**Курсовая работа:**

1. Обоснование дипломного проекта – первая часть пояснительной записки (начать к концу сентября):

* Описание темы (что под ней подразумевается) – постановка задачи;
* Исследование предметной области (анализ проблематики; задача, которую мы решаем нашим дипломом) – определение методов, уточнение требований к программному модулю – зачем, почему, для кого, предпосылки;
* Обзор аналогов и выводы из обзора (их описание, обоснование и сравнение);
* Формирование функциональных требований к дипломному проекту.

1. Функциональные требования (выбор технических средств разработки).
2. Модель будущего дипломного проекта (диаграммы, алгоритмы).
3. Техническое задание.
4. Разработка дипломного проекта (март).
5. Техническое описание.
6. Экономическая оценка дипломного проекта – экономика отрасли.
7. Пояснительная записка (2-3 приложения: Модель, ТЗ, Тех. Описание) – около 70 страниц.

Затем получаем отзыв от заказчика (лицензия) и руководителя.

**Защита**: документы + презентация.

**Мастер-классы** от экспертов и внутренних лиц по презентации своего диплома.

1 АНАЛИЗ поставленной задачи

1.1 Формулировка поставленной задачи

Задачей дипломного проектирования является разработка (функциональное моделирование) приложения (далее - система) для управления проектами с использованием технологии интеллект-карт.

Назначение разработки – обеспечение пользователей инструментом визуализации при распределении задач и контроле за их выполнением в рамках управления проектом.

Целевой аудиторией является в первую очередь, команды, осуществляющие проектную деятельность.

1.2 Обзор предметной области

Управление проектами подразумевает под собой совокупность планирования, реализации, а также контроля за ходом выполнения проекта.

Процесс управления проектом включает в себя следующие этапы:

* этапы процессов инициализации;
* этапы процессов планирования;
* этапы процессов организации исполнения;
* этапы процессов завершения.

В проекте выделяют две роли: руководитель команды и участник команды.

Руководитель команды отвечает за распределение задач по участникам команды, а также за контроль сроков выполнения этапов проекта.

Далее, детальнее рассмотрим этапы процессов планирования и организации исполнения проекта.

Процессы планирования проекта начинаются от момента запуска проекта и реализуются вплоть до завершающих стадий. Это многократные процедуры, реализуемые на каждой фазе решения проектной задачи. Целями этих процессов являются: детальная разработка содержания, выработка плана действий по управлению проектом и составление календарного расписания работ.

Рассмотрим некоторые из этапов создания планов проекта:

* разработка структуры проекта. в наиболее полном варианте в этап входит создание деревьев целей, задач, организационной структуры;
* определение последовательности работ. данный этап позволяет выстроить логику взаимосвязей операций;
* календарное планирование проекта;
* выполнение вспомогательных планировочных мероприятий. в настоящий этап включается разработка планов поставок, коммуникаций и других обеспечительных планов.

Средством, с помощью которого можно визуализировать структуру проекта, является интеллект-карты.

Методология интеллект-карт представляет собой инструмент активизации мыслительного процесса через построение радиантных диаграмм, способствующий построению индивидуальных деревьев мозгового штурма, которые способствуют генерации идей и быстрому запоминанию информации.

Существуют различные области использования интеллект-карт, такие как:

* планирование;
* презентации;
* запоминание;
* обучение;
* мозговой штурм;
* принятие решений.

При управлении проектом необходимо выделять его главные и вспомогательные (составные, детализированные) задачи, визуализация которых способствует облегчению коллективного мозгового штурма. Для решения данной проблемы могут быть использованы интеллект-карты, позволяющие визуализировать ассоциации разных уровней, устанавливая связи типа родитель-потомок от центральной задачи к периферийной, с несколькими уровнями детализации.

Процессы планирования сроков проекта может решить система посредством установки сроков выполнения задач.

Процессы организации исполнения проекта находятся исключительно в зоне управленческих компетенций руководителя проекта. Постановка задач, координация и оперативное реагирование – все это должно выполняться на этапах данной группы процессов.

Рассмотрим некоторые из этапов организации исполнения проекта.

* обеспечение координации работ и исполнителей. этап своей целью имеет обеспечение четкого взаимодействия участников за счет установленных приоритетов задач, согласований с функциональными руководителями, качественной информационной поддержки команды;
* организация распределения информации. распределение и движение информации по участникам проекта;

Процессы распределения информации, обеспечения координации задач и исполнителей проекта может решить система посредством создания карточек, в которых содержатся задачи, а также их подробное описание.

При анализе этапов выполнения проекта были выделены следующие проблемы

* сложность организации командной работы;
* невозможность удаленной работы в проекте;
* отсутствие механизма контроля за выполнением задач;
* сложность детализации описания при постановке задач;
* сложность коммуникации между участниками проекта;
* сложность разработки структуры проекта ввиду необходимости визуализации дерева задач.

Разрабатываемая система нацелена на решение вышеописанных проблем.

(нет готовых решений, у нас получается система, которая решает эти проблемы, в итоге получается готовое проблемно-ориентированное решение)

1.3 Сравнение аналогов

В ходе обзора аналогов были выбраны следующие решения:

1. «Trello».
2. «Asana».
3. «Jira».
4. «GanttProject»
5. «Wrike»
6. «Mindmeister».
7. «Mindmup».
8. «Coggle»
9. «FreeMind»
10. «MindNode»

«Trello» - прямой аналог. Это разработка компании «Fog Creek Software», предназначенная для управления проектами небольших групп. Проекты представлены в виде «досок», в которых есть разбиение на столбцы, в которых содержатся карточки с задачами.

Отличительной особенностью «Trello» является возможность расширения функционала путем интегрирования сторонних приложений в разрабатываемый проект.

Этот аналог не подходит для решения задачи, поставленной перед проектируемым ПО в силу того, что отсутствует возможность представления данных в виде интеллект-карты. Вследствие чего отсутствует возможность отображения задач при «мозговом штурме» без использования сторонних приложений.

«Asana» - прямой аналог. Это разработка представляет собой мобильное и веб-приложение для управления проектами в небольших командах. Приложение представлено в виде списка задач, которые прикреплены к проекту.

Отличительной особенностью «Asana» является возможность взаимодействия с календарем посредством отображения на календарь сроков выполнения задач, а также их создания.

Этот аналог не подходит для решения задачи, поставленной перед проектируемым ПО в силу того, что отсутствует возможность представления данных в виде интеллект-карты. Вследствие чего отсутствует возможность отображения задач при «мозговом штурме» без использования сторонних приложений.

«Jira» - прямой аналог. Это разработка компании «Atlassian», коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами.

Отличительной особенностью «Jira» является возможность создания различных отчетов, отображающих состояние выполнения проекта.

Этот аналог не подходит для решения задачи, поставленной перед проектируемым ПО в силу того, что отсутствует возможность представления данных в виде интеллект-карты. Вследствие чего отсутствует возможность отображения задач при «мозговом штурме» без использования сторонних приложений.

«GanttProject» - прямой аналог. Это программа, предназначенная для планирования проектов на основе построения диаграмм Гантта и диаграмм типа PERT.

Отличительной особенностью «GanttProject» является возможность импорта / экспорта документов Microsoft Project.

Этот аналог не подходит для решения задачи, поставленной перед проектируемым ПО в силу того, что отсутствует возможность представления данных в виде интеллект-карты. Вследствие чего отсутствует возможность отображения задач при «мозговом штурме» без использования сторонних приложений.

«Wrike» - прямой аналог. Это разработка компании «Wrike», облачный сервис для совместной работы и управления проектами, предназначена для команд и целых организаций.

Отличительной особенностью «Wrike» является возможность создания различных отчетов, отображающих состояние выполнения проекта, а также составления планов в виде диаграммы Ганта.

Этот аналог не подходит для решения задачи, поставленной перед проектируемым ПО в силу того, что отсутствует возможность представления данных в виде интеллект-карты. Вследствие чего отсутствует возможность отображения задач при «мозговом штурме» без использования сторонних приложений.

В силу того, что рассматриваемые прямые аналоги схожи по функционалу, выделим основные функции представленных систем для управления проектами.

Перечень общих функций рассмотренных продуктов:

1. Создание карточек, а также возможность их детального описания: добавление описания карточки, а также добавления списков задач.
2. Прикрепление файлов и ссылок к карточкам. К каждой карточке можно прикрепить файлы в различных форматах или ссылки на сторонние ресурсы.
3. Добавление комментариев к карточкам в текстовом формате, а также прикрепить ссылку на сторонний ресурс.
4. Установка и контроль за сроками выполнения задач. Для каждой карточки можно установить срок выполнения, а также проконтролировать его соблюдение.
5. Добавление участников к разработке проекта. К карточкам можно прикреплять одного или нескольких участников, которые будут отображаться в проекте.

«Mindmeister» - косвенный аналог. Это разработка компании «MeisterLabs GmbH», предназначенная для создания интеллект-карт.

«Mindmup» - косвенный аналог. Это разработка компании «Sauf Pompiers Ltd», предназначенная для создания интеллект-карт.

«Coggle» - косвенный аналог. Это разработка, предназначенная для создания интеллект-карт. Также, в программе присутствует возможность создания блок-схем.

«FreeMind» - косвенный аналог. Это программа предназначена для создания интеллект-карт. Представлена для настольных операционных систем.

«MindNode» - косвенный аналог. Это разработка компании «IdeasOnCanvas GmbH», предназначенная для создания интеллект-карт.

В силу того, что рассматриваемые косвенные аналоги схожи по функционалу, они предназначены для представления данных в виде интеллект-карт. Из чего можно сделать вывод, что они не подходят для решения задачи, поставленной для проектируемого ПО ввиду отсутствия функционала для управления проектами.

Перечень общих функций рассмотренных продуктов:

1. Создание интеллект-карт: отображение данных в виде деревьев для более эффективного восприятия информации.
2. Поиск по веткам: среди всех веток можно найти концевой узел или подветку по заданному значению.
3. Добавление участников к построению карты. К картам можно прикреплять одного или нескольких участников, которые будут отображаться в проекте.
4. Добавление текстовых заметок к рабочей области. К рабочей области можно прикрепить заметки с текстом, которые объясняют или поясняют информацию на карте.
5. Прикрепление файлов к карте. К карте можно прикрепить файлы в различных форматах.

Сравнение функций представленных аналогов приведено в таблице. При оценке используются следующие критерии:

* совместная разработка – возможность участия в проекте нескольких людей одновременно;
* прикрепление файлов – возможность прикрепления дополнительных материалов (файлы, ссылки) к задачам;
* платформы – какие платформы (операционные системы) поддерживает приложение;
* стоимость – размер платы (в рублях) за использование продукта в месяц;
* поиск по данным – возможность фильтрации информации по средствам поиска по имеющимся данным;
* установка сроков выполнения – возможность установки сроков выполнения задач (дата и время, которые обозначают крайний срок выполнения);
* представление данных в виде интеллект-карт – возможность представления дынных в виде интеллект-карт;
* создание команд – возможность создания команд, объединяющих пользователей для совместной работы;
* отображение сроков на календарь – возможность отображения сроков выполнения в виде календаря с метками задач;
* создание диаграмм – возможность создания графического представления данных в виде диаграмм;
* авторизация – необходимость предварительной регистрации в приложении для его использования.

Таблица 1 – Сравнение функционала аналогов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Прямые аналоги | | | | | Косвенные аналоги | | | | |
| Trello | Asana | Jira | GanttProject | Wrike | Mindmeister | Mindmup | Coggle | FreeMind | MindNode |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Совместная разработка | + | + | + | + | + | + | + | + | - | + |
| Прикрепление файлов | + | + | + | - | + | - | - | - | - | - |
| Платформы | Windows, Mac, iOS, Android, Web | iOS, Android, Web | Windows, Linux, Web | Windows, Linux, Mac | Web | iOS, Android, Web | Web | Web | Windows, Linux, Mac | Mac, iOS |
| Стоимость | 700 р/месяц | 700 р/месяц | 700 р/месяц | Бесплатно | 640 р/месяц | 550 р/месяц | 200 р/месяц | 320 р/месяц | Бесплатно | 2560 р |
| Поиск по данным | + | + | + | + | + | + | + | + | + | + |
| Установка сроков выполнения | + | + | + | - | + | - | - | - | - | - |
| Представление данных в виде интеллект-карт | - | - | - | - | - | + | + | + | + | + |
| Создание команд | + | + | + | + | + | - | - | + | - | - |
| Отображение сроков на календарь | - | + | + | - | + | - | - | - | - | - |
| Создание диаграмм | - | - | + | + | + | + | + | + | + | - |
| Авторизация | + | + | + | + | + | + | - | + | - | + |

В ходе обзора аналогов был сделан вывод, что в качестве образца для проектируемого ПО можно выделить «Trello», потому что в нем данные представляются наиболее удобно, каждый проект разделен на направления(столбцы), а сами направления разделены на задачи(карточки), в которых присутствует подробное описание задачи и сроков ее выполнения. Также, в данном аналоге присутствует возможность расширения функционала, что способствует более детальному описанию проекта.

Однако, данный аналог не подходит для рассмотренной предметной области, поскольку в нем отсутствует реализация методологии интеллект-карт, вследствие чего невозможно осуществить представление задач в виде интеллект-карт, а также автоматическое преобразование задач в интеллект-карты.

функциональные требования

На основе анализа предметной области и обзора аналогичных решений сформирован общий функционал разрабатываемой системы:

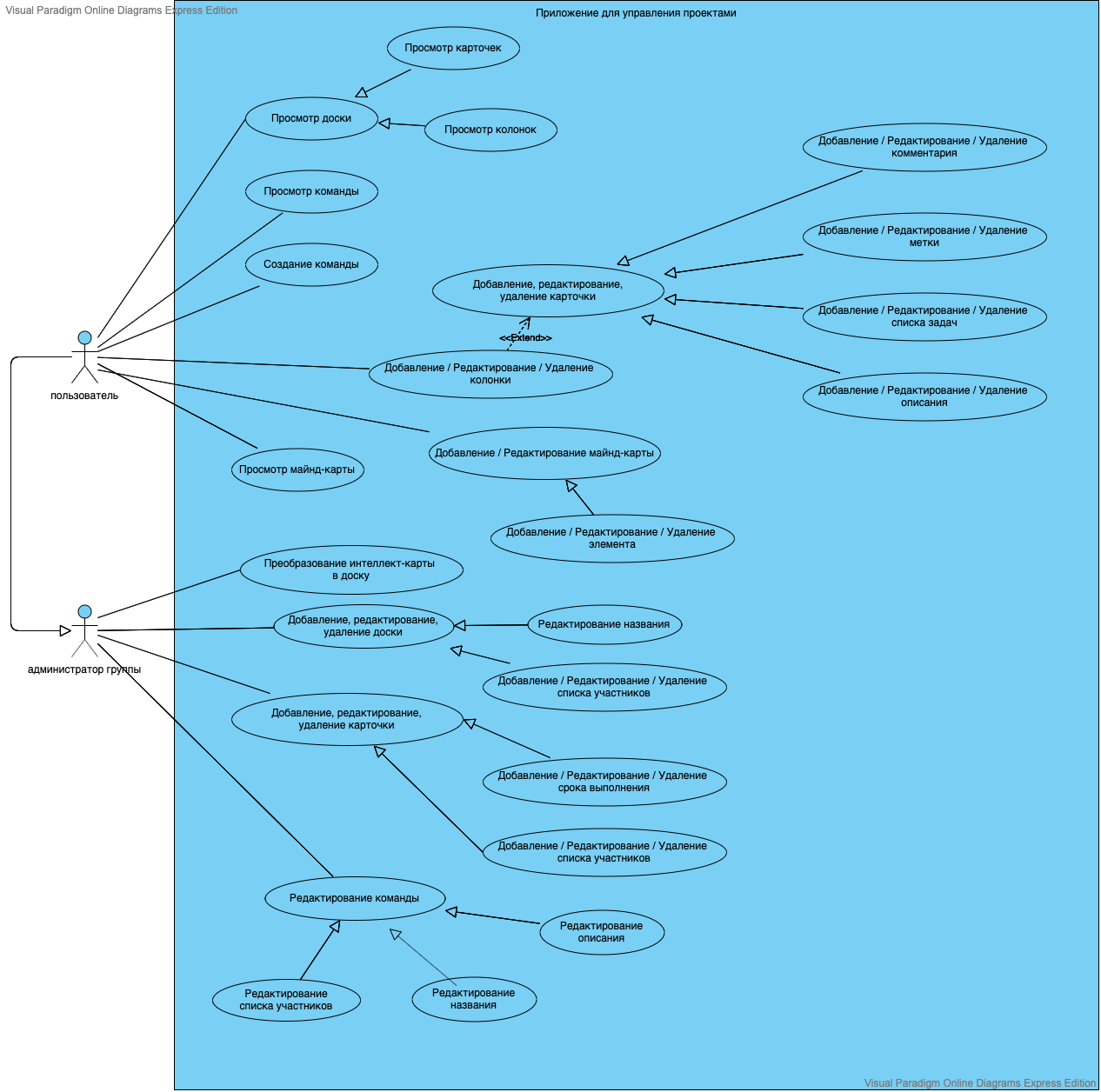
* + регистрация в системе. При регистрации следующие параметры должны быть обязательными к заполнению;
    1. логин;
    2. пароль;
    3. имя;
    4. фамилия.
  + авторизация в системе по логину и паролю;
  + создание команд;
  + создание/редактирование/удаление карточек. Карточка содержит следующие поля;
    1. название;
    2. описание;
    3. список задач;
    4. срок выполнения;
    5. список участников;
    6. метки.
  + создание/редактирование/удаление комментариев к карточкам;
  + создание/удаление/редактирование интеллект-карт;
  + преобразование интеллект-карты в доску. Элементы первого уровня вложенности преобразуются в столбцы, второго – в карточки, третьего – в списки задач;
  + просмотр информации о команде;
  + просмотр содержания доски, к которой прикреплен участник команды.

Для администратора команды должны быть реализованы дополнительные функции:

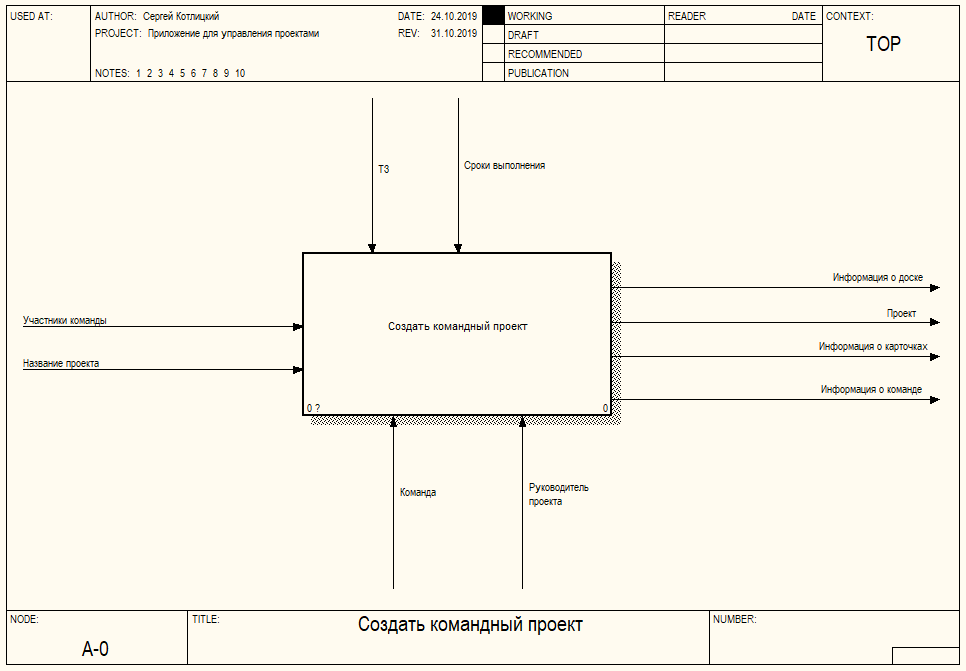
* + создание доски;
  + редактирование настроек доски;
    1. название доски;
    2. список участников.
  + редактирование информации о команде;
    1. название команды;
    2. описание команды;
    3. список участников.
  + редактирование списка участников карточки;
  + редактирование срока выполнения задачи в карточке.
  + просмотр всех досок команды;

На основе функциональных требований были составлены функциональные модели системы:

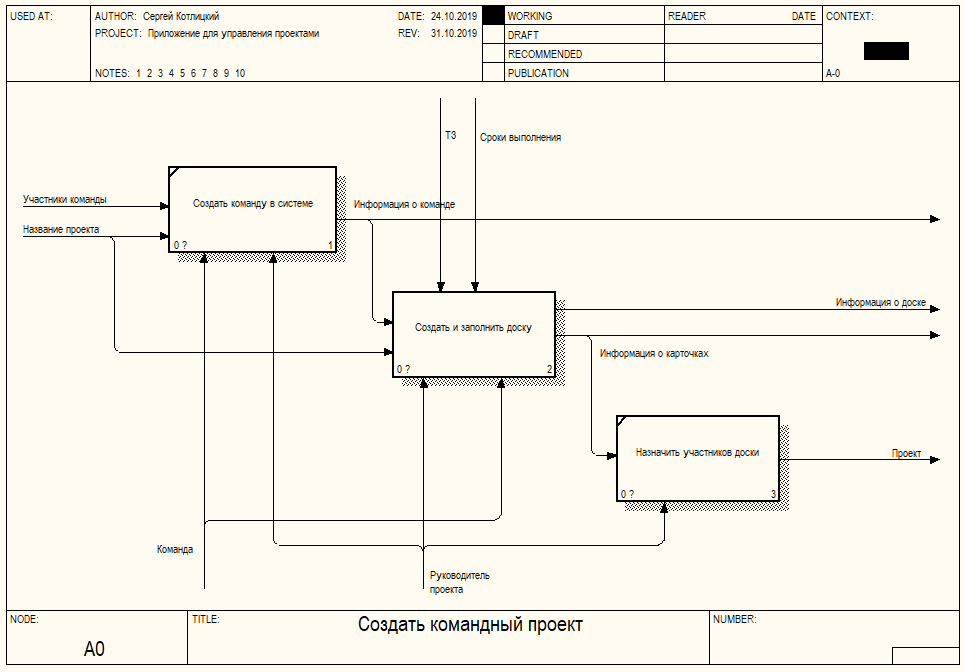
* + диаграмма вариантов использования (Рисунок 1);
  + idef0 диаграмма (Рисунок 2-4);
  + диаграмма последовательностей (Рисунок 5).



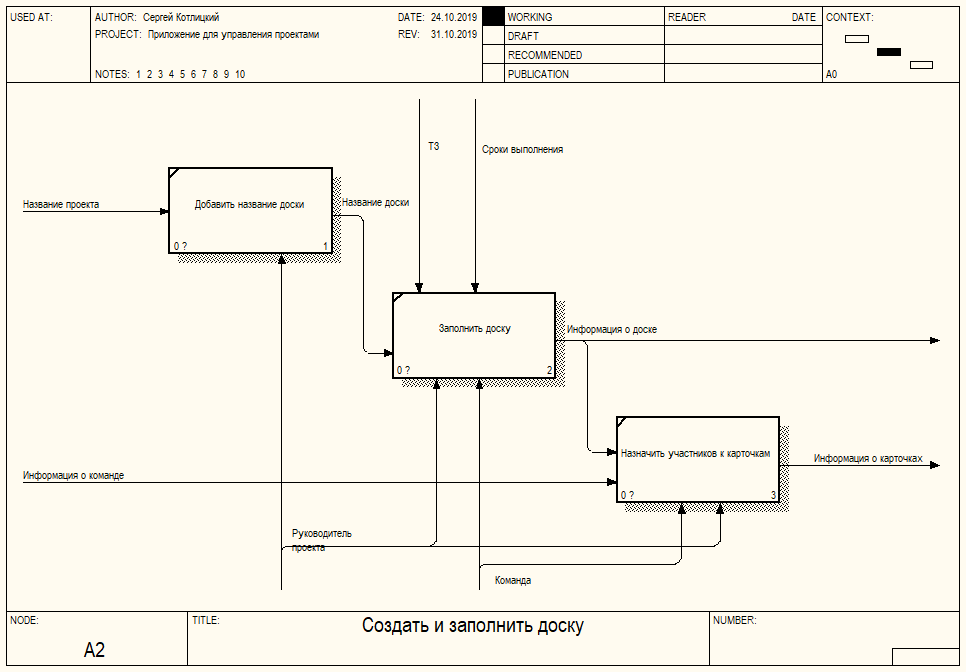
*Рисунок 1 Диаграмма вариантов использования*



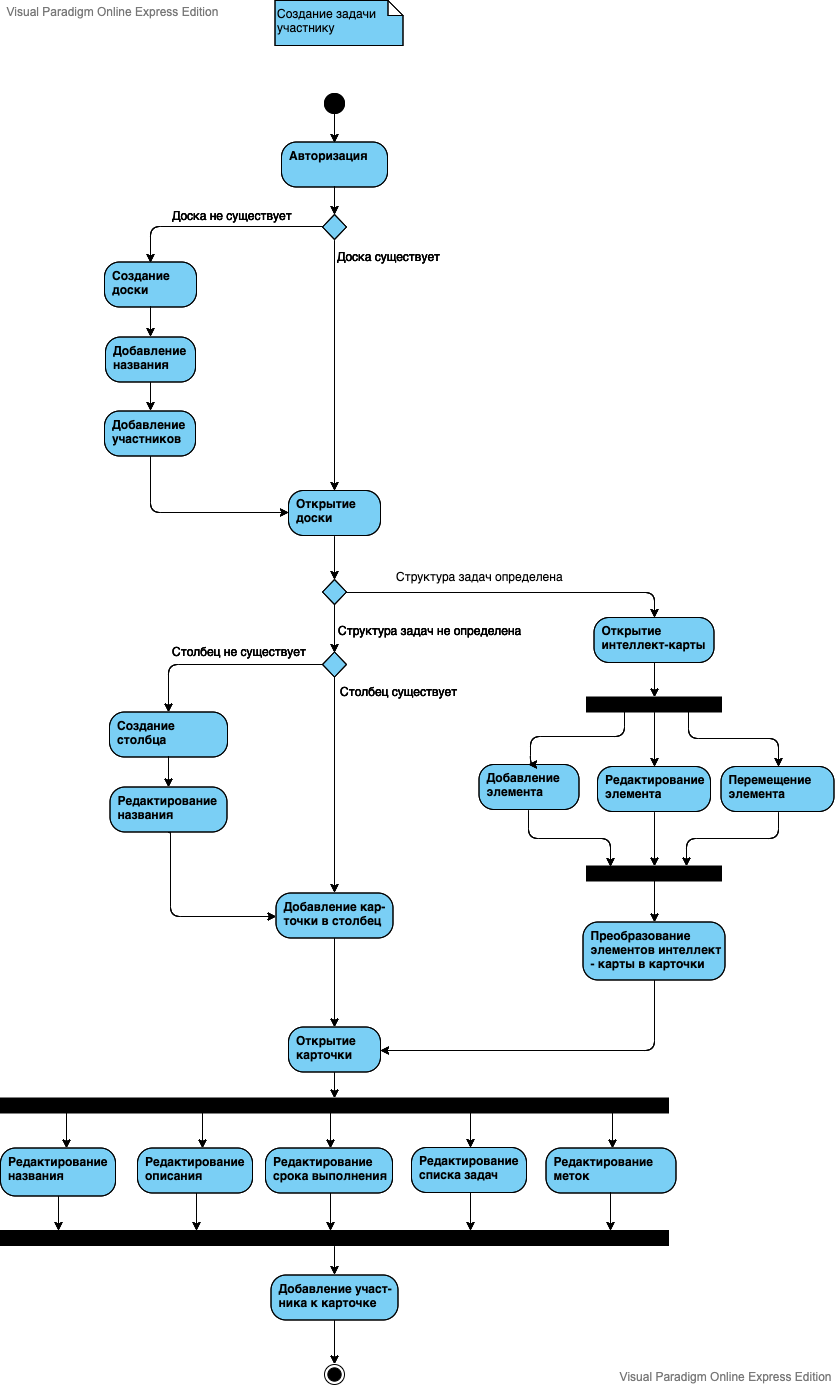
*Рисунок 2 IDEF0 диаграмма*



*Рисунок 3 IDEF0 диаграмма*



*Рисунок 4 IDEF0 диаграмма*



*Рисунок 5 Диаграмма последовательностей*

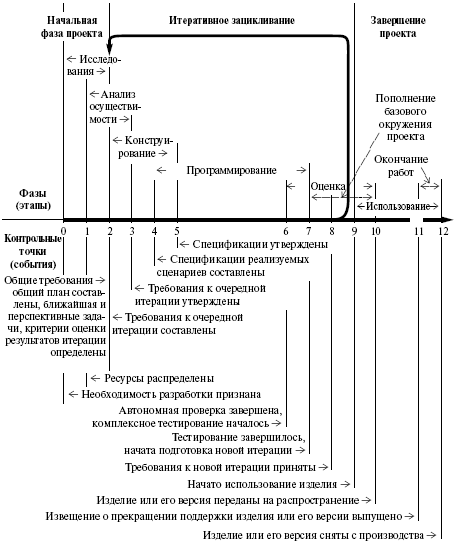
выбор модели жизненного цикла

Разработка системы планируется в несколько этапов: сначала внедрение базового функционала, затем постепенное наращивание дополнительного. Без базового функционала отсутствует возможность разрабатывать дополнительный, так как при просчетах в базовом функционале системы необходимо перестраивать всю систему.

При выборе модели жизненного цикла программного обеспечения подразумевается, что при разработке проекта есть команда, состоящая из следующих лиц:

1. Руководитель проекта.
2. Проектировщик подсистем.
3. Разработчики.
4. Тестировщик.

Исходя из того, что есть команда разработчиков, а также, что при разработке будет использоваться постепенное наращивание функционала выберем итерационную модель жизненного цикла программного обеспечения (Рисунок 6).



*Рисунок 6 Модель жизненного цикла ПО*

Данная модель жизненного цикла подходит для реализации системы для управления проектами, так как разработка функционала программы по этапам наиболее оптимальна.

Данная модель жизненного цикла имеет ряд преимуществ:

* При разработке ПО фазы проекта могут выполняться параллельно, что уменьшает время реализации;
* Продукт может реализоваться в несколько этапов: сначала основной функционал программы, затем наращивание дополнительного;
* Существуют контрольные точки каждого из фаз разработки проекта, что уменьшает возможные просчеты в момент перехода на следующие фазы разработки.

СПИСОК ИСТОЧНИКОВ

1. Составы ключевых этапов проекта // Projectimo URL: http://projectimo.ru (дата обращения: 12.10.2019).
2. Коцюба И. Ю., Шиков А. Н. ИНТЕЛЛЕКТ-КАРТЫ КАК СРЕДСТВО E-ДИДАКТИКИ В КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЯХ ОБУЧЕНИЯ // Образовательные технологии и общество. 2015.
3. Обзор инструментов для создания интеллект-карт // TexTerra URL: https://texterra.ru/ (дата обращения: 17.10.2019).
4. Коцюба И. Ю. Методы и алгоритмы поддержки нечетких семантических моделей для компьютерных тренажеров: дис. ... канд. тех. наук: 05.13.06. Санкт-Петербург, 2018.
5. Основы менеджмента программных продуктов // Интуит URL: https://www.intuit.ru (дата обращения: 18.11.2019).