

项目名称：点菜宝

文档名称：管理活动分析文档

HUSTZL

侯皓斐 软件2003 班 U202010851

版本：V1.0

文档信息

文件状态： <input type="checkbox"/> 草稿 <input checked="" type="checkbox"/> 正式发布 <input type="checkbox"/> 正在修改	文件标识：	HUSTZL_001
	当前版本：	V1.0
	作者：	侯皓斐
	发布日期：	< 2023.4.27 >

文档更改记录

版本	更改日期	更改人	更改原因	说明
V1.0	2023.4.27	侯皓斐		初版

摘要

本文旨在编写一个瀑布模型来实现点菜宝程序的管理文档。

瀑布模型是一种经典的软件开发过程模型，包括需求分析、设计、编码、继承和测试，部署和维护等不同的阶段。在本文中，我们将使用瀑布模型来管理点菜宝程序的开发过程。

在需求分析阶段，我们将收集用户需求和功能要求，并定义项目的范围和目标。

在设计阶段，我们将根据需求分析阶段的结果设计系统架构和界面，并制定详细的开发计划。

在编码阶段，我们将根据设计阶段的结果编写代码，并进行单元测试和集成测试。

在继承和测试阶段，我们将对系统进行功能测试、性能测试和用户验收测试，确保系统符合用户需求和质量标准。

再部署阶段，我们配置环境（包括硬件、软件的安装、环境变量的设置等）保证开发的软件用户能够正常使用。

最后，在维护阶段，我们将提供持续的技术支持和维护服务，修复系统中发现的漏洞和错误，并进行必要的更新和升级。

在每个阶段中，我们将定义所有活动名称、活动次序、活动执行人、关键活动输入输出、关键活动规范以及它们之间的关系，以确保项目的顺利开展和高质量的交付。

目录

- 1.引言..... 1
 - 1.1 文档目的..... 1
 - 1.2 背景和项目概述..... 1
 - 1.3 文档范围..... 1
- 2.瀑布模型概述..... 2
 - 2.1 瀑布模型定义..... 2
 - 2.2 瀑布模型不同阶段..... 2
 - 2.3 点菜宝与瀑布模型..... 2
- 3.需求分析阶段..... 3
 - 3.1 活动名称及次序..... 3
 - 3.2 活动执行人..... 3
 - 3.3 关键活动输入输出..... 4
 - 3.4 关键活动规范..... 4
- 4.设计阶段..... 4
 - 4.1 活动名称及次序系统架构设计..... 4
 - 4.2 活动执行人..... 5
 - 4.3 关键活动输入输出..... 5
 - 4.4 关键活动规范..... 5
- 5.实现阶段..... 6
 - 5.1 活动名称及次序..... 6
 - 5.2 活动执行人..... 7
 - 5.3 关键活动输入输出..... 7
 - 5.4 关键活动规范..... 7
- 6.继承与测试阶段..... 7
 - 6.1 活动名称及次序模块集成..... 8
 - 6.2 活动执行人..... 8
 - 6.3 关键活动输入输出..... 8
 - 6.4 关键活动规范..... 8

7.部署阶段.....	9
7.1 活动名称及次序.....	9
7.2 活动执行人.....	9
7.3 关键活动输入输出.....	10
7.4 关键活动规范.....	10
8.维护阶段.....	10
8.1 活动名称及次序.....	11
8.2 活动执行人.....	11
8.3 关键活动输入输出.....	11
8.4 关键活动规范.....	11
9.项目管理与团队协作.....	12
9.1 项目管理方法.....	12
9.2 团队协作.....	12
9.3 变更管理.....	13
10.结论.....	13

1. 引言

1.1 文档目的

本文档的主要目的是为点菜宝项目的开发团队提供一个详细的瀑布模型实现框架，以确保项目按照预定的进度和质量顺利进行。本文档将详细描述瀑布模型的各个阶段，以及在这些阶段中的活动名称、活动次序、活动执行人、关键活动输入输出、关键活动规范，以及他们的关系。通过阅读本文档，项目团队成员可以清晰地了解项目的整体流程、任务分工和执行细节。

1.2 背景和项目概述

点菜宝是一款针对餐饮行业的点菜软件，旨在提高顾客点餐效率、优化餐厅管理和提升顾客用餐体验。该软件可以让顾客通过智能手机或平板电脑进行点餐，同时也方便餐厅工作人员管理订单和库存。点菜宝项目由一支跨领域的团队负责开发，包括产品经理、开发人员、设计师、测试人员等。

为了确保项目的顺利进行，团队选择了瀑布模型作为开发方法。瀑布模型是一种经典的软件开发过程模型，将开发过程划分为多个阶段，每个阶段完成后再进入下一个阶段。这种顺序性的开发方式有利于项目管理和进度控制。

1.3 文档范围

本文档覆盖了点菜宝项目的瀑布模型实现过程中的所有关键阶段，包括需求分析、设计、实现、集成与测试、部署和维护。同时，文档还涉及项目管理和团队协作方面的内容。需要注意的是，本文档主要关注于瀑布模型的实现框架，而非具体的技术实现细节。对于技术实现细节，将在各个阶段的具体文档中进行详细描述。应该注意的是项目团队成员没有可能全职做一种任务，其必然要身兼多职。

2. 瀑布模型概述

2.1 瀑布模型定义

瀑布模型（Waterfall Model）是一种线性顺序的软件开发过程模型，它将软件开发过程划分为一系列相互关联的阶段。在瀑布模型中，每个阶段的输出作为下一个阶段的输入，且每个阶段在整个项目周期中只进行一次。这种自上而下的开发方式有助于清晰地定义项目的起点和终点，便于项目管理和进度控制。

2.2 瀑布模型不同阶段

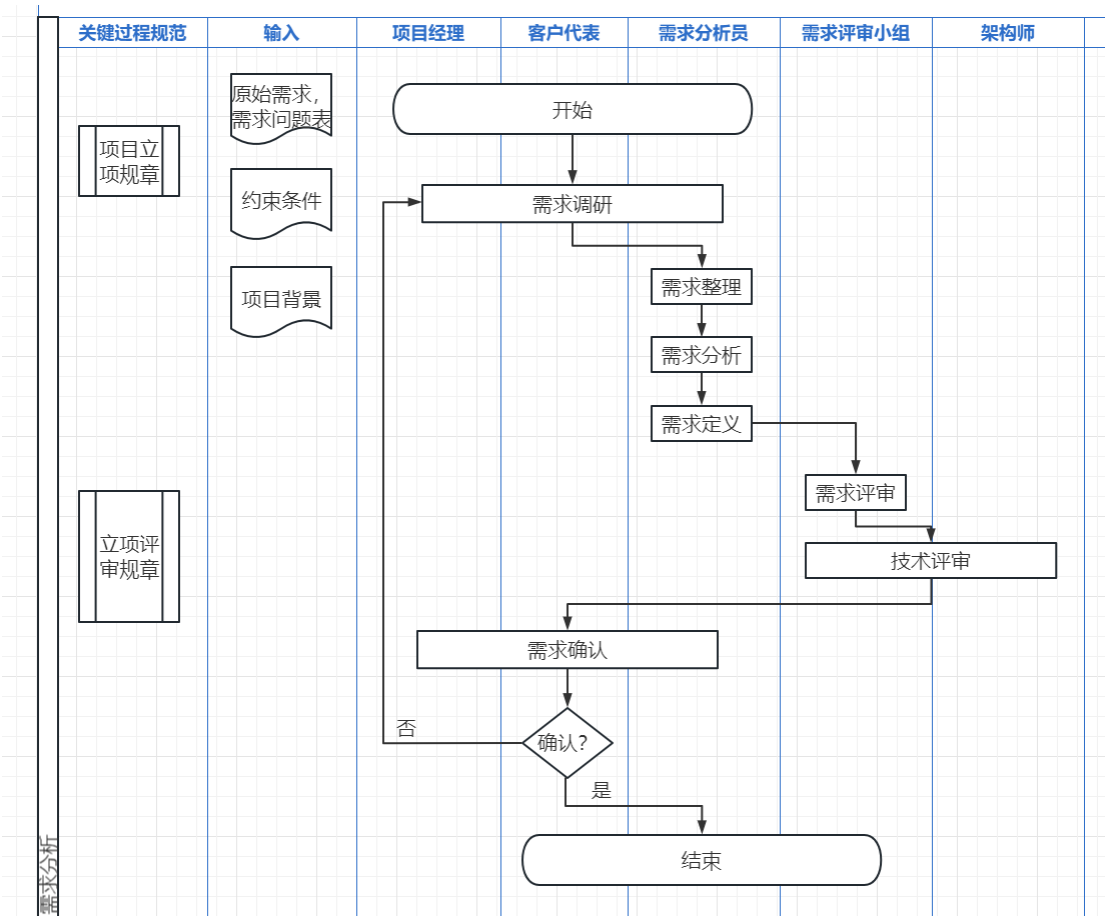
瀑布模型通常包括以下几个阶段：

1. 需求分析：确定项目的需求，包括功能需求和非功能需求。需求分析的主要产出是需求规格说明书，为后续阶段提供指导。
2. 设计：根据需求规格说明书进行系统设计，包括架构设计和详细设计。设计阶段的产出是设计文档，为实现阶段提供指导。
3. 实现：根据设计文档编写代码，将设计转化为实际的软件。实现阶段的产出是可执行的软件程序。
4. 集成与测试：对软件进行集成和测试，确保软件的正确性和可靠性。测试阶段的产出是测试报告和已修复的软件。
5. 部署：将软件部署到生产环境，让最终用户可以使用。部署阶段的产出是可运行的软件和部署文档。
6. 维护：对已部署的软件进行维护，包括修复漏洞、添加新功能和优化性能。维护阶段的产出是更新后的软件和维护报告。

2.3 点菜宝与瀑布模型

点菜宝项目采用瀑布模型作为开发方法，以确保项目在约束下按照预定的进度和质量顺利进行。具体细节请见开发模型确定原因_瀑布模型文档。

3. 需求分析阶段



3.1 活动名称及次序

- 1. 需求调研
- 2. 需求整理
- 3. 需求定义
- 4. 需求评审
- 5. 技术评审
- 6. 需求确认
- 7. 立项结束

3.2 活动执行人

- 1. 产品经理：负责收集和整理业务需求，编写需求规格说明书，参与需求评审和修订。

2. 客户代表：提供业务需求和反馈，参与需求评审。
3. 需求分析师：协助产品经理分析需求，确定系统需求和约束条件，参与需求评审。
4. 需求评审小组：较为专业和独立的评审小组，指出需求分析的不足，为用户，项目经理，架构师提供意见。
5. 架构师：提供技术方面的建议和评估，参与技术评审。

3.3 关键活动输入输出

1. 输入：业务需求、项目背景和约束条件。
2. 输出：需求规格说明书（包括功能需求和非功能需求）。

3.4 关键活动规范

需求分析规范，项目立项和评审制度，关键有如下几点

1. 需求收集应全面、详细，涵盖所有相关业务场景。
2. 需求分析过程中应考虑项目的技术可行性、成本和时间限制。
3. 需求规格说明书应清晰、完整、易于理解，以便为后续阶段提供明确的指导。
4. 需求评审过程应确保需求的准确性、一致性和完整性。

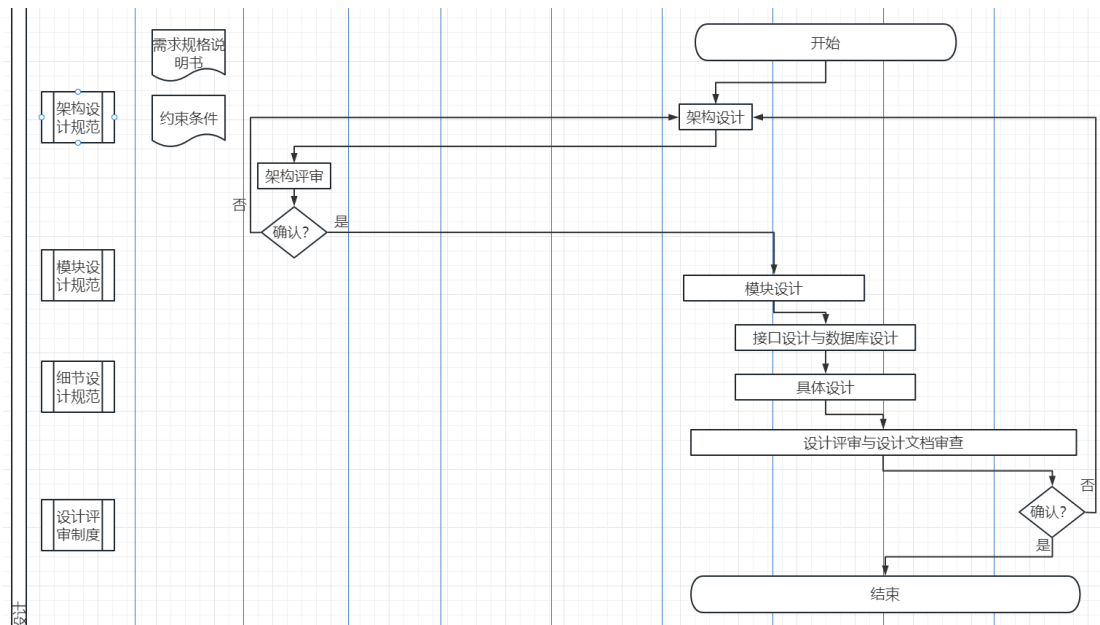
4. 设计阶段

4.1 活动名称及次序系统架构设计

1. 架构设计
2. 架构评审
3. 模块设计
4. 接口设计
5. 数据库设计
6. 具体设计
7. 设计评审

8. 设计文档编写
9. 设计文档审查

4.2 活动执行人



1. 系统架构师
2. 软件设计师
3. 数据库管理员
4. 产品经理
5. 开发团队成员
6. 设计评审小组

4.3 关键活动输入输出

1. 输入：需求规格说明书和约束条件。
2. 输出：系统架构设计文档、模块设计文档、接口设计文档、数据库设计文档。

4.4 关键活动规范

系统架构设计规范：确保系统具有可扩展性、可维护性和性能优化，满足需求规格说明书中的非功能性需求。

模块设计规范：根据系统架构设计，划分软件模块，明确模块间的关系和职责，确保模块职责单一、低耦合和高内聚。

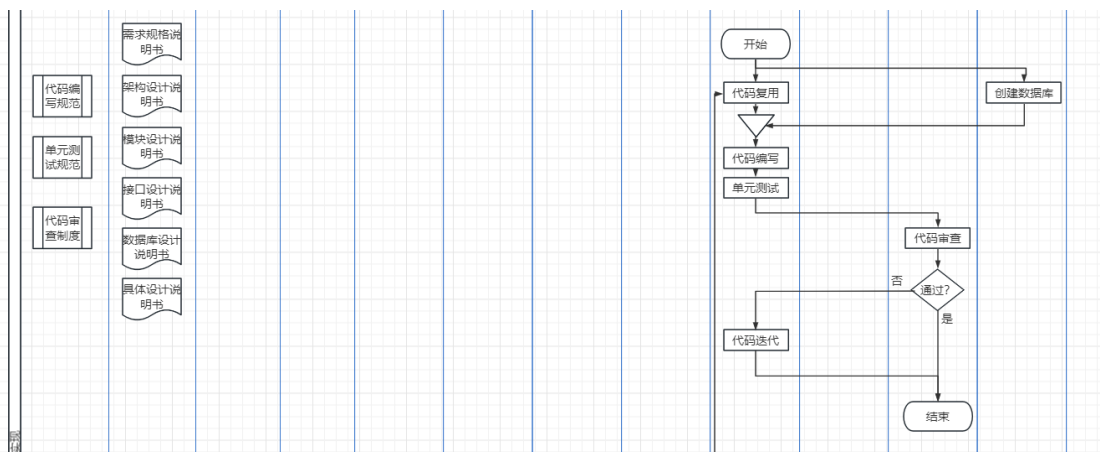
接口设计规范：为各个模块定义清晰的接口，确保接口易于理解和使用，满足系统的可扩展性和可维护性需求。

数据库设计规范：根据需求规格说明书中的数据需求，设计数据库表结构和关系，确保数据的完整性、一致性和安全性。

设计文档编写规范：编写清晰、完整的设计文档，包括系统架构、模块设计、接口设计和数据库设计等内容，作为实现阶段的指导。

设计文档审查制度：对设计文档进行审查，确保设计满足需求规格说明书中的要求，同时遵循设计规范和最佳实践。

5. 实现阶段



5.1 活动名称及次序

1. 代码复用
2. 创建数据库
3. 编写代码
4. 单元测试
5. 代码审查
6. 修复代码审查中发现的问题
7. 合并代码至主分支

5.2 活动执行人

1. 软件开发人员：负责编写代码、进行单元测试、参与代码审查和修复问题。
2. 代码审查人员：负责对代码进行审查，确保代码质量和规范性。
3. 数据库管理员：建立数据库

5.3 关键活动输入输出

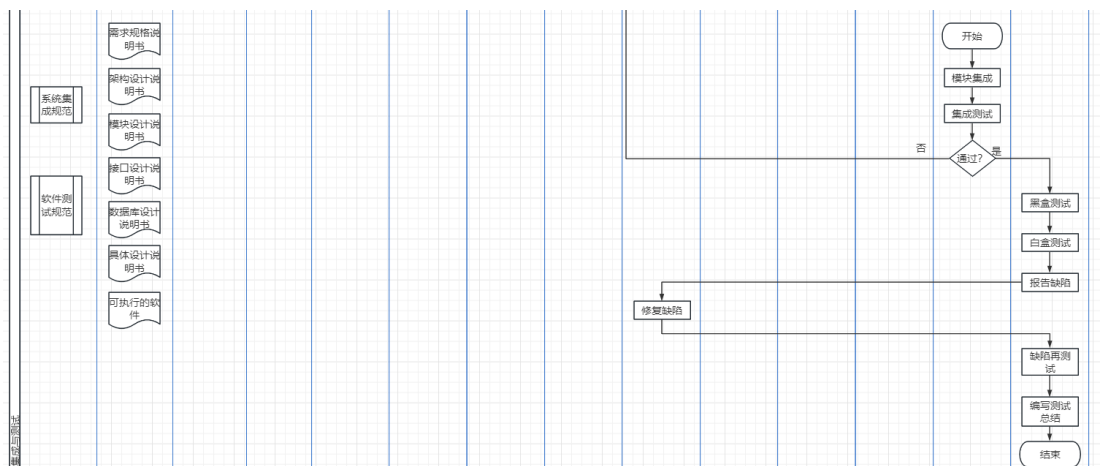
1. 输入：需求规格说明书和约束条件。设计文档（包括系统架构、模块设计、接口设计和数据库设计）。
2. 输出：可执行的软件程序、单元测试报告、代码审查报告。

5.4 关键活动规范

代码编写规范，单元测试规范，代码审查制度，关键有如下几点

1. 遵循项目规定的编码规范和风格。
2. 代码具有良好的可读性、可维护性和可扩展性。
3. 每个功能模块都应进行详细的单元测试，确保其正确性。
4. 代码审查过程应遵循项目规定的审查标准，确保代码质量。
5. 合并代码至主分支时，确保无冲突，遵循项目的版本控制规范。

6. 继承与测试阶段



6.1 活动名称及次序模块集成

1. 模块集成
2. 集成测试
3. 黑盒测试
4. 白盒测试
5. 报告缺陷
6. 修复缺陷
7. 缺陷重新测试
8. 测试报告撰写
9. 提交测试报告

6.2 活动执行人

1. 系统集成人员：负责软件模块的集成和集成测试。
2. 软件测试人员：负责编写自动化测试用例、执行自动化测试、手动测试、报告缺陷、撰写测试报告。
3. 软件开发人员：负责修复缺陷。

6.3 关键活动输入输出

1. 输入：可执行的软件程序、设计文档、需求规格说明书。
2. 输出：测试报告、已修复的软件。

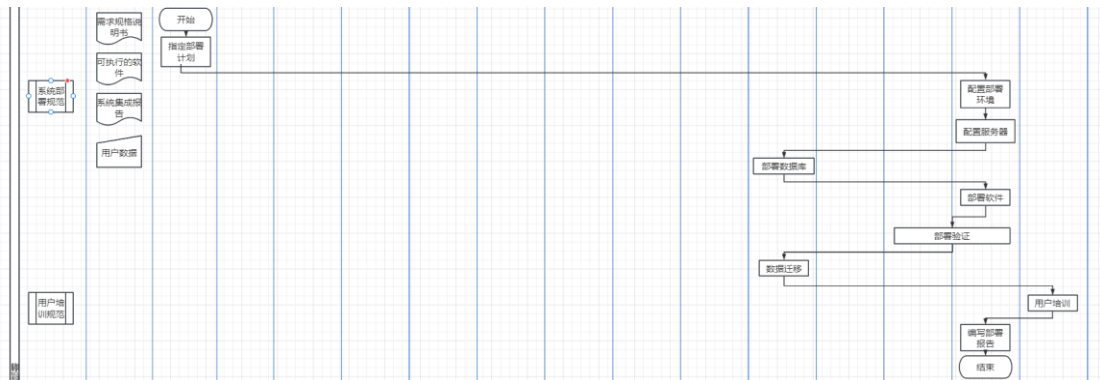
6.4 关键活动规范

软件集成规范，软件测试规范，关键有如下几点

1. 遵循项目规定的集成和测试策略。
2. 模块集成应确保正确的接口对接和数据传递。
3. 自动化测试用例应覆盖主要功能和关键路径。
4. 手动测试应关注用户体验和业务场景。
5. 缺陷报告应包含详细的复现步骤、预期结果和实际结果。
6. 修复缺陷后，应对修改后的软件进行重新测试以确保问题已解决。

7. 测试报告应详细记录测试过程、测试结果和遗留问题。

7. 部署阶段



7.1 活动名称及次序

1. 制定部署计划
2. 准备部署环境
3. 配置服务器和数据库
4. 部署软件
5. 验证部署
6. 数据迁移
7. 监控部署
8. 用户培训
9. 部署完成报告

7.2 活动执行人

1. 项目经理：负责制定部署计划。
2. 运维工程师：负责准备部署环境、配置服务器和数据库、部署软件、监控部署。
3. 测试工程师：负责验证部署。
4. 数据库管理员：负责数据迁移。
5. 培训师：负责用户培训。

7.3 关键活动输入输出

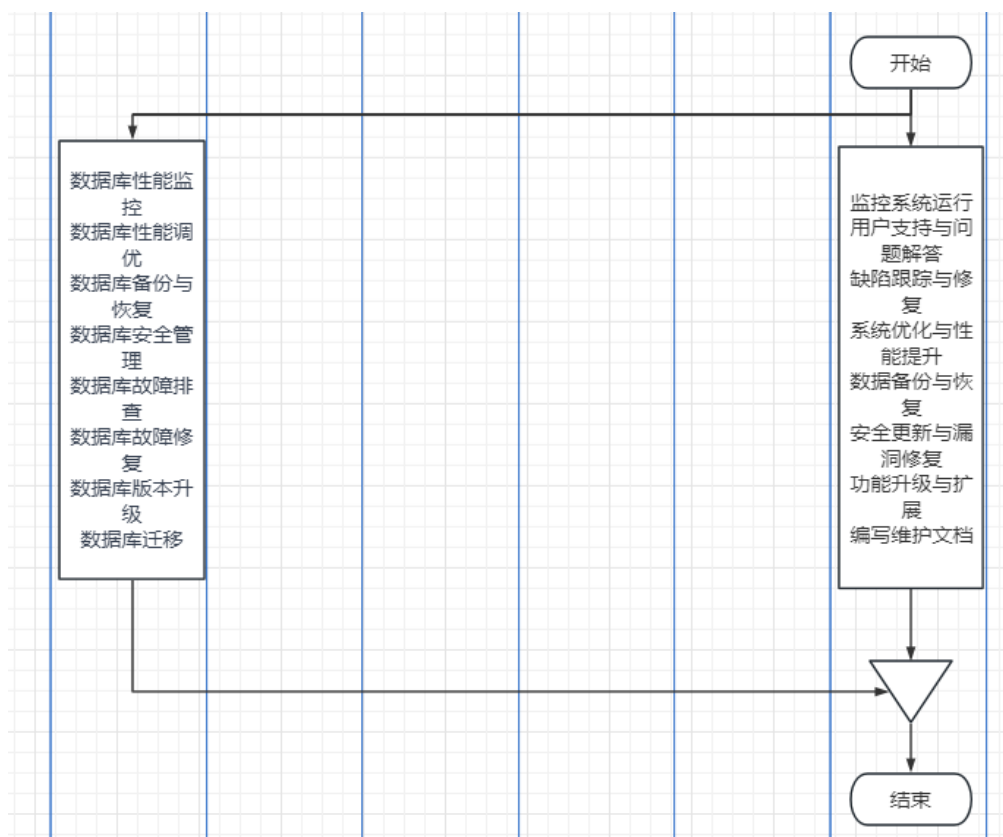
1. 输入：已测试和修复的软件、设计文档、需求规格说明书、测试报告。
2. 输出：部署完成报告、已部署的软件、培训资料。

7.4 关键活动规范

软件部署规范，用户培训规范，关键有如下几点

1. 遵循项目规定的部署策略和标准。
2. 部署环境应符合系统需求和兼容性要求。
3. 服务器和数据库配置应满足性能和安全需求。
4. 验证部署应确保软件正确运行和功能完整。
5. 数据迁移应确保数据的完整性和一致性。
6. 用户培训应包含软件操作和常见问题解答。

8. 维护阶段



8.1 活动名称及次序

下面的活动无次序之分。

1. 监控系统运行
2. 用户支持与问题解答
3. 缺陷跟踪与修复
4. 系统优化与性能提升
5. 数据备份与恢复
6. 安全更新与漏洞修复
7. 功能升级与扩展
8. 编写维护文档
9. 数据库性能监控与调优
10. 数据库备份与恢复
11. 数据库安全管理
12. 数据库故障排查与修复
13. 数据库架构优化
14. 数据库版本升级与迁移
15. 协助开发团队

8.2 活动执行人

1. 运维工程师：负责监控系统运行、系统优化与性能提升、数据备份与恢复、安全更新与漏洞修复。
2. 数据库管理员：数据库运维

8.3 关键活动输入输出

1. 输入：已部署的软件、设计文档、需求规格说明书、测试报告、用户反馈。
2. 输出：更新后的软件。

8.4 关键活动规范

软件维护规范，数据库维护规范，关键有如下几点

1. 遵循项目规定的维护策略和标准。
2. 系统运行监控应确保系统稳定性和可用性。
3. 用户支持应及时响应用户需求和问题。
4. 缺陷修复应在规定的时间内完成。
5. 系统优化应提高软件性能和用户体验。
6. 数据备份与恢复应确保数据的安全性和完整性。
7. 安全更新与漏洞修复应遵循安全最佳实践。
8. 功能升级与扩展应与项目需求和目标保持一致。

9. 项目管理与团队协作

9.1 项目管理方法

在点菜宝项目中，采用瀑布模型进行项目管理。瀑布模型是一种线性顺序的软件开发过程，从需求分析到设计、实现、集成与测试、部署和维护。该模型的特点是在一个阶段完成后，才能进入下一个阶段。瀑布模型适用于需求明确、稳定的项目。

9.2 团队协作

团队协作工具和流程在项目中起到关键作用，它们有助于提高团队沟通和协作效率。点菜宝项目团队可采用以下工具与流程：

项目管理工具：例如 Jira、Trello 或 Microsoft Project 等，用于规划、分配和跟踪项目任务。

文档协作工具：例如 Confluence、Google Docs 或 Microsoft SharePoint 等，用于创建、共享和编辑项目文档。

代码托管与版本控制工具：例如 Git、Subversion 或 Mercurial 等，用于代码的版本控制、合并和共享。

沟通工具：例如飞书等，用于团队之间的实时沟通与讨论。

流程方面，团队成员需定期开展项目进度会议，及时报告工作进度和遇到的

问题。项目经理应密切关注项目的整体进展，并在必要时进行调整。

9.3 变更管理

变更管理是项目管理的重要组成部分，旨在确保项目中的变更得到有效控制。请参考变更管理流程图，及其相关文档。

10. 结论

本文档详细阐述了点菜宝项目采用瀑布模型进行管理的过程。通过对每个阶段的活动、活动执行人、关键活动输入输出、关键活动规范以及它们之间的关系进行分析，为项目团队提供了清晰的指导。同时，文档还简单介绍了项目管理与团队协作的方法、工具和流程，以及变更管理的策略，确保项目的顺利进行。

点菜宝项目作为一个以瀑布模型为基础的软件开发项目，通过严格按照各阶段的活动规范和标准，保证了项目的质量和进度。在今后的开发过程中，项目团队应继续遵循本文档所描述的管理方法和规范，确保项目能够按时交付并满足用户需求与约束。