**Содержание**

[1 Введение 2](#_Toc193409938)

[2 Общие сведения 4](#_Toc193409939)

[3 Назначение и цели разработки 5](#_Toc193409940)

[3.1 Функциональное назначение 5](#_Toc193409941)

[4 Требования к программе или программному изделию 6](#_Toc193409942)

[4.1 Требования к функциональным характеристикам 6](#_Toc193409943)

[4.1.1 Требования к составу выполняемых функций 6](#_Toc193409944)

[4.2 Требования к надежности 6](#_Toc193409945)

[4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы 6](#_Toc193409946)

[4.2.2 Время восстановления после отказа 7](#_Toc193409947)

[4.2.3 Отказы из-за некорректных действий оператора 8](#_Toc193409948)

[4.3 Условия эксплуатации 8](#_Toc193409949)

[4.3.1 Климатические условия эксплуатации 8](#_Toc193409950)

[4.3.2Требования к видам обслуживания 9](#_Toc193409951)

[4.4 Требования к составу и параметрам технических средств 9](#_Toc193409952)

[4.5 Требования к маркировке и упаковке 9](#_Toc193409953)

[4.6 Требования к транспортированию и хранению 9](#_Toc193409954)

[4.7 Специальные требования 9](#_Toc193409955)

[5 Требования к программной документации 11](#_Toc193409956)

[5.1 Предварительный состав программной документации 11](#_Toc193409957)

[6 Стадии и этапы разработки 12](#_Toc193409958)

[7 Порядок контроля и приемки 14](#_Toc193409959)

# **1 Введение**

В современных условиях управления проектами системы управления проектами (СУП) становятся ключевым инструментом для планирования, организации и контроля выполнения задач. Основная цель СУП — обеспечить эффективное управление проектами различного уровня сложности, минимизировать риски, повысить прозрачность процессов и улучшить взаимодействие между всеми участниками проекта.

СУП позволяет структурировать процесс управления проектом, делая его последовательным и управляемым. Она предоставляет возможности для детального планирования, распределения ресурсов, управления сроками и контроля выполнения задач. Внедрение СУП способствует повышению эффективности работы команд, позволяет более точно оценивать затраты и сроки реализации проектов, а также своевременно реагировать на возникающие риски.

В рамках данного проекта методы Agile рассматриваются как вспомогательное средство, дополняющее возможности СУП и позволяющее адаптироваться к изменяющимся требованиям и обстоятельствам. Agile-подходы, такие как Scrum и Kanban, применяются для повышения гибкости управления, обеспечения оперативной обратной связи и повышения удовлетворенности заказчика. Использование этих методологий в сочетании с СУП позволяет командам оперативно реагировать на изменения, минимизировать влияние рисков и поддерживать высокое качество результатов.

Целью настоящего технического задания является разработка системы управления проектами, которая опирается на принципы СУП и использует методы Agile для повышения адаптивности и гибкости процессов. Такая система обеспечит комплексный подход к управлению проектами, поддерживая последовательность и структурированность СУП, а также гибкость и динамичность Agile.

Основные области применения системы:

* IT-проекты: управление разработкой программного обеспечения с учетом гибких подходов к изменяющимся требованиям.
* Образовательные проекты: планирование и контроль выполнения задач в образовательных учреждениях и учебных центрах.
* Корпоративное управление: организация работы команд в рамках корпоративных проектов и инициатив.

Система, разрабатываемая в рамках данного проекта, станет универсальным инструментом для повышения эффективности управления проектами, сокращения времени выполнения задач и улучшения взаимодействия между всеми участниками проекта. В рамках данного технического задания будут определены ключевые требования к функциональности, архитектуре, интерфейсу и интеграционным возможностям системы, а также рассмотрены аспекты ее внедрения и поддержки.

# **Общие сведения**

Разрабатываемая система управления проектами (СУП) предназначена для обеспечения эффективного планирования, контроля и управления проектами различного уровня сложности. Основное функциональное назначение системы заключается в поддержке структурированного подхода к управлению проектами, включая планирование задач, распределение ресурсов и контроль выполнения.

В качестве вспомогательного средства система интегрирует методы Agile, такие как Scrum и Kanban, что позволяет повысить гибкость управления, адаптироваться к изменениям и обеспечивать оперативную обратную связь. Такое сочетание подходов дает возможность сочетать последовательность и структурированность СУП с гибкостью и адаптивностью Agile.

Система предоставляет возможности для взаимодействия и коммуникации между участниками проекта, включая комментирование задач, обмен сообщениями, формирование отчетов и аналитики.

Система предназначена для применения в различных организациях и командах, которые нуждаются в надежном инструменте для управления проектами. Она поддерживает масштабируемость и адаптируется под проекты различного масштаба — от небольших команд до крупных организаций. Высокая производительность, безопасность данных и интуитивно понятный интерфейс делают систему эффективным решением для комплексного управления проектами и повышения качества взаимодействия между участниками.

# **Назначение и цели разработки**

## **Функциональное назначение**

Система управления проектами предоставляет возможность создания, редактирования и назначения задач участникам проекта, а также отслеживания их статуса выполнения, например, "To Do", "In Progress" или "Done". Пользователи могут приоритизировать задачи и распределять их по спринтам, что обеспечивает гибкость в управлении проектами. Для улучшения взаимодействия между участниками система включает инструменты для обсуждения задач, такие как комментарии, упоминания и уведомления, а также поддерживает возможность совместной работы над задачами в режиме реального времени.

Кроме того, система предоставляет функциональность для формирования отчетов о выполнении задач, прогрессе проекта и использовании ресурсов. Это позволяет анализировать ключевые метрики, такие как скорость выполнения задач, что способствует постоянному улучшению процессов и повышению эффективности работы команды.

**4 Требования к программе или программному изделию**

**4.1 Требования к функциональным характеристикам**

4.1.1 Требования к составу выполняемых функций

**Управление задачами**:

* Создание, редактирование и назначение задач участникам проекта.
* Отслеживание статуса выполнения задач (например, "To Do", "In Progress", "Done").
* Возможность приоритизации задач и их распределения по спринтам

**Взаимодействие и коммуникация**:

* Интеграция инструментов для обсуждения задач (комментарии, упоминания, уведомления).
* Возможность совместной работы над задачами в режиме реального времени.

**Аналитика и отчетность:**

* Формирование отчетов о выполнении задач, прогрессе проекта и использовании ресурсов.
* Анализ ключевых метрик для улучшения процессов.

**4.2 Требования к надежности**

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Устойчивость к нагрузкам:

* Система должна быть способна обрабатывать высокие нагрузки пользовательского трафика без значительного снижения производительности.
* Проведение тестирования на прочность и масштабируемость для определения максимальной нагрузочной способности системы.

Целостность данных:

* Обеспечение целостности данных путем использования методов проверки целостности и шифрования данных во время их передачи и хранения.

Безопасность:

* Реализация мер безопасности, включая механизмы аутентификации и авторизации пользователей, контроль доступа к данным и мониторинг безопасности системы.
* Использование современных методов защиты от вредоносного программного обеспечения и атак извне, таких как защита от DDoS-атак и SQL-инъекций.

Мониторинг и аналитика:

* Реализация системы мониторинга и аналитики для постоянного контроля за работой системы, выявления аномалий и быстрого реагирования на потенциальные угрозы и проблемы.

4.2.2 Время восстановления после отказа

Обеспечение быстрого и эффективного восстановления работы системы после сбоев или атак с минимальным временем простоя для пользователей является критически важным аспектом проектирования и эксплуатации современных информационных систем. В рамках дипломной работы необходимо разработать и реализовать механизмы, которые позволят минимизировать время восстановления системы после возникновения нештатных ситуаций, таких как аппаратные сбои, программные ошибки, кибератаки или человеческие ошибки.

Для достижения этой цели предлагается использовать следующие подходы:

Резервное копирование и восстановление данных. Регулярное создание резервных копий критически важных данных с использованием инкрементальных и дифференциальных методов для минимизации времени восстановления.

Реализация отказоустойчивых архитектур. Использование кластеризации, репликации данных и распределенных систем, которые позволяют системе продолжать работу даже при выходе из строя отдельных компонентов.

Автоматизация процессов восстановления. Разработка сценариев и инструментов для автоматического обнаружения сбоев, устранения их последствий и восстановления работоспособности системы без необходимости вмешательства человека.

Мониторинг и анализ инцидентов. Внедрение систем мониторинга, которые позволят оперативно выявлять сбои и атаки, а также анализировать их причины для предотвращения повторных инцидентов.

Тестирование восстановления. Проведение регулярных тестовых учений по восстановлению системы после сбоев для проверки эффективности разработанных механизмов и выявления возможных узких мест.

4.2.3 Отказы из-за некорректных действий оператора

Отказы, вызванные некорректными действиями оператора, являются одной из распространенных причин сбоев в работе информационных систем. Такие отказы могут возникать из-за ошибок ввода данных, неправильной настройки параметров системы, случайного удаления или изменения критически важных файлов, а также из-за несанкционированных действий, вызванных недостатком знаний или невнимательностью оператора. Для минимизации рисков, связанных с человеческим фактором, в рамках дипломной работы предлагается реализовать комплекс мер, направленных на предотвращение и устранение последствий таких отказов.

**4.3 Условия эксплуатации**

4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

4.3.2Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

**4.4 Требования к составу и параметрам технических средств**

Для компьютеров:

Минимальные требования:

Процессор: Intel Core i3 или аналогичный.

Оперативная память: 4 ГБ DDR4.

Хранилище данных: SSD или HDD с емкостью не менее 128 ГБ.

Графический процессор: Интегрированный.

Рекомендуемые требования:

Процессор: Intel Core i5 или аналогичный.

Оперативная память: 8 ГБ DDR4.

Хранилище данных: SSD с емкостью не менее 256 ГБ.

Графический процессор: Дискретный или интегрированный с поддержкой OpenGL.

**4.5 Требования к маркировке и упаковке**

Дополнительных требований к маркировке и упаковке не предъявляется.

**4.6 Требования к транспортированию и хранению**

Требования к транспортированию и хранению не предъявляются.

**4.7 Специальные требования**

Разрабатываемая система не предъявляет специальных требований к аппаратному или программному обеспечению, что делает ее универсальной и легко адаптируемой для различных условий эксплуатации. Система спроектирована с учетом принципов кроссплатформенности и может быть запущена на любом современном оборудовании, поддерживающем стандартные операционные системы, такие как Windows, Linux или macOS. Для работы системы не требуется специализированное оборудование или уникальные условия, что значительно упрощает ее внедрение и использование.

Архитектура системы построена на основе модульного подхода, что позволяет гибко настраивать ее под конкретные задачи без необходимости изменения исходного кода. Система не зависит от конкретных версий программного обеспечения или библиотек, так как использует стандартные протоколы и интерфейсы взаимодействия. Это обеспечивает ее совместимость с большинством существующих решений и платформ.

Кроме того, система не требует наличия выделенных серверов или мощных вычислительных ресурсов. Она может быть развернута как на локальных машинах, так и в облачных средах, что делает ее доступной для широкого круга пользователей, включая малые и средние предприятия. Благодаря минимальным требованиям к инфраструктуре, система может быть запущена практически в любом месте, где есть доступ к стандартным вычислительным ресурсам и интернету (если требуется сетевое взаимодействие).

Таким образом, отсутствие специальных требований к оборудованию, программному обеспечению или условиям эксплуатации делает систему максимально гибкой и удобной для использования в различных сценариях, что является одним из ее ключевых преимуществ.

**5 Требования к программной документации**

**5.1 Предварительный состав программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* руководство пользователя;
* листинг программы;
* руководство системного программиста
* пояснительная записка

Специальные требования к программной документации не предъявляются.

**6 Стадии и этапы разработки**

Разработка должна быть проведена в три стадии:

1. техническое задание;

2. технический (и рабочий) проекты;

3. внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* постановка задачи;
* определение и уточнение требований к техническим средствам;
* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и документации на нее;
* согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

разработка, согласование и утверждение порядка и методики испытаний;

проведение приемо-сдаточных испытаний;

корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

| **Стадия разработки** | **Этап разработки** | **Сроки исполнения** | **Ответственный** |
| --- | --- | --- | --- |
|  | | | |
| Эскизный проект | Разработка эскизного проекта |  | Квашонкин Никита Романович |
| Согласование и утверждение эскизного проекта |  | Квашонкин Никита Романович |
| Технический проект | Разработка технического проекта |  | Квашонкин Никита Романович |
| Утверждение технического проекта |  | Квашонкин Никита Романович |
| Рабочий проект | Разработка программы |  | Квашонкин Никита Романович |
| Разработка программной документации |  | Квашонкин Никита Романович |
| Испытания программы |  | Квашонкин Никита Романович |
| Внедрение | Подготовка и передача программы |  | Квашонкин Никита Романович |

**7 Порядок контроля и приемки**

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.