



广东金融学院课程教案

课程名称：软件测试课程设计

课程代码：154442008

开课单位：计算机学院

制定人：周宇文

审核人：周瑞红

制定时间：2026-09-13

广东金融学院教务处 制

一、课程简介

课程类别：专业选修

授课对象：本科 计算机科学与技术、软件工程、数据科学与大数据技术

学时与学分：32 学时，2 学分

使用教材：朱少民. 软件测试方法和技术（第 4 版）[M]. 北京：清华大学出版社，2022.

参考教材：朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.；朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.；朱少民. 软件测试知识地图[Z]. 2022.；陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.；Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.；Kaner C, Bach J, Pettichord B. Lessons Learned in Software Testing[M]. New York: John Wiley & Sons, 2001.；Crispin L, Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams[M]. Boston: Addison-Wesley, 2008.；Meszaros G. xUnit Test Patterns: Refactoring Test Code[M]. Boston: Addison-Wesley, 2007.

二、教学目的与教学要求：

性质：本课程是软件工程与软件测试交叉的实践性课程，面向软件测试岗位能力培养，是计算机类专业的专业课程。

目的：学生通过本课程的学习，掌握软件测试的基本理论、方法、技术与常用工具，熟悉软件测试的流程与规范；能够针对具体软件系统开展测试需求分析、测试计划制定、测试用例设计与执行、缺陷跟踪与测试报告撰写；培养规范化、工程化的测试实践能力与团队协作能力，并在课程中融入工程伦理与职业道德教育，培养学生的质量意识、诚信意识团队精神、社会责任感和科技报国的情怀。

任务：完成软件测试基本理论与方法的学习与实验；以一个具体软件系统为对象完成课程设计项目，包括测试需求分析、测试计划、测试用例设计、测试执行、缺陷报告与测试总结，并进行项目答辩。

第 1 次课 2 学时

本次课教学重点：

软件缺陷的代价与质量成本观

软件测试的必要性（真实事故案例）

软件测试的定义与学科形成

本次课教学难点：

区分测试、调试、质量保证三者的边界

理解测试左移与成本非线性增长

本次课教学内容：

第 1 章 课程导学与引论：为什么必须做软件测试

学习目标：理解软件缺陷与质量成本的概念；理解软件测试的必要性与价值；掌握软件测试的基本定义；了解本课程的考核方式与课程设计任务。

一、软件测试的必要性：从真实事故说起

• 《狮子王》安装失败、Intel 奔腾浮点除法错误、火星探测器坠毁、波音星际客机、骑士资本 4.6 亿美元损失、AWS S3 四小时宕机、Therac-25 放疗仪致死——缺陷代价可能远超开发成本。

- 软件缺陷客观存在：需求理解偏差、设计疏漏、编码错误、集成不当与运行环境差异都会引入缺陷。

- 测试是产品发布前的质检关卡，缺少把关的软件等于把风险直接交给用户。

- 国产游戏《血狮》（1997）宣传远超前于被验证的产品：大量用户无法安装运行、兼容性差、功能与宣传不符，成为反面教材。

二、为什么要进行软件测试

- 测试的直接目的是保证软件质量：软件总存在缺陷，只有通过测试才能发现缺陷，只有发现缺陷才可能将其清除。

- 缺陷发现得越迟，修复成本越高且呈非线性增长；测试左移的本质是尽早发现、降低总成本。

- 缺陷带来的损失包括直接经济损失、人身安全风险、企业信誉受损与法律合规责任。

- 测试不仅发现问题，也提供质量信息，支撑发布决策与风险评估。

三、什么是软件测试

- 定义：在指定条件下运行或评审系统与组件，观察或记录结果，对照预期作出评价。

- 测试包含静态测试（评审、走查、静态分析）与动态测试（执行程序观察结果）。

- 狭义测试把测试等同于程序测试；广义测试认为软件不只是可执行程序，文档、数据与配置同样需要验证。

- 正向思维（Bill Hetzel）：测试是评价程序或系统特性、确认是否达到预期结果的活动。

- 反向思维（Glenford J. Myers）：测试是为了发现错误而执行程序的过程，测试的目的是发现缺陷而非证明正确。

四、软件测试学科的形成与发展

- 从“调试即测试”到独立学科：测试由开发者的附属活动发展为独立的工程实践与职业岗位。

- 以缺陷预防为导向的现代测试观：测试贯穿需求、设计、编码、集成、发布与运维全生命周期。

- 质量内建思想：把质量责任嵌入研发流程，而不是依赖末端把关。

五、测试与质量保证（SQA）

- SQA 面向过程，通过规范、评审、度量与过程改进保证质量体系的建立与执行。

- 测试面向产品，通过验证与确认获取产品质量信息。

- 两者互补：没有过程保证的测试只能反复救火，没有测试的 SQA 缺乏事实依据。

六、测试与开发的关系

- 开发与测试是并行协作关系：测试在需求与设计阶段即介入（测试左移），而非开发完成后才开始。

- 测试驱动开发（TDD）主张先写测试再写实现：以测试明确需求、驱动设计、形成回归保护网。

- 在敏捷团队中，开发与测试逐渐融合，提倡整个团队对质量负责。

七、测试岗位与能力要求

- 岗位方向：功能测试、自动化测试、性能测试、安全测试、测试开发与质量工程师。

- 能力结构：测试理论与方法、编程与工具、业务理解、分析与沟通、工程规范意识。

- 从就业市场看，测试岗位需求稳定，自动化与测试开发是主要增长方向。

八、课程导学与课程设计任务

- 课程定位：专业选修、32 学时 / 2 学分，第 1-17 周含实验，期末以课程设计作品与答辩考核。

- 主线安排：原理方法 → 测试技术 → 项目实践，教材为《本课程教材（第 4 版）》。

• 课程设计任务：以一个具体软件系统为对象，完成测试需求分析、测试计划、用例设计、执行、缺陷报告与总结答辩。

• 考核构成：考勤与课堂表现 10% + 实验与阶段任务 40% + 课程设计（计划与用例 15% + 执行与缺陷 15% + 报告与答辩 20%）。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：多媒体讲授+案例分析（支撑课程目标 1）

案例讨论：结合《血狮》案例讨论：如果当年有规范的测试流程，哪些问题可能被提前发现？

- 1、采用多媒体讲授，配合真实事故案例与现场提问。
- 2、先让学生猜测事故损失量级，再给出真实数据，形成认知冲击。
- 3、通过宣传与现实的对比，引导学生建立发布前必须验证的意识。
- 4、最后给出本课程主线：原理方法→测试技术→项目实践。

课程思政设计——思政维度：家国情怀·职业道德·质量责任

融入载体：国外重大事故（Therac-25 放疗仪致死、波音星际客机、骑士资本 4.6 亿美元、AWS S3 宕机）与国产游戏《血狮》的对比案例。

课堂实施：先让学生估算事故损失量级再给真实数据，讨论“如果没有测试把关，交付给用户的是什么”；引到测试人员的职业底线。

评价观测点：能说出测试与公共利益的关系；课堂发言中体现质量责任意识。

作业布置：

- 1、整理 5 个真实软件缺陷事故，标注缺陷类型（兼容/精度/集成/日期/数据等）。
- 2、用自己的话写出软件测试的定义，并说明它与调试、质量保证的区别。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、安装并熟悉课程实验环境（浏览器、截图工具、待测小程序）。
- 2、完成第一次测试体验：找出给定小程序中至少 2 个缺陷并截图记录。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 2 次课 2 学时

本次课教学重点：

测试分类与测试级别
静态测试与动态测试
黑盒测试与白盒测试

本次课教学难点：

不同测试级别的目标与执行主体
黑盒与白盒方法的选择依据

本次课教学内容：

第2章 软件测试的基本概念

学习目标：掌握软件测试的分类与测试级别；理解静态测试与动态测试的区别；理解黑盒白盒与灰盒测试；了解软件测试工作范畴与岗位分工。

一、软件缺陷与软件质量

- 软件缺陷：系统或组件中存在的、导致其无法满足预期需求或产生不正确结果的瑕疵。

- 缺陷来源贯穿全生命周期：需求二义与遗漏、设计缺陷、编码错误、接口不匹配、环境与数据问题。

- 软件质量内涵包括功能性、可靠性、易用性、效率、可维护性与可移植性等特性（参照 ISO/IEC 25010）。

- 质量成本分为预防成本、评价成本与失效成本（内部/外部），缺陷越早发现总成本越低。

二、软件测试的分类与测试级别

- 按是否运行程序：静态测试与动态测试。

- 按对内部结构的了解程度：黑盒测试、白盒测试、灰盒测试。

- 按测试目的：功能测试、性能测试、兼容性测试、安全性测试、可靠性测试、易用性测试等。

- 测试级别：单元测试、集成测试、系统测试、验收测试，各阶段对象、目标与执行主体不同，构成完整测试链。

三、静态测试与动态测试

- 静态测试不运行程序：文档评审、代码走查、代码审查、静态分析工具检查。

- 静态测试可在编码前发现问题，成本最低；对文档与规范质量依赖较大。

- 动态测试通过执行程序并观察结果，验证功能与非功能需求是否满足。

- 两者互补：静态测试覆盖“无法执行”的问题，动态测试验证运行时的真实行为。

四、主动测试与被动测试

- 主动测试：测试者主动设计并执行用例，按计划施压，发现问题的可控性高。

- 被动测试：在系统真实运行中被动收集信息（日志、监控、用户反馈），用于发现计划外问题。

- 生产环境的监控与灰度发布属于典型的被动测试实践。

五、黑盒、白盒与灰盒测试

- 黑盒测试依据需求与业务规则，从外部输入输出验证功能，不关心内部实现。

- 白盒测试基于代码结构，关注语句、分支、条件与路径覆盖。

- 灰盒测试结合两者：了解部分内部结构（如接口、数据库）以设计更有效的用例。

- 选择依据：需求稳定性、代码可获取性、风险等级与投入成本。

六、软件测试工作范畴与岗位分工

- 工作范畴：测试需求分析、测试设计、测试执行、缺陷管理、测试环境与工具建设、测试度量与报告。

- 分工：测试工程师、测试开发工程师、性能/安全专项测试工程师、测试经理。

- 测试活动贯穿研发全流程，需与产品、开发、运维紧密协作。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：多媒体讲授+课堂练习（支撑课程目标 1、2）

案例讨论：同一段代码缺陷：用黑盒视角与白盒视角分别能发现什么？

1、以同一功能、不同视角的对比讲解测试分类。

2、现场演示一次静态审查与一次动态执行的区别。

3、组织学生对给定功能点分类归属进行快速抢答。

课程思政设计——思政维度：科学精神·质量观·诚信

融入载体：软件缺陷与质量成本曲线、ISO/IEC 25010 质量特性、缺陷分类实践。

课堂实施：用“同一缺陷不同视角”的对比练习训练分类思维；讨论“发现缺陷不报告”对团队与用户的后果。

评价观测点：缺陷分类是否准确；是否如实记录缺陷而不隐去不利结果。

作业布置：

- 1、为“学生选课系统”分别列出适合黑盒与白盒测试的功能点。
- 2、画出单元—集成—系统—验收四级测试的关系图。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、使用等价类与边界值思路，为登录功能设计 10 条用例。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 3 次课 2 学时

本次课教学重点：

错误猜测法与探索式测试

基于缺陷模式的测试（DPBT）

测试方法分类框架

本次课教学难点：

探索式测试的会话组织与记录

把个人经验转化为可复用用例

本次课教学内容：

第 3 章 软件测试方法（一）：基于直觉与经验

学习目标：掌握错误猜测法与探索式测试；理解基于缺陷模式的测试思想；能根据场景选择测试方法。

一、软件测试方法分类框架

- 测试方法可按依据划分为基于直觉与经验、基于输入域、基于组合、基于逻辑、基于模型与形式等类别。

- 方法选择取决于需求形态、风险等级、可获取的模型信息与投入成本，实际项目多为组合使用。

- 任何方法都服务于同一个目标：以可接受的成本发现更多、更严重的缺陷。

- 错误猜测法、探索式测试属于经验驱动，适合需求不完整、时间受限的早期阶段。

二、错误猜测法（Error Guessing）

- 依据测试者对系统易错点的经验判断，直接列出高风险场景并构造用例。

- 常见猜测点：空值与超长输入、边界值、非法格式、重复提交、并发操作、异常中断、权限越权。

- 优点是快速、成本低；缺点是依赖个人经验，覆盖不可度量、难以复用。
- 应把猜测结果沉淀为检查表（Checklist）与缺陷模式，交由团队共享复用。

三、探索式测试（Exploratory Testing）

- 学习、设计、执行三者同步进行，边理解系统边设计并立即执行的测试方式。
- 以会话（Session）为组织单位，采用时间盒管理，配合章程（Charter）明确测试目标与范围。
- 测试记录应包含：测试思路、操作路径、发现的问题与后续待验证点。
- 适合需求快速变化、文档不足或者需要补充脚本化用例盲区的场景。
- 与脚本化测试互补：脚本化保证回归与覆盖，探索式负责发现未知缺陷。

四、基于缺陷模式的测试（DPBT）

- 把历史缺陷抽象为可识别的缺陷模式，如空指针、数组越界、资源未释放、边界处理错误。
- 按模式检索代码或设计中的可疑点，据此设计针对性用例，提高缺陷命中率。
- 缺陷模式可来源于缺陷库、根因分析与行业缺陷分类（如正交缺陷分类 ODC）。
- 该思想同时服务于缺陷预防：在评审与设计阶段即按模式排查。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+探索式测试实践（支撑课程目标 2）

案例讨论：针对“文件上传”功能，用错误猜测法列出 10 个可能的出错点。

- 1、先由学生自由列举易错点，教师归纳为分类框架。
- 2、用真实缺陷案例说明经验为什么有效、也会失效。
- 3、组织一次短时探索式测试，现场记录发现。

课程思政设计——思政维度：科学精神·批判思维·工匠品质

融入载体：错误猜测法、探索式测试会话、真实缺陷模式（空指针、资源未释放、边界越界）。

课堂实施：组织 30 分钟探索式测试会话，要求学生先提出假设再验证，记录“猜错的假设”。

评价观测点：能否基于证据下结论、敢于否定自己的假设；会话记录是否规范完整。

作业布置：

- 1、查找并阅读一个缺陷模式（如空指针、资源未释放），说明如何在测试中触发它。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、以小组为单位开展 30 分钟探索式测试会话，记录测试思路与发现。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 4 次课 2 学时

本次课教学重点：

基于模型的测试（MBT）
形式化测试方法
测试方法的选择与组合

本次课教学难点：

模型抽象与状态迁移覆盖准则
方法成本与收益的权衡

本次课教学内容：

第3章 软件测试方法（二）：基于模型与形式

学习目标：理解基于模型测试的原理与流程；能用状态机模型生成测试用例；了解形式化测试方法及其适用场景。

一、基于输入域的测试方法

- 等价类划分：把输入域划分为若干等价类，每类取代表值，用少量用例覆盖全部有效与无效类。
- 边界值分析：错误最易出现在边界上，取最小值、略大于最小值、正常值、略小于最大值、最大值及其越界点。
- 判定表与因果图：用于处理输入条件之间存在组合与约束的逻辑关系。
- 正交试验与组合覆盖：用两两组合（Pairwise）等方法在组合爆炸与覆盖之间取得平衡。

二、基于逻辑覆盖的白盒方法

- 语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖、路径覆盖，覆盖强度依次递增。
- 基本路径测试：以控制流图计算环路复杂度，导出独立路径集合以确定最少测试用例数。
- 循环测试关注零次、一次、多次、最大次数及越界循环。
- 覆盖率高不等于缺陷少，需与需求覆盖、数据覆盖结合判断测试充分性。

三、基于模型的测试（MBT）

- 从需求或设计抽取状态机、流程图、决策表、序列图等模型，由模型生成测试用例。
- 状态迁移测试需明确状态、事件、迁移与守卫条件，覆盖准则包括全状态、全迁移、全路径等。
- 基于场景的测试以典型业务流为主线，覆盖基本流与各种备选流、异常流。
- MBT 的收益在于需求与用例同步、变更后可快速重生成；代价是建模与工具投入。

四、形式化测试方法

- 以严格数学语义描述系统行为与测试条件，如有限状态机、Petri 网、Z/VDM 规约。
- 适用于安全关键系统（航天、医疗、轨交），可支撑证明式的正确性验证。
- 建模与验证成本高，对人员数学与工具能力要求高。
- 实践中常与常规测试结合，用于核心逻辑而非全系统。

五、测试方法的选择与组合

- 按风险排序：高风险、高复杂度、频繁变更的模块优先采用更严格的方法。
- 黑盒方法主导功能与业务验证，白盒方法补充代码结构与逻辑覆盖。
- 静态方法与动态方法互补，评审与静态分析可在编码前发现问题、成本最低。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+建模练习（支撑课程目标2）

案例讨论：把一个登录—锁定—解锁业务流程画成状态机，并生成覆盖各迁移的用例。

1、以状态机建模为主线，边讲边画。

- 2、学生分组完成模型并互评覆盖是否完整。
- 3、对比不同方法在同一需求上的用例数量与质量。

课程思政设计——思政维度：严谨求实·规范意识·科技报国

融入载体：状态机建模、形式化方法在航天/轨道交通等安全关键系统中的应用。

课堂实施：以国产大飞机、高铁列控系统为例，说明严格建模与验证如何守护公共安全。

评价观测点：模型是否完整、覆盖准则是否清晰；能否说明形式化方法的适用边界。

作业布置：

- 1、为给定的订单状态机设计覆盖全部迁移的测试路径。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、使用状态转换图完成 15 条用例设计，并说明覆盖准则。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 5 次课 2 学时

本次课教学重点：

传统测试过程（V 模型）

敏捷测试与测试左移

测试规范与过程改进

本次课教学难点：

敏捷迭代中的测试节奏安排

准入准出条件的量化

本次课教学内容：

第 4 章 软件测试流程和规范

学习目标：理解传统测试过程与敏捷测试过程；掌握测试准入准出条件；了解测试规范与过程改进。

一、传统的软件测试过程

- V 模型：需求分析、概要设计、详细设计、编码对应单元测试、集成测试、系统测试、验收测试，测试与开发阶段一一对应。
- W 模型：开发与测试并行，测试在需求阶段即介入并贯穿全过程。
- TMap：以结构化方式组织测试生命周期，强调计划、设计、执行、评估与改进的闭环。
- 传统过程强调阶段评审、文档与准入准出条件。

二、敏捷测试过程

- 敏捷宣言与原则：个体与互动、可工作的软件、客户合作、响应变化优先于流程与文档。
- 敏捷测试特征：测试与开发同步、迭代内完成验证、自动化回归支撑持续交付。

- 测试左移（需求与设计阶段介入）与测试右移（生产环境监控与反馈）。
- 基于会话的测试管理（SBTM）用于组织探索式测试，兼顾灵活性与可追溯性。

三、软件测试学派与测试规范

- 主要学派：分析学派（以需求与风险为驱动）、标准学派（以规范与流程为依据）、质量学派、上下文驱动学派、敏捷学派。
- 测试规范包括测试计划、用例规范、缺陷报告规范、测试报告规范与配置管理要求。
- 规范的价值在于保证一致性、可追溯性与可复用性，避免过度形式化。

四、测试过程改进

- 以度量驱动改进：缺陷逃逸率、缺陷密度、用例有效性、回归成本等。
- 常见模型与框架：TMMi、TPI Next、CTP，用于评估并提升测试过程成熟度。
- 改进要点：缺陷根因分析、测试资产复用、自动化覆盖与质量门禁建设。
- 过程改进应结合团队规模与项目特点，小步迭代、持续验证效果。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+流程设计练习（支撑课程目标 1、4）

案例讨论：同一个需求变更：瀑布流程与敏捷流程下测试工作分别如何组织？

- 1、对比两种流程的时间线与角色分工。
- 2、让学生为本组项目设计一页式测试流程。
- 3、讨论没有文档算不算测试规范这一常见争议。

课程思政设计——思政维度：规则意识·法治观念·持续改进

融入载体：GB/T 15532、ISO/IEC 29119 等测试标准，V 模型与敏捷测试过程、缺陷根因分析。

课堂实施：对比“无流程救火”与“规范流程”的成本差异；讨论标准为什么是行业共同语言。

评价观测点：能否按标准要素编写流程文档；是否主动提出过程改进措施。

作业布置：

- 1、绘制本课程项目的测试流程（含准入准出条件）。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、为本组课程设计项目制定一页式测试计划与流程。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 6 次课 2 学时

本次课教学重点：

单元测试目标与用例设计

JUnit 框架使用

集成策略与 CI

本次课教学难点：

测试替身（Mock/Stub）的使用

接口覆盖与集成顺序

本次课教学内容：

第5章 单元测试与集成测试

学习目标：掌握单元测试的目标与用例设计方法；能使用 JUnit 等框架编写并运行测试；理解集成策略与持续集成。

一、单元测试的概念与内容

- 单元测试针对软件最小可测单位（函数、类、方法）验证其逻辑正确性。
- 测试内容：模块接口、局部数据结构、边界条件、独立路径、错误处理路径。
- 单元测试通常由开发者完成，先于集成测试，是缺陷发现成本最低的环节。
- 驱动模块（Driver）与被测模块（Stub）用于替代未完成的上下层依赖。

二、代码评审与静态分析

- 代码评审包括走查、审查、同行评审等，以人工阅读结合检查表发现缺陷。
- 静态分析工具检查编码规范、复杂度、潜在空指针与资源泄漏，可在提交前自动拦截。
- 评审需明确检查项、缺陷记录方式与闭环跟踪，避免流于形式。

三、测试框架与单元测试工具

- xUnit 家族（JUnit、pytest、NUnit）提供断言、测试用例组织与测试报告能力。
- 测试替身分为桩（Stub）、模拟对象（Mock）、伪造对象（Fake），用于隔离外部依赖。
- Mock 用于验证交互行为，Stub 用于提供固定输入，避免过度使用导致测试脆弱。
- 推荐实践：一个用例一个断言目标、命名可读、可重复执行、不依赖执行顺序。

四、集成测试策略

- 集成测试验证模块间接口与协作，关注数据传递、调用时序、全局数据结构与错误传播。
- 集成方式：一次性集成（大爆炸）、自顶向下、自底向上、三明治集成。
- 自顶向下需要桩，自底向上需要驱动，三明治结合两者优点。
- 现代工程更倡导持续集成：小步提交、自动构建与自动化回归，尽早暴露集成缺陷。

五、持续集成与持续测试

- 持续集成要求代码频繁合入主干，每次合入自动触发构建、单元测试与静态检查。
- 持续测试把测试左移到需求与编码阶段，右移到生产环境监控。
- 流水线质量门禁：编译失败、用例失败、覆盖率下降或严重缺陷未修复即阻断发布。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+JUnit 实验（支撑课程目标 2、3）

案例讨论：为什么模块单测都通过，系统仍然出错？结合火星探测器接口案例讨论。

- 1、现场演示 JUnit 测试的编写—运行—失败—修复闭环。
- 2、让学生观察覆盖率报告并讨论覆盖率高的测试是否一定好。
- 3、以航天事故说明接口测试的重要性。

课程思政设计——思政维度：严谨细致·团队协作·责任边界

融入载体：单元测试用例、代码评审检查表、持续集成质量门禁。

课堂实施：开展小组代码互评，要求“对事不对人”，每条意见给出依据与修改建议。

评价观测点：评审意见是否具体可执行；是否接受他人批评并修订代码。

作业布置：

1、为给定 Java 方法编写单元测试，覆盖正常与异常分支。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、在 IDE 中运行 JUnit 测试并查看覆盖率；把测试接入一次本地 CI 流程。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第7次课 2学时

本次课教学重点：

功能测试思路与方法
UI 与 API 功能自动化
回归测试与精准测试

本次课教学难点：

端到端用例的粒度与维护
自动化脚本稳定性

本次课教学内容：

第6章 系统功能测试

学习目标：掌握功能测试的思路与方法；能设计端到端功能用例；了解 UI/API 自动化与回归测试。

一、系统功能测试的目标与过程

- 系统测试在完整系统上验证功能与非功能需求是否满足规格与用户预期。
- 过程：测试需求分析 → 测试计划 → 用例设计 → 环境准备 → 执行 → 缺陷跟踪 → 报告。
- 功能测试关注“做什么”，以黑盒视角依据需求规格、业务规则与用户场景验证。
- 需明确测试准入条件（版本、环境、数据就绪）与准出条件（用例通过率、缺陷收敛）。

二、功能测试用例设计

- 以需求条目为追踪单元，保证每条需求至少被一条用例覆盖，建立需求—用例追踪矩阵。
- 综合使用等价类、边界值、判定表、场景法、错误猜测法设计用例。
- 用例要素：编号、前置条件、测试数据、操作步骤、预期结果、优先级。
- 业务流用例覆盖正常主流程与各类异常分支，如支付失败、库存不足、超时。

三、界面与易用性相关功能验证

- 界面测试检查布局、控件、文案、提示信息、输入校验与多分辨率适配。
- 检查交互一致性：按钮可用状态、必填校验、错误提示是否明确可行动。
- 可用性验证关注任务完成效率与用户出错率，可与少量用户试用结合。

四、接口测试与自动化

- 接口测试绕过界面直接验证服务契约，覆盖参数组合、鉴权、异常码与幂等性。

- 常用工具：Postman、JMeter、requests/pytest 等，可纳入持续集成流水线。
- 自动化适合稳定、重复、回归量大与数据驱动的场景，不适合需求频繁变动部分。
- 自动化收益需评估维护成本，避免用例脆弱导致“自动化负债”。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+接口/UI 测试实验（支撑课程目标 2、3）

案例讨论：电商下单流程中，哪些环节适合 UI 自动化、哪些更适合 API 自动化？

- 1、以业务流程为主线梳理测试点。
- 2、现场演示接口测试与 UI 测试的执行差异。
- 3、组织学生对自动化收益进行估算讨论。

课程思政设计——思政维度：用户思维·服务意识·诚信

融入载体：需求追踪矩阵、功能用例设计、界面与可用性验证；“测试放水/数据造假”警示案例。

课堂实施：角色扮演：学生分别扮演用户、测试与开发，就“这个缺陷要不要发版”进行辩论。

评价观测点：用例是否真正覆盖用户场景；辩论中是否坚持事实依据而非立场。

作业布置：

- 1、为“购物车结算”设计 8 条端到端功能用例。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、使用 Postman 或 JMeter 完成一组接口功能测试并保存测试集。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 8 次课 2 学时

本次课教学重点：

性能测试指标与场景
安全测试方法
兼容性与易用性测试

本次课教学难点：

性能瓶颈定位
安全测试的边界与合规

本次课教学内容：

第 7 章 专项测试：性能、安全、兼容与易用性

学习目标：掌握性能测试指标与场景设计；了解安全测试基本方法；掌握兼容性、可靠性与易用性测试要点。

一、性能测试的目标与指标

- 性能测试为发现性能问题或获取性能指标而开展，需在真实或逼近真实环境下施加

特定负载。

- 核心指标：响应时间、吞吐量（TPS/QPS）、并发用户数、资源利用率、错误率。
- 常见性能问题内因：资源耗尽、资源泄漏（如内存泄漏）、锁竞争、慢查询与算法低效。
- 外部表现：页面卡顿、超时、请求失败、CPU 或内存持续高位。

二、性能测试类型与实施

- 类型：基准测试、负载测试、压力测试、稳定性（疲劳）测试、容量测试、并发测试。
- 实施步骤：确定指标与场景 → 构造脚本与数据 → 环境准备与监控 → 施压 → 分析调优 → 回归验证。
- 性能测试必须同时监控应用、中间件、数据库与主机资源，才能定位瓶颈。
- 调优后需回归确认，防止以牺牲其他指标换取局部改善。

三、安全性测试

- 安全性缺陷即脆弱性（Vulnerability），又称弱点或漏洞；来源包括恶意（木马、陷门、逻辑炸弹）与非恶意（校验错误、隐秘通道）。
- 安全属性：保密性、完整性、可用性（CIA），三者可能相互制约。
- 常见漏洞：注入（SQL/命令）、跨站脚本 XSS、越权访问、敏感信息泄露、不安全配置。
- 测试方法：威胁建模、渗透测试、漏洞扫描、代码审计、模糊测试；贯穿研发全生命周期。
- Web 安全测试应覆盖身份认证、会话管理、权限控制、输入校验与日志审计。

四、兼容性测试

- 兼容性测试验证软件在不同硬件、操作系统、浏览器、数据库、分辨率与网络环境下的可用性。
- 维度：平台兼容、浏览器兼容、版本前后兼容、数据兼容与向下兼容。
- 策略：按用户占比确定测试矩阵，优先覆盖主流与关键组合，配合虚拟机与云真机。
- 重点关注升级安装、数据迁移与新旧版本互操作导致的失败。

五、可靠性与易用性测试

- 可靠性测试关注长时间运行、异常恢复、故障注入与容错能力，指标如 MTBF、MTTR。
- 易用性测试从效率、易学性、易记性、出错率与满意度评价产品可用程度。
- A/B 测试通过对比不同方案在真实用户上的数据表现辅助设计决策。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+JMeter 压测实验（支撑课程目标 2、3）

案例讨论：Uber 隐私泄露与 AWS 宕机：分别属于哪类专项测试范围？

- 1、以指标、场景、工具、分析四步讲解性能测试。
- 2、演示 JMeter 脚本与结果图表解读。
- 3、强调安全测试的合规边界。

课程思政设计——思政维度：国家安全观·工程伦理·绿色低碳

融入载体：网络安全法、数据安全法、个人信息保护法；性能测试资源指标与信创兼容适配（国产操作系统/数据库）。

课堂实施：用真实泄露与宕机事件分析“保密性、完整性、可用性”的权衡；讨论压力测试不得危害真实生产系统。

评价观测点：能否识别合规边界；是否在实验方案中主动提出数据脱敏与授权要求。

作业布置：

- 1、为一个 Web 系统列出性能测试场景与监控指标。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、使用 JMeter 对给定接口做并发压测，记录并解释结果曲线。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第9次课 2学时

本次课教学重点：

本地化技术问题

本地化功能与界面测试

多语言与区域设置

本次课教学难点：

编码与排版差异引发的缺陷

时区、货币、日期格式处理

本次课教学内容：

第8章 软件本地化测试

学习目标：理解本地化的技术问题；掌握本地化功能与界面测试方法；能识别多语言与区域设置缺陷。

一、本地化测试的基本概念

- 本地化（L10n）是在国际化（I18n）基础上，把产品适配到目标语言与地区文化的过程。

- 本地化测试验证本地化后的产品在语言、格式、功能与法律合规方面是否正确、完整、可用。

- 与翻译验证的区别：翻译验证关注文字，本地化测试覆盖语言、界面、功能、数据与文化全维度。

二、翻译验证

- 检查译文准确性、术语一致性、语境恰当性与专业表达规范性。
- 检查语言细节：标点规范、大小写、换行与截断、特殊符号与文化心理适宜性。
- 典型问题：漏译、错译、术语不统一、译文过长导致界面溢出或遮挡。

三、数据格式与区域设置验证

- 日期与时间格式、度量衡单位、货币符号与精度、数字千分位与小数分隔符。
- 复数规则、姓名与地址格式、时区与夏令时处理。
- 字符集与编码：多语言字符显示、排序规则、输入法兼容性。

四、本地化功能测试

- 验证本地化版本的功能完整性与一致性：安装卸载、核心业务流、错误提示与帮助文档。

- 验证区域相关的功能差异：支付方式、税率、隐私政策与法律条款合规。
- 双字节与双向文字（如阿拉伯语 RTL）界面布局验证。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+本地化检查实践（支撑课程目标 2、3）

案例讨论：把 iPhone 日期设置为 1970-01-01 会变砖，这属于什么类型的缺陷？

- 1、对比同一界面在中英文环境下的差异截图。
- 2、演示区域设置引起的日期与货币格式问题。
- 3、让学生动手切换语言与区域做检查。

课程思政设计——思政维度：文化自信·跨文化交流·语言规范

融入载体：本地化测试中的翻译验证、日期/度量衡/复数规则、阿拉伯语 RTL 布局。

课堂实施：以国产软件出海为例，讨论“把中国产品讲给世界听”需要怎样的语言与规范意识。

评价观测点：本地化检查项是否覆盖语言、格式与文化三个维度。

作业布置：

- 1、列举 5 个本地化/国际化常见缺陷并说明验证方法。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、切换系统语言与区域，对给定应用做一次本地化检查并提交缺陷清单。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 10 次课 2 学时

本次课教学重点：

自动化内涵与实现原理
自动化框架类型
自动化实施与维护

本次课教学难点：

框架选型与分层设计
自动化投入产出评估

本次课教学内容：

第 9 章 测试自动化及其框架

学习目标：理解测试自动化的内涵与适用条件；了解常见自动化框架类型；能搭建最小可运行的自动化用例。

一、自动化测试的适用条件

- 适合：需求稳定、执行频繁、回归量大、结果可判定、数据可参数化的测试。
- 不适合：需求频繁变更、一次性验证、依赖主观判断（如界面美观）的测试。
- 自动化率不是目标，投入产出比与维护成本才是决策依据。

二、自动化测试框架的分层

- 常见分层：测试脚本层、测试对象/页面抽象层、数据层、驱动与报告层。
- 关键字驱动、数据驱动与行为驱动（BDD）是常用的框架组织思路。
- 页面对象模型（PO）隔离界面变化，降低脚本维护成本。
- 框架需具备：用例组织、断言、数据管理、日志、报告、失败重跑与并行执行能力。

三、典型框架与工具链

- Web UI：Selenium、Playwright、Cypress；移动端：Appium；接口：Postman、pytest、RestAssured。
- 测试运行器与报告：pytest、TestNG、Allure；持续集成：Jenkins、GitLab CI、GitHub Actions。
- 选型考虑语言栈、社区活跃度、并行与跨端能力、与流水线的集成成本。

四、自动化测试的设计原则与常见问题

- 用例相互独立、可重复执行、不依赖执行顺序与外部人工准备。
- 定位方式优先使用稳定属性，避免依赖易变的样式与绝对路径。
- 常见问题：用例脆弱、等待处理不当、数据污染、失败定位困难与缺乏维护。
- 自动化用例同样需要评审、版本管理与定期清理。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+自动化环境搭建（支撑课程目标 2、3）

案例讨论：什么情况下不适合做自动化？请举例说明。

- 1、以投入产出为核心讨论自动化的价值与边界。
- 2、演示一个最小自动化脚本从编写到运行。
- 3、让学生调研不同工具的适用场景。

课程思政设计——思政维度：创新精神·技术自主可控·效率观

融入载体：自动化测试框架分层、页面对象模型；国产操作系统与国产测试工具的应用案例。

课堂实施：围绕“哪些测试值得自动化、哪些不值得”做成本收益辨析，鼓励提出改进方案。

评价观测点：能否给出自动化取舍的量化理由；是否提出可落地的改进点。

作业布置：

- 1、调研一款自动化测试工具，写出选型对比（至少 3 个维度）。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、搭建 Selenium/Playwright 环境，完成一个最小可运行的自动化用例。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 11 次课 2 学时

本次课教学重点：

测试需求分析方法
测试目标与准则
测试估算与进度安排

本次课教学难点：

需求到测试点的转化
工作量估算与风险识别

本次课教学内容：

第 10 章 测试需求分析与测试计划

学习目标：掌握测试需求分析方法；能把业务需求转化为可测指标；能编写测试计划。

一、测试需求分析

• 测试需求来源于需求规格、设计文档、用户场景与标准规范，需转化为可验证的测试项。

- 识别测试项的可测性：输入、输出、约束、边界与异常处理是否明确。
- 需求可追踪性矩阵用于保证需求—用例—缺陷—报告全链路可追溯。
- 需求评审阶段即应介入测试视角，尽早发现二义性、遗漏与不可测需求。

二、测试目标与测试策略

- 测试目标应可度量：缺陷发现率、用例通过率、需求覆盖率、遗留缺陷等级分布。
- 测试策略明确测试范围（测什么、不测什么）、测试类型组合、准入准出与风险应对。
- 按风险分配资源：高风险模块投入更多测试类型与更严格覆盖。

三、测试计划的内容

- 计划要素：测试范围与目标、测试项与优先级、方法与环境、进度与人员分工、风险与应对、交付物。
- 进度安排需与开发里程碑对齐，预留回归测试与缺陷修复验证时间。
- 计划不是一次性文档，需随需求变更与执行情况滚动更新。

四、测试估算与风险控制

- 估算方法：基于用例数、历史数据、专家判断与工作分解结构。
- 时间不足时的应对：按风险裁剪范围、加大自动化回归、明确遗留风险并向干系人披露。
- 测试风险包括需求不清、环境不可用、数据不可得、人员能力与进度压缩。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+测试计划编写（支撑课程目标 2、4）

案例讨论：需求文档里“系统应快速响应”如何转化为可测试的指标？

- 1、以一条模糊需求为例，现场演练细化为可测指标。
- 2、展示测试计划的完整目录结构。
- 3、组织学生对计划的风险项进行补充。

课程思政设计——思政维度：责任担当·风险意识·契约精神

融入载体：测试需求分析、测试策略与测试计划、资源与进度估算。

课堂实施：给出一个“时间被压缩一半”的项目情境，要求学生按风险裁剪范围并书面披露遗留风险。

评价观测点：计划要素是否完整；风险披露是否诚实、可执行。

作业布置：

- 1、为课程设计题目编写测试需求清单与测试计划（1-2 页）。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、评审一份真实需求说明，输出测试点与测试计划初稿。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 12 次课 2 学时

本次课教学重点：

测试用例构成与设计方法
用例组织与管理
用例评审与维护

本次课教学难点：

用例粒度与覆盖度的平衡
用例库长期维护

本次课教学内容：

第 11 章 设计和维护测试用例

学习目标：掌握测试用例的构成要素；能编写规范用例并组织评审；理解用例维护与复用。

一、测试用例的构成与作用

- 测试用例是为特定测试目标而设计的一组前置条件、输入数据、操作步骤与预期结果的集合。

- 作用：明确“测什么、怎么测、预期是什么”，提升测试的有效性、可复用性、可评估性与可管理性。

- 用例是测试执行与缺陷判定的依据，也是测试资产与知识传递的载体。

二、测试用例的基本元素与书写标准

- 最基本内容：用例编号、测试项、前置条件、测试数据、操作步骤、预期结果。

- 其他重要属性：测试目的与类型、优先级（高/中/低）、所属层次（父/子/孙）、估计用时、设计者与版本。

- 书写要求：避免含糊表述，预期结果必须可判定（Pass/Fail 明确），一条用例聚焦一个验证点。

- 良好用例的特征：能最大程度发现隐藏缺陷、执行效率高、可重复、易维护。

三、测试用例设计原则与颗粒度

- 设计原则：代表性、典型性、覆盖需求、优先针对设计与功能弱点。

- 颗粒度权衡：粗颗粒度用例数量少但定位困难；细颗粒度定位清晰但维护成本高。

- 通过数据驱动把测试数据与步骤分离，兼顾覆盖广度与维护效率。

- 设计方法综合使用等价类、边界值、判定表、场景法与错误猜测法。

四、测试用例的组织与维护

- 测试集（Test Suite）由一组相关用例及其测试环境构成，用于满足特定执行要求。

- 组织维度：按功能模块、按测试类型、按优先级、按业务流程，便于分层执行与回

归选择。

- 维护内容：需求变更后同步更新用例、清理失效用例、补齐缺陷对应的回归用例。
- 用例需评审：可执行性、覆盖充分性、预期结果明确性与冗余情况。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+用例设计与评审（支撑课程目标 2、4）

案例讨论：同一个功能，写 5 条用例与写 50 条用例，如何判断哪个更合适？

- 1、展示好用例与差用例的对比样例。
- 2、组织小组交叉评审并给出修改意见。
- 3、演示测试管理工具中的用例组织方式。

课程思政设计——思政维度：精益求精·用户视角·批评与自我批评

融入载体：测试用例元素与书写标准、用例评审清单、缺陷回归用例补齐。

课堂实施：组织用例评审会：作者自评一同伴评一教师点评，要求给出可判定的预期结果。

评价观测点：用例预期结果是否可判定；能否接受评审意见并闭环修改。

作业布置：

- 1、为“注册”功能编写 12 条规范测试用例并交叉评审。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、使用测试管理工具（如 TestLink）录入用例并执行一轮。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 13 次课 2 学时

本次课教学重点：

测试环境与数据管理

测试工具链与持续集成

缺陷跟踪与配置管理

本次课教学难点：

环境一致性保障

测试数据的构造与脱敏

本次课教学内容：

第 12 章 部署测试基础设施

学习目标：理解测试基础设施的组成；能准备测试环境与测试数据；掌握缺陷跟踪流程。

一、测试环境与基础设施

- 测试环境需尽量贴近生产配置，包括操作系统、中间件、数据库版本与网络拓扑。
- 基础设施资源包括硬件（机架式服务器、刀片式服务器）、软件（操作系统、数据库、Web 服务器、测试工具）、数据与网络资源。

- 环境管理要求：版本可追溯、可快速重建、环境间相互隔离，避免相互干扰。
- 建议用配置管理工具与脚本实现环境自动化部署，减少人工搭建差异。

二、测试数据准备

- 测试数据包括原有数据、正确数据与错误数据，需覆盖正常、边界与异常场景。
- 数据获取途径：生产脱敏数据、脚本生成、工具造数与手工构造。
- 须遵守数据安全与隐私合规，敏感字段脱敏，禁止真实个人信息外泄。
- 数据应可复位，保证用例可重复执行与结果可比较。

三、虚拟机与容器技术

- 虚拟机通过 Hypervisor 在一台物理机上运行多个独立操作系统，分裸金属型与宿主型。
- 容器共享宿主内核，启动快、资源开销小，适合快速交付测试环境。
- 镜像与快照能力便于环境复制、回滚与并行测试，显著降低环境准备成本。

四、缺陷管理与跟踪

- 缺陷生命周期：新建 → 指派 → 修复 → 验证 → 关闭，异常时重新打开或延期处理。
- 缺陷报告要素：标题、复现步骤、实际与预期结果、环境版本、截图日志、严重程度与优先级。
- 严重程度反映技术影响，优先级反映修复紧迫性，两者需分别评定。
- 缺陷跟踪工具（如 Bugzilla、Jira、禅道）提供流程流转、统计分析与度量。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+环境与缺陷流程演练（支撑课程目标 3、4）

案例讨论：在我机器上是好的——如何用基础设施手段避免这类问题？

- 1、以 Docker 为例演示环境一致性。
- 2、展示缺陷生命周期与状态流转图。
- 3、组织一次缺陷提交与验证的模拟演练。

课程思政设计——思政维度：规范操作·数据合规·诚实守信

融入载体：测试环境与数据准备、个人信息保护与数据脱敏、缺陷生命周期与跟踪工具。

课堂实施：演练缺陷报告撰写与流转，强调“不夸大、不隐瞒、可复现”；讨论违规使用真实数据的法律后果。

评价观测点：缺陷报告是否客观可复现；实验数据是否脱敏、来源合规。

作业布置：

- 1、画出本组项目的测试环境与缺陷处理流程图。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、搭建 Docker 测试环境；在缺陷跟踪工具中完成一次缺陷全流程演练。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习

例题选择与个别辅导的依据。

第 14 次课 2 学时

本次课教学重点：

测试执行与准入准出

缺陷报告规范

测试结果评估与报告

本次课教学难点：

缺陷严重程度与优先级判定

质量评估与发布建议

本次课教学内容：

第 13 章 测试执行与结果评估、报告

学习目标：掌握测试执行的组织方式；能编写规范的缺陷报告；能完成测试结果评估与测试报告。

一、测试执行与结果记录

- 执行前确认准入条件：版本正确、环境与数据就绪、用例已评审。
- 执行中如实记录实际结果、执行时间、环境与失败现象，保留截图与日志证据。
- 用例状态：通过、失败、阻塞、未执行；阻塞需说明原因与解除条件。
- 失败用例应先做自检（数据、环境、脚本），再判定为缺陷，避免误报。

二、缺陷报告与跟踪

- 缺陷报告的核心是可复现：步骤最小化、描述客观、证据充分、期望结果明确。
- 缺陷分类与定级依据影响范围、严重程度与出现频率，便于研发排期。
- 跟踪闭环：修复后回归验证，确认无回归并关闭，遗留缺陷需评估发布风险。
- 定期做缺陷分析：缺陷密度、分布、根因与趋势，反哺设计与测试改进。

三、测试进度与质量管理

- 进度管理方法：里程碑与任务分解、燃尽图、每日站会同步阻塞与风险。
- 进度与质量、成本相互制约，压缩测试时间必然提高遗留缺陷风险，需显式权衡。
- 质量度量：需求覆盖率、用例执行率与通过率、缺陷发现与修复趋势、遗留缺陷等级。

- 测试报告需给出明确的发布建议与遗留风险说明。

四、测试报告与结果评估

• 测试报告内容：测试范围与依据、环境与数据、执行统计、缺陷汇总、结论与建议、风险与遗留问题。

- 结论必须有数据支撑：用例执行情况、缺陷分布与趋势、关键需求验证结果。
- 评估需区分“测试完成”与“质量达标”：执行完毕不代表可以发布。
- 报告面向不同干系人：管理层关注结论与风险，开发关注缺陷细节，需分层表述。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+测试报告撰写（支撑课程目标 3、4）

案例讨论：一个偶现缺陷如何写出高质量报告并推动修复？

- 1、展示缺陷报告模板与反例。
- 2、演练从执行记录到测试报告的汇总过程。
- 3、讨论测试通过与可以发布的区别。

课程思政设计——思政维度：实事求是·责任担当·合规意识

融入载体：测试执行记录、缺陷定级与跟踪、测试报告与发布建议。

课堂实施：给定一组含失败用例与遗留缺陷的执行数据，要求写出有依据的发布建议

并公开说明风险。

评价观测点：结论是否有数据支撑；是否如实报告未通过项与遗留风险。

作业布置：

- 1、编写一份完整测试报告（含覆盖率与缺陷分析）。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、执行一轮系统测试，提交缺陷并输出测试报告。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 15 次课 2 学时

本次课教学重点：

测试技术发展趋势
测试职业发展路径
综合案例复盘

本次课教学难点：

新技术在测试中的应用边界
从案例中提炼可迁移方法

本次课教学内容：

第 14 章 软件测试展望与综合案例

学习目标：了解软件测试技术发展趋势；了解测试岗位能力要求；能够综合运用所学方法分析案例。

一、测试技术的发展趋势

- 测试左移与右移：需求与设计阶段介入，生产环境持续监控与反馈闭环。
- 智能化测试：用大模型辅助用例生成、脚本维护、缺陷定位与测试数据分析。
- 精准测试与变更影响分析：依据代码变更与调用关系选择回归范围，降低回归成本。
- 质量内建：把质量责任嵌入研发流程与工程能力，而非依赖末端把关。

二、测试组织与岗位能力

- 岗位方向：功能测试、自动化测试、性能测试、安全测试、测试开发与质量工程师。
- 能力结构：测试理论与方法、编程与工具、业务理解、分析与沟通、工程规范意识。
- 职业发展强调持续学习与工程质量思维，测试开发能力是重要增长点。

三、综合案例复盘

- 以本学期真实缺陷事故案例为主线，复盘缺陷产生的根因与可预防环节。
- 对照测试流程与规范，评估案例中缺失的测试活动与改进措施。
- 把案例经验转化为检查表与测试策略，应用于课程设计项目。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：讲授+案例复盘（支撑课程目标 1、4）

案例讨论：AI 能替代测试人员吗？结合本章案例谈谈你的判断。

- 1、以趋势、岗位、案例三段式展开。
- 2、组织开放式辩论：AI 对测试岗位的影响。
- 3、引导学生规划个人能力发展路径。

课程思政设计——思政维度：科技报国·创新自信·终身学习

融入载体：测试左移右移、精准测试、AI 辅助测试的发展趋势；国产软件质量提升历程。

课堂实施：讨论“AI 会取代测试工程师吗”，引导学生明确人的判断责任与技术伦理边界。

评价观测点：能否结合国家需求与技术趋势谈个人发展规划。

作业布置：

- 1、调研一项测试新技术，写 500 字评述其对测试岗位的影响。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第 3 版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、完成课程设计项目自测，准备项目汇报材料。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

第 16 次课 2 学时

本次课教学重点：

测试方案与用例设计展示
测试执行结果与缺陷分析
项目答辩与互评

本次课教学难点：

测试结论的合理性与证据完整度
答辩表达与问题回应

本次课教学内容：

综合实践 课程设计项目汇报与答辩

学习目标：能够完整呈现测试方案与执行结果；能够分析缺陷并给出质量结论；能够进行工程表达与答辩。

各组按统一模板汇报课程设计成果，重点说明测试策略、用例设计依据与缺陷发现过程。

评分依据包括方案合理性、用例质量、执行严谨性、报告规范性与答辩表现。

通过互评与教师点评，总结共性问题与改进方向，形成课程学习闭环。

教学组织（含课堂教学方法、辅助手段、师生互动、时间分配、板书设计、重点如何突出、难点如何解决等）：

教学方式：汇报答辩+教师点评（支撑课程目标 1-4）

案例讨论：互评：如果你是项目经理，是否同意该项目发布？给出理由。

- 1、每组限时汇报，组间提问与互评。

- 2、教师按评分表打分并现场点评。
- 3、汇总共性问题与优秀实践，形成课程总结。

课程思政设计——思政维度：学术诚信·团队协作·职业素养

融入载体：课程设计项目验收与答辩：测试计划、用例、执行记录、缺陷清单、测试报告。

课堂实施：答辩中设置“诚信提问”：数据如何取得、结论如何验证、有无未完成项与遗留风险。

评价观测点：作品数据是否真实可追溯；团队分工是否明确、答辩是否如实回应。

作业布置：

- 1、提交课程设计报告与测试资产（用例、脚本、缺陷记录、测试报告）。

本次课推荐和参考文献

- 1、朱少民. 全程软件测试（第3版）[M]. 北京：人民邮电出版社，2019.
- 2、朱少民. 致命 Bug——软件缺陷的灾难与启示[M]. 北京：人民邮电出版社，2016.
- 3、朱少民. 软件测试知识地图[Z]. 2022.
- 4、陈能技. 软件测试技术大全：测试基础、流行工具、项目实战[M]. 北京：人民邮电出版社，2008.
- 5、Myers G J, Sandler C, Badgett T. The Art of Software Testing[M]. 3rd ed. Hoboken: John Wiley & Sons, 2011.

本次课实践教学设计：

- 1、完成项目汇报答辩，提交全部课程设计成果物。

课后自我总结分析：

根据课堂提问、实验完成情况与作业反馈，记录学生易错点与薄弱环节，作为下次课复习例题选择与个别辅导的依据。

（教案一律使用 A4 纸打印）