

软件测试基础 (PDF测试岗位课程)





课程概览

课程编号 基本描述

课程名称 *缺陷管理培训*

课程级别 *初级/中级*

本课程提供本课程提供软件测试的基础理论介绍，包括测试基本概念、测试方法、测试阶段和活动、测试类型和关键技术、公司所用测试工具。学到测试在提高软件质量上的重要性，是测试入门的基础课程。

课程目标：

- 通过本课程了解测试的基础概念和基本原理、测试阶段和活动
- 了解测试方法和测试过程
- 了解公司所用到的测试工具
- 通过学习认识到测试的重要性和意义

主要学习内容/要点：

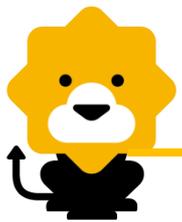
- 测试基本概念和方法
- 测试阶段和活动
- 测试类型和关键技术
- 公司所用测试工具

目标人群和课程时间：

- 信息体系初级专业技术人员，序列含咨询顾问、项目经理、IT工程师、产品经理
- 课程类型：课堂培训
- 时长：2小时

授课要求和课程特色：

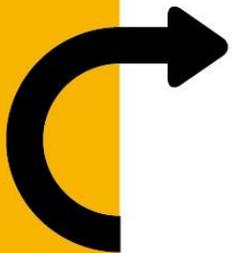
- 人数要求：无
- 分组要求：无
- 案例要求：无
- 提前需要学习的内容要求：无



课程目标

完成本课程的学习之后，你能够：

- 了解测试的基础概念和基本原理
- 了解测试方法和测试的流程阶段
- 了解测试类型与关键技术及常用测试工具
- 通过学习认识到测试的重要性和意义



第一部分：测试基础概念

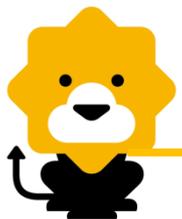
第二部分：测试方法

第三部分：测试流程和阶段

第四部分：测试类型和关键技术

第五部分：苏宁软件测试流程

第六部分：内容回顾



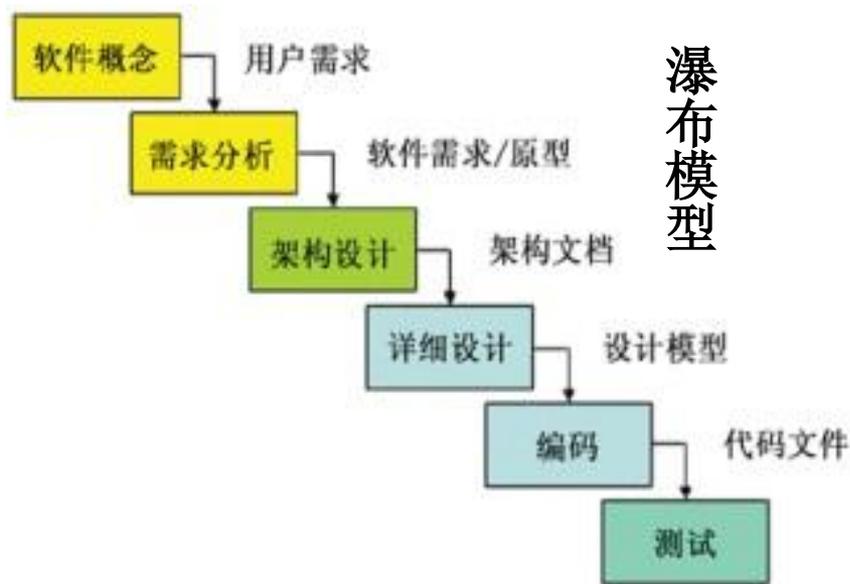
第一部分 软件测试的基本概念

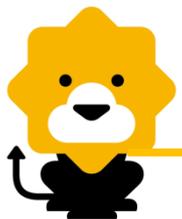
在当前角色下，谈谈你对软件测试的理解？

- 开发工程师
- 测试工程师
- 产品经理
- 用户

一些对软件测试的误解：

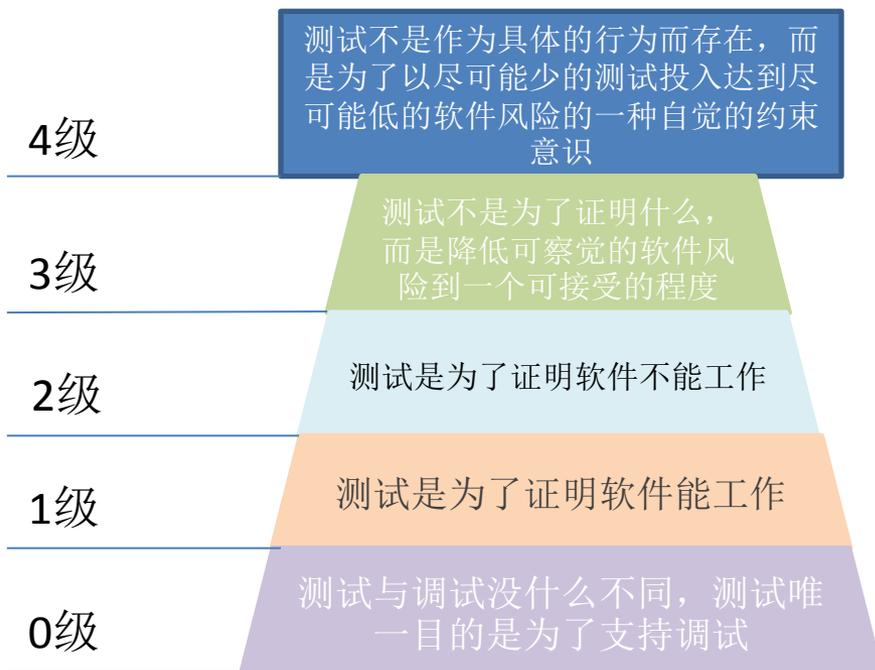
- “发布后软件出现了BUG，是软件测试人员的错。”
- “软件测试对于人员的要求不高，技术含量低。”
- “软件测试很简单，随便点点就好了，谁来都可以做。”
- “有时间就多测一些，没时间就少测一点。”
- “软件测试是测试工程师的事，与开发、产品无关。”
- “在软件的整个生命周期中，软件测试只是开发过程中的一个阶段。”





为什么要进行测试？

对于为什么要进行软件测试的5个层次的不同理解：



从0级到4级的认识提高基本上反应了从发现错误到预防错误的趋势。



测试的概念（IEEE中对测试的定义）

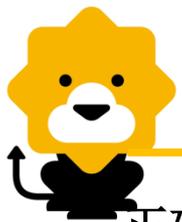
软件测试的定义？

使用人工或自动手段来运行或测试某个系统的过程，目的在于检验该系统是否满足规定的需求或是弄清预期结果与实际结果之间的差别。

软件测试的定义：软件测试是在规定的条件下对程序进行操作，以发现错误，对软件质量进行评估。

即：**软件测试是为了发现错误而执行程序的过程。**

软件测试的目的：不仅是为了发现软件缺陷与错误，是软件质量保证的关键，是对软件质量进行度量与评估，判断风险，并提高软件质量。



测试的原则

正确的理解：

- 软件测试是为了发现缺陷而执行程序的过程
- 一个好的测试案例，在于能发现至今尚未发现的缺陷
- 一个成功的测试是发现了至今为发现的多个缺陷的测试

软件测试的一些原则：

1. 应当把“尽早地和不断地进行软件测试”作为软件测试者的座右铭。
2. 一个测试案例必需包括两部分：对程序的输入数据的描述和对应正确输出结果的描述。
3. 程序员应当避免测试自己编写的程序。
4. 应当彻底检查每个测试的执行结果，以免遗漏。
5. 在设计测试案例时，应当包括合理(有效/预料)的输入和不合理(无效/未预料)的输入条件。
6. 集群现象：程序某部分存在更多错误的可能性，与该部分已发现错误的数量成正比。
7. 妥善保留测试计划、测试案例、缺陷记录和最终分析报告，为维护提供方便。
8. 检查程序是否“做了其不应该做的”、“未做其应该做的。”
9. 计划测试工作时不应默认假定不会发现错误。
10. 软件测试是一项极富创造性、极具智力挑战性的工作。



测试的核心是什么？

- *The systems meet the users' needs = “**has the right system been built**” 实现正确的系统（需求的正确性）*
- “实现正确的系统”是指我们要验证系统功能时首先需要保证需求是正确的，这是我们去验证系统的前提；
- *The systems meet the business requirements = “**has the system been built right**” 正确地实现系统（实现的正确性）*
- “正确的实现系统”是指通过测试来验证用户提出的需求是否被系统正确的实现



确认和验证

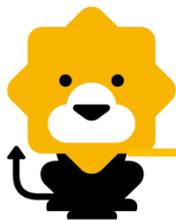
确认和验证的定义

- 确认（所收集需求的正确性）
 - 用来说明工作内容正确，遵循真正的用户需求
 - 通过检查或规定目标客观证据来确认具体预期用途的特定需求已经被履行
- 验证（系统实现的正确性）
 - 用来说明工作内容达到目标通过验证它是否依据上一步的工作内容
 - 通过检查或规定目标客观证据来确认具体需求已经被实现
- 通常**确认比验证更重要**
 - 我们可以想像一个极端的例子：“正确地建立了一个错误的系统！”



确认和验证

- 确认贯穿于测试活动的整个过程。需求分析、设计、开发、验证都需要不断确认需求
- 确认经常是测试活动中的薄弱环节
 - 确认要求很好地理解用户需求
 - 确认经常被开发、测试人员忽视
 - 确认要求大量的沟通和行业经验
 - 传统的测试技术更加侧重于验证
- 好的测试人员应该具备很强的需求确认意识



确认失败的后果

确认失败的后果：



客户是这样描述需求的



项目经理是这么理解的



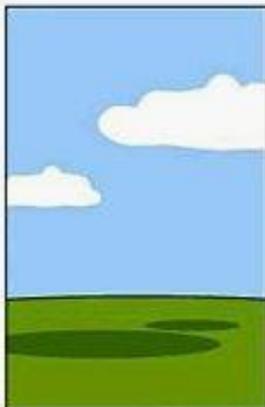
分析师是这么设计的



程序员是这么编写的



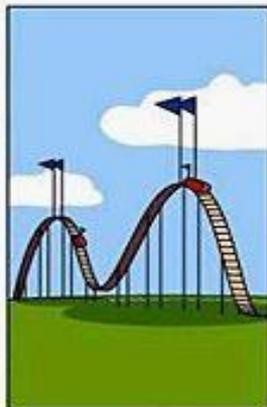
商业顾问是这么描绘的



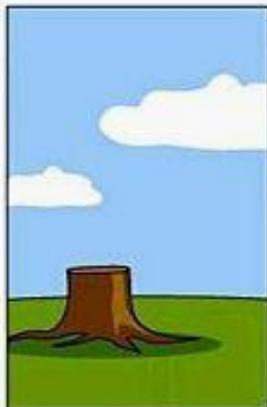
项目书写出来是这样的



操作中用了这样的工具



客户是这么建造的



提供的支持就是这个样子

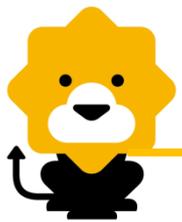


而这才是
客户真正需要的



测试的基本概念

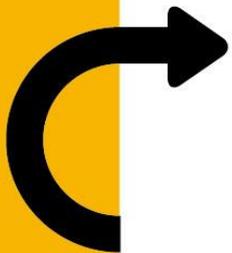
- 软件测试的对象：
 - (1) 软件测试并不只是程序测试
 - (2) 软件开发过程中所产生的需求规格说明书、概要设计规格说明书、详细设计规格说明书以及源程序都是软件测试的对象
 - (3) 软件测试贯穿于软件定义和开发的整个过程
- 通常，软件测试涉及的关键问题包括四个方面：
 - (1) 测试由谁来执行 —— Who
 - (2) 测试什么 —— What
 - (3) 什么时候进行测试 —— When
 - (4) 怎样进行测试 —— How



软件测试职业素质

作为软件测试工程师，应具备：

- ①具有探索精神
- ②具有创造性
- ③坚持不懈精神
- ④故障排除专家
- ⑤判断准确
- ⑥追求完美
- ⑦沟通能力
- ⑧耐心和细心



第一部分：测试基础概念

第二部分：测试方法

第三部分：测试流程和阶段

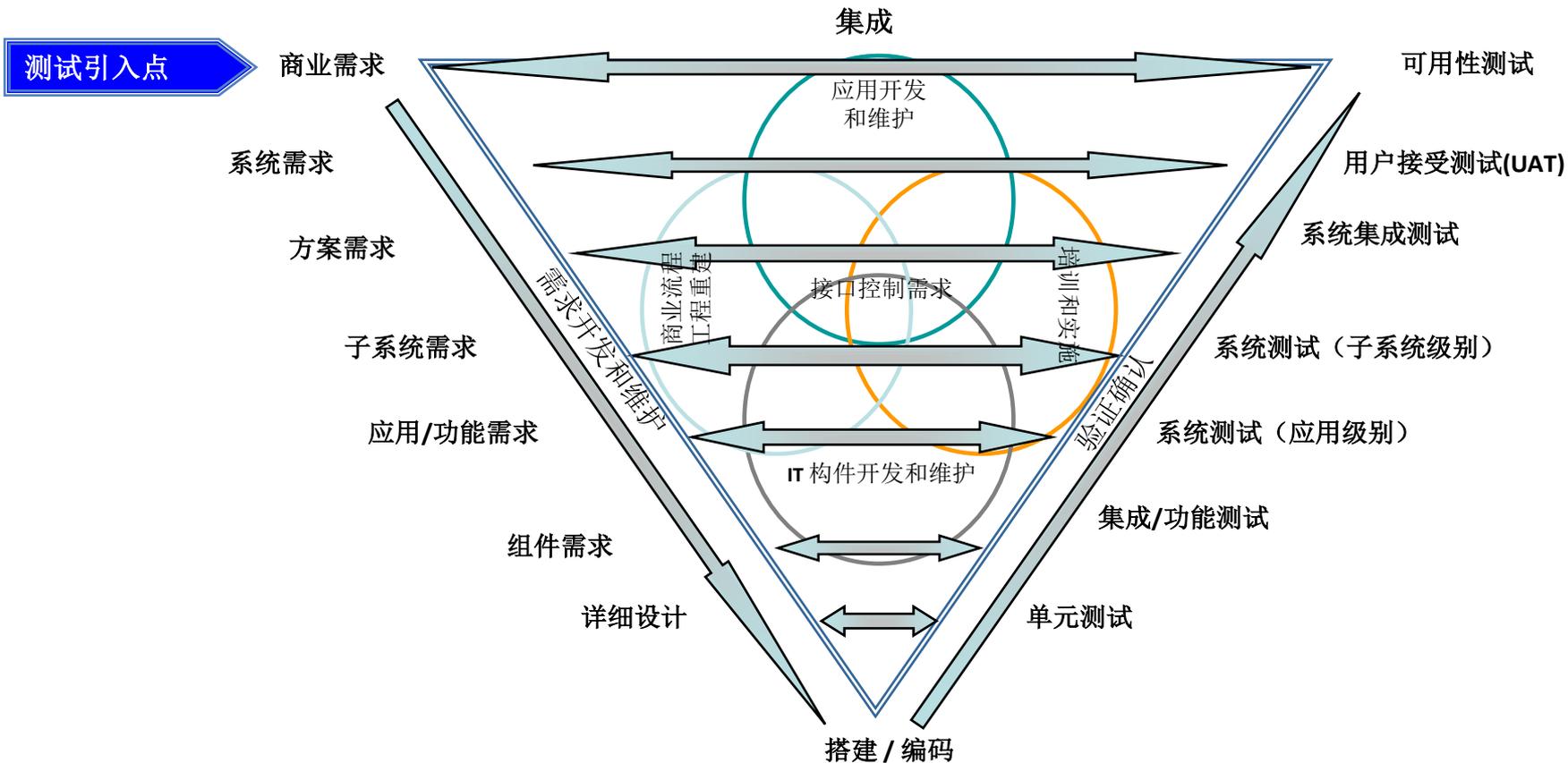
第四部分：测试类型和关键技术

第五部分：苏宁软件测试流程

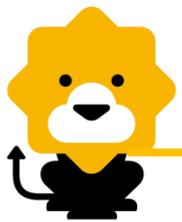
第六部分：内容回顾



第二章 软件测试的方法



需求确认是从初始的需求定义开始并贯穿至最终客户的接受，它对大型复杂项目的成功是至关重要的。验证需求可以降低风险并最小化重复的工作。



测试方法的分类



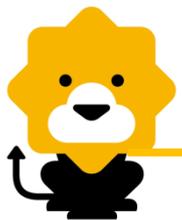
静态测试 - 指不运行被测应用本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查应用的正确性。

- 需求走查或者需求的最终评审
- 设计或者代码审查
- 测试计划审查
- 测试案例评审



动态测试 - 指通过运行被测应用，检查运行结果与预期结果的差异，并分析运行效率等来检测应用的实际情况。

- 在运行的系统中执行测试案例
- 模拟真实用户使用的场景测试
- 在生产环境在并行测试



静态测试的几种主要方法



评审：
来自同级或者其他人的非正式的检查，为产品提供控制和保证，如同级评审。



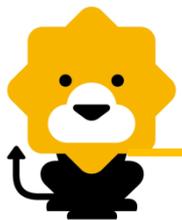
走查：
比评审正式，通过参与会议明确问题，并须采取的行动以及必要的跟踪。



审查：
最正式的静态测试方法，也最消耗时间和资源。



桌面检查：
作者自查产品，如自查代码或测试案例。



动态测试的几种主要方法



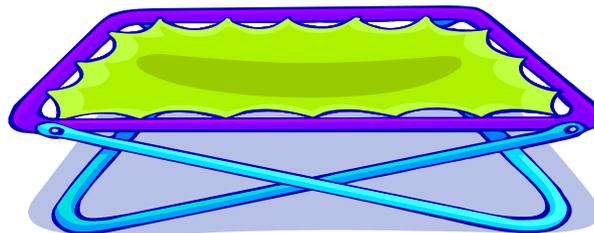
■ 白盒测试

深入到代码一级的测试，检查程序内部逻辑结构相关信息



■ 黑盒测试

着眼于程序外部的结构，不考虑内部逻辑结构，从用户角度出发

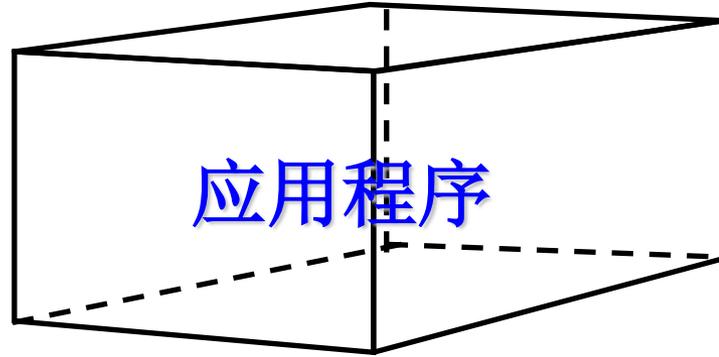


■ 错误猜测

没有用到任何特殊的方法，利用直觉和经验猜测出错的可能类型



白盒测试的定义



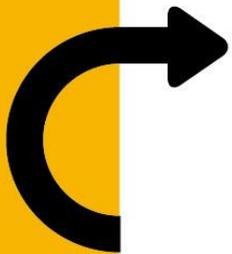
- 白盒测试需要完全了解程序结构和处理过程，它按照程序内部逻辑测试程序，检验程序中每条通路是否按预定要求正确工作
- 白盒测试将被测程序看作一个打开的盒子，测试者能够看到被测源程序，可以分析被测程序的内部结构，此时测试的焦点集中在根据其内部结构设计测试案例
- 白盒测试要求是对某些程序的结构特性做到一定程度的覆盖，或者说这种测试是“基于覆盖率的测试”



黑盒测试的定义



- 黑盒测试是在程序接口进行测试，它只是检查程序功能是否按照规格说明书的规定正常使用
- 黑盒测试的基本观点是：任何程序都可以看作是从输入定义域映射到输出值域的函数过程，被测程序被认为是一个打不开的黑盒子，黑盒中的内容（实现过程）完全不知道，只明确要做到什么
- 黑盒测试主要根据规格说明书设计测试案例，并不涉及程序内部构造和内部特性，只依靠被测程序输入和输出之间的关系或程序的功能设计测试案例



第一部分：测试基础概念

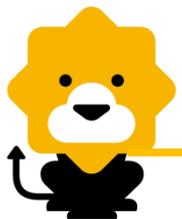
第二部分：测试方法

第三部分：测试流程和阶段

第四部分：测试类型和关键技术

第五部分：苏宁软件测试流程

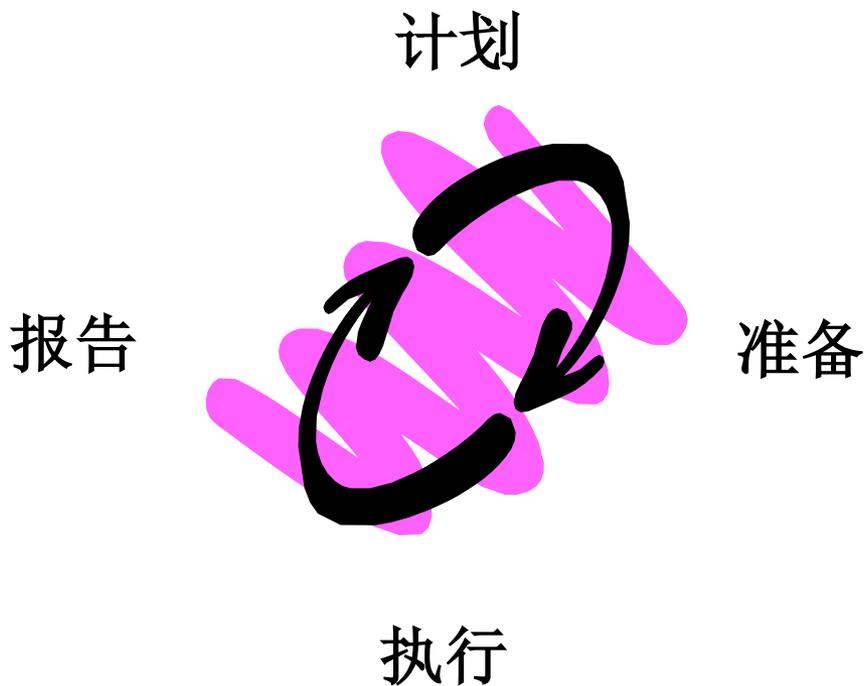
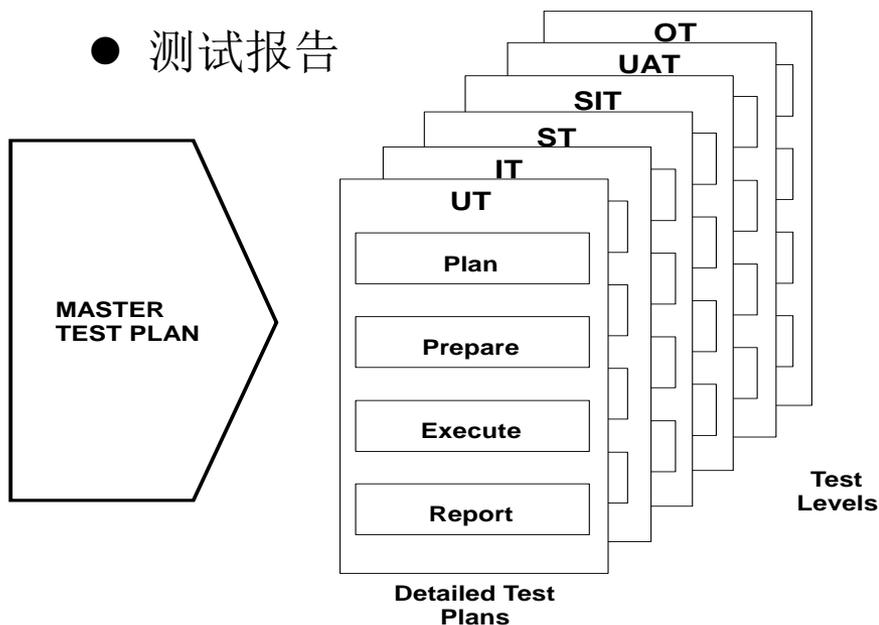
第六部分：内容回顾



全生命周期测试-测试流程

基本测试流程包括4步：

- 测试计划
- 测试准备
- 测试执行
- 测试报告





测试流程-主要任务

• 测试计划

- 测试目标
- 资源范围
- 前提条件
- 测试规划
 - 测试阶段
 - 测试类型
- 准入准出标准
- 风险分析分析
- 测试环境
- 测试数据
- 测试时间安排
- 主要交付件

• 测试准备

- 维护项目计划
- 资源
- 需求跟踪矩阵
- 执行计划
- 测试案例
 - 全量案例
 - 冒烟案例
- 测试数据
- 测试环境
- 状态报告

• 测试执行

- 维护项目计划
- 执行测试案例
- 更新跟踪矩阵
- 更新测试案例
- 报告测试状态
- 缺陷管理
- 测试执行记录
 - 单号
 - 截图

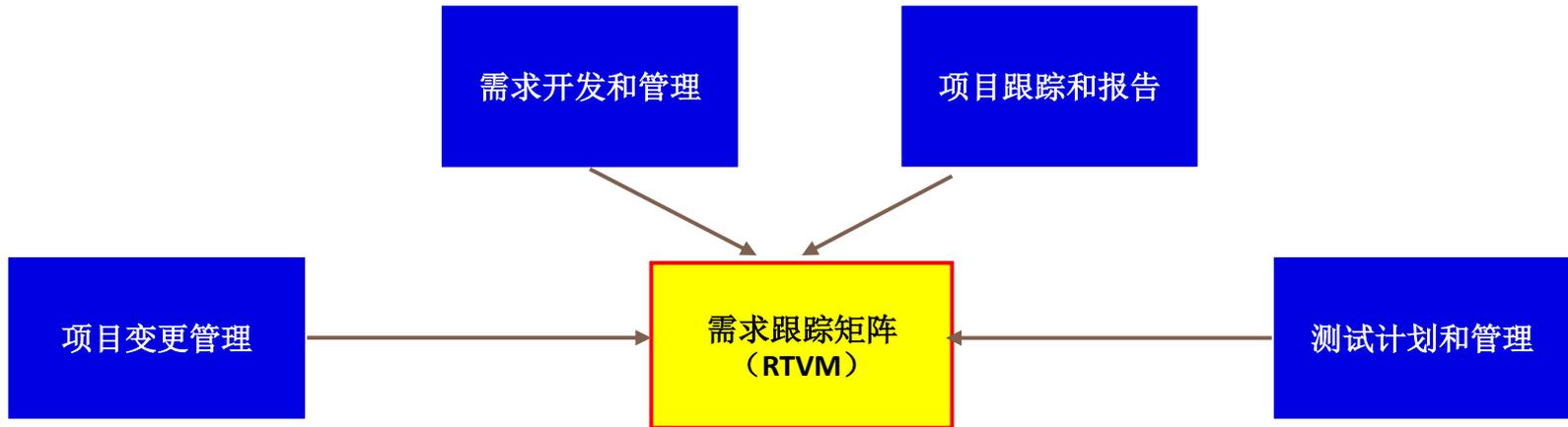
• 测试报告

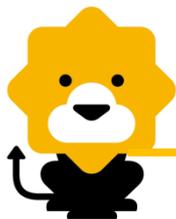
- 测试范围
- 测试案例执行总结
- 缺陷总结
- 测试覆盖总结
- 风险与质量分析
- 测试结论



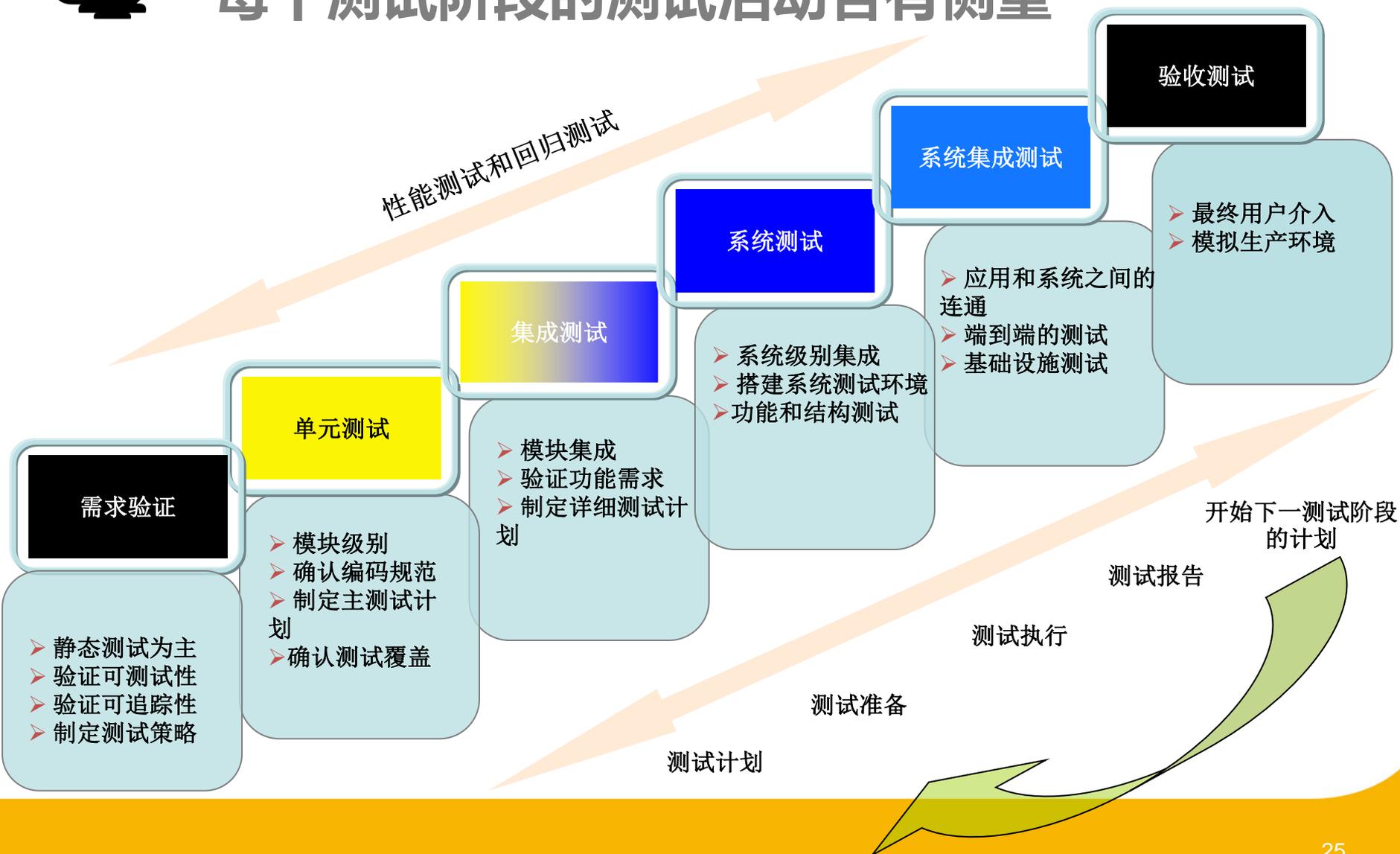
使用需求跟踪建立需求和测试的对应关系

- 及时地为所有项目需求提供一个完整的概述, 每个需求的内容和这些需求已经发生的变更
- 验证为完成业务/系统需求所编写的子需求和工作件
- 跟踪需求到测试, 确保所有的需求都被解决
- 确定如何测试/验证工作件, 并提供给团队一个机制来保证:
 - 设计被完成
 - 测试案例的编写被完成
 - 测试执行被完成



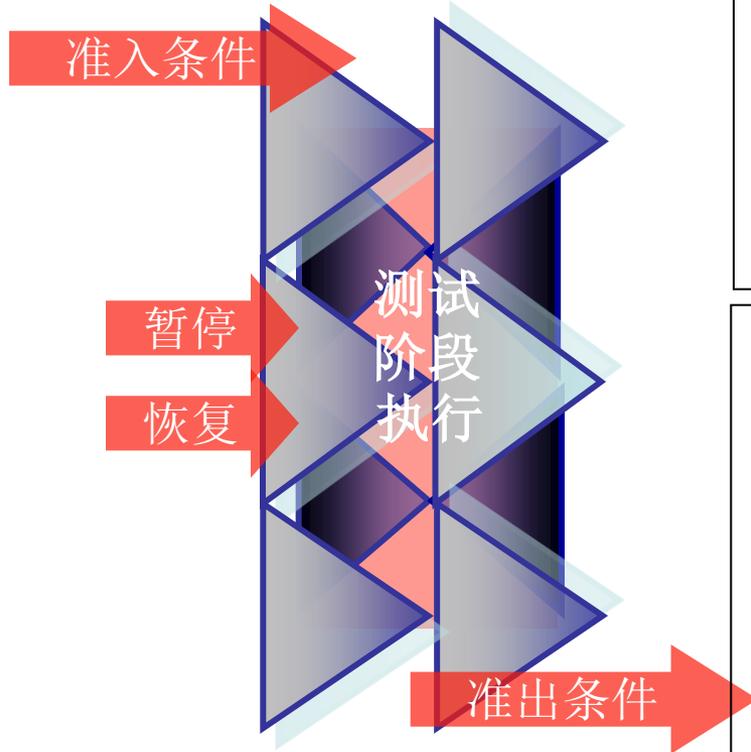


在应用开发生命周期中，涵盖了以下测试阶段， 每个测试阶段的测试活动各有侧重





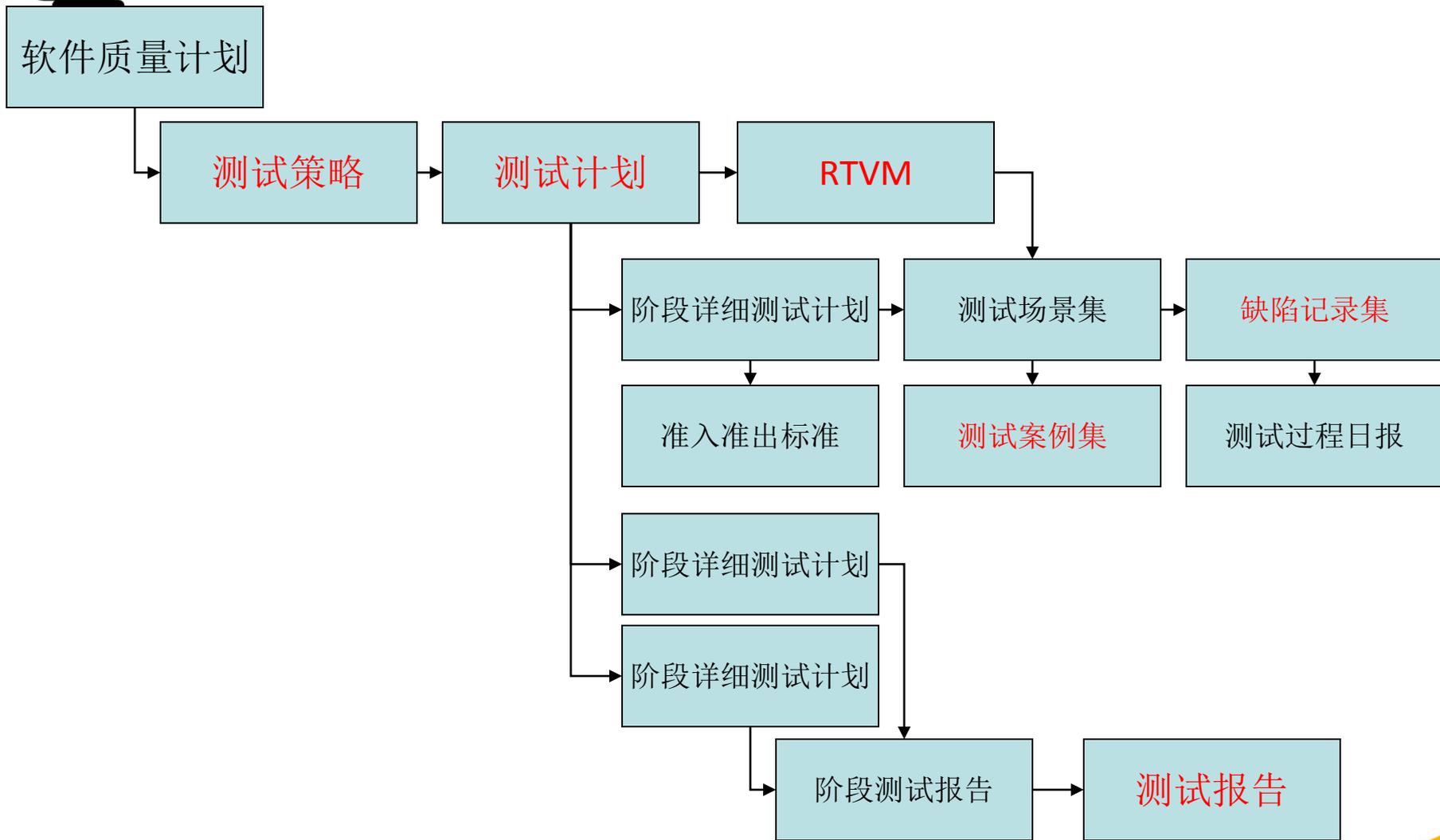
测试准入和准出

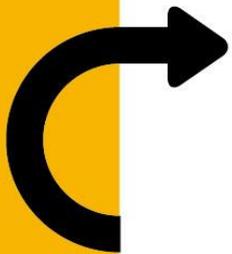


- 对于有效管理任一活动的流程的方法是应该建立准入和准出条件，测试活动也不例外，为每个测试阶段建立有效明确的标准：
 - **准入条件**就是指能够开始某一活动所需要具备的因素
 - **准出条件**就是指能够结束某一活动所需要具备的因素
- 以下是基本标准：
 - 准入：有文档证明具备测试策略和主测试计划
 - 有文档证明项目具有如下工作件：
 - 详细测试计划
 - 需求跟踪矩阵及测试案例
 - 冒烟测试通过
 - 准出：测试案例已经全部被正确执行
 - 严重程度为严重、阻塞和致命的缺陷已经被修复并被测试通过
 - 遗留问题均得到领导的同意



测试的主要交付成果





第一部分：测试基础概念

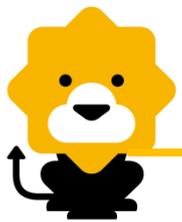
第二部分：测试方法

第三部分：测试流程和阶段

第四部分：测试类型和关键技术

第五部分：苏宁软件测试流程

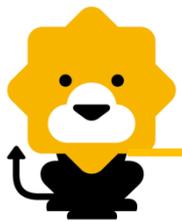
第六部分：内容回顾



为什么需要测试关键领域

- 穷尽测试通常不可能（不现实或不经济）
- 项目往往没有足够的资源
- 对业务角度和技术角度任何一方的忽略，都会造成风险
- 通过风险分类和分级来分配资源和测试工作量是一个好的选择





什么是测试关键领域

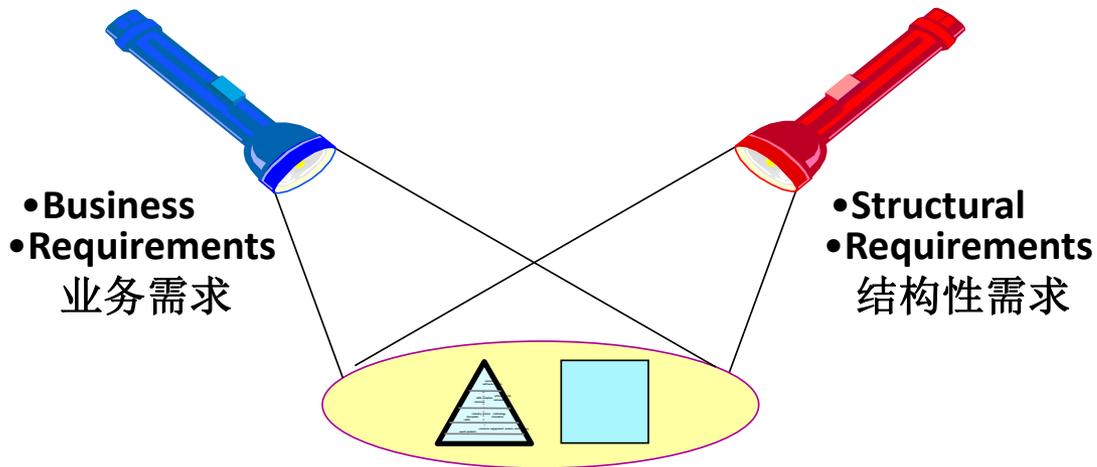
- **测试关键领域：** 为确保满足业务和技术的需求，而必须对一些系统特征所做的测试
- **测试焦点：**
 - 什么是对系统用户最重要的因素（最终用户，开发人员，操作/维护人员等）
 - 什么是风险最大的部分
- **测试关键领域证明：** 通过认真的计划，技术协作，以及与客户的沟通和协议，对关键领域的测试可以确保正确的系统被正确的建立，使测试的投资可以得到最大的回报





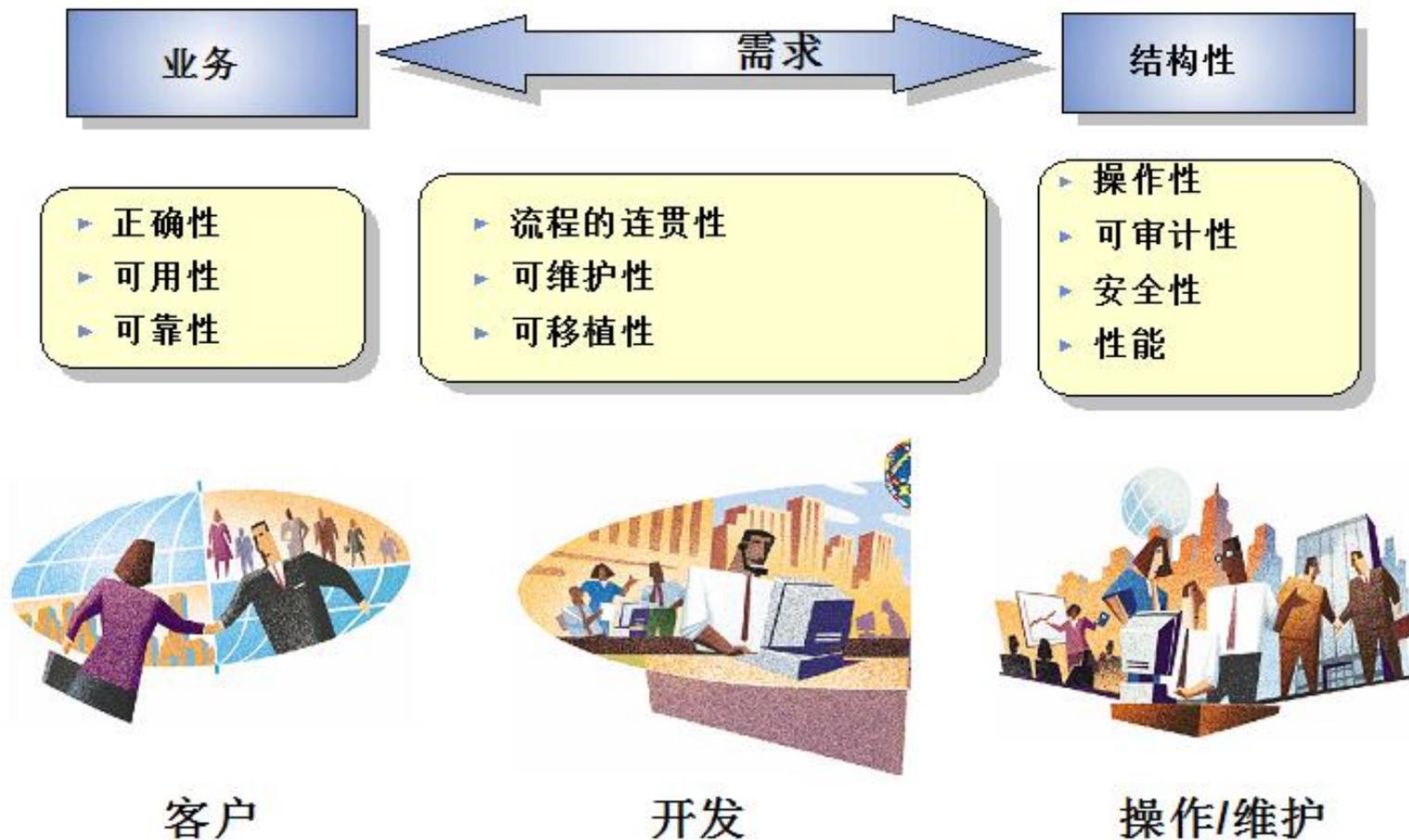
测试关键领域 (Test Focus Areas)

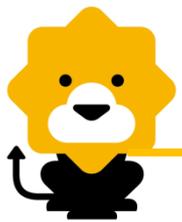
测试关键领域的确定必须同时考虑业务需求和结构/技术需求，来确保两方面都被满足





10个常用的测试关键领域





什么是测试类型

- 测试类型是用来测试业务和技术需求的各种手段，以验证正确的系统（符合业务需求）被正确的建立（技术结构健全）
- 测试类型包括：
 - 功能性测试
 - 结构性测试





功能性测试

功能性测试是用来保障系统属性满足用户需求的测试

- **Audit and Controls** 审计和控制的测试
 - 验证控制的充足和有效性，及数据处理结果的完整性
- **Conversion** 转换测试
 - 验证转换的程序，数据和过程与旧的事物的协调性，如新旧系统切换
- **User Documentation and Procedures** 用户文档和流程的测试
 - 验证说明文档的准确性，用户文档包括操作说明书
- **Error Handling** 错误处理的测试
 - 验证系统发现和应对异常的能力
- **Function** 功能测试
 - 确保业务功能需求被实现了。验证每个功能操作都和具体需求，内外部设计一致。
- **Interface/Inter-system** 接口/系统间的测试
 - 验证应用程序和系统功能连接正确。



功能性测试（续）

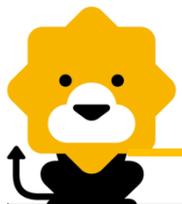
- **Installation** 安装的测试
 - 任何安装和运行环境不同于开发地点的都需要做安装测试。影响用户体验
- **Parallel** 平行测试
 - 平行测试用来比较新旧系统中运行相同数据的结果。
- **Regression** 回归测试
 - 验证当系统的一部分发生改变时，没有影响到其它不需要改变的部分。
- **Transaction Flow** 交易流的测试
 - 验证交易流从进入系统到完成（出系统）整个流程的完整和正确性。
- **Usability** 可用性的测试
 - 验证最终的产品用户友好并易于使用。



结构性测试

结构性测试是用来保障系统技术健全的测试

- **Backup and Recovery** 备份和恢复的测试
 - 验证应用程序失败后重启的能力。如何从程序错误、硬件失效和数据错误中恢复过来。
- **OPC/JCL Testing** OPC/JCL的测试
 - 验证JCL或者Job执行的正确性。
- **Performance** 性能测试
 - 验证应用程序在类生产环境下达到期望的预期的性能指标。
- **Security** 安全的测试
 - 验证应用软件可为数据提供足够级别的保护和保密功能。
- **Stress/Volume** 压力/容量测试
 - 验证在数据量峰值的情况下，应用软件的性能指标可以接受。



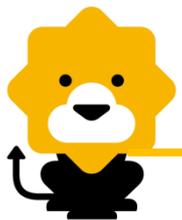
测试关键领域VS测试类型

测试类型 \ 测试关键领域	功能性测试											结构性测试				
	Audit & Cont.	Conversion	Documentation	Error-Handling	Function	Interface	Installation	Parallel	Regression	Trans. Flow	Usability	Backup & Rec	Oper./Job Strm.	Performance	Security	Stress/Volume
可审计性	●		●	●						●		●			●	
流程连贯性					●	●	●		●			●	●			
正确性	●	●	●	●	●	●	●	●	●	●		●	●		●	
可维护性			●									●	●			
可操作性						●	●						●			
性能						●		●			●		●	●		●
便携性						●	●									
可靠性		●		●	●	●		●	●	●		●				
安全性	●				●	●									●	
可用性			●	●		●					●					

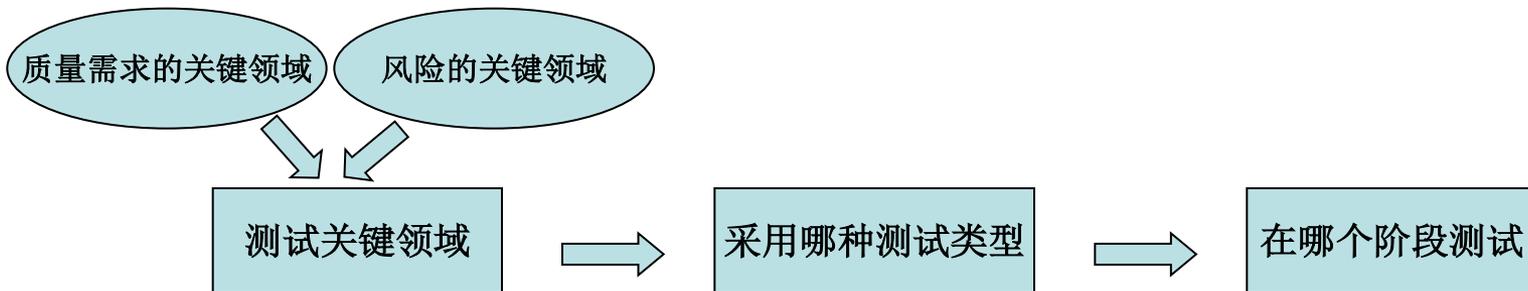


测试类型VS测试阶段

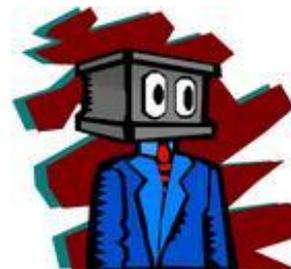
测试类型 \ 阶段	单元测试	集成测试	系统测试	系统集成测试	验收测试	操作性测试
审计和控制测试		●	●	●		
转换测试	●		●	●		●
文档和流程测试			●	●	●	●
错误处理测试	●	●	●	●	●	●
功能测试	●	●	●	●	●	
安装测试			●			●
接口和系统交互测试			●	●	●	●
平行测试			●		●	●
回归测试	●	●	●	●	●	●
交易流测试			●	●	●	
可用性测试			●		●	
备份和恢复性测试	●		●	●		●
作业流处理测试			●	●	●	●
操作性测试			●	●		●
性能测试			●	●		●
安全性测试	●		●	●		●
压力/容量测试			●	●		

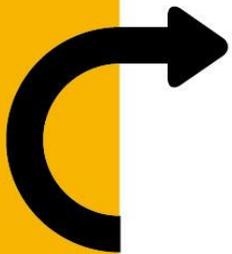


小结：识别和对应测试关键领域和测试类型的重要性



- 如果没有测试关键领域的识别和测试类型与之对应，将失去对后续测试工作的指导性，将失去和客户、需求人员及各团队之间对测试工作目标的认同，或失去此认同的基线
- 如果没有将测试类型分布到适当的测试阶段，将导致各测试阶段分工不清，测试不足或者测试过度





第一部分：测试基础概念

第二部分：测试方法

第三部分：测试流程和阶段

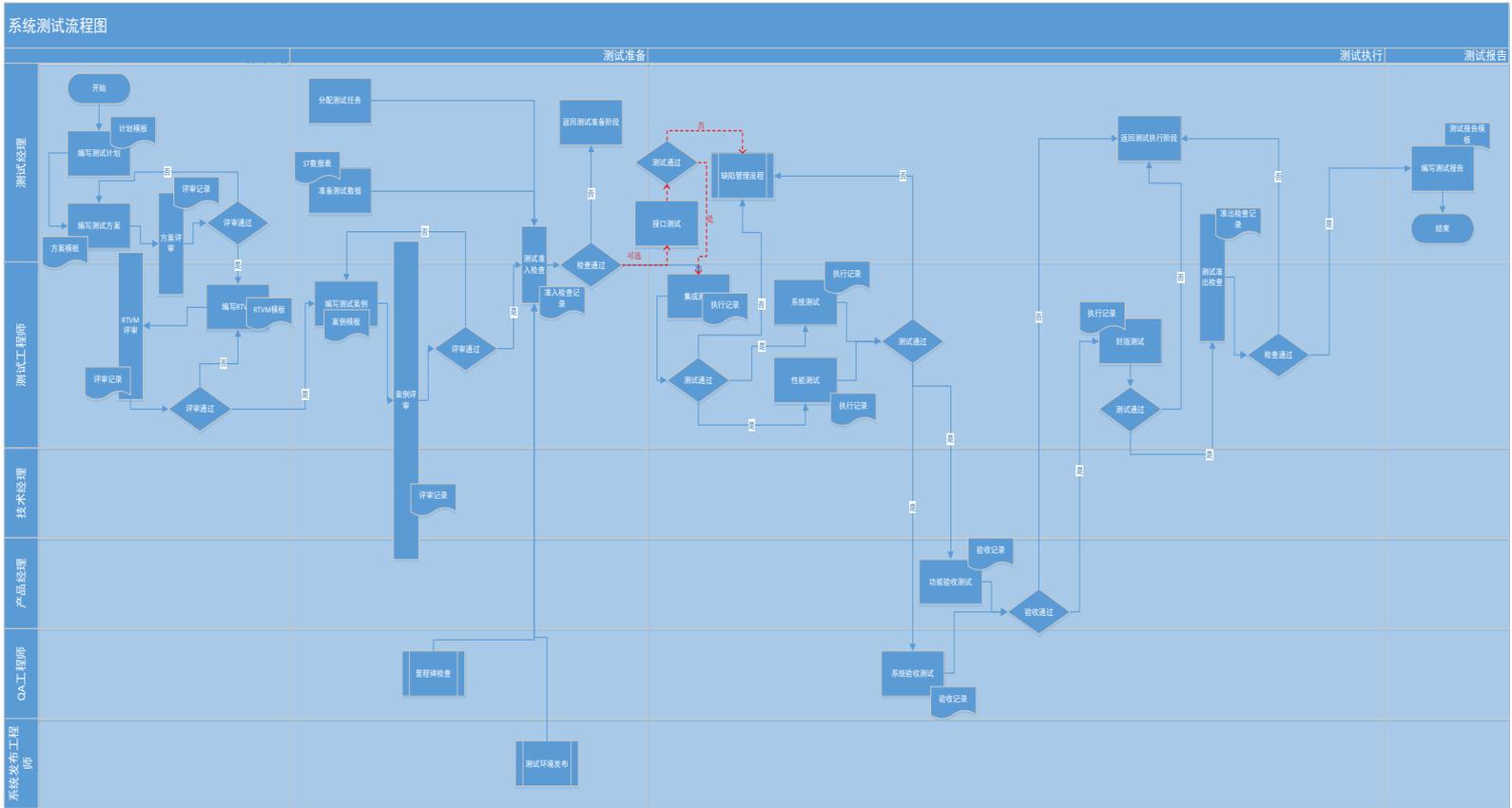
第四部分：测试类型和关键技术

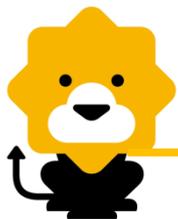
第五部分：苏宁软件测试流程

第六部分：内容回顾



苏宁软件测试流程





苏宁软件测试流程-测试计划阶段

测试计划阶段一般在项目初期，开发概设阶段。测试计划阶段主要任务：完成测试计划，测试方案，RTVM文档，并终稿通过。测试计划阶段流程活动

活动	责任人	输出	目的
编写测试计划	测试经理	《测试计划初稿》 《测试计划评审稿》	制定整个测试活动的时间计划，资源，节点任务
编写测试方案	测试经理	《测试方案初稿》 《测试方案评审稿》	为测试定义策略和计划，为后续的测试活动提供指导
编写RTVM矩阵	测试工程师	《RTVM初稿》 《RTVM评审稿》	整理完成所有功能点并识别出测试所需要验证的功能点



测试计划模板



系统测试方案模板



性能测试方案模板



性能测试申请_项目_yyyy_mm_d



RTVM (模板)



苏宁软件测试流程-测试准备阶段

测试准备阶段需要完成测试案例编写，测试数据准备，测试环境准备，测试准入检查工作。测试准备阶段流程活动：

活动	责任人	输出	目的
编写测试案例	测试工程师	《测试案例初稿》 《测试案例评审稿》	根据需求和设计文档编写测试场景及案例，使测试执行人员可以按步骤直接执行测试案例
准备测试数据	测试经理	《测试数据列表》	为测试执行活动准备测试数据，保证系统执行正常开展
准备测试环境	运维工程师	运行正常的测试环境	维护测试环境，保证测试执行工作能正常开展
准备准入检查	测试经理	准入检查表	对测试相关条件的准备情况进行检查，识别风险，确保测试能够按计划开始执行



测试案例（模板）



测试案例评审表



ST测试数据表



发布实施流程及规范



测试准入标准模板



冒烟测试表-测试部



苏宁软件测试流程-测试执行阶段

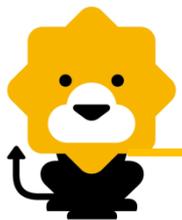
测试人员根据测试案例对系统进行测试执行，保证系统质量。测试执行分为功能测试，系统测试，性能测试等。

活动	责任人	输出	目的
缺陷管理	测试工程师	Bug单	确保所有缺陷都被及时有效的进行处理
功能测试	测试工程师	测试执行记录、截图	执行测试案例，检验系统返回结果与期望结果是否一致，验证系统各模块功能是否按系统需求文档实现
系统集成测试	测试工程师	测试执行记录、截图	执行测试案例，检验系统返回结果与期望结果是否一致，验证系统及系统间各功能是否按需求实现



苏宁软件测试流程-测试执行阶段

活动	责任人	输出	目的
性能测试	性能测试工程师	测试脚本	执行测试案例，检验系统返回结果与期望结果是否一致，验证系统性能是否符合业务需求
验收测试	QA工程师，产品经理	验收测试执行记录	对项目进行端到端的系统验收测试，保证系统核心功能能正确使用。
封版测试	测试工程师	测试执行记录、截图	收回开发人员对当前项目提交代码的权限，进行系统执行测试案例，检验系统返回结果与期望结果是否一致，验证系统及系统间各功能是否按需求实现



苏宁软件测试流程-测试执行阶段

活动	责任人	输出	目的
测试准出检查	测试经理	准出检查表	根据测试准出标准逐一检查，验证系统是符合准出条件



缺陷管理流程说明



缺陷管理流程



测试日报模板

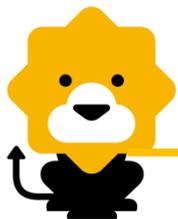


性能测试工作日报



苏宁易购性能测试测试准出标准模板
准入模板





苏宁软件测试流程-测试报告

测试报告描述了软件测试过程中出现的问题及处理结果，对过程中发现的问题进行分析，并得出测试结论。测试报告是测试阶段最后的文档产出物，包括产品质量和测试过程的评价，测试报告基于测试中的数据采集以及对最终的测试结果分析。是领导评估项目是否上线的核心依据之一。

活动	责任人	输出	目的
测试报告编写	测试经理	测试报告文档	根据测试情况进行分析，给出测试结论



测试报告模板



通用性能测试报告模板



第一部分：测试基础概念

第二部分：测试方法

第三部分：测试流程和阶段

第四部分：测试类型和关键技术

第五部分：苏宁软件测试流程

第六部分：内容回顾



课件要点：

1. 软件测试的定义是什么？软件测试有何目的？
2. 软件测试的核心包含哪两个方面，以及它们各自的定义是什么？
3. 软件测试的对象是什么？软件测试涉及哪几个方面的关键问题？
4. 测试方法分为哪两类？静态测试和动态测试的主要方法分别是哪些？
5. 白盒测试和黑盒测试的定义分别是什么？
6. 测试阶段主要分为哪几个？
7. 测试关键领域的确定需要满足哪两方面的需求？
8. 测试类型包含哪两方面？
9. 苏宁目前所用的缺陷管理和需求管理工具是什么？

Thanks!

