中山大學
SUN YAT-SEN UNIVERSITY

# Chapter 4
# Software Dynamic Testing

Software Testing: Approaches & Technologies
School of Data & Computer Science, Sun Yat-sen University

# Outline

- 4.1 白盒测试
- 4.2 黑盒测试
- 4.3 灰盒测试
- 4.4 测试用例设计
- 4.5 单元测试
- 4.6 集成测试
- 4.7 确认测试
- 4.8 系统测试
- 4.9 动态测试工具

# Outline

- **4.4 测试用例设计**
  - 测试用例概述
  - 测试用例编写要素
  - 测试用例编写模版
  - 测试用例分级
  - 测试用例设计的误区

# 4.4 测试用例设计

- **测试用例概述**
  - 测试用例的概念
    - ✧ 测试用例是为特定的目的而设计的一组测试输入、执行条件和预期的结果，是测试方案、方法、技术和策略的体现。
      - ◌ 测试用例是执行动态测试的最小实体，其内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并形成文档。
      - ◌ 测试用例就是设计一个场景，检验软件程序在这种场景下是否能够正常运行并且达到程序所设计的执行结果。
    - ✧ 测试用例的设计和编制是软件测试活动中最重要的活动，高质量的测试用例是软件测试质量的根本保障。

# 4.4 测试用例设计

- **测试用例概述**
  - 高质量测试用例
    - ✧ 高质量测试用例所应具备的特点
      - ◌ 正确性；完整性；描述准确性、清晰性和简洁性；可维护性；适应性；可再现性等等。
    - ✧ 正确性
      - ◌ 测试用例尽量输入用户应用的实际数据以验证系统是否满足需求规格说明书的要求；测试用例中的测试点应保证至少覆盖需求规格说明书中的各项功能。
    - ✧ 完整性
      - ◌ 测试用例不可遗漏任何基本功能项。
      - ◌ 测试用例还应该体现临界测试、压力测试、性能测试等方面的测试。

# 4.4 测试用例设计

- **测试用例概述**
  - 高质量测试用例 (续)
    - ✧ 描述能力
      - ◌ 按测试用例的输入实施测试后，根据测试用例描述的输出得出正确/准确的结论。
        - 不能出现模糊不清的语言。
      - ◌ 好的测试用例描述清晰，每一步骤都应有相应的作用，有很强的针对性，不应出现一些冗繁无用的操作步骤。
    - ✧ 可维护性
      - ◌ 测试用例的维护：软件开发过程中发生需求变更时，需要对测试用例进行修改、增加、删除等，以符合相应改变的测试要求。
      - ◌ 测试用例应具备可维护的功能。

# 4.4 测试用例设计

- **测试用例概述**
  - 高质量测试用例 (续)
    - 适应性
      - 测试用例应该适应特定的测试环境，以及符合整个团队的测试技术水平。
    - 可再现性
      - 不同测试者在同样测试环境下使用同样的测试用例能得出相同的结论。
    - 其它
      - 可追朔性、可移植性等也是对编写测试用例的要求。

# 4.4 测试用例设计

- **测试用例设计的基本原则**
  - 基于测试需求的原则
    - ✧ 按照测试类别的不同要求设计测试用例。如：
      - ◌ 单元测试依据详细设计说明；
      - ◌ 集成测试依据概要设计说明；
      - ◌ 配置项测试依据软件需求规格说明；
      - ◌ 系统测试依据用户需求说明 (系统/子系统设计说明、软件开发计划等)。
    - ✧ 测试内容具备完整性，具有可操作性。
  - 基于测试方法的原则
    - ✧ 对于不同的测试要求，选择采用相应的测试方法，如等价类划分、边界值分析、猜错法、因果图等方法。

# 4.4 测试用例设计

- **测试用例设计的基本原则**
  - 测试用例的代表性
    - ✧ 测试用例能够代表并覆盖各种合理和不合理的、合法和非法的、边界内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
    - ✧ 测试执行结果的正确性是可判定的，每一个测试用例都具有相应的期望结果。
  - 测试执行的可再现性
    - ✧ 对同样的测试用例，系统的执行结果是相同的，即可再现的。
  - 测试充分性和效率兼顾的原则

# 4.4 测试用例设计

- **测试用例设计的具体原则**
  - 测试用例和测试目标的功能点一一对应
    - ✧ 测试用例对应的功能点称为该用例的测试点。
  - 测试用例具有易读性
    - ✧ 测试用例应该从执行者的角度去设计，指明测试的具体位置。
  - 测试用例具有尽量小的执行粒度
    - ✧ 粒度小的测试用例所覆盖的边界定义将更加清晰，测试结果对产品问题的指向性更强。
    - ✧ 测试用例间的耦合度越低，彼此之间的干扰也就越低。
    - ✧ 符合上述要求的测试用例比较容易调试和分析，可以降低测试用例的维护成本。
  - 测试用例具有明确的期望结果
    - ✧ 测试用例明确指明一个测试操作之后应该产生的结果。

# 4.4 测试用例设计

- **测试用例设计的具体原则**
  - 测试步骤清晰可读
    - ✧ 测试用例的使用有清晰明确的指导步骤。
  - 将具有相类似功能的测试用例抽象并归类
    - ✧ 这是由于软件测试过程无法进行穷举测试。
  - 避免冗长和复杂的测试用例。
    - ✧ 总体设计思路是先易后难，循序渐进，确保正常情况下基本功能能够走通。如：
      - ◌ 先进行基本功能测试，再进行复杂功能测试；
      - ◌ 先进行一般用户使用测试，再进行特殊用户使用测试；
      - ◌ 先进行正常情况的测试，再进行异常情况 (内存和硬件的冲突、内存泄漏、破坏性测试等) 的测试。
  - 尝试用测试用例替代产品文档。
  - 测试用例设计要考虑单次投入成本和多次使用成本。

# 4.4 测试用例设计

- **测试用例覆盖内容**

1. 正确性测试
2. 容错性 (健壮性) 测试
3. 完整 (安全) 性测试
4. 接口测试
5. 数据库测试
6. 边界值测试
7. 压力测试
8. 等价划分测试
9. 错误推测
10. 效率
11. 可理解 (操作) 性测试
12. 可移植性测试
13. 回归测试
14. 比较测试

# 4.4 测试用例设计

- **测试用例覆盖内容**
  - 针对不同的测试类型和测试阶段，测试用例编写的侧重点有所不同。
    - ✧ 模块 (组件、控件) 测试、组合 (集成) 测试和系统测试要重点测试 内容 1、2、6、8、9、13；
    - ✧ 单元 (模块) 测试和组件、控件测试要重点测试内容 5；
    - ✧ 集成测试重点进行接口数据输入及逻辑测试，即测试内容 4；
    - ✧ 系统测试重点测试内容 3、7、10、11、12、14；
    - ✧ 完成基础功能测试用例设计后，其相关测试内容不需要在其他测试项目重复测试；
    - ✧ 随着测试经验的积累和测试过程的进展，可以不断补充完善测试项目用例的测试内容。

# 4.4 测试用例设计

- **测试用例编写要素**
  - 名称和标识
  - 来源追踪
    - ✧ 说明测试所依据的内容来源。例如系统测试依据的用户需求，配置项测试依据的软件需求，集成测试和单元测试依据的软件设计。
  - 用例说明
    - ✧ 简要描述测试的对象、目的和所采用的测试方法。
  - 初始化要求
    - ✧ 硬件配置、软件配置、测试配置、参数配置及其他对于测试用例的特殊说明。
  - 输入内容
    - ✧ 在测试用例执行中发送给被测对象的所有测试命令、数据和信号等。

# 4.4 测试用例设计

- **测试用例编写要素**
  - 期望的测试结果
    - ✧ 说明测试用例执行中由被测软件所产生的期望测试结果。
  - 评价测试结果的准则
    - ✧ 判断测试用例执行中产生的中间结果和最后结果是否符合期望结果。
  - 操作过程
    - ✧ 实施测试用例的执行步骤。
  - 前提和约束
    - ✧ 在测试用例说明中施加的所有前提条件和约束条件。标识特别限制、参数偏差或异常处理的情况，说明对测试用例的影响。
  - 测试终止条件
    - ✧ 说明测试正常终止和异常终止的条件。

# 4.4 测试用例设计

- **测试用例编写模版** (ANSI/IEEE 829-1983标准)
  - 标识符
    - ✧ 每个测试用例有一个唯一的标识符，它将成为所有和测试用例相关的文档/表格引用和参考的基本元素，这些文档/表格包括设计规格说明书、测试日志表、测试报告等。
  - 测试项
    - ✧ 测试用例应该准确地描述所需要测试的项及其特征，测试项应该比测试设计说明书中所列出的特性描述更加具体。
  - 测试环境要求
    - ✧ 用来表征执行该测试用例所需要的测试环境。

# 4.4 测试用例设计

- **测试用例编写模版** (ANSI/IEEE829-1983标准)
  - 输入标准
    - ✧ 用来执行测试用例的输入需求。这些输入可能包括数据、文件或操作，必要时相关的数据库、文件也必须被罗列。
  - 输出标准
    - ✧ 标识按照指定的环境和输入标准得到的期望输出结果。如果可能的话，尽量提供适当的系统规格说明书来证明期望的结果。
  - 测试用例之间的关联
    - ✧ 用来标识该测试用例与其它的测试 (或其它测试用例) 之间的依赖关系。

# 4.4 测试用例设计

- **测试用例的设计步骤**
  - 测试需求分析
    - ✧ 从软件需求文档中，提取出被测目标软件/模块的需求。
    - ✧ 测试需求在软件需求基础上进行归纳、分类或细分得到，方便测试用例设计。
      - ◌ 测试需求包含了软件需求，具有可测试性。
    - ✧ 测试用例集与测试需求的关系是多对一的，即一个或多个测试用例集对应一个测试需求。
  - 业务流程分析
    - ✧ 软件测试不单纯是功能测试，还需要对软件的内部处理逻辑进行测试。
    - ✧ 为了不遗漏测试点，需要清楚了解软件产品的业务流程。

# 4.4 测试用例设计

- **测试用例的设计步骤**
  - 测试用例设计
    - ✧ 测试用例设计的类型包括功能测试、边界测试、异常测试、性能测试、压力测试等。
    - ✧ 在测试用例设计中，除了功能测试用例外，应尽量考虑边界、异常、性能的情况，以便发现更多的隐藏问题。
  - 测试用例评审
    - ✧ 测试用例设计完成后，为了确认测试过程和方法是否正确、是否有遗漏的测试点，需要进行测试用例的评审。
    - ✧ 测试用例评审一般由测试主管安排，参加的人员包括：
      - ◌ 测试用例设计者、测试主管、项目经理、开发工程师、其它相关开发测试工程师。
    - ✧ 测试用例评审完毕后，测试工程师根据评审结果，对测试用例进行修改，并记录修改日志。

# 4.4 测试用例设计

- **测试用例的设计步骤**
  - 测试用例更新完善
    - ✧ 测试用例在软件的生命周期中不断更新与完善。测试用例更新完善的条件包括：
      - ◌ 产品新增功能或更新需求；
      - ◌ 在测试过程中发现设计测试用例时考虑不周；
      - ◌ 软件交付使用后客户反馈软件缺陷，而缺陷是由于测试用例存在漏洞造成的；
      - ◌ 软件版本升级更新。
    - ✧ 测试用例文档需要保留用例的更改记录。

# 4.4 测试用例设计

- **测试用例的分级**
  - 测试用例的级别表明该用例的重要程度。
    - ✧ 定义测试用例的级别
      - ◌ 测试用例的级别描述该测试用例在测试执行过程中的重要程度，包括优先执行规定。
    - ✧ 测试用例的级别并不对应测试用例可能造成的后果的严重性
      - ◌ 测试用例的重要性对应于测试用例的基本程度。
        - 例如一个可能导致系统死机的测试用例未必是高级别的，因为其触发条件可能相当特别。
    - ✧ 测试用例执行的优先程度首先取决于该用例所测试的用户需求点的紧急程度、使用频率以及重要程度。
      - ◌ 测试用例的优先级首先要继承测试需求点的优先级别。
        - 先对测试需求进行优先级定义，再对需求点对应的系列测试用例的优先级别进行定义。

# 4.4 测试用例设计

- **测试用例的分级**
  - 基本级 (第1级)
    - ✧ 该级别的测试用例涉及被测系统的基本功能。
    - ✧ 该级别的测试用例的数量应受到控制。
    - ✧ 该级别的划分依据是该用例执行的失败会导致多项重要功能无法运行，这些功能是发生概率较高且经常用到的一些功能。
    - ✧ 该级别的测试用例在每一轮版本测试中都必须执行。

# 4.4 测试用例设计

- **测试用例的分级**
  - 重要级 (第2级)
    - ✧ 该级别的测试用例用于测试被测系统的重要功能。
    - ✧ 该级别的测试用例数量较多。
    - ✧ 该级别的划分针对一些交互相关、应用场景广泛、使用频率较高的功能的正常实现测试用例。
    - ✧ 该级别的测试用例在非回归的系统测试版本中全部进行验证，以保证系统所有的重要功能都是正常实现的。
    - ✧ 在测试过程中可以根据当前版本的具体情况决定是否在回归测试中全部执行或者部分执行这些测试用例。

# 4.4 测试用例设计

- **测试用例的分级**
  - 一般级 (第3级)
    - ✧ 该级别的测试用例涉及系统的一般功能。
    - ✧ 该级别的测试用例数量较多。
    - ✧ 该级别的划分针对使用频率较低于第2级的测试用例。
    - ✧ 该级别的测试用例在非回归的系统测试版本中不一定都要进行验证，而且在系统测试的中后期并不一定需要对每个版本都进行该级别的测试。
  - 特殊级 (第4级)
    - ✧ 该级别的测试用例数量一般很少。
    - ✧ 该级别的划分针对特殊的预置条件和数据设置的测试用例。
    - ✧ 虽然某些测试用例发现过较严重的错误，但那些用例的触发条件非常特殊，仍然应该被置于第4级测试用例中。
    - ✧ 该级别经过测试经理同意可以不进行测试。

# 4.4 测试用例设计

- **测试用例的优先级设计**
  - 测试优先级最高的需求点
    - ✧ 在根据用户需求和需求分析文档提取测试需求时，必须了解所有需求中，哪些是用户急需使用的部分，哪些是用户使用频繁的部分，以及哪些是系统最不能出现错误的部分等等。
    - ✧ 这些部分就是测试优先级最高的需求点。
  - 影响测试用例优先级的因素
    - ✧ 继承测试需求点的优先级别；
    - ✧ 测试用例在用户实际使用中的使用概率及频率；
    - ✧ 测试用例导致被测软件出错的概率；
    - ✧ 测试用例导致系统发生错误后对系统的危害性。

# Thank you!