

---

Software Testing

# System Testing (1)

---

School of Computer Science & Engineering  
Sun Yat-sen University

Instructor: Guoyang Cai  
email: [isscgymail@mail.sysu.edu.cn](mailto:isscgymail@mail.sysu.edu.cn)

*Approaches & Technologies*



中山大學  
SUN YAT-SEN UNIVERSITY



## OUTLINE

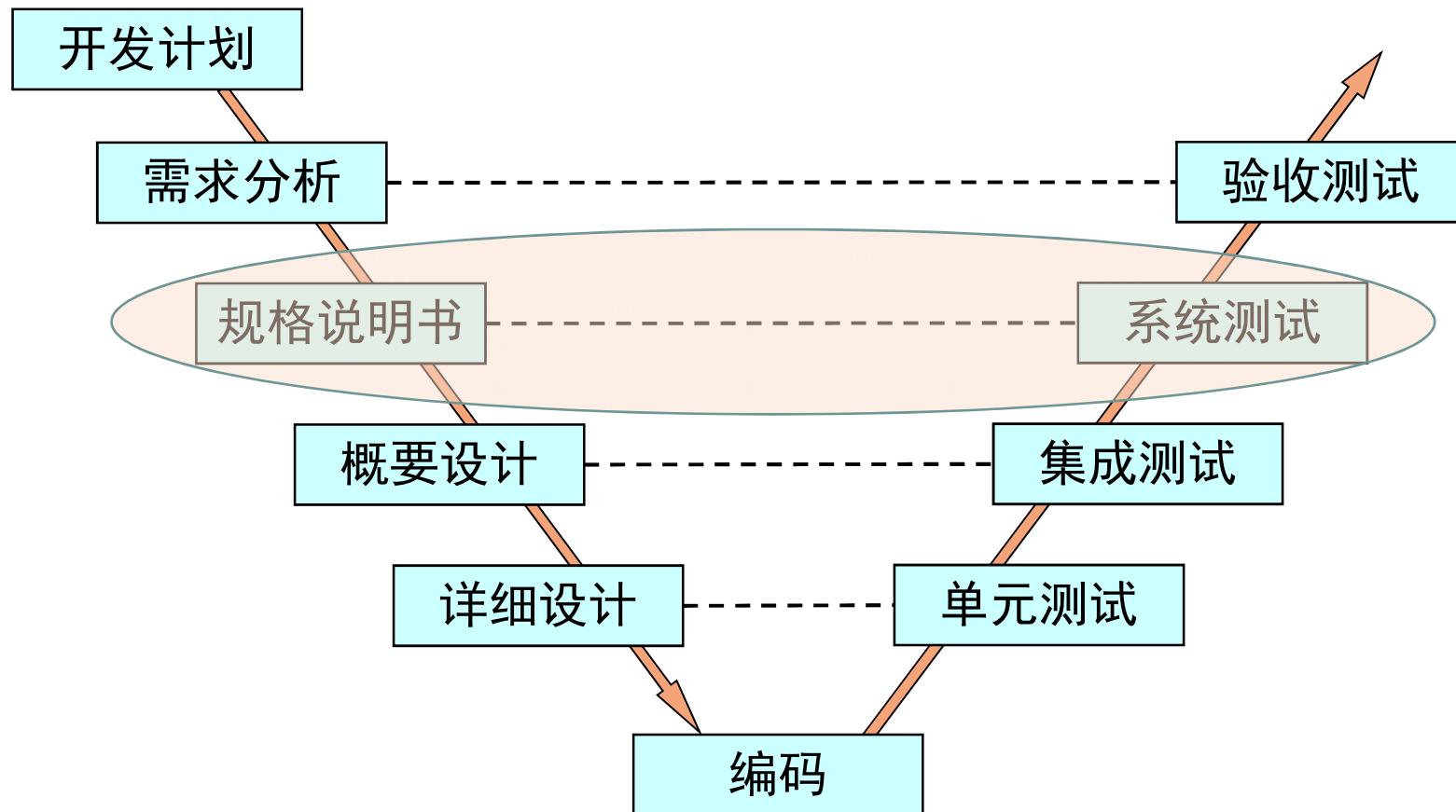


- 7.1 系统测试概述
- 7.2 系统测试内容分类
- 7.3 系统测试步骤
- 7.4 系统测试的测试用例分类设计
- 7.5 系统故障模型与攻击测试
- 7.6 软件攻击突破口测试
- 7.7 软件故障注入测试
- 7.8 系统可用性测试和完整性测试
- 7.9 动态测试工具



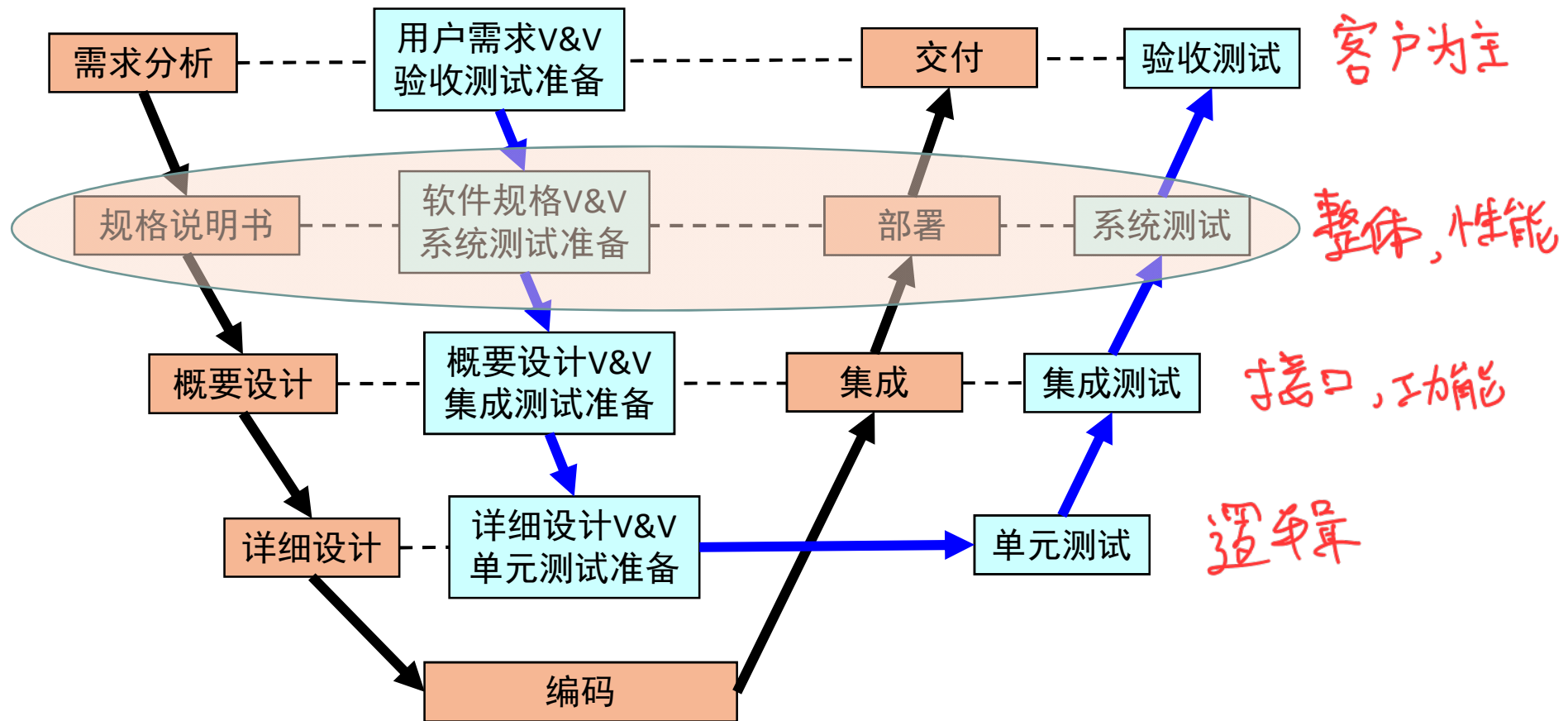
## 系统测试的概念

### V 模型



## 系统测试的概念

### W 模型





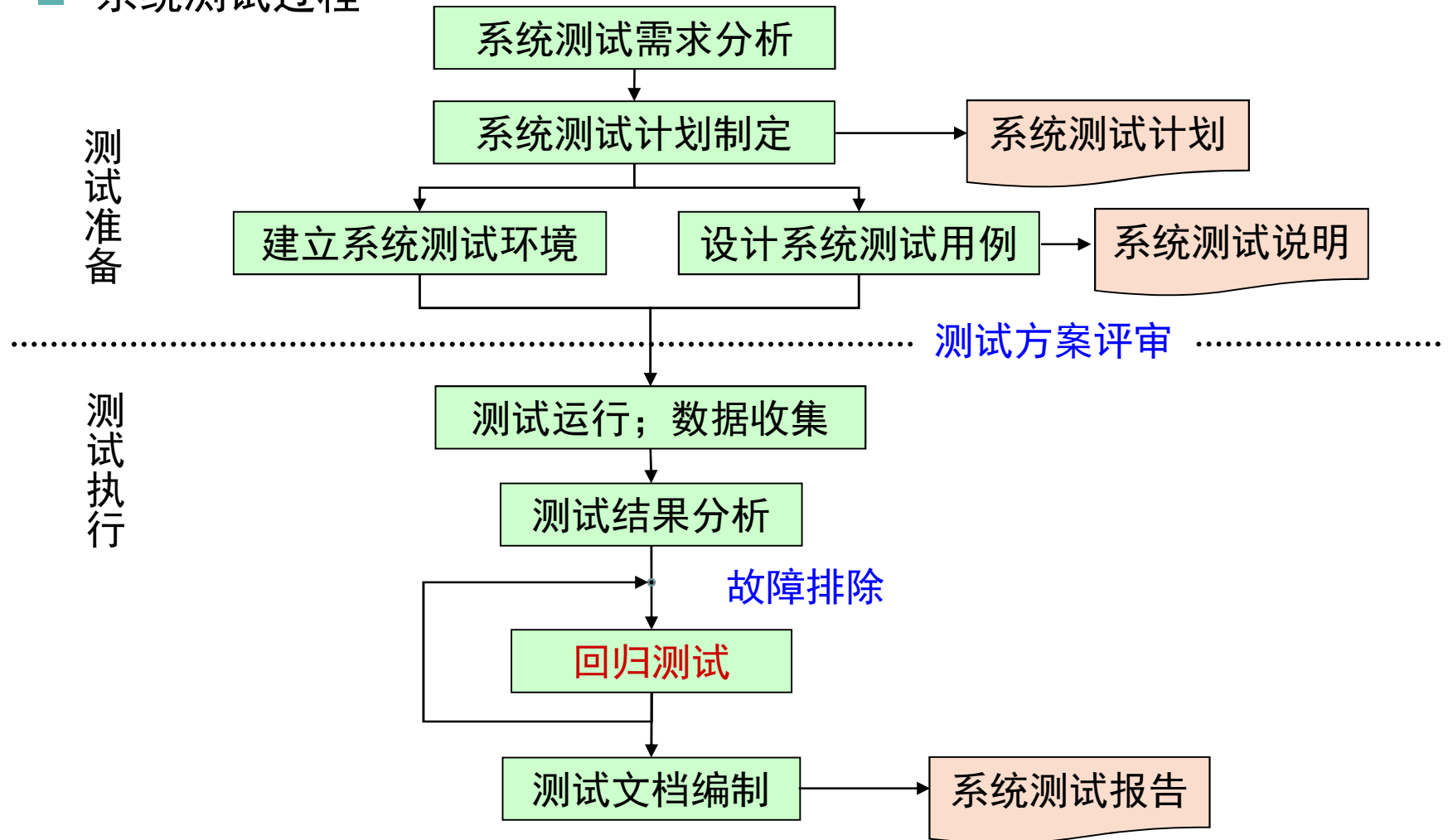
### ■ 系统测试的概念

- 系统测试将被测软件作为一个运行系统的整体，判断其是否符合规定。
  - 被测软件与计算机硬件和软件支持平台、数据平台、操作人员等系统元素结合形成完整的运行系统，并在实际运行(使用)环境下进行测试。
  - 系统测试按照规格说明书的需求进行功能和性能测试，以发现缺陷并度量产品质量，进一步检验系统文档的完整性。
    - 单元或模块之间的接口错误由单独的集成测试发现。
  - 系统测试一般使用黑盒测试技术，由独立的测试人员完成。



## ■ 系统测试的概念

### ■ 系统测试过程





## ■ 系统测试的概念

### ■ 系统测试的阶段性任务

活动名称	输入工作	输出工作	角色
分析测试需求	软件需求文档	系统测试需求	测试分析员
制定测试计划	软件需求文档 软件项目计划	系统测试计划	测试设计员
设计测试用例	系统测试计划 软件设计文档	系统测试用例	测试设计员
准备测试脚本	系统测试计划 系统测试用例	系统测试脚本	测试设计员
执行测试过程	测试计划、被测软件 测试用例、测试脚本	系统测试结果	测试员
评估测试结果	系统测试结果	软件测试报告	测试设计员 测试员



### ■ 系统测试的需求分析

- 系统测试需求分析的主要根据是系统需求文档 (软件需求规格说明书), 用于确定系统测试的内容或具体对象。
- 系统测试需求的形成
  - 系统测试需求必须是可观察、可测量、可评价的行为;
  - 系统需求文档中的每一个功能性描述将派生一个或多个系统测试需求, 测试用例和测试需求之间不一定形成一对一的关系;
  - 系统需求文档中的性能描述、安全性描述等也将派生出一个或多个系统测试需求;
  - 此外, 系统需求文档中的每一个描述信息可以生成至少一个系统测试需求。





### ■ 系统测试的需求分析

#### ■ 系统测试的功能测试需求

- 系统功能性测试需求来源于系统需求文档对被测试对象的功能性说明。
  - 对于软件需求规格说明书中的每一个功能描述，将至少派生一个系统功能性测试需求。
- 每个系统需求用例至少会派生一个系统功能性测试需求。
  - 对于每个系统需求用例事件流，系统测试需求的详细列表至少会包括一个系统功能性测试需求。



### ■ 系统测试的需求分析

#### ■ 系统测试的性能测试需求

- 系统性能测试需求来源于系统需求文档对测试对象的指定性能行为。
  - 系统性能需求在软件按需求规格说明书中的性能描述部分或其它补充说明中声明。
  - 系统性能通常被描述为对响应时间和资源使用率的某种评测。
- 系统性能需要在各种条件下进行评测，这些条件包括：
  - 不同的工作量和/或系统条件
  - 不同的用例/功能
  - 不同的配置



- 系统测试的需求分析
  - 其它系统测试需求
    - 配置测试
    - 安全性测试
    - 容量测试
    - 强度测试
    - 故障恢复测试
    - 负载测试等



### ■ 系统测试策略

- 测试策略用于说明某项特定测试工作的一般方法和目标。一个好的测试策略应该包括下列内容：
  - (1) 测试目标
  - (2) 实施的测试类型
  - (3) 采用的测试技术
  - (4) 用于评估测试结果和测试是否完成的标准
  - (5) 对测试策略所述的测试工作存在影响的特殊事项
- 系统测试策略主要针对系统测试需求确定测试目标、测试类型以及实施测试的方法和技术。



### ■ 系统测试策略

#### ■ 测试目标

- 确认软件满足软件需求规格说明书的要求

#### ■ 测试类型

- 系统测试类型一般包括**功能测试**、**性能测试**、接口测试、负载测试、强度测试、容量测试、安全性测试、**配置测试**、故障恢复测试、**安装测试**、文档测试、用户界面测试。
- 功能测试、性能测试、配置测试和安装测试一般情况下必需执行。其它测试类型可以根据软件项目的具体要求进行裁剪。

#### ■ 采用的测试技术

- 主要采用黑盒测试技术设计测试用例



### ■ 系统测试环境

- 软件运行的三种环境：**开发环境**、**测试环境**、**用户环境**。
  - 测试环境在测试计划和测试用例中需要事先定义和规划。
  - 规划良好的测试环境通常接近于用户环境，同时兼顾开发环境的成本和效率。
- 系统测试的硬件环境和软件环境
  - 硬件环境
    - 包括实施测试必需的由服务端设备、客户端设备、网络连接设备以及打印机/扫描仪等辅助硬件设备所构成的环境。
  - 软件环境
    - 支持被测软件运行的操作系统、数据库以及其他支持软件系统 (例如网络系统、浏览器等) 构成的环境。
    - 共存软件：考虑数据共享、内存冲突以及其它影响。
  - 其他专用环境以及支撑上述各种环境的软硬件产品的版本。



### ■ 系统测试环境

#### ■ 测试环境规划

- 分析用户环境中可能对被测软件的运行产生影响的硬件和软件配置，在此基础上建立测试环境。

#### ■ 测试环境的要求

- 尽可能接近用户真实环境
- 选用满足系统运行最低要求的软硬件配置
- 选用流行的的软硬件配置
- 环境保持纯净和独立性
- 环境不受病毒或其它安全风险的干扰



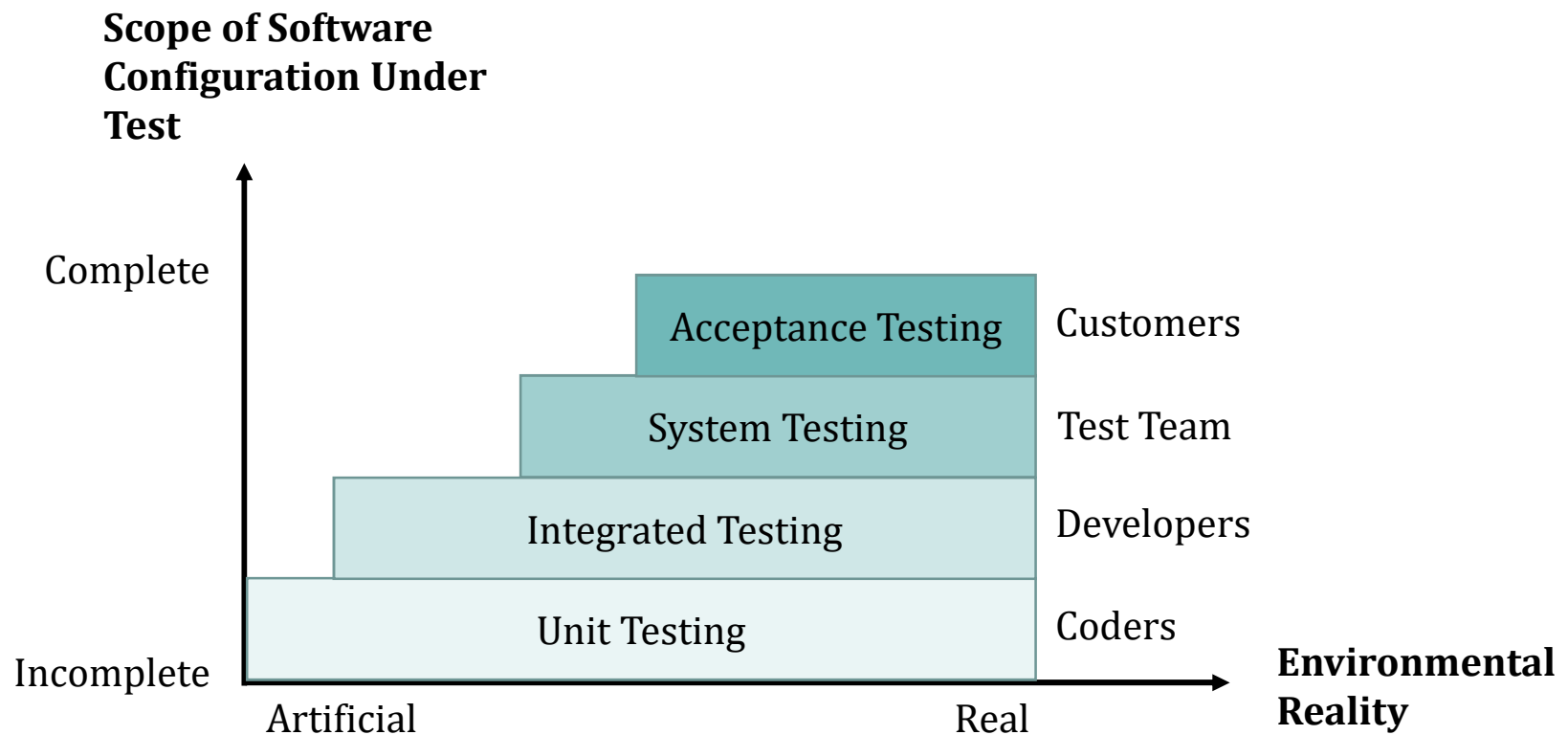
# 系统测试概述



## ■ 系统测试环境

### ■ 测试环境规划

#### ■ 测试环境的要求







### ■ 系统测试环境

#### ■ 测试环境规划

Attribute	Level			
	Unit	Integration	System	Acceptance
People	Developers	Developers & Testers	Testers	Testers & Users
Hardware OS	Programmers' Workbench	Programmers' Workbench	System Test Machine or Region	Mirror of Production
Cohabiting Software	None	None	None/Actual	Actual
Interfaces	None	Internal	Simulated & Real	Production
Source of Test Data	Manually Created	Manually Created	Production & Manually Created	Production
Volume of Test Data	Small	Small	Large	Large
Strategy	Unit	Group of Units/Builds	Entire System	Simulated Production



### ■ 系统测试环境

#### ■ 建立测试环境的步骤

- 确认硬件以及必要的设备配备适当；
- 确认操作系统以及必要的数据环境配备适当；
- 安装应用程序；
- 安装测试工具 (必要时需要开发专用工具)；
- 设置专用文件；
  - 包括将这些文件与测试所需的数据相对应
- 建立与应用程序通信的实用程序。



### ■ 系统测试的交付文档

#### ■ 系统测试的交付文档包括

- 系统测试计划
- 系统测试计划评审报告
- 系统测试用例
- 系统测试用例评审报告
- 系统测试脚本
- 系统测试脚本评审报告
- 系统测试报告
- 系统测试报告评审报告
- 缺陷问题清单

#### ■ 交付文档的形式可以根据需要进行裁减或改造

- 例如：只提交系统测试计划、系统测试设计、系统测试结果、缺陷问题报告。



### ■ 系统测试的技术要求

#### ■ 系统测试一般应符合以下技术要求：(GB/T 15532-2008)

- (1) 系统的每个特性应至少被一个正常测试用例和一个被认可的异常测试用例所覆盖；
- (2) 测试用例的输入应至少包括有效等价类值、无效等价类值和边界数据值；
- (3) 应逐项测试系统 / 子系统设计说明规定的系统的功能、性能等特性；
- (4) 应测试软件配置项之间及软件配置项与硬件之间的接口；
- (5) 应测试系统的输出及其格式；
- (6) 应测试运行条件在边界状态和异常状态下，或在人为设定的状态下，系统的功能和性能；
- (7) 应测试系统访问和数据安全性；
- (8) 应测试系统的全部存储量、输入/输出通道和处理时间的余量；



### ■ 系统测试的技术要求

#### ■ 系统测试一般应符合以下技术要求：(续)

- (9) 应按系统或子系统设计文档的要求，对系统的功能、性能进行强度测试；
- (10) 应测试设计中用于提高系统安全性、可靠性的结构、算法、容错、冗余、中断处理等方案；
- (11) 对完整性级别高的系统，应对其进行安全性、可靠性分析，明确每一个危险状态和导致危险的可能原因，并对此进行针对性的测试；
- (12) 对有恢复或重置功能需求的系统，应测试其恢复或重置功能和平均恢复时间，并对每一类导致恢复或重置的情况进行测试；
- (13) 对不同的实际问题应外加相应的专门测试。

#### ■ 对具体的系统，可根据软件测试合同(或项目计划)及系统的重要性、完整性级别等要求对上述内容进行裁剪。



## ■ 系统测试设计

- 系统测试设计的主要任务是设计测试用例。
  - 测试用例的衡量标准：多、快、好、省。
  - 测试用例库是宝贵的经验积累的结果。
  - 软件攻击 (Software Break) 是在系统测试中常用的一种测试用例设计方法。
- 系统测试的设计目标
  - 测试效率
  - 测试结果
- 系统测试设计的计划性
  - 高度的计划性有助于提高结果的可靠性，降低管理风险。
    - 缺点是降低了测试人员的经验作用和灵感发挥。
  - 以智力为核心、以系统为框架的原则
    - 智力：逻辑性、灵活性和技巧性
    - 系统：组织性、计划性和流程性



- 系统测试流程
  - 系统测试的控制过程
  - 系统测试的度量过程



### ■ 系统测试项目

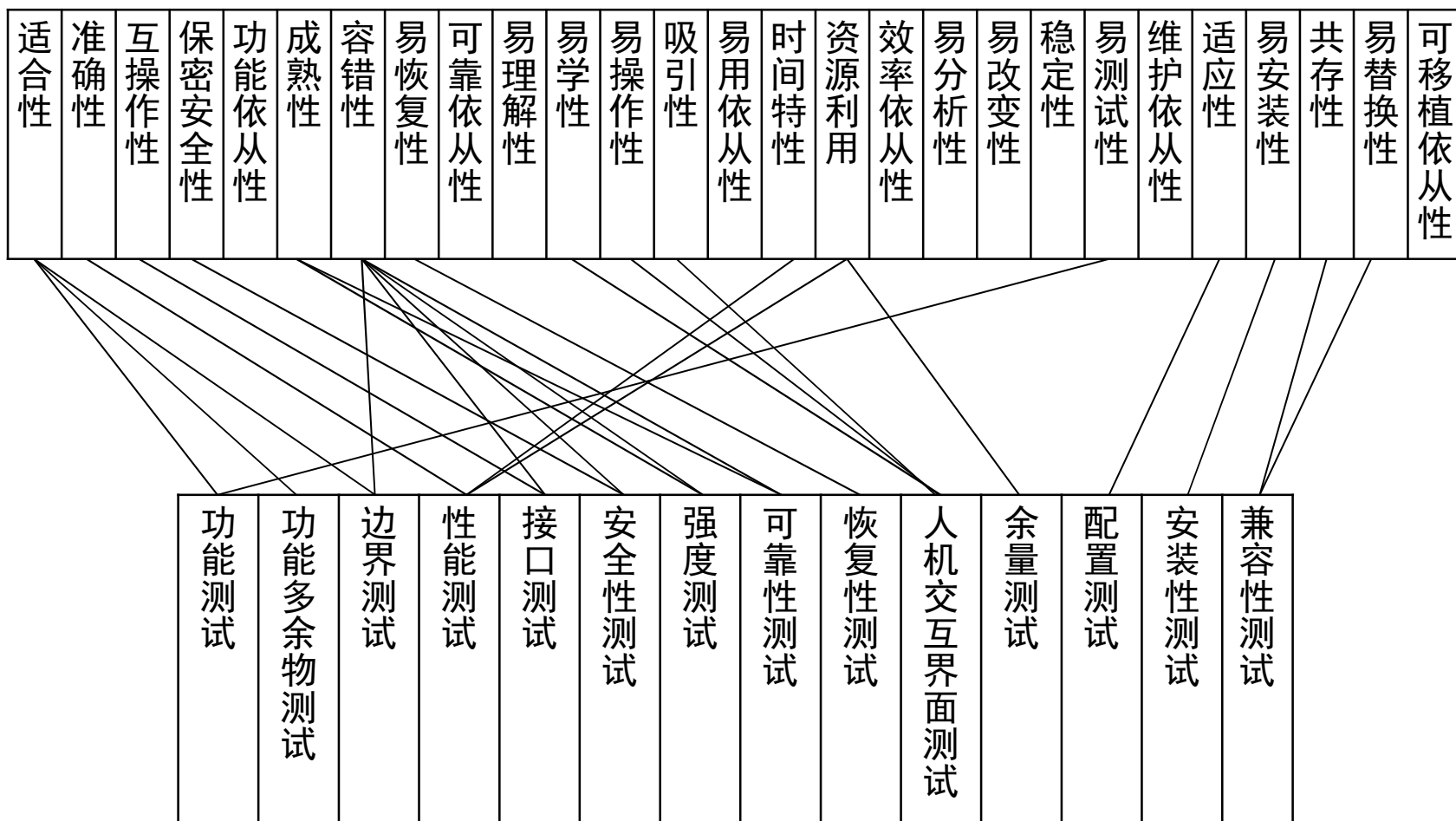
- GB/T 15532-2008：按 GB/T 16260.1 质量特性分类的系统测试内容。
- 按传统分类的系统测试内容包括下列18项：

- |         |         |
|---------|---------|
| ■ 功能测试  | ■ 安全测试  |
| ■ 性能测试  | ■ 恢复测试  |
| ■ 负载测试  | ■ 回归测试  |
| ■ 压力测试  | ■ 健全测试  |
| ■ 疲劳测试  | ■ 交付测试  |
| ■ 易用性测试 | ■ 演练测试  |
| ■ 安装测试  | ■ 背靠背测试 |
| ■ 配置测试  | ■ 度量测试  |
| ■ 文档测试  | ■ 比较测试  |



## ■ 系统测试项目

## ■ GB/T 16260.1 质量特性与传统分类测试内容之间的对应关系.





### ■ 功能测试

#### ■ 目标

- 对产品的功能进行测试。
  - 检验可实现性；
  - 检验实现的正确性。

#### ■ 方法

- 覆盖产品的功能。

#### ■ 工具

- 回归测试时可以使用工具。



### ■ 性能测试

#### ■ 目标

- 对产品的性能进行测试。
  - 检验性能是否达标；
  - 检验性能是否能够保持。

#### ■ 方法

- 覆盖系统的性能需求，一般和负载测试结合使用。

#### ■ 工具

- 在需要大访问量时候尤其需要使用工具。



### ■ 负载测试

#### ■ 目标

- 人为设置**高负载** (大数据量、大访问量)。
  - 检查系统在高负载下是否发生功能或性能上的故障。

#### ■ 方法

- 人为生成大数据量，并利用工具模拟频繁并发访问。

#### ■ 工具

- 一般需要使用工具。



### ■ 压力测试

#### ■ 目标

- 人为设置系统环境，使系统**资源紧缺**。
  - 检查系统在系统资源紧张情况下是否发生功能或者性能上的故障。

#### ■ 方法

- 人为减少可用的系统资源。
  - 系统资源包括：内存、硬盘、网络、CPU、数据库反应时间 ...

#### ■ 工具

- 一般需要使用工具。



### ■ 疲劳测试

#### ■ 目标

- 在一段连续时间内 (比如 72小时) 保持系统功能的频繁使用。
  - 检查系统在**持续长时间**运行状态下是否发生功能或者性能上的故障。

#### ■ 方法

- 人为设置不同功能的连续重复操作。

#### ■ 工具

- 一般需要使用工具。



### ■ 易用性测试

#### ■ 目标

- 检查系统界面和功能是否易于学习掌握；
- 检查使用方式是否规范一致；
- 检查提示信息是否清晰准确；
- 检查误导用户的可能性。

#### ■ 方法

- 一般与功能测试结合使用。

#### ■ 工具

- 用户操作、观察 (录像)、反馈并评估的方式。



### ■ 安装测试

#### ■ 目标

- 检查系统安装过程是否能够安装所有需要的文件/数据并进行必要的系统设置；
- 检查系统安装过程是否会破坏其他文件或配置；
- 检查系统安装过程是否可以中止并恢复现场；
- 检查系统是否能够正确卸载并恢复现场；
- 检查安装和卸载过程的用户提示和功能是否出现错误；
- 有时候将安装测试作为功能测试的一部分。





### ■ 配置测试

#### ■ 目标

- 设置不同的硬件配置、不同的操作系统和应用软件环境。
  - 检查系统在符合需求说明的不同系统配置下是否发生功能或者性能上的故障。

#### ■ 方法

- 一般需要建立专门的测试实验室。
  - 需要附加成本。



### ■ 文档测试

#### ■ 目标

- 检查系统的文档是否齐全；
- 检查是否有多余文档或者死文档；
- 检查文档文法是否规范，内容是否符合 5C 要求
  - Correct
  - Clear
  - Concise
  - Complete
  - Consistent

#### ■ 方法

- 一般由单独的文档测试人员进行评审。



### ■ 安全测试

#### ■ 目标

- 检查系统是否带有病毒；
- 检查系统是否正确加密；
- 检查系统在非授权的内部或外部用户访问或故意破坏时候是否出现错误。



### ■ 恢复测试

#### ■ 目标

- 人为使系统发生灾难 (系统崩溃、硬件损坏、病毒入侵等), 检查系统是否能恢复被破坏的环境和数据。



### ■ 回归测试

- 回归测试是一种选择性重新测试，目的是检查先前测试发现的软件缺陷是否得到了正确的处理，并验证被测软件在修改之后各项功能依然正确 (确认未引入新的缺陷)。
- 目标
  - 检查系统变更后是否引入新的错误或旧的错误重新出现。
- 方法
  - 对系统修改部分进行针对性测试以保证其符合需求；
  - 运行一系列的测试以确认现有功能仍然符合需求。
- 工具
  - 一般需要使用工具 (特别是自动化测试工具)；
  - 一般依赖于测试用例库和缺陷报告库。



### ■ 健全测试

#### ■ 目标

- 检查系统的功能和性能是否**基本**可以正常使用。
  - 确定是否可以继续进行系统测试的其他内容。

#### ■ 方法

- 正常安装系统；
- 使用正常情况下的测试用例对主要功能进行测试；
- 同时检查系统文档是否齐全。



### ■ 交付测试

#### ■ 目标

- 关闭所有缺陷报告；
- 确保系统达到预期的交付标准。

#### ■ 方法

- 一般需要结合回归测试，并谨慎处理新出现的缺陷。
- 交付测试也称为稳定期测试，可以独立于系统测试分类。



### ■ 演练测试

#### ■ 目标

- 在系统交付给用户之前，利用相似的用户环境进行测试。





### ■ 背靠背测试

#### ■ 目标

- 设置一组以上的测试团队，在互相不进行沟通的情况下独立执行相同的测试项目，评估测试团队的效果并期望发现更多的错误；
- 需要附加成本。



### ■ 度量测试

#### ■ 目标

- 在系统中人为地设置错误 (播种), 并根据被发现的比例来估计系统中遗留的错误数量。



### ■ 比较测试

#### ■ 目标

- 对本产品的旧版本或者竞争产品作同等测试，确定当前系统的优势和劣势。

#### ■ 比较测试属于系统测评的内容。

- 例：BenchMarking 是一种特殊的比较测试。



### ■ 小结

- 前述18项测试内容的选择与测试目标、测试资源、软件系统特点和业务环境有关。
- 系统测试可以由独立第三方测试。
  - 目的：
    - 加强软件质量保证工作
    - 提供对软件产品的客观评价
  - 优点：
    - 专业性优势
    - 独立性优势
  - 缺点：
    - 需要大量附加成本
    - 存在商业技术机密的保护问题



### ■ 系统测试的基本步骤

#### ■ 系统测试的基本步骤包括：

- (1) 理解软件和测试目标，制订测试计划；
- (2) 设计测试用例；
- (3) 运行测试用例并处理测试结果；
- (4) 评估测试用例和测试策略。



### ■ 系统测试的基本步骤

#### (1) 理解软件和测试目标，制订测试计划

##### ■ 目的

- 建立**软件故障模型**；
- 了解测试目标，确定测试策略和测试计划。

##### ■ 任务

- 了解软件的功能和业务背景、用户环境；
- 了解软件的开发背景和系统结构、技术选型；
- 了解软件的质量历史、版本变化；
- 了解测试目标和资源限制；
- 确定测试策略和测试计划。



### ■ 系统测试的基本步骤

#### (1) 理解软件和测试目标，制订测试计划 (续)

##### ■ 方法

- 阅读软件使用手册
  - 理解软件运行环境和用户行为；
  - 了解同类软件的功能和使用；
  - 了解软件要解决的问题域和解域 (业务背景知识)。
- 试运行软件
  - 熟悉软件功能；
  - 确定软件基本可以测试。
- 了解软件体系结构、技术选型、开发环境和工具；
- 阅读早期版本测试报告，以及单元和集成测试报告；
- 确定测试人员限制和时间限制，制定初步测试策略和测试计划，确定测试结束标准。



### ■ 系统测试的基本步骤

#### (1) 理解软件和测试目标，制订测试计划 (续)

##### ■ 结果

- 建立软件故障模型，指导发现下列错误：
  - 用户环境与测试环境不一致；
  - 没有测试过的代码或功能；
  - 没有测试过的输入组合、极端环境或功能使用方法、使用顺序。
- 建立测试策略和初步测试计划：
  - 了解测试目标和资源限制；
  - 按照出现错误的性能价格比制定设计测试用例的优先级；
  - 确定初步的测试计划。





### ■ 系统测试的基本步骤

(1) 理解软件和测试目标，制订测试计划 (续)

#### ■ 结果 (续)

- 确定测试结束标准 (退出机制)。
  - 已经解决的错误没有重现
  - 所有缺陷报告已经关闭
  - 所有测试用例全部执行完毕
  - 遗留的未发现错误数量可以被接受
  - 错误播种、错误发生曲线分析、历史数据
  - 市场和管理因素、预算和时间耗尽因素
  - 不属于技术层面的结束标准
  - 测试失败或者部分失败的表现



### ■ 系统测试的基本步骤

#### (2) 设计测试用例

##### ■ 目的

- 设计能够尽可能多、快、好、省地发现错误的测试用例。

##### ■ 任务

- 理解软件故障模型；理解已有测试用例库；设计具体的测试用例。

##### ■ 方法

- 软件攻击法
  - 一般情况下采用的方法；
  - 基于故障模型：经验、历史数据/错误、软件开发和运行环境。
- 因果图法、边界值分析法
  - 针对小型系统或者系统的简单功能采用的方法。



### ■ 系统测试的基本步骤

#### (2) 设计测试用例 (续)

##### ■ 结果

- 获得测试用例。
- 测试用例文档由简介和测试用例两部分组成。
  - 简介部分描述测试目的、测试范围、定义术语、参考文档、概述等；
  - 测试用例部分逐一系列示各个测试用例；
  - 测试用例包含的要素：参见 [Chap.6.1 Design of Test Cases](#)。
- 例：IE4.0 的测试用例数目达到10万。



### ■ 系统测试的基本步骤

#### (3) 运行测试用例并处理测试结果

##### ■ 目的

- 使用测试用例发现错误并关闭错误。

##### ■ 任务

- 运行测试用例并记录结果；
- 评估测试结果并记录缺陷；
- 处理缺陷直至缺陷关闭。
  - 修改、延迟处理、不修改、不是错误。



### ■ 系统测试的基本步骤

#### (3) 运行测试用例并处理测试结果 (续)

##### ■ 方法

- 选择测试用例库中的测试用例运行；
- 选择新设计的测试用例运行；
- 录制/回放或者笔录中间步骤和结果；
- 记录执行过程中的灵感：但不要轻易改变本次任务；
- 分析测试结果，并尽量重现和优化错误步骤；
- 填写缺陷报告，尽可能提供错误分析和修改建议；
- 认真审核错误处理结果并及时关闭缺陷。

##### ■ 结果

- 运行结果纪录；
- 新的测试用例设计思路纪录；
- 缺陷报告。



### ■ 系统测试的基本步骤

#### (4) 评估测试用例和测试策略

##### ■ 目的

- 检验测试用例和测试策略的有效性，必要时进行修改完善
- 增加测试经验

##### ■ 任务

- 根据测试结果修改、合并测试用例，必要时将其文档化；
- 维护测试用例库：
  - 增加新的测试用例 (尤其是揭示了错误的测试用例)；
  - 慎重删除不必要的测试用例；
  - 修改刚刚使用的测试用例 (根据测试结果进行修订)；
  - 合并部分测试用例。
- 根据测试结果修正测试策略和测试计划，产生新的测试用例设计思路。



### ■ 系统测试的基本步骤

#### (4) 评估测试用例和测试策略 (续)

##### ■ 方法

##### ● 基于经验

- 用例发现的问题；问题出现的原因；
- 用例如何发现问题；能否更快地发现问题；
- 用例可否合并；
- 设计新的测试用例。

##### ● 流程控制

- 借助工具实现测试用例库的维护。

##### ■ 结果

- 优化的测试用例库；
- 优化的软件故障模型；
- 优化的测试策略和测试计划；
- 增加的测试经验和新的测试用例设计思路。



## Lecture 23. System Testing (1)

# End of Lecture

