**ПРОЕКТ**

ОГЛАВЛЕНИЕ

[Название проекта 2](#_Toc131172267)

[Цели проекта 2](#_Toc131172268)

[Результаты проекта 2](#_Toc131172269)

[Критерии достижения цели 2](#_Toc131172270)

[Допущения 3](#_Toc131172271)

[Ограничения 3](#_Toc131172272)

[Альтернативные способы достижения цели 3](#_Toc131172273)

[Модель разработки 3](#_Toc131172274)

[Критерии: 4](#_Toc131172275)

[Обоснование выбора: 4](#_Toc131172276)

[Методология разработки 4](#_Toc131172277)

[Критерии: 4](#_Toc131172278)

[Обоснование выбора: 4](#_Toc131172279)

[Технологии для реализации проекта 5](#_Toc131172280)

[Основные этапы разработки проекта 5](#_Toc131172281)

# Название проекта

Task Manager с отслеживанием времени выполнения задач.

# Цели проекта

* Обеспечение возможности отслеживания выполнения каждой задачи для контроля своевременного их выполнения;
* Упрощение и ускорение процесса распределения задач;
* Обеспечение возможности создания отчетов для оценки производительности разработчика.

# Результаты проекта

* Разработанное и функционирующее приложение "Task Manager с трекером времени выполнения задач".
* База данных приложения, содержащая две таблицы:

а) таблицу проектов, которая содержит информацию о проектах, их описании, статусе выполнения, крайнего срока выполнения;

б) таблицу задач, которая содержит информацию о задачах, их описании, статусе выполнения, крайнего срока выполнения, трекера времени;

* Обеспечение качества продукта модульным тестированием, чтобы гарантировать корректность функционирования приложения;
* Увеличение эффективности работы пользователя за счет отслеживания времени, затраченного на выполнение задач и проектов;

# Критерии достижения цели

* Стабильная работа приложения без сбоев и ошибок;
* Реализация функции отслеживания времени выполнения задач, которая позволит пользователям контролировать время, затраченное на каждую задачу;
* Увеличение производительности и эффективности работы пользователя на определенный процент;
* Снижение количества просроченных задач и увеличение количества выполненных задач за определенный период времени;
* Сокращение времени, затраченного на прогнозирование и планирование задач и проектов;
* Улучшение контроля над процессом выполнения задач и проектов за счет системы уведомлений о дедлайнах, изменениях статуса задач и отчетов о производительности разработчика, включающих в себя общее количество выполненных задач, затраченное время и достигнутые результаты.
* Уменьшение количества ошибок и несоответствий в процессе выполнения задач и проектов.

# Допущения

* Целевая платформа приложения – ПК на ОС Windows;
* Microsoft не прекратит поддержку платформы .NET;
* Приложением будет пользоваться один пользователь;
* Пользователь будет использовать приложение только на одном устройстве;
* Пользователь будет иметь возможность создавать, редактировать и удалять задачи и проекты в приложении;

# Ограничения

* Приложение разрабатывается на платформе .NET;
* Разрабатываемое приложение доступно только одному пользователю;
* База данных будет храниться локально на компьютере пользователя;
* Запуск и остановка отслеживания выполнения задач, а также их создание будут выполняться вручную пользователем.

# Альтернативные способы достижения цели

* Использование готовых решений, которые уже интегрированы с нужными системами;
* Разработка приложения с использованием другого языка программирования;
* Создание веб-приложения, чтобы решить проблему обеспечения доступа к приложению из разных мест;
* Создание таблиц в Excel и отслеживание выполнения задач вручную.

# Модель разработки

Выбрана инкрементная модель разработки, так как требования для системы достаточно ясны и определены заранее. Она подразумевает разбиение проекта на небольшие части (инкременты), каждый из которых разрабатывается отдельно и может быть протестирован и принят заказчиком.

## Критерии:

* Снижение рисков;
* Жесткий график;
* Возможность промежуточной оценки результатов.

## Обоснование выбора:

Основное обоснование выбора инкрементной модели разработки заключается в простоте её использования, возможности сдачи проекта по частям. Она хорошо подходит для небольших проектов, в которых требования достаточно определены на старте, но могут измениться в процессе.

# Методология разработки

Выбрана методология Kanban, так как она является одной из популярных методологий гибкой разработки программного обеспечения и основана на принципе управления потоком работы и визуализации процесса разработки.

## Критерии:

* Гибкость: Kanban является гибкой методологией, которая позволяет быстро реагировать на изменения в проекте;
* Необходимость обеспечить четкое планирование и контроль над процессом разработки: Kanban предоставляет возможность визуализировать процесс разработки, что позволяет лучше понимать, какие задачи нужно выполнить, и как они связаны между собой;
* Временные ограничения разработки: методология Kanban позволяет эффективно управлять потоком работы, контролировать его и оптимизировать, тем самым экономя время разработки.

## Обоснование выбора:

Выбор методологии Kanbanобосновывается тем, что она позволяет эффективно управлять потоком работы и визуализировать процесс разработки. Это особенно важно для проектов, где требуется быстро реагировать на изменения и адаптироваться к новым условиям. Кроме того, Kanban является очень гибкой методологией, которая позволяет **разработчику** более эффективно работать над улучшением качества продукта. Наконец, простота в использовании способствует более эффективной работе и повышению производительности.

# Технологии для реализации проекта

* .NET Framework – платформа для разработки приложений на языке C#;
* WPF (Windows Presentation Foundation) – библиотека для создания графического интерфейса пользователя;
* MVVM (Model-View-ViewModel) – архитектурный шаблон для разработки приложений;
* NUnit – фреймворк для написания автоматических тестов;
* Git – система контроля версий, которая позволяет отслеживать изменения в коде и управлять ими.

# Основные этапы разработки проекта

* Анализ требований: определение функциональных требований к приложению, определение требований к его производительности и удобству использования;
* Проектирование: разработка архитектуры приложения, определение интерфейса пользователя и проектирование базы данных;
* Разработка: непосредственная разработка приложения с использованием выбранной методологии и инструментов;
* Тестирование: проведение модульного и интеграционного тестирования.
* Релиз: готовность приложения к выпуску и его установка на компьютеры пользователей;
* Поддержка: техническая поддержка приложения, включая исправление ошибок и улучшение функциональности.