



# 中华人民共和国金融行业标准

JR/T 0175—2019

---

## 证券期货业软件测试规范

Specification for securities and futures industry software test

2019 - 09 - 30 发布

2019 - 09 - 30 实施

---

中国证券监督管理委员会 发布



# 目 次

前言.....	III
引言.....	IV
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 总体要求.....	2
5 单元测试.....	17
6 集成测试.....	18
7 系统测试.....	20
8 系统集成测试.....	21
9 验收测试.....	21
附录 A（规范性附录） 测试流程.....	24
A.1 测试估计.....	24
A.2 测试计划.....	24
A.3 测试设计.....	24
A.4 测试执行.....	26
A.5 测试报告.....	26
附录 B（资料性附录） 软件测试模板.....	28
B.1 测试工作量估计书.....	28
B.2 测试计划.....	29
B.3 测试需求.....	30
B.4 测试用例.....	31
B.5 测试缺陷列表.....	32
B.6 评审记录.....	33
B.7 测试总结报告.....	33
B.8 管理者报告.....	35
B.9 里程碑总结报告.....	36
附录 C（规范性附录） 测试技术.....	38
C.1 黑盒测试.....	38
C.2 白盒测试.....	44
C.3 灰盒测试.....	45
附录 D（规范性附录） 静态测试方法.....	46
D.1 概述.....	46
D.2 评审.....	46

附录 E（规范性附录）业务分类.....	49
附录 F（规范性附录）缺陷管理.....	50
F.1 缺陷管理目的.....	50
F.2 缺陷管理的生命周期.....	50
F.3 缺陷管理流程图.....	52
F.4 缺陷严重度划分.....	53
F.5 缺陷优先级划分.....	53
F.6 根本原因举例.....	53
参考文献.....	55

## 前 言

本标准按照GB/T 1.1-2009给出的规则起草。

本标准由全国金融标准化技术委员会证券分技术委员会（SAC/TC180/SC4）提出。

本标准由全国金融标准化技术委员会(SAC/TC180)归口。

本标准起草单位：中国证券监督管理委员会信息中心、中国证券监督管理委员会证券投资基金机构监管部、大连商品交易所、上海证券交易所、深圳证券交易所、上海期货交易所、郑州商品交易所、中国金融期货交易所、证券期货业信息技术测试中心（大连）、中国银河证券股份有限公司、南方基金管理有限公司。

本标准主要起草人：张野、刘铁斌、蒋东兴、周云晖、马晨、高红洁、王恺、郭郢、陈楠、王凤海、许强、胥海涛、刘军、孙瑞超、陆素源、蒋凯、陈彦、喻华丽、万春波、顾军妹、陈亮、邹杏忠、徐玲、于三禄、刘相富、鲁继东、戴鹏、刘进、汪璇璇、董琳、徐艳、刘洁如、韩秀玲、邓志远、刘丹、唐沛来、邓廷勋、邱星、牛云峰。

## 引 言

证券期货业市场的发展和平稳运行高度依赖行业信息系统。规范统一行业信息系统的测试标准，有助于规范行业信息系统的测试管理，提高行业软件测试过程的规范化程度，促进行业软件测试水平整体提升，从而降低行业信息系统运行风险，促进证券期货业的可持续性健康发展。

本标准具有鲜明的行业特征，将行业经验、实践与测试理论紧密结合，使标准更加易于落实和实施，能够有效指导行业软件测试活动、提高测试质量，其中：

- a) 测试管理与测试流程基于《CMMI 软件能力成熟度集成模型》和《TMMI 测试成熟度模型》编制，并在此基础上提供以下适应行业特点的内容：
  - 1) 提供各测试阶段的工作要求，更细致地指导测试工作开展，见 4.2.1；
  - 2) 明确各测试文档规范、基本内容并提供参考模板，见 4.2.2 及附录 B；
  - 3) 提供测试管理细则，明确各项管理工作具体要求和实施内容，以便测试管理工作切实开展，见 4.6；
  - 4) 针对行业特点细化测试流程中各阶段工作内容及要求，见附录 A；
  - 5) 细化各测试类型及测试级别的测试准入/准出要求，用于更加细致地指导测试具体工作，见 4.5、5.5、6.5、7.5、8.5、9.5。
- b) 系统内容与测试类型、测试级别与测试类型的基本要求遵循 GB/T 15532-2008《计算机软件测试规范》并根据行业特点明确以下关系：
  - 1) 系统内容与测试类型之间的关系，便于行业机构根据软件产品可能涉及的测试内容选择适合的测试类型，同时增加应急演练、联网测试、全市场测试及选型测试四个测试类型。
  - 2) 测试级别与测试类型的对应关系，便于行业机构根据其角色、软件产品特点等实际情况，选择适合的测试级别和测试类型，确保行业机构履行测试工作职责、保证软件产品质量。
- c) 本标准内容符合 JR/T 0146-2016（所有部分）《证券期货业信息系统审计指南》系列标准中与软件测试相关的审计要求。

# 证券期货业软件测试规范

## 1 范围

本标准规定了证券期货业信息系统建设过程中的总体要求、单元测试、集成测试、系统测试、系统集成测试、验收测试等测试活动的内容。

本标准适用于证券期货业市场核心机构（以下简称市场核心机构）、证券期货基金经营机构（以下简称市场经营机构）、证券期货信息技术服务机构（以下简称市场服务机构）开展证券期货业计算机软件和外部信息系统测试工作。

注1：市场核心机构，如证券期货交易所、证券登记结算机构、期货市场监管中心等；

注2：市场经营机构，如证券公司、期货公司、基金公司等；

注3：市场服务机构，如软件开发商、信息商、服务商等。

## 2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 11457-2006 信息技术 软件工程术语

GB/T 15532-2008 计算机软件测试规范

GB/T 29834.3-2013 系统与软件维护性 第3部分：测试方法

JR/T 0145-2016 资本市场交易结算系统核心技术指标

## 3 术语和定义

GB/T 11457-2006界定的以及下列术语和定义适用于本文件。为了便于使用，以下重复列出了GB/T 11457-2006中的某些术语和定义。

### 3.1

**测试级别 test level**

根据软件开发生命周期、测试对象及测试人员不同，将软件生命周期中的测试活动分为不同的级别。

### 3.2

**测试流程 test process**

一个完整的测试任务按顺序经历的过程。

### 3.3

**测试用例 test case**

为特定目标编写的测试输入、执行操作及期望结果的集合。

注：改写GB/T 11457-2006，定义2.1695

### 3.4

**测试轮次 test cycles**

项目中在一个完整、全量发布版本上将项目计划的所有可执行用例执行一遍的次数。

3.5

**通关测试** production readiness test

新系统、新业务、新产品上线启用前，由市场核心机构组织并由全市场参与者共同参与，在生产环境上模拟系统/业务/产品上线首日行为的测试。

3.6

**第三方测试机构** the third party testing institute

处于用户和产品开发商利益之外、根据有关标准或规范对产品进行客观质量检验的专业测试机构。

3.7

**裁剪** tailor

对相关测试活动、测试方法、测试内容、测试类型、测试输出文档等进行调整、增加、删除、替换、顺序变更等。

4 总体要求

4.1 测试目的

验证软件满足软件开发合同、系统/子系统设计文档、系统需求规格说明书、系统设计说明书、产品说明书、运维手册和操作手册等软件质量要求。

通过测试发现软件缺陷，为软件产品的质量评价提供依据。

4.2 测试流程

4.2.1 测试阶段的工作要求

各测试级别、测试类型的测试应遵循此测试流程。此测试流程分为测试估计、测试计划、测试设计、测试执行、测试报告五个阶段，各阶段的活动、输入输出见附录A。

各测试阶段应遵守的工作要求见表1。

表 1 工作要求

测试阶段	工作要求
测试估计	a) 功能测试和非功能测试的测试范围合理、覆盖全面、无冗余； b) 估计工作量时，测试策略选择合理、有效； c) 功能测试及非功能测试均完成测试工作量估计，且有充分的估计依据； d) 如有测试工具开发、验收测试支持、测试管理等其他工作，均完成测试工作量估计，且有充分的估计依据。
测试计划	a) 明确功能与非功能的测试范围、测试策略、测试执行的准入/准出要求； b) 测试策略与测试工作量估计书中的测试策略存在明显差异时，应有合理的解释说明； c) 详细测试进度安排合理、里程碑时间点明确，避免测试不足或过度测试； d) 测试环境部署策略符合系统规划要求，使用的测试工具名称、版本和用途描述清晰； e) 明确在测试活动中可能发生的风险及其影响和应急措施； f) 明确测试过程中的问题管理流程、变更管理流程、进度管理流程、缺陷管理流程。



表 1 工作要求（续）

测试阶段	工作要求
测试设计	a) 测试需求：应 100%覆盖业务需求和系统需求规格说明书，并适当参考相关开发设计文档； b) 测试用例：应 100%覆盖测试需求或按照测试级别不同 100%覆盖相关设计、需求文档，如单元测试、集成测试覆盖相关开发设计文档，系统测试覆盖系统需求规格说明书等； c) 非功能测试场景的设计中充分考虑可能产生影响的业务数据量，且在真实生产环境下有效、在测试环境下可测。
测试执行	a) 明确记录每条测试用例的执行结果，未执行的用例应写明未执行原因； b) 对于核心、重要的系统及功能应对测试执行过程以截图或者保留日志的方式进行留痕； c) 执行过程中发现的缺陷及问题，应及时记录在缺陷管理工具或者缺陷列表中，并进行及时处理，测试结束后无非最终状态的缺陷。
测试报告	a) 明确功能及非功能的测试范围，如与测试计划中存在差异，可进行解释说明； b) 明确测试结论，从功能及非功能角度分别描述具体的通过标准及实际测试结果，给出是否通过的结论； c) 写明测试环境部署情况，测试环境与生产环境的差异分析； d) 明确测试版本、测试执行情况、对缺陷进行汇总及分析； e) 进行改善建议、经验教训总结，为后续工作提供参考。

#### 4.2.2 测试文档规范

各测试阶段的输出文档，宜包括的基本内容见表2，可根据实际需要适当裁剪，具体模板参见附录B。

表 2 文档的基本内容

阶段	输出文档	文档的基本内容
测试估计	测试工作量估计书	各系统模块的测试设计复杂度、测试执行复杂度、测试用例估计个数、测试设计工作量、测试执行工作量、非功能测试工作量等。
测试计划	测试计划	测试目标、测试范围、测试策略、测试进度计划、测试环境、测试方法、测试管理、测试报告等。
测试设计	测试需求	需求功能、验证点、验证方式等。
	测试用例	测试用例ID、优先级、场景说明、方法/函数、正常系/异常系、前提条件、测试目的、测试步骤、测试数据、期望结果等。
测试执行	测试缺陷列表	缺陷ID、缺陷描述、再现步骤、期望结果、实际结果、测试轮次、测试类型、发现版本、发现日期、缺陷负责人、缺陷提交人、严重度、优先级、缺陷状态、根本原因、解决方案、关闭日期等。
	测试执行结果	测试用例的实际执行结果、通过或不通过、对应的缺陷ID、测试人员、实际测试日期、测试步骤截图等。
测试报告	测试总结报告	测试范围、测试结论、测试环境与生产环境的差异分析、功能测试/非功能测试执行情况、缺陷汇总及遗留问题分析、测试结果分析、经验教训、改善建议等。
测试管理	管理者报告	进度情况、工作量偏差、质量目标达成情况、风险与问题等。
	里程碑总结报告	里程碑的进度、成本和质量情况、风险与问题、经验教训、改善建议等。

### 4.3 测试类型

#### 4.3.1 功能测试

功能测试的主要目的、测试内容及测试技术如下：

a) 目的

以系统需求规格说明书为依据对其中要求的功能实现、业务操作等系统功能特性进行验证，目的是确保系统在指定的条件下满足客户的功能性和易用性要求。

b) 测试内容/关注点

测试内容/关注点包含实现的功能正确、不存在无用的功能和不存在功能的遗漏，具体如下：

- 1) 实现的功能正确：与系统需求规格说明书上要求的功能一致；
- 2) 不存在无用的功能：实现的功能范围未超出系统需求规格说明书描述的范围；
- 3) 不存在功能的遗漏：实现的功能范围能完全覆盖系统需求规格说明书描述的范围。

c) 测试技术

以黑盒测试技术为主，灰盒测试技术为辅。

本标准中提及的黑盒测试技术、白盒测试技术及灰盒测试技术见附录 C。

#### 4.3.2 性能测试

性能测试的主要目的、测试内容及测试技术如下：

a) 目的

通过模拟真实、高压等各种负载条件的业务场景，对系统的各项性能指标进行评估，通常包括负载测试、压力测试、容量测试、业务响应时间测试等。

b) 测试内容/关注点

测试内容/关注点包含负载测试、压力测试、容量测试和业务响应时间测试，具体如下：

- 1) 负载测试：对系统不断地增加压力或持续保持最大安全负载，直到系统的某项或者多项性能指标达到安全临界值（例如某种资源已经达到饱和状态等），获得系统在不同负载条件下的性能指标；
- 2) 压力测试：超过安全负载的情况下，对系统不断地施加压力，通过确定一个系统的瓶颈或不能接收用户请求的性能点，来获得系统可提供的最大服务级别的测试；
- 3) 容量测试：使用反映系统容量特征的极限值（如最大并发用户数、数据库记录数等）对系统施加压力，通过观察系统在极限容量状态下没有出现任何软件故障或还能保持主要功能正常运行来检测系统的容量指标；
- 4) 业务响应时间测试：被测系统在稳定压力持续运行过程中或达到一定容量后进行业务操作，记录系统处理所花费的时间，以此检测业务响应时间符合性能需求。

c) 测试技术

具体如下：

- 1) 负载测试、压力测试和业务响应时间测试的测试方法是通过模拟单用户和多用户并发的场景，获取软件系统功能处理的延时、吞吐率及处理时间。利用自动化测试工具进行数据构造、负载压力构造，通常选择具有代表性或者核心业务的模块、访问量较高的业务构造各种性能测试场景，从而进行单机性能测试或并发性能测试。测试指标包括订单峰值吞吐速率、成交峰值吞吐速率、订单持续吞吐速率、成交持续吞吐速率、订单处理延时、日结算处理能力等；

- 2) 容量测试的测试方法是在不同的系统配置以及不同的业务场景下构造相应的容量数据，评估系统的极限容量和硬件资源占用率。其中，极限容量包括最大用户数、最大处理量、最大事务数（执行单元）、最大吞吐率、最大文件容量等；硬件资源占用率包括内存、CPU、硬盘、网络带宽、设备等。测试指标包括日订单处理容量、日成交处理容量等。

#### 4.3.3 可靠性测试

可靠性测试的主要目的、测试内容及测试技术如下：

##### a) 目的

可靠性测试是通过模拟系统可能出现的各种异常或故障等场景，来验证系统在异常处理、故障容错、数据恢复及容灾等方面的能力，并对系统的可靠性进行评估，目的是确保系统在特定的环境部署条件下满足高可用性要求，通常包括成熟性测试、容错性测试、稳定性测试和可恢复性测试。

##### b) 测试内容/关注点

测试内容/关注点包含成熟性测试、容错性测试、稳定性测试和可恢复性测试，具体如下：

- 1) 成熟性测试：当软件系统中出现代码判断错误、集成中的接口错误、通讯报文错误、系统中业务逻辑错误以及其他在设计、开发等环节中造成的错误时，验证系统能消除错误造成的影响并仍可正常工作；
- 2) 容错性测试：当引入外部错误，如违规操作、删除数据、强行终止等严重的行为使系统(包括硬件、软件及附属程序)发生异常时，验证系统在异常情况下应具有防护性措施(处理异常的方法包括：系统自动处理和人工干预处理)；
- 3) 稳定性测试：通过持续性测试的方法验证系统在一定压力下长时间运行的稳定性；
- 4) 可恢复性测试：系统在失效状态下，恢复至正常状态的能力。失效状态应是系统崩溃或者宕机停止所有功能，而不是带故障的运行状态；正常状态应是至少保证系统的主要功能可全部运行。获取系统恢复时间(RTO)、数据恢复时间(RPO)，RTO、RPO定义见JR/T 0145-2016。

##### c) 测试技术

具体如下：

- 1) 成熟性测试、容错性测试的测试方法是通过人工模拟故障场景，如进程崩溃及挂起、网络断开及延迟、服务器宕机等场景，或通过工具或手工制造异常操作或异常数据输入等，验证系统或组件在出现故障时进行有效处理的能力；
- 2) 稳定性测试的测试方法是7\*24小时压力测试，即仿照生产的业务配比关系，根据实际生产需要，保持不同倍数下单压力，每天运行被测系统24小时，连续运行7天。在测试期间可通过查看系统进程状态、执行功能操作、查看运行日志信息的方式来验证系统运行状态的正确性；
- 3) 可恢复性测试的测试方法是模拟在出现故障或灾难导致系统失效时，通过分析系统日志或流水计算得到系统恢复时间(RTO)和数据恢复时间(RPO)指标。

#### 4.3.4 安全性测试

安全性测试的主要目的、测试内容及测试技术如下：

##### a) 目的

通过安全测试手段和方法，验证软件的安全特性实现与预期一致、检验软件产品对各种攻击情况的防范能力，从而保障产品的安全质量。

b) 测试内容/关注点

具体如下：

- 1) 配置管理：按照信息安全等级保护要求对操作系统、应用程序、数据库、网络设备等进行了安全配置；
- 2) 资源利用：应用程序内存使用和资源竞争问题，如变量未初始化、数组越界、内存溢出、资源泄露、进程线程异常等；
- 3) 身份鉴别：对登录操作系统、数据库系统和应用系统的用户进行了身份标识和鉴别，如应使用强密码策略、限定连续登录尝试次数、使用有效验证码等；
- 4) 访问控制：用户应根据自己的权限大小来访问系统资源（如服务器、目录、文件等），不得越权访问；
- 5) 数据保护：应用程序存储敏感信息的情况，存储的敏感信息应进行加密，包括但不限于密码、密钥等敏感信息应加密写入缓存、文件系统、数据库中等；
- 6) 通信安全：应用程序使用安全通信协议（如 SSL、TLS）的情况，对敏感信息的传输应进行加密等；
- 7) 会话管理：本地及服务器端对会话超时的设置，会话结束后应用程序使用过的敏感信息应从内存中删除，合法用户的会话不被劫持等；
- 8) 数据验证：应用程序在使用前正确验证来自客户端或外界输入数据的情况，避免发生如跨站脚本攻击、命令注入、文件注入、SQL 注入等漏洞；
- 9) 安全审计：对网络系统中的网络设备运行状况、网络流量、用户行为等进行日志记录；审计记录应包括事件的日期和时间、用户、事件类型、事件成功及其他审计相关的信息等。

c) 测试技术

常见的安全性测试技术有安全功能检查、代码安全测试、漏洞扫描、渗透测试和模糊测试，具体如下：

- 1) 安全功能检查是指测试人员通过对相关系统管理人员进行访谈和对测试对象（如相关系统、各类设备、基线标准、开发设计文档）进行观察、验证、分析以确认测试对象符合信息安全等级保护、相关信息安全要求以及安全功能需求的过程；
- 2) 代码安全测试分为静态检测和动态检测。静态检测是使用代码检测软件进行代码静态扫描，并对扫描后结果进行人工分析来发现定位漏洞的过程；动态检测即在运行时进行错误检查，主要通过将已经写好的程序转换成一个新程序（其功能与原程序是等价的），新程序包含有一些额外的代码，它们的作用是在程序执行期间检查错误；
- 3) 漏洞扫描是指利用专业的漏洞扫描工具，基于公共漏洞和暴露（CVE）漏洞数据库或特征库，通过扫描、探测等方法对目标系统的安全情况进行检测，发现可利用的漏洞；
- 4) 渗透测试是通过模拟恶意黑客的攻击方法对目标系统进行模拟入侵，以找出系统中的脆弱点、技术缺陷和漏洞；
- 5) 模糊测试是一种通过提供非预期的输入并监视异常结果来发现软件故障的方法，是一个自动的或半自动的过程，通常包括识别目标、识别输入、生成模糊测试数据、执行模糊测试、监视异常、确定可利用性等阶段。

#### 4.3.5 可移植性测试

可移植性测试的主要目的、测试内容及测试技术如下：

a) 目的

通过模拟部署在不同的运行平台来验证系统运行的适应性，目的是确保系统在运行环境发生变化时运行及处理可满足预期要求，通常包括可安装性测试、兼容性测试、适应性测试和可替换性测试。

b) 测试内容/关注点

测试内容/关注点包含可安装性测试、兼容性测试、适应性测试和可替换性测试，具体如下：

- 1) 可安装性测试：在目标环境对软件安装程序所进行的测试，包括操作系统安装软件或在客户个人电脑上安装软件产品的安装向导软件。主要关注：安装文档应齐全、软件安装的程序文件应齐全、文件格式应与安装指导中要求的文件格式相同；所有预置的数据应齐全；软硬件环境的配置应合理；所有文件应正确产生并确有所需要的内容等；
- 2) 兼容性测试：如果不存在相互依赖关系的计算机系统可在同一环境（例如：同一个硬件平台）中运行，而不影响彼此的行为（如资源冲突），则称之为兼容，包括应用软件与硬件平台、应用软件与操作系统、操作系统与硬件平台间的兼容性，例如：当新的或升级之后的软件被大量装入已经安装了应用程序的环境（例如：服务器）时，应执行兼容性测试；
- 3) 适应性测试：软件系统能在所有特定的目标环境（硬件、软件、中间件、操作系统等）中正确运行，包括软件系统在不同处理器配置下的适应性、在台式机与专业服务器下的适应性、在不同操作系统下的适应性、在不同网络环境下的适应性；
- 4) 可替换性测试：软件系统中组件能被替换的能力以及软件升级引起的新旧版本的可替换性。

c) 测试技术

可安装性测试、兼容性测试、适用性测试和可替换性测试的测试方法如下：

- 1) 可安装性测试的测试方法：模拟软件系统在目标环境中进行安装、升级及卸载等操作，验证其正常运行能力；
- 2) 兼容性测试的测试方法：模拟软件系统在目标环境中与其他应用程序、升级后操作系统等软硬件共存的场景，验证其正常运行能力；
- 3) 适应性测试的测试方法：模拟软件系统在不同的特定目标环境下运行的场景，验证其正常运行能力；
- 4) 可替换性测试的测试方法：模拟系统在不同软件组件或者版本上运行的场景，验证其正常运行能力，尤其对于以第三方商业软件为特定组件的软件系统。

#### 4.3.6 可维护性测试

可维护性测试的主要目的、测试内容及测试技术如下：

a) 目的

通过评审系统设计结构、开发语言、系统文档及用户文档来评估系统后续易于修改、优化及能满足新需求的开发，目的是确保系统在交付后直至到期被淘汰的整个时期内易于理解和改进。

b) 测试内容/关注点

具体如下：

- 1) 软件系统具有错误的易分析性，内部：检验设计、文档、代码的书写规格，代码注释和错误标注规范性和充分性等，外部：检验错误发生时系统的响应结果，便于发现和准确定位问题；



- 2) 软件系统具有使指定的修改可被实现的能力，包括编码、设计和文档的更改，实现参数文件、数据库、表、操作方式、个性化配置；
  - 3) 修改后的软件系统仍然能达到所需功能要求和运行指标。
- c) 测试技术
- 常见的维护性测试技术有：专家评审法、技术测试法和用户调查法，具体见 GB/T 29834.3-2013。
- 1) 专家评审法是一种主观的评审方法。评审时，应根据被评审对象和评审目的，设计评审项目表，列出栏目、分值、权重和打分规则。专家根据自身的经验与认知，进行判断打分，然后根据专家的权重和统计规则计算最终评分；
  - 2) 技术测试法是一种客观的评分方法。技术测试时，可依据被测对象和测试目的，选择适用的自动化测试工具，也可人工进行手动测试。技术测试获得的结果是一种量化的测量结果；
  - 3) 用户调查法是一种面向用户群的问卷征询方法。用户调查时，应根据调查的目的和特定的用户群，设计调查表，让被调查对象填写并反馈，调查表回收数应达到一定的数量，并不低于发出的适当比例。然后对回收的调查表进行汇总计算，其计算值作为用户调查的结果。

#### 4.3.7 应急演练

应急演练的主要目的、测试内容及测试技术如下：

- a) 目的  
充分分析可能出现的各种突发事件并对其制定相应应急预案，通过评审及演练，检验预案的实用性、可用性、可靠性，相关人员应急行动实施的有效性以及避免事故、防止事故、抵抗事故的能力，目的是确保在上线过程或上线后突发重大事故、设备故障等情况下，相关人员可及时反应、妥善处理，从而使事故造成的影响和损失降到最低。
- b) 测试内容/关注点  
具体如下：
  - 1) 应急预案具有完整性、正确性、一致性及易理解性，符合应急要求；
  - 2) 突发事件场景考虑全面，应急过程所用工具、脚本可用，按照应急预案中描述的操作步骤可将系统恢复至正常运行状态；
  - 3) 系统恢复时间（RTO）、数据恢复时间（RPO）符合应急要求。与可靠性测试的区别在于测试环境不同，且测试场景更贴近于生产已经发生过的或者可能发生的紧急情况；
- c) 测试技术  
以黑盒测试为主，通过应急预案静态检查、应急预案模拟演练、分析系统日志或流水等方法，保证应急预案的有效性和正确性。

#### 4.3.8 联网测试

联网测试的主要目的、测试内容及测试技术如下：

- a) 目的  
市场核心机构组织部分或全部市场参与方，在测试环境或准生产环境开展的接口处理及业务流程测试，目的是验证交易所和各参与方的接口处理及业务流程的正确性，以及网络的连通性。此测试类型可根据实际情况进行裁剪。
- b) 测试内容/关注点

具体如下：

- 1) 验证机构间接口含义理解及处理的一致性、正确性；
  - 2) 验证机构间业务及数据交互流程的正确性。
- c) 测试技术
- 以黑盒测试为主，参测各方通过场景设计对测试关注点进行验证。

#### 4.3.9 全市场测试

全市场测试也称为全网测试，其主要目的、测试内容及测试技术如下：

- a) 目的
- 利用非交易日，市场核心机构组织部分或全部市场参与方，利用生产环境进行联网测试，目的是通过模拟真实的市场行为来验证交易所和各参与方的系统功能、性能、可靠性以及整体业务流程等功能测试和非功能测试与需求规格说明书的一致性，确定系统具备上线条件。
- b) 测试内容/关注点
- 具体如下：
- 1) 各参与者系统和具体业务的功能及流程应正确，性能、可靠性等非功能测试应满足需求规格说明书；
  - 2) 系统上线所需线路等技术条件应完备并可正常使用；
  - 3) 参与市场业务各方上线条件应已完备；
  - 4) 新程序或系统上线后，相关系统能正常适应；
  - 5) 可根据实际情况需要，增加通关测试。
- c) 测试技术
- 以黑盒测试为主，参测各方通过场景设计对测试关注点进行验证。

#### 4.3.10 选型测试

选型测试的主要目的、测试内容及测试技术如下：

- a) 目的
- 在软件竞品选择的过程中，使用测试理论和方法对软件竞品进行多种技术指标的评测，通过多个维度对软件竞品进行评测对比，为软件竞品选择提供参考指标数据。
- b) 测试内容/关注点
- 测试内容/关注点包含软件可靠性选型、软件性能选型、安全选型和软件功能选型，具体如下：
- 1) 软件可靠性选型：使用相同的可靠性测试方法针对不同软件竞品进行可靠性测试，检测不同竞品的可靠性指标差异；
  - 2) 软件性能选型：使用相同硬件环境和相同的性能测试方法针对不同的软件竞品进行性能测试，检测不同竞品的性能指标差异；
  - 3) 安全选型：使用相同的安全测试方法针对不同的软件竞品进行安全测试，检测不同竞品的安全指标差异；
  - 4) 软件功能选型：根据产品需求对软件功能进行功能测试，检测不同竞品在满足相同需求条件下功能实现的差异性。
- c) 测试技术
- 根据选型目的选择一种或多种测试方法进行选型测试，具体测试方法见 4.3.1、4.3.2、4.3.3、4.3.4。

#### 4.3.11 选择原则

行业机构根据软件产品可能涉及的测试内容需选择适合的测试类型，具体选择原则见表3，其中【类别】不可裁剪，【内容】可根据行业特点进行裁剪，相关的测试内容宜使用画圈的测试类型。

表3 各测试类型的测试内容

类别	内容	测试类型									
		功能测试	性能测试	可靠性测试	安全性测试	可移植性测试	可维护性测试	应急演练	联网测试	全市场测试	选型测试
基础架构	通信线路	○	○	○	○					○	○
	网络结构	○	○	○	○						○
	主机硬件	○	○	○	○	○	○				○
	操作系统及编译环境	○	○	○	○	○	○				○
	存储设备	○	○	○	○	○	○				○
	灾备架构	○	○	○			○	○			○
应用软件	数据交换接口	○	○	○	○				○	○	○
	数据存储方式	○	○	○	○						○
	应用程序	○	○	○	○	○	○				○
	数据库等第三方应用	○	○	○	○	○	○				○
业务规则	根据实际业务情况填写，如交易规则、清算业务	○								○	
上线运维	上线步骤						○	○		○	
	应急方案						○	○			
	数据迁移	○						○		○	
	运维管理						○	○		○	
注：“○”表示选取											

#### 4.4 测试级别

##### 4.4.1 工作目标

根据软件生命周期和测试周期自然形成的阶段，测试级别包含单元测试、集成测试、系统测试、系统集成测试、验收测试。

待上线系统应经过五个测试级别及全市场测试之后，才能达到上线要求，允许上线。

每个测试级别的测试应根据4.2的要求展开相关工作。

各测试级别的工作目标见表4。

表4 测试级别的工作目标

测试级别	目标
单元测试	验证程序及模块内部逻辑符合系统设计说明书，又称为模块测试，是动态测试的最早期阶段。



表4 测试级别的工作目标（续）

测试级别	目标
集成测试	将应用内部的模块组合在一起进行测试，测试接口，各类集成以确保接口数据不会丢失，一个模块的功能不会影响其他模块的功能，各模块组合起来能达到预期要求等。
系统测试	针对单一被测系统以及系统内部组件间接口进行的测试，通过功能测试和非功能测试相结合的方式验证该系统应正确运行、满足项目需求规格说明书，从而确保系统功能及操作性上的成熟度和健康度。
系统集成测试	针对多个被测系统之间数据交互以及系统间接口进行的测试，验证整个系统承载的业务流程实现正确及系统整体与外界的硬件、网络间合作运行正确，从而确保整个系统在生产环境中的运行可达到客户预期。
验收测试	在模拟的客户环境中，验证产品能满足客户的需求。

#### 4.4.2 选择原则

根据行业机构角色及软件产品来源的不同，明确各机构的测试职责范围，以指导行业机构履行职责、把控软件质量，具体如下：

a) 市场核心机构与市场经营机构：

- 1) 对于完全外购的软件产品，因其需经过配置、联调才可与其他系统整体协同工作，故应对外购软件产品进行系统集成测试及验收测试；
- 2) 对于部分模块或组件为外购的软件产品，因其外购模块或组件需与自主研发或其他产品集成，故应对外购模块或组件进行集成测试、系统测试、系统集成测试及验收测试；
- 3) 对于完全自主研发的软件产品，应进行单元测试、集成测试、系统测试及验收测试；
- 4) 对于接入期货公司信息系统的外部信息系统，期货公司要对其进行必要的系统集成测试及验收测试，保证外部信息系统的合规性和安全性。

b) 市场服务机构：对于向行业提供的软件产品，应至少进行单元测试、集成测试及系统测试。

各证券期货业机构可委托第三方测试机构完成所需的验收测试工作，但是应遵循以下原则：

- 1) 市场服务机构不能代替用户对本机构研发或代理销售的软件产品进行验收测试。
- 2) 对于完全外购或部分外购软件产品的验收测试应由用户或其委托的第三方测试机构完成。
- 3) 第三方测试机构的测试过程均需符合本标准要求。
- 4) 市场监管机构可依据本标准要求，监督、检查相关机构执行情况。

测试级别和测试类型的对应关系见表5。

表5 测试级别与测试类型的对应关系

测试类型	测试级别					备注
	单元测试	集成测试	系统测试	系统集成测试	验收测试	
功能测试	○	○	○	○	○	
性能测试	○	○	○		○	
可靠性测试	○	○	○		○	
安全性测试	○		○		○	

表 5 测试级别与测试类型的对应关系（续）

测试类型	测试级别					备注
	单元测试	集成测试	系统测试	系统集成测试	验收测试	
可移植性测试			○		○	
可维护性测试			○		○	
应急演练			○		○	
联网测试				○	○	
全市场测试						验收测试完成后，进入上线准备阶段，应和系统的关联方进行全市场测试。在采购前期或采购过程中实施，不对应任何测试级别。

各个测试级别的具体描述见5～9章。

#### 4.5 测试准入/准出要求

##### 4.5.1 准入要求

在测试执行前，应达到准入要求后，才可展开测试执行。测试准入要求如下：

- 上一个测试级别达到准出要求；
- 测试计划已经通过审核；
- 测试用例设计完成并通过审阅；
- 测试用例已完全 100%覆盖测试需求；
- 阻碍进一步测试的问题已经解决，如冒烟测试通过；
- 测试环境（软件、硬件）搭建完成、测试工具已安装配置完成，发布的程序代码已经被安装到测试环境中。

##### 4.5.2 准出要求

测试执行阶段的准出将标志着整个测试实施过程的结束，故该阶段的准出要求主要关注以下三个方面：

- 用例执行情况；
- 缺陷解决情况；
- 系统质量情况。

具体的测试准出要求见表6，可根据不同的测试级别进行调整。

表 6 测试执行准出要求

测试类型	准出要求
功能测试	a) 计划实施的测试范围 100%覆盖； b) 计划执行的测试用例 100%执行完成，未执行的用例已写明原因； c) 所有提交的缺陷都被记录并有最终状态，测试执行结束后仍未修复完成的缺陷作为遗留缺陷在测试报告中记录说明未修复原因及修复计划； d) 缺陷整体呈收敛；经过讨论确认应修改的缺陷均已修复。

表 6 测试执行准出要求（续）

测试类型	准出要求
性能测试	a) 测试执行情况：可执行场景 100%执行完毕，未执行场景均有相关说明并评审通过； b) 缺陷解决情况：无非最终状态缺陷； c) 监控指标情况：监控并收集了用户关注的关键的资源指标和系统指标，其中资源指标包括 CPU、内存、I/O、带宽等，系统指标包括并发用户数、响应时间、事务成功率、超时错误率等； d) 测试报告情况：清晰的评价系统当前性能，判断系统满足预期性能需求的情况；指出系统可能存在的性能问题，定位性能瓶颈；判定系统性能表现，预见系统负载压力承受力； e) 测试工具情况：使用既有工具的，需确保测试脚本在当前系统版本条件下可反复稳定运行；自行研发工具的，需确保测试工具至少可产生系统最大承载力 1 倍及以上的压力，在常规压力条件下至少可稳定加压 1 小时以上，至少具备响应时间和事务成功率等指标的监控能力。
可靠性测试	a) 测试执行情况：可执行的测试场景全部执行完毕，未执行场景均有相关说明并评审通过； b) 缺陷解决情况：无非最终状态缺陷； c) 系统及组件的容错、恢复能力和成熟性指标满足需求； d) 系统灾备处理能力满足需求。
安全性测试	a) 安全性测试整体要求：应完成代码级安全漏洞扫描、系统级安全漏洞扫描，可选完成拒绝服务式攻击测试； b) 代码级安全漏洞扫描：至少覆盖 OWASP Top 10、CWE、SANS 等安全静态标准，并给出不合规代码的修复建议； c) 系统级安全漏洞扫描：至少覆盖了 CVE 已发现的安全漏洞以及隐患漏洞，对于关键性的严重级别漏洞应通过渗透测试验证； d) 拒绝服务式攻击测试：至少完成资源比拼型（ICMP Flood、ACK Flood、Connection Flood）、软件缺陷利用型（SYN Flood、ARP Spoof、Ping of Death）等类型的拒绝服务式攻击测试，并针对被测系统表现情况给出改善建议。
可移植性测试	a) 测试执行情况：可执行的测试场景全部执行完毕，未执行场景均有相关说明并评审通过； b) 缺陷解决情况：无非最终状态缺陷； c) 软件的安装/升级程序、安装/升级文档齐全，按照安装文档的指导能正确安装/升级程序； d) 软件能在其支持的所有硬件平台、操作系统、网络环境中安装/升级、使用。
可维护性测试	a) 软件系统的需求、设计等过程文档清晰规范，不存在引起歧义的描述；相应的专家评审意见均需确认且不存在遗留未修改的问题； b) 软件的代码实现规范，注释描述清晰，注释率不低于 20%，函数、类、全局变量以及逻辑复杂的代码段、算法等应有相应注释； c) 每个函数最大代码行不超过 1000 行（非空、非注释行）；圈复杂度不超 10，如有必要超过的情况应经相关专家评审通过。
应急演练	a) 应急方案演练成功； b) 未发现重大缺陷或发现缺陷已经处理完成。
联网测试	a) 接口联网测试成功； b) 未发现重大缺陷或发现缺陷已经处理完成。
全市场测试	a) 全市场联网测试成功； b) 未发现重大缺陷或发现缺陷已经处理完成。

表 6 测试执行准出要求（续）

测试类型	准出要求
选型测试	a) 测试执行情况：所有测试场景全部执行完毕，每个测试场景在不同备选竞品上均执行完毕且不存在测试场景只运行在某个或某几个备选竞品的情况，未执行场景均有相关说明并评审通过； b) 测试报告情况：竞品评定指标数据均采集完成，未填写项均有相关说明并评审通过；清晰地阐述竞品指标执行步骤和指标结果，对相同指标项应明确写出不同备选竞品的测试数据差异。

#### 4.6 测试管理

##### 4.6.1 角色职责

从测试级别和测试类型的角度，描述在相关测试活动中的责任主体和参与者，如表7、表8所示。

表 7 测试级别参与团队

测试级别	责任主体
单元测试	开发团队。
集成测试	
系统测试	测试团队。
系统集成测试	
验收测试	需求提出方、市场服务机构。

表 8 测试类型参与团队

测试类型	责任主体	参与者
功能测试	需求提出方、开发团队、测试团队。	需求提出方、开发团队、测试团队、运维团队及供应商等干系人。
性能测试		
可靠性测试		
安全性测试		
可移植性测试		
可维护性测试		
应急演练	需求提出方、运维团队。	测试发起方或组织方应根据实际的业务需求和测试需求选择必要的关联方参与测试。关联方主要包括但不限于市场核心机构、市场经营机构、市场服务机构等市场各参与方。
联网测试		
全市场测试		
选型测试	需求提出方、运维团队。	需求提出方、开发团队、测试团队、运维团队及供应商等干系人。

参与测试的团队中，各角色的职责如表9所示。

表 9 各角色的职责

团队	角色	职责
需求提出方	客户	提出业务需求，对系统需求规格说明书进行确认，对出现分歧的业务需求进行确认，完成验收测试。

表9 各角色的职责（续）

团队	角色	职责
开发团队	项目经理	制定项目整体日程安排，协助测试环境准备。
	需求分析人员	对业务需求进行分析并完成可测的系统需求规格说明书。
	开发人员	根据业务需求文件及系统设计文档进行编码；对代码引起的缺陷进行分析及修正等。
测试团队	测试经理	组建和管理测试团队，把控项目测试整体质量、进度、资源/成本、风险，协调过程中各种问题等。
	测试人员	参与业务需求调研，对系统需求规格说明书进行静态测试（见附录D）；测试需求的编写和维护；测试用例编写、执行、记录测试结果、完成对发现的缺陷进行提交、验证等；负责测试环境部署与维护；参与测试框架设计，分析测试工具需求，完成测试工具开发和维护。
支持团队	配置管理人员	制定配置管理计划、配置库，进行配置管理； 对由修正缺陷所引起的应用系统/文档的变更等进行版本控制及发布； 对其他原因引起的需求变更、技术变更等进行配置管理。
	质量保证人员	策划质量保证工作，参考组织级要求对项目过程和工作产品进行评审和审计，跟踪发现不符合问题直至关闭。
运维团队	运维经理	协调和安排整个项目的运维工作； 负责项目非功能需求的提出和验收等； 负责项目上线计划编写及与相关方沟通上线相关工作。
注：开发团队和测试团队需为相互独立的两个团队，在项目实施过程中需保证两个团队人员的分离。		

#### 4.6.2 质量管理

在测试项目实施过程中，应建立相应的测试质量管理体系，对测试目标、流程、各角色的职责以及实施过程进行规范和监控，保证测试项目实施过程中的文档、过程的规范性，最终保证测试项目的质量。

测试项目质量控制管理包括测试过程控制管理，质量，成本，进度（QCD）数据监控等。质量保证工作涉及测试项目实施各个阶段的活动，在测试项目实施过程中，通过独立的第三方质量保证负责人对测试项目过程质量进行监控，客观地检查和监控“过程及产品的质量”。

测试项目各测试阶段的输出文档主要包括：测试估计、测试计划、测试需求、测试用例/方案、测试数据/脚本、缺陷列表、测试结果及测试报告。在测试项目过程中，通过测试团队内、外专家或者客户评审控制各文档的质量，保证文档内容的可行性、可操作性、完整性、准确性，并保留相关的评审记录。

在测试项目实施的过程中，通过定期例会、周月报等机制，测试经理向所有干系人汇报测试进展情况，对于出现的异常情况，重点分析并制定纠正措施；并监控缺陷的状态、趋势，跟踪缺陷消除措施的制定并跟踪措施的执行结果。

#### 4.6.3 进度管理

依据测试计划进行进度跟踪与监控，对测试任务进行分析，找到关键路径，并分析此关键路径上的关键点，控制好这些关键点，保证任务最终达成。

在进度管理过程中向相关方及时汇报进度情况：

- a) 定期汇报
  - 1) 测试人员定期向测试经理汇报任务完成的进度情况，并更新进度管理表；
  - 2) 测试经理定期根据测试项目整体的进度情况，整理定期报告，标注每项计划任务的进行状态、完成百分比、计划完成日期、后续计划安排，以及测试项目的风险和问题，通过邮件等方式汇报给项目干系人。
- b) 管理者报告  
定期向项目管理者汇报测试项目的进度、风险、问题以及应对解决措施，管理者在测试过程中推进过程中对重大问题和风险进行决策、解决项目难题、协调项目资源，监控测试项目运行状况。
- c) 里程碑报告  
在测试项目的里程碑或关键点时，可组织测试项目相关人员进行会议方式沟通，主要汇报测试项目进度、完成情况及风险问题等，并将会议记录发送给测试项目相关人员。

#### 4.6.4 资源管理

在测试计划阶段根据资源管理要求，识别并管理测试过程中所需的人员资源、软/硬件资源和测试环境，宜包括下面几项工作：

- a) 在测试计划中策划测试团队成员数量、软件和硬件的需求，过程中跟踪各项资源的变动情况；
- b) 定期评价测试团队成员的知识和技能，必要时安排培训；
- c) 定期跟踪测试项目软件及硬件环境的变化，评价环境的变化对测试项目的影响；
- d) 当发现问题时，及时协调并跟踪解决。

#### 4.6.5 风险管理

风险管理的目的是避免或降低由于风险给组织和测试项目带来的损失。

- a) 风险管理流程  
风险管理流程包含风险识别、风险分析、制定措施和风险跟踪，具体如下：
  - 1) 风险识别：在进行风险识别前应尽可能收集测试项目的相关资料，测试经理组织测试团队成员和其他利益共同者参与风险识别活动，风险识别活动贯穿测试项目全生命周期；
  - 2) 风险分析：对识别出来的风险进行分析，包括风险的类型、严重性、可能性及生存期，确定风险优先级；
  - 3) 制定措施：依据风险优先级，制定风险的缓解措施、应急措施及跟踪频率；
  - 4) 风险跟踪：依据风险的跟踪频率，定期对风险进行跟踪，直至其发生或消失。
- b) 风险分类  
根据风险的影响内容区分风险分类，具体分类见表 10。

表 10 风险分类

类别	影响
质量	该类风险主要影响测试项目的产品质量。
工作量	该类风险主要影响测试项目的工作量和成本。
进度	该类风险主要影响测试项目的进度。
所有	该类风险对进度、工作量、质量都有影响。

- c) 风险优先级确定方法  
风险优先级和确定方法如下：

- 1) 严重性：代表风险一旦发生对测试项目造成的威胁程度，严重性分为“致命”、“严重”、“一般”、“轻微”四个级别；
- 2) 可能性：风险发生的概率，分为“高”、“中”、“低”三个级别；
- 3) 优先级：风险的优先级，分为“A”、“B”、“C”、“D”四个级别；
- 4) 风险优先级确定方法见表 11。

表 11 风险优先级确定方法

严重性	可能性		
	高	中	低
致命	A	B	C
严重	B	C	D
一般	C	C	D
轻微	D	D	D

d) 风险管理要求见表 12。

表 12 风险管理要求

风险优先级	是否应制定风险缓解措施	是否应制定风险应急方案	跟踪频率
A	是	是	每天或者每周
B	是	是	每周
C	是	否	每双周
D	否	否	每月

5 单元测试

5.1 测试对象

单元测试是针对软件设计的最小单位——程序模块进行的测试，在面向对象的软件开发中，以 Class（类）作为测试的最小单元；在结构化的软件开发中，以模块（函数、过程）作为测试的最小单元。

“单元”具有明确的功能、规格说明、与其他部分的明确的接口说明，可清晰地与同一程序的其他单元划分开来。

在概要设计中应进行识别。

5.2 测试目的

能发现各模块内部可能存在的各种错误，保证模块功能符合设计需求。由于单元测试是动态测试的最早期阶段，所以有助于早期识别和修复缺陷，消除单元模块的不确定性。

5.3 测试内容/关注点

由编码人员在实现相应的模块后进行测试执行。

测试内容及关注点应按照GB/T 15532-2008，5.4。

5.4 测试方法

测试者应依据详细设计说明书和源程序清单，了解该模块的I/O条件和模块的逻辑结构。  
主要采用白盒测试技术，辅助于黑盒测试技术。

### 5.5 测试准入/准出要求

测试准入要求见4.5.1，准出要求在4.5.2基础上，应根据附录E中业务分类，A类语句覆盖率宜达到100%；B类、C类语句覆盖宜达到90%。

## 6 集成测试

### 6.1 测试对象

待测试系统的模块间接口；  
软件单元集成到软件系统的组装过程。

### 6.2 测试目的

验证各模块之间、模块与已集成的软件系统之间的接口关系，保证将所有模块按照设计要求组装成为系统后能正常运行。

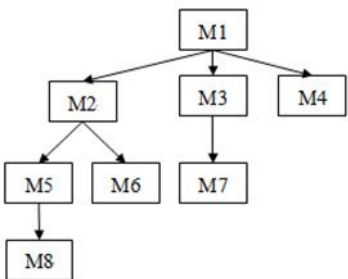
### 6.3 测试内容/关注点

测试各模块间接口，保证接口数据传递正常，接口数据不会丢失。  
一个模块的功能不会影响到其他模块的功能。  
各模块组合起来能达到预期要求的父功能。  
分析单个模块的误差累积起来的放大情况，应在可接受的程度内。

### 6.4 测试方法

测试者应依据概要设计说明书和系统设计说明书，了解各模块的接口关系。  
主要采用灰盒测试技术。  
有三种集成测试方法，分别如下：

a) 自顶向下的方法（如图1）



**策略一：**按照软件的控制层次结构，以深度优先的策略（首先确认一条主控路径）

示例：首先把M1、M2、M5和M8集成在一起，再将M6集成起来，然后考虑中间和右边的路径。

**策略二：**按照软件的控制层次结构，以广度优先的策略（水平地向下移动）

示例：首先把M1、M2、M3和M4与主控模块集成在一起，再将M5和M6和其他模块集成起来。

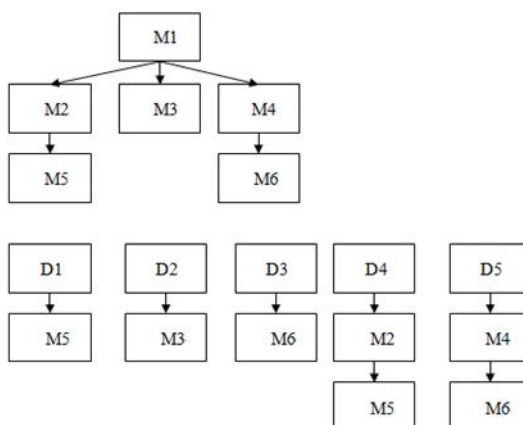
**分析：**



- 1) 这种组装方式将模块按软件的控制层次结构，从主控模块开始，以深度优先或广度优先的策略，逐步把各个模块集成在一起；
- 2) 优点：在测试过程中可首先实现和验证一个完整的软件功能，能较早地发现主要控制和决策中的缺陷；
- 3) 缺点：**桩模块**——用以代替被测模块调用的子模块；
- 4) 很难维护一个纯粹的自顶向下的策略。

图 1 集成测试方法-自顶向下

b) 自底向上的方法（如图 2）



分析：

- 1) 这种组装的方式是从程序模块结构的最底层的模块开始组装和测试；
- 2) 优点：测试用例的设计相对简单；
- 3) 缺点：软件的整体形象到最后才会形成；
- 4) 驱动模块——相当于被测模块的主程序。它接收测试数据，把这些数据传送给被测模块，最后再输出测试结果；
- 5) 这是一种更本能的测试方法。

图 2 集成测试方法-自底向上

c) 一次性组装方法

一次性组装方法分析如下：

- 1) 它是一种非增殖式组装方式。把所有通过测试的模块组装在一起进行测试，最终得到要求的系统。是最普通和效果最差的方法；
- 2) 如果每个模块只进行草率的测试，整个系统发生灾难的可能性将是巨大的。系统越大这种可能性越大；
- 3) 对于高质量的简单的系统来说，这种方法就很有意义。

对于高质量的简单的系统来说，通常采用一次性的集成方法，快捷有效；对于相对复杂的系统来说，应根据软件的特点和工程的进度，选用适当的测试策略，可混合使用两种策略，上层模块用自顶向下的方法，下层模块用自底向上的方法。

## 6.5 测试准入/准出要求

测试准入要求见4.5.1，准出要求在4.5.2基础上，系统组件单元之间的所有调用宜达到100%测试覆盖率。

## 7 系统测试

### 7.1 测试对象

系统此时以完整的形象展现出来，针对整个产品系统进行测试。

### 7.2 测试目的

验证系统满足系统需求规格说明书，找出与系统需求规格说明书不符或与之矛盾的地方；保证系统功能及非功能按照系统需求规格说明书要求工作。

### 7.3 测试内容/关注点功能测试

验证被测试系统符合系统需求规格说明书规定的每一项功能、校验。  
详细内容见4.3.1。

#### 7.3.2 性能测试

验证被测试系统符合系统需求规格说明书规定的性能指标。  
详细内容见4.3.2。

#### 7.3.3 可靠性测试

验证被测试系统符合系统需求规格说明书规定的可靠性指标。  
详细内容见4.3.3。

#### 7.3.4 安全性测试

验证被测试系统符合系统安全性要求。  
详细内容见4.3.4。

#### 7.3.5 可移植性测试

验证被测试系统符合系统需求规格说明书规定的可移植性指标；  
详细内容见4.3.5。

#### 7.3.6 可维护性测试

验证被测试系统符合系统需求规格说明书规定的可维护性指标。  
详细内容见4.3.6。

#### 7.3.7 应急演练

验证被测试系统符合系统需求规格说明书规定的应急演练要求。  
详细内容见4.3.7。

## 7.4 测试方法

测试者应依据系统需求规格说明书。  
主要采用黑盒测试、灰盒测试技术。

## 7.5 测试准入/准出要求

测试准入要求在4.5.1基础上，应根据附录E中的业务分类，确保在测试设计方面符合以下要求：

- a) 针对A类业务，应充分识别可能产生的输出结果以及影响输出结果的各种输入条件，全面梳理和校验输入条件及输出结果的逻辑关系，同时应覆盖全部业务场景下被测系统的数据流及业务流并进行校验。
- b) 针对B类业务，应充分识别可能产生的输出结果以及影响输出结果的各种输入条件，梳理和校验输入条件及输出结果的关键逻辑关系，同时应覆盖关键业务场景下被测系统的数据流及业务流并进行校验。
- c) 针对C类业务，应覆盖关键业务场景下被测系统的数据流及业务流并进行校验。
- d) 测试设计应考虑正向和反向用例、边界值场景。
- e) 新增/修改功能应重点关注功能实现正确性，包括正常及异常处理等，还应关注对非功能的影响；回归范围主要根据被测系统的修改方案从技术和业务层面进行影响分析、确定回归范围。

准出要求在4.5.2基础上应确保测试用例至少执行一轮，并且对测试执行关键步骤留痕，便于确认、核对测试结果。

## 8 系统集成测试测试对象

### 系统间的接口及数据传输。

### 8.2 测试目的

对由多个系统构成的整体系统进行验证，确保系统间业务交互、数据联通的正确性和完整性，宜在类生产环境中或者与生产环境有可比性的环境中进行。

关键缺陷被识别并修复，避免在用户验收测试阶段进行昂贵的返工。

### 8.3 测试内容/关注点

测试系统间接口，保证系统间接口数据传递正常、不会丢失。

一个系统的功能不会影响其他系统的功能。

各系统组合起来能达到系统整体目标。

单个系统的误差累积起来会放大，从而达到不能接受的程度。

### 8.4 测试方法

测试者应依据系统架构说明书。

主要采用黑盒测试、灰盒测试技术。

### 8.5 测试准入/准出要求

测试准入要求在4.5.1基础上，确保在测试设计方面符合以下要求：

- a) 系统之间数据接口的测试设计应覆盖所有接口及接口间传送的数据类型和数据流向。
- b) 梳理系统间业务依赖关系，覆盖相关业务场景下被测系统的数据流及业务流并进行校验。
- c) 测试设计应考虑正向和反向用例、边界值场景。

准出要求在4.5.2基础上应确保测试用例至少执行一轮。

## 9 验收测试测试对象

验收测试的参与者应是用户、需求提出方或其他授权人员，验收测试人员应具备很强的业务或技术背景，他们从业务及技术需求的角度，决定是否接受这个系统/产品，依据技术指标对软件功能、性能、可靠性、安全性等方面进行验收测试。

验收测试团队需区别于项目开发团队、系统测试团队及运维团队，验收测试可委托第三方测试机构完成，遵循的原则见4.4.2。

## 9.2 测试目的

有助于确保业务及技术需求得到满足，确保软件产品或者产品组件在用户环境下，实现了用户的需求。

## 9.3 测试内容/关注点功能测试

确认被测试系统符合客户业务需求规定的每一项功能、校验。  
详细内容见4.3.1。

### 9.3.2 性能测试

确认被测试系统符合客户业务需求规定的性能指标。  
详细内容见4.3.2。

### 9.3.3 可靠性测试

确认被测试系统符合客户业务需求规定的可靠性指标。  
详细内容见4.3.3。

### 9.3.4 安全性测试

确认被测试系统符合系统安全性要求。  
详细内容见4.3.4。

### 9.3.5 可移植性测试

确认被测试系统符合客户业务需求规定的可移植指标。  
详细内容见4.3.5。

### 9.3.6 可维护性测试

确认被测试系统符合客户业务需求规定的可维护性指标。  
详细内容见4.3.6。

### 9.3.7 应急演练

确认被测试系统符合客户业务需求规定的应急演练要求。  
详细内容见4.3.7。

### 9.3.8 联网测试

确认被测试系统符合客户业务需求规定的联网测试要求。  
详细内容见4.3.8。

## 9.4 测试方法

测试者应依据客户业务需求。  
主要采用黑盒测试技术。

#### 9.5 测试准入/准出要求

测试准入要求在4.5.1基础上，确保在测试设计方面符合以下要求：

- a) 明确被测系统上线使用时的业务场景，并将受测试数据或其他因素影响可能存在的业务场景分支进一步拆分，形成完整的业务场景库，业务场景应覆盖到关键处理流程分支，包括正常、异常以及系统特殊处理。
  - b) 针对各业务场景，应充分识别和校验该业务场景所涉及的功能操作、测试数据及逻辑关系。
- 准出要求在4.5.2基础上应确保测试用例至少执行一轮。

## 附录 A

### (规范性附录)

### 测试流程

#### A.1 测试估计

##### 目标:

估算测试的工作量，预分配测试资源及测试活动时间。

##### 作业活动:

- a) 根据项目整体日程规划及应用系统质量要求选择适当的测试级别（见 4.4）；
- b) 分析业务需求，估计测试工作量；
- c) 根据项目整体日程规划及测试工作量估计结果，制定项目测试里程碑、周期；
- d) 对测试工作量估计书，项目测试里程碑、周期进行评审。

输入输出（见表A.1）：

表 A.1 输入输出-测试估计

活动	输入	输出
测试估计	-业务需求 -系统质量要求 -项目整体日程规划	-测试工作量估计书 -项目测试里程碑、周期

#### A.2 测试计划

##### 目标:

- a) 策划所有测试活动，便于后续监控与跟踪；
- b) 在项目初期尽早确定测试策略。

##### 作业活动:

- a) 对业务需求进行进一步分析，拟定测试计划；
- b) 对测试计划进行评审。

输入输出（见表A.2）：

表 A.2 输入输出-测试计划

活动	输入	输出
测试计划	-业务需求 -项目整体日程规划及里程碑	-测试计划（或称测试方案）

#### A.3 测试设计

##### A.3.1 测试需求分析

##### 目标:

- a) 以业务需求、系统需求规格说明书等所有与被测功能相关的资料为基础，以系统设计、缺陷修复方案等相关文档为参考，确定应测试的内容、测试方法，形成完整、准确、清晰的测试需求；
- b) 测试需求 100%覆盖业务需求及系统需求规格说明书。

**作业活动：**

- a) 根据业务需求及系统需求规格说明书，进行测试需求分析；
- b) 根据分析结果，进行测试需求整理；
- c) 对测试需求进行评审；
- d) 可根据实际需要进行裁剪，如无此活动，可直接进行测试用例设计。

**输入输出**（见表A.3）：

**表 A.3 输入输出-测试需求分析**

活动	输入	输出
测试设计	-业务需求 -系统需求规格说明书	-测试需求

### A.3.2 测试用例设计

**目标：**

- a) 依据测试需求或系统需求规格说明书，进行测试设计，要求明确测试数据需求，为测试执行做好准备；
- b) 测试用例 100%覆盖需求。

**作业活动：**

- a) 有测试需求分析活动时，根据测试需求撰写测试用例；
- b) 无测试需求分析活动时，根据系统需求规格说明书撰写测试用例；
- c) 对测试用例进行评审。

**输入输出**（见表A.4）：

**表 A.4 输入输出-测试用例设计**

活动	输入	输出
测试设计	-测试需求 -系统需求规格说明书	-测试用例
测试用例设计原则及要素见[GB/T 15532-2008, 4.5]		

### A.3.3 测试数据

**目标：**

为测试执行准备测试数据，便于测试执行快速、有效开展。

**作业活动：**

- a) 依据测试需求及测试用例要求，准备测试数据；
- b) 验证测试数据；
- c) 可根据实际需要进行裁剪。

**输入输出**（见表A.5）：

表 A.5 输入输出-测试数据

活动	输入	输出
测试设计	-测试需求 -测试用例 -原始数据	-测试数据

## A.3.4 测试环境

## 目标：

- a) 确保测试环境与开发环境分离；
- b) 依据测试计划中的测试环境搭建策略，部署、维护测试环境。

## 作业活动：

- a) 依据测试环境搭建策略，申请搭建测试环境需要的软件、硬件；
- b) 根据部署要求，部署测试环境，并在测试过程中，维护测试环境，确保系统测试环境可用。

## A.4 测试执行

## 目标：

- a) 根据测试计划，按照测试用例描述在测试环境中完成执行操作；
- b) 尽可能多地发现缺陷，并进行缺陷管理。

## 作业活动：

- a) 执行测试用例，并记录测试结果；
- b) 分析测试结果，提交问题/缺陷（缺陷管理见附录 F）；
- c) 对问题/缺陷进行分析、处理、验证；
- d) 过程中若发生需求变更，或者发现用例设计不足之处，应更新、完善测试需求和测试用例；
- e) 执行结束后，对测试通过的版本做好记录。

输入输出（见表A.6）：

表 A.6 输入输出-测试执行

活动	输入	输出
测试执行	-测试计划 -测试用例 -测试数据	-测试结果 -缺陷列表 -测试通过的版本

## A.5 测试报告

## 目标：

总结测试过程，为下一测试级别的测试过程改进提供参考。

## 作业活动：

- a) 对测试过程（包括测试环境部署情况、功能/非功能测试执行情况、测试轮次/内容、测试用例汇总等）、测试结果及问题/缺陷进行收集、整理、分析，得出测试结论，编写测试总结报告；
- b) 遗留缺陷已经给出后续处理计划；



c) 对测试报告进行评审。

输入输出（见表A.7）。

表 A.7 输入输出-测试报告

活动	输入	输出
测试报告	-测试过程 -测试结果 -缺陷列表	-测试总结报告

附 录 B  
(资料性附录)  
软件测试模板

B.1 测试工作量估计书

测试工作量估计书如图 B.1 所示。

测试工作量估计																		
序号	测试级别	系统分类	系统	模块/功能		系统类型/功能特点	功能类型	回归力度	标准执行轮次	计划执行轮次	设计复杂度	执行复杂度	需求分析与设计静态测试(人日)	预估用例数(个)	预估测试设计工作量(人日)	测试数据构造工作量(人日)	预估测试执行工作量(人日)	工作量合计(人日)

图 B.1 测试工作量估计

B.2 测试计划

测试计划如图 B.2 所示。

1	简介
1.1	项目概述
1.2	基本假设
1.3	参考资料
2	测试目标
3	测试范围
4	测试策略
4.1	测试级别
4.2	业务功能
4.3	非功能
4.4	测试依据版本
4.5	风险评估
4.6	测试方法
5	测试计划
5.1	角色及职责
5.2	测试工作里程碑
5.3	测试组人员需求
6	测试环境
6.1	测试系统
6.2	测试数据
6.3	测试工具
7	测试管理与报告程序
7.1	测试管理
7.1.1	问题追踪管理流程
7.1.2	变更管理流程
7.1.3	进度追踪流程
7.2	测试报告

图 B.2 测试计划

B.3 测试需求

测试需求如图 B.3 所示。

交易类型	一级功能	二级功能	三级功能	验证点	验证方式	函数/存储过程	数据库表	备注
期货				从业务角度描述各功能需要验证关注点和重点	从系统角度描述如何操作验证该功能的业务关注点，若有多种验证方式会选择易于验证的一种或全部描述	从代码角度列出该功能实现所涉及的函数/存储过程	从数据表角度列出该功能相关的内存/物理数据库表	
期权								
期货/期权								

图 B.3 测试需求

B.4 测试用例

测试用例如图 B.4 所示。

项目名称				测试用例总数										
测试功能描述				通过个数										
测试用例版本				失败个数										
准备人				未执行个数										
准备日期														
参考														
前提条件 / 测试设置：														
1.														
2.														
测试用例 ID	需求对应	优先级	前提条件	测试目的描述	测试步骤概述	测试数据	期望结果	实际结果	Pass/Fail/NT	Build 版本	缺陷 ID	测试人员	测试日期	备注

图 B.4 测试用例

B.5 测试缺陷列表

测试缺陷列表如图 B.5 所示。

缺陷ID	缺陷标题	缺陷描述	测试轮次	发现日期	识别时机	测试类型	交易类型	模块	缺陷发现版本	缺陷负责人	缺陷提交人	缺陷状态	是否核心缺陷	严重度	业务领域项	优先级	子系统	根本原因	解决方案描述	可测版本	缺陷关闭版本	关闭日期

图 B.5 测试缺陷列表

B.6 评审记录

评审记录如图 B.6 所示。

评审内容									
评审日期		文档版本					文档页数		
项目阶段		准备阶段		工作量（人时）			总工时（人时）		
评审负责人				问题个数			缺陷总数		
记录人		正式		工作量（人时）			问题总数		
评审人员		评审时		问题个数			剩余问题总数		
		准备阶段工时投入比率					评审效率（页/人时）		
		准备阶段问题发现比率					文档缺陷密度（个/页）		
NO.	问题位置	问题描述	提出者	是否项目外专家	提出时期	问题类型	解决措施	责任人	状态

图 B.6 评审记录

B.7 测试总结报告

测试总结报告如图 B.7 所示。

XXXX 项目

测试总结报告

图 B.7 测试总结报告

1 简介
1.1 概述
1.2 涉及范围
2 测试结论
3 测试执行汇总
3.1 环境部署
3.1.1 功能测试
3.1.2 非功能测试
3.1.2.1 系统框架图
3.1.2.2 环境部署图
3.1.2.3 软/硬件环境
3.2 功能测试情况概述
3.2.1 冒烟测试情况
3.2.2 测试执行情况
3.2.3 测试用例汇总
3.3 非功能测试情况概述
3.3.1 测试版本说明
3.3.2 测试执行情况
3.3.2.1 代码检测与分析
3.3.2.1.1 静态检测
3.3.2.1.2 动态检测
3.3.2.2 可靠性
3.3.2.2.1 测试目标完成情况
3.3.2.2.2 测试技术及方法总结
3.3.2.2.3 测试场景统计
3.3.2.3 性能
3.3.2.3.1 测试目标完成情况
3.3.2.3.2 测试技术及方法总结
3.3.2.3.3 测试场景统计
3.3.2.4 可移植性
3.3.2.4.1 测试目标完成情况
3.3.2.4.2 测试技术及方法总结
3.3.2.4.3 测试场景统计
3.3.2.5 可维护性
3.3.2.5.1 测试目标完成情况
3.3.2.5.2 测试技术及方法总结
3.3.2.5.3 测试场景统计
3.3.2.6 安全性
3.3.2.6.1 测试目标完成情况
3.3.2.6.2 测试技术及方法总结
3.3.2.6.3 测试场景统计
3.3.2.7 生产流水反演

图B.7 测试总结报告（续）



#### 4 测试结果汇总

##### 4.1 功能测试结果

###### 4.1.1 测试用例执行结果

###### 4.1.2 测试用例执行详情

##### 4.2 非功能测试结果

###### 4.2.1 代码检测与分析结果

###### 4.2.2 可靠性测试结果

###### 4.2.3 性能测试结果

###### 4.2.4 可移植性测试结果

###### 4.2.5 可维护性测试结果

###### 4.2.6 安全性测试结果

###### 4.2.7 生产流水反演结果

##### 4.3 缺陷汇总

#### 5 测试结果分析

##### 5.1 缺陷统计与分析

###### 5.1.1 按缺陷严重程度

###### 5.1.1.1 冒烟测试

###### 5.1.1.2 正式执行

###### 5.1.2 按测试周期缺陷分布情况

###### 5.1.3 按测试版本缺陷分布情况

###### 5.1.4 按缺陷根本原因

###### 5.1.5 按模块缺陷分布情况

###### 5.1.6 按业务领域分布情况

###### 5.1.7 是否核心业务缺陷分析

###### 5.1.8 无效缺陷分析

##### 5.2 遗留问题及解决计划

#### 6 经验教训&改善建议

##### 6.1 经验教训

##### 6.2 改善建议

图 B.7 测试总结报告（续）

## B.8 管理者报告

管理者报告如图 B.8 所示。

进 度	当前阶段			延迟情况				
	详细进度							
成 本	活动类型	需求	设计	实现	测试	项目管理	其他	合计
	计划人月							
	实际人月							
	使用率实际/ 计划							
	偏差原因分析:							
质 量	活动类型	需求	架构设计	概要设计	系统测试	验收测试		
	缺陷密度							
	分析:							
风 险 与 问 题	风险描述			优先级		跟踪描述		

图 B.8 管理者报告

B.9 里程碑总结报告

里程碑总结报告如图 B.9 所示。

题 目	XXXX 项目 XX 里程碑总结报告		
日 期	年 月 日~ 年 月 日		
作 者			
发 送			
<b>1 里程碑情况概述</b>			
里程碑名称:			
计划开始日期		计划结束日期	
实际开始日期		实际结束日期	
里程碑任务:			
计划完成任务及工作产品		完成情况	
--备注: 如有必要进行项目计划变更, 请在此处记录原因。			

图B. 9 里程碑总结报告

**2.6 评审过程**

本里程碑过程性能分析：

活动	评审效率/ 评审规模	缺陷密度/缺 陷个数	异常情况分析
需求评审			是否正常。如果存在异常，分析原因及采取的针对性措施。
设计评审			
代码评审			
测试用例评审			

· 经验教训小结：有什么需要改进的、杜绝的、推广的。

**2.7 测试过程**

基于过程异常率的数据，分析测试过程性能。

**2.8 项目资源**

分析项目中使用的资源（人力、环境等）是否满足项目的要求。

**2.9 风险**

等级	总数	未关闭数	发生数	未关闭 or 发生风险的列举
A				
B				
C				
D				

· 风险措施效果的总结：

**2.10 成本**

对照项目计划总结项目成本的使用情况。

**2.11 质量保证**

说明检查项数量，不符合的数量和不符合问题的分布状况及延期解决状况等，分析不符合问题产生的原因。

**2.12 配置管理**

- 基线建立及升级、产品发布活动：
- 配置审计活动：

**3 下里程碑工作安排**

里程碑名称：

里程碑任务：

图 B.9 里程碑总结报告（续）

附 录 C  
(规范性附录)  
测试技术

## C.1 黑盒测试

### C.1.1 等价类划分

等价类划分是把所有可能的输入数据，即程序的输入域划分成若干部分，然后从每一部分中选取少数有代表性的数据作为测试用例。

等价类是指某个输入域的子集合，在该子集合中，各个输入数据对于揭露程序中的错误都是等效的，测试某等价类的代表值就等价于对这一类其他值的测试：

- a) 有效等价类：是指对于程序的规格说明来说，是合理的，有意义的输入数据构成的集合；
- b) 无效等价类：是指对于程序的规格说明来说，是不合理的，无意义的输入数据构成的集合。

划分等价类的原则：

- a) 如果输入条件规定了取值范围或值的个数，则可确立一个有效等价类和两个无效等价类所有可能的输入数据，例： $0 < A < 10$ ；
- b) 如果输入条件规定了输入值的集合或者规定了“应如何”的条件，这时可确立一个有效等价类和一个无效等价类，例： $A=1, 3, 5$  或者  $A > 10$ ；
- c) 如果输入条件是一个布尔量，则可确定一个有效等价类和一个无效等价类，例： $A=0$  or  $1$ ；
- d) 在规定了输入数据的一组值（假设  $n$  个），并且程序要对每个输入值分别进行处理的情况下，可确立  $n$  个有效的等价类和一个无效的等价类，例：CASE 语句；
- e) 如果规定了输入数据应遵守的规则，则可确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则），例：文本框中应输入数字；
- f) 在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价类进一步的划分为更小的等价类，例：三角形问题。

在确立了等价类后，可建立等价类表，列出所有划分出的等价类输入条件：有效等价类、无效等价类，然后从划分出的等价类中按以下三个原则设计测试用例：

- a) 为每一个等价类规定一个唯一的编号；
- b) 设计一个新的测试用例，使其尽可能多地覆盖尚未被覆盖的有效等价类，重复这一步，直到所有的有效等价类都被覆盖为止；
- c) 设计一个新的测试用例，使其仅覆盖一个尚未被覆盖的无效等价类，重复这一步，直到所有的无效等价类都被覆盖为止。

### C.1.2 边界值分析

边界值分析就是对输入或输出的边界值进行测试的一种方法。

边界值分析是对等价类划分方法的补充，人们从长期的测试工作经验得知，大量的错误是发生在输入或输出范围的边界上，而不是在输入范围的内部。因此针对各种边界情况设计测试用例，可更有效地查出更多的错误。

边界分析的原则：

- a) 边界值分析不是从某等价类中随便挑一个作为代表，而是使这个等价类的每个边界都要作为测试条件，因此在等价类的边界上以及两侧的情况设计测试用例；
- b) 首先应确定边界情况，通常输入和输出等价类的边界，就是应着重测试的边界情况；
- c) 在边界上，选取正好等于、刚刚大于、或刚刚小于边界的值作为测试数据，不能选取等价类中的典型值或任意值作为测试数据。

### C.1.3 错误推测法

错误推测法是基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性地设计测试用例的方法

基本思想：

- a) 例举出程序中所有可能有的错误和容易发生错误的特殊情况，根据他们选择测试用例；
- b) 根据以往模块或者产品中经常出现的错误或者经验。

### C.1.4 因果图

因果图是一种利用图解法分析输入的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况，最终生成判定表（decision table）。

能帮助我们按一定步骤，高效率地选择测试用例，同时还能为我们指出，系统需求规格说明书描述中存在的问题。

#### a) 基本步骤

- 1) 分析系统需求规格说明书描述中，哪些是原因（即输入条件或输入条件的等价类），哪些是结果（即输出条件），并给每个原因和结果赋予一个标识符；
- 2) 分析系统需求规格说明书描述中的语义，找出原因与结果之间、原因与原因之间的对应关系并根据这些关系，画出因果图；
- 3) 由于语法或环境限制，有些原因与原因之间、原因与结果之间的组合情况不可能出现，为表明这些特殊情况，在因果图中用一些记号标明约束或限制条件；
- 4) 把因果图转换成判定表；
- 5) 判定表的每一列拿出来作为依据，设计测试用例。

#### b) 基本符号

通常在因果图中用 $c_i$ 表示原因，通常置于图的左部；用 $e_i$ 表示结果，通常在图的右部； $c_i$ 和 $e_i$ 均可取值“0”或“1”，“0”表示某状态不出现，“1”表示某状态出现。如图C.1所示。

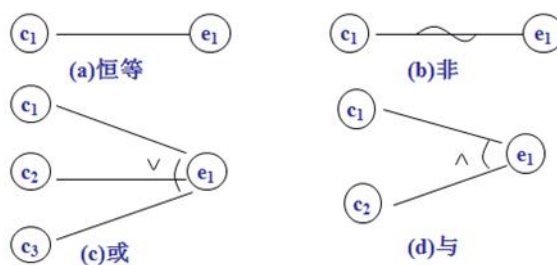


图 C.1 因果图

- 1) 恒等：若  $c_i$  是 1，则  $e_i$  也是 1；否则  $e_i$  为 0；
- 2) 非：若  $c_i$  是 1，则  $e_i$  是 0；否则  $e_i$  是 1；

- 3) 或：若  $c_1$  或  $c_2$  或  $c_3$  是 1，则  $e_i$  是 1；否则  $e_i$  为 0。“或”可有任意个输入；  
4) 与：若  $c_1$  和  $c_2$  都是 1，则  $e_i$  为 1；否则  $e_i$  为 0。“与”也可有任意个输入。

c) 约束符号

输入状态相互之间还可能存在某些依赖关系，称为约束。

为了表示原因与原因之间，结果与结果之间可能存在的约束条件，在因果图中可附加一些表示约束条件的符号。如图 C. 2 所示。

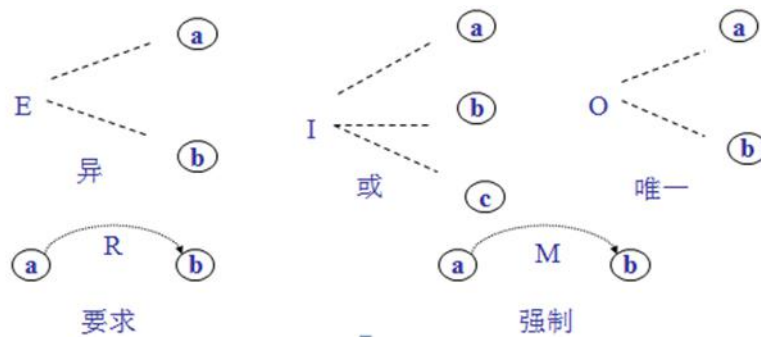


图 C. 2 约束符号

输入条件的约束有以下 4 类：

- 1) E 约束（异）：a 和 b 中至多有一个可能为 1，即 a 和 b 不能同时为 1；  
2) I 约束（或）：a、b 和 c 中至少有一个应是 1，即 a、b 和 c 不能同时为 0；  
3) O 约束（唯一）：a 和 b 应有一个，且仅有一个为 1；  
4) R 约束（要求）：a 是 1 时，b 应是 1，即不可能 a 是 1 时 b 是 0。

输出条件约束类型：

输出条件的约束只有 M 约束（强制）：若结果 a 是 1，则结果 b 强制为 0。

d) 实战举例

某个软件的系统需求规格说明书中包含下面的要求：

第一列字符应是 A 或 B，第二列字符应是一个数字，在此情况下进行文件的修改。但如果第一列字符不正确，则给出信息 L；如果第二列字符不是数字，则给出信息 M。

设计步骤 1

根据规格说明画出因果关系表和因果图，如表 C. 1、图 C. 3 所示。

表 C. 1 实战举例 - 因果关系表

编号	原因（条件）	编号	结果（动作）
1	第一列是字符 A	21	修改文件
2	第一列是字符 B	22	给出信息 L
3	第二列字符是一数字	23	给出信息 M
11	中间原因		

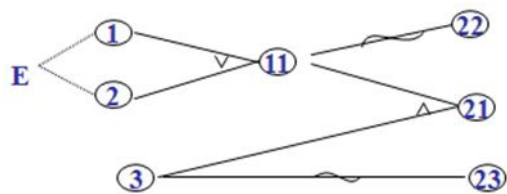


图 C.3 实战举例 - 因果关系图

设计步骤2

根据因果图建立判定表（如表C. 2的条件和动作部分），即按条件的各种组合情况产生对应的动作。  
原因1与原因2不可能同时成立，故可排除这种情况。

设计步骤3

从判定表可设计出6个测试用例（如表C. 2的测试用例部分）。

表 C.2 实战举例 - 测试设计

	1	2	3	4	5	6	7	8
条件 (原因)	1	1	1	1	1	0	0	0
	2	1	1	0	0	1	1	0
	3	1	0	1	0	1	0	1
	11			1	1	1	1	0
动作 (结果)	22			0	0	0	0	1
	21			1	0	1	0	0
	23			0	1	0	1	0
测试用例			A3 A8	AB A?	B5 B4	BN B!	C2 X6	SD P\$

C.1.5 场景法

软件用事件触发来控制流程，事件触发时的情景便形成了场景，而同一事件不同的触发顺序和处理结果就形成事件流。这种在软件设计方面的思想引入到软件测试中，可比较生动地描绘出事件触发时的情景，有利于测试设计者设计测试用例，同时使测试用例更容易理解和执行。

场景法分析测试用例如图C. 4所示。

可分为基本流和备选流：图中经过用例的每条路径都用基本流和备选流来表示，直黑线表示基本流，是经过用例的最简单的路径。备选流用不同的色彩表示，一个备选流可能从基本流开始，在某个特定条件下执行，然后重新加入基本流中（如备选流1和3）；也可能起源于另一个备选流（如备选流2），或者终止用例而不再重新加入到某个流（如备选流2和4）。

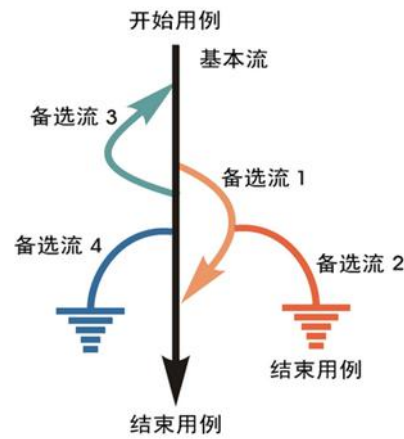


图 C.4 场景法分析测试用例

实战举例：  
图C. 5所示是定单请求的流程示意图。

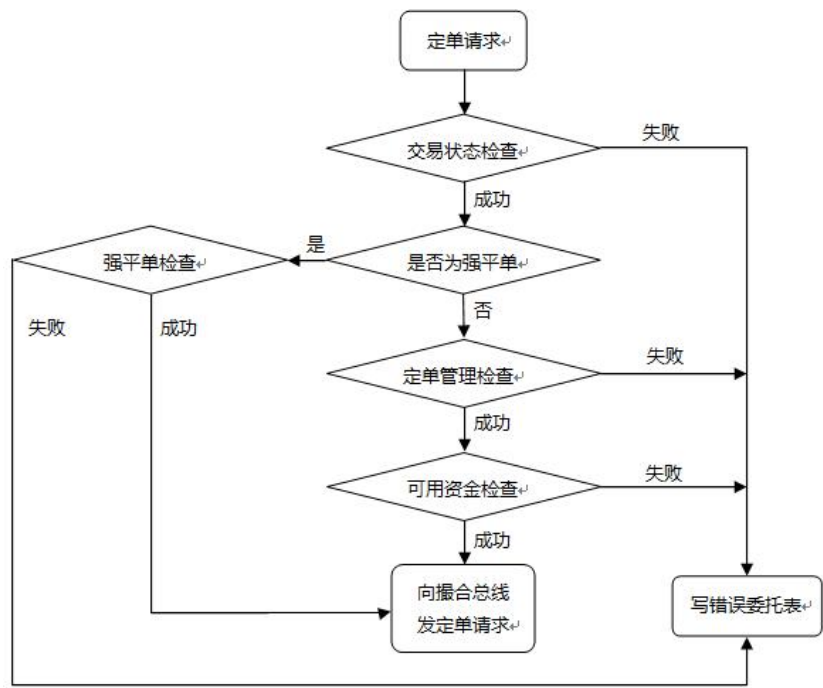


图 C.5 定单请求流程示意图

设计步骤1  
确定基本流和备选流，进行场景设计，表C. 3所示为定单请求功能生成的场景。

表 C.3 测试场景表

场景名称	所属流	
场景1——向撮合总线发定单请求	基本流	
场景2——是强平单且强平单检查成功	基本流	备选流1



表 C.3 测试场景表（续）

场景名称	所属流	
场景3——是强平定单且强平单检查失败	基本流	备选流1 备选流2
场景4——交易状态检查失败	基本流	备选流3
场景5——定单管理检查失败	基本流	备选流4
场景6——可用资金检查失败	基本流	备选流5

## 设计步骤2

进行用例设计，对于这5个场景中的每一个场景应确定测试用例。可采用矩阵或决策表来确定和管理测试用例。表C.4所示，每行代表一条测试用例，而每列则代表测试用例的信息。本示例中，对于每个测试用例，存在一个测试用例ID、场景/条件（或说明）、测试用例中涉及的所有数据元素（作为输入或已经存在于数据库中）以及预期结果。

表 C.4 测试用例表

测试用例ID	场景/条件	交易状态检查	是否为强平单	强平单检查	定单管理检查	可用资金检查	预期结果
CW1	场景1: 向撮合总线发定单请求	V	V	n/a	V	V	定单请求成功
CW2	场景2: 是强平定单且强平单检查成功	V	I	V	n/a	n/a	定单请求成功
CW3	场景3: 是强平定单且强平单检查失败	V	I	I	n/a	n/a	定单请求失败，写错误委托表
CW4	场景4: 交易状态检查失败(交易状态为初始化)	I	n/a	n/a	n/a	n/a	定单请求失败，写错误委托表
CW5	场景4: 交易状态检查失败(交易状态为连续交易暂停)	I	n/a	n/a	n/a	n/a	定单请求失败，写错误委托表
CW6	场景4: 交易状态检查失败(交易状态为闭市)	I	n/a	n/a	n/a	n/a	定单请求失败，写错误委托表
CW7	场景5: 定单管理检查失败	V	V	n/a	I	n/a	定单请求失败，写错误委托表
CW8	场景6: 可用资金检查失败	V	V	n/a	V	I	定单请求失败，写错误委托表
注1: V（有效）用于表明此条件应是 VALID（有效的）才可执行基本流							
注2: I（无效）用于表明此条件下将激活所需备选流							
注3: n/a（不适用）表明此条件不适用于测试用例							

## 设计步骤3

进行数据设计，一旦确定了所有的测试用例，就可确定实际数据值（在测试用例实施矩阵中）并且设计测试数据。

## C.1.6 正交实验法

正交实验法是依据Galois理论，从大量的（实验）数据（测试例）中挑选适量的，有代表性的点（例），从而合理地安排实验（测试）的一种科学实验设计方法。

利用正交实验设计测试用例的步骤：

- a) 提取功能说明，构造因子——状态表

把影响实验指标的条件称为因子，而影响实验因子的条件叫因子的状态，利用正交实验设计方法来设计测试用例时，首先应根据被测试软件的系统需求规格说明书找出影响其功能实现的操作对象和外部因素，把他们当作因子，而把各个因子的取值当作状态。对系统需求规格说明中的功能要求进行划分，把整体的概要性的功能要求进行层层分解与展开，分解成具体的有相对独立性的基本的功能要求。这样就可把被测试软件中所有的因子都确定下来，并为确定各个因子的权值提供参考的依据。确定因子与状态是设计测试用例的关键。因此要求尽可能全面的正确的确定取值，以确保测试用例的设计做到完整与有效。

b) 加权筛选，生成因素分析表

对因子与状态的选择可按其重要程度分别加权。可根据各个因子及状态的作用大小，出现频率的大小以及测试的需要，确定权值的大小。

c) 利用正交表构造测试数据集

正交表的推导依据Galois理论（这里省略，需要时可查数理统计方面的教材）。

### C.1.7 组合测试法

组合测试法（Combinatorial Test）是一种测试用例生成方法。它是将被测系统抽象出来的变量的取值进行组合并生成一组测试用例的过程，它将被测试应用抽象为一个受到多个因素影响的系统，其中每个因素的取值是离散且有限的。多因素（N-way,  $N \geq 2$ ）组合测试可覆盖任意N个因素的所有取值组合，在理论上可发现由N个因素共同作用引发的缺陷。组合测试法包含正交矩阵法、结对组合法等多种具体实施方法。证券期货业内推荐采用结对组合法（即AllPair或者Pairwise）进行测试用例的生成，结对组合法可根据设定的条件自动组合生成高覆盖、低冗余的测试用例，擅于发现每两个因素同时作用时引发的缺陷，在测试质量和测试效率上达到了较好的平衡。

## C.2 白盒测试

### C.2.1 语句覆盖

设计若干个测试用例，运行所测程序，使得每一可执行语句至少执行一次。

### C.2.2 判定覆盖

设计若干个测试用例，运行所测程序，使得程序中每个判断的取真分支和取假分支至少经历一次（又叫分支覆盖）。

### C.2.3 条件覆盖

设计若干个测试用例，运行所测程序，使得程序中每个判断的每个条件的可能取值至少执行一次。

### C.2.4 判定-条件覆盖

设计足够的测试用例，使得判断中每个判定的所有可能取值至少执行一次，同时每个判断的所有可能结果也至少执行一次。

### C.2.5 条件组合覆盖

设计足够的测试用例，运行所测程序，使得每个判断的所有可能的条件取值组合至少执行一次。

#### C.2.6 路径覆盖

设计足够的测试用例，覆盖程序中所有可能的路径。

#### C.2.7 基本路径

在程序控制流图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例的方法，应保证测试中程序的每个可执行语句至少执行一次。

#### C.3 灰盒测试

介于黑盒测试和白盒测试之间的一种测试技术。

附 录 D  
(规范性附录)  
静态测试方法

D.1 概述

静态测试是指不运行被测系统本身，仅通过分析或检查等来保证程序正确性的方法。

静态测试对象包括代码和软件开发生命周期中的各种文档，如：系统需求规格说明书、系统架构设计、数据库设计、概要设计、详细设计、原型、测试计划、测试用例、用户手册、接口设计规范等。接口设计规范的评审应考虑机构内接口和机构间接口两类规范。针对代码的静态测试有评审，主要关注逻辑正确性、功能正确性及编码规范等，还可使用专业工具进行代码静态检查，主要关注程序结构及逻辑正确性、表达式正确性、编码规范及安全漏洞等。

静态测试贯穿于整个软件开发生命周期，好的静态测试可保证尽早地发现缺陷，并对缺陷进行修改及验证，从而降低项目成本，是早期缺陷预防的一个重要手段。

静态测试方法主要是评审，详细说明见D.2。

D.2 评审

D.2.1 审查

审查是最系统化，也是最严谨的一种评审方式。审查的特性和作用如表D.1。

表 D.1 审查的特性和作用

特性	作用
主持评审的是评审负责人，而不是工作产品生产者。	能客观地对待工作产品的评审。
读者（材料陈述者）对要评审的资料进行宣读及阅读	通过读者（材料陈述者）对要评审的资料的宣读及阅读可审查可理解性和可读性。
多个评审人员	不同的审查者会发现不同的问题，3到7个评审人员为最佳。

D.2.2 技术复审

技术复审是一种非正式的评审，是由一组技术专家对重点内容进行评审的方式。技术复审的特性和作用如表D.2。

表 D.2 技术复审的特性和作用

特性	作用
可不召开会议	1) 生产率高； 2) 对评审人员自身的知识、技能、自律要求比较高。
应事先确定评审重点，比如工作产品需满足的标准、接口或依赖关系等。	使得评审的目的更有针对性，更能集中资源发现问题。

### D.2.3 走查

走查是一种非正式的评审，是由工作产品生产者将该工作产品向一组评审人员介绍，并希望他们给出意见。当评审的首要目的是使别人了解工作产品时，走查是比较适合的。走查的特性和作用如表D.3。

表 D.3 走查的特性和作用

特性	作用
省略多个角色和多个步骤（总体的会议和跟踪审查的步骤可省略）	节省时间和成本。
工作产品生产者直接将工作产品向一组评审人员介绍	1) 工作产品生产者处于主导地位，主要是为了满足工作产品生产者的需要； 2) 可能导致评审人员忽略某些没有被重点提到的部分。
允许一组有资历的人来评审工作产品	专门的评审人员在评审的技巧、能力，流程掌握上更加专业。

### D.2.4 评审方法的区别

不同的评审方式评审的内容和过程也不同，主要区别如下表D.4、表D.5所示。

表 D.4 内容区别

特性	审查	技术复审	走查
主持	评审负责人	技术负责人	工作产品生产者
材料陈述者	读者	作者或组员	工作产品生产者
使用记录员	是	是	可能
文档化的评审记录	是	是	可能
设置专门的评审角色	是	否	否
使用评审检查表	是	可能	可能
评审问题的跟踪	评审负责人	技术负责人	无
数据收集与分析	是	可能	可能

表 D.5 过程区别

方法	策划	准备	开会	纠错	验证
审查	○	○	○	○	○
技术复审	○	○		○	○
走查	○			○	
注：“○”表示选取					

D.2.5 评审方法的选择

选择适合的评审方法主要考虑：组织评审的有效性和效率。技术复审在技术文档和重要的管理文档方面有效性突出；走查能增加对代码评审的覆盖率；审查更适合于将要成为基准的工作产品。根据评审的侧重点不同，选择表D.6中不同的评审方法。

表 D.6 评审方法选择表

工作产品	走查	技术复审	审查
测试工作量估计书		○	
测试计划		○	○
单元测试用例	○		
集成测试用例	○	○(重点部分)	
系统测试用例	○	○(重点部分)	
系统集成测试用例	○	○(重点部分)	
验收测试用例		○	
测试总结报告		○	
注：“○”表示选取			

附 录 E  
(规范性附录)  
业务分类

按照被测系统（含传统终端及移动端）所涉及业务的影响范围和重要程度，可将被测试系统的业务划分为不同的业务分类，见表E. 1。

表 E. 1 业务分类

序号	业务分类	说明
1	A 类	发生故障后，对资本市场实时交易活动产生直接影响的业务，包括但不限于交易、行情、风控、初始化、资金及持仓结算等相关业务，涉及实时通讯平台类的基础组件等。
2	B 类	发生故障后，对资本市场非实时交易活动产生影响的业务，包括但不限于交割、监察、会员/客户管理、投资管理、业务配置管理、TA、估值、场外业务等相关业务。
3	C 类	仅利用交易或业务数据进行查询或进行自身逻辑计算的非核心业务，包括但不限于数据查询、互联网信息、内部办公、运维监控、营销服务等相关业务。

## 附 录 F

### (规范性附录)

### 缺陷管理

#### F.1 缺陷管理目的

缺陷管理的目的是：

- a) 对测试执行过程中所发现缺陷的记录及解决方案实施流程化管理；
- b) 对所发现缺陷的根本原因进行分析。

缺陷管理是用来度量测试流程及测试品质的一个重要指标。

一个稳健的缺陷追踪流程有助于应用系统整体项目达到最终的成功。

#### F.2 缺陷管理的生命周期

缺陷状态描述见表F.1，并不是全部的缺陷都要经历以下所有的状态，并且由于在整个开发生命周期中存在迭代，所以有些缺陷可能要多次经历相同的状态。

表 F.1 缺陷状态描述

序号	状态	描述
1	新建	发现并新建缺陷。
2	打开	缺陷被分析审阅确认为真实的缺陷。
3	分配	缺陷被指定给适当开发人员进行分析。
4	修复	确定是缺陷/问题：且根本原因已确定；开发已修正并在开发环境验证通过。
5	可测	确定是缺陷/问题：修正方案已被正式发布，可进行复测、验证。
6	解决	缺陷确认已被解决。
7	关闭	被解决的缺陷可被关闭。
8	取消	系统设计及功能符合业务需求。
9	重复	已有类似缺陷存在。
10	拒绝	缺陷不可重现或并不是问题或与现系统无关。
11	重新打开	缺陷没有被解决、已解决的问题又重新出现或原来遗留的缺陷可进行修复。
12	延迟修复	确定是缺陷/问题：但由于各种局限性（系统、架构、项目进度等），缺陷暂时不修复，遗留到后期。



表F. 2为状态转换矩阵，描述了各个缺陷状态的转换关系。

表 F. 2 缺陷状态转换矩阵

结束状态	开始状态									
	新建	打开/重新打开	分配	修复	可测	解决	延迟修复	取消	拒绝	关闭
打开	●									
分配		●								
修复			●							
可测				●						
解决					●					
重新打开					●	●	●	●	●	●
取消	●									
重复	●	●	●							
拒绝		●	●							
延迟修复		●	●							
关闭						●				
注：“●”表示选取										

F.3 缺陷管理流程图

缺陷管理的流程见图F.1。

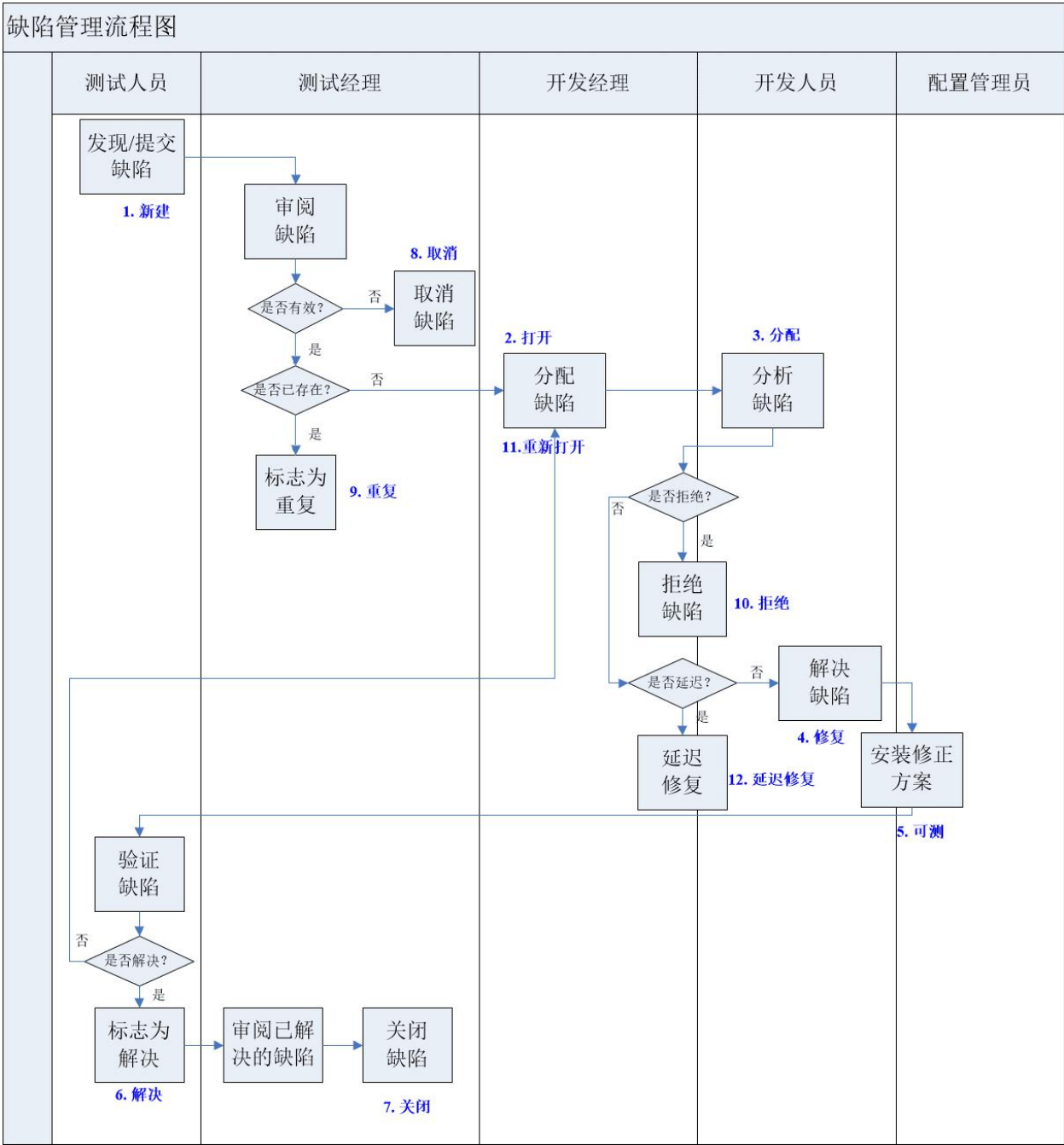


图 F.1 缺陷管理流程

F.4 缺陷严重度划分

缺陷严重度划分见表F.3。

表 F.3 缺陷严重度

严重度	对业务的影响程度	应用系统是否可用	描述
致命问题	致命	否	系统宕机，崩溃，不可用。
严重问题	严重	是	功能失效、缺失；计算，逻辑错误等。
一般问题	高	是	功能可用，但功能处理存在错误。
轻微问题	中	是	不影响功能运行，GUI 问题，提示信息等。
改善建议	低	是	改善建议。

F.5 缺陷优先级划分

优先级划分见表F.4。

表 F.4 缺陷优先级

优先级	对业务的影响程度	描述
5-紧急	致命	缺陷对系统引起重大问题，应尽快解决； 在没有提供修正方案前，测试不能继续执行。
4-极高	严重	在下个交付之前应解决； 有变通方法可使用，但是缺陷应在进行下一轮测试前被解决。
3-高	高	在下个交付之前应解决； 有变通方法可使用，但是缺陷应在进行下一轮测试前被解决。
2-中	中	在下个交付之前可选择的解决； 有变通方法可使用，可选择推后解决，但本测试级别完成前应被解决。
1-低	低	在下个交付之前可不解决； 有变通方法可使用，且在没有提供修正方案前应用系统可进行下一测试级别。

F.6 根本原因举例

测试执行过程中提出的缺陷或问题可能由多种原因引起，表F.5列出的根本原因可作为参考，也可将根本原因描述为更接近编码级别的原因，如日志问题、精度问题、命名规则问题、死循环等，具体以实际情况及缺陷管理工具为主。

表 F.5 根本原因示例

根本原因	对应缺陷状态	描述
历史遗留	修复/ 延迟修复	原有系统或其他系统遗留缺陷，非本项目引起。
架构		由于系统构架问题引起的缺陷。
设计		由于系统详细设计问题引起的缺陷。
代码		由于编码问题引起的缺陷。
集成	修复	由于系统集成问题引起的缺陷。
环境	修复	由于测试环境配置问题引起的缺陷。
发布		由于版本发布环节引起的缺陷。
需求		由于项目相关方对需求理解有歧义引起的既修改需求又改代码； 明显的需求编写错误。
数据	拒绝	由于基础原始数据准备引起的缺陷。
测试		由于错误的测试执行引起的缺陷。
非缺陷		功能与需求及设计相符或缺陷不可重现或与现系统无关。

## 参 考 文 献

- [1] GB/T 25000.10-2016 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第10部分:系统与软件质量模型
- [2] GB/T 25000.51-2016 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第51部分:就绪可用软件产品(RUSP)的质量要求和测试细则
- [3] JR/T 0060-2010 证券期货业信息系统安全等级保护基本要求(试行)
- [4] JR/T 0146.1-2016 证券期货业信息系统审计指南第1部分:证券交易所
- [5] JR/T 0146.2-2016 证券期货业信息系统审计指南第2部分:期货交易所
- [6] JR/T 0146.3-2016 证券期货业信息系统审计指南第3部分:证券登记结算机构
- [7] JR/T 0146.4-2016 证券期货业信息系统审计指南第4部分:其他核心机构
- [8] JR/T 0146.5-2016 证券期货业信息系统审计指南第5部分:证券公司
- [9] JR/T 0146.6-2016 证券期货业信息系统审计指南第6部分:基金管理公司
- [10] JR/T 0146.7-2016 证券期货业信息系统审计指南第7部分:期货公司
- [11] ISO/IEC 17025 检测和校准实验室能力的通用要求 General requirements for the competence of testing and calibration laboratories
- [12] CMMI软件能力成熟度集成模型 SEI (Software Engineering Institute) 发布, 美国卡内基梅隆大学的软件工程研究所(SEI)
- [13] TMMI 测试成熟度模型模型 TMMI 基金会发布
-