|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | Логотип КБК копия | **Автономная некоммерческая организация профессионального образования**  **КАЛИНИНГРАДСКИЙ БИЗНЕС-КОЛЛЕДЖ** |   **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  **Специальность** 09.02.07 Информационные системы и программирование  **Тема: «Разработка модуля рабочих заданий для инструмента формирования недельно-суточных заданий с внедрением диаграммы Ганта»**  **Выполнил:** студент группы 17-ИСП-4  Зуев Виталий Сергеевич  **Руководитель ВКР**  Дубинин Андрей Валентинович  ***«Допустить к защите»***  Заведующий кафедрой информационных технологий  Ю.А. Япарова  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Дата защиты «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.  Оценка: «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» |

|  |  |
| --- | --- |
| Логотип КБК копия | **Автономная некоммерческая организация профессионального образования**  **КАЛИНИНГРАДСКИЙ БИЗНЕС-КОЛЛЕДЖ** |

Специальность 09.02.07 Информационные системы и программирование  
Форма обучения: Очная  
Группа 17-ИСП-4

**ЗАДАНИЕ**  
по выпускной квалификационной работе

студенту Зуеву Виталию Сергеевичу

1. Тема выпускной квалификационной работы «Разработка модуля рабочих заданий для инструмента формирования недельно-суточных заданий с внедрением диаграммы Ганта»

2. Срок сдачи готовой работы на кафедру «10» июня 2020 г.

3. Цели работы (исходные данные): разработать модуль рабочих заданий с внедрением диаграммы Ганта для настольного приложения «Инструмент формирования недельно-суточных заданий»

4. Содержание разделов выпускной квалификационной работы

1. Анализ проблемы и постановка задачи дипломного проекта
   1. Анализ объекта исследования
      1. Анализ неэффективного управления проектом
      2. Анализ влияния планирования на проект с целью улучшения его эффективности
   2. Постановка задачи дипломного проекта
      1. Анализ субъекта
      2. Постановка задачи
   3. Выбор языка и инструментария разработки
      1. Выбор языка
      2. Выбор инструментария
2. Анализ требований и определение спецификаций
   1. Выбор парадигмы программирования
   2. Декомпозиция задачи
   3. Диаграмма прецедентов
   4. Диаграмма деятельности для расшифровки содержания прецедентов
   5. Диаграмма состояний
   6. Диаграмма классов
   7. Требования к программе
3. Проектирование программного продукта
   1. Архитектура приложения
      1. Подсистема для работы с рабочими заданиями
      2. Подсистема команд
      3. Подсистема пользовательского интерфейса
   2. Диаграммы архитектуры приложения
      1. Диаграмма компонентов
      2. Диаграмма объектов
      3. Обзорная диаграмма взаимодействия
      4. Диаграмма пакетов
4. Реализация
   1. Интерфейс пользователя
5. Тестирование программного продукта
6. Экономическое обоснование

5. Перечень графического и (или) иллюстрационного материала:

Диаграмма прецедентов

Диаграмма деятельности

Диаграмма состояний

Диаграмма классов

Диаграмма последовательностей

Диаграмма компонентов

Диаграмма объектов

Обзорная диаграмма взаимодействия

Диаграмма пакетов

Экранные формы модуля для настольного приложения

Снимки кода тестов

6. Консультанты по разделам выпускной квалификационной работы

|  |  |  |  |
| --- | --- | --- | --- |
| Раздел | Ф.И.О. Консультанта | Подпись | Дата |
| Анализ проблемы и постановка задачи дипломного проекта | Дубинин А.В. |  | 15.02.2020 |
| Анализ требований и определение спецификаций | Дубинин А.В. |  | 06.03.2020 |
| Проектирование программного продукта | Дубинин А.В. |  | 27.03.2020 |
| Реализация | Дубинин А.В. |  | 15.05.2020 |
| Тестирование программного продукта | Дубинин А.В. |  | 29.05.2020 |
| Экономическое обоснование | Дубинин А.В. |  | 05.06.2020 |

7. Дата выдачи задания «10» февраля 2020 г.

8. Руководитель выпускной квалификационной работы \_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

9. Задание получил «10» февраля 2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

# Содержание

[Содержание 5](#_Toc43674139)

[Введение 7](#_Toc43674140)

[1. Анализ проблемы и постановка задачи дипломного проекта 9](#_Toc43674141)

[1.1. Анализ объекта исследования 9](#_Toc43674142)

[1.1.1. Анализ неэффективного управления проектом 9](#_Toc43674143)

[1.1.2. Анализ влияния планирования на проект с целью улучшения его эффективности 11](#_Toc43674144)

[1.2. Постановка задачи дипломного проекта 14](#_Toc43674145)

[1.2.1. Анализ субъекта 14](#_Toc43674146)

[1.2.2. Постановка задачи 17](#_Toc43674147)

[1.3. Выбор языка и инструментария разработки 17](#_Toc43674148)

[1.3.1. Выбор языка 17](#_Toc43674149)

[1.3.2. Выбор инструментария 19](#_Toc43674150)

[2. Анализ требований и определение спецификаций 21](#_Toc43674151)

[2.1. Выбор парадигмы программирования 21](#_Toc43674152)

[2.2. Декомпозиция задачи 21](#_Toc43674153)

[2.3. Диаграмма прецедентов 23](#_Toc43674154)

[2.4. Диаграмма деятельности для расшифровки содержания прецедентов 24](#_Toc43674155)

[2.5. Диаграмма состояний 25](#_Toc43674156)

[2.6. Диаграмма классов 26](#_Toc43674157)

[2.7. Требования к программе 28](#_Toc43674158)

[3. Проектирование программного продукта 29](#_Toc43674159)

[3.1. Архитектура приложения 29](#_Toc43674160)

[3.1.1. Подсистема для работы с рабочими заданиями 29](#_Toc43674161)

[3.1.2. Подсистема команд 29](#_Toc43674162)

[3.1.3. Подсистема пользовательского интерфейса 30](#_Toc43674163)

[3.2. Диаграммы архитектуры приложения 31](#_Toc43674164)

[3.2.1. Диаграмма компонентов 31](#_Toc43674165)

[3.2.2. Диаграмма объектов 31](#_Toc43674166)

[3.2.3. Обзорная диаграмма взаимодействия 32](#_Toc43674167)

[3.2.4. Диаграмма пакетов 33](#_Toc43674168)

[4. Реализация 34](#_Toc43674169)

[4.1. Интерфейс пользователя 34](#_Toc43674170)

[5. Тестирование программного продукта 47](#_Toc43674171)

[6. Экономическое обоснование 51](#_Toc43674172)

[Заключение 58](#_Toc43674173)

[Список использованных источников 59](#_Toc43674174)

[Список сокращений 60](#_Toc43674175)

# Введение

Информационная система (далее ИС) – это система, предназначенная для хранения, поиска и обработки информации, а также соответствующие организационные ресурсы (человеческие, технические, финансовые и т.д.), обеспечивающие и распространяющие информацию.

Наиболее широко информационные системы и технологии используются в производственной, управленческой и финансовой деятельности. Например, в строительной отрасли.

Основной составляющей строительной отрасли (примерно 90% вложений) является стадия ведения проектов. В первую очередь это касается строительных и компаний-разработчиков, а также компаний, специализирующихся на развитии территорий и создании инфраструктуры строительства. Во многом проекты капитального строительства зданий и сооружений и создания сопутствующей инфраструктуры зависят от общего подхода к процессу. То, насколько качественно компания сможет управлять проектами, их рисками, сроками, бюджетом и многими другими составляющими, и будет влиять на бизнес в целом. Кроме человеческого фактора нельзя не принимать в расчет вспомогательные инструменты - информационные системы для управления строительством. Соответственно такие ИС должны являться интерактивным план-графиком подрядчика, в котором у пользователя будет возможность просматривать запланированные, действующие и выполненные работы, оценивать затраченные и необходимые в дальнейшем ресурсы, а также координировать работу строительных команд и выдавать заказчику требуемую информацию о состоянии строительства на различных его этапах. Однако, сама по себе ИС, работающая на основе таблиц, не так уж и удобна в использовании, поэтому для наглядности отслеживания процессов производства в проектах применяется диаграмма Ганта.

Диаграмма Ганта (a Gantt Chart) — это визуальный способ отображения запланированных задач. Горизонтальные графики широко используются для планирования проектов любых размеров в разных отраслях и сферах. Это удобный способ показать, какая работа планируется к выполнению в определенный день и время. Gantt Charts также помогают командам и менеджерам проектов контролировать даты начала и окончания любого проекта. Все в одном пространстве.

Большинство менеджеров проектов, знают о диаграммах Ганта. Диаграммы Ганта помогают повысить производительность и эффективность рабочих процессов и обеспечить своевременное выполнение задач. Такие графики могут быть применены к небольшим проектам и самым сложным.

Целью данной работы является разработка модуля рабочих заданий для инструмента формирования недельно-суточных заданий с внедрением диаграммы Ганта. Данный модуль будет позволять оптимально планировать затраты ресурсов и времени на выполнение поставленных задач, предоставление быстрого поиска и фильтрации по различным атрибутам работ, создание связей и цепочек работ для составления и отслеживания рабочего процесса.

# Анализ проблемы и постановка задачи дипломного проекта

# Анализ объекта исследования

# Анализ неэффективного управления проектом

Каждое предприятие своей целью ставит получение как можно большего дохода при минимальных затратах (такой подход называется бережливым производством), но добиться этого получается не всегда и зачастую появляются дополнительные расходы из-за различных внешних и внутренних факторов, основным из которых можно выделить недостаточное планирование и неэффективное управление проектом.

Основными проблемами, возникающими при реализации проекта, являются:

* Слабо-налаженные коммуникации между проектными отделами, что приводит к увеличению сроков разработки проекта;
* Нехватка имеющихся ресурсов;
* Конфликты, возникающие при распределении имеющихся ресурсов, если организация занимается осуществлением нескольких проектов одновременно;
* Невыполнение в срок;
* Превышение установленного бюджета;
* Потеря актуальности долгосрочных проектов к моменту их завершения;
* Несоответствие стратегиям организации;
* Недооценка влияния возможных рисков.

Эти проблемы, как последствие, приводят к потерям. Потери – это операции, на которые затрачиваются временные и материальные ресурсы без добавления ценности товару или услуге для конечного потребителя. В соответствии с концепцией бережливого производства в бизнес-процессах возникают следующие виды потерь:

* Неиспользование потенциала сотрудников. (Основополагающий фактор эффективности непрерывного совершенствования –вовлечение в этот процесс всего персонала компании. Знание особенностей сотрудников и грамотное использование их особенностей способно значительно увеличить скорость и качество производства. Так же к этому критерию можно и отнести бездействие сотрудников при отсутствии соответствующей их специализации задачи);
* Потери на транспортировку. (Потери в результате перемещения материалов и заготовок между производственными участками и цехами, не добавляющими ценности конечному продукту. По отношению к строительным проектам, это обычно несвоевременный подвоз материалов, инструментов и оборудования по причине некорректного планирования, что ведет к потерям на обслуживание запасов);
* Потери от брака, излишних отходов и переделок. (Типичными примерами могут служить ошибки в документах и доработки, некорректный ввод данных в информационные системы, несогласованность работы подразделений, дефекты и брак на этапе изготовления и после передачи изделия заказчику);
* Потери на обслуживание запасов. (Потери в результате приобретения сырья и материалов в объеме, превышающем необходимый на текущий плановый период, такие как затраты на складское хранение, убытки в результате ухудшения свойств материалов вследствие их длительного хранения, запасы готовой продукции, сырья, незавершенного производства между технологическими стадиями и на рабочем месте);
* Потери на перемещениях и движениях персонала. (Потери в результате нерациональной организации рабочего пространства, что отражается в виде лишних движений персонала для поиска документов, материалов, инструмента, отсутствия у предметов постоянного места, нерациональной организации рабочих мест);
* Потери от простоев. (Потери в результате простоя персонала или оборудования в ожидании информации, ресурсов, очередной технологической операции. Это могут быть сбои в логистической системе, отсутствие необходимой документации, ожидание указаний руководства по причине отсутствия текущих задач, недостаточный уровень делегирования полномочий);

# Анализ влияния планирования на проект с целью улучшения его эффективности

В настоящее время достаточно активно проявляется интерес к оценке проектной деятельности и эффективности управления проектами. Следовательно, совокупная оценка результатов бизнеса необходима не только в общем, но и по определенным программам и проектам. Таким образом, вопросы обеспечения оптимального проекта, определения долей проектов и их весов в затратах, рентабельности и сроках отвлечения средств на исполнение (в том числе и затраты на кадры) формируют особый подход к оценке эффективности проектного управления.

Наиболее важным фактором эффективного производства и управления проектами является грамотное планирование и сопровождение бизнес-процессов на всех его этапах.

Оптимизация планирования проекта строительства получила значительное внимание в течение последних 20 лет. В результате, множество методов и алгоритмов были разработаны для решения конкретных сценариев или проблем.

Планирование процессов выполнения для строительного проекта является сложной и трудной задачей. Выбор ресурсов (например, труда, основных средств) является наиболее важной частью планирования и должен рассматриваться по-разному, с ограничениями плана и работы, которая будет осуществляться в процессе выполнения данного плана. Так проекты являются уникальными по своей природе, создание графика строительных задач планировщиком, например, следует рассмотреть целый ряд условий, таких как: технологические, организационные методы и ограничения при выполнении, а также от наличия ресурсов для обеспечения того, чтобы потребности клиента и требования в отношении времени, стоимости и качества были выполнены. Ели эти правила не соблюдать, то можно понести ущерб из-за штрафов, прописанных в трудовых и временных условиях договора.

Процессы управления проектами включают в себя:

* Инициацию – принятие решения о начале выполнения проекта;
* Планирование – определение целей и критериев успеха проекта и разработка рабочих схем их достижения;
* Исполнение – координация людей и других ресурсов для выполнения плана;
* Анализ – определение соответствия плана и исполнение проекта и приятие решений о необходимости применения корректирующих воздействий;
* Управление – определение необходимых корректирующих воздействий, их согласование и применение;
* Завершение – формализация выполнения проекта и подведение его к упорядоченному финалу.

Планирование строительства связано с оптимальным уточнением деятельности в течение долгого времени и распределения ресурсов соответственно. Критерий продолжительности выполнения работ, является решающим фактором при учете клиентом оценки и выбором ставки, со сроком проекта. Подрядчики, почти всегда, стремятся свести к минимуму продолжительность выполнения проекта, чтобы получить преимущество при оценке тендерного предложения.



Рис.1.1. Составляющие системы планирования и контроля производства/строительства.

Любая деятельность в целом зачастую имеет зависимость от времени, затрат и ресурсов, которые являются ограниченными. Решение задачи оптимального планирования с учетом или без учета этих ограничений с течением времени меняются. Зависимость в ходе выполнения работ одно из самых основных ограничений, которое существует в строительных проектах. В процессе строительства, деятельность не может начаться, пока все его главные аспекты не будут завершены. Кроме того, время начала каждого вида деятельности не может быть позднее, чем последнего, чтобы закончить проект в рамках требуемого времени. Рабочее пространство всегда ограничено в проекте строительства. Рабочая зона может потребоваться в нескольких различных видах деятельности, в одно и тоже время. Таким образом, чтобы определить, как оптимально управлять рабочей зоной, чтобы облегчить планирование деятельности необходимо влиять на проект производства. Ресурсы являются наиболее влиятельными ограничениями в строительстве, так как они определяют целесообразность применения сроков реализации проекта, а также оптимальны ли они. Сжатое расписание сильно зависит от наличия и количества ресурсов. Ограниченная информация, которая состоит из чертежей, спецификаций, оценки риска и безопасности, разрешений для работы, также имеет значительное влияние на проблему планирования в строительстве. Информационный поток между деятельностью, часто упускается из виду.

# Постановка задачи дипломного проекта

# Анализ субъекта

В ходе дипломной работы необходимо разработать модуль создания и управления рабочими заданиями для десктоп программы инструмента формирования недельно-суточных заданий, который будет предназначен для использования строительным подрядчиком «АтомСтройЭкспорт». Данный программный продукт разрабатывается фирмой «Неолант-Тенакс».

С 2004 года группа компаний «НЕОЛАНТ» (Россия), поставщик № 1 решений для жизненного цикла промышленных объектов, осуществляет комплексную поддержку управления регионами и предприятиями в России и мире и предоставляет услуги по направлениям:

* Разработка и внедрение российских ГИС, САПР и PLM-систем.
* Цифровой инжиниринг: проектирование/сооружение, конструирование/изготовление.
* Разработка государственных и корпоративных информационных систем на заказ: мониторинг, анализ, прогнозирование.
* Информационное и имитационное моделирование территорий и предприятий.

Специалисты ГК «Неолант» с 2012 года активно разрабатывают информационные системы для атомной отрасли, в том числе и систему мониторинга и оперативного управления капитальным строительством, в которую входит инструмент формирования недельно-суточных заданий как приложение-клиент для взаимодействия с системой.

Система мониторинга и оперативного управления капитальным строительством – интернет-портал, предназначенный для поддержки принятия управленческих решений при капитальном строительстве (КС).

Целью создания инструментария является повысить эффективность управления капитальным строительством АЭС, что означает сократить сроки и стоимость, повысить качество проекта сооружения АЭС.

В ходе проекта специалисты «НЕОЛАНТ» и «НИАЭП» совместно разработали методологию мониторинга и оперативного управления капитальным строительством (КС) в атомной отрасли. В её основе – две идеи:

* Формирование системы аналитических показателей, отражающих возможные отклонения фактических показателей процессов сооружения АЭС от плановых. Эта система демонстрирует состояние процессов капитального строительства – проектирование, закупки, поставки, строительно-монтажные и пуско-наладочные работы – в разрезе сроков, стоимости и качества;
* Предоставление участникам проекта КС актуальной информации в реальном времени для поддержки принятия управленческих решений и обеспечения эффективных коммуникаций между всеми участниками проекта.

Далее «НЕОЛАНТ» реализовал ИТ-решение, воплотившее разработанную методологию:

* Для расчета аналитических показателей используется платформа бизнес-аналитики SAS BI. Реализованный на её основе аналитический модуль обеспечивает выявление «узких мест» в процессах сооружения АЭС, поддерживая декомпозицию показателей от сводных значений до исходных данных. Для удобства восприятия и анализа сводные значения отображаются по принципу «светофора» – красный цвет означает критическое отклонение фактических значений от плановых, зеленый – нахождение фактических значений в пределах допустимого отклонения от плана;
* Для доступа к информации по проекту и к информационным системам заказчика создан интернет-портал – единое информационное пространство для совместной и индивидуальной работы участников сооружения АЭС;
* Важнейшей составляющей IT-реализации методологии стала система визуализации. Она позволяет в цифрах, графиках и реалистичных изображениях ознакомиться с актуальной информацией: сравнить фактические показатели с плановыми, проектные 3D модели и сферические панорамы строительной площадки с календарно-сетевыми графиками и проектной документацией. В системе реализовано синхронное отображение инженерных 3D моделей объектов проекта КС – «как спроектировано», с их реальным видом «как построено» на основе сферических панорам. Это позволяет сравнить ситуацию на стройплощадке с запланированной и оперативно оценить ход работ.;
* Для удобства взаимодействия с субподрядными строительно-монтажными организациями в системе реализовано формирование недельно-суточных заданий, а также сбор данных о фактическом состоянии работ с помощью мобильных компьютеров/планшетов. Данные на планшетах синхронизируются с порталом, что обеспечивает участникам проекта получение информации о фактически выполненных работах в режиме реального времени.

# Постановка задачи

Для улучшения качества и эффективности планирования строительной деятельности и сопровождения бизнес-процессов была поставлена цель – разработать модуль рабочих заданий, представляющий собой подробный календарный план, отображающий графически последовательность запланированных, выполняемых и выполненных работ с информацией о расходах, для настольного клиентского приложения – инструмента формирования недельно-суточных заданий.

Создание данного модуля позволит:

* Формировать последовательность работ для каждого рабочего задания;
* Оптимизировать календарный план выполнения работ, что позволит улучшить эффективность проекта;
* Производить мониторинг выполнения работ и отслеживать все бизнес-процессы, связанные со строительством.

# Выбор языка и инструментария разработки

# Выбор языка

**C#** (произносится *си шарп*) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимны функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++,  Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Мною был выбран этот язык по 3 факторам:

1. Скорость разработки - синтаксис c# позволяет быстро создавать сложные механизмы при разработке
2. Технология WPF – разработанная для c# является одной из лучших технологий для создания настольных программ
3. Компилируемый язык – c# перенял скорость и мощность c++ и почти ей не уступает что позволяет совершать миллионы операций в секунду.

# Выбор инструментария

**Microsoft Visual Studio** — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual studio был выбран учитывая 2 фактора:

1. Это единственный инструмент где присутствует технология WPF и ее графический дизайнер
2. Это мощный инструмент с множеством функций и удобством разработки приложений на .NET платформе

# Анализ требований и определение спецификаций

# Выбор парадигмы программирования

При выборе между функциональным и объектно-ориентированным подходом были рассмотрены такие факторы как:

1. Объектно-ориентированное программирование (ООП) является более «традиционной» парадигмой. С её помощью разработано несчётное количество программ, в том числе огромные промышленные системы в финансовых организациях, телекоммуникации на производстве, складах, транспорте.
2. Функциональное программирование может, как облегчить, так и усложнить разработку ПО, особенно если программа не большая, ввиду того что требуется больше времени за удержанием всех функций чистыми и написанием высших функций.

Поэтому был выбран объектно-ориентированный подход с небольшим содержанием функционального программирования в виде чистых функций.

# Декомпозиция задачи

Основными частями будут являться интерфейс и логика программы. Для интерфейсной части возникают задачи как:

1. Проработка общего стиля программы – шрифтов, цветовой гаммы, элементов управления;
2. Реализация стандартов качества интерфейса, например, возможность перетаскивать работы влево/вправо по временной шкале на диаграмме Ганта, а не только при изменении дат старта и финиша работ, добавление всплывающих подсказок с актуальной информацией по работам;

Для логики программы стоят задачи:

1. Реализовать визуализацию загруженных рабочих заданий;
2. Реализовать создание новых рабочих заданий;
3. Реализовать создание цепочек последовательностей работ с различными типами связей;
4. Реализовать создание ограничений для старта работ («жесткий старт», «старт не раньше»)
5. Реализовать алгоритм выстраивания работ согласно их типам связей;
6. Реализовать поиск, сортировку и фильтрацию работ по их атрибутам;

# Диаграмма прецедентов

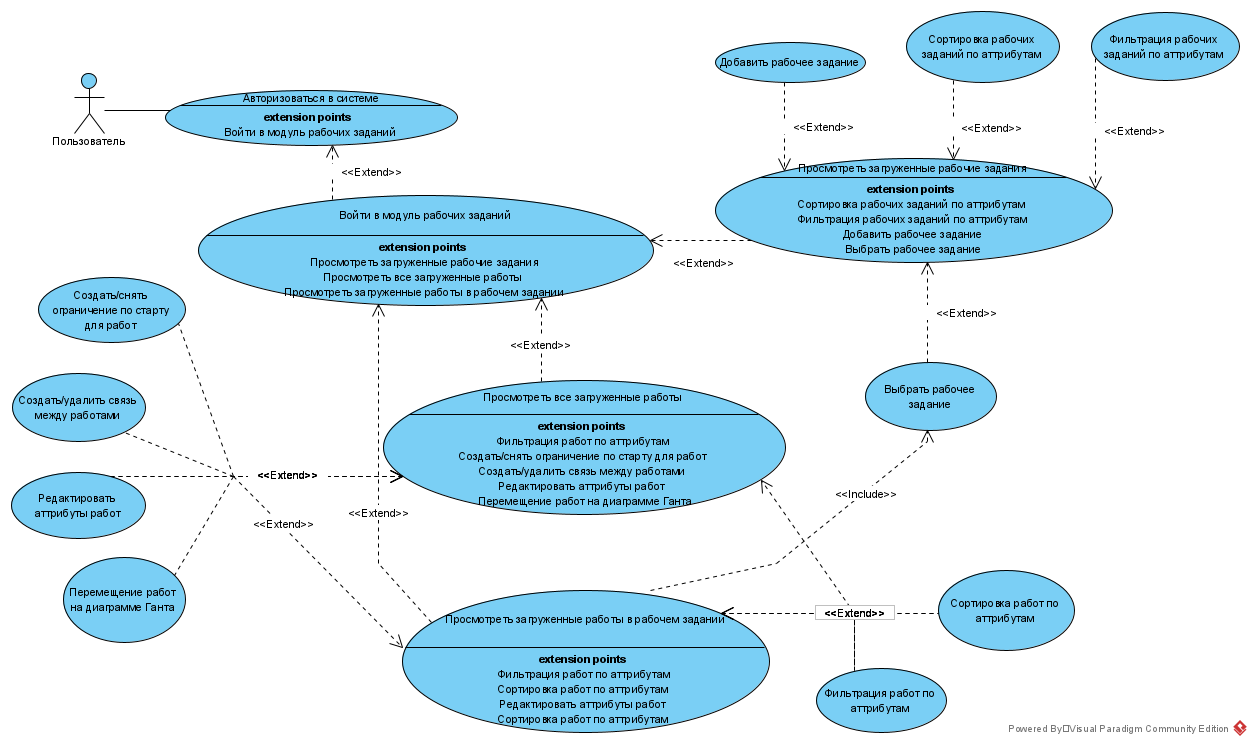


Рис. 2.1. Диаграмма прецедентов.

# Диаграмма деятельности для расшифровки содержания прецедентов

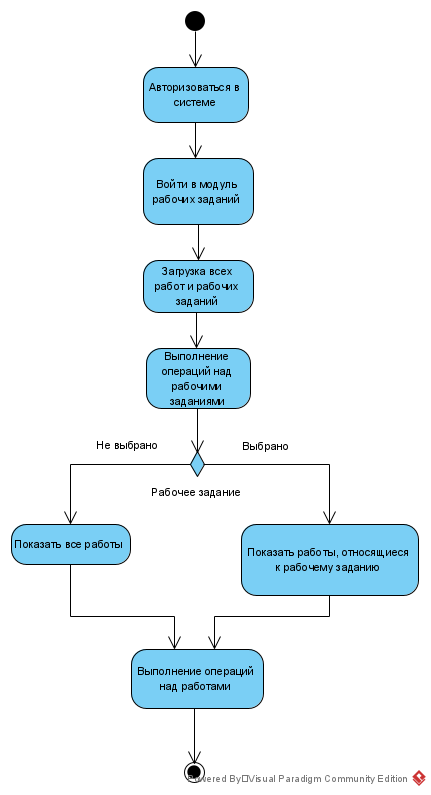
****

Рис. 2.2. Диаграмма деятельности.

# Диаграмма состояний

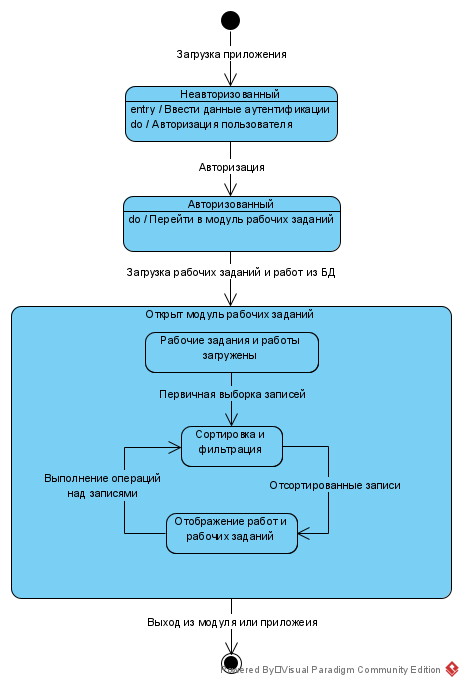


Рис. 2.4. Диаграмма состояний.

# Диаграмма классов

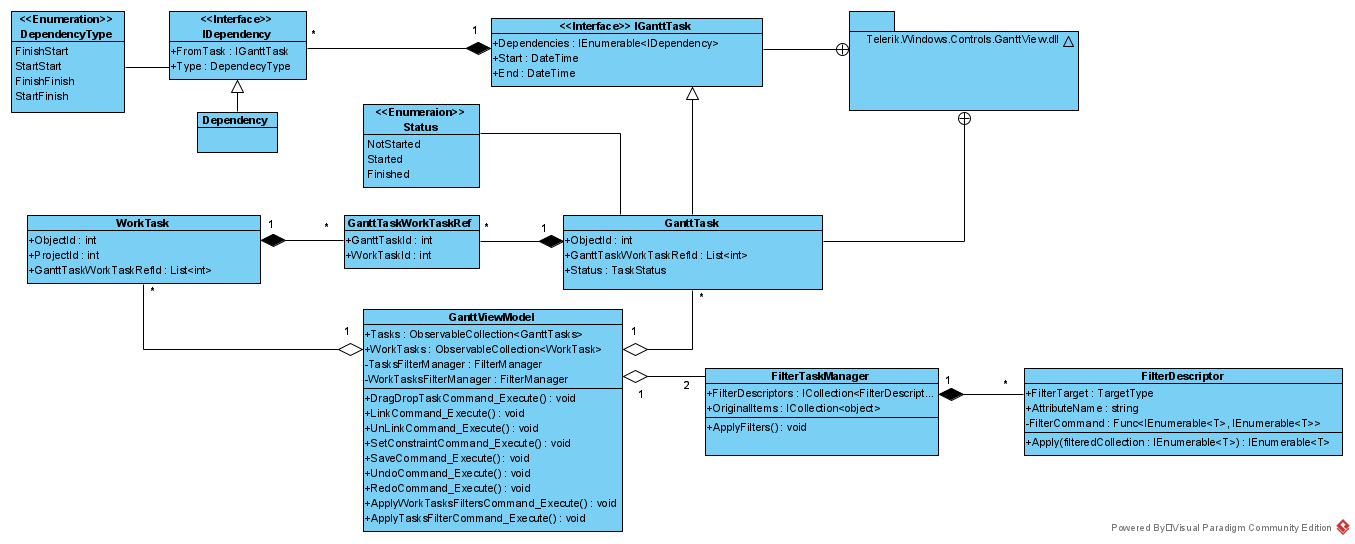


Рис. 2.5. Диаграмма классов.

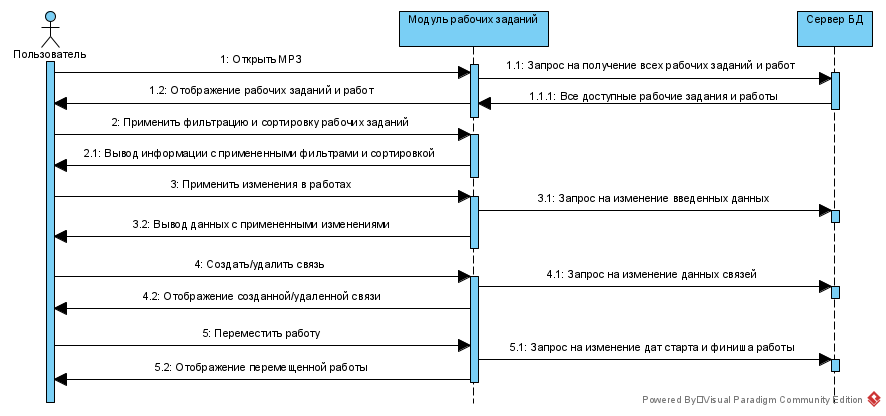


Рис. 2.5. Диаграмма последовательностей.

# Требования к программе

Разрабатываемый модуль должен удовлетворять следующим требованиям:

* Просматривать и редактировать все загруженные рабочие задания;
* Создавать новые рабочие задания;
* Просматривать все загруженные работы;
* Добавлять/удалять работы в рабочих заданиях;
* Создавать/удалять связи между работами;
* Редактировать доступные поля работ;
* Перемещать работы на диаграмме Ганта;
* Создавать ограничения по старту для работ;
* Корректно сохранять все изменения в БД.

# Проектирование программного продукта

# Архитектура приложения

Приложение «Инструмент формирования недельно-суточных заданий» состоит из 10 модулей, одним из которых и является разрабатываемый модуль рабочих заданий, который состоит из следующих подсистем:

* Основная подсистема для создания и редактирования рабочих заданий, сортировки и фильтрации и внесения изменений в атрибуты работ;
* Подсистема команд для отслеживания изменений, их отмены и сохранения в базу данных;
* Подсистема пользовательского интерфейса.

# Подсистема для работы с рабочими заданиями

Основная подсистема модуля, отвечающая за вычисление и взаимопередачу данных между пользовательским интерфейсом и базой данных. Подсистема для работы с рабочими заданиями является логической моделью всего, что видит на экране пользователь. Именно здесь формируются все команды от пользователя на изменение данных, которые в дальнейшем передаются в подсистему команд.

Основная задача – это формирование и редактирование списков рабочих заданий и работ для текущего комплексного сетевого графика.

# Подсистема команд

В связи с тем, что все элементы управления в пользовательском интерфейсе настроены на использование команд, то их функционал в проекте расширен глобально для возможности контролировать все изменения, совершаемые пользователем в процессе работы с приложением и модулем рабочих заданий в частности.

Данная подсистема включает в себя такие операции как «Undo» (отмена последнего в очереди изменения), «Redo» (восстановление отмененного в очереди изменения) и «Save» (сохранение текущих изменений). Так как некоторые действия пользователя могут неочевидно повлиять на зависимые параметры, то эти операции помогают ему сократить время на корректный ввод данных.

# Подсистема пользовательского интерфейса

Эта подсистема займет большое количество времени разработки, однако, именно она и будет определять удобство работы с программой. Будет использована технология WPF совместно с пакетом услуг Telerik. Эти технологии помогут в наименьший срок разработать дружелюбный, многофункциональный, масштабируемый интерфейс. Основное назначение этой подсистемы будет заключаться в предоставлении посредника между пользователем и логикой работы программы.

Основной задачей будет являться определение общего стиля интерфейса и возможность реализовать все функции других подсистем.

Преимущественной сложностью предвидится поддержание принципов паттерна MVVM.

# Диаграммы архитектуры приложения

# Диаграмма компонентов

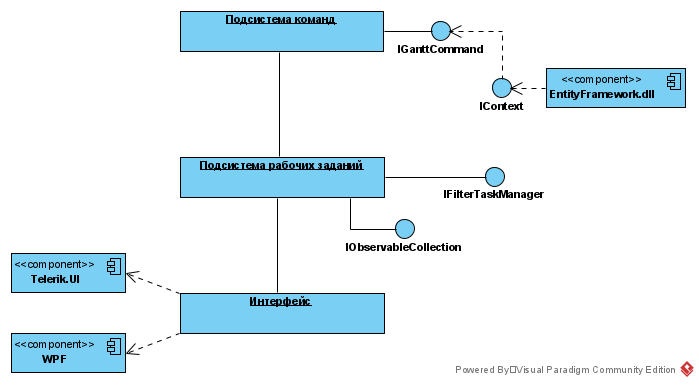


Рис. 3.1. Диаграмма компонентов.

# Диаграмма объектов

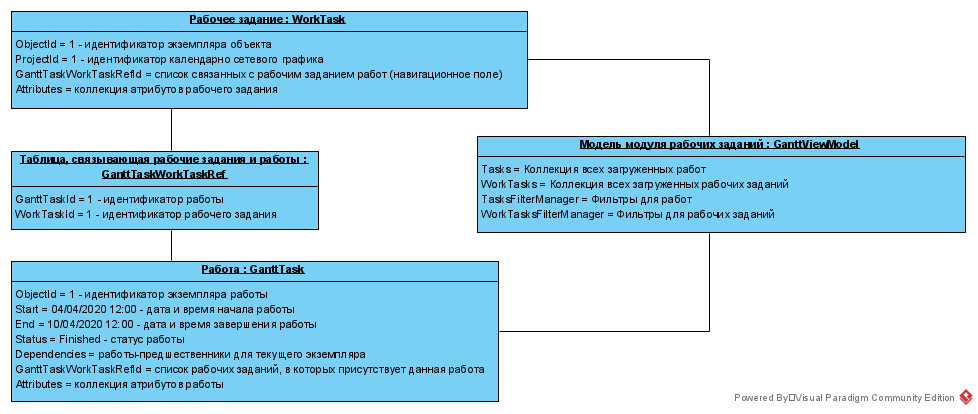


Рис. 3.2. Диаграмма объектов.

# Обзорная диаграмма взаимодействия

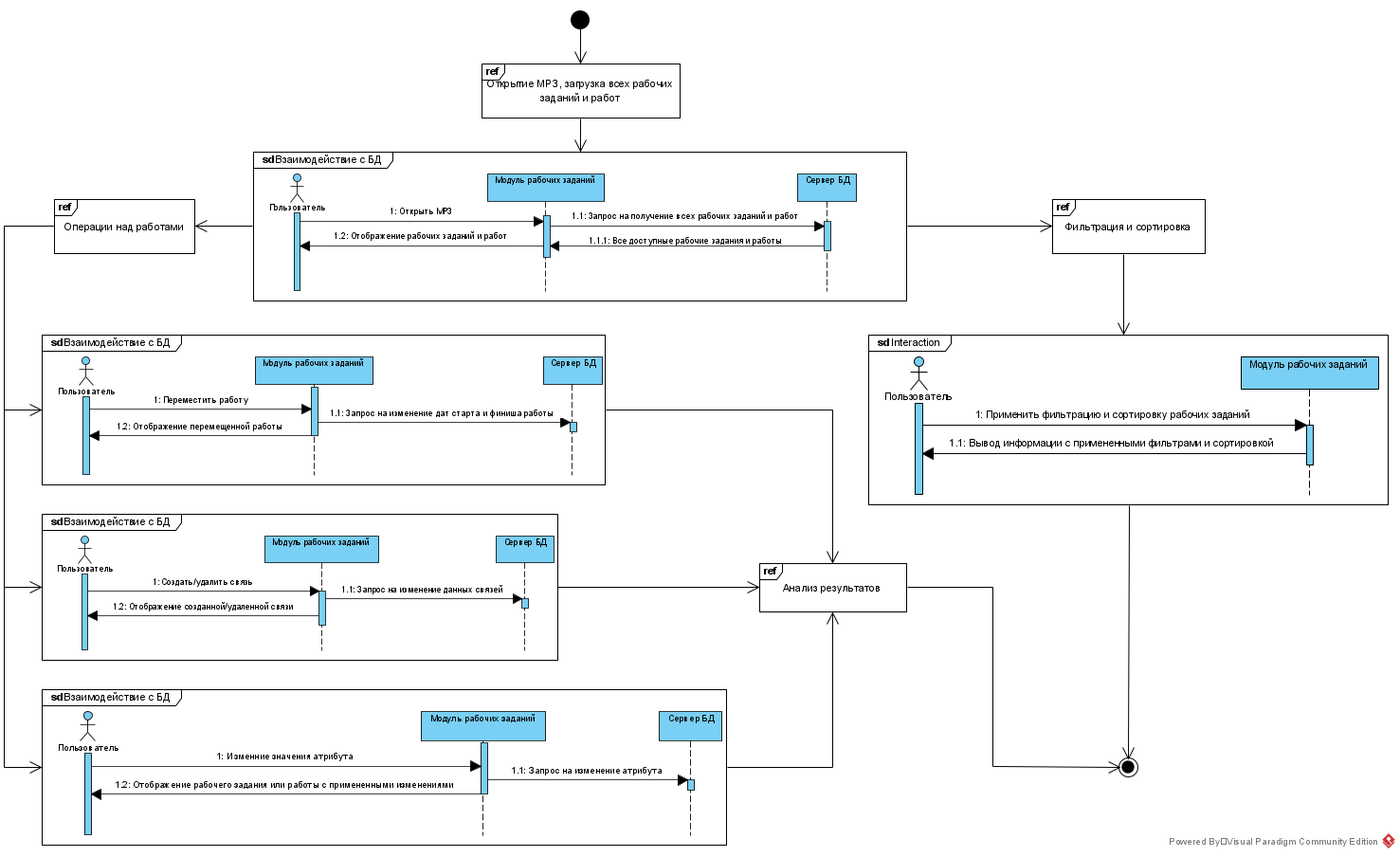


Рис. 3.4. Обзорная диаграмма взаимодействия

# Диаграмма пакетов

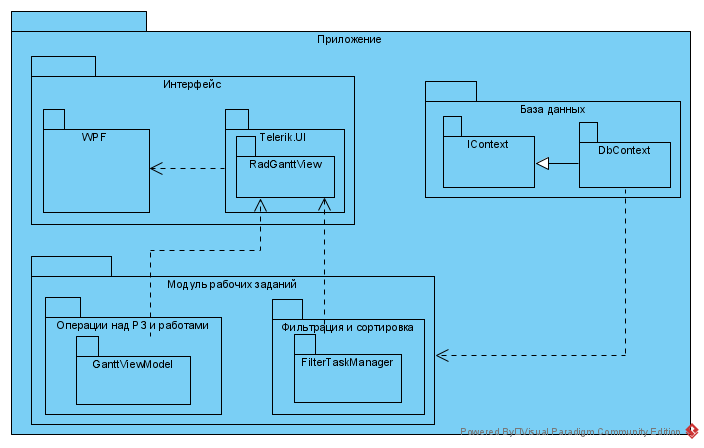


Рис. 3.5. Диаграмма пакетов

# Реализация

# Интерфейс пользователя

**После загрузки приложение встретит пользователя формой авторизации (рис 4.1). Введя аутентификационные данные, пользователь авторизуется под своей учетной записью, которой администратор выдал соответствующую роль для работы с приложением.**



Рис 4.1. Форма авторизации пользователя

Далее пользователю предлагается выбрать модуль, с которым он будет работать (рис. 4.2). В данном контексте ВКР будет демонстрироваться модуль «Рабочие задания».



Рис. 4.2. Окно выбора модуля

Этот модуль предоставляет полный функционал по формированию рабочих заданий (далее РЗ), просмотру и управлению ими.

После загрузки модуля (рис. 4.3) слева будет отображаться список РЗ, в котором пользователь может их выбирать, создавать новые или удалять, и открыта вкладка «Информация», с помощью которой доступны просмотр и редактирование описания рабочих заданий (рис. 4.4).

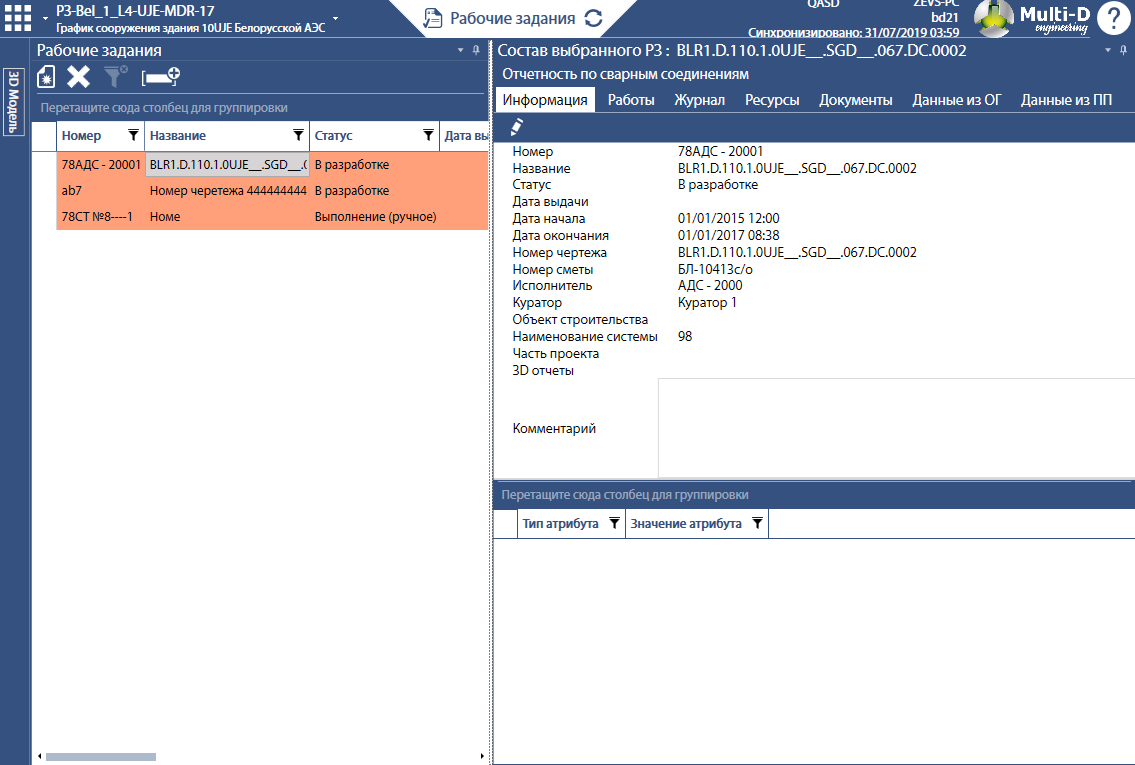


Рис. 4.3. Просмотр РЗ и вкладка «Информация»

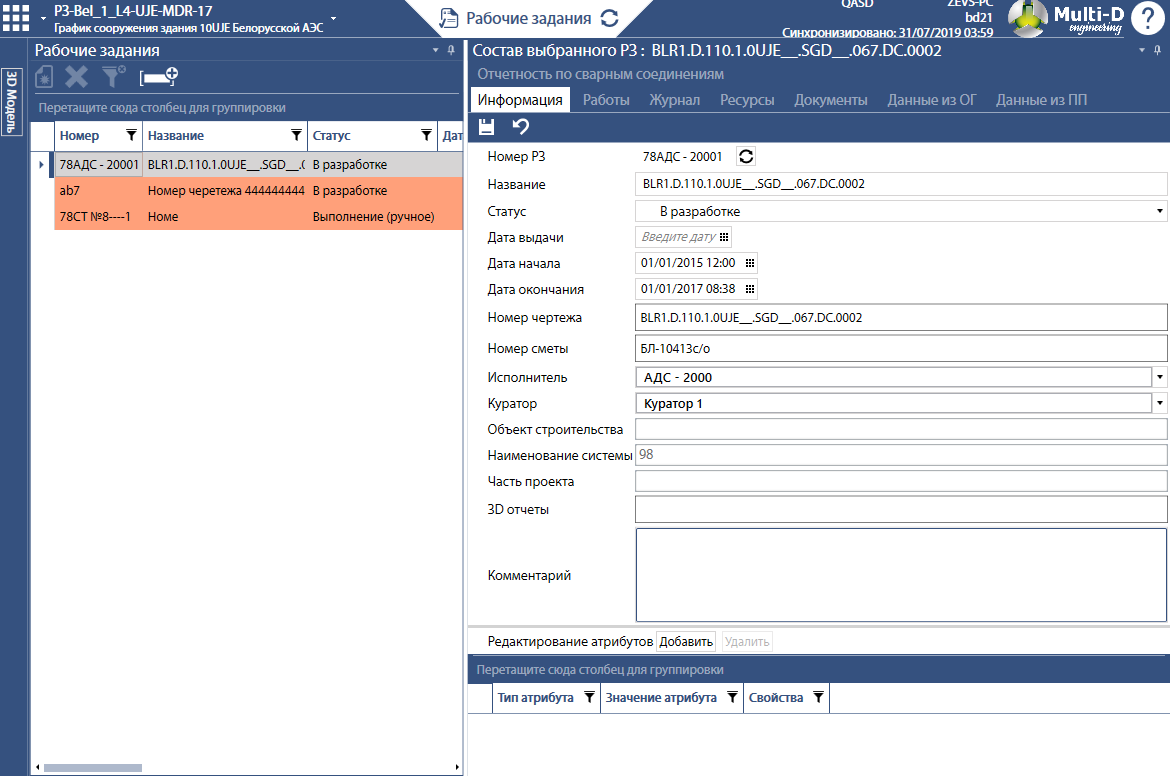


Рис. 4.4. Создание/редактирование РЗ

Теперь, когда РЗ было создано, пользователь может перейти во вкладку «Работы» для просмотра списка всех доступных работ (рис. 4.5 – 4.6) и формирования РЗ, назначив необходимые работы в РЗ (рис. 4.7).

На этой вкладке предоставляется следующая информация:

* название работы;
* идентификатор работы;
* в скольких РЗ состоит работа;
* запланированная дата старта;
* запланированная дата финиша;
* длительность работы в часах;
* другие выбранные пользователем атрибуты.

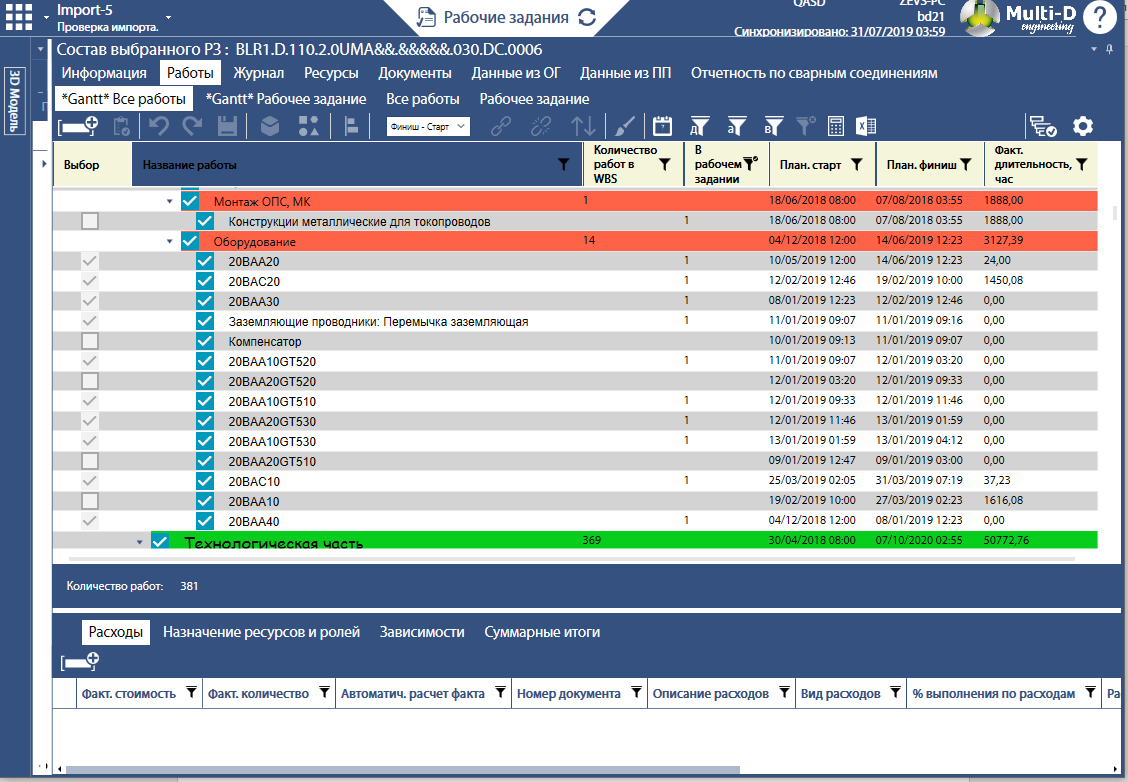


Рис. 4.5. Общий вид вкладки «Работы»

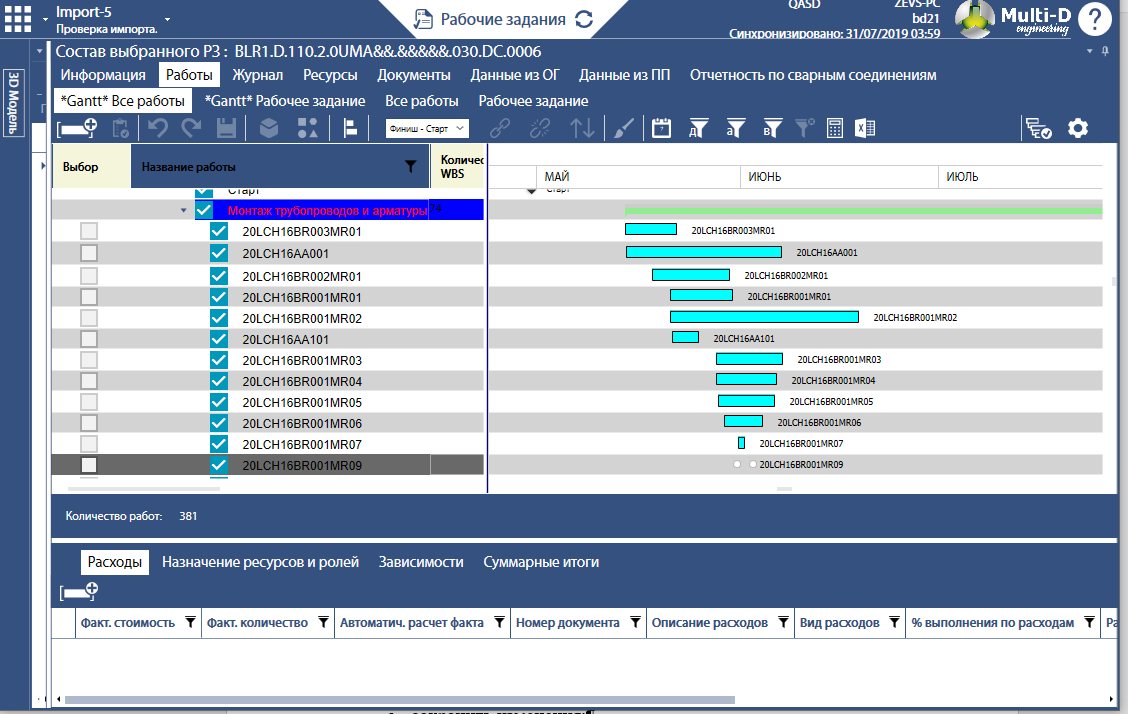


Рис. 4.6. Общий вид вкладки «Работы» с включенной диаграммой Ганта

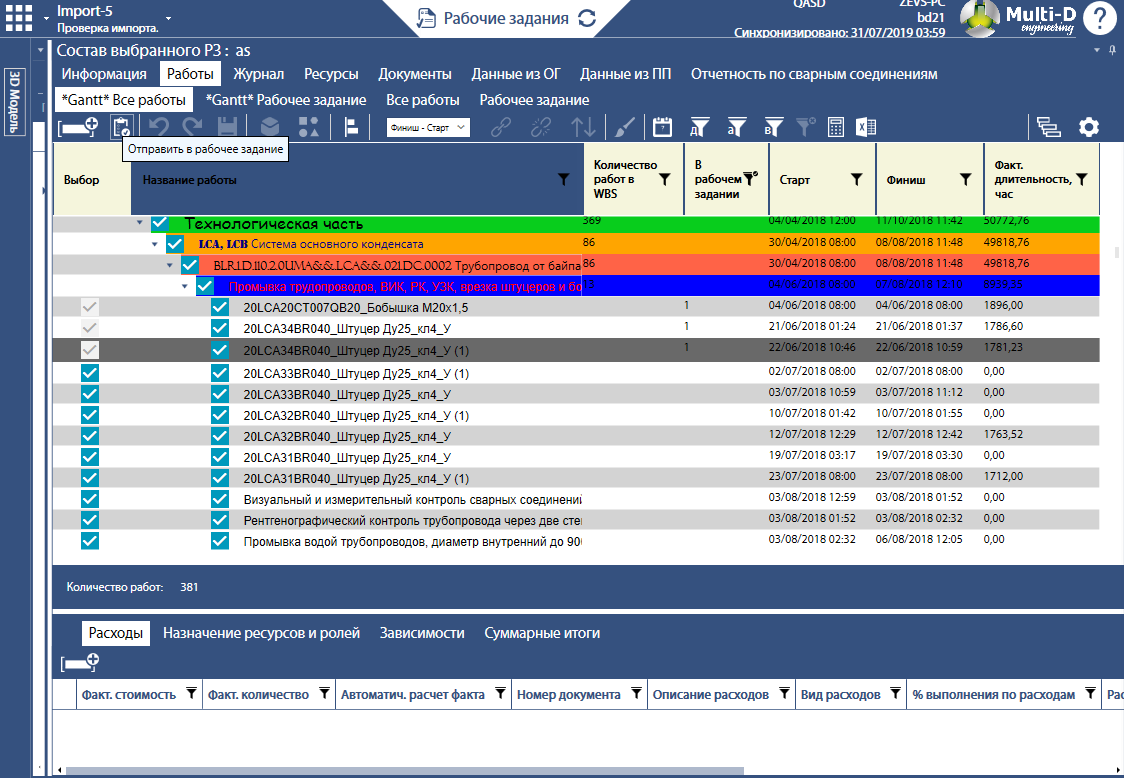


Рис. 4.7. Назначение работ в выбранное РЗ

Помимо формирования РЗ не менее важно выстроить работы в необходимом порядке в соответствии с требуемым графиком их выполнения по плану, что можно выполнить как путем прямого ввода плановых дат старта и финиша, так и методом перетаскивания работы влево-вправо на диаграмме Ганта (рис. 4.8).

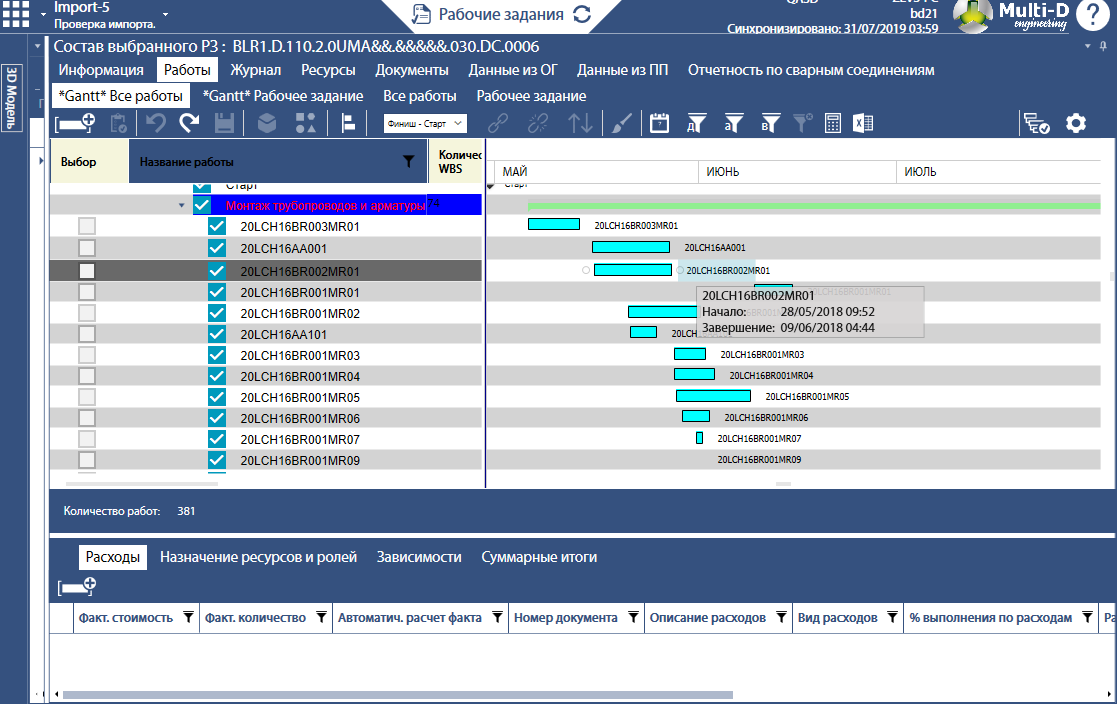


Рис. 4.8. Изменение планового старта работы методом Drag and Drop

Тем не менее, даже если пользователь вручную и выстроит все работы как ему необходимо, то они все равно останутся независимыми друг от друга, что неверно с точки зрения планирования. Необходимо создать связи (зависимости) между работами (рис. 4.9). В результате связанные работы выстроятся в порядке, соответствующий выбранному типу связи.

Для каждого типа связи есть свой алгоритм выстраивания работ, который определяет старт или финиш следующей за ним работы, а именно:

* Финиш-Старт – дата старта последующей работы должна быть больше или равна дате финиша предыдущей работы (рис. 4.10);
* Старт-Старт - дата старта последующей работы должна быть больше или равна дате старта предыдущей работы (рис. 4.11);
* Финиш-Финиш – дата финиша последующей работы должна быть больше или равна дате финиша предыдущей работы (рис. 4.12);
* Старт-Финиш - дата финиша последующей работы должна быть больше или равна дате старта предыдущей работы (рис. 4.13)

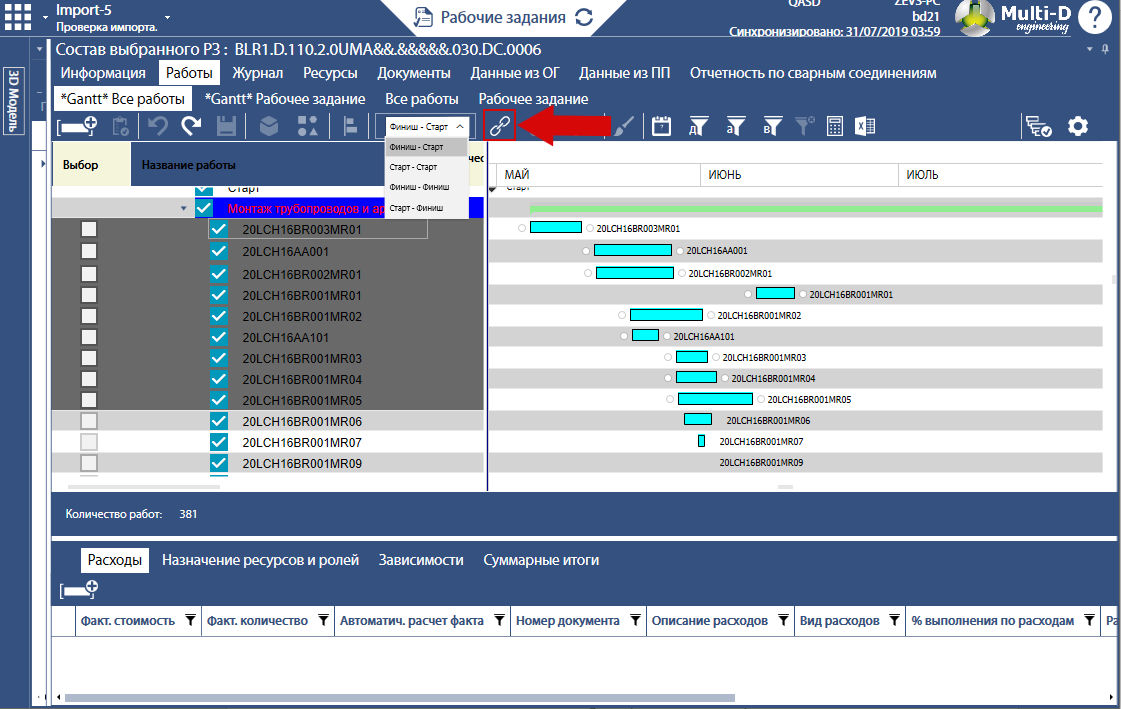


Рис. 4.9. Создание связей между выбранными работами   
типом связи «Финиш-Старт»

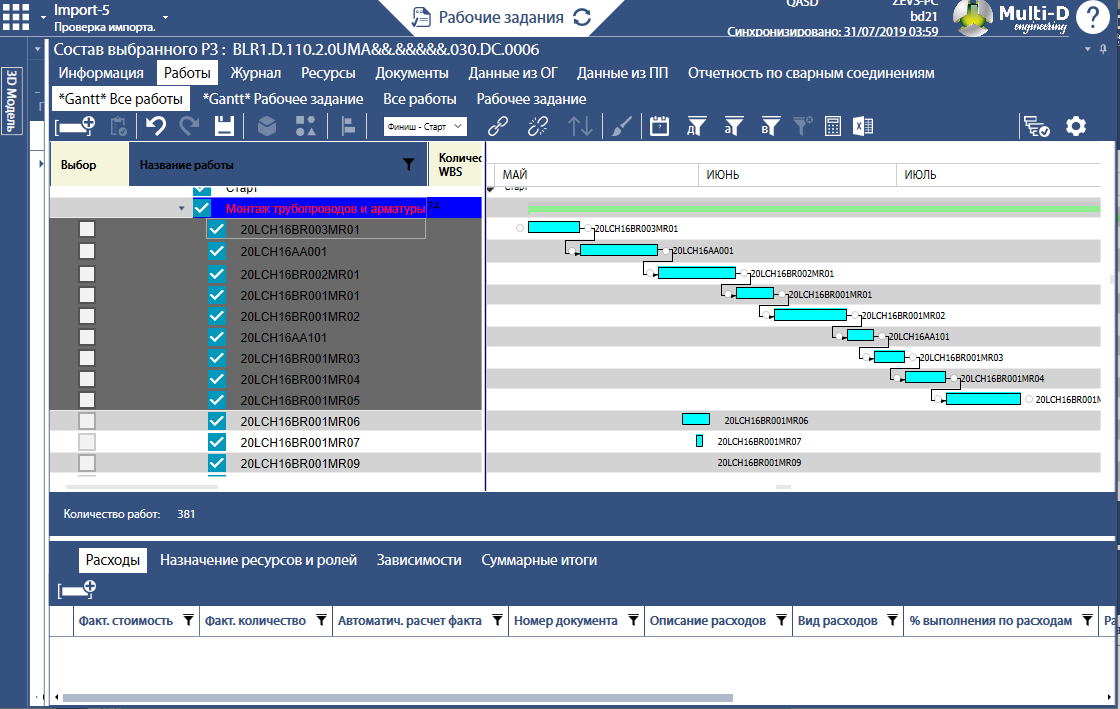


Рис. 4.10. Результат создания связей между выбранными работами  
типом связи «Финиш-Старт»

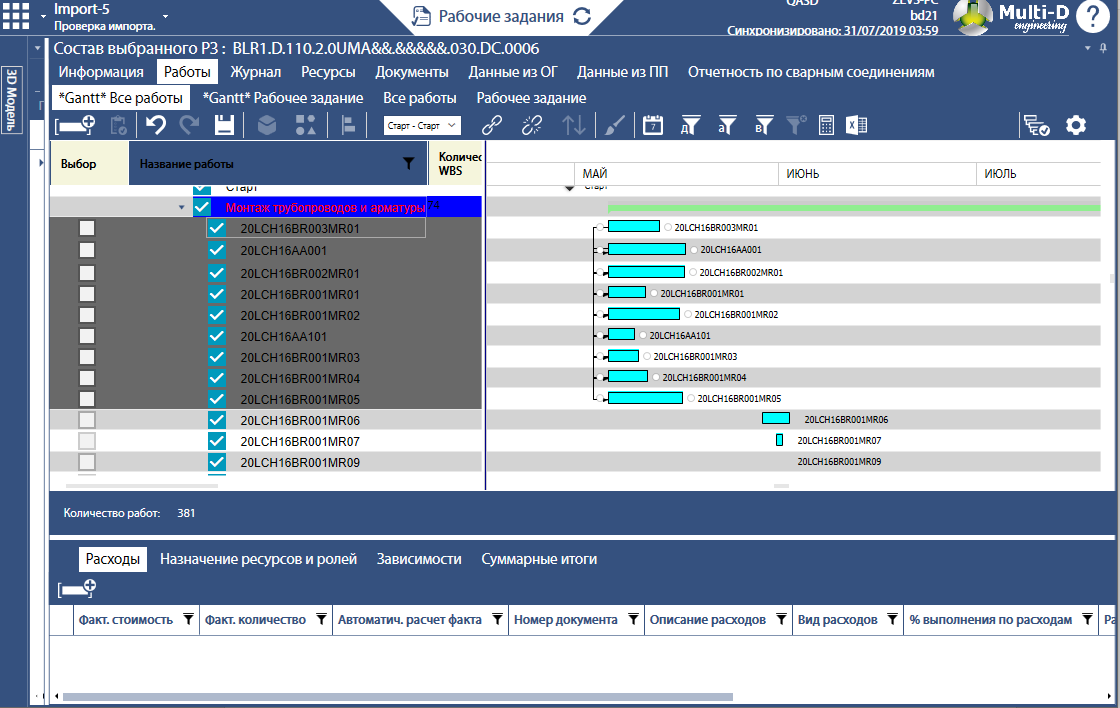


Рис. 4.11. Результат создания связей между выбранными работами  
типом связи «Старт-Старт»

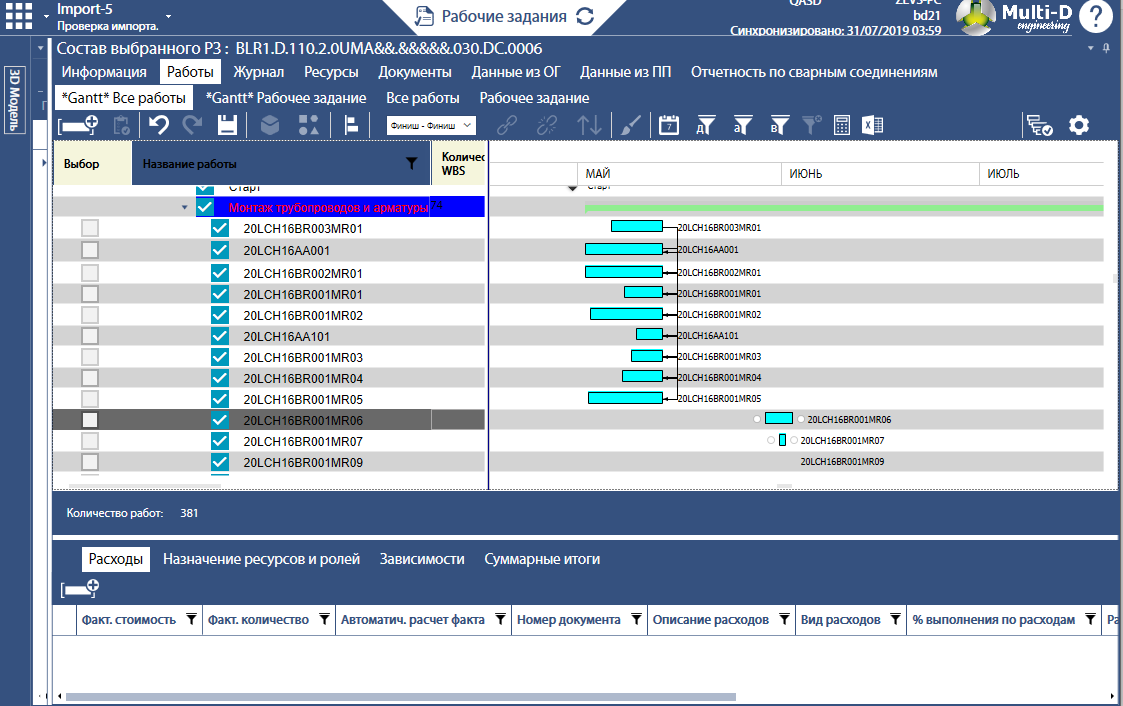


Рис. 4.12. Результат создания связей между выбранными работами  
типом связи «Финиш-Финиш»

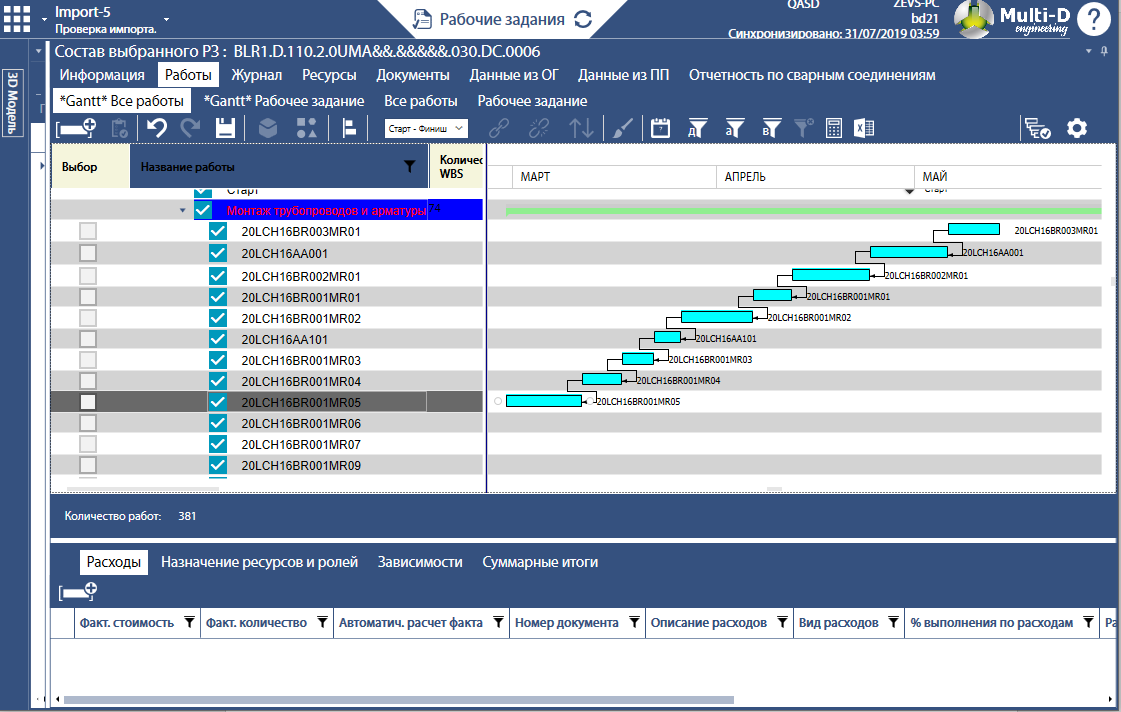


Рис. 4.13. Результат создания связей между выбранными работами  
типом связи «Старт-Финиш»

Работы в графике необязательно могут иметь только одного «предшественника» или «последователя», но визуально такие связи можно не увидеть явно по причине наложения одной линии связи на другую или из-за широкой цепочки работ, в которой уже не видно все связанные работы. Для такого случая была реализована вкладка «Зависимости», находящаяся в нижней панели вкладок, в которой отображаются все предшествующие и последующие работы для выделенной работы (рис. 4.14)

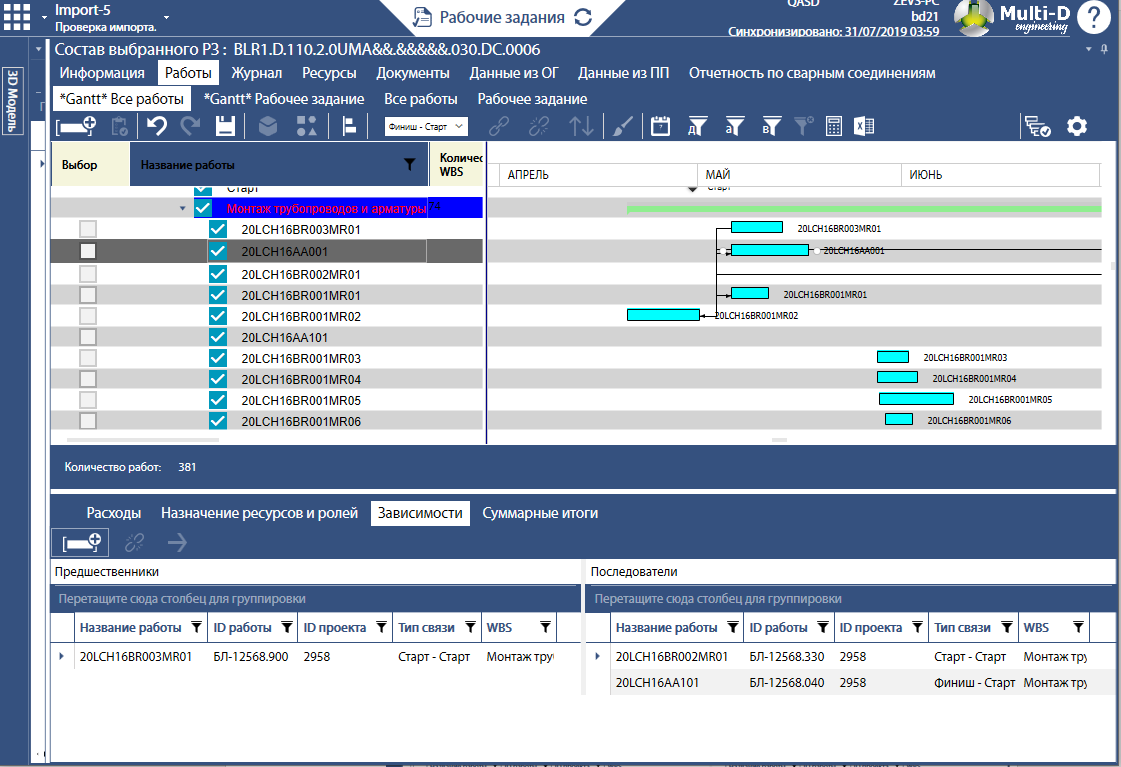


Рис. 4.14. Вкладка «Зависимости»

Также имеется функционал, который ограничивает дату старта работ по конкретной дате и типу ограничения, а именно:

* Жесткий старт – дата старта работы не может быть меньше или больше даты, указанной в ограничении (рис. 4.15);
* Старт не раньше – дата старта работы не может быть меньше даты, указанной в ограничении (рис. 4.16).

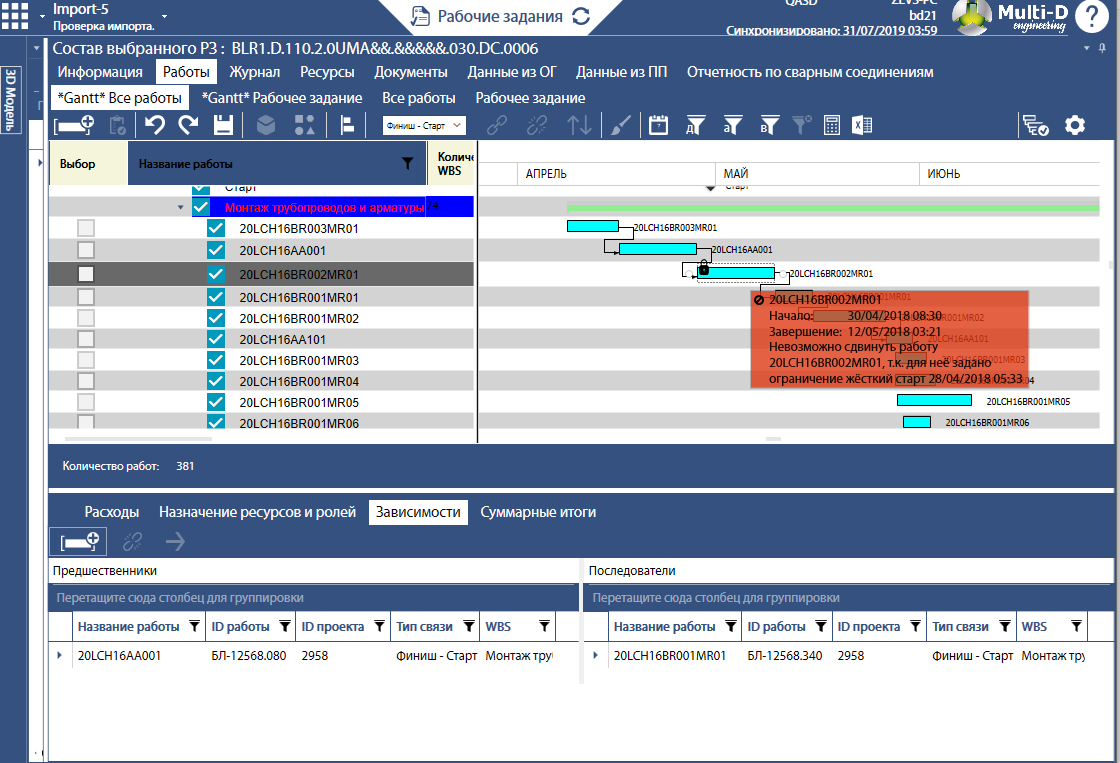


Рис. 4.17. Для работы было задано ограничение «Жесткий старт».

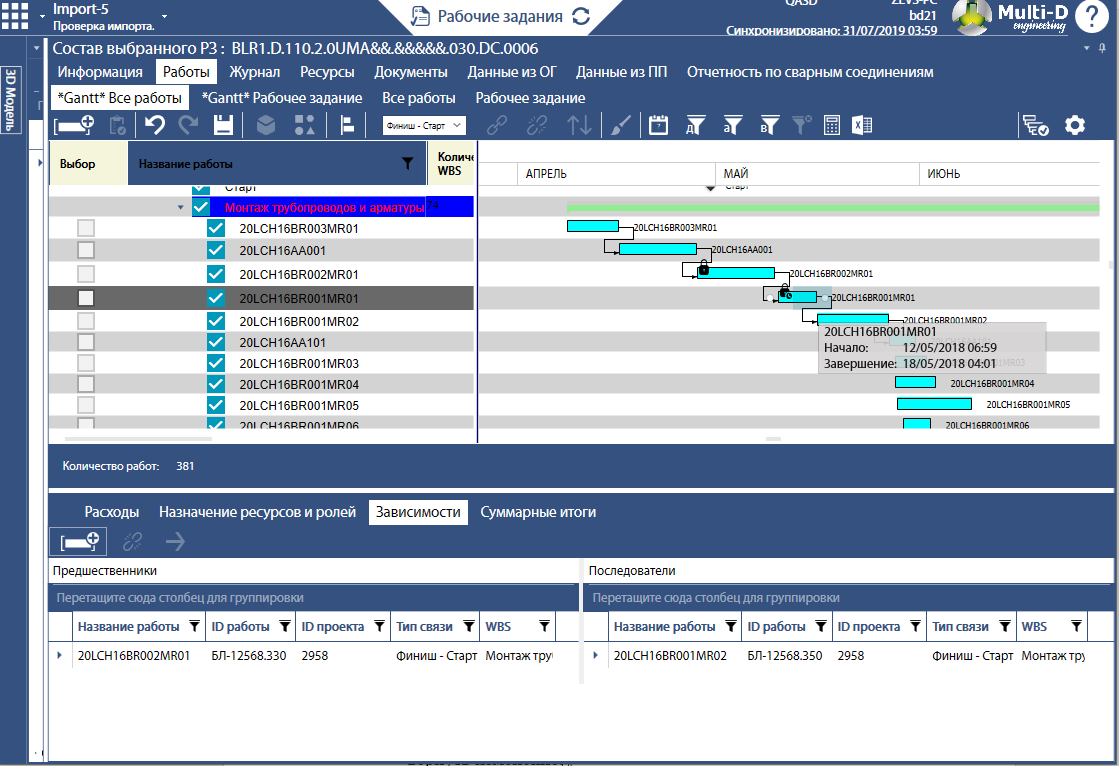


Рис. 4.18. Для работы было задано ограничение «Старт не раньше».

Заданные ограничения при создании связей между работами не позволят работам перемещаться при их нарушении (рис. 4.19).

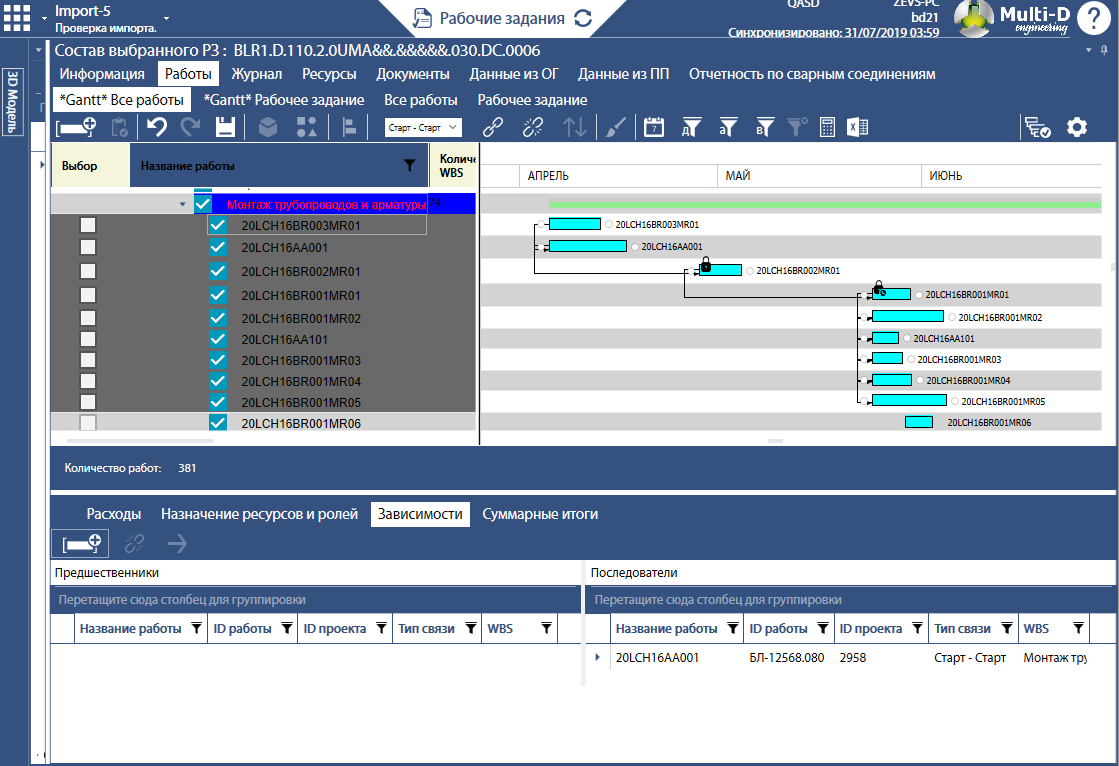


Рис. 4.19 Создание связей типа «Старт-Старт» с выставленными ограничениями для двух работ.

В модуле РЗ реализован функционал фильтрации и сортировки по атрибутам работ, который позволяет быстро найти интересующие записи из списка и выстроить их в порядке по возрастанию или по убыванию их значений (рис. 4.20). Фильтрация и сортировка являются многоуровневыми, и пользователь может использовать сразу несколько атрибутов для отображения списка.

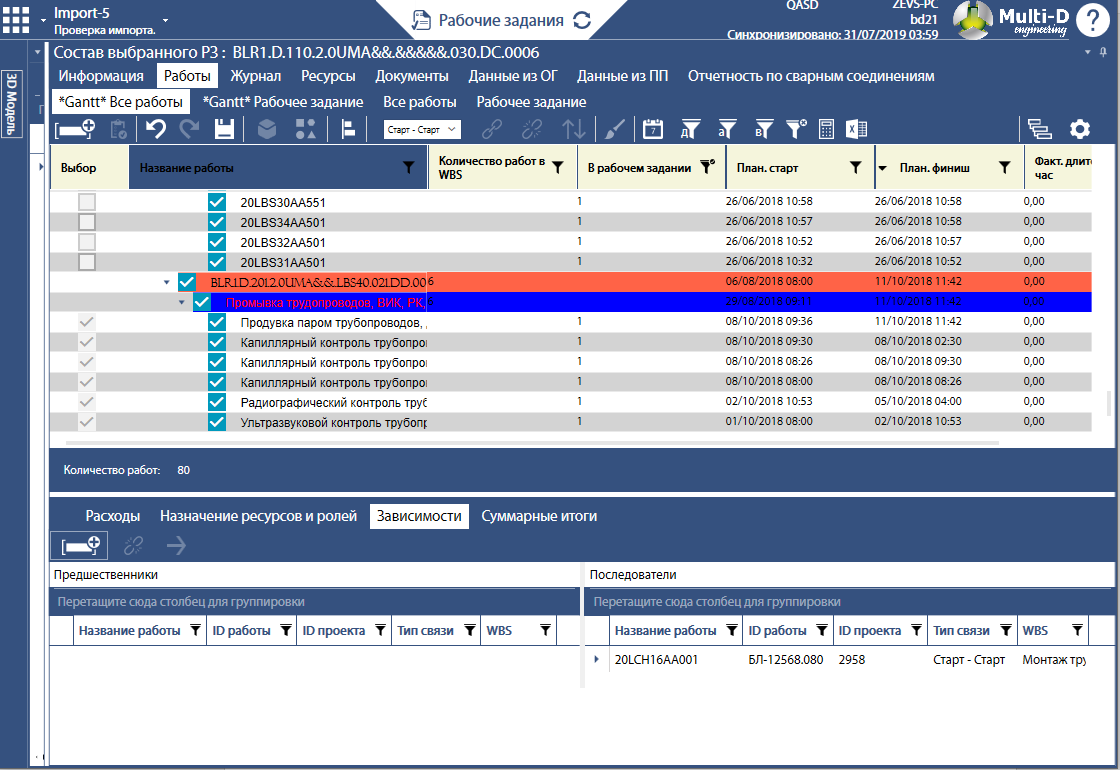


Рис. 4.20 Применение фильтрации и сортировки.

# Тестирование программного продукта

Модульное тестирование (англ. unittesting) –это процесс, позволяющий проверить на корректность отдельные модули исходного кода программы. Такой вид тестирования позволяет достаточно быстро проверить, не привело ли очередное изменение кода к появлению ошибок в уже оттестированных местах программы.

Использование модульного тестирования как раз подходит для проверки результатов выполнения команд перемещения работ, создания и удаления связей между работами.

Было разработано 3 теста, каждый из которых проверяет работу своей команды.

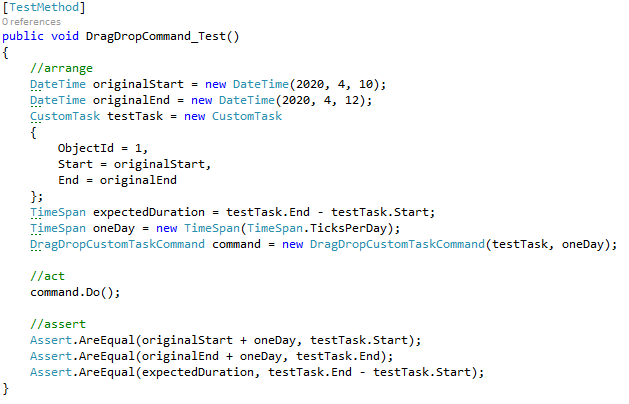


Рис. 5.1. Тест для проверки команды перетаскивания работы.

Тест представляет собой создание экземпляра некоторой работы, важные атрибуты для которого являются дата старта и дата конца работы. Разница между этими датами очевидно является длительностью работы, и она не должна будет измениться. В итоге дата старта и дата конца работы должны сместиться на один день вперед.



Рис. 5.2. Тест для проверки команды создания связи.

При создании связи типа «Финиш-Старт» между двумя работами, которые имеют одинаковые даты старта и конца, проверяется, что:

* работа-последователь получает только одну ссылку;
* работа-последователь получает ссылку на требуемую работу-предшественника;
* создается связь требуемого типа («Финиш-Старт»);
* дата начала работы-последователя равна дате конца работы-предшественника;
* дата конца работы-последователя равна сумме его даты начала и длительности;
* длительность работы последователя не изменилась.



Рис. 5.3. Тест для проверки команды удаления связи.

При удалении связи любого типа между двумя работами проверяется, что:

* у работы-последователя осталась одна ссылка;
* ссылка, не затронутая командой удаления, осталась у работы-последователя;
* у работы-последователя удалена требуемая ссылка;
* работа-предшественник не сместилась;
* работа-последователь не сместилась.

# Экономическое обоснование

Одним из наиболее удобных методов решения задач по оценке и анализу структуры затрат и потерь рабочего времени, разработке мероприятий по совершенствованию организации труда и повышению его производительности за счет устранения потерь и нерациональных затрат рабочего времени является фотография рабочего времени.

Фотография рабочего времени – это вид наблюдения, при котором измеряют все без исключения затраты времени исполнителя за определенный период работы. В результате получают точный срез: чем именно и в течении какого времени занимался конкретный сотрудник.

Фотография рабочего времени может применяться вне зависимости от наличия или отсутствия норм времени, выработки и т. д. на предприятии. Применение фотографии дает практический результат для нормирования труда всех категорий персонала – руководителей, профессионалов, специалистов, технических служащих и рабочих.

Рассчитаем, насколько изменится производительность труда пользователя ИФНСЗ, если он будет использовать в своей работе модуль РЗ с применением алгоритма оптимизации связанных работ и диаграммы Ганта.

Для начала приведем пример фотографии рабочего дня с применением алгоритма оптимизации связанных работ в модуле РЗ (Таблица 6.1).

Таблица 6.1

Фотография рабочего дня пользователя ИФНСЗ с применением алгоритма оптимизации связанных работ в модуле РЗ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Наименование затрат рабочего времени | Время | | | Выработка, единица работы |
| Начало | Окончание | Всего минут |
|  | Приход на работу | - | 9:50 |  |  |
|  | Приготовление рабочего места | 9:50 | 10:00 | 10 |  |
|  | Запуск приложения и модуля РЗ | 10:00 | 10:10 | 10 |  |
|  | Просмотр текущей задачи (оптимизировать планирование 200 связанных работ) | 10:10 | 10:20 | 10 |  |
|  | Сортировка и фильтрация требуемых для связи работ | 10:21 | 10:25 | 4 |  |
|  | Выделение по прямому порядку работ и создание связей | 10:25 | 10:26 | 1 |  |
|  | Выполнение задачи | 10:26 | 10:27 | 1 | 50 оп/сек |
|  | Выполнение других задач | 10:27 | 12:00 | 93 |  |
|  | Обед | 12:00 | 13:00 | 60 |  |
|  | Просмотр новой задачи (выполнить перемещение 20 цепочек работ, состоящей из 5 записей) | 13:00 | 13:10 | 10 |  |
|  | Выполнение задачи | 13:10 | 13:12 | 2 | 20 операций  15 оп/мин |
|  | Выполнение других задач | 13:12 | 17:45 | 273 |  |
|  | Уборка рабочего места, выключение компьютера | 17:45 | 18:00 | 15 |  |
|  | Завершение рабочего дня | 18:00 |  |  |  |

Общее время рабочего дня – 490 минут

* Подготовительно-заключительное время – 35 минут (7,1%)
* Оперативное рабочее время – 392 минут (80%)
* Время на основные операции (с алгоритмом оптимизации) – 3 минуты (0,6%)
* Время на отдых – 60 минут (12,2%)

Теперь фотография того же рабочего дня, но без применения алгоритма оптимизации связанных работ в модуле РЗ (Таблица 6.2).

Таблица 6.2

Фотография рабочего дня пользователя ИФНСЗ без применения алгоритма оптимизации связанных работ в модуле РЗ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Наименование затрат рабочего времени | Время | | | Выработка, единица работы |
| Начало | Окончание | Всего минут |
|  | Приход на работу | - | 9:50 |  |  |
|  | Приготовление рабочего места | 9:50 | 10:00 | 10 |  |
|  | Запуск приложения и модуля РЗ | 10:00 | 10:10 | 10 |  |
|  | Просмотр текущей задачи (оптимизировать планирование 200 связанных работ) | 10:10 | 10:20 | 10 |  |
|  | Сортировка и фильтрация требуемых для связи работ | 10:21 | 10:25 | 4 |  |
|  | Выделение по прямому порядку работ и создание связей | 10:25 | 10:26 | 1 |  |
|  | Выполнение задачи | 10:26 | 11:00 | 34 | 6 оп/мин |
|  | Выполнение других задач | 11:00 | 12:00 | 60 |  |
|  | Обед | 12:00 | 13:00 | 60 |  |
|  | Просмотр новой задачи (выполнить перемещение 20 цепочек работ, состоящей из 5 записей) | 13:00 | 13:10 | 10 |  |
|  | Выполнение задачи | 13:10 | 13:30 | 20 | 20 операций  1 оп/мин |
|  | Выполнение других задач | 13:30 | 17:45 | 255 |  |
|  | Уборка рабочего места, выключение компьютера | 17:45 | 18:00 | 15 |  |
|  | Завершение рабочего дня | 18:00 |  |  |  |

Общее время рабочего дня – 490 минут

* Подготовительно-заключительное время – 35 минут (7,1%)
* Оперативное рабочее время – 341 минут (69,5 %)
* Время на основные операции (с алгоритмом оптимизации) – 54 минуты (11%)
* Время на отдых – 60 минут (12,2%)

Если мы сравним 2 таблицы (таблицу 6.1 и таблицу 6.2), то увидим, что пользователь в обоих случаях справился с одной и той же задачей за разное время, причем во первом случае это заняло всего 3 минуты. Все потому, что:

* 1. Для оптимизации цепочки работ из 200 элементов пользователю приходилось вручную проверять тип связи и набирать дату старта и дату конца для каждой отдельной работы.
  2. Для перемещения 20 цепочек связанных работ приходилось выполнять практически те же действия, что и для оптимизации, однако связанные работы также приходилось искать вручную.

Можно сделать вывод, что работа пользователя ИФНСЗ в модуле РЗ значительно облегчилась, что позволяет сотруднику сосредоточиться на новых приоритетных задачах.

Теперь посчитаем, насколько увеличивается производительность труда пользователя, если пользуется в работе алгоритмом оптимизации:

, где

%

,

15

100

341

3

54

100

















*ОВф*

*ВОн*

*ВОф*

*П*

∆П = Производительность труда

*ВОф* – время, затраченное на основные операции (без применения алгоритма создания последовательности)

*ВОн* – время, затраченное на основные операции (с применением алгоритма создания последовательности)

*ОВф* – фактическое оперативное рабочее время

Таким образом, можно сделать вывод, что применение модуля РЗ с диаграммой Ганта и алгоритмом создания последовательности играет важную роль для строительного подрядчика. Без применения функционала модуля РЗ, производительность труда пользователя упала бы на 15 процентов (для вышеуказанного графика рабочего времени). Время, которое должно было уходить на выполнение основных функций, тратилось на решение основных задач своими силами. Из-за избыточной нагрузки сотрудник быстрее устает на работе, что непосредственно влияет на качество его работы. При использовании модуля РЗ сотрудник имеет возможность делать небольшие перерывы в работе или приступить к выполнению новых приоритетных задач, что повышает качество его работы. Все временные показатели в таблице представлены условно, в реальной работе поставленные задачи могут отличаться.

# Заключение

В ходе выполнения данной работы был создан модуль рабочих заданий для инструмента формирования недельно-суточных заданий (настольное приложение), отвечающее потребностям компании «АтомСтройЭкспорт».

Созданный модуль и его функционал поможет строительным подрядчикам быстро и эффективно планировать задачи, затрачивая минимальное количество времени на разработку и формирование РЗ без потерь их эффективности, тем самым позволяя сосредоточиться на других аспектах строительства.

В процессе выполнения данной работы было проведено исследование эффективности управления рабочими командами с целью увеличения производительности выполняемых работ и как это может повлиять на финансовое обеспечение строительного проекта.

Также был получен опыт проектирования, разработки, тестирования ПО и опыт использования таких технологий как:

1. WPF
2. MVVM
3. Telerik

# Список использованных источников

1. Албахари Д., Албахари Б. «C# 7.0. Справочник. Полное описание языка» 2010 г.
2. Петцольд Ч. Microsoft Windows Presentation Foundation 2008г.
3. Сурков С. А. Работа с персоналом: зависимость значимости от времени. Управление персоналом, 2003
4. Комов А. В. Понятие и принципы управления проектами – 2018г.
5. Хугаева А. Д. Формирование корпоративной системы управления проектами в организации – 2018г.
6. Сайт Википедия – свободная энциклопедия [Электронный ресурс]. http://ru.wikipedia.org

# Список сокращений

CLR – common language runtime (общеязыковая исполняющая среда)

MVVM – model - view – viewmodel

WPF – windows presentation foundation

ПО – программное обеспечение

СУБД – система управления базами данных

РЗ – рабочее задание

МРЗ – модуль рабочих заданий

ИФНСЗ – инструмент формирования недельно-суточных заданий