SOFTWARE TESTING

Software Testing

Classification of Software Testing

School of Computer Science & Engineering
Sun Yat-sen University

Instructor: Guoyang Cai
email: isscgy@mail.sysu.edu.cn

Approaches & Technologies

中山大學
SUN YAT-SEN UNIVERSITY

OUTLINE

- 2.1 软件缺陷
- 2.2 软件测试概述
- 2.3 软件测试的过程和方法
- 2.4 基于软件生命周期的软件测试方法
- 2.5 软件测试的分类与分级

软件测试的分类与分级

软件测试的分类方法

- 按照与软件内部结构的相关性分类
  - 白盒测试；黑盒测试；灰盒测试。
- 按照是否执行程序代码分类
  - 静态测试；动态测试。
- 按照测试执行时是否需要人工干预分类
  - 人工测试；自动化测试。
- 按照软件开发过程的阶段分类
  - 需求分析阶段；设计阶段；编码阶段；测试阶段；安装阶段；维护阶段。
- 按照测试实施的组织者分类
  - 开发方测试 (α 测试)；用户测试 (β 测试)；第三方测试。
- 按照 SDLC 的测试阶段进行分级
  - 单元测试；集成测试；功能/确认测试；系统测试；验收测试。

软件测试的分类与分级

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - IEEE-STD-610.12-1990
    - CI: Configuration Item (CI) is an aggregation (聚集) of hardware, software, or both that is designated for (用于) configuration management and treated as a single entity in the configuration management process.
    - CU: Configuration Unit is the lowest-level entity of a configuration item or component that can be placed into, & retrieved from, a configuration management library system.
    - CSCI: Computer Software Configuration Item (CSCI) is an aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process.

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - IBM
    - A *configuration item* (CI) is any service component, infrastructure element, or other item that needs to be managed in order to ensure the successful delivery of services.
    - Each CI has several characteristics :
      - A classification, or type, which indicates what kind of item it is.
      - Attributes, which vary by classification and describe the characteristics of the individual CI.
      - A status value, which represents the CI's state in the lifecycle used for CIs of this classification.
      - Relationships, which indicate how the CI is related to other CIs.
      - An owner, the person who is responsible for the CI.
    - CIs vary in complexity, size, and type. They can range from an entire service, which may consist of hardware, software, and documentation, to a single program module or a minor hardware component. The lowest-level CI is usually the smallest unit that will be changed independently of other components.

软件测试的分类与分级

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - From Project Management Viewpoint
    - A Computer Software Configuration Item (CSCI) is a major software component of a system that is designated by the Buyer for configuration management to ensure integrity of the delivered product. It may exist at any level of the hierarchy, where interchangeability is required. Each CSCI is to have (as appropriate) individual design reviews, individual qualification certification, individual acceptance reviews and separate user manuals.

软件测试的分类与分级

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - 软件配置 (Software Configuration, SC)
    - 软件配置是软件开发过程中产生的所有信息，包括由各种形式(机器可读或人工可读) 和各种版本的文档、报告、程序及其数据所组成的集合。软件配置描述了软件的技术状态。
    - 软件配置的内容：
      - 源代码、目标代码以及数据结构；
      - 需求文档、技术文档、管理文档等；
      - 软件测试过程中所产生的工作成果，例如测试计划、测试用例、自动化测试执行脚本、测试数据、测试报告等；
      - 软件开发过程中使用的环境 (如操作系统、数据库系统、各类支撑软件等)。
    - 软件配置的内容可能随着软件开发和维护工作的进展而不断变化，应当妥善管理。

软件测试的分类与分级

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - 软件配置项 (Software Configuration Item, SCI or CSCI)
    - 软件配置的每一个项目称为一个软件配置项。
    - 软件配置项是软件配置管理的基本单位。
  - 软件配置管理 (Software Configuration Management, SCM)
    - SCM 是标识和确定系统中软件配置项的管理过程，它通过协调在同一软件项目中不同人员的软件工作产品来实现：
      - 识别、定义软件配置项并指定基线；
      - 控制软件配置项的投放和变更，记录并报告软件配置项的状态和变更请求；
      - 保证软件配置项的完整性、一致性和正确性；
      - 履行必要的审批手续，控制软件的存储、处理和交付。

软件测试的分类与分级

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - 基线 (Baseline)
    - Baseline: In configuration management, a "baseline" is an agreed description of the attributes of a product, at a point in time, which serves as a basis for defining change. An "alteration" is a movement from this baseline state to a next state.
      - Generally, a baseline may be a single work product, or set of work products that can be used as a logical basis for comparison. A baseline may also be established as the basis for subsequent select activities when the work products meet certain criteria. Such activities may be attributed with formal approval.
    - Baseline Identification: The identification of significant changes from the baseline state.
    - Ref. to *Capability Maturity Model Integration* (CMMI, by CMU-SEI)

软件测试的分类与分级

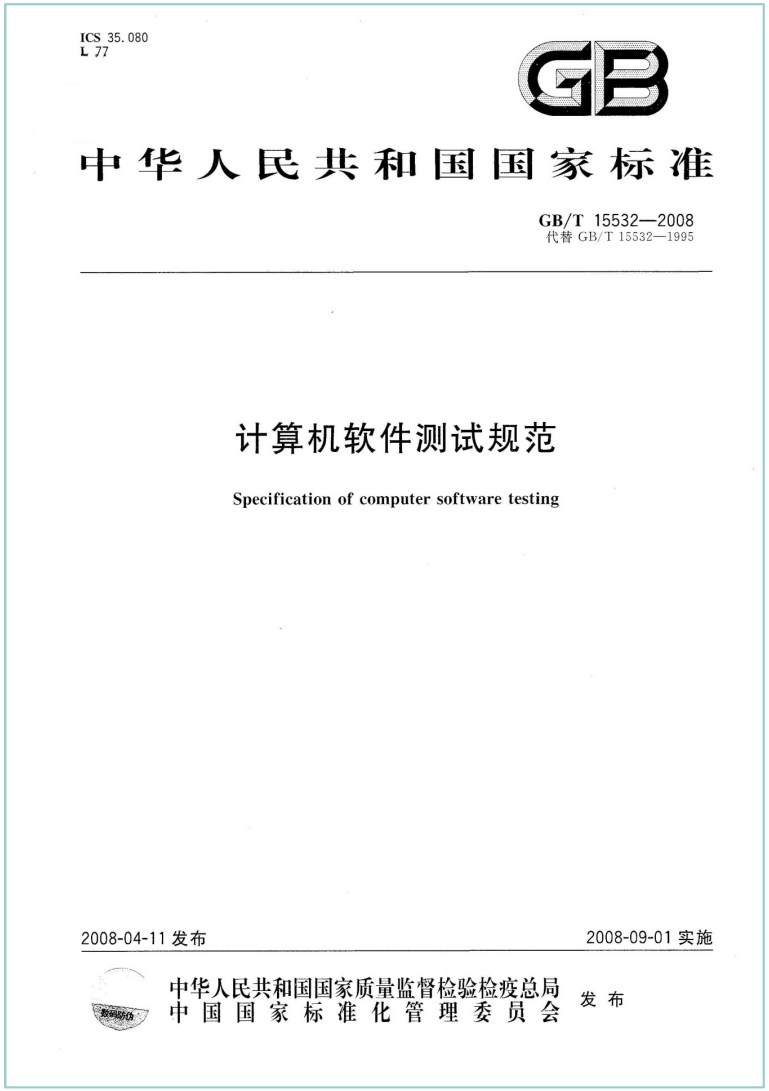
- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - 基线 (Baseline)
    - 软件配置项概念中的基线指的是软件技术状态基线。
    - 基线描述了受软件配置管理控制的某个软件开发阶段的结束点的软件成分的技术状态。该状态已经通过正式审核，是下一步软件开发的基础。
    - 任何一个软件配置项，一旦形成文档并审议通过即成为基线。
    - 对已经成为基线的软件配置项进行修改时，必须按照特殊的、正式的过程进行评价和确认。
    - 对未成为基线的软件配置项，可以进行非正式的修改。

软件测试的分类与分级

- 计算机软件配置项 CSCI (Computer Software Configuration Item)
  - 软件配置项测试
    - 测试目的
      - 检验软件配置项与软件需求规格说明的一致性。
    - 测试人员
      - 软件的供应方组织，由独立于软件开发团队的人员主持，软件开发人员配合实施。
    - 技术依据
      - 软件需求规格说明 (含接口需求规格说明)。
    - 测试内容
      - 依据软件质量模型中包括功能性、可靠性、可用性、效率、维护性和可移植性中的质量特性及子特性中的可测试项，根据软件需求的要求选定进行。
      - 测试内容可以根据被测对象的实际情况进行剪裁。

软件测试的分类与分级

- GB/T 15532-2008《计算机软件测试规范》



ICS 35.080
L 77

中华人民共和国国家标准

GB/T 15532—2008
代替 GB/T 15532—1995

计算机软件测试规范

Specification of computer software testing

2008-04-11 发布　　2008-09-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

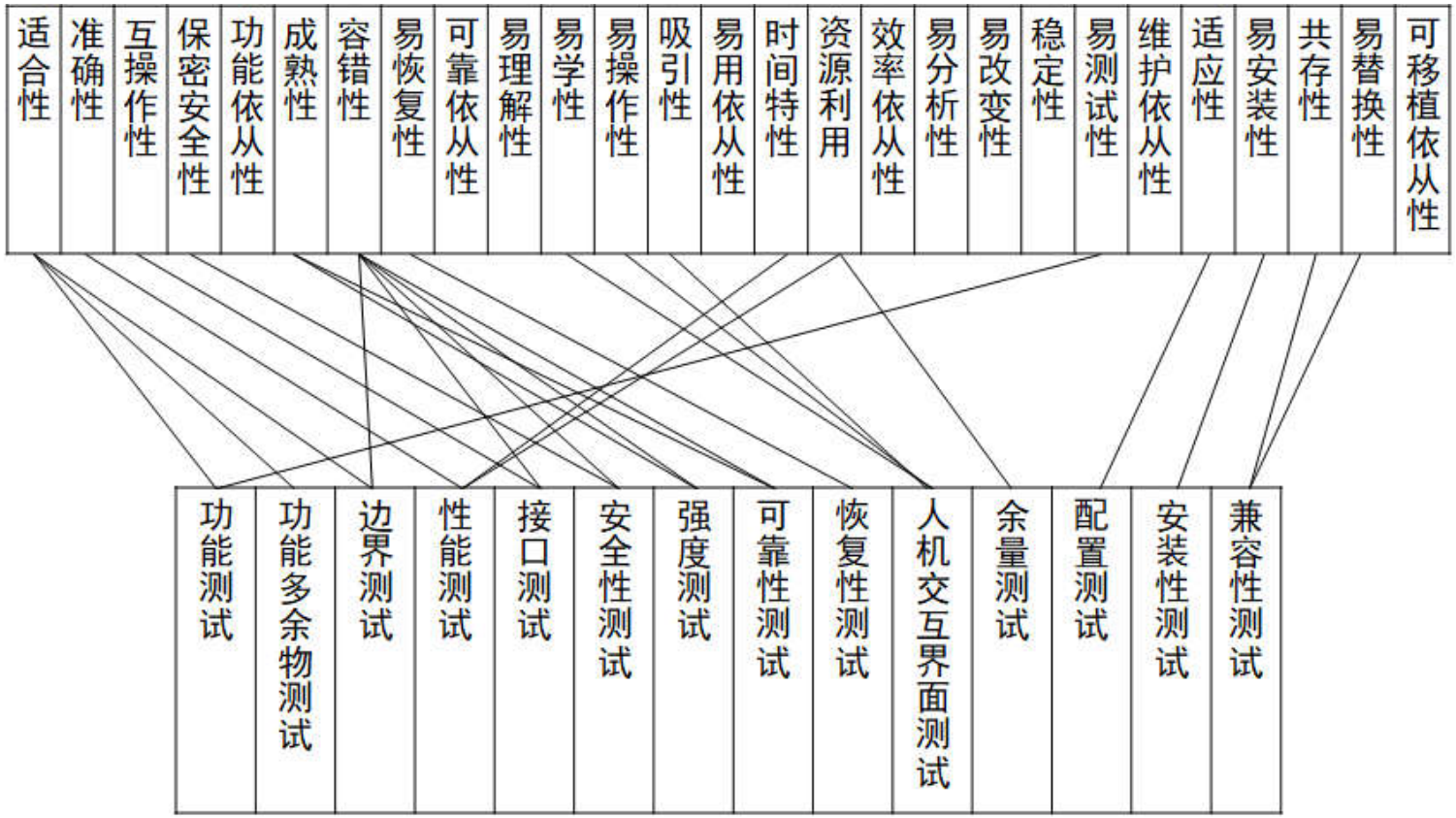
软件测试的分类与分级

- GB/T 15532-2008《计算机软件测试规范》
  - GB/T 15532-2008 规定的6个测试类别：
    - 单元测试
    - 集成测试
    - 软件配置项测试 (也称软件合格性测试、功能测试或确认测试)
    - 系统测试
    - 验收测试
    - 回归测试
  - 可以根据软件的规模、类型、完整性级别选择执行测试类别。

软件测试的分类与分级

- GB/T 15532-2008《计算机软件测试规范》
  - 对 GB/T 15532-2008 规定的测试类别的描述结构包含：
    - 测试对象和目的
    - 测试的组织和管理
    - 测试技术要求
    - 测试内容
    - 测试环境
    - 测试方法
    - 测试过程和文档要求

- GB/T 15532-2008《计算机软件测试规范》
  - GB/T 15532-2008 规定的测试内容主要依据 GB/T 16260.1 规定的质量特性进行，有别于传统的测试内容。它们之间的对应关系如下：



基于 CSCI 的软件测试

基于 CSCI 的传统测试种类 (内容)

- 功能测试
- 余量测试
- 性能测试
- 外部接口和人机交互界面测试
- 强度测试
- 可靠性测试
- 安全性测试
- 恢复性测试
- 边界测试
- 功能多余物测试
- 安装性测试
- 本地化语言测试
- 应用基准测试
- 其它测试

在实际测试中并非在每个测试类别上都必须完成上述所有的测试内容，而应根据被测软件的复杂性、关键等级和当前的测试类别选定执行的测试。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类
    - 功能测试
      - 功能测试主要对软件需求规格说明定义的功能需求进行测试，确认被测软件与功能需求的一致性，发现不一致性。
        - 确定每一个软件功能必须被一个测试用例或一个被认可的异常所覆盖。
      - 使用基本数据类型和数据值进行测试。
        - 使用一系列合理的数据类型和数据值运行，测试超负荷、饱和及其它“最坏情况”的结果。
      - 使用假想的数据类型和数据值运行被测程序，测试其排斥不规则输入的能力。
      - 确定软件的每个功能的合法边界值和非法边界值都必须有测试用例专门测试。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 余量测试
      - 测试程序的全部存储量、输入/输出通道及处理时间的余量满足需求规格说明的要求。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 性能测试
      - 性能测试主要对软件需求规格说明定义的性能需求进行测试，说明在一定工作负荷和配置条件下，被测软件的响应时间及处理速度等特性，发现与性能需求的不一致性。
        - 测试程序在获得定量结果时程序计算的精确性
        - 测试程序在有速度要求时完成功能的时间
        - 测试程序完成功能所能处理的数据量
        - 测试程序各部分的协调性，如高速、低速操作的协调
        - 测试软/硬件因素是否限制了程序的性能
        - 测试程序的负载潜力
        - 测试程序运行占用空间

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 外部接口和人机交互界面测试
      - 外部接口和人机交互界面测试主要对平台各个服务域提供的应用编程接口、应用程序接口、外部环境接口以及人机交互界面的符合性和可用性进行测试。
      - 在进行外部接口和人机交互界面测试时，要求：
        - 测试所有外部接口，检查接口信息的格式及内容；
        - 测试所有人机交互界面提供的操作和显示界面,并以非常规操作、误操作、快速操作来检查界面的可靠性，以最终用户为背景检验界面显示的清晰性；
        - 如果有用户手册或操作手册，应对照手册逐条进行操作和观察。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 强度测试
      - 强度测试在预先规定的一个时间内，在软件设计能力的极限状态，进而超出此极限状态下，运行软件的所有功能。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 可靠性测试
      - 可靠性测试是在有实际使用代表性的环境中，为进行软件可靠性估计而对其进行的功能测试。
        - 可靠性测试必须保证对软件的各种使用功能的覆盖。
      - 可靠性测试必须按软件使用概率分布随机选择测试实例。
      - 可靠性测试必须保证程序的输入覆盖，包括：
        - 输入域覆盖：覆盖重要的输入变量值，并且所有被测输入值域的概率之和必须大于软件可靠度要求。
        - 相关输入变量可能性组合的覆盖：确保相关输入变量的相互影响不会导致软件失效。
        - 设计输入空间与实际输入空间之间区域的覆盖：即不合法输入域的覆盖。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 可靠性测试 (续)
      - 被测软件的测试环境 (包括硬件配置和软件支撑环境) 应和预期的实际使用环境尽可能一致。
      - 对于可能导致软件运行方式改变的一些边界条件 (如堆栈溢出) 和环境条件 (如系统加电、掉电、电磁干扰等) 必须进行针对性测试。
      - 测试时应记录测试结果、运行时间和判断结果。如果软件失效，还应记录下失效现象和时间。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 安全性测试
      - 安全性测试主要对平台软件配置项的安全性进行测试。
        - 安全性测试用于说明被测软件系统的安全机制是否存在，是否起到了应有的作用，是否达到了规定的安全级别等。
      - 安全性测试内容包括系统安全评估和系统入侵测试两个部分：
        - 系统安全评估主要涉及标识与鉴别、访问控制、审计、特权管理、可信通路、隐通道等。
        - 系统入侵测试主要涉及系统脆弱性分析、系统安全漏洞检测等。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 安全性测试 (续)
      - 安全性测试的基本要求
        - 进行软件安全性分析，并且在软件需求说明中明确每一个危险状态及导致危险的可能原因，在测试中全面检验软件在这些危险状态下的反应。
        - 单独测试安全性关键部件，确认满足安全性要求。
        - 对软件设计中用于提高安全性的结构、算法、容错、冗余、中断处理等方案进行针对性测试。
        - 测试应尽可能在符合实际使用的条件下进行。
        - 异常测试：除在正常条件下测试外，应在异常条件下测试软件，以表明不会因可能的单个或多个输入错误而导致系统处于不安全状态。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 安全性测试 (续)
      - 异常测试的内容
        - 硬件及软件输入故障模式测试
        - 边界、界外及边界接合部的测试
        - “0”、穿越“0”以及从两个方向趋近于“0”的输入值测试
        - 在最坏情况配置下的最小和最大输入数据率测试 (以确定系统的固有能力及对这些环境的反应)
        - 接口测试必须包括在安全性关键操作中的操作员错误测试 (以验证安全系统对这些错误的响应)
        - 测试双工切换、多机替换的正确性和连续性
        - 测试防止非法进入系统的能力
        - 测试保护系统数据完整性的能力

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 恢复性测试
      - 对有恢复或重置 (RESET) 功能的软件，必须验证恢复或重置功能，对每一类导致恢复或重置的情况进行测试。
    - 边界测试
      - 测试程序在输入域 (或输出域)、数据结构、状态转换、过程参数、功能界限等的边界或端点情况下运行状态。
    - 功能多余物测试
      - 验证被测软件实现的功能以及功能边界都是在软件需求规格说明中明确声明的
      - 所有的程序输出都应有意义并在软件需求规格中指明

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 安装性测试
      - 安装性测试对平台软件配置项的可安装性/可卸载性进行测试。
      - 安装性测试通过安装/卸载程序或按照安装/卸载规程进行软件配置项的安装/卸载，发现安装/卸载过程的错误，验证软件配置项的可安装性/可卸载性。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 中文能力测试
      - 中文能力测试属于本地化语言测试的内容，主要对平台软件配置项的中文支持能力进行测试，验证软件配置项是否能够全面、正确地支持和处理中文。
      - 中文能力测试的基本要求：
        - 测试中英文转换后的正文长度变化是否对软件有影响
        - 测试软件是否能够完全处理中西文字符集
        - 测试对中文是否存在正文过滤现象
        - 测试操作系统语言是否支持中文
        - 测试中文排序规则是否正确
        - 测试大小写转换功能对中文是否有影响

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 应用基准测试
      - 应用基准测试 (配置测试) 主要用于测试平台软件配置项的综合性能。
      - 应用基准测试构造一组能够反映某个领域应用特点的典型应用程序，即面向特定应用领域的应用基准程序，在平台上运行该组程序以测试平台软件配置项的综合性能。
        - 综合性能包括软件配置项的适用性、准确性、成熟性、稳定性、时间行为、资源行为、适应性、易学性、易操作性等。
      - 应用基准测试主要用来验证平台软件系统对典型应用的综合支持能力，因此，应用基准程序必须是能够反映应用领域特点的典型、甚至标准的应用程序。

- 基于 CSCI 的软件测试
  - 基于 CSCI 的传统测试种类 (续)
    - 其他测试
      - 负载测试
        - 测试软件在重负荷下的表现，例如测试一个 Web 站点在大量的负荷下，何时系统的响应会退化或失败。
      - 压力测试
        - 评估在超越最大负载的情况下系统将如何运行，目标是发现在高负载的条件下应用程序的缺陷。
      - 疲劳测试
        - 采用系统稳定运行情况下能够支持的最大并发用户数，持续执行一段时间业务，通过综合分析交易执行指标和资源监控指标来确定系统处理最大工作量强度性能。

软件测试的分类与分级

- 软件测试分级的概念
  - 软件测试分级的依据
    - 按照软件测试的要求、目的、关注点、被测对象、工作产品及测试人员的不同，可以进行相应的软件测试级别的划分。
  - 主要的软件测试分级
    - 按照软件生命周期测试阶段的分级
    - 按照软件错误的分级 (软件错误对软件测试通过的影响)
    - 完整性测试的分级
    - 软件测试用例的分级 (参见：chap.6.1 测试用例设计)
  - 软件测试分级的作用
    - 软件测试分级可以有效控制软件的复杂性，强化测试的针对性，提高测试管理的科学性，最终确保软件测试的质量。

软件测试的分类与分级

- 按照软件生命周期测试阶段的分级
  - 分级层次 (GB/T 15532-2008)
    - 单元测试、集成测试、配置项测试 (也称软件合格性测试、功能测试或确认测试)、系统测试、验收测试和回归测试。
    - 回归测试可出现在上述每个测试级别中，并贯穿整个软件生命周期。
  - 在不同的测试级别上都需要明确：
    - 测试对象和目的
    - 测试用例设计的基础
      - 测试用例参考的软件产品或测试需求
    - 发现的典型缺陷和失效
    - 测试工具的需求和支持
    - 不同的测试技术和方法

软件测试的分类与分级

- 按照软件生命周期测试阶段的分级
  - 单元测试/组件测试/模块测试
    - 针对单个软件单元的测试都可以称为单元测试。
    - 单元测试包括功能测试和特定的非功能测试。
      - 如资源行为测试 (内存泄露)、健壮性测试、基于结构的测试 (如分支覆盖) 等。
    - 单元测试的设计输入主要是单元详细规格说明、软件设计规格说明或数据模型等。
    - 在编写代码之前就开始准备测试和自动化测试用例是单元测试常用的一种方法。
      - 该方法被称之为测试驱动或测试驱动开发 (TDD)。

- 按照软件生命周期测试阶段的分级
  - 单元测试/组件测试/模块测试 (续)
    - 单元测试的任务包括
      - 模块局部数据结构测试，模块接口参数边界值测试，模块中所有独立执行路径测试，以及模块的各条错误处理路径测试等。
    - 测试环境：程序代码编写完成并通过评审和编译检查后，便可开始单元测试。
      - 单元测试与开发工程密切相关，通常由开发人员自己执行单元测试。
      - 单元测试主要采用白盒测试方法结合黑盒测试方法，通常从程序内部结构出发设计测试用例。

- 按照软件生命周期测试阶段的分级
  - 单元测试/组件测试/模块测试 (续)
    - 单元测试中需要考虑各方面的因素有：
      - 检查单元模块接口参数，是单元测试的基础；
      - 检查局部数据结构，用来保证临时存储在模块内的数据在程序执行过程中的完整、正确；
      - 在模块中应对每一条独立执行路径进行测试，单元测试的基本任务是保证模块中每条路径至少执行一次；
        - 比较、判断与控制流常常紧密相关；
      - 好的软件设计应能预见各种出错条件，并预设各种出错处理路径，出错处理路径同样需要认真测试。

软件测试的分类与分级

- 按照软件生命周期测试阶段的分级
  - 集成测试
    - 集成测试也叫组装测试、联合测试。
      - 集成测试旨在暴露接口以及集成组件之间或系统之间交互时存在的缺陷。
      - 集成测试是单元测试的逻辑扩展，测试关注点是组件之间的接口，以及组件与系统其它部分 (如操作系统、数据库系统、硬件系统) 的相互作用。

- 按照软件生命周期测试阶段的分级
  - 集成测试 (续)
    - 根据测试对象规模的不同，集成测试可分为组件集成测试和系统集成测试两类：
      - 组件集成测试针对不同的软件组件之间的相互作用进行测试，一般在单元测试之后进行。
      - 系统集成测试针对不同系统之间的相互作用进行测试，一般在系统测试之后进行。
        - 系统集成测试的范围越大，对缺陷的定位就越困难，系统的风险就越大。
      - 为了降低在软件开发生命周期后期发现严重缺陷而产生的风险，集成程度应该逐步增加。

- 按照软件生命周期测试阶段的分级
  - 集成测试 (续)
    - 集成测试关注模块之间的接口，而不是模块本身的功能。
      - 黑盒测试和白盒测试的方法都可以应用在集成测试上。
    - 集成测试的对象是已经经过单元测试的软件组件，主要内容是功能性、可靠性、易用性、效率、可维护性和可移植性。
      - 集成测试的依据是软件的概要设计规格说明。
    - 集成测试采用的测试策略是非渐增式集成测试模式和渐增式集成测试模式。
      - 包括自底向上集成、自顶向下集成、核心子系统先行集成以及随意集成等。

- 按照软件生命周期测试阶段的分级
  - 配置项测试
    - 必要时在高层控制流图中作结构覆盖测试；
    - 逐项测试软件需求规格说明所规定的配置项的功能、性能等特性；配置项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；
    - 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；
    - 测试配置项的输入输出及其格式；
    - 测试人机交互界面提供的操作和现实界面，包括用非常规操作、无操作、快捷操作、快速操作测试界面的可靠性；
    - 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，配置项的功能和性能；
    - 测试配置项的安全性和数据的安全保密性；

- 按照软件生命周期测试阶段的分级
  - 配置项测试 (续)
    - 测试配置项的所有外部 I/O 接口 (包括和硬件之间的接口)；
    - 测试配置项的全部存储、输入/输出通道的吞吐能力和处理时间的余量；
    - 按照软件需求规格的要求，对配置项的功能、性能进行强度测试；
    - 测试设计中用于提高配置项的安全性和可靠性方案，如结构、算法、容错、冗余、中断处理等；
    - 对安全性关键的配置项进行安全性分析，明确每一个危险状态和导致危险的可能原因，并对此进行针对性测试；
    - 对有恢复或重置功能需求的配置项，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；
    - 对不同的实际问题外加相应的专门测试。
      - 参见本节“基于 CSCI 的软件测试”

- 按照软件生命周期测试阶段的分级
  - 系统测试
    - 系统测试将已经集成好的软件系统作为计算机系统的一部分，与计算机系统的硬件、支持软件、数据和人员等系统元素结合起来，在实际运行环境下对计算机系统进行一系列严格有效的测试。
    - 系统测试对测试环境的要求：集成测试完成后，系统已经全部完成组合，在尽可能和目标运行环境一致的情况下进行测试。
    - 系统测试的目的是确认整个系统是否满足了系统需求规格说明中的功能需求和非功能需求，以及满足的程度。
    - 常见的系统测试包括：压力测试、容量测试、性能测试、安全测试、容错测试等。

软件测试的分类与分级

- 按照软件生命周期测试阶段的分级
  - 验收测试
    - 验收测试通常由使用系统的用户为主导实施。
      - 验收测试的目的是确保系统功能、系统的某部分或特定的系统非功能特征满足验收准则。
      - 发现缺陷不是验收测试的主要目标。
    - 验收测试的主要测试类型有
      - 根据合同的验收测试
      - 用户验收测试
      - 运行 (验收) 测试
      - 现场测试

- 按照软件生命周期测试阶段的分级
  - 维护测试
    - 软件被用户接受并运行一段时间后，需要做某些修正、改变或扩展。在这种情况下进行的测试称为维护测试。
    - 维护测试是在一个运行的系统上进行的，属于回归测试类型。

- 软件测试中的软件错误分级
  - 目的
    - 对软件错误进行级别定义或分级的目的，在于为软件测试工作提供科学的指导，提高软件测试的目的性，从而确保软件测试的质量。
    - 不同的行业和企业、不同的应用领域以及不同的错误类型有不同的错误分级方法。
    - 软件错误分级涉及到两个方面：错误分类及错误级别划分。

- 软件测试中的软件错误分级
  - 软件错误分类
    - 按软件生命周期各阶段分类
      - 用户需求错误、产品需求错误、设计错误、编码错误、数据错误、发行错误等。
    - 按软件使用特性分类
      - 功能错误、性能错误、界面错误、流程错误、数据错误、提示错误、常识错误以及其他错误。
    - 按《GB/T 32422-2015 软件工程 软件异常分类指南》分类

- 软件测试中的软件错误分级
  - 软件错误级别划分
    - 第1级错误：严重缺陷
      - 妨碍由基线要求所规定的运行或任务的主要功能的完成；
      - 妨碍操作员完成运行或任务的主要功能；
      - 危及相关人员安全。
    - 第2级错误：较严重缺陷
      - 给由基线要求所规定的运行或任务的主要功能的完成造成不利的影响，以致降低系统效能，而且没有变通的解决办法；
      - 给操作员完成由基线要求所规定的运行或任务的主要功能造成不利的影响，以致降低系统效能，且没有变通的解决办法。

- 软件测试中的软件错误分级
  - 软件错误级别划分 (续)
    - 第3级错误：一般性缺陷
      - 给由基线要求所规定的运行或任务的主要功能的完成造成不利的影响，以致降低系统效能，但已知有变通的解决办法；
      - 给操作员完成由基线要求所规定的运行或任务的主要功能造成不利的影响，以致降低系统效能，但已知有变通的解决办法。
    - 第4级错误：较小缺陷
      - 给操作员带来不方便或麻烦，但不影响其完成由基线要求所规定的运行或任务的主要功能。
    - 第5级错误：其他缺陷
      - 所有的其他错误。

- 软件测试中的软件错误分级的实例化
  - 第1级：严重缺陷。软件错误使得应用系统崩溃或系统资源严重不足。
    - 系统停机 (含软件、硬件) 或非法退出，且无法通过重启恢复；
    - 系统死循环；
    - 数据库发生死锁或程序原因导致数据库断开连接；
    - 系统关键性能不达标；
    - 数据通讯错误或接口不通；
    - 错误操作导致程序中断。

- 软件测试中的软件错误分级的实例化
  - 第2级：较严重缺陷。软件错误可能导致如：
    - 重要事务无法正常处理、功能不符合用户需求；
    - 重要计算错误；
    - 业务流程错误或不完整；
    - 使用某事务处理导致业务数据紊乱或丢失；
    - 业务数据保存不完整或无法保存到数据库；
    - 周边接口出现故障 (需考虑接口时效/数量等综合情况)；
    - 服务程序频繁需要重启 (每天2次或以上)；
    - 批处理报错中断导致业务无法正常开展；
    - 前端未合理控制并发或连续点击动作，导致后台服务无法及时响应；
    - 在产品声明支持的不同平台下，出现部分重要事务无法处理或处理错误。

- 软件测试中的软件错误分级的实例化
  - 第3级：一般性缺陷。软件错误可能导致如：
    - 部分事务处理存在问题，虽不影响业务继续开展，但造成使用障碍；
    - 初始化未满足客户要求或初始化错误；
    - 功能点能实现，但结果错误；
    - 数据长度不一致，无数据有效性检查或检查不合理，数据来源不正确；
    - 显示/打印的内容或格式错误；
    - 删除操作不给提示；
    - 个别事务处理系统反应时间超出正常合理时间范围；
    - 日志记录信息不正确或应记录而未记录；
    - 在产品声明支持的不同平台下，出现部分一般事务无法处理或处理错误。

- 软件测试中的软件错误分级的实例化
  - 第4级：较小缺陷。软件错误导致一些操作不便如：
    - 系统某些查询、打印等实时性要求不高的辅助功能无法正常使用；
    - 界面错误；
    - 菜单布局错误或不合理；
    - 焦点控制不合理或不全面；
    - 光标、滚动条定位错误；
    - 辅助说明描述不准确或不清楚；
    - 提示窗口描述不准确或不清楚；
    - 日志信息不够完整或不清晰，影响问题诊断或分析。

- 软件测试中的软件错误分级的实例化
  - 第5级：其他缺陷。软件错误导致系统辅助功能问题如：
    - 缺少产品使用、帮助文档、系统安装或配置方面需要信息；
    - 联机帮助、脱机手册与实际系统不匹配；
    - 系统版本说明不正确；
    - 长时间操作未给用户进度提示；
    - 提示说明未采用行业规范语言；
    - 显示格式不规范；
    - 界面不整齐；
    - 软件界面、菜单位置、工具条位置、相应提示不美观，但不影响使用。

Lecture 11. Classification of Software Testing

End of Lecture