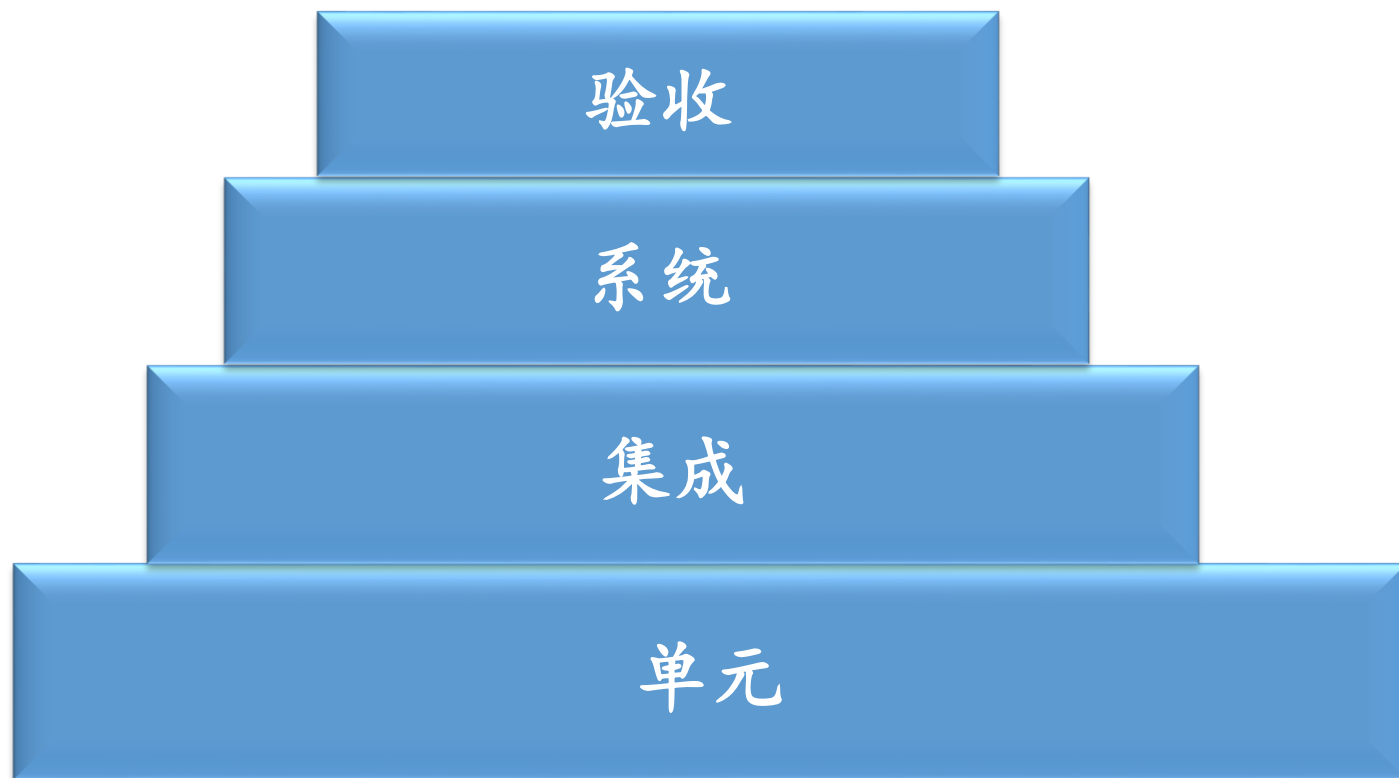
➤概念：一种在用户故事创建阶段就设定验收准则和测试的技术

➤概念：使开发人员聚集与测试软件行为是否符合预期

➤意义

- 帮助开发人员生成测试代码
- 针对业务需求定义精准的单元测试

测试金字塔



➤ 单元测试

- 尽可能达到100%判定覆盖率，仔细评审不可达路径
- 对所有代码执行静态分析
- 没有未解决的严重缺陷
- 所有单元测试都是自动化的
- 重要特性满足约定限制

➤集成测试

- 所有功能需求都经过测试，包括正向和逆向测试，测试的数量基于规模、复杂度和风险
- 所有单元之间的接口都经过测试
- 根据约定的测试程度，所有质量风险都已覆盖
- 所有发现的缺陷都已报告
- 在可能的情况下，所有回归测试都是自动化的，所有自动化测试都存放在一个公共库中

➤ 系统测试

- 用户故事、特性和功能的端到端测试
- 覆盖了所有用户角色
- 覆盖了系统最重要的质量特性（如：性能、健壮性、可靠性）
- 在类似生产环境中完成测试，在一定程度上包括了所有软硬件和所支持的配置
- 根据测试约定的测试程度覆盖了所有质量风险
- 在可能的情况下，所有回归测试都是自动化的
- 所有发现的缺陷都已报告并被修复
- 没有未解决的严重缺陷

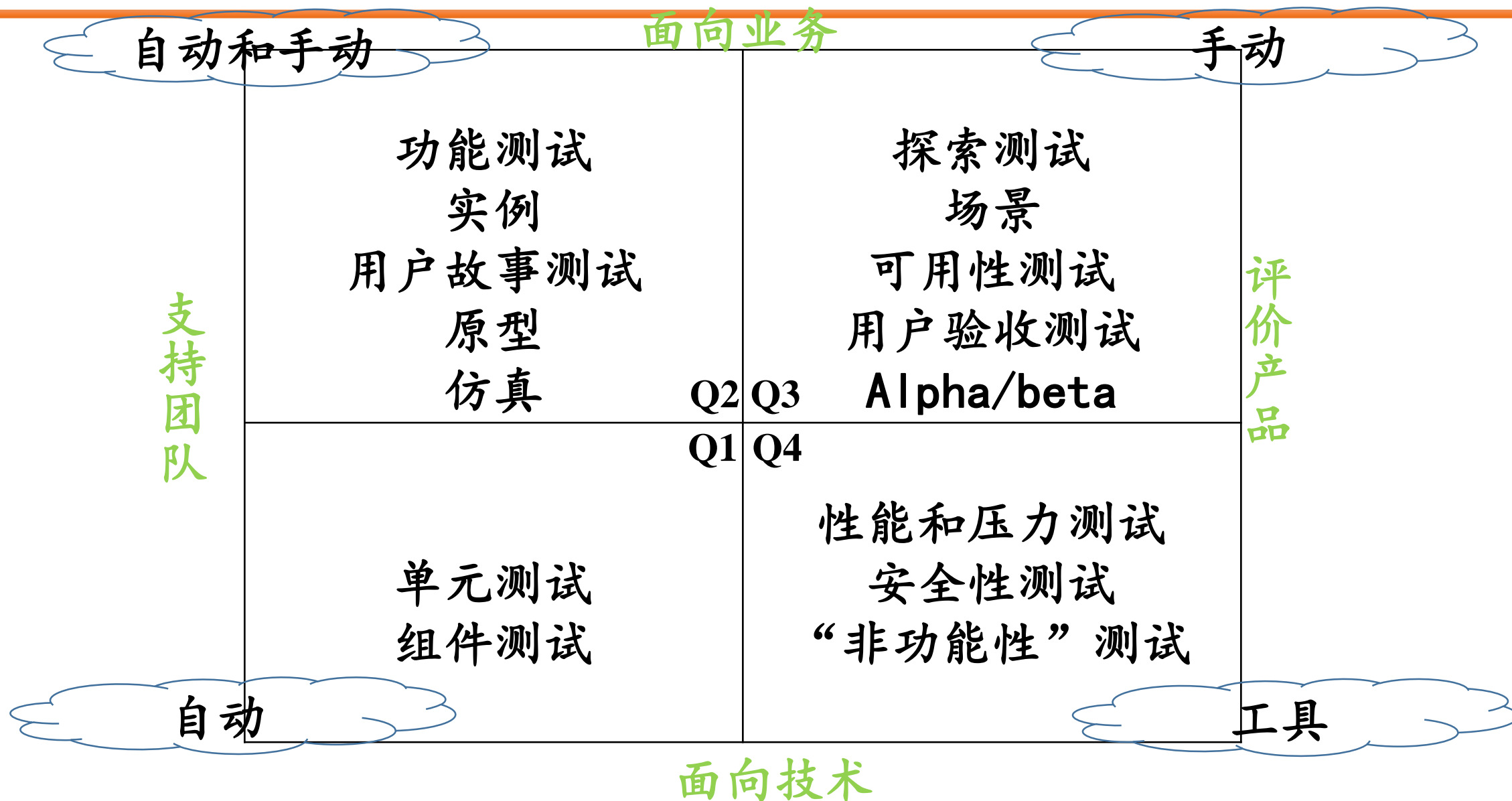
➤ 验收测试

- 迭代所选择的用户故事全部完成，能被测试团队理解，并且有详细的、可测试的验收准则

什么是敏捷测试象限

- 通过不同类型的测试达到不同目的
- 根据测试类型的不同将其分到不同的象限中

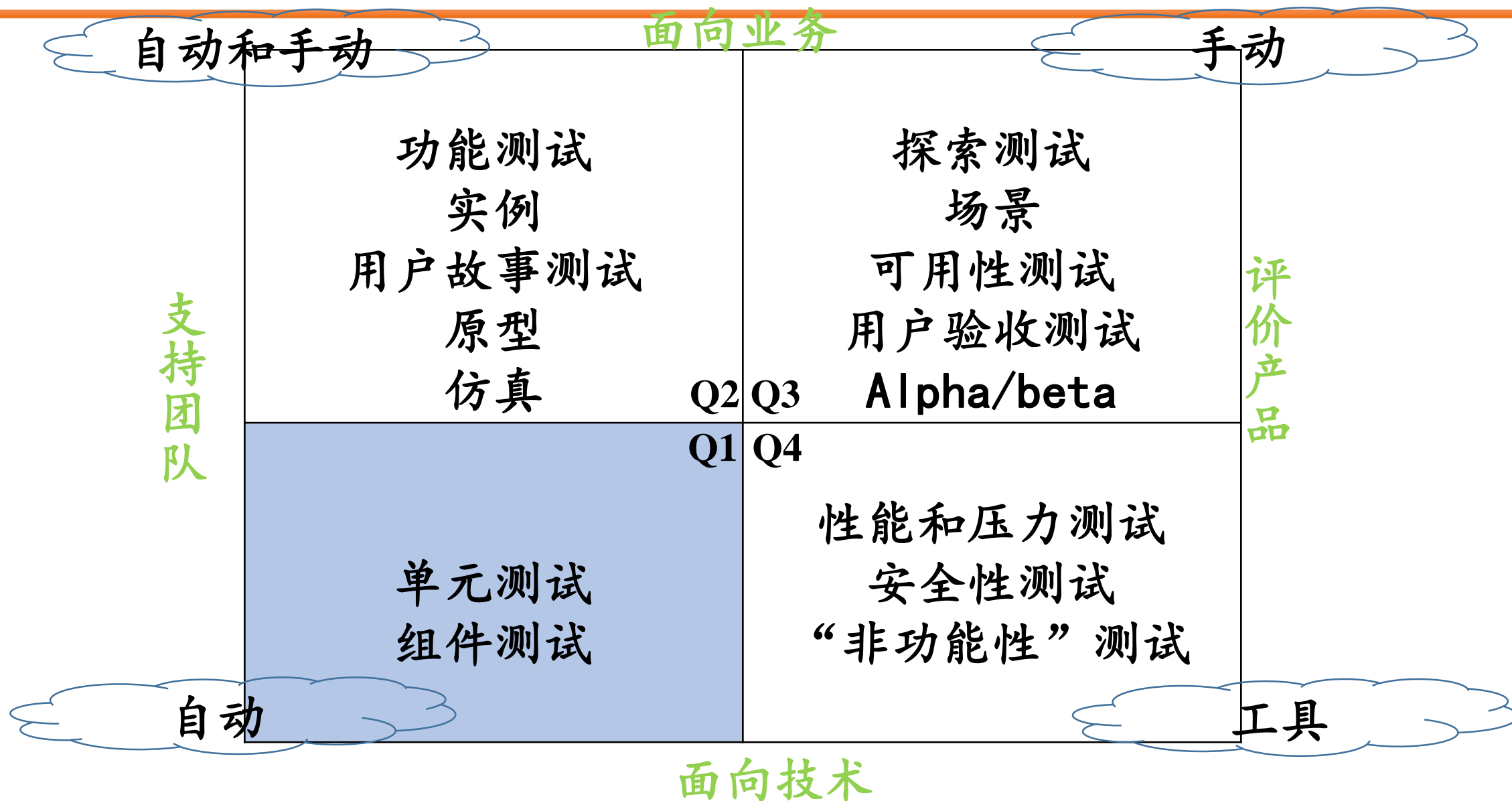
什么是敏捷测试象限



为什么划分测试象限

- 帮忙测试人员，确保他们已经考虑了为了创造价值而需要的所有不同类型的测试

第一象限介绍



➤ 第一象限

● 单元测试

- 验证代码段的行为，小至单独对象或方法

● 组件测试

- 通过测试类和方法间的交互帮助巩固系统的一个可部署部分的整体设计

● TDD(Test-Driven Development)

为什么编写这些测试

- 提高效率
- 让测试人员的工作更容易
- 设计时谨记测试
- 即时反馈

第一象限中可以借助的工具

- 源代码控制
- 优秀集成开发环境 (IDE)
- 构建工具
- 单元测试工具

- 什么是敏捷测试象限
- 为什么有敏捷测试象限
- 第一象限相关知识
 - 什么是单元测试和集成测试
 - 为什么进行单元测试和集成测试

Question