

基线需求、升级a、升级b已完成成果物介绍

汇报人：文小库
2024-01-02



目录

- 需求文档
- 测试安装部署文档
- 演示视频
- 代码成果
- 测试用例设计

contents

目 录

- 用例清单、前置条件
- 未完成工作及达成率
- 小组成员分工
- 源代码管理
- maven构建介绍、pom文件介绍

contents

01

需求文档



需求描述

介绍需求的来源、产生原因和重要性。

界定需求的边界和涉及的方面。

明确需求要解决的问题和要达到的目标。



功能点

1

功能点一

详细描述该功能点的功能、输入、输出和实现方式。

2

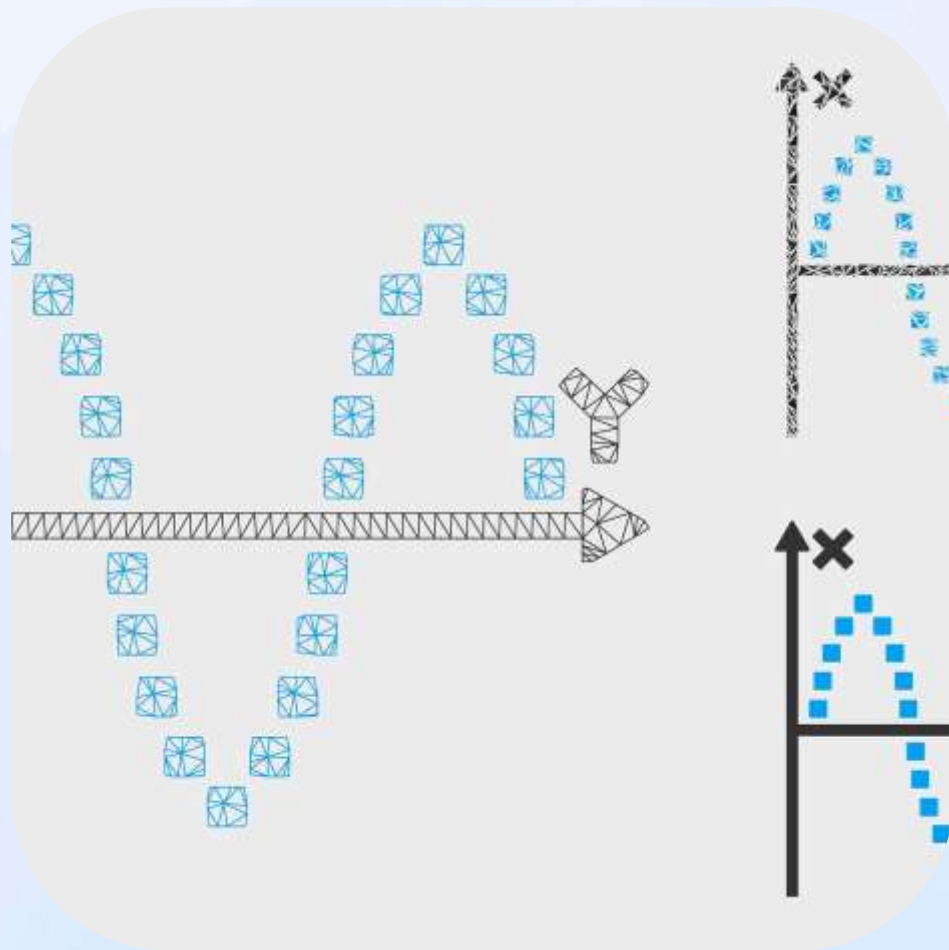
功能点二

详细描述该功能点的功能、输入、输出和实现方式。

3

功能点三

详细描述该功能点的功能、输入、输出和实现方式。





测试要求

测试环境

说明测试所需的硬件、软件和网络环境。

测试数据

提供测试所需的数据来源、数据格式和数据量。



测试方法

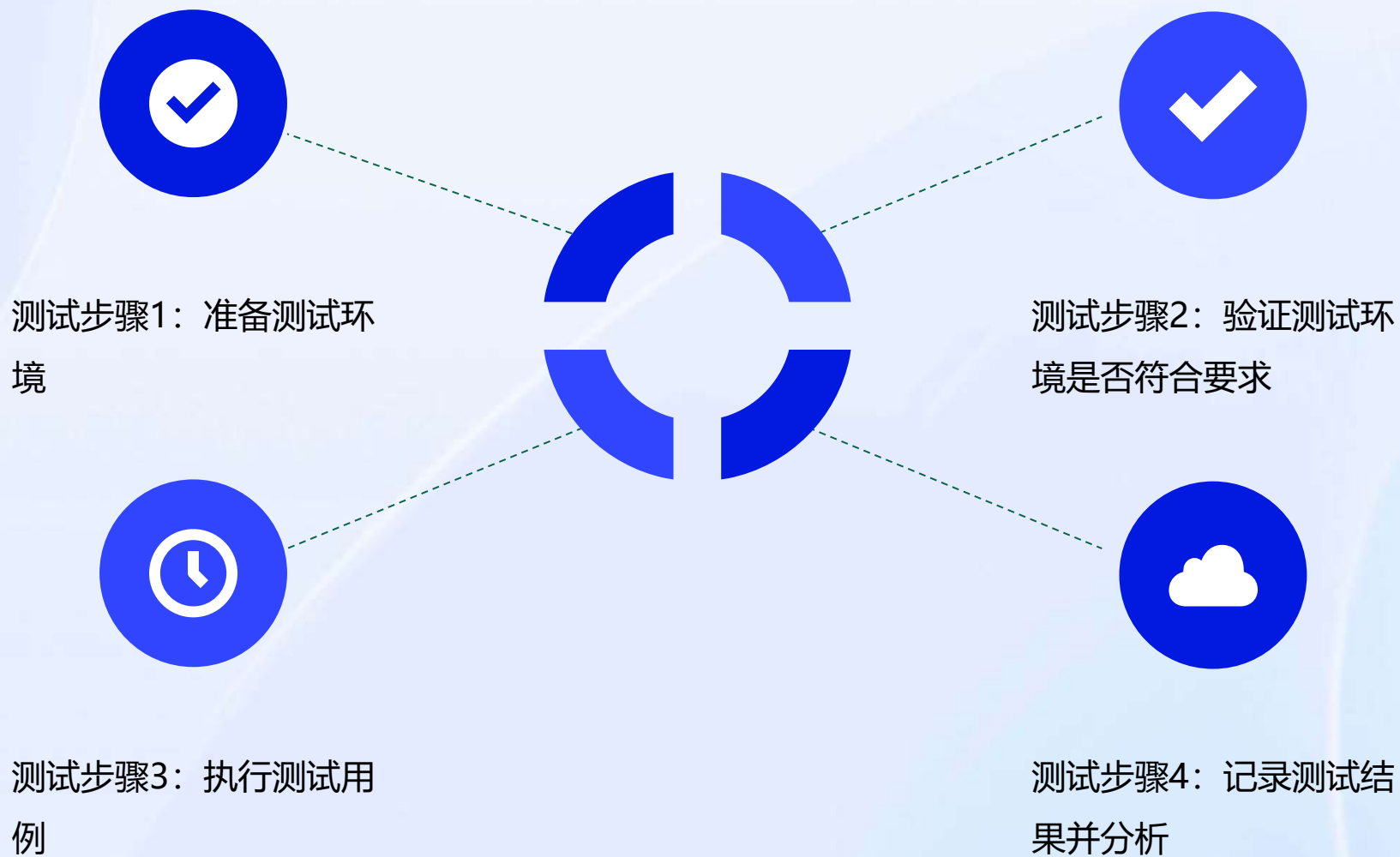
详细描述测试的步骤、测试用例设计方法和测试结果分析方法。

02

测试安装部署文档



测试步骤





安装步骤



安装步骤1：下载安装包并解压



安装步骤3：按照提示进行安装



安装步骤2：运行安装程序



安装步骤4：完成安装并重启



部署说明

01

部署说明1：确
定部署环境

02

部署说明2：配
置部署参数

部署说明3：执
行部署脚本

03

04

部署说明4：验
证部署结果

03

演示视频

功能展示



01

功能1

详细描述功能1的具体内容，如它如何满足用户需求，它的工作原理等。

02

功能2

详细描述功能2的具体内容，如它如何满足用户需求，它的工作原理等。

03

功能3

详细描述功能3的具体内容，如它如何满足用户需求，它的工作原理等。



操作流程

● 流程1

详细描述流程1的具体内容，如它的步骤、操作方式等。

● 流程2

详细描述流程2的具体内容，如它的步骤、操作方式等。

● 流程3

详细描述流程3的具体内容，如它的步骤、操作方式等。





特点解说

特点1

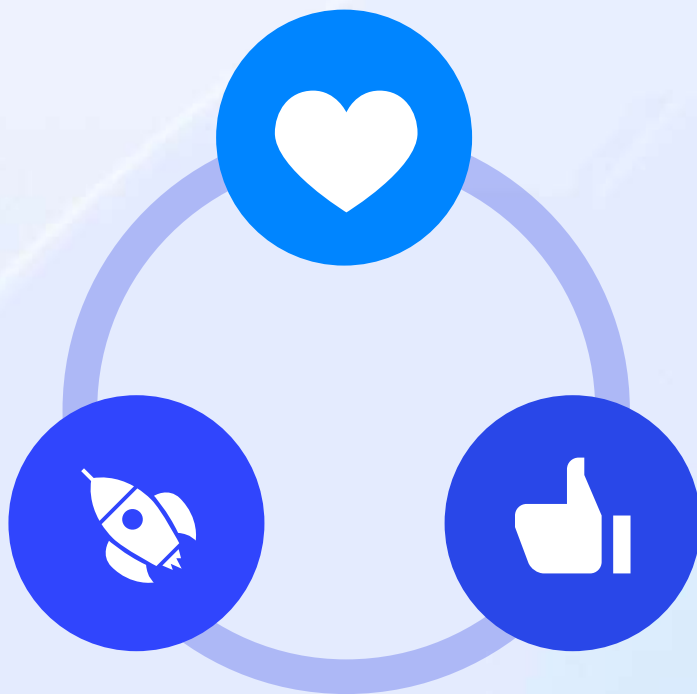
详细描述特点1的具体内容，如它与其他产品的区别、优点等。

特点2

详细描述特点2的具体内容，如它与其他产品的区别、优点等。

特点3

详细描述特点3的具体内容，如它与其他产品的区别、优点等。



04

代码成果



代码完成情况

总结词

代码完成情况良好，所有需求点均已实现，无遗漏。

详细描述

经过对代码的审查和测试，确认所有基线需求、升级a和升级b的代码开发任务均已完成，且功能正常。

。



代码质量保障

总结词

代码质量得到了有效保障，遵循了编码规范和最佳实践。

详细描述

代码编写过程中，遵循了公司内部的编码规范和最佳实践，确保了代码的可读性、可维护性和可扩展性。同时，通过代码审查和单元测试，及时发现并修复了潜在的代码质量问题。



代码测试情况

总结词

代码测试覆盖率高，测试用例完整且通过率高。

VS

详细描述

对所有完成的代码进行了详尽的测试，包括单元测试、集成测试和系统测试。测试用例覆盖了所有需求点，且通过率高。同时，对测试过程中发现的问题进行了及时的修复和再次测试，确保了代码的稳定性和可靠性。

05

测试用例设计



用例覆盖率



需求覆盖率

确保测试用例覆盖了所有基线需求，没有遗漏任何功能或场景。



异常覆盖率

测试用例应覆盖正常和异常情况，包括边界条件、错误处理和安全考虑。



性能覆盖率

测试用例应考虑性能需求，包括响应时间、吞吐量、资源利用率等。



用例设计原则

01

明确性

测试用例应明确描述测试目标、输入、输出和测试步骤。

02

完整性

测试用例应覆盖所有相关功能和场景，避免遗漏。

03

可重复性

测试用例应具有可重复性，以便进行回归测试和持续集成。

04

可维护性

测试用例应易于理解和维护，方便后续修改和扩展。





用例执行计划



执行人员

确定测试用例的执行人员，包括测试人员、开发人员等。



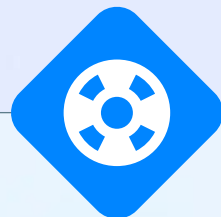
执行环境

准备测试所需的软硬件环境，包括测试工具、测试数据等。



执行时间

制定测试用例的执行时间表，确保按时完成测试任务。



执行流程

明确测试用例的执行流程，包括测试准备、执行、结果记录和缺陷跟踪等步骤。

06

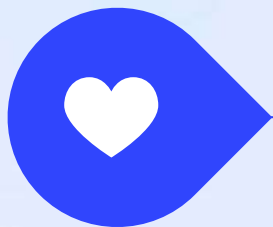
用例清单、前置条件



用例列表

用例1

用户登录功能



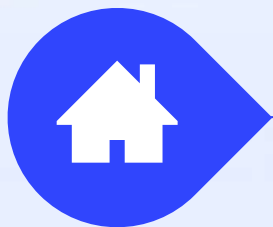
用例2

用户注册功能



用例3

商品浏览功能



用例4

商品购买功能





用例执行步骤



用例1执行步骤

1. 用户输入用户名和密码

。

2. 系统验证用户名和密码
是否正确。



用例执行步骤

1

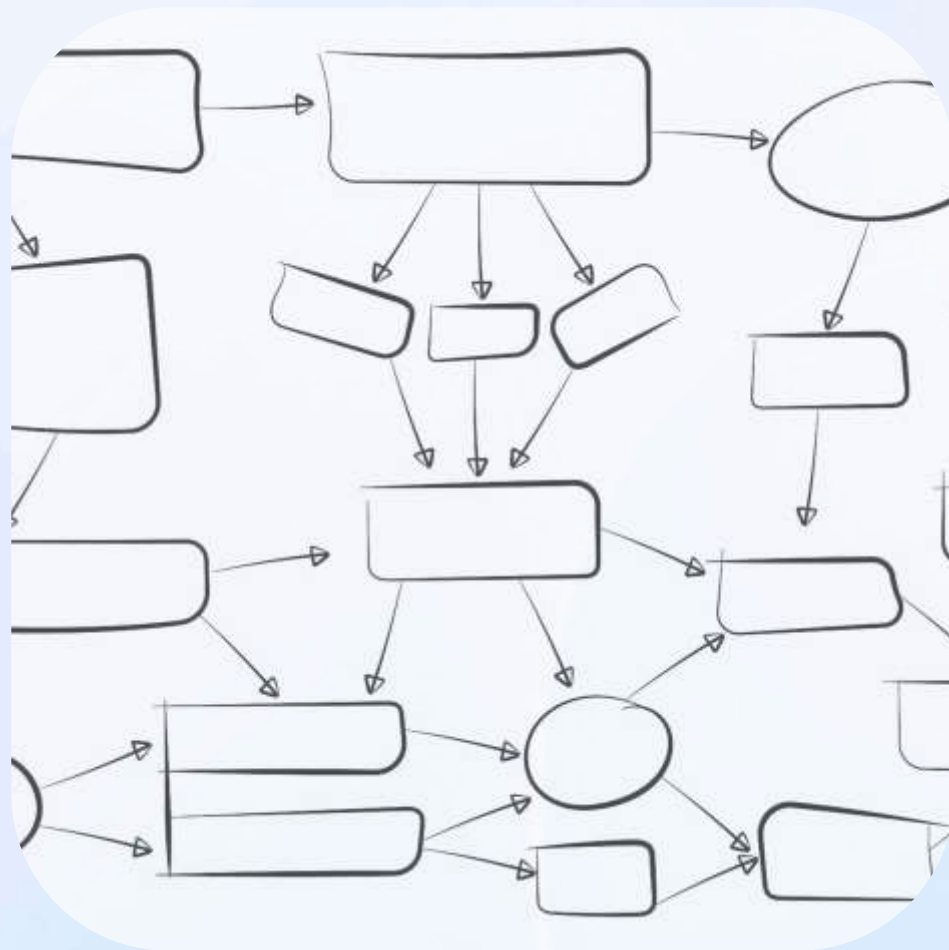
3. 如果验证通过，用户登录成功；否则，登录失败。

2

用例2执行步骤

3

1. 用户填写注册信息，包括用户名、密码、邮箱等。





用例执行步骤



01

2. 系统验证用户填写的信息是否符合规范。

02

3. 如果验证通过，用户注册成功；否则，注册失败

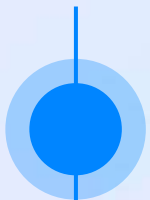
。

03

用例3执行步骤



用例执行步骤



1. 用户进入商品浏览页面。



2. 用户选择感兴趣的商品。



3. 系统展示所选商品的详细信息。



用例执行步骤

01

用例4执行步骤

02

1. 用户选择要购买的商品，并填写收货地址。

03

2. 系统处理用户的购买请求。

04

3. 系统完成支付处理，并通知用户支付结果。



前置条件说明

前置条件1

用户已连接到互联网。



前置条件2

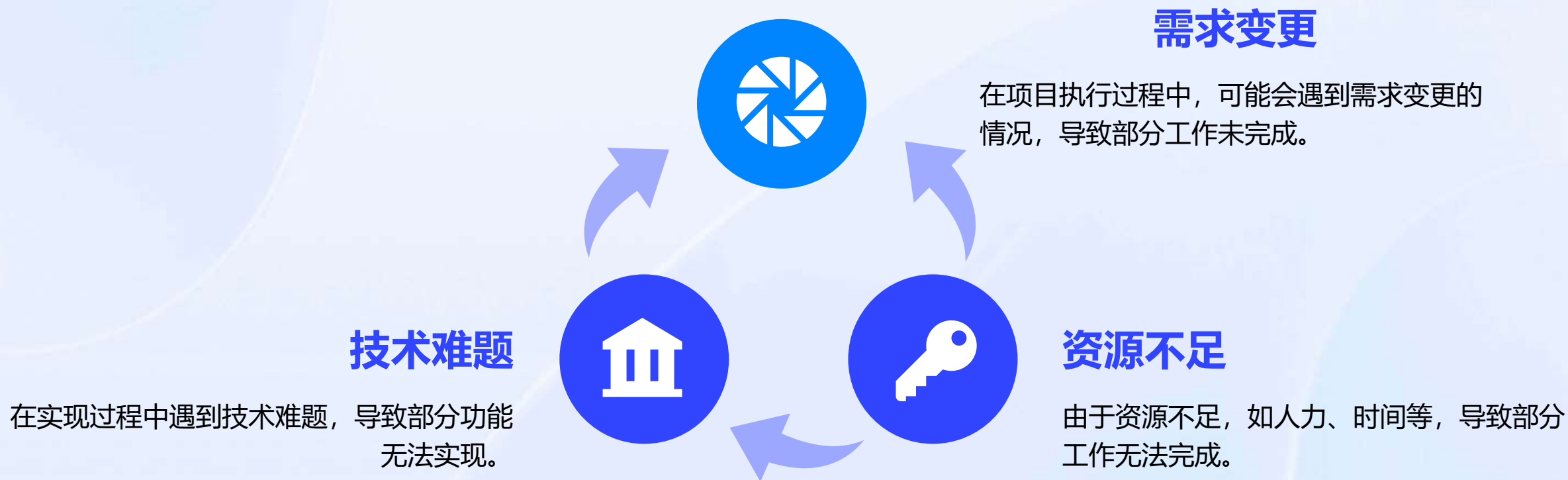
系统服务器正常运行。



07

未完成工作及达成率

未完成工作内容





达成率计算方法



任务完成数量

统计已完成的任务数量，并将其作为达成率的计算基础。

总任务数量

统计项目总的任务数量，包括已完成和未完成的任务。

达成率计算公式

达成率 = (任务完成数量 / 总任务数量) * 100%。



达成率计算依据

任务清单

根据项目开始时制定的任务清单，统计每个任务的完成情况。



实际完成情况

根据项目执行过程中的实际完成情况，调整任务清单，并更新达成率。



质量评估

在计算达成率时，还需考虑任务完成的质量，确保达成率的准确性。

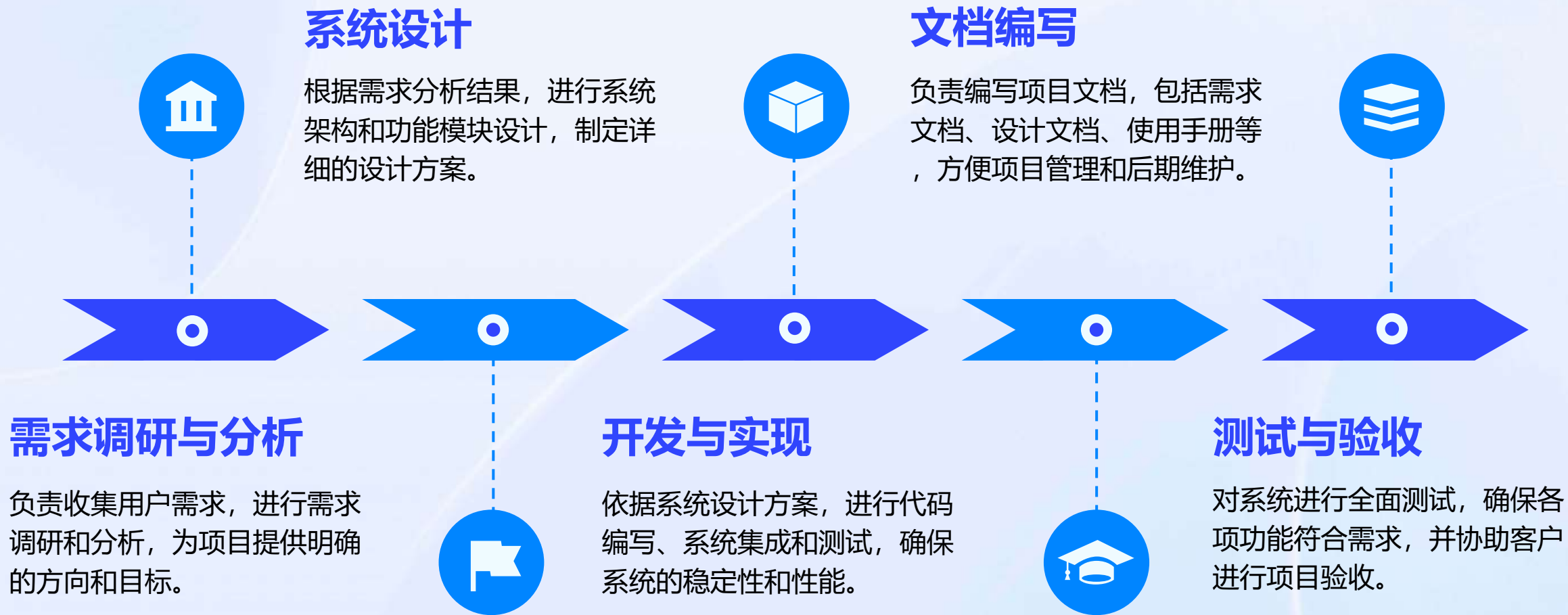


08

小组成员分工



分工情况说明



工作量统计

需求调研与分析

3人, 工作量占比20%

系统设计

2人, 工作量占比15%

开发与实现

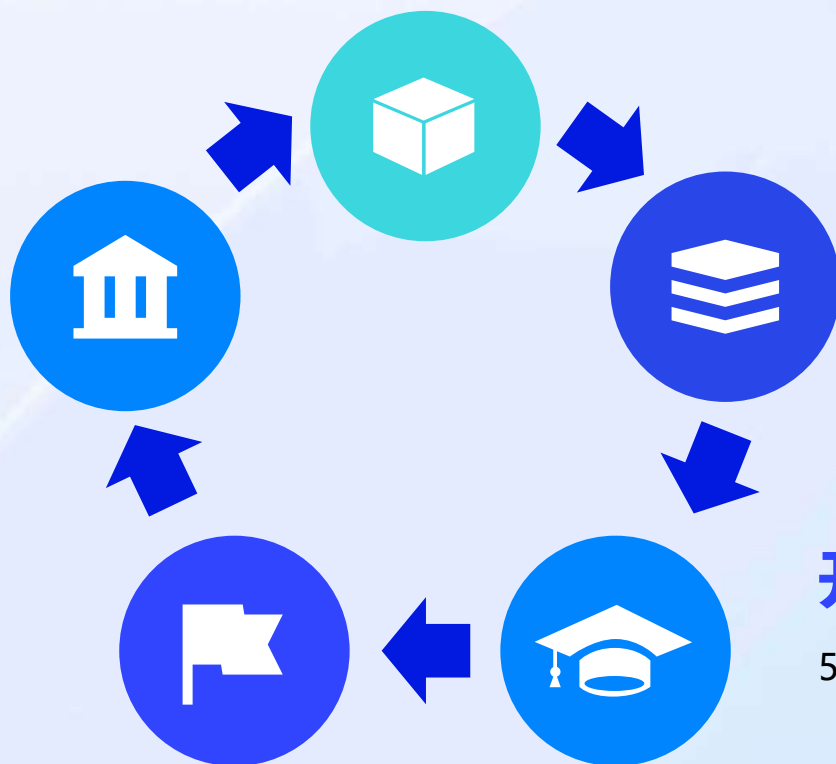
5人, 工作量占比35%

文档编写

2人, 工作量占比15%

测试与验收

3人, 工作量占比20%





工作质量评估

需求调研与分析

准确把握客户需求，
为项目提供明确方向，
评估为优秀。

系统设计

设计方案合理、全面，
考虑了系统的扩展性和可维护性，
评估为良好。

开发与实现

代码质量高，系统运行稳定，性能良好，
评估为优秀。

文档编写

文档内容完整、准确，
方便项目管理和后期维护，
评估为良好。

测试与验收

测试全面、细致，协助客户顺利完成项目验收，
评估为优秀。



09

源代码管理



提交情况统计

提交次数统计

对项目源代码的提交次数进行统计，以了解代码的更新频率和团队的合作情况。

提交者分布

分析每次提交的作者，了解团队成员的贡献程度和合作模式。

提交内容分类

对每次提交的内容进行分类，如bug修复、新功能开发、文档更新等，以便更好地理解项目的开发进度和方向。





版本控制说明

版本控制工具

介绍项目所使用的版本控制工具，如Git，以及该工具的主要功能和特点。

版本管理策略

说明项目的版本管理策略，包括分支管理、标签使用、合并策略等，以确保代码的稳定性和可维护性。

冲突解决机制

提供解决版本控制中常见的冲突的方法和步骤，以确保团队成员之间的协同工作顺利进行。



代码规范说明



命名规范

规定代码中变量、函数、类等命名规则，以提高代码的可读性和可维护性。

代码格式化

提供代码的缩进、排版等格式化标准，以确保代码风格的一致性。

注释规范

说明代码中的注释规则，包括注释的位置、内容和格式，以帮助其他开发者更好地理解代码的功能和实现方式。

10

maven构建介绍、pom文件介绍



maven构建过程



解析项目依赖关系

Maven会解析项目中的pom.xml文件，确定项目所依赖的库和插件，并下载所需的依赖库和插件。



编译源代码

Maven会自动编译项目中的Java源代码，并将编译后的class文件输出到target目录下。



打包项目

Maven可以根据项目类型（如jar、war等）自动打包项目，生成相应的可执行文件或部署包。



运行测试

Maven可以运行项目中的单元测试和集成测试，确保代码质量。



pom文件解析

项目坐标

pom.xml文件中定义了项目的坐标，包括groupId、artifactId、version等，用于唯一标识一个项目。

项目构建配置

pom.xml文件中可以配置项目的编译、测试、打包等阶段，定义自动化构建流程。

项目依赖

pom.xml文件中可以声明项目所依赖的库和插件，Maven会自动下载和管理这些依赖。

项目信息

pom.xml文件中可以包含项目的名称、描述、版本等信息，方便了解项目的基本情况。





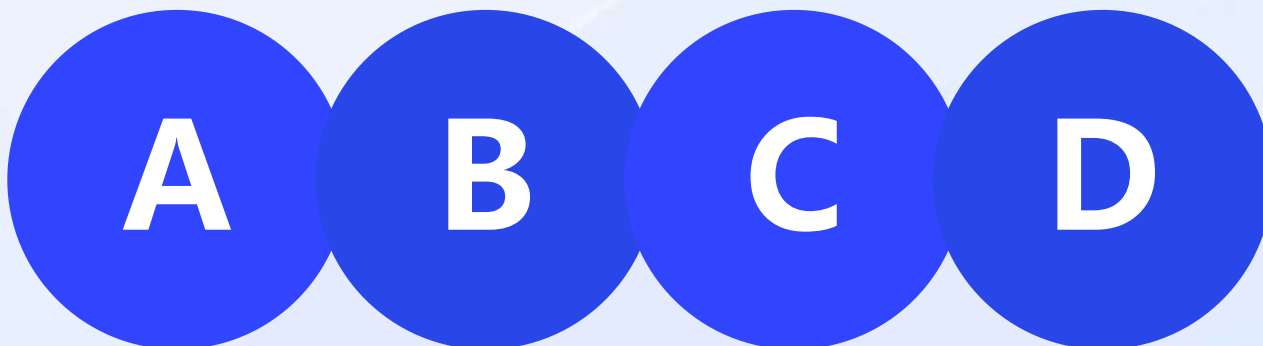
pom文件优势

统一管理

使用pom.xml文件统一管理项目的依赖库和插件，方便团队成员之间的协作。

可扩展性

pom.xml文件中可以配置多个模块，实现多模块项目的构建和管理。



自动化构建

通过配置pom.xml文件，可以实现自动化编译、打包、测试等构建过程，提高开发效率。

版本控制

将pom.xml文件纳入版本控制系统（如Git），方便跟踪项目的构建历史和版本信息。

THANKS

感谢观看

