



# 《基于优先级队列的 TO DO 软件》

## 可行性分析报告

姓名： 姚宏伟

专业： 计算机科学技术

班级： B21 计科 10 班

指导教师： 孙 红

2023 年 3 月

# 一、引言

## （一）背景

在快节奏的现代生活中，个人和专业任务管理变得日益重要。随着数字化转型的加速，市场上涌现了各种 To Do 类软件，旨在帮助用户高效地规划和跟踪他们的日常活动。然而，这些软件大多受到了最初的“奇妙清单”模式的影响，该模式在灵活性和优先级管理方面存在明显的不足。

本项目的产生背景基于对现有 To Do 软件的深入分析和市场调研，发现用户尤其是学生群体在使用这些工具时面临着任务积累和优先级混乱的问题。这些问题不仅影响了用户的工作效率，也对他们的心理健康产生了负面影响。因此，本项目旨在开发一种新的任务管理模式，通过技术创新解决现有模式的局限，提供一个更加灵活和高效的任务管理解决方案。

## （二）项目概述

- 项目适用环境
  - 目的：创建一个真正的“代办事项” Todo 软件，避免“奇妙清单”式的日程安排模式，专注于非日程安排类任务。
  - 适用范围：理论上适用于所有原有 Todo 软件能承担的任务类型。
- 软件用途
  - 功能改进：继承 Todo 类软件的基本用途，同时引入创新功能，提升用户体验和效率。
- 项目运行环境
  - 跨平台兼容性：软件设计为跨平台工作，以适应不同设备和工作环境的需求。

### （三）文档概述

本文档存在的意义为提出、研究、与探讨这类新式 Todo 软件的概念与可行性，以及为之后的开发环节作出清晰合理的规划，本文档的读者应当遵守保密的要求，在软件初步开发完成之前，文档的读者不可将文档内容在未经作者的允许的情况下泄露给第三方。

## 二、引用文件

## 三、可行性分析的前提

### （一）项目的要求

- 技术要求：软件必须能够在包括 Windows、macOS、Linux、iOS 和 Android 在内的各种操作系统上无缝运行。此外，应用程序应具备高度的用户交互性和友好的用户界面。
- 性能要求：软件应能快速响应用户操作，即使在处理大量任务时也能保持稳定。同时，必须实现跨平台的数据同步功能，确保用户在不同设备间的任务状态能够实时更新。
- 安全要求：必须确保用户数据的安全性和隐私保护，遵守相关的数据保护法规。

### （二）项目的目标

- 主要目标：开发一个以优先级队列为基础的任务管理软件，能够帮助用户根据任务的重要程度进行有效规划。
- 次要目标：提高用户的任务管理效率，减少任务积累，避免优先级混乱。

### （三）项目的环境、条件、假定和限制

- 环境：项目将在多平台环境中开发，包括桌面和移动操作系统，以适应不同用户的需求。

- 条件：项目开发需要一支具备跨平台开发经验的团队，并且有足够的时间和资源来完成开发。同时，必须有能够实现数据同步的技术支持。
- 假定：假设用户愿意尝试新的任务管理方法，并且市场对于这种新型软件有需求。
- 限制：项目可能受到时间和预算的限制，以及现有技术的限制。特别是数据同步功能可能需要额外的时间和资源来确保其稳定性和安全性。

#### （四）进行可行性分析的方法

- 技术可行性：评估软件的技术实现可能性，包括所需技术的可用性和团队的技术能力。
- 经济可行性：分析项目的成本效益，确保投入产出比合理。
- 法律可行性：考虑软件开发过程中可能涉及的法律问题，如版权、专利和隐私法规。

### 四、可选的方案

#### （一）原有方案对比

- 微软 Todos 软件
  - 优点：提供了基本的任务管理功能，如任务创建、分类和提醒，用户界面简洁直观。
  - 缺点：缺少灵活的时间规划和优先级队列功能，无法自动优化任务顺序，导致用户在任务管理上遇到困难。

#### （二）可复用部分

- API 复用：我们的系统可以复用微软的 API，将经过我们系统优化后的任务列表推送到 Todos 软件，作为当天的日程安排，以解决跨平台兼容的问题。此外，在周

期结束时，我们的系统将读取任务的完成情况，并根据用户的实际进度重新安排下一个周期的任务，从而实现更加个性化和高效的任务管理。

### **（三）可选择的方案 1：基于优先级的动态任务管理系统**

- 设计：开发一个动态调整任务优先级的系统，它会根据任务的紧急程度、重要性以及用户的行为模式自动排序任务。
- 特点：系统将提供一个灵活的界面，允许用户根据个人偏好调整优先级算法的参数，以适应不同的工作和生活方式。

### **（四）可选择的方案 2：基于优先级的智能任务推荐系统**

- 设计：创建一个智能系统，它不仅根据优先级队列管理任务，还能根据用户的历史数据和习惯推荐最佳完成任务的时间。
- 特点：系统将使用机器学习算法来预测用户的空闲时间，并在最适合的时刻提醒用户完成高优先级的任务。

### **（五）选择最终方案的准则**

- 用户体验：在选择最终方案时，我们将重点考虑系统的易用性、对用户日常生活的实际帮助，以及系统的可维护性和可扩展性。
- 技术可行性：评估系统的技术实现难度，确保所选方案能够在预定的时间和预算内完成。
- 市场需求：考虑市场对新型任务管理工具的需求，选择能够满足用户需求并具有市场竞争力的方案。

## **五、所建议的系统**

### **（一）对所建议系统的说明**

本系统旨在通过优先级队列的方法，提供一个更加灵活和高效的任务管理工具。与传

统的 To Do 软件相比，本系统将允许用户根据任务的重要性和紧急性来动态调整任务的优先级，而不是简单地按照时间顺序排列。此外，系统将采用先进的算法来预测用户的行为模式，并据此优化任务的安排，使用户能够更加合理地规划时间和精力。

## （二）数据流程和处理流程

数据流程将从用户输入任务开始，系统将根据任务的属性（如截止日期、重要性标签等）将其放入相应的优先级队列中。用户界面将实时显示当前的任务列表，用户可以随时调整任务的属性或优先级。处理流程包括任务的自动排序、提醒设置以及与用户日历的同步等功能。系统还将记录用户的任务完成情况，以便未来的行为预测和任务优化。

## （三）与原系统的比较（若有原系统）

与微软的 Todos 软件相比，本系统在任务管理的灵活性和智能性方面进行了显著的改进。例如，本系统不仅提供了基于优先级的任务排序，还引入了基于用户行为的智能推荐功能。此外，本系统还支持跨平台的数据同步，确保用户在任何设备上都能访问最新的任务列表。

## （四）影响（或要求）

### 1. 设备

- 数据同步：为实现跨平台数据同步，初步考虑使用 git 存储库系统。但对于商业化使用，需要考虑购买云存储服务，以保证数据的安全和可靠同步。

### 2. 软件

- 技术栈选择：开发软件的选择将取决于项目的具体需求和所选的编程语言。
- 商业许可证：如果项目用于商业目的，必须购买相应的开发软件商业许可证。

### 3. 运行

- API 维护：需要考虑 API 更新可能带来的维护成本。
- 服务器成本：内容同步服务器可能产生的流量和存储成本。
- 人员工资：开发和维护团队的潜在工资开销。

### 4. 开发

- 编程语言：考虑使用 Rust 或 C#等跨平台兼容性强的语言。如果需要前端，则可能需要额外的前端技术栈。
- 技术选择：可以考虑 Java/Kotlin 等 JVM 工具链，或者 C++，但后者可能会增加开发成本。
- 简化前端：考虑使用 Todos API 时，可能省略复杂的前端开发，转而使用简单的命令行界面（CLI）作为管理页面。

### 5. 环境

- 跨平台：软件应能够在 Windows、macOS、Linux、iOS 和 Android 等多个平台上运行。

### 6. 经费

- 开源项目：作为公益开源项目启动，内置官方提供的云同步方案作为可选收费服务。
- 收费模型：收费价格将基于服务器租赁成本，并加上适当比例以保证盈利。
- 赞助：接受无偿赞助，若转为商业用途，将根据开发人员和资源配置计算成本，并按市场标准支付工资。

## 六、经济可行性

### （一）投资

### （二）预期的经济收益

1. 一次性收益
2. 非一次性收益
3. 不可定量的收益
4. 收益/投资比
5. 投资回收周期

### （三）市场预测

## 七、技术可行性

## 八、法律可行性

## 九、用户使用可行性

## 一〇、其他的与项目有关的问题

## 一一、注解

## 一二、附录