

AIoT人工智能项目实战

- 人脸识别测试设计

辛慧

15309220868

目录

- 测试基本概念
- 软件测试模型
- 软件测试方法
- 测试工具简介

测试基本概念

- 软件测试定义
- 软件测试目的
- 软件测试原则
- 软件测试分类
- 测试用例于执行

软件测试定义

- 软件测试就是在软件投入运行前，对软件需求分析、设计规格说明和编码的最终复审，是软件质量保证的关键步骤。
- 早期定义：软件测试是为了发现软件中错误而运行软件的过程。（1979年，G.J.Myers）
- 后期定义：使用人工或者自动的手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或者是弄清预期结果与实际运行结果之前的差别。（1983年，IEEE）
- 定义：软件测试是在一定的软件、硬件、网络环境下，遵循相对规范的软件测试流程，使用合理的测试方法以及适当的测试工具运行或测试某个系统的过程。

软件测试目的

- 软件测试的目的（业界有两种权威定义）
 - 第一、评价一个程序和系统的特性或能力，并确定它是否达到预期的结果。
 - 第二、是在软件分发到最终用户手中之前，以最少的时间和人力找出软件中潜在的各种错误和缺陷。
 - 软件测试目的：检验软件是否满足规定的需求或者是弄清预期结果与实际运行结果之间的差别。

软件测试原则

- 所有测试的标准都是建立在用户需求之上
- 尽早定义好产品的质量标准的
- 测试应尽早介入
- 穷尽测试是不可能的
- 测试应贯穿于软件整个生命周期
- 应有第三方或独立的测试团队

软件测试分类

- 按照开发阶段划分

- 1) 单元测试

单元测试（Unit Testing, UT）又称模块测试，是对软件基本单元（软件设计最小单位）进行正确性进行检验，检查代码语法、格式和逻辑上的错误，验证程序是否符合规范，尽早发现单元内部可能存在的各种缺陷。

- 2) 集成测试

集成测试（Integration Testing, IT）又称组装测试或联合测试，是单元测试在集成时是否有缺陷。是单元测试的逻辑扩展，通过测试识别组合单元时出现的问题。

集成测试的目标就是检测系统是否达到需求；对业务流程及数据流的处理是否符合标准；检测系统对业务流处理是否存在逻辑不严谨或者错误；检测需求是否存在不合理的标准及要求。

软件测试分类

- 按照开发阶段划分（续）

- 3) 系统测试

系统测试（System Testing, ST）是将已集成好的软件系统，作为整个计算机系统的一个元素，与支持软件、计算机硬件、外设、数据等其他系统元素结合在一起，在模拟实际使用环境下，对计算机系统进行一系列的测试活动。

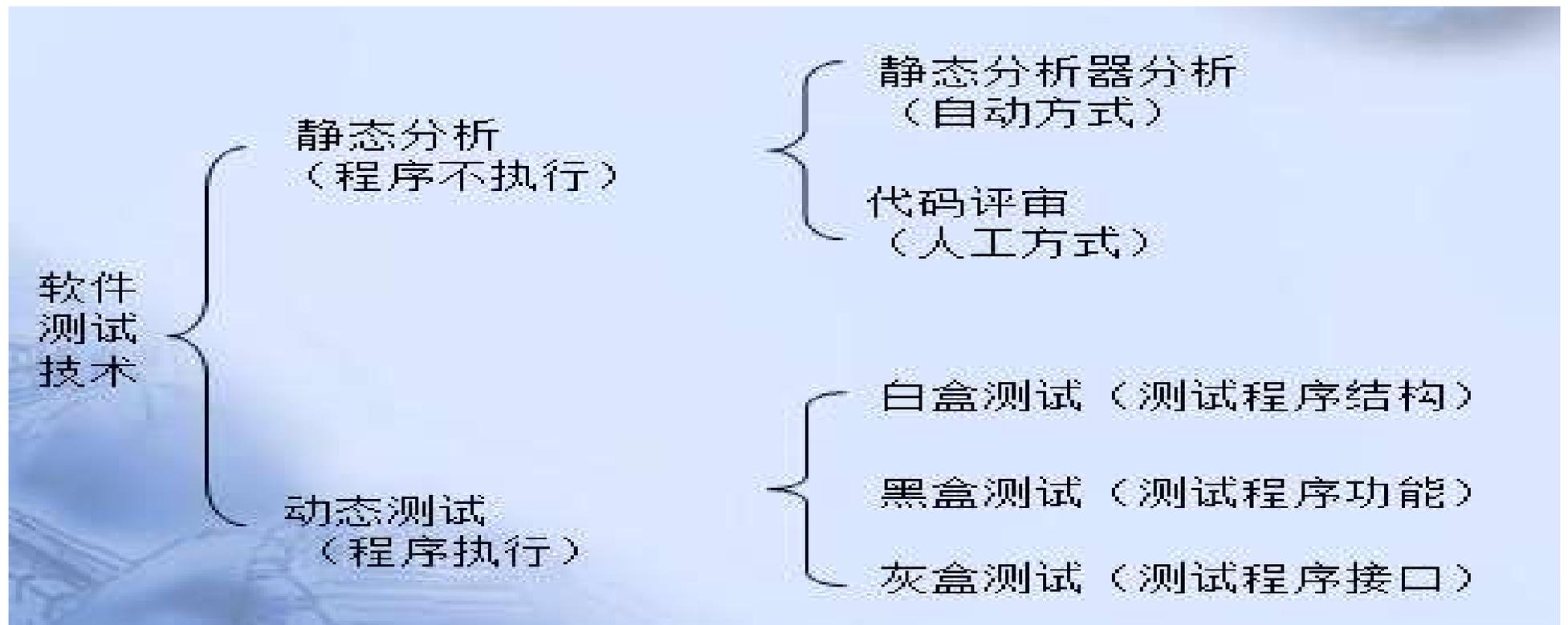
- 4) 验收测试

验收测试（Acceptance Testing, AT）也称交付测试，是在软件产品完成了单元测试、集成测试和系统测试之后，产品发布之前所进行的软件测试活动，是技术测试的最后一个阶段。

验收测试目的是验证系统是否达到了用户规格说明书中的要求，并试图尽可能的发现软件中存在的缺陷，从未为软件进一步改善提供帮助，保证系统或软件产品最终被用户接受。

软件测试分类

- 按照测试方法划分



软件测试分类

- 按照测试方法划分（续）

- 1) 静态测试

静态测试是指不运行程序，通过人工对程序和文档进行分析与检查。静态测试实际上是对软件中的需求说明书、设计说明书、程序代码等进行评审的一个过程。

- 2) 动态测试

动态测试是指通过人工或使用工具运行程序进行检查、分析程序的执行状态和程序的外部表现，一般包括白盒测试、黑盒测试和灰盒测试。

软件测试分类

■ 按照测试方法划分（续）

➤ 1>白盒测试

白盒测试又称结构测试。白盒测试可以把程序看成装在一个透明的盒子里，也就是清楚的了解程序结构和处理过程，检查是否所有的结构及路径都是正确的，检查软件内部动作是否按照设计说明的规定正常运行。

➤ 2>黑盒测试

黑盒测试是把程序看成一个黑盒子，完全不考虑程序内部结构和处理过程。通常在程序界面处进行测试，它只是检查程序或软件是否按照需求规格说明书的规定正常运行。

➤ 3>灰盒测试

灰盒测试时介于白盒测试和黑盒测试之间的测试。灰盒测试主要关注输出对输入的正确性；同时也关注内部表现，但这种关注不像白盒测试那种详细、完整。灰盒测试结合了白盒测试和黑盒测试的要素。

软件测试分类

- 按照测试测试实施组织划分

- 1) 开发方测试

开发方测试是开发方在软件开发环境下，通过检测和提供客观证据，证实软件是否满足规定的需求。

- 2) 用户测试

用户测试是在实际应用环境下，用户通过运行和使用软件找出软件使用过程中发现软件的缺陷和问题，检测与核实软件实现是否符合用户的预期要求，并把信息反馈给开发者。

- 3) 第三方测试

第三方测试又称为“独立测试”，是介于软件开发和用户方之间的测试组织的测试。也就是由在技术、管理和财务上与开发方和用户方相对独立的组织进行的软件测试。一般情况下是在模拟用户真实应用环境下，进行软件确认测试。

测试用例与执行

- 什么是测试用例?

- 测试用例目前没有经典的定义。比较通常的说法是：指对一项特定的软件产品测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等所形成的文档。

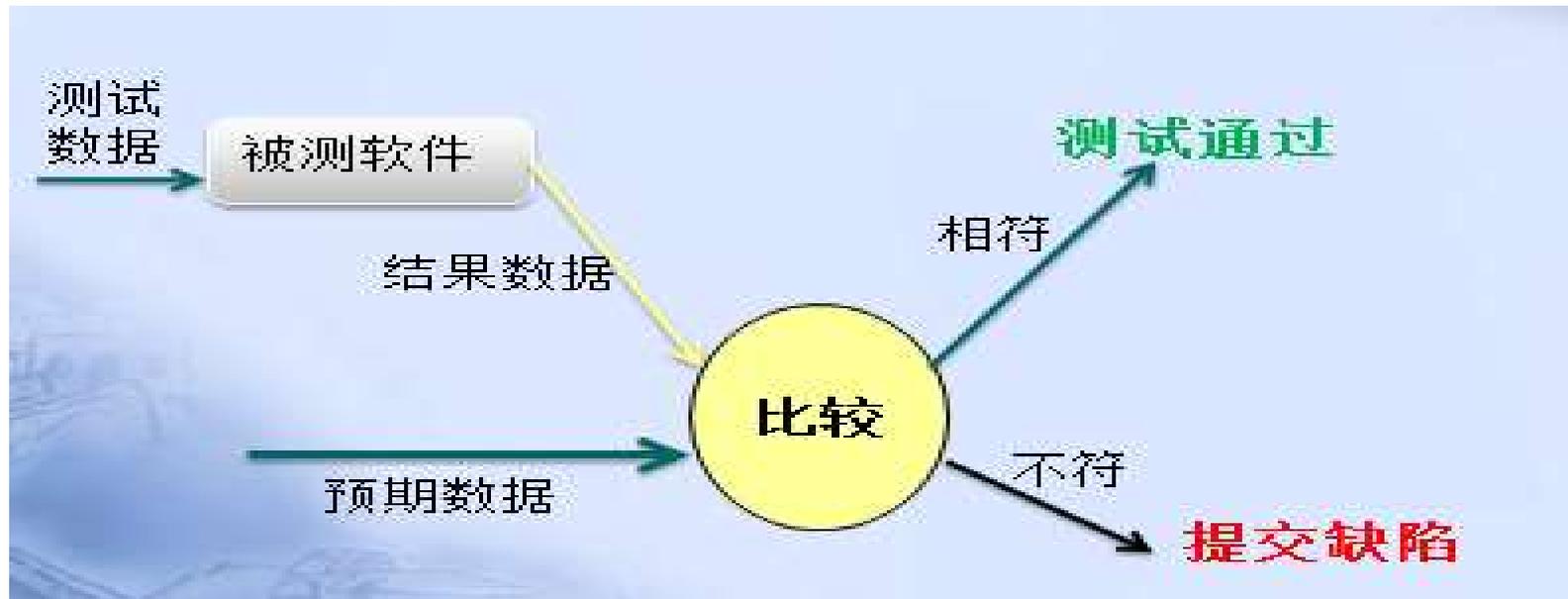
测试用例编号	CALC-ST-SRS001-001
测试项目	测试计算器的加法功能
测试标题	一个数在合法的取值范围，另一个数在最大取值范围外
重要级别	高
预置条件	启动计算器软件
输入	参数1: 1 参数2: + 参数3:65537 参数4: =
执行步骤	用计算机键盘依次输入上述参数
预期输出	参数1: 65538

测试用例与执行

- 什么是测试执行?

- 测试执行就是根据测试用例运行被测软件。

- 一个测试用例或测试用例集的测试执行:



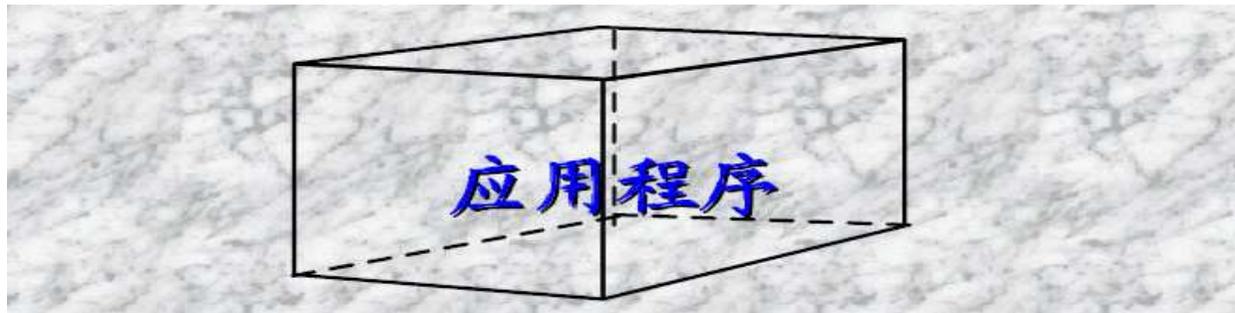
软件测试方法

- 白盒测试
- 黑盒测试
- 白盒测试与黑盒测试比较
- 静态/动态分析

白盒测试

- 什么是白盒测试

白盒测试可以把程序看成装在一个透明的盒子里，也就是清楚的了解程序结构和处理过程，检查是否所有的结构及路径都是正确的，检查软件内部动作是否按照设计说明的规定正常运行。



- 为什么要进行白盒测试

- (1) 逻辑错误和不正确假设与一条程序路径被运行的可能性成反比。
- (2) 我们经常相信某逻辑路径不可能被执行，而事实上它可能在正常的基础上被执行。
- (3) 笔误是随机的。

白盒测试

- 白盒测试主要是想对程序模块进行如下检查：
 - 1、对程序模块的所有独立的执行路径至少测试一遍。
 - 2、对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测一遍。
 - 3、在循环的边界和运行的界限内执行循环体。
 - 4、测试内部数据结构的有效性，等等。

白盒测试

- 白盒测试优点：
 - 1.迫使测试人员去仔细的思考软件的实现；
 - 2.可以检测代码中的每条分支和路径；
 - 3.揭示隐藏在代码中的错误；
 - 4.对代码的测试比较彻底。
- 白盒测试缺点：
 - 1.昂贵（投入成本太高）
 - 2.无法检测代码中遗漏的路径和数据敏感性错误
 - 3.不验证规格的正确性

黑盒测试

- 什么是黑盒测试
- 黑盒测试也叫功能测试，这是因为在黑盒测试中，主要关注被测软件的功能实现，而不是内部逻辑。依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。



黑盒测试

- 为什么要进行黑盒测试

白盒测试在测试的早期采用，黑盒测试在主要用于测试后期。黑盒测试故意不考虑控制结构，而是注意信息域。

黑盒测试主要用于回答以下问题：

- 1、是否有不正确或遗漏的功能？
- 2、在接口上，输入是否能正确的接受？
- 3、能否输出正确的结果？
- 4、是否有数据结构错误或外部信息（例如数据文件）访问错误？
- 5、性能上是否能够满足要求？
- 6、是否有初始化或终止性错误？

黑盒测试

- 黑盒测试常用技术

功能测试的测试数据的选择和测试结果的描述是以软件的功能属性为基础的，即以用户角度去验证软件是否符合功能需求。尽管黑盒测试是围绕着用户需求文档进行的，但是黑盒测试也不一定必须要有用户来参与测试。

- 无用户参与的黑盒测试

常见的无用户参与的黑盒测试有：功能性测试、容量测试、安全性测试、负载测试、恢复性测试、标杆测试、稳定性测试、可靠性测试等。

- 有用户参与的黑盒测试

常见的有用户参与的测试有：实验室测试和外场测试。

黑盒测试

- 黑盒测试优点：

1. 对于更大的代码单元来说（子系统甚至系统级）比白盒测试更高效。
2. 测试人员不需要了解实现的细节，包括特定的编程语言。
3. 测试人员和编码人员时彼此独立的。
4. 从用户的角度进行测试，很容易被大家理解和接受。
5. 有助于暴露任何规格不一致或有歧义的问题。
6. 测试用例可以在规格完成之后马上进行。

黑盒测试

- 黑盒测试缺点：

- 1.只有一小部分可能的输入测试到，要测试每个可能的输入流几乎是不可能的；
- 2.没有清晰简明的规格，测试用例是很难设计的；
- 3.会有很多程序路径没有被测试到；
- 4.不能直接针对特定的程序段，修改问题时，这些程序可能非常复杂。

白盒测试与黑盒测试比较

- 考虑对象不同

白盒测试只考虑测试软件代码，它不保证完整的希求规格是否被满足。

黑盒测试只考虑测试软件的需求规格，它不保证实现的所有部分是否被测试到。

- 技术要求不同

白盒测试要求测试人员要熟悉软件代码以及软件内部的逻辑结构。

黑盒测试要求测试人员熟悉软件需求规格说明书，不要求熟悉代码及软件内部逻辑结构。

- 投入测试成本

白盒测试比黑盒测试成本要高很多。

静态动态分析

- 静态分析(Static analysis)

就是对软件的源代码进行研读，查找错误或收集一些度量数据，不需要对代码进行编译和执行

- 动态分析(Dynamic analysis)

就是通过观察软件运行时的动作，来提供执行跟踪，时间分析，以及测试覆盖度方面的信息。

软件测试工具

请看例程演示

感谢
欢迎提问