

软件测试与质量保证

2.1 静态测试技术概述

张宇霞 副研究员



目录

CONTENTS

01

静态测试

02

详细设计的静态测试

03

小结



目录

CONTENTS

01

静态测试

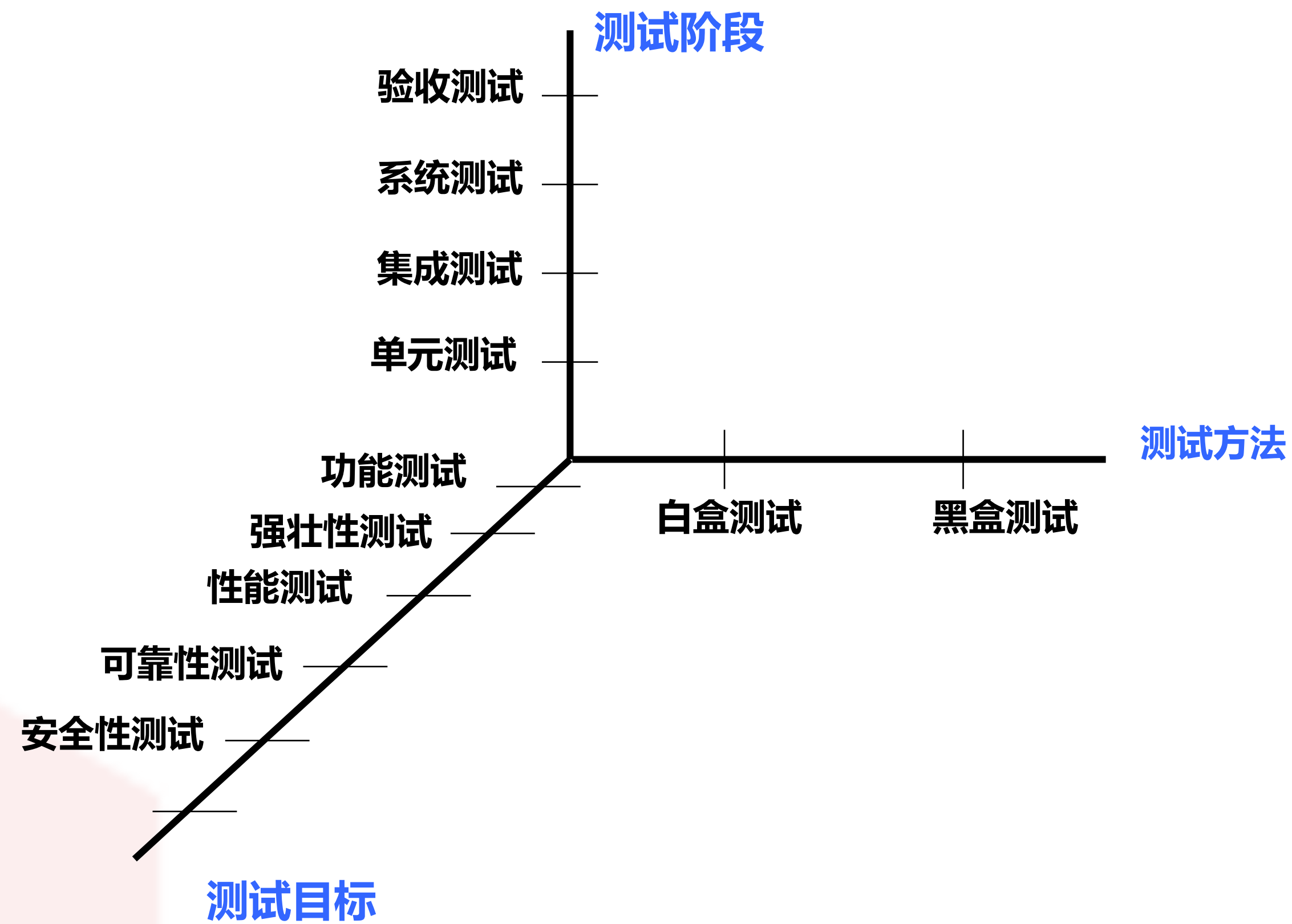
02

详细设计的静态测试

03

小结





多维度分类：

测试方法

测试阶段

测试目标



静态测试

■ 动态测试

- 指通过**运行被测程序**，检查运行结果与预期结果的差异，并分析运行效率、正确性和健壮性等性能。
- 动态测试方法由三部分组成：构造测试用例、**执行程序**、分析程序的输出结果。

■ 静态测试

- 指**不运行被测程序**本身，仅通过分析或检查源程序（或其他软件制品）的语法、结构、过程、接口等来检查程序/设计的正确性。
- 通常不需要测试用例，但需要静态检查表。

■ 静态测试的目标及内容：

- 检查需求文档，是否切合实际，是否自相矛盾，是否清晰明确。
- 检查详细设计文档，是否符合概要设计的要求。
- 检查代码风格，是否符合规范。
- 检查程序设计和结构，是否合理。
- 检查业务逻辑，是否有逻辑缺陷。

■ 静态测试方法：

- **同行评审**：由软件工作产品创建者的同行们检查该工作产品，识别产品的缺陷，改进产品的不足。
- **数据流测试**：检查程序的控制流程，根据事件的顺序，探索跟踪变量的值以及变量的使用情况等。

■ 同行评审：由软件工作产品创建者的**同行**们检查该工作**产品**，识别产品的缺陷，改进产品的不足。

- 同行：指项目成员和具有同等开发专业技能的并熟知工件的人员。具体来说指项目组成员或公司内其他项目组成员以及少量公司外的专家。
- 产品：指最终产品的组成部分，包括源代码、设计、文档等。



静态测试

■ 同行评审的形式

- 走读 (Walkthrough)
- 小组评审(Team Review)
- 审查(Inspection)

■ 走读、小组评审和审查都有**正式**评审和**非正式**评审两种方式



■ 走读：同行评审中最自由的一种形式

- 主要目的：评价软件制品 (比如软件代码)。
- 其他目的：技术的交换、参与人员的技术培训、设计思想的介绍等。
- 评价什么：代码正确性、效率、**可读性**。
- 走读成员：项目内部的其他开发人员。
- 方式：对照检查表 (checklist) 检查代码。

小组评审

■ 组织

- 参与者主要是公司技术领导或权威及公司外部专家。

■ 适用范围

- 主要适用于需求文档和概要设计。

■ 目标

- 给出意见和建议：确认某个制品是否符合要求，是否可以进入下一个阶段，如何提高制品的质量，如何使之符合要求。

■ 组织

- 公司内部设计、开发、测试、质量等部门中工作性质相关的员工。

■ 形式

- 它遵循一个严格的过程，人员经过培训，检查过程有标准。

■ 适用范围与目的

- 检查针对实际的产品或半成品，目的是发现存在的错误。

三种同行评审方式的区别

分类	走读	小组评审	审查
目标	检测缺陷; 找出可替换的方法	评估产品的价值和 市场可行性; 评估需求规格和设 计规格的正确性	检测和识别缺陷
组织者/会议 主持人	作者	公司技术领导	测试或质量主管
评审员	同项目组成员	公司内技术权威或 公司外行业专家	项目负责人、公司设 计人员、质量管理人 员、测试人员
人员规模	2-7	3人或更多	3-6人
数据收集	可选	可选	正式要求

三种同行评审方式的区别

分类	走读	小组评审	审查
记录员	可选（作者自己）	必须专门设置	必须专门设置
输出报告	可选	可选 (建议性意见)	缺陷列表; 缺陷总结
活动计划	有	有	有
活动前准备	评审员会议前 不需要阅读作品	评审员会议前 阅读作品	评审员会议前 阅读作品
适用范围	代码、详细设计	概要设计、需求规格	代码、详细设计、 概要设计

是否需要主动介绍

■ 正式评审，YES

- 在评审会议上，工作产品的开发者向评审成员介绍工作产品，以征求评审成员的意见和批准。

■ 非正式评审，NO

- 由工件开发者与一个或多个评审人员进行，往往不需要召开会议，由评审人员自己安排时间进行评阅后提交建议或结论。

① 计划阶段：明确评审目的，确定参加人员并分配角色和职责。

➤ 作者、组织者、评审员、记录员

② 评审实施：主要是会议形式。

③ 评审情况统计：对会议中提出的问题进行统计整理，并形成文档。

④ 问题跟踪：作者根据评审中提出的问题进行产品修改，相关人员进行检查。

静态测试是不需要运行被测试软件的一种测试手段

同行评审是静态测试的主要手段

同行评审可以分为走读、小组评审和审查

目录

CONTENTS

01

静态测试

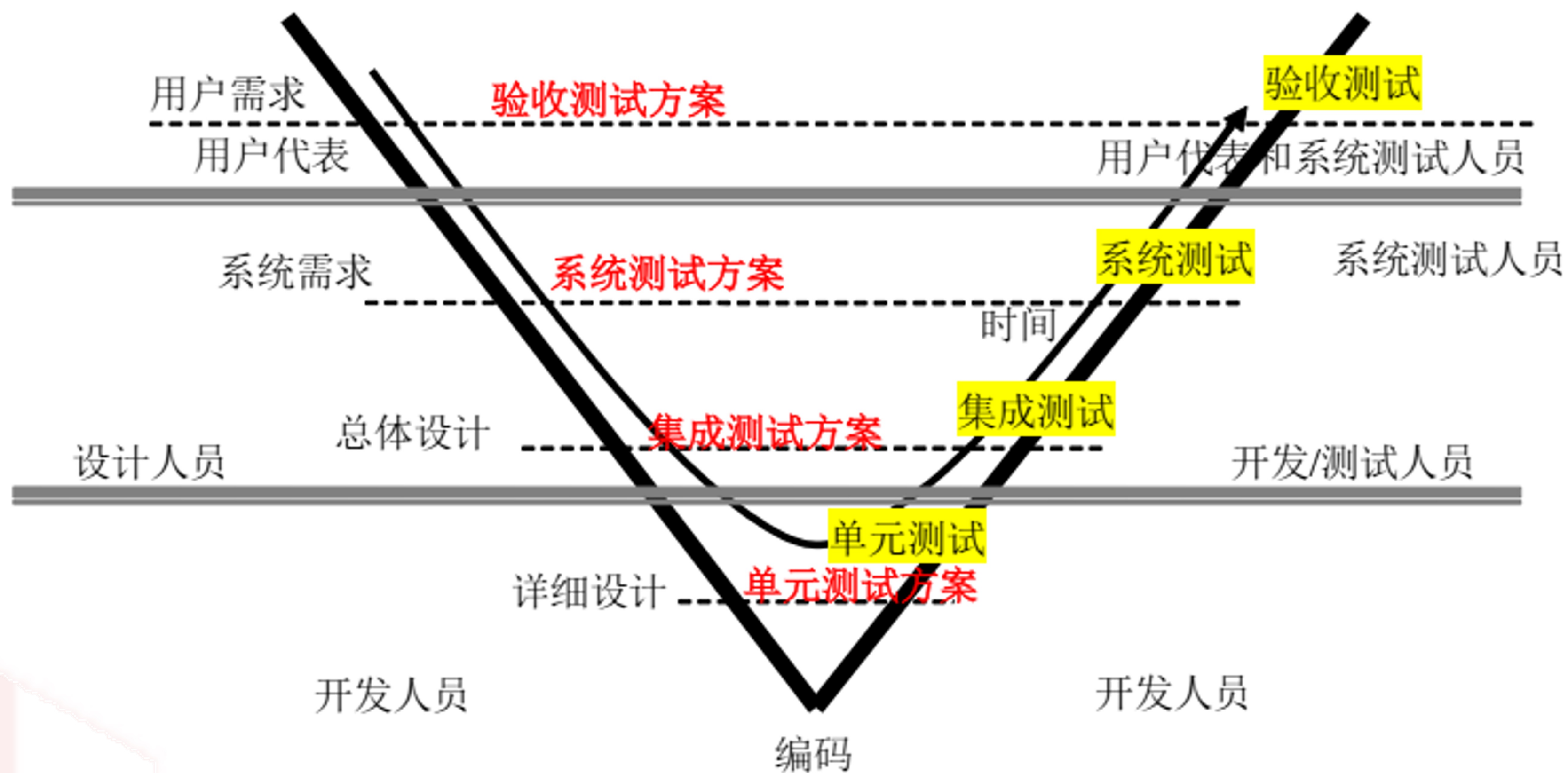
02

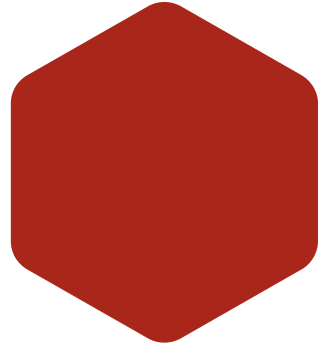
详细设计的静态测试

03

小结







详细设计说明书

- 为每个模块确定采用的算法，选择某种适当的工具表达算法的过程，写出模块的详细过程性描述。
- 确定每一模块使用的数据结构。
- 确定模块接口的细节。
- 编写详细设计说明书

系统名称—详细设计说明书

目录

1. 引言	1
1.1 术语	1
1.2 缩略语	1
2. 模块概述	3
3. 模块的功能点	3
4. 模块结构设计	4
4.1 模块内部构成	4
4.2 模块的实现	6
4.3 变更机能的实现	6
5. 模块接口设计	6
5.1 外部接口设计	6
5.2 内部接口设计	6
6. 使用的系统资源	7
6.1 ROM/RAM 使用量	7
6.2 中断的定义和使用	7
6.3 TIMER 资源的使用	7
6.4 OS 资源的使用	7
6.5 使用的其他资源	7
7. 限制条件	7
8. 模块详细设计	8
8.1 文件一览	8
8.2 函数一览	8
8.3 函数调用关系	9
8.4 宏定义	9
8.5 消息一览	10
8.6 数据类型定义	10
8.7 全局变量定义	10
8.8 数据表设计	10
8.9 接口函数设计	11
8.9.1 函数 1 (实际函数名)	11
8.9.2 函数 2	12
8.10 内部函数设计	12
8.10.1 函数 1 (实际函数名)	12
8.10.2 函数 2	12
8.10.3 函数 3	13
8.11 流程图	14
8.11.1 函数 1 (实际函数名)	14
8.11.2 函数 2	15
9. 附件	16
9.1 模块技术点整理	16
9.2 参考文档	16

详细设计的静态测试

- **测试依据：概要设计（总体设计）**

- 主要测试详细设计同概要设计的一致性，以及详细设计的合理性。

- **测试方法：采用同行评审的形式为审查或走读。**



检查要素	检查内容
清晰性	是否所有的单元和进程的设计目的都已文档化
	单元设计，包括数据流、接口描述是否清楚
	单元的整体功能是否描述清楚
完整性	是否描述了所采用的标准
	是否确定了每个单元应用的算法
	是否列出了单元的所有调用
规范性	文档是否遵从了公司的标准
	单元设计是否使用了要求的方法和工具
一致性	在程序单元和单元的接口中数据成员的名称是否保持一致
	所有接口之间，接口和接口规格书之间是否保持一致
	详细设计是否同概要设计一致



详细设计的静态测试

检查要素	检查内容
正确性	是否有逻辑错误
	是否所有的条件都被处理 (>, =, <0, switch case)
数据	是否所有声明的数据都已经被使用
	数据结构是否已经描述
	如果有对共享数据、文件的修改，对数据的访问是否按照正确的共享协议进行 (例如：信号灯)
可靠性	对所有错误情况都安排了有意义的消息反馈
	特殊情况下的返回码是否和文档中定义的全局返回码一致
	是否考虑了异常情况 (如磁盘空间不足、网络掉线)
可追溯性	是否每个详细设计单元都可以追溯到概要设计



检查要素	检查内容
接口	参数表是否在数量、类型和顺序上保持一致
	是否所有的输入输出都已经正确定义并检查过
	所传递参数的顺序是否描述清楚
	参数传递的机制是否确定
	通过接口传递的常量和变量是否与单元设计的相同 (例如：函数中定义的常量不能在所调用的函数中被修改)
	是否以度量单位描述了参数的值区间、准确性和精度



目录

CONTENTS

01

静态测试

02

详细设计的静态测试

03

小结



软件测试不一定要跑程序

同行评审是最常见的静态测试方法

详细设计依赖于静态测试

谢谢！

