

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики

Направление подготовки 09.04.02 Информационные системы и технологии

Кафедра Информационных систем и технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка программных средств для оценки трудозатрат и длительности проектов по созданию программного обеспечения

УДК _____

Студент

Группа	ФИО	Подпись	Дата
8ИМ5А	Алешков Андрей Викторович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
зав. лаб. каф. ИСТ	Копнов М.В.	К.Т.Н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Попова Светлана Николаевна			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры экологии и безопасности жизнедеятельности	Акулов Петр Анатольевич			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Информационных систем и технологий	Мальчуков Андрей Николаевич	К.Т.Н		

Томск – 2017 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)
Общепрофессиональные компетенции	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
Профессиональные компетенции	
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.
Общекультурные компетенции	
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики

Направление подготовки 09.04.02 Информационные системы и технологии

Кафедра Информационных систем и технологий

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ИМ5А	Алешков Андрей Викторович

Тема работы:

Разработка программных средств для оценки трудозатрат и длительности проектов по созданию
программного обеспечения

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

Перечень подлежащих исследованию, проектированию и разработке вопросов	Необходимо изучить методы и средства оценки трудозатрат и длительности проектов. Провести анализ и выбрать наиболее подходящий метод для создания программных средств.
Перечень графического материала	Презентация работы
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Попова Светлана Николаевна
Социальная ответственность	Акулов Павел Анатольевич
Раздел на иностранном языке	Мирошниченко Евгений Александрович
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Аналитический обзор (Analytical review)	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
зав. лаб. каф. ИСТ	Копнов М.В.	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Алешков Андрей Викторович		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики

Направление подготовки 09.04.02 Информационные системы и технологии

Уровень образования магистр

Кафедра Информационных систем и технологий

Период выполнения осенний / весенний семестр 2016/2017 учебного года

Форма представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
19.03.15	Глава «Обзор литературы»	10
28.03.15	Глава «Объекты и методы исследования»	15
20.04.15	Глава «Аналитический обзор»	25
20.05.15	Глава «Разработка приложения»	25
6.06.15	Глава «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	15
8.06.15	Глава «Социальная ответственность»	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
зав. лаб. каф. ИСТ	Копнов М.В.	к.т.н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Информационных систем и технологий	Мальчуков А.Н.	к.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ИМ5А	Алешков Андрей Викторович

Институт	кибернетики	Кафедра	ИСТ
Уровень образования	магистр	Направление	Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, представленной в научных публикациях и статьях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	Оценка потенциальных потребителей исследования, SWOT-анализ, создание графика проведения и бюджета НИ, оценка ресурсной, финансовой и экономической эффективности НИ
2. Разработка устава научно-технического проекта	
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Планирование этапов работ, определение трудоёмкости и построение календарного графика, формирование бюджета
4. Определение ресурсной, финансовой, экономической эффективности	Оценка сравнительной эффективности исследования

Перечень графического материала (с точным указанием обязательных чертежей):

1. Сегментирование рынка
2. Диаграмма FAST
3. Матрица SWOT
4. График проведения и бюджет НТИ
5. Оценка ресурсной, финансовой и экономической эффективности НТИ
6. Потенциальные риски

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Попова Светлана Николаевна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Алешков Андрей Викторович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИМ5А	Алешков Андрей Викторович

Институт	ИК	Кафедра	Информационных систем и технологий
Уровень образования	Магистр	Направление/специальность	09.04.02 Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:

<i>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</i>	<i>Объектами исследования являются методы оценки длительности проектов, способы разработки мобильных приложений, различные языки программирования, шаблоны архитектуры мобильных приложений.</i> <i>Целью магистерской диссертации является изучение методов для оценки трудозатрат и длительности проектов, разработка приложения для оценки трудозатрат и длительности проектов на предприятиях по разработке программного обеспечения.</i>
--	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1 Производственная безопасность: 1.1 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения; 1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.	<i>Производственная безопасность на стадии разработки тестовых наборов.</i> <i>1.1 В качестве вредных факторов выделены:</i> – отклонение показателей микроклимата; – недостаточная освещенность рабочей зоны. <i>1.2 В качестве опасных факторов выделены:</i> – опасность возникновения пожара; – опасность поражения электрическим током;
2 Экологическая безопасность: 2.1 Анализ воздействия объекта на окружающую среду; 2.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.	<i>2.1 Влияние объекта исследования на окружающую среду:</i> – утилизация бумажных отходов и поломанной техники; <i>2.3 Мероприятия по защите окружающей среды:</i> – отказ использования бумаги в пользу ведения записей в электронном формате.
3 Безопасность в чрезвычайных ситуациях: 3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения; 3.2 Выбор наиболее типичной ЧС; 3.3 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.	<i>3.1 Возможные чрезвычайные ситуации:</i> – пожар; – кибертерроризм; <i>3.2 Типичная чрезвычайная ситуация:</i> – пожар; – кибертерроризм. <i>3.3 Мероприятия по предотвращению наиболее типичной ЧС – пожара, согласно нормативным документам: НПБ 105-03 и ППБ 01-03.</i>

4 Правовые и организационные вопросы обеспечения безопасности: 4.1 Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; 4.2 Организационные мероприятия при компоновке рабочей зоны.	4.1 Описание правовых норм для работ, связанных с работой за ПЭВМ согласно следующим документам: – Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015). 4.2 Влияние разработанных тесовых наборов на ПО: – уменьшение времени оценки трудозатрат и длительности проектов; – увеличение прибыли предприятия; – возможность удаленной оценки трудозатрат и длительности проектов.
--	---

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры экологии и безопасности жизнедеятельности	Акулов Петр Анатольевич			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Алешков Андрей Викторович		

РЕФЕРАТ

Выпускная квалификационная работа содержит 98 с., 31 рис., 14 табл., 27 источников, 2 приложения.

Ключевые слова: МЕТОДЫ ОЦЕНКИ, ИНТЕГРАЦИЯ, JIRA, FIREBASE, PLANNING POKER, РАЗРАБОТКА, АНАЛИЗ, API, МОДЕЛИРОВАНИЕ.

Объектами исследования являются методы оценки длительности проектов, способы разработки мобильных приложений и шаблоны архитектуры мобильных приложений.

Целью работы является проведение сравнительного анализа существующих методов оценки длительности проектов, вариантов архитектуры мобильных приложений, разработка приложения для оценки длительности задач выбранным методом.

В процессе исследования проводились сбор и анализ существующих методов оценки длительности задач. Был проведен анализ возможных вариантов разработки нативных и кроссплатформенных приложений, анализ архитектур мобильных приложений, анализ языков программирования используемы для разработки мобильных приложений. Было разработано мобильное приложения для оценки длительности задач методом Planning Poker. Были изучены программные средства Jira, среда разработки X-code, облачная база данных Firebase.

В результате использования приложения «PlanningPoker» сократится время оценки длительности задач в проекте, появится возможность записывать и в последующем оценивать скорость принятия решений сотрудниками и их дальновидность в принятии решений.

Область применения: разработанное приложение планируется использовать на предприятии ООО «СибирьСофтПроект».

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

API: Application Programming Interface.

МКП: Метод критического пути.

ИТ: Информационные технологии.

jira: JIRA это продукт, предназначенный для организации процесса контроля запросов и задач, имеющий часть функциональности обычно присущей большим и дорогим системам управления проектами.

firebase: Это поставщик облачных сервисов и приложений. Облачная база данных.

оценка проекта: Это определение его значимости, ценности, выявление пользы и выгоды от его реализации в сопоставлении с альтернативами.

задача: Проблемная ситуация с явно заданной целью, которую необходимо достичь.

планирование: Это определение ясных и точных задач (и составляющих их рабочих заданий), служащих для достижения конечной, поставленной цели.

программное обеспечение: Программы, процедуры, правила и любая соответствующая документация, относящиеся к работе вычислительной системы.

оценка: Действие по применению конкретного теста к конкретному программному модулю.

мобильное приложение: Это программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах.

СОДЕРЖАНИЕ

Реферат	9
Определения, обозначения, сокращения	10
Введение.....	12
1 Обзор литературы	14
2 Объекты и методы исследования	19
2.1 Обзор методов оценки проектов	19
2.2 Обзор типов разработки	23
2.3 Обзор языков программирования	25
2.4 Обзор шаблонов архитектур приложения.....	25
3 Аналитический обзор	31
3.1 Выбор метода оценки проектов	31
3.2 Выбор функционала разрабатываемого продукта.....	32
3.3 Выбор типа разработки	36
3.4 Выбор языка программирования.....	38
3.5 Выбор шаблона архитектуры приложения	40
4 Разработка приложения	42
4.1 Архитектура клиент-серверного приложения	42
4.2 Разработка алгоритмов работы приложения	45
4.3 Разработка основных экранов приложения	46
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение....	52
6 Социальная ответственность	67
Заключение	81
Список использованных источников	82
ПРИЛОЖЕНИЕ А	85
ПРИЛОЖЕНИЕ Б.....	96

ВВЕДЕНИЕ

Проблема оценки трудозатрат и длительности выполнения проектов наиболее важна в разработке программного обеспечения. Чем точнее и быстрее происходит оценка, тем более качественно и быстро будет выполнен проект, и соответственно предприятие получит больше прибыли. Для оценки трудозатрат существует множество решений – desktop, web и мобильных приложений.

Для оценки трудозатрат, менеджером проектов может использовать различные методы, такие как: PERT, CPM, Planning Poker. Все эти методы имеют свои особенности и необходимо выбрать наиболее подходящий для компании метод.

Таким образом, актуальностью проведения работы является уменьшение времени, которое предприятие тратит на оценку трудозатрат и длительности проектов. Чем меньше предприятие тратит времени на различные прикладные операции, без потери качества выполняемых операций, тем быстрее оно выполнит проект.

Целью работы является разработка программного средства для оценки трудозатрат и длительности проектов по созданию программного обеспечения. Программное средство должно соответствовать следующим требованиям:

- отображение не оцененных задач для оценки их длительности;
- отображение сотрудников участвующих в проекте;
- возможность оценки трудозатрат из приложения;
- рассылка уведомлений включенным в голосование сотрудникам;
- отображение результатов оценки длительности администратору;
- интеграция с Jira.

Объектами исследования являются методы оценки длительности проектов, способы разработки мобильных приложений, различные языки программирования и шаблоны архитектуры мобильных приложений.

Практическая значимость результатов ВКР заключается в упрощении процесса проведения оценки трудозатрат и длительности, необходимых для их реализации проекта.

Область применения: будет востребовано компаниями, занимающимися созданием программного обеспечения.

1 ОБЗОР ЛИТЕРАТУРЫ

В настоящее время рынок мобильных приложений стремительно растет, все больше операций в интернете осуществляется не со стационарных компьютеров, а через мобильные устройства, подключенные к глобальной сети. В большинстве случаев именно со смартфонов и планшетов осуществляется большинство необходимых операций.

Кроме того мобильные приложения, в отличие от десктоп и веб приложений, являются очень удобными для постоянного использования. Эксперты в разработке программных продуктов, нередко отсутствуют в офисе. Как правило, именно в эти моменты компания остро испытывает дефицит их профессиональных компетенций. В этом случае при наличии нужных мобильных приложений у сотрудника будет возможность решить неотложный вопрос. Одним из таких вопросов является оценка трудозатрат и длительности проектов. Существует множество методов оценки трудозатрат и длительности проектов, но основными из них являются: PERT (Project Evaluation and Review Technique), CPM (Critical Path Method), а так же метод Planning Poker.

В настоящее время существует множество мобильных приложений реализующих данные методы. Рассмотрим их подробнее:

Planning Poker leanify Ltd – это мобильное приложение позволяющее проводить голосование среди других пользователей[1].



Рисунок 1.1– Planning Poker – leanify Ltd

Основными особенностями являются:

- отсутствие кроссплатформенности;
- ограниченный бесплатный функционал;
- отсутствие настроек голосования;
- возможность передачи текстовых сообщений во время голосования;
- возможность скрыть неиспользуемые/лишние карты;
- поддержка операционных систем IOS и Android.

iPERT InfiniTour – это приложение для оценки трудозатрат и длительности проектов с помощью метода PERT[2].



Рисунок 1.2 – Приложение iPERT

Основными особенностями являются:

- отсутствие кроссплатформенности;
- платное приложение;
- возможность расчета командного и индивидуального рейтинга PERT;
- возможность сравнения рейтинга PERT среди пользователей;
- поддержка операционных систем IOS.

Scrum Poker Cards artArmin – это бесплатное мобильное приложение для ОС Android[3].

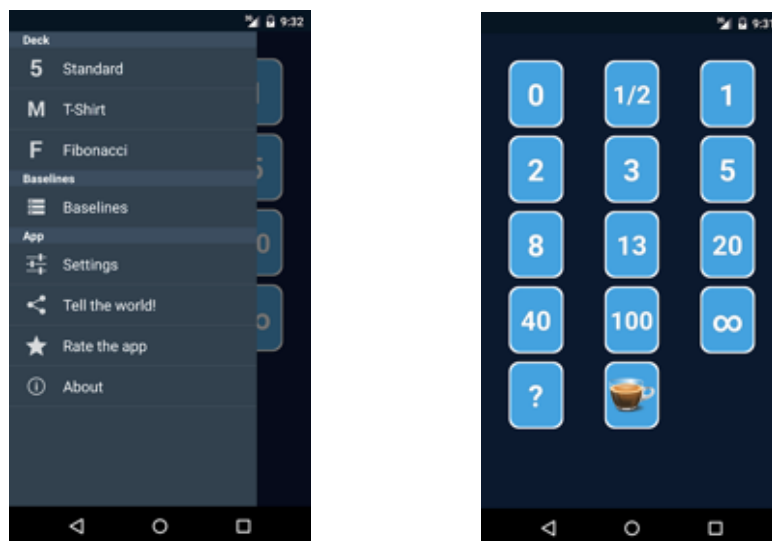


Рисунок 1.3 – Scrum Poker Cards – artArmin

Основными особенностями являются:

- необходимость предварительной регистрации;
- отсутствие гибкой настройки голосования;
- отсутствие возможности приглашать в голосование не зарегистрированных пользователей.

Planning Poker CESAR – это бесплатное мобильное приложение, поддерживающее операционные системы IOS и Android[4].

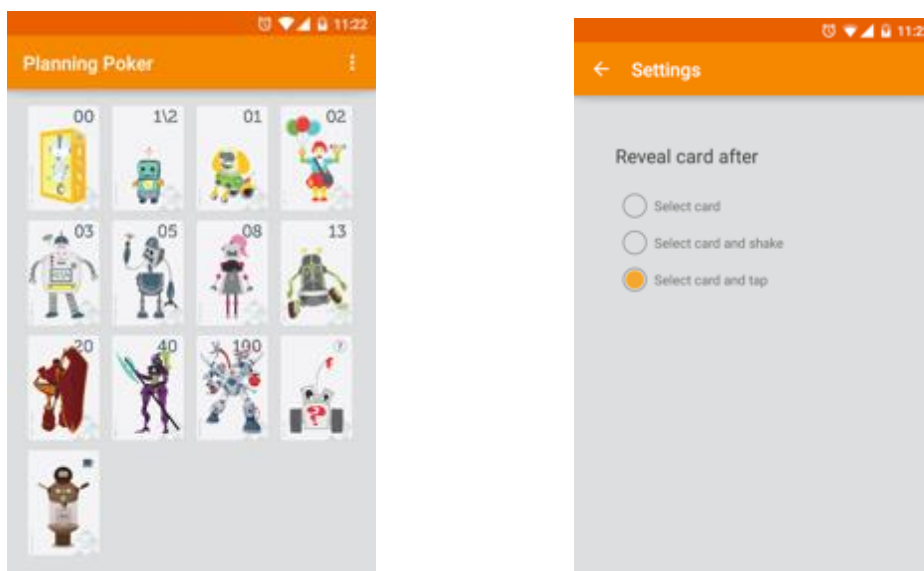


Рисунок 1.4 – Planning Poker – CESAR

Основными особенностями являются:

- возможность приглашения не зарегистрированного пользователя принять участие в голосовании;

- возможность обмена сообщениями во время голосования.

Scrum Time Planning Poker Endava – это мобильное приложение для проведения оценки трудозатрат методом Planning Poker[5].



Рисунок 1.5 – Scrum Time – Planning Poker – Endava

Основными особенностями являются:

- отсутствие платы за пользование;
- наличие предварительной регистрации;
- несколько колод для голосования;
- поддержка операционной системы Android и IOS.

PERT Ilmasoft Pro – это программа для оценки длительности задач методами PERT и CPM[6].



Рисунок 1.6 – PERT – ILMASOFT PRO

Основными особенностями являются:

- описание метода PERT;
- описание метода CPM;
- наличие калькулятора для метода PERT;
- наличие калькулятора для метода CPM;
- присутствует платный контент.

Время от времени, эксперты в разработке программных продуктов принимают участие в профильных конференциях и семинарах, таким образом отсутствуют в офисе. Как правило, именно в эти моменты компания остро испытывает дефицит их профессиональных компетенций. В этом случае при наличии нужных мобильных приложений у него будет возможность решить неотложный вопрос.

Таким образом, платформой для реализации программы оценки трудозатрат и длительностей проектов было выбрано мобильное приложение.

2 ОБЪЕКТЫ И МЕТОДЫ ИССЛЕДОВАНИЯ

2.1 Обзор методов оценки проектов

Рассмотрим существующие методы оценки проектов:

1) Метод PERT позволяет определить ожидаемую продолжительность (ОП) работ проекта на основе трех вероятностных оценок времени:

- оптимистическая оценка (ОО);
- пессимистическая оценка (ПО);
- наиболее вероятная оценка (НВО).

Для учета вероятностного характера оценок продолжительности работ можно использовать метод пересмотра и оценки программы [7].

Эксперты имеют возможность дать три оценки длительности работ, что позволяет в различной степени учесть риски, влияющие на их выполнение. Каждой оценке длительности назначаются весовые коэффициенты. В результате рассчитывается средневзвешенная (наиболее ожидаемая оценка - НОО) продолжительность работы с агрегированным уровнем риска.

$$НОО = \frac{ОО + 4 * НВО + ПО}{6}$$



Рисунок 2.1 – Графическое представление распределения продолжительностей

Стандартное отклонение позволяет определить максимальную и минимальную ожидаемую продолжительность:

$$\text{Ст. отклонение} = \frac{\text{Пессимистическая оценка} - \text{Оптимистическая оценка}}{6}$$

Это дает возможность Менеджеру проекта построить несколько графиков выполнения проекта:

- оптимистический;
- наиболее вероятный;
- пессимистический;
- ожидаемый PERT.

Риски изменения состава или логической структуры работ в методе не учитываются.

2) Метод графической оценки и пересмотра программ (GERT) позволяет учесть риск изменения состава работ при наступлении определенных событий или по результатам выполнения предшествующих работ [8]. В сетевой модели GERT могут создаваться точки ветвления или точки выбора, после которых планируются несколько независимых цепочек работ, не все из которых выполняются.

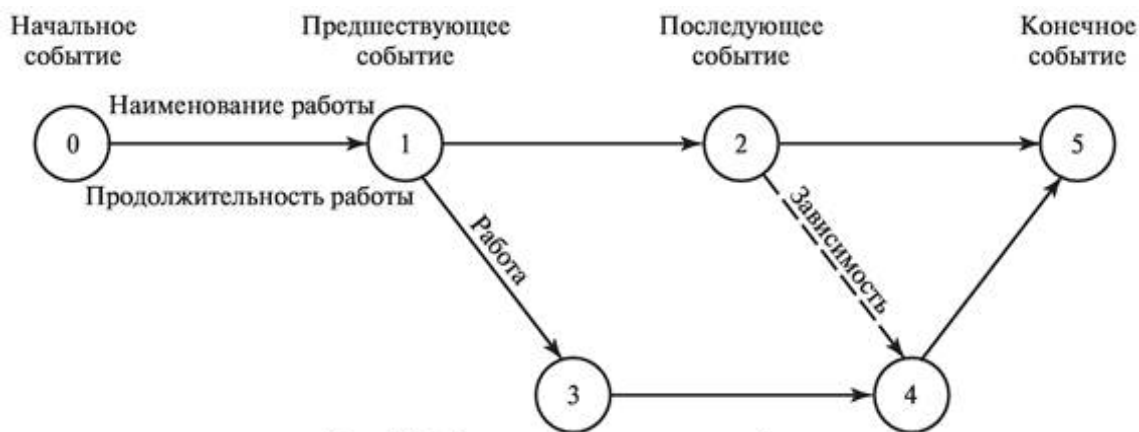


Рисунок 2.2 – Метод GERT

Сетевая модель представляет собой вероятностную сеть, учитывающую возможность различного состава работ проекта.

Учет рисков, влияющих на продолжительность работ, осуществляется также, как и в методе PERT, то есть по результатам вычисления средневзвешенной оценки продолжительности на базе трех оценок, выданных экспертами.

3) Метод Монте-Карло (Monte Carlo Analysis – MCA) использует модель проекта, например сетевой график, для анализа поведения[9]. С этой целью МСА делает случайную выборку из распределения вероятности для каждой операции при условии выполнения проекта сотни и даже тысячи раз, что позволяет получить статистическое распределение и приблизительное значение длительности проекта (рис. 3).

С помощью таких распределений можно количественно описать риски, характеризующие различные варианты расписаний, альтернативных стратегий исполнения, путей выполнения операций и даже отдельные операции.

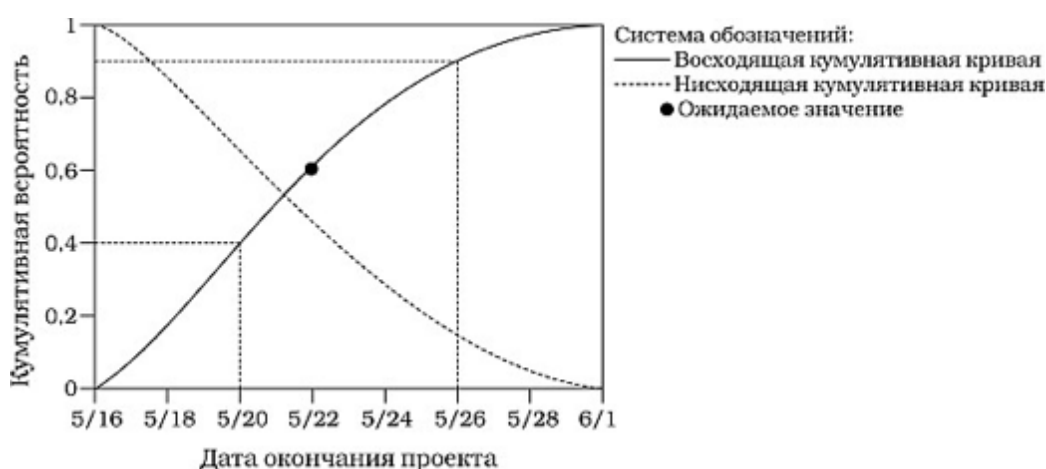


Рисунок 2.3 - Кумулятивное распределение длительности проекта, полученное в ходе анализа Монте-Карло

Обычно МСА имеет дело с рисками для расписания, стоимости и потока денежной наличности, хотя могут анализироваться и другие аспекты, например качество конечного продукта проекта.

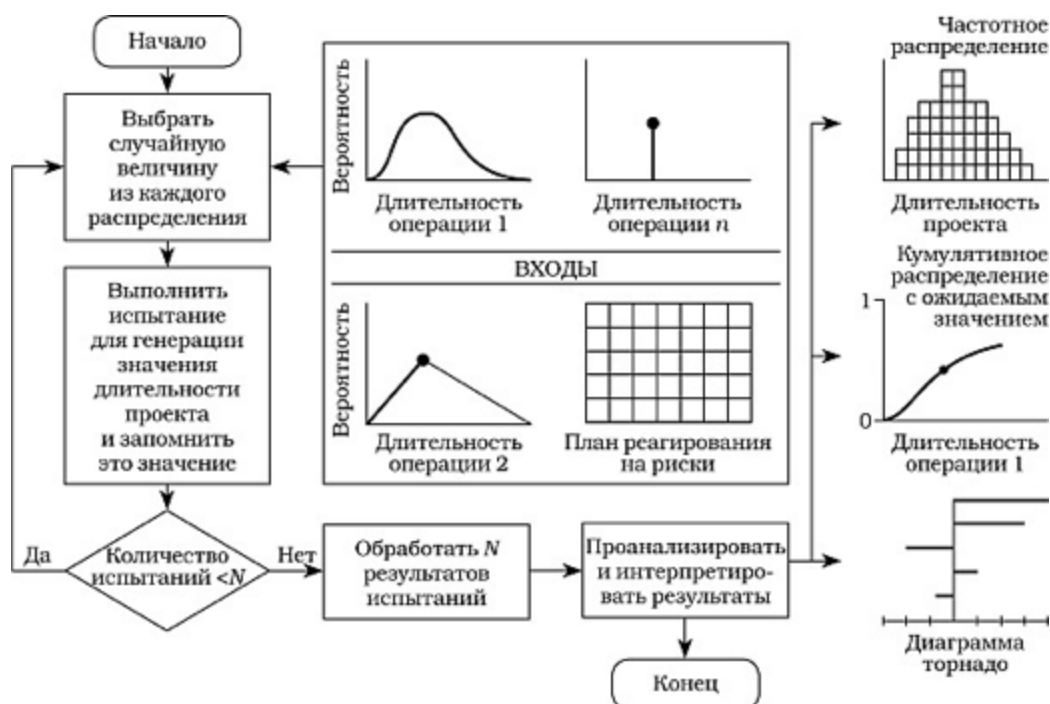


Рисунок 2.4 - Проведение анализа Монте-Карло для рисков расписания

Ключевая роль в анализе Монте-Карло принадлежит следующим информационным элементам:

- плану управления рисками;
- плану реагирования на риски;
- логической схеме (сетевой диаграмме) или расписанию проекта;
- распределениям вероятностей.

4) Метод Planning Poker Покер использует технику оценки, основанную на достижении договорённости, для оценки сложности предстоящей работы или относительного объёма решаемых задач при разработке программного обеспечения[10].

В основе Покер планирования лежат следующие принципы:

- коллективное обсуждение;
- коллективное согласие;
- вовлечение экспертов;
- группирование и Относительность размеров;
- накапливание и использование опыта предыдущих оценок.

Покер планирование включает в себя группирование или сортировку элементов одинакового размера, а также определяет относительные размеры этих элементов. Для упрощения оценки последующих элементов задачи, команда использует исторические данные и практики оценки.

Перед началом проведения Покер планирования команда должна решить какую шкалу или ряд чисел использовать для оценки. В Покер планировании чаще всего используется оценочная шкала, предложенная Майком Коном и основанная на видоизмененном ряде чисел Фибоначчи: 1, 2, 3, 5, 8, 13, 20, 40 и 100.

Альтернативная шкала, которой пользуются команды, основывается на ряде чисел в степени 2: 1, 2, 4, 8, 16, 32, 64 и так далее.

2.2 Обзор типов разработки

Разработка мобильных приложений, становится все более и более востребованной, таким образом разработчики ищут новые возможности для быстрой и качественной реализации проектов. Долгое время разработка мобильных приложений являлась исключительно нативной, это обусловлено достаточным качеством и быстродействием существующих IDE.

Таким образом, для того, что бы написать качественные программы для двух наиболее распространенных платформ (Android, IOS) необходимо было изучать такие языки программирования, как: Java, Kotlin, Swift[11], Objective-C[12]. Благодаря необходимости знания множества языков, компанией Facebook был представлен фреймворк ReactNative[13], для написания на котором требовалось знание JavaScript.

В ReactNative, в качестве языка программирования используется JavaScript, или несколько иной подход смешивающий синтаксис JavaScript и HTML по типу JSX. Инструменты для ReactNative – это в основном текстовый редактор, отладчик Chrome и несколько других инструментов для сборки и тестирования.

Для нативной разработки iOS необходимо знание языка Swift или Objective-C, а для Android – Java или Kotlin. Инструменты для создания приложения находятся в IDE каждой платформы: для iOS это X-code, а для Android это Android studio. Так же для программирования в каждой из этих IDE, необходимо знать как работать в этой среде, как работают системы отладки и сборки.

ReactNative берет свое начало в веб-разработке и заимствует свою базовую части именно оттуда, в то время как для нативной разработки такой базовой части не существует. Это исключительно нативные сенсорные платформы, которые необходимо изучать. Так же стоит отметить, что когда изучена одна из IDE, вторая изучается достаточно быстро.

Фреймворки являются очень значимым показателем выбора в сторону нативной разработки. Необходимо изучить как работают все интересующие вас функции, но в отличии от ReactNative, все эти функции задокументированы и реализованы.

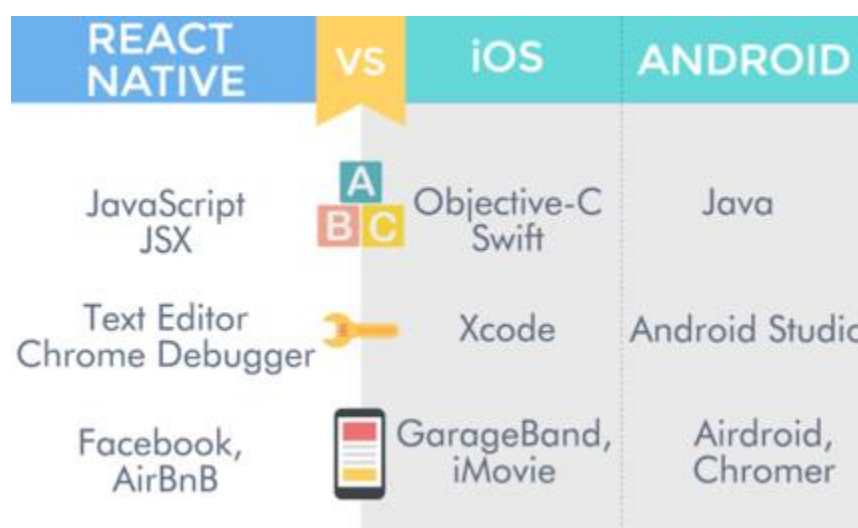


Рисунок 2.5 – Нативная и кроссплатформенная разработка

Swift и ReactNative совсем недавно появились в программировании, при этом каждый из них отлично документирован. Официальная документация apple, позволяет разобраться во всех тонкостях нового языка, с примерами применения и полезными советами начинающим разработчикам. В то же время

официальная документация Facebook по ReactNative, тоже раскрывает все возможности фреймворка, с примерами кода и типовой документацией.

2.3 Обзор языков программирования

Создание нативного IOS приложения, подразумевает выбор одного из языков программирования. В настоящее время существует 2 языка для нативной IOS разработки, это Swift и Objective-C.

Objective-C появился в 1984 году, он использовался Apple для операционных систем IOS и OS X, и являлся основным решением для разработчиков.

В 2014 году, Apple представил новый, современный язык программирования – Swift. Этот новый язык программирования был разработан для IOS, MacOS, tOS, watshOS и Linux. Кроме того он был разработан для работы с существующими библиотеками написанными на Objective-C, для продуктов Apple.

Если судить по активности пользователей на GitHub (одна из популярных платформ для ИТ-проектов), более 2500 проектов переключились на Swift в 2014 году. В настоящее время, популярность этого языка растет очень быстро, но несмотря на все улучшения и нововведения Swift, Objective-C пользуется большой популярностью среди IOS разработчиков.

Таким образом в 2017 году, большинство стартапов начали свою деятельность именно на Swift, в отличии от больших корпораций.

2.4 Обзор шаблонов архитектур приложения

Существует множество различных архитектурных шаблонов для разработки мобильных IOS приложений. Выбор архитектуры будущего приложения очень важный этап проектирования. Правильный выбор архитектуры, позволит проводить поиск ошибок, а так же оперативно реагировать на изменения внутри класса.

В настоящее время есть множество вариантов шаблонов проектирования архитектуры:

- MVC;
- MVP;
- MVVM;
- VIPER.

MVC, MVP, MVVM предполагают включение объектов приложения в одну из трех категорий:

- Models – ответственные за данные домена или уровень доступа к данным, который манипулирует данными;
- Views – ответственны за уровень представления (Graphics User Interface), все элементы с префиксом “UI”;
- Controller, Presenter, ViewModel – выполняет роль посредника между Model и View, отвечающего за изменения Model. Реагируя на действия пользователя, выполняемые во View и обновляя View с изменениями из Model.

Далее подробнее рассмотрим каждый из представленных выше шаблонов.

При использовании шаблона традиционного MVC, View не имеет состояния, он просто отображается контроллером после изменения модели[14]. Таким образом, этот шаблон подразумевает полную перезагрузку страницы после любого действия пользователя.

Таким образом, реализация традиционного MVC в приложении IOS не имеет смысла из-за архитектурной проблемы – все три объекта связаны друг с другом и каждая сущность знает о двух других.

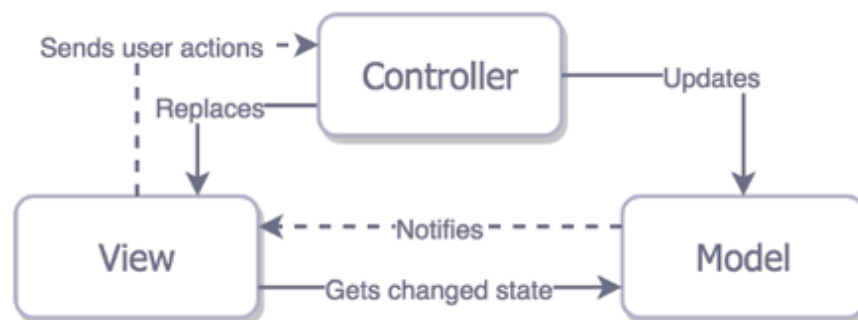


Рисунок 2.6 – Традиционный MVC

В шаблоне архитектуры Apple MVC контроллер будет являться посредником между View и Model, что бы эти два объекта не знали о существовании друг друга[15]. Наиболее часто используемым объектом будет являться контроллер. В этом есть плюсы, потому что в приложении должно быть место для всей сложной бизнес-логики, которая не вписывается в Model.

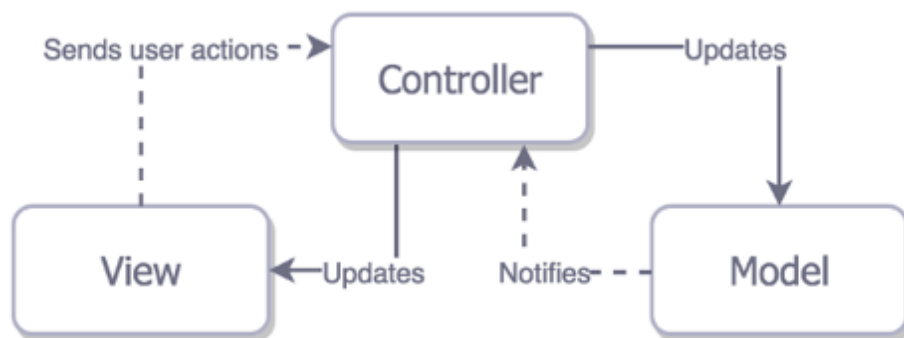


Рисунок 2.7 – Apple MVC

В реальности шаблон Apple MVC работает по-другому. Перегруженный контроллер содержит в себе большое количество бизнес-логики, превращая Model-View-Controller в Massive-View-Controller. Cocoa MVC поощряет писать Massive-View-Controller, потому что они очень сильно вовлечены в жизненный цикл View, что приводит к неразделимости этих двух сущностей.

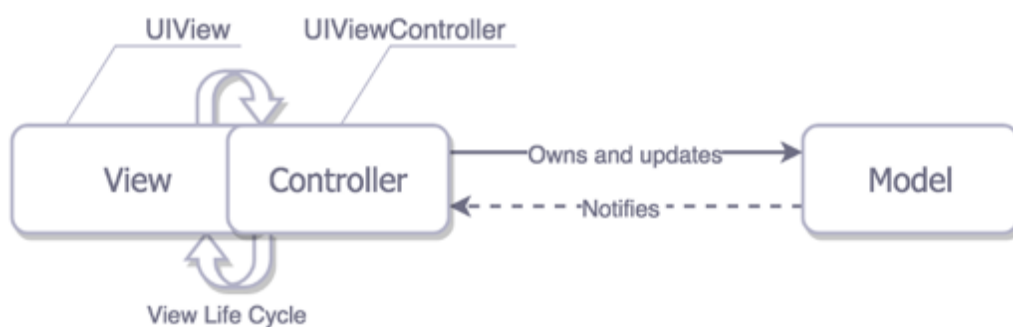


Рисунок 2.8 – Практический Apple MVC

MVP (Cocoa MVC's promises delivered) похож на Apple MVC, но это не так[16]. В Apple MVC – View связан с контроллером, в то время как медиатор MVP – Presenter, не имеет ничего общего с жизненным циклом контроллера, поэтому в Presenter нет кода для установки макета экрана, но он отвечает за обновление View с актуальными данными и состоянием.

MVP это архитектурный шаблон, показывающий проблемы сборки, которая возникает из-за наличия трех фактически разделенных слоев.

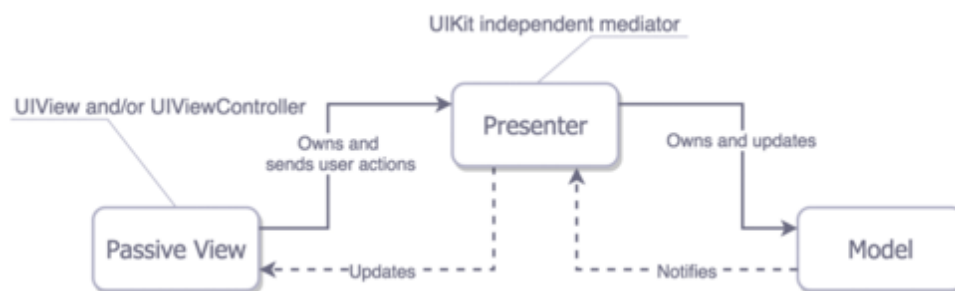


Рисунок 2.9 – MVP

Существует и другой формат MVP – MVP Supervising Controller[16]. Этот вариант включает прямую зависимость View и Model, в то время как Presenter (Supervising Controller) все еще обрабатывает действия из View и способен его изменять. Разделение смутной ответственности, является недопустимым, так же как и сильная связь View и Model.

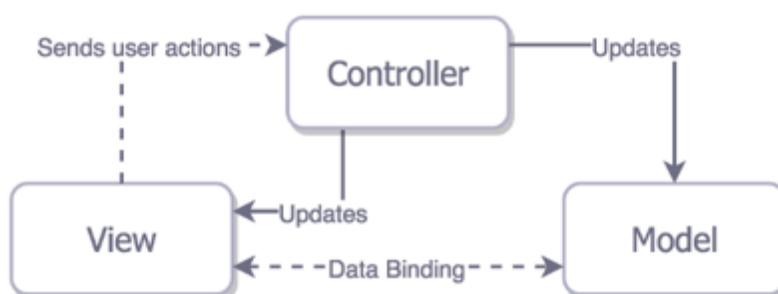


Рисунок 2.10 - MVP (With Bindings and Hooters)

MVVM рассматривает контроллер View, как View, а так же подразумевает отсутствие связи между View и Model[17]. View-Model обеспечивает независимое представление View и его состояния. View-Model вызывает изменения в Model и обновляет себя с обновленной Model, а поскольку View и View-Model соединены, View соответственно тоже обновляется.

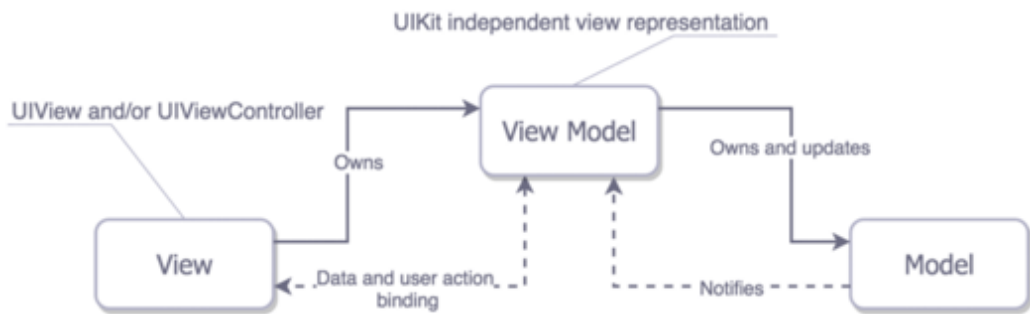


Рисунок 2.11 – MVVM

VIPER позволяет полностью разделить обязанности, и обеспечить полную независимость модулей[18]. Он состоит из:

- View – отвечающее за исполнение произошедших в Presenter изменений;
- Interactor – содержит бизнес-логику, связанную с данными (Entities) или сетевыми устройствами. В его обязанности входят такие операции, как создание новых экземпляров объектов, или выборку их с сервера. Для этих целей используются службы и менеджеры, не являющиеся частью модуля VIPER – они являются внешней зависимостью;
- Presenter – содержит связанную с UI (независимую от UIKit) бизнес-логику, вызывает методы в Interactor;
- Entities – в этом классе хранятся простые объекты данных;
- Router – отвечает за переходы между различными модулями VIPER.

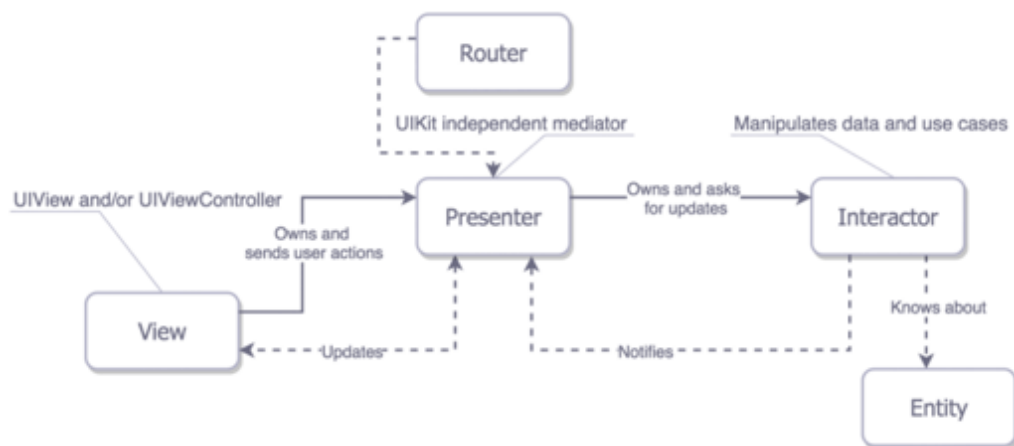


Рисунок 2.12 - VIPER

Модуль VIPER может соответствовать как одному экрану в приложении, так и всему приложению, четких указаний как выбрать размер модуля не приводится.

VIPER – это первый шаблон в котором явно рассматривается навигационная ответственность, которая решается при помощи Router.

3 АНАЛИТИЧЕСКИЙ ОБЗОР

3.1 Выбор метода оценки проектов

При детальном рассмотрении методов оценки проектов, были выделены основные особенности каждого из них.

Метод PERT позволяет дать три вида оценок: пессимистичную, наиболее вероятную, и оптимистичную. Метод может определить вероятности завершения проекта в необходимый период времени и в заданные сроки, и рассчитать значение стандартного отклонения длительности и максимальную и минимальную ожидаемую продолжительность каждой задачи и всего проекта.

Метод GERT применим в случае, когда последующие задачи начинаются после завершения некоторого числа из предшествующих задач. Метод графической оценки и пересмотра программ (GERT) учитывает риск изменения состава работ при наступлении определенных событий или по результатам выполнения предшествующих работ. Следовательно, можно учитывать риски на уровне проекта в целом, а не только риски на уровне отдельных работ. В результате моделирования методом GERT получаются графики, показывающие вероятность различной длительности и неопределенности работ по проекту.

Метод Монте-Карло основан на получении большого числа реализаций случайного процесса, формирующийся так, что бы вероятностные характеристики совпали с характеристиками решаемых задач. Метод Монте-Карло имеет ряд преимуществ:

- простота вычислений, необходимых для анализа Монте-Карло, даже сложные уравнения или моделирование корреляций и других взаимозависимостей не создают трудностей;
- легкость использования, так как есть программные пакеты, которые автоматизируют проведение МСА, то есть компьютер может сам выполнять весь объем работ, необходимых для генерации результирующего распределения;

- высокая скорость процесса – изменения модели, которая анализируется методом Монте-Карло позволяет сравнить данную модель с предшествующими, а исследования поведения любой из моделей является не трудоемким процессом;

Метод Монте-Карло имеет такие недостатки, как сложность, так как построена на концепции вероятности, с которой многие менеджеры проектов еще не овладели. Приблизительность выходных распределений вероятности в случае малого числа итераций, однако проблема легко решается путем увеличения количества итераций вплоть до достижения требуемого уровня точности.

Метод Planning Poker позволяет быстро оценить ту или иную задачу, так как количество вариантов для голосования сильно ограничено и голосующие быстро приходят к необходимому решению.

Покер планирование стимулирует команду к обсуждению элементов задачи для достижения согласия по оценкам объема работ над ними. При этом обязательно в обсуждение привлекают экспертов, входящих в команду.

В результате исследования был выбран метод PlanningPoker, потому что при его реализации существует возможность быстро оценить задачу всей командой. Этот метод минимизирует эффект привязки путем опроса каждого участника команды таким образом, что никто не знает результатов голосования до определенного времени.

3.2 Выбор функционала разрабатываемого продукта

Разрабатываемый продукт обладает следующим функционалом:

- возможность оценки трудозатрат и длительности проектов;
- отображение не оцененных задач в Jira;
- отображение сотрудников в проекте Jira;
- возможность записи итоговой оценки в Jira;
- рассылка уведомлений включенным в голосование сотрудникам;
- отображение результатов оценки длительности администратору;

- гибкая настройка голосования;
- возможность автоматического запуска следующей запланированной для голосования задачи;
- предоставление администратору графических данных о результатах голосования;
- интеграция с Firebase.

Для определения отличительных особенностей создаваемого программного средства от существующих аналогов, была создана таблица, в которой представлены сравнительные характеристики программ для оценки трудозатрат и длительности проектов, использующих выбранный ранее метод PlanningPoker. Как видно из таблицы 3.1 в сравнении участвуют и WEB приложения, для возможности ознакомления с их функционалом. Функционал приложений достаточно схож с друг другом, как у мобильных, так и у WEB приложений. В результате главной отличительной особенностью разрабатываемой системы является возможность интеграции с клиентом JIRA, для выгрузки не оцененных задач, а так же для записи результатов голосования.

Таблица 3.1 – Сравнение аналогов приложений PlanningPoker

Название	Производитель	Поддержка ОС	Стоимость	Пробный период	Необходимость предварительной регистрации	Интеграция с Jira, Redmine, TFS и т.д.	Варианты колод для оценки	Ограничение численности голосующих
PlanITpoker [19]	PlanITpoker	Windows, Mac	Free	-	-	-	+	До 50 человек
PlanningPoker.com[20]	Mountain Goat Software	Windows, Mac	Pro = 49.95 \$ Enter = 0	1 месяц (Pro)	+	-	Pro + Enter -	Pro до 50 Enter < 5
Planning Poker	SSP	IOS	Free	-	-	Jira	-	До 30 человек
Scrum Poker Cards (Agile)	artArmin	Android	Free	-	+	-	+	Нет информации
Planning Poker	CESAR	Android, IOS	Free	-	-	-	+	Нет информации
Scrum Time - Planning Poker	Endava	Android, IOS	Free	-	+	-	+	Нет информации

Продолжение таблицы 3.1

Название	Производитель	Гибкая настройка голосования	Возможность автоматического последовательного запуска голосования	Возможность приглашения не зарегистрированного пользователя	Предоставление графических данных по голосованию	Возможность обмена сообщениями во время голосования
PlanITpoker	PlanITpoker	+	-	+	+	+
PlanningPoker.com	Mountain Goat Software	Pro + Standard - Enter -	Pro + Standard - Enter -	Pro + Standard - Enter -	Pro + Standard - Enter -	Pro + Standard + Enter +
Planning Poker	SSP	+	+	-	+	-
Scrum Poker Cards (Agile)	artArmin	-	-	-	-	-
Planning Poker	CESAR	-	-	+	-	+
Scrum Time Planning Poker -	Endava	-	-	-	-	-

3.3 Выбор типа разработки

При проведении анализа рынка возможностей разработки мобильных приложений, были выбран нативный вариант IOS разработки, так как для разработки приложения необходимо выбрать строго типизированный, компилируемый, объектно-ориентированный язык. Java и Objective-C/ Swift отлично для этого подходят. ReactNative официально не поддерживается ни Apple, ни Google. Это означает, что есть большая вероятность, что последующие обновления операционных систем Android и IOS могут ограничить, или исключить работу ReactNative. К тому же, многие эксперты утверждают, что некоторые нативные приложения, в настоящее время невозможно повторить при помощи ReactNative. Так же ReactNative в настоящее время не имеет возможности разрабатывать приложения для Apple Watch, и предпосылок к этому нет. Все нововведения IOS и Android будут оставаться впереди ReactNative, в отличие от нативной разработки, когда создатели операционной системы параллельно помогают создавать IDE, с всевозможными новшествами. Необходимо иметь ввиду долговечность проекта. Apple и Google выпускают собственные IDE для собственных ОС, а Facebook не выпускает собственных ОС. Таким образом проект развиваемый при помощи нативной разработки выглядит более стабильным и долговечным.

Таким образом, плюсы кроссплатформенной разработки ReactNative заключаются в том, что ReactNative существенно упрощает разработку сразу под 2 платформы мобильных устройств, тем самым значительно сокращает время работы. В базовой структуре, фреймворк производит различия между двумя платформами, это очень важно, потому как нативная разработка под IOS и Android очень сильно отличается, что ведет к удорожанию продукта из-за необходимости найма специалистов знающих Swift / Objective-C и специалистов по Java/ Kotlin. Следовательно все базовые API отличаются друг от друга, например использование GPS, создание анимации пользовательского интерфейса, даже создание телефонных вызовов отличается. Для вхождения в

разработку с помощью ReactNative, нет необходимости осваивать новые языки если JavaScript, HTML, CSS уже изучены, а программирование напоминает создание веб интерфейса для веб версий. Инструменты и архитектура приложения такие же как и в веб- разработке.

Для кроссплатформенной разработки необходимо знать, что типичное мобильное приложение имеет два основных компонента – приложение и его серверная часть. Независимо от того нативное это, или кроссплатформенное приложение, база данных будет едина, в то время как ее проектирование составляет почти 50% работ по проекту. В результате выбор метода разработки мобильного приложения не является основной проблемой в проекте.

Программист умеющий писать для Android и для IOS нативные программы, при первоначальном написании для одной из платформ приложения, потратит на него значительно больше времени, чем для написания под оставшуюся платформу. Скорость будет достигнута, благодаря тому, что будет значительно меньше времени тратиться на написание кода, благодаря отстроенной архитектуре приложения, правильной абстракцией классов, правильно расстановкой компонентов пользовательского интерфейса и рефакторинга кода при каких либо изменениях. Таким образом при написании первого приложения, будут решены наиболее важные вопросы в проектировании и программировании.

В ReactNative, помимо архитектурных вопросов необходимо решать вопросы, как на уровне совместимости платформы, так и на уровне проектирования. В итоге будет необходимо решать куда более много серьезных вопросов чем в нативной разработке.

При повторном использовании написанного ранее кода, на других языках программирования (C/C++) или с использованием нативной игровой платформы – ReactNative сможет использовать их, но необходимо будет написать собственный код, благодаря чему теряется смысл в переиспользовании нативных проектов. Написание этих библиотек заново будет

требовать высоким уровнем знаний о нативной разработке, для того, что бы была возможность воссоздать работу низкоуровневого кода.

Наиболее известные продукты написанные на ReactNative (Facebook, Instagram, AriBnB), были написаны экспертами нативной разработки. Таким образом, для начинающих разработчиков мобильных приложений нативная разработка является наиболее простой в освоении.

3.4 Выбор языка программирования

Проведя анализ преимуществ и недостатков Swift и Objective-C предпочтительней оказался Swift.

Swift является более производительным языком, позволяющим быстрее создавать новые приложения. В таблице 3.2 представлены основные различия этих языков.

Таблица 3.2 – Сравнение языков программирования Swift и Objective-C

	Swift	Objective-C
Работа с другими языками программирования	Синтаксис похож на другие известные языки программирования, такие как: Python, Go, JavaScript.	Синтаксис намного отличается от других языков программирования, поэтому достаточно труден в изучении.
Длина кода	Код является более рациональным, что снижает его длину и время разработки.	Реализация одинаковых функций, требует в 2 раза больше строк кода чем на Swift.
Интерактивность	Можно создавать приложения в интерактивном режиме.	Нельзя создавать приложения в интерактивном режиме.
Сообщество использующее язык	Swift намного проще в изучении, но профессионалов в настоящее время, намного меньше чем в Objective-C.	Большая часть приложений IOS индустрии, написана на Objective-C.

Продолжение таблицы 3.2

Безопасность	Безопасность является одним из основных понятий Swift. Его конструкции исключают несколько типов потенциальных ошибок в Objective-C.	Не так безопасен как Swift.
Классы и подклассы	В Swift, класс не имеет подклассов.	В Objective-C класс имеет подкласс.
Кортежи	Кортежи позволяют группировать несколько элементов в один.	В Objective-C кортежей не существует.
Открытый исходный код	Swift – это язык с открытым исходным кодом. Так же он используется в Linux, что потенциально создает возможность его применения в разработке программ для Android OS.	Не обладает исходным кодом, и ориентирован исключительно на устройства от Apple.
Полнота и стабильность языка	Разработка Swift еще не завершена, и Apple постоянно выпускают, и добавляют новые возможности.	Objective-C уже создан и стабильно используется.
Описание методов в коде	Можно нажать на имя класса для просмотра его заголовка, и вложенных функций.	Нет подобных функций.

При использовании Swift, процесс разработки упрощается, становясь доступным для изучения начинающими программистами, благодаря чему уменьшаются затраты на высоко квалифицированных специалистов. Так же не маловажную роль играет поддержка развития языка, со стороны Apple.

Основные преимущества Swift, можно выделить быстрый рост и открытый код. Свифт выбирают потому, что он бесплатен, и благодаря этому доступен большому числу разработчиков. На рисунке 3.1 показан график роста аудитории разработчиков Swift с момента запуска.



Рисунок 3.1 – График роста аудитории Swift

Уменьшение времени написания приложения, позволяет сократить расходы, и увеличить прибыль компании. Swift подходит для написания новых приложений, в самые короткие сроки.

3.5 Выбор шаблона архитектуры приложения

В результате анализа архитектурных шаблонов были сделаны следующие выводы:

- 1) Традиционный MVC – не применим к современной IOS разработке.
- 2) Apple MVC является лучшим архитектурным шаблоном, с точки зрения скорости разработки, но заметно усложняет поддержку приложения при его увеличении.

- 3) MVP предлагает хорошую тестируемость, но и значительное увеличение кода. Его особенности заключаются в распределение – большая часть обязанностей, разделены между Presenter и Model, при этом у View остается минимум задач. Так же преимуществом является тестируемость – можно протестировать большую часть бизнес-логики из за практических пустого View. MVP, как же как и традиционный MVC является недопустимым для использования в IOS разработке.

- 4) Особенности MVVM заключаются в распространении – View в MVVM имеет больше обязанностей, чем View в MVP, поскольку MVVM обновляет состояние из ViewModel, а второй только передает все события в

Presenter и не обновляет себя. В тестируемости – ViewModel ничего не знает о View что позволяет без проблем его тестировать. View также может быть протестирован, но поскольку он зависит от UIKit – тестирование его не целесообразно. Таким образом, MVVM сочетает в себе преимущества всех рассмотренных ранее подходов к архитектуре IOS приложений, в то же время не требует дополнительного написания кода для обновления View. Тестируемость так же находится на хорошем уровне;

5) Viper превосходит все представленные архитектурные паттерны в распределённой, тестируемости, и простоте освоения новых модулей. Основными особенностями Viper является распространение – VIPER отличается отличной способностью к распределению обязанностей, тестируемость – при лучшем распределении, получается лучшая тестируемость. Так же удобство использования – маленькое количество кода в блоках VIPER, позволяет без труда переключаться с одной задачи на другую, увеличивая тем самым полезную работу программиста.

Можно сделать вывод, что для наилучшего распределения обязанностей, и покрытия приложения тестами приложения, больше всего подходит архитектурный шаблон – Viper.

4 РАЗРАБОТКА ПРИЛОЖЕНИЯ

4.1 Архитектура клиент-серверного приложения

Разрабатываемое мобильное приложение для оценки трудозатрат представляет собой клиент двухзвенной клиент-серверной архитектуры (рисунок 4.1). Двухзвенной называется архитектура, распределяющая три базовых компонента между клиентом и сервером.

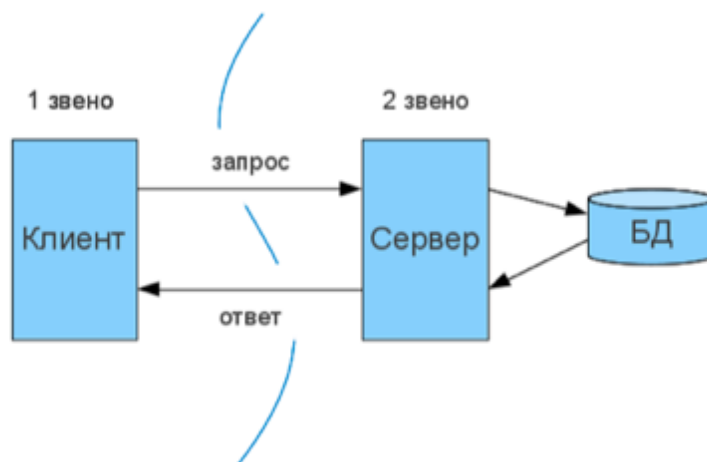


Рисунок 4.1 – Архитектура приложения

Клиент – это программное обеспечение предоставляющее пользователю интерфейс для взаимодействия с системой. Клиент не взаимодействует с базой данных напрямую, так же он не содержит большого количества бизнес-логики. Клиент отвечает за получение и отображение данных полученных с сервера. Так же клиент отвечает за отправку полученных от пользователя данных, или выполняет запрос к данным по команде пользователя.

В приложении для IOS, все взаимодействия с пользователем происходят при помощи сенсорного дисплея.

Архитектура IOS приложения представляет собой 4 слоя:

- 1) Cocoa Touch – содержит в себе множество фреймворков, которые отвечают за внешний вид приложения, а так же обеспечивает базовую инфраструктуру приложения и поддержку основных технологий, таких как: многозадачность, сенсорный ввод, push- уведомления;

2) Media layer – содержащий в себе аудио и видео технологии, используемые для реализации мультимедийных функций приложения;

3) Core Services – состоит из основных системных сервисов для приложений, где основными сервисами являются Core Foundation и Foundation frameworks, определяющие основные типы используемые всеми приложениями, а так же он содержит в себе технологии для работы с местоположением, iCloud, интернет сетями;

4) Core OS layer – включает в себя такие сервисы как безопасность, локальная идентификация и Bluetooth.

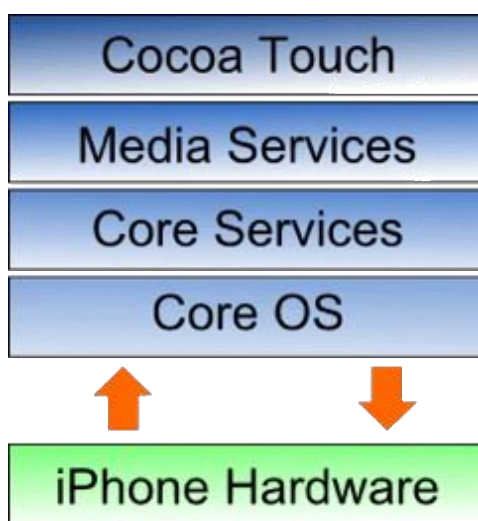


Рисунок 4.2 – Архитектура IOS приложения.

Сервер – расположен на втором уровне логики, содержит большую часть бизнес-логики системы. Сервер взаимодействует с базой данных, после сего генерирует JSON модели и отправляет их клиенту через интерфейс API. Так же происходит процесс обмена информацией в другую сторону, когда пользователь изменяет данные в приложении клиента, затем они отсылаются на сервер, после чего изменяется база данных.

На рисунке 4.3 изображена схема взаимодействия приложения с серверами Jira и Firebase.



Рисунок 4.3 – Схема работы приложения

Разработанное мобильное приложение взаимодействует с сервером Jira для отправки клиенту информации о неоценённых задачах и пользователях:

- время создания задачи;
- общая информация о задаче;
- имени и фамилии пользователя;
- никнейм пользователя;
- права доступа пользователя.

Так же получать информацию об оценке выставленной администратором голосования.

Сервер Firebase взаимодействует с приложением для получения следующей информации:

- информацию о имени и фамилии голосовавшего пользователя;
- информацию о оценке пользователя;
- информацию о времени старта голосования;
- информацию о времени окончания голосования.

А так же передачи информации приложению:

- информацию о именах и фамилиях пользователей участвовавших в голосовании;
- информацию о оценках пользователей участвовавших в голосовании;
- информацию о времени завершения голосований пользователями;

- информацию о скорости оценивания задач пользователями.

4.2 Разработка алгоритмов работы приложения

Была произведена разработка двух алгоритмов, для реализации программы оценки трудозатрат и длительности проектов и длительности проектов.

1) Алгоритм проведения голосования.

Данный алгоритм был необходим на начальном этапе понимания целей разрабатываемой системы. В нем кратко изложена логика от декомпозиции задач, до записи в них полученных данных о трудозатратах. Алгоритм представлен на рисунке 4.4.



Рисунок 4.4 – Алгоритм проведения голосования.

2) Алгоритм работы программы.

Разработанный алгоритм показывает всю бизнес-логику, находящуюся на стороне клиентского приложения, такую как:

- идентификация пользователя – клиентское приложение определяет причастность пользователя к группе «User» или «Administrator» и в зависимости от причастности к одной из них приложение меняет логику работы под нужный тип пользователя;

- загрузка доступных голосований;

- процедура старта голосования – при создании голосования администратором, всем включенным в голосование пользователям отправляется сообщение о времени его начала, тексте задачи для голосования и названии голосования;

- процедура оценки полученных голосов – после проведения голосования, приложение запрашивает данные с Firebase, производит подсчеты общего среднего, и выводит результаты для показа администратору.

С данным алгоритмом можно ознакомиться в приложении Б.

4.3 Разработка основных экранов приложения

Разработка приложения проводилась в среде X-code, при помощи языка Swift, по паттерну VIPER. Разметка элементов экранов проводилась при помощи Constrainit, разметка масштабируется в зависимости от размера экрана при помощи «Auto-Layout».

Для реализации графиков была использована библиотека «Scrollable-GraphView».

Экран авторизации позволяет пользователю ввести логин и пароль используемый в Jira. Экран авторизации содержит 3 Label, 1 Button и 2 TextField.

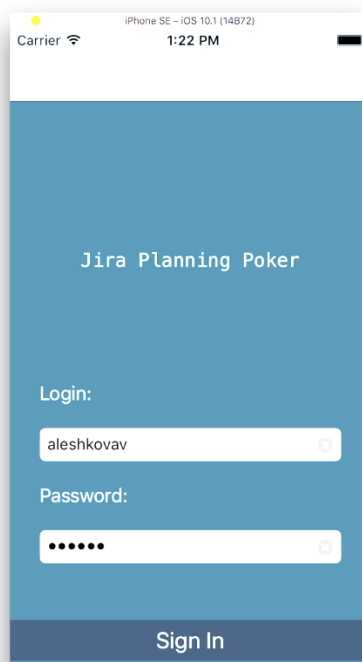


Рисунок 4.5 – Экран авторизации

Экран выбора проекта позволяет администратору голосования выбрать необходимый для голосования проект, после чего происходит переход на экран выбора задач. Экран состоит из нижнего Tab-bar отвечающего за выбор необходимой операции – создание голосования или голосование. Так же экран включает в себя TableView с атрибутами ячейки: Название проекта, краткое описание проекта.

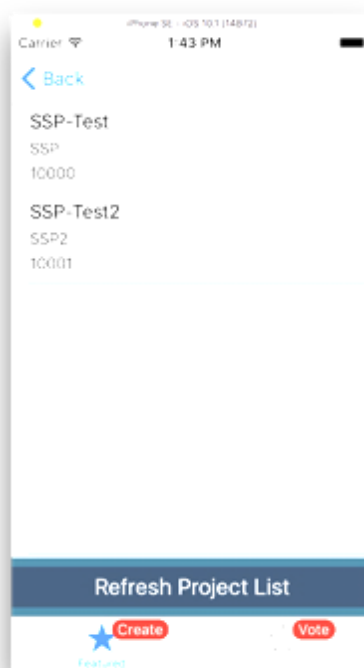


Рисунок 4.6 – Экран выбора проекта

Экран выбора задачи показывает задачи, не имеющие оценок в выбранном проекте.

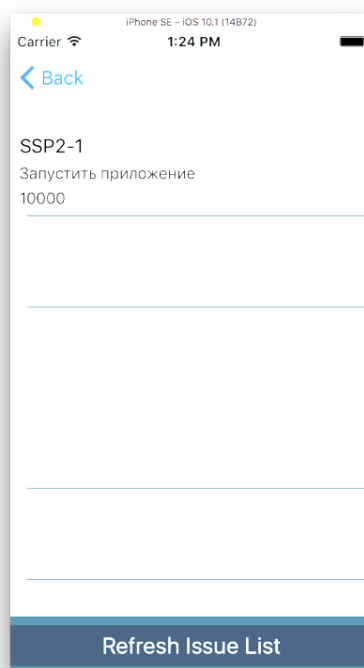


Рисунок 4.7 - Экран выбора задачи

Экран выбора пользователей показывает всех сотрудников задействованных в проекте. До голосования допускается любой выбранный администратором пользователь. При нажатии администратором на кнопку

«Start voting», приложение автоматически создает запись в базе данных Firebase для последующего сохранения в ней результатов выбранных пользователей.

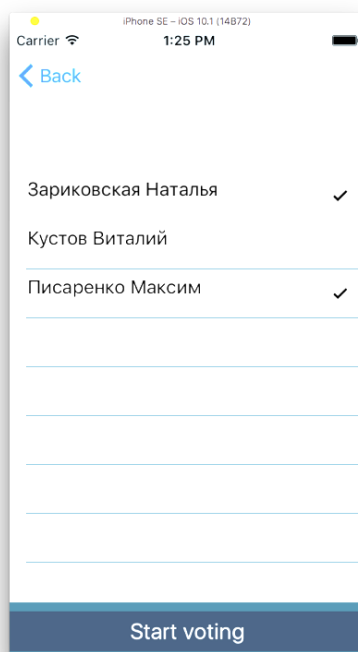


Рисунок 4.8 – Экран выбора пользователей

На экране голосования пользователь выбирает одну из оценок по заданной задаче. При завершении голосования, клиентское приложение записывает в созданную администратором запись Firebase, выбранное значение.

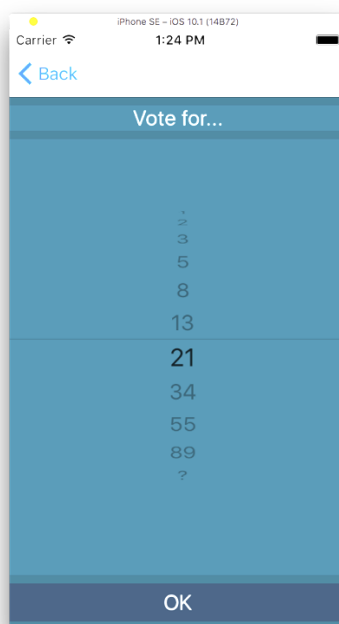


Рисунок 4.9 – Экран голосования

Экран «Графическое представление результатов» помогает визуально оценить какие оценки были выставлены. В случае если администратор голосования видит очень большую разницу в оценках, можно провести повторный раунд голосования.

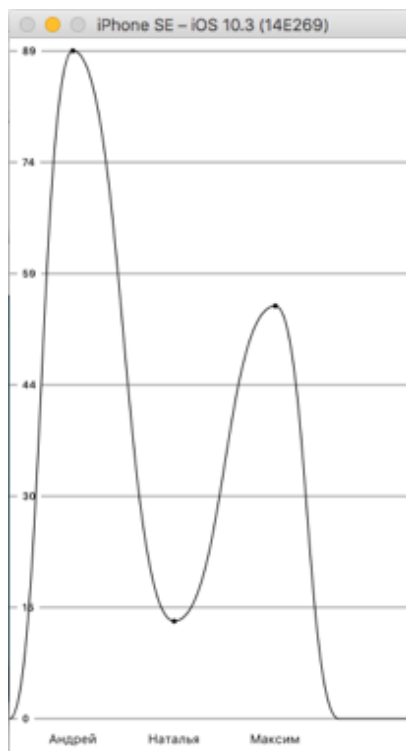


Рисунок 4.10 – Экран с графической информацией

После проведения голосования и ознакомления с результатами, администратор может отправить предложенную приложением оценку, или ввести заново свою.

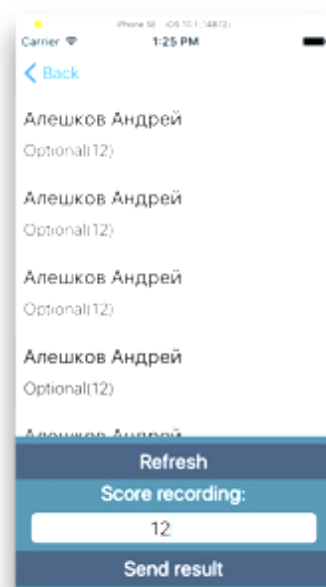


Рисунок 4.11 – Экран отправки результатов

В приложении так же был реализован метод оценки длительности проектов «PERT». Экран состоит из трех TextField, семи Label и одной Button.



Рисунок 4.12 – Метод оценки длительности проектов - PERT

Метод оценки длительности разработки PERT, предварительно помогает самостоятельно оценить задачи, предлагаемые на голосование.

5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1. Потенциальные потребители результатов исследования

Целевым предприятием разрабатываемого программного средства, в первую очередь является компания ООО «СибирьСофтПроект», занимающаяся разработкой программного обеспечения.

Целевым рынком для результатов выпускной квалификационной работы является рынок по разработке программного обеспечения, оценке трудозатрат и длительности проектов.

5.2 Организация и планирование работ

5.2.1 Структура работ в рамках научного исследования

При организации работ в рамках данной работы необходимо планировать занятость каждого участника проекта в работе. На данном этапе определяется полный перечень работ, распределение времени работ между всеми участниками. В качестве структуры, показывающей необходимые данные, используется линейный график работ, представленный в таблице 5.1.

Таблица 5.1 – Перечень этапов работ и распределение исполнителей

Основные этапы	№	Содержание работ	Исполнитель и
Разработка задания на выполнение	1	Постановка целей и задач	НР
	2	Составление списка задач для выполнения.	НР, И
	3	Подбор и изучение материалов по теме.	НР, И
Исследование	4	Выбор метода оценки трудозатрат и длительности проектов.	И
	5	Выбор типа разработки программных средств.	И
	6	Выбор языка программирования.	И
	7	Выбор шаблона архитектуры приложения.	И
Разработка мобильного	8	Изучение среды программирования Xcode.	И
	9	Изучение необходимых библиотек.	И

приложения	10	Написание алгоритмов работы программы.	И
	11	Разработка бизнес-логики приложения.	И
	12	Разработка экранов приложения.	И
	13	Тестирование и отладка.	И
	14	Анализ результатов разработки.	НР, И
Анализ и оформление результатов	15	Оформление работы	И
	16	Подведение итогов	НР, И

НР – научный руководитель;

И – инженер – в его роли действуют исполнитель ВКР.

5.2.2 Продолжительность этапов работ

Для определения трудоемкости выполнения работ, необходимо на основе экспертной оценки ожидаемой трудоемкости выполнения каждой работы рассчитать длительность работ в рабочих и календарных днях для каждого из вариантов исполнения работ последующим формулам:

$$t_{\text{ож}i} = \frac{3t_{\text{min}i} + 2t_{\text{max}i}}{5} \quad (5.1)$$

где $t_{\text{ож}i}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\text{min}i}$ – минимально возможная трудоемкость, чел.-дн.;

$t_{\text{max}i}$ – максимально возможная трудоемкость, чел.-дн.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях ведется по формуле:

$$T_{\text{РД}} = \frac{t_{\text{ож}}}{K_{\text{ВН}}} \cdot K_{\text{Д}}, \quad (5.2)$$

где $t_{\text{ож}}$ – продолжительность работы, дн.;

$K_{\text{ВН}}$ – коэффициент выполнения работ, учитывающий влияние внешних на соблюдение предварительно определенных длительностей, в рамках данной ВКР установим $K_{\text{ВН}} = 1$;

$K_{\text{Д}}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ, примем $K_{\text{Д}} = 1,1$.

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{\text{КД}} = T_{\text{РД}} \cdot T_K, \quad (5.3)$$

где $T_{\text{РД}}$ – продолжительность выполнения этапа в календарных днях;

T_K – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях. Он рассчитывается по формуле:

$$T_K = \frac{T_{\text{КАЛ}}}{T_{\text{КАЛ}} - T_{\text{ВД}} - T_{\text{ПД}}}, \quad (5.4)$$

Где $T_{\text{КАЛ}}$ – календарные дни, $T_{\text{КАЛ}} = 365$;

$T_{\text{ВД}}$ – выходные дни, $T_{\text{ВД}} = 52$;

$T_{\text{ПД}}$ – праздничные дни, $T_{\text{ПД}} = 10$.

Подставив значения в формулу 5.4, получим следующий результат:

$$T_K = \frac{365}{365 - 52 - 10} = 1,205. \quad (5.5)$$

Все рассчитанные значения представлены в таблице 5.3.

Для наглядного отображения графика и распределения работ между участниками проекта использована диаграмма Ганта.

Для построения диаграммы календарных работ используются данные, из таблицы 5.2.

Таблица 5.2 - Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях Т _{рд}		Длительность работ в календарных днях Т _{рд}	
	t _{min} , чел-дни		t _{max} , чел-дни		t _{ож} i, чел-дни					
	НР	И	НР	И	НР	И	НР	И	НР	И
Постановка целей и задач	2	–	5	–	3,2	–	3,52	–	4,241	–
Составление списка задач для выполнения.	2	2	5	5	3,2	3,2	3,52	3,52	4,241	4,241
Подбор и изучение материалов по теме.	1	3	3	15	1,8	7,8	1,98	8,58	2,385	10,338
Выбор метода оценки трудозатрат и длительности проектов.	–	5	–	15	–	9		9,9	0	11,92
Выбор типа разработки программных средств.	–	2	–	6	–	3,6		3,96	0	4,771
Выбор языка программирования.	–	7	–	12	–	9		9,9	0	11,929
Выбор шаблона архитектуры приложения.	–	5	–	14	–	8,6		9,46	0	11,399
Изучение среды программирования Xcode.	–	8	–	17	–	11,6		12,76	0	15,375
Изучение необходимых библиотек.	–	1	–	2	–	1,4		1,54	0	1,855
Написание алгоритмов работы программы.	–	1	–	3	–	1,8		1,98	0	2,385
Разработка бизнес-логики приложения.	–	5	–	14	–	8,6		9,46	0	11,399
Разработка экранов приложения.	–	7	–	12	–	9		9,9	0	11,929
Тестирование и отладка.	–	2	–	6	–	3,6		3,96	0	4,771
Анализ результатов разработки.	3	12	7	21	3,9	15,6	4,29	17,16	5,169	20,677
Оформление работы	–	1	–	2	–	1,4		1,54	0	1,855
Подведение итогов	1	1	2	3	1,4	1,8	1,54	1,98	1,855	2,385
Итого:	9	62	22	147	13,5	96	14,85	105,6	17,89 1	127,229

Таблица 5.3 – Календарный план-график проведения работ

№ Э	$T_{\text{кд}} \text{ НР}$	$T_{\text{кд}} \text{ И}$	Февраль			Март			Апрель			Май			Июнь		
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	4,241	0	■														
2	4,241	4,241		■													
3	2,385	10,338			■												
4	0	11,92				■											
5	0	4,771					■										
6	0	11,929						■									
7	0	11,399							■								
8	0	15,375								■							
9	0	1,855									■						
10	0	2,385										■					
11	0	11,399											■				
12	0	11,929												■			
13	0	4,771													■		
14	5,169	20,677														■	
15	0	1,855															■

НР – ■ ; И – ■

5.2.3 Расчет накопления готовности проекта

В данном пункте будет описан расчет текущих состояний работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего этапа выполнен общий объем работ по проекту в целом.

Степень готовности определяется формулой (5.5)

$$СГ_i = \frac{ТР_i^H}{ТР_{общ.}} = \frac{\sum_{k=1}^i ТР_k}{ТР_{общ.}} = \frac{\sum_{k=1}^i \sum_{j=1}^m ТР_{kj}}{\sum_{k=1}^i \sum_{j=1}^m ТР_{kj}}, \quad (5.6)$$

где $ТР_i^H$ – накопленная трудоемкость i -го этапа проекта по его завершении;

$ТР_{общ.}$ – общая трудоемкость проекта;

$ТР_k$ – трудоемкость k -го этапа проекта, $k = \overline{1, i}$;

$ТР_{kj}$ – трудоемкость работ, выполняемых j -м участником на k -м этапе, здесь $j = \overline{1, m}$ – индекс исполнителя, в данной работе $m = 2$, так как в разработке проекта участвуют научный руководитель и исполнитель.

Применительно к таблице (5.3) величины $ТР_{ij}$ ($ТР_{kj}$) находятся в столбцах ($Т_{РД}$ (НР), $j = 1$) и ($Т_{РД}$ (И), $j = 2$). $ТР_{общ.}$ равна сумме чисел из итоговых клеток этих столбцов. Пример расчета $ТР_i$ (%) и $СГ_i$ (%) на основе этих данных содержится в таблице (5.4).

Таблица 5.4 – Нарастание технической готовности работы и удельный вес каждого этапа

Этап	$ТР_i$, %	$СГ_i$, %
Постановка целей и задач	2,92	2,92
Составление списка задач для выполнения.	5,84	8,77
Подбор и изучение материалов по теме.	8,77	17,53
Выбор метода оценки трудозатрат и длительности проектов.	8,21	25,75
Выбор типа разработки программных средств.	3,29	29,04
Выбор языка программирования.	8,22	37,26

Этап	ТР _i , %	СГ _i , %
Выбор шаблона архитектуры приложения.	7,85	45,11
Изучение среды программирования Xcode.	10,59	55,71
Изучение необходимых библиотек.	1,28	56,98
Написание алгоритмов работы программы.	1,64	58,63
Разработка бизнес-логики приложения.	7,85	66,48
Разработка экранов приложения.	8,22	74,70
Тестирование и отладка.	3,29	77,99
Анализ результатов разработки.	17,81	95,80
Оформление работы	1,28	97,08
Подведение итогов	2,92	100,00

5.3 Расчет сметы затрат на выполнение проекта

В состав бюджета выполнения работ входит стоимость всех расходов, необходимых для их выполнения. При формировании бюджета используется группировка затрат по следующим статьям:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные начисления;
- оплата услуг связи;
- прочие (накладные расходы) расходы.

5.3.1 Расчет накопления готовности проекта

Данная статья включает стоимость всех материалов, используемых при разработке проекта: приобретаемые материалы, необходимые для создания ВКР. Также включаются затраты на канцелярские принадлежности и диски.

Все расчеты сведены в таблицу 5.5.

Таблица 5.5 – Расчет материальных затрат

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, (З _м), руб.
		И	И	И
диск CD/RW	Шт	2	30	60
бумага формата А4	Уп.	1	290	290
тонер для принтера	Шт.	1	250	250
Итого				600

Транспортно-заготовительные расходы (ТЗР) составляют 5 % от отпускной цены материалов, тогда расходы на материалы с учетом ТЗР равны

$$С_{\text{мат}} = 600 \cdot 1,05 = 630 \text{ руб.}$$

5.3.2 Расчет заработной платы

Данная статья расходов включает в себя заработную плату научного руководителя и студента, а также премии и доплаты. Расчет выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя.

Величина месячного оклада научного руководителя (МОНР) получена из открытых данных, размещенных на официальном сайте Национального исследовательского Томского политехнического университета. Величина месячного оклада инженеров (МОИ) берется как месячный оклад инженера кафедры.

Основной расчет фонда заработной платы выполняется по формуле:

$$ЗП_{\text{дн-т}} = \text{МО} / N, \quad (5.7)$$

где МО – месячный оклад, руб.;

N – количество рабочих дней в месяц, при шестидневной рабочей неделе – $N = 24,91$, а при пятидневной рабочей неделе – $N = 20,58$.

Среднедневная заработная плата научного руководителя равна:

$$ЗП_{\text{дн-т}} = \frac{26\,300}{24,91} = 1\,055,8 \frac{\text{руб.}}{\text{раб. день}}.$$

А среднедневная тарифная заработная плата инженеров равна

$$ЗП_{\text{дн-т}} = \frac{7\,864,11}{20,58} = 382,12 \frac{\text{руб.}}{\text{раб. день}}.$$

Затраты времени по каждому исполнителю в рабочих днях взяты из таблицы 5.2. Для перехода от тарифной суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку необходимо будет тарифную сумму заработка исполнителя, связанной с участием в проекте умножить на интегральный коэффициент. Интегральный коэффициент находится по формуле:

$$K_{\text{и}} = K_{\text{пр}} \cdot K_{\text{доп.ЗП}} \cdot K_{\text{р}}, \quad (5.8)$$

где $K_{\text{пр}}$ – коэффициент премий, $K_{\text{пр}} = 1,1$;

$K_{\text{доп.ЗП}}$ – коэффициент дополнительной зарплаты, при шестидневной рабочей неделе $K_{\text{доп.ЗП}} = 1,188$, а при пятидневной рабочей неделе $K_{\text{доп.ЗП}} = 1,113$;

$K_{\text{р}}$ – коэффициент районной надбавки, $K_{\text{р}} = 1,3$.

Результаты вычислений представлены в таблице 5.6.

Таблица 5.6 – Затраты на заработную плату

Исполнитель	Оклад, руб./месэ	ЗП _{дн-т} , руб./раб.день	Затраты времени, раб.дни	Коэффици- циент	Фонд з/платы, руб.
НР	26 300	1 055,8	14,85	1,699	26637,99237
И	7 864,11	382,12	105,6	1,59	64159,47648
Итого:					90797,46885

5.3.3 Отчисления на социальные нужды

Отчисления на социальные нужды установлены в размере 30,2% от заработной платы. Размер отчислений рассчитываются по формуле:

$$C_{\text{соц}} = C_{\text{ЗП}} \cdot 0,302, \quad (5.9)$$

где $C_{\text{ЗП}}$ – размер заработной платы.

Подставив необходимые значения в формулу 5.10 получим:

$$C_{\text{соц}} = 90797,46885 \cdot 0,302 = 27420,83 \text{ руб.}$$

5.3.4 Расчет затрат на электроэнергию

Затраты на электроэнергию рассчитываются по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot t_{\text{об}} \cdot Ц_{\text{э}}, \quad (5.10)$$

где $P_{об}$ – мощность, потребляемая оборудованием, кВт;

$t_{об}$ – время работы оборудования, час;

$\Pi_э$ – тариф на 1 кВт·час. Для ТПУ $\Pi_э = 5,8$ руб./кВт · час.

Время работы оборудования вычисляется на основе итоговых данных таблицы 5.3 для инженера ($T_{рд}$) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{об} = T_{рд} \cdot K_t, \quad (5.11)$$

где K_t – коэффициент использования оборудования по времени, $K_t = 0,9$.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном} \cdot K_C, \quad (5.12)$$

где K_C – коэффициент загрузки;

$P_{ном}$ – номинальная мощность оборудования, кВт. Для технологического оборудования малой мощности $K_C = 1$.

Таблица 5.7 – Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $\Sigma_{об}$, руб.
ПК инженера	760,32	0,09	442,34
Итого:			442,34

5.3.5 Расчет амортизационных расходов

Для расчета амортизационных расходов используется формула:

$$C_{ам} = \frac{H_A \cdot \Pi_{об} \cdot t_{рф} \cdot n}{F_D}, \quad (5.13)$$

где H_A – годовая норма амортизации единицы оборудования;

$\Pi_{об}$ – балансовая стоимость единицы оборудования с учетом ТЗР, стоимость ПК инженера – 59 140 руб.;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта,

$t_{рф} = 105 \cdot 8 = 845$ часов;

n – число задействованных однотипных единиц оборудования;

F_d – действительный годовой фонд времени работы соответствующего оборудования, $F_d = 298 \cdot 8 = 2384$ часа.

N_A определяется по формуле:

$$N_A = \frac{1}{CA}, \quad (5.12)$$

где CA – срок амортизации, который можно получить из постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы» Для электронно-вычислительной техники CA свыше 2 лет до 3 лет включительно. В данной работе примем $CA=2,5$ года. Тогда

$$N_A = \frac{1}{2,5} = 0,4.$$

Таким образом,

$$C_{AM}(ПК) = \frac{0,4 \cdot 49\,140 \cdot 845 \cdot 1}{2384} = 8384,78 \text{ руб}$$

Итого начислено амортизации 8384,78 руб.

5.3.6 Расчет расходов на услуги связи

Расходы на услуги связи определены наличием подключения к сети Интернет на компьютере, использованном в данной работе.

Ежемесячная оплата, согласно тарифу, составляет 350 рублей. В соответствии с таблицей 5.2, трудоемкость выполняемой задачи составляет четыре календарных месяца. Таким образом, сумма расходов на услуги связи составляет $5 \cdot 350 = 1750$ руб. Общая сумма расходов $C_{св} = 1750$ руб.

5.3.7 Расчет прочих расходов

Прочие расходы следует принять равными 10% от суммы всех предыдущих расходов. Они находятся по формуле:

$$C_{проч} = (C_{мат} + C_{ЗП} + C_{соц} + C_{эл.об.} + C_{AM} + C_{св}) \cdot 0,1, \quad (5.13)$$

Где $C_{мат}$ – расходы на материалы, руб.;

$C_{ЗП}$ – основная заработная плата, руб.;

$C_{соц}$ – расходы на единый социальный налог, руб.;

$C_{эл.об.}$ – расходы на электроэнергию, руб.;

C_{AM} – амортизационные расходы, руб.;

$C_{св}$ – расходы на услуги связи, руб.

Подставив полученные выше результаты, получим:

$$C_{\text{проч}} = (630 + 90797,46885 + 27420,83 + 442,34 + 8384,78 + 1\,750) \cdot 0,1 = 12942,5419 \text{ руб.}$$

5.3.8 Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта.

Таблица 5.8 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	630
Основная заработная плата	$C_{\text{ЗП}}$	90797,46885
Отчисления в социальные фонды	$C_{\text{соц}}$	27420,83
Расходы на электроэнергию	$C_{\text{эл.об.}}$	442,34
Амортизационные отчисления	C_{AM}	8384,78
Расходы на услуги связи	$C_{св}$	1 750
Прочие расходы	$C_{\text{проч}}$	12942,5419
Итого:		142367,961

Таким образом, затраты на разработку составили $C = 142367,961$ руб.

5.3.9 Расчет прибыли

Результатом работы является приложение осуществляющее оценку трудозатрат и длительности проектов.

Предположим, что в компании есть четыре уровня зарплат, каждый вышестоящий уровень отличается от нижестоящего в два раза.

Предположим что при каждой оценке трудозатрат будут принимать участие по пять человек из каждого уровня зарплаты. На каждую оценку традиционным образом, затрачивается час рабочего времени, в то время как на оценку с помощью разработанного программного средства затрачивается половина часа. В месяц происходит около трех групповых оценок трудозатрат проектов, таким образом, при участии двадцати человек, компания экономит около тридцати часов рабочего времени программистов. Средний уровень часа работы разработчика составляет около одной тысячи рублей. Таким образом, общая прибыли за год будет равна $30000 \cdot 12 - 142367,961 = 217632,039$.

5.3.10 Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли.

В нашем случае это $(360\,000) \cdot 0,18 = 64\,800$ руб.

5.3.11 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае

$$C_{\text{НИР(КР)}} = 142367,961 + 217632,039 + 64\,800 = 567\,167,961 \text{ руб.}$$

5.4 Оценка экономической эффективности проекта

В рамках данной работы оценить экономическую эффективность проекта невозможно.

5.4.1 Определение срока окупаемости

Срок окупаемости используется, как показатель эффективности проекта. Чем меньше срок окупаемости, тем эффективнее проект. Для расчета используется формула:

$$PP = \frac{C}{PP_{\text{ч}}}, \quad (5.14)$$

где C – затраты на разработку, руб.;

$PP_{\text{ч}}$ – годовая чистая прибыль, руб.

Подставив полученные выше результаты, получим:

$$PP = \frac{142367,961}{217632,039} = 0,65 \text{ года.}$$

Исходя из низкого срока окупаемости, можно сделать вывод, что проект эффективен.

Следует отметить, что основное направление данной работы – это не получение коммерческой выгоды, а достижение социального экономического эффекта.

5.4.2 Оценка научно-технического уровня НИР

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для оценки научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР, используется метод балльных оценок. Балльная оценка заключается в том, что каждому фактору по принятой шкале присваивается определенное количество баллов. Обобщенную оценку проводят по сумме баллов по всем показателям. На ее основе делается вывод о целесообразности НИР.

Сущность метода заключается в том, что на основе оценок признаков работы определяется интегральный показатель (индекс) ее научно-технического уровня по формуле:

$$I_{\text{НТУ}} = \sum_{i=1}^3 R_i \cdot n_i, \quad (5.13)$$

где $I_{\text{НТУ}}$ – интегральный индекс научно-технического уровня;

R_i – весовой коэффициент i -го признака научно-технического эффекта;

n_i – количественная оценка i -го признака научно-технического эффекта, в баллах.

Частные оценки уровня n_i и их краткое обоснование даны в таблице 5.10.

Таблица 5.9 Оценки научно-технического уровня НИР

Значимость	Фактор НТУ	Уровень фактора	Выбранный балл	Обоснование выбранного балла
0,3	Уровень новизны	Внедряемая	6	Экономит время, сокращает число машин и ресурсов, которые используются компанией в целях достижения высокого качества продукта.
0,2	Теоретический уровень	Разработка программного продукта	3	Изучение области методов и средств разработки программных продуктов
0,5	Возможность реализации	В течение первых лет	8	Сложные операции и доработка функционала продукта

Интегральный показатель научно-технического уровня для данного проекта составляет:

$$I_{НТУ} = 0,3 \cdot 6 + 0,2 \cdot 3 + 0,5 \cdot 8 = 6,4.$$

По полученным данным можно сделать вывод, что проект имеет высокий уровень научно-технического эффекта.

6 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

В разделе социальная ответственность содержится информация о производственной и экологической безопасности при выполнении и оформлении данной магистерской работы.

Выполнение в рамках магистерской диссертации проекта заключалось в анализе методов и средств автоматизированного тестирования и разработкой тестовых наборов для программного обеспечения. В соответствии с этим, производилось взаимодействие студентов-дипломников с ПК.

Целью раздела является выявление и анализ вредных и опасных факторов труда пользователя персонального компьютера (ПК) и разработка мер защиты от них, оценка условий труда и микроклимата рабочей среды. Также рассматриваются вопросы создания оптимальных условий труда, техники безопасности, пожарной профилактики и охраны окружающей среды.

6.1 Производственная безопасность на стадии разработки тестовых наборов

Научно-исследовательская деятельность выполнялась в помещении организации ООО «СибирьСофтПроект», которое оснащено видео-дисплейными терминалами (ВДТ), персональными электронно-вычислительными машинами (ПЭВМ), компьютерными столами, стульями, огнетушителями, кондиционером, противопожарной сигнализацией и датчиками дыма.

Производственный фактор считается вредным, если его воздействие может привести к заболеванию человека, и считается опасным, если его воздействие может привести к травме [25].

Производственные факторы классифицируются по группам элементов: физические, химические, биологические и психофизические. Для данной работы целесообразно рассмотреть физические и психофизические вредные и

опасные факторы производства, характерные для рабочей зоны программиста и пользователя. Выявленные факторы представлены в таблице 6.1.

Таблица 6.1 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
1) Работа за ПК	1) Отклонение показателей микроклимата; 2) Недостаточная освещенность рабочей зоны.	1) Опасность возникновения пожара; 2) Опасность поражения электрическим током.	1) СН 2.2.4/2.1.8.562-96; 2) СанПиН 2.2.4.548-96; 3) СанПиН 2.2.2/2.4.1340-03; 4) СП 52.13330.2011; 5) ГОСТ Р 12.1.019-2009 ССБТ; 6) СНиП 21-01-97.

6.2 Вредные производственные факторы

1) Отклонение показателей микроклимата

Микроклиматом производственных помещений является сочетание температуры, подвижности воздуха, влажности, температуры окружающих поверхностей и их тепловым излучением. Микроклимат определяет теплообмен организма человека и оказывает существенное влияние на функциональное состояние системы самочувствие, организма, работоспособность и здоровье.

Метеорологические условия для рабочей зоны производственных помещений регламентируются ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны» и СанПиН 2.2.4.548—96 «Гигиенические требования к микроклимату производственных помещений».

Принципиальное значение в нормах имеет раздельное нормирование каждого компонента микроклимата: температуры, влажности, скорости движения воздуха. В рабочей зоне должны обеспечиваться параметры микроклимата, соответствующие оптимальным и допустимым значениям. В таблице 6.1 [24] приведены параметры микроклимата. В нашем случае работа

была легкой и проводилась в теплое время года. Температура колебалась в пределах 21-27 градусов, относительная влажность в пределах 45-58. Следовательно, рабочая зона имела допустимые нормы.

Таблица 6.2 – Оптимальные значения характеристик микроклимата

Период года	Температура воздуха, °С		Температура поверхностей, °С		Относительная влажность, %	Скорость движения воздуха, м/с
	оптимальная	допустимая	оптимальная	допустимая		
Холодный	22-24	20-22	21-25	19-23	40-60	0,1
Теплый	23-25	20-28	22-26	19-27	40-60	0,1

2) Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни которого регламентируются СП 52.13330.2011.

Причины недостаточной освещенности: недостаточность искусственного и естественного освещения, пониженная контрастность.

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени. Недостаточность освещения увеличивает количество допускаемых ошибок и утомляемость, снижает производительность труда и приводит к появлению болезней зрения.

Разряд зрительных работ программиста и оператора ПЭВМ относится к разряду III и подразряду Г (работы высокой точности). В таблице 6.3 показаны нормативы искусственного освещения при работах заданной точности.

Таблица 6.3 – Требования к освещению помещений промышленных предприятий для операторов ПЭВМ [26]

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта зрительного, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Искусственное освещение		
						Освещённость, лк		При системе общего освещения
						При системе комбинированного освещения		
						всего	В том числе от общего	
Высокой точности	0,264	III		Средний, большой	Светлый, средний	400	200	200

При создании и поддержании благоприятных условий освещения рабочие места для операторов ПЭВМ должны соответствовать санитарно-эпидемиологическим правилам СанПиН 2.2.2/2.4.1340-03. Что бы рассеять естественное освещение, нужно на окнах рабочих помещений использовать жалюзи. Источником искусственного освещения должны быть лампы накаливания – для местного освещения и люминесцентные лампы. [27].

6.3 Опасные производственные факторы

1) Опасность возникновения пожара

Возникновение пожара – опасный производственный фактор, т.к. пожар наносит большой материальный ущерб на предприятии, и часто сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНиП 21-01-97.

В помещении с ПЭВМ повышен риск возникновения пожара, из-за множества факторов: присутствие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Виды источников воспламенения:

- искры при разряде статического электричества;
- искра от электрооборудования;
- искра от удара и трения;
- открытое пламя [23].

Для профилактики организации действий при пожаре нужно проводить комплекс организационных мер: регулярно обеспечивать проверки пожарной сигнализации, первичных средств пожаротушения; проводить инструктаж и тренировки по действиям в случае пожара; не загромождать или блокировать пожарные выходы; выполнять правила техники безопасности и технической эксплуатации электроустановок; во всех служебных помещениях устанавливать

«Планы эвакуации людей при пожаре и других ЧС», которые регламентируют действия персонала при возникновении пожара.

Что бы предотвратить пожар, помещение с ПЭВМ нужно оборудовать первичными средствами пожаротушения: углекислотными огнетушителями типа ОУ-2 или ОУ-5; пожарной сигнализацией, и, при необходимости, автоматической установкой объемного газового пожаротушения [21].

2) Опасность поражения электрическим током

Поражение электрическим током - опасный производственный фактор и, так как программист имеет дело с электрооборудованием, то электробезопасности на его рабочем месте необходимо уделять особое внимание. Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасностью является система организационных и технических мероприятий и средств, которая обеспечивает защиту людей от вредных и опасных воздействий электрической дуги, электрического тока, электромагнитного поля и статического электричества.

При поражении электрическим током опасно еще и тем, что человек не имеет возможности без специальных приборов обнаружить напряжение дистанционно.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

Обучение и инструктаж безопасным методам труда, проверка знаний правил безопасности и инструкций в соответствии с занимаемой должностью применительно к выполняемой работе являются основными организационными мероприятиями по обеспечению безопасности.

К мероприятиям по предотвращению возможности поражения электрическим током можно отнести:

- с целью защиты от поражения электрическим током, который возникает между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
- при включенном сетевом напряжении работы на задней панели корпуса приборов должны быть запрещены;
- все работы по устранению неисправностей должен производить квалифицированный персонал;
- необходимо постоянно следить за исправностью электропроводки [21, 22].

6.4 Экологическая безопасность

1) Влияние на окружающую среду

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта, а также самого продукта в результате его реализации на производстве.

В ходе выполнения ВКР и дальнейшем использовании алгоритмов отсутствуют выбросы каких-либо вредных веществ в атмосферу и гидросферу, следовательно, загрязнение воздуха и воды не происходит.

Так же и люминесцентные лампы, которые применяются для искусственного освещения рабочих мест, требуют особой утилизации, т.к. в них присутствует от 10 до 70 мг ртути, относящейся к чрезвычайно-опасным химическим веществам и которая может стать причиной отравления живых организмов, а также загрязнения атмосферы, гидросферы и литосферы. Срок работы таких ламп около 5-ти лет, после чего их нужно сдавать на переработку в специальные пункты приема.

Во время разработки и написания ВКР образовывался мусор, такой как: канцелярские принадлежности, бумажные отходы, неисправные комплектующие персонального компьютера, люминесцентные лампы.

2) Мероприятия по защите окружающей среды

Для уменьшения вредного влияния на литосферу необходимо производить сортировку отходов и обращаться в службы по утилизации для дальнейшей переработки или захоронения.

В основном, организации, занимающиеся приёмом и утилизацией ртути содержащих отходов, принимают люминесцентные лампы в массовых количествах. Лампа состоит из электронного блока — выгодный компонент для реставрации и утилизации; колба и цоколь также ценное сырьё. По стране утилизацией «ртутных» ламп занимаются более 50 фирм, но единственное их условие — деньги, которые вы должны заплатить за вывоз.

Такие лампы нельзя выкидывать в мусоропровод или уличные контейнеры, а нужно отнести в свой районный ДЕЗ (Дирекция единичного заказчика) или РЭУ (Ремонтно-эксплуатационное управление), где есть специальные контейнеры. Там они принимаются бесплатно, основанием должна служить утилизация в соответствии с Управлением Федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека по Томской области. Пункты приёма отработавших свой срок люминесцентных ламп по городам можно найти в интернете. [26]

Переработка макулатуры представляет собой многоэтапный процесс, цель которого заключается в восстановлении бумажного волокна и, зачастую, других компонентов бумаги (таких как минеральные наполнители) и использование их в качестве сырья для производства новой бумаги.

Организации, занимающиеся покупкой сломанных компьютеров на запчасти, готовы платить за запчасти деньги, которые они сэкономят на покупке новых деталей, необходимых для ремонта. Компьютерная техника (или ее компоненты) может также заинтересовать тех, кто скупает старые платы и радиодетали для получения из них после переработки драгоценных и

редких металлов. Многие сетевые гипермаркеты электронной техники периодически устраивают программу утилизации. Также можно самостоятельно отвезти сломанный компьютер в пункт приема металлолома не составит труда. Такие точки приема есть в каждом городе.

Также следует отменить, что разрабатываемая в рамках ВКР система позволяет создавать сложные по структуре документы технических задания для компаний, разрабатывающих информационные системы, вести документооборот и переписку между участниками непосредственно в самой системе в электронном формате, что экономит бумагу.

6.5 Безопасность в чрезвычайных ситуациях

6.5.1 Основные чрезвычайные ситуации в офисном помещении

Чрезвычайные ситуации бывают техногенного, природного, биологического, социального или экологического характера.

При работе в кабинете могут возникнуть следующие классификации чрезвычайных ситуаций:

- преднамеренные/непреднамеренные;
- техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения/природные – связанные с проявлением стихийных сил природы;
- экологические – это аномальные изменения состояния природной среды, такие как загрязнения биосферы, разрушение озонового слоя, кислотные дожди/ антропогенные – являются следствием ошибочных действий людей;
- биологические – различные эпидемии, эпизоотии, эпифитотии;
- социальные – это обстановка на определенной территории, сложившаяся в результате опасного социального явления, которое повлекло в результате человеческие жертвы, ущерб здоровью, имуществу или окружающей среды;
- комбинированные.

6.5.2 Типичные чрезвычайные ситуации

1) Пожар

Наиболее вероятной чрезвычайной ситуацией, которая возникает при работе с ПЭВМ, является пожар, так как в современных ЭВМ очень высокая плотность размещения элементов электронных схем. Соединительные провода и кабели располагаются в большой близости друг от друга, и при протекании электрического тока по ним выделяется большое количество теплоты, поэтому возможно оплавление изоляции и возникновение возгорания.

Биологические, так как программист работает в кабинете и контактирует с большим количеством людей, в том числе с другими сотрудниками, то велик риск заражения одного сотрудника от другого (чем больше людей, тем выше риск). В связи с большим скоплением народа в одном помещении появляется необходимость в непрерывном проветривании, что приводит к образованию сквозняков, что так же может сказаться на здоровье.

Возникновение других видов ЧС – маловероятно [26].

2) Кибертерроризм

Терроризм – это метод, посредством которого организованная группа или партия стремятся достичь провозглашенные ими цели через систематическое использование насилия.

Компьютерный терроризм (кибертерроризм) — использование компьютерных и телекоммуникационных технологий (прежде всего, интернета) в террористических целях.

В киберпространстве используются различные способы для совершения кибертерракта:

- несанкционированный доступ к государственным и военным секретам, банковской и личной информации;
- причинение ущерба отдельным физическим элементам информационного пространства, например, разрушение сетей электропитания, создание помех;

- использование специальных программ для разрушения аппаратных средств;
- кражи или уничтожения информации, программ и технических ресурсов при преодолении защитных систем, внедрения вирусов, программных закладок;
- воздействие на программное обеспечение и информацию;
- раскрытие публикации закрытой информации и угроза ее использования;
- захват каналов СМИ для распространения дезинформации, слухов и демонстрации силы террористической организации и объявления своих требований;
- уничтожение или активное подавление линий связи, неправильная адресация, перегрузка узлов коммуникации;
- проведение информационно-психологических операций.

Использование документов и данных хранящихся в системы, включая личные переписки участников, дают киберпреступникам возможность использовать их в злых умыслах.

В результате чего, физические или юридические лица, чьи данные были использованы в злых умыслах, могут быть подвержены шантажу и вымогательству со стороны киберпреступников, что может негативно сказаться на репутации и финансовому состоянию компаний разрабатывающие информационные системы. Также в создаваемых документах технических заданий могут содержаться информации о государственной или коммерческой тайне.

6.5.3 Действия в результате возникновения чрезвычайной ситуации и мер по ликвидации ее последствий

При работе компьютерной техники выделяется много тепла, это может приводить к пожароопасной ситуации. Серьёзную опасность представляют

электроизоляционные материалы, которые используются для защиты от механических воздействий отдельных радиодеталей.

В связи с этим, участки, на которых используется компьютерная техника, по пожарной опасности относятся к категории пожароопасных «В».

Меры, соблюдение которых поможет исключить с большой вероятностью возможность возникновения пожара:

- что бы понизить воспламеняемость и способность распространять пламя кабели покрывают огнезащитным покрытием;
- при ремонтно-профилактических работах строго соблюдаются правила пожарной безопасности;
- помещения, в которых должны располагаться ПЭВМ проектируют I или II степени огнестойкости;
- помещения, в которых эксплуатируются устройства ПЭВМ, должно оборудоваться первичными средствами пожаротушения и быть обеспечено инструкциями по их применению. Для пожаротушения можно использовать углекислотные огнетушители типа ОУ-2, ОУ-5(описание ниже), а также порошковый тип. Пенные огнетушители не допускается применять, так как жидкость пропускает ток;
- устройства ПЭВМ нужно устанавливать в стороне от отопительных и нагревательных приборов (на расстоянии не менее 1 м и в там, где не затруднена их вентиляция и отсутствуют прямые солнечные лучи);
- разрабатываются организационные меры по обучению персонала навыкам ликвидации пожара имеющимися в наличии средствами тушения пожара до прибытия пожарного подразделения [22].

При пожаре людям необходимо покинуть помещение в течение минимального времени.

В помещениях с компьютерной техникой, нельзя применять воду и пену так как существует опасность повреждения или полного выхода из строя электронного оборудования.

При тушении пожаров необходимо применение углекислотных и порошковых огнетушителей, обладающих высокой скоростью тушения, большим временем действия, возможностью тушения электроустановок, высокой эффективностью борьбы с огнем. Воду разрешено применять только во вспомогательных помещениях [21].

6.6 Правовые и организационные вопросы обеспечения безопасности

1) Описание правовых норм для работ, связанных с работой за ПЭВМ

Регулировка отношений работник – работодатель, которые касаются оплаты труда, трудового распорядка, регулировка труда женщин, детей, людей с ограниченными способностями и т.д., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Продолжительность рабочего времени не должна превышать 40 часов в неделю.

Порядок исчисления нормы рабочего времени на определенные календарные периоды (месяц, квартал, год) в зависимости от установленной продолжительности рабочего времени в неделю определяется федеральным органом исполнительной власти, осуществляющим функции по выработке государственной политики и нормативно-правовому регулированию в сфере труда.

Продолжительность ежедневной работы (смены) не может превышать:

- для работников от 15 до 16 лет – 5 часов, от 16 до 18 лет – 7 часов;
- для учащихся общеобразовательных учреждений, образовательных учреждений начального и среднего профессионального образования, совмещающих в течение учебного года учебу с работой, от 14 до 16 лет – 2,5 часа, от 16 до 18 лет – 4 часов;
- для инвалидов – в соответствии с медицинским заключением, выданным в порядке, установленном федеральными законами и иными нормативными правовыми актами российской федерации.

Для работников, занятых на работах с вредными и (или) опасными условиями труда, где установлена сокращенная продолжительность рабочего времени, максимально допустимая продолжительность ежедневной работы (смены) не может превышать:

- при 36-часовой рабочей неделе - 8 часов;
- при 30-часовой рабочей неделе и менее - 6 часов.

Продолжительность работы (смены) в ночное время сокращается на один час без последующей отработки. К работе в ночное время не допускаются: беременные женщины; работники, не достигшие возраста 18 лет, за исключением лиц, участвующих в создании и (или) исполнении художественных произведений, и других категорий работников в соответствии с настоящим Кодексом и иными федеральными законами.

В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для питания и отдыха. Это время и его продолжительность должно устанавливаться правилами внутреннего трудового распорядка или по соглашению между работником и работодателем.

Работникам должны предоставляться выходные дни (еженедельный непрерывный отдых).

Организация-работодатель должна выплачивать заработную плату работникам. Удержание заработной платы возможно только в случаях, установленных ТК РФ ст. 137. При задержке заработной платы более чем на 15 дней, может приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [23].

Основными элементами рабочего места программиста являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура, мышь; вспомогательными - пюпитр, подставка для ног.

Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования [21].

На рабочем месте, оснащенном ЭВМ, должно быть пространство для пользователя не менее 850 м. Для стоп должно быть предусмотрено пространство по глубине и высоте не меньше 150 мм, по ширине – не менее 530 мм. Расположение ЭВМ на рабочем месте должно быть таким, чтобы поверхность экрана находилась на расстоянии 400 – 700 мм от глаз пользователя. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы [26].

Высота поверхности сиденья должна регулироваться в пределах 400 – 550 мм. Ширина и глубина его поверхности должна быть не менее 400 мм. Поверхность сиденья должна быть плоской, передний край – закругленным. Сиденье и спинка кресла должны быть полумягкими, с несскользящим, не электризующимся и воздухопроницаемым покрытием, материал которого обеспечивает возможность легкой очистки от загрязнения.

Опорная поверхность спинки стула должна иметь высоту 280 – 320 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм. Расстояние сцинки от переднего края сиденья должно регулироваться в пределах 260 – 400 мм.

Рациональной рабочей позой может считаться такое расположение тела, при котором ступни работника расположены на плоскости пола или на подставке для ног, бедра сориентированы в горизонтальной плоскости, верхние части рук – вертикальный угол локтевого сустава колеблется в пределах 70 – 90, запястья согнуты под углом не более чем 20, наклон головы – в пределах 15 – 20, а также исключены частые ее повороты [26].

2) Влияние разработанного приложения на работу компании

Основным направлением реализации программного продукта для оценки трудозатрат и длительности проектов, является уменьшение временных издержек предприятия, а так же возможность привлекать к оценке трудозатрат работников отсутствующих в офисе.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был проведен анализ существующих методов оценки длительности и трудозатрат проектов по разработке программного обеспечения. Проведен сравнительный анализ аналогов разрабатываемого приложения, проведен анализ возможности нативной и кроссплатформенной разработки. Проведен анализ языков программирования и анализ шаблонов архитектуры мобильных приложений. Так же было разработано приложение для оценки трудозатрат и длительности проектов.

Для разработки приложения для оценки трудозатрат и длительности проектов по разработке программного обеспечения был изучен метод оценки трудозатрат – Planning poker, метод оценки длительности проектов – PERT. Были изучены основные функции предоставляемые Jira API, были изучены и использованы возможности базы данных Firebase. Был изучен язык программирования Swift и разработано мобильное приложение использующее архитектурный шаблон VIPER, позволяющее проводить оценку трудозатрат и длительности проектов. Приложение позволяет сотрудникам голосовать по необходимым задачам не присутствуя на рабочем месте, таким образом результаты голосования являются наиболее точными и адекватными.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Planning Poker – Leanify Ltd [Электронный ресурс]. - режим доступа: URL: <http://leanify.com/planning-poker/> (дата обращения – 15.03.2017)
2. iPERT [Электронный ресурс]. - режим доступа: URL: <https://itunes.apple.com/ru/app/ipert/id465224517?mt=8> (дата обращения – 15.03.2017)
3. Scrum Poker Cards – artArmin [Электронный ресурс]. - режим доступа: URL: <https://itunes.apple.com/ru/app/scrum-poker-cards-agile/id474276586?mt=8> (дата обращения – 15.03.2017)
4. Planning Poker – Cesar [Электронный ресурс]. - режим доступа: URL: <https://itunes.apple.com/tr/app/planning-poker-cesar/id1032755956?l=tr> (дата обращения – 15.03.2017)
5. Scrum Time – Planning Poker – Endava [Электронный ресурс]. - режим доступа: URL: <https://itunes.apple.com/us/app/scrum-time-planning-poker/id844162336?mt=8> (дата обращения – 15.03.2017)
6. PERT – ILMASOFT PRO [Электронный ресурс]. - режим доступа: URL: <https://play.google.com/store/apps/details?id=com.professional.pert> (дата обращения – 15.03.2017)
7. Метод PERT [Электронный ресурс]. - режим доступа: URL: Form.ru/метод-pert/ (дата обращения – 20.03.2017)
8. АЛЬТЕРНАТИВНЫЙ МЕТОД GERT [Электронный ресурс]. - режим доступа: URL: http://studopedia.ru/2_91348_alternativniy-metod-GERT.html (дата обращения – 23.03.2017)
9. Методы статистического моделирования (метод Монте-Карло) [Электронный ресурс]. - режим доступа: URL: <http://csc.sibsutis.ru/sites/csc.sibsutis.ru/files/courses/pvt/%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5.pdf> (дата обращения – 25.03.2017)

10. PLANNING POKER — ИНСТРУМЕНТ СТРАТЕГИЧЕСКОГО ПЛАНИРОВАНИЯ [Электронный ресурс]. - режим доступа: URL: <http://plushki-interneta.ru/planning-poker-instrument-strategicheskogo-planirovaniya/> (дата обращения – 27.03.2017)
11. Swift. Язык программирования с открытым кодом [Электронный ресурс]. - режим доступа: URL: <https://www.apple.com/ru/swift/> (дата обращения – 15.04.2017)
12. Введение в Obj-C [Электронный ресурс]. - режим доступа: URL: <http://macbug.ru/cocoa/objc> (дата обращения – 17.04.2017)
13. React Native Tutorial: Build app with JavaScript [Электронный ресурс]. - режим доступа: URL: <https://www.raywenderlich.com/126063/react-native-tutorial> (дата обращения – 15.04.2017)
14. Шаблон проектирования MVC [Электронный ресурс]. - режим доступа: URL: <https://webformyself.com/shablon-proektirovaniya-mvc/> (дата обращения – 17.04.2017)
15. Model-View-Controller (MVC) in IOS: A Modern Approach [Электронный ресурс]. - режим доступа: URL: <https://www.raywenderlich.com/132662/mvc-in-ios-a-modern-approach> (дата обращения – 17.04.2017)
16. IOS Architecture Patterns [Электронный ресурс]. - режим доступа: URL: <https://techblog.badoo.com/blog/2016/03/21/ios-architecture-patterns/> (дата обращения – 17.04.2017)
17. Реализация MVVM в IOS с помощью RxSwift [Электронный ресурс]. - режим доступа: URL: <https://habrahabr.ru/post/273455/> (дата обращения – 13.04.2017)
18. The Book of VIPER [Электронный ресурс]. - режим доступа: URL: <https://github.com/strongself/The-Book-of-VIPER> (дата обращения – 5.04.2017)
19. PlanItPoker [Электронный ресурс]. - режим доступа: URL: <http://www.planitpoker.com/> (дата обращения – 17.04.2017)

20. PlanningPoker.com [Электронный ресурс]. - режим доступа: URL: <http://www.planningpoker.com> (дата обращения – 17.04.2017)
21. Чрезвычайные ситуации при работе с ПЭВМ // Студопедия — Ваша школопедия. URL: http://studopedia.ru/8_107307_osveshchenie-pomeshcheniy-vichislitelnih-tsentrov.html (дата обращения: 10.03.2017).
22. Долин П.А. Справочник по технике безопасности. М.: Энергоатомиздат, 1984 г. – 824 с.
23. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 3.07.2016) // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901807664> (дата обращения: 11.03.2017).
24. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
25. Охрана труда. Основы безопасности жизнедеятельности // www.Grandars.ru. URL: <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/ohrana-truda.html> (дата обращения: 11.03.2017).
26. Попов В.М. Психология безопасности профессиональной деятельности: учебное пособие / В. М. Попов; Новосибирский государственный технический университет. – Новосибирск: Изд-во Новосибирского государственного технического университета, 1996 г. – 155 с.
27. СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормы. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901865498> (дата обращения: 10.03.2017).

ПРИЛОЖЕНИЕ А

Part 1

Analytical review

Студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Алешков Андрей Викторович		

Консультант кафедры Информационных систем и технологий:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Мирошниченко Евгений Александрович	К.т.н.		

Консультант – лингвист кафедры ИЯ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель каф.ИЯ	Горбатова Татьяна Николаевна			

2.1 Choice of projects estimation method

During detailed research of estimation methods, were pointed out the main features of each one of them.

PERT method allows us to give 3 types of estimations: pessimistic, most probable and optimistic. This method allows to define probabilities of finishing project within specified periods of time and to specified time line. It also allows us to calculate standard deviation of duration and, respectively, maximum and minimum expected duration on the level of separate work and the whole project.

GERT method is applied in cases where subsequent tasks can start after ending a number of previous tasks. Method of graphical estimation and program review (GERT) allows to consider risk of work content change when specific events occur or by the results of previous work. Thus, we can consider not only risks (uncertainty) on the level of separate works, but also on the level of the whole project. GERT method simulation result is several diagrams, reflecting the probability of different duration and uncertain project work content.

Monte Carlo method is based on obtaining large number of realization of random process, which is formed in a way to have its probable characteristics to coincide with related characteristics of active task. Monte Carlo method has several advantages:

- calculations, that are needed for Monte Carlo analysis, are quite simple, even inserting complicated equations or correlation modeling and other interdependencies does not create additional difficulties;
- ease-of-use, as there are good software packages, automating MCA, then there is computer which is able to perform all the work to generate resultant distribution;
- model modification, analyzed by Monte Carlo method is fast process, allowing to compare current model with previous one and the study of each model behavior is not complicated.

Monte Carlo method has such disadvantages as complexity, as it's based on probability concept, which is yet unknown to many project managers. Approximation of output distributions of probability in case of small number of iterations, however this problem is easily solved by increasing the number of iterations until the required level of accuracy is reached.

Planning Poker method allows us to give fast estimation to one or another task, as the number of votation is limited and voters come to necessary decision quickly.

Poker planning stimulates the team to discuss the task elements to reach agreement on the work amount estimation on them. It should be noted, that experts of the team are necessarily included in discussion.

As the result of the research, Planning Poker method was chosen, because it gives possibility of fast task estimation by the whole team while its realization. This method minimizes the binding effect by enquiring each member of the team in the way that no one knows the result of votation till specified time.

2.2 Choice of functions of the product being developed

The product has following functions:

- opportunity to estimate working hours and project duration;
- display of unvalued tasks in Jira;
- display of members in Jira project;
- possibility of final estimation record in Jira;
- notifications mailout to members included in votation;
- results of duration estimations display to administrator;
- flexible voting setting;
- possibility of next scheduled task automatic launch for voting;
- providing administrator with graphical data of voting results;
- Firebase integration.

To define distinctive characteristics of the created software instrument from existing analogs, was made a table, which presents programs comparative characteristics to estimate labor contribution and projects duration, using previously chosen Planning Poker Method. As you can see from the table 3.1 WEB applications are also included to give opportunity to get acquainted with their functionality. Functionality is quite similar with each other, both in mobile and WEB applications. As a result, main distinctive feature of created software is integration possibility with JIRA client to unload unrated tasks and to record voting results.

Table 3.1 – Comparison of analog applications PlanningPoker

Name	Producer	OS support	Cost	Trial period	Necessity of preliminary registration	Integration with Jira, Redmine, TFS etc.	Options for estimation	Voters number restriction
PlanITpoker[19]	PlanITpoker	Windows, Mac	Free	-	-	-	+	Up to 50 people
PlanningPoker.com[20]	Mountain Goat Software	Windows, Mac	Pro = 49.95 \$ Enter = 0	1 month (Pro)	+	-	Pro + Enter -	Pro до 50 Enter < 5
Planning Poker	SSP	IOS	Free	-	-	Jira	-	Up to 30 people
Scrum Poker Cards (Agile)	artArmin	Android	Free	-	+	-	+	No information
Planning Poker	CESAR	Android, IOS	Free	-	-	-	+	No information
Scrum Time - Planning Poker	Endava	Android, IOS	Free	-	+	-	+	No information

Continuation of Table 3.1

Name	Producer	Flexible voting setting	Possibility of automatic consequent votation launch	Possibility to invite unregistered user	Providing graphical voting data	Possibility of message exchange during voting
PlanITpoker	PlanITpoker	+	-	+	+	+
PlanningPoker.com	Mountain Goat Software	Pro + Standard - Enter -	Pro + Standard - Enter -	Pro + Standard - Enter -	Pro + Standard - Enter -	Pro + Standard + Enter +
Planning Poker	SSP	+	+	-	+	-
Scrum Poker Cards (Agile)	artArmin	-	-	-	-	-
Planning Poker	CESAR	-	-	+	-	+
Scrum Time Planning Poker -	Endava	-	-	-	-	-

2.3 Choice of development type

While analyzing market of possibilities of mobile applications' development, was chosen native version of IOS development, as for application development it's necessary to choose strictly typed, compilable, object-oriented language. Java and Objective-C/ Swift fit for that perfectly. It means, that there is high probability that consequent updates for Android and IOS systems might limit or exclude ReactNative work. Furthermore, many experts claim that at present time, it's impossible to repeat some native application using ReactNative. Currently, ReactNative also does not have ability to create applications for Apple Watch and there are no prerequisites for this. All IOS and Android innovations will leave ReactNative behind, unlike the native development, when creators of operating system along with it help to create IDE with all kinds of innovations. It is necessary to pay attention to longevity of project. Apple and Google produce their own IDE for their OS, but Facebook does not issue its own OS. Thus, the project that is being developed with help of native development looks more stable and long-lasting.

Thus, the advantages of cross-platform development ReactNative are that ReactNative significantly simplifies development of 2 platforms for mobile devices at the same time, thereby reduces the working time. In basic structure, framework makes difference between 2 platforms, it is very important, because native development for IOS and Android is very different, what brings to the cost increase due to necessity to hire specialists knowing Swift / Objective-C and Java/ Kotlin. Therefore, all basic APIs are different from, for example GPS usage, animation creation of user's interface, even creation of phone calls is different. To enter the development process with the help of ReactNative there is no necessity to learn new languages if you are familiar with JavaScript, HTML, CSS, and programming reminds creating of WEB interface for WEB versions. The instruments and architecture of application are the same with WEB development.

It is necessary to know for cross-platform development, that typical mobile application has two main components – application and its server part. Regardless whether it is native or cross-platform application, data base will be one, at the same

time its engineering counts up to 50 % of all project work. As a result, the choice of development method of mobile application is not the main problem of the project.

Programmer, who can write native programs for Android and IOS, will spend much more time for initial writing for one of the application platforms than for the remaining platform. The speed will be reached due to the fact that it will take less time to write the code for already built architecture of the application, proper classes' abstraction, correct arrangement of user's interface components and code refactoring for any changes. Thus, while writing the first application the most essential factors of engineering and programming will be solved.

In ReactNative, apart from structural questions there is a need to solve problems on the platform compatibility level and also on the engineering level. In the end, it will be necessary to find solutions for much more serious obstacles than in native development.

Reusability of the code written before, using other programming languages (C/C++) or using native game platform - ReactNative can use it, but it will be necessary to write your own code, what makes it meaningless to reuse native projects. Writing of all these libraries again will require high level of knowledge of native development to have opportunity to recreate low level code.

The most well-known products written on ReactNative (Facebook, Instagram, AriBnB), were written by the experts of native development. Thus, for starting developers of mobile applications native development is the easiest to learn.

2.4 The choice of programming language

Having analyzed the advantages and disadvantages of Swift and Objective-C, Swift turned out to be more preferable.

Swift is more productive language, which allows to create new applications faster. In the table 3.2 the main differences of those two languages are presented.

Table 3.2 – Comparison of Swift и Objective-C programming languages

	Swift	Objective-C
Work with other programming languages	Syntax is similar to other famous programming languages, such as: Python, Go, JavaScript.	Syntax is much more different from other programming languages, so it is quite difficult to learn.
Code length	The code is more rational, what reduces its length and development time.	Realization of the same functions requires twice as much lines of the code than Swift.
Interactivity	Can create applications in interactive regime.	Can not create applications in interactive regime.
Community using the language	Swift is much easier to learn, but the number of specialists is currently much less than in Objective-C.	Most part of IOS applications industry is written using Objective-C.
Security	Security is one of the Swift main concepts. Its structure excludes several types of potential mistakes in Objective-C.	Not as secure as Swift.
Classes and subclasses	In Swift, class does not have subclasses.	In Objective-C class has subclasses.
Tuple	Tuples allow grouping several elements in one.	There are no tuples in Objective-C.
Opened source code	Swift – is a language with opened source code. It is also used in Linux, what potentially creates possibility to use it in developing programs for Android OS.	Does not have source code and is oriented only on Apple devices.
Completeness and language stability	Swift development has not been finished yet and Apple constantly releases and adds new options.	Objective-C has been created already and is being used consistently.
Description of methods in code	Can click on the name of the class to see its header and nested functions.	There are no such functions.

Using Swift simplifies development process, what makes it available for starting programmers to study, thereby reducing expenses for highly qualified professionals. Apple support of the language also plays an important role.

Fast growth and opened code can be named as main advantages of Swift. Swift is chosen because it is free and thus, available for larger amount of developers. Picture 3.1 demonstrates the growth diagram of developer's audience since the launch of Swift.



Picture 3.1 – Diagram of Swift's audience growth

Reducing the time of application writing allows to reduce expenses and to increase company's profit. Swift is suitable for writing applications in the shortest periods of time.

2.5 The choice of application architecture template

As a result of architectural template analysis, the following conclusions were made:

2.5.1 Traditional MVC is not applicable for modern IOS development.

2.5.2 Apple MVC is the best architectural template, in terms of development speed, but it significantly complicates application support with its growth.

2.5.3 MVP offers good testability, but along with significant code increase. Its distinctive feature is distribution – most part of responsibilities is divided between Presenter and Model, while View has minimum of tasks. Testability is also an advantage, you can test most part of domain logic due to the emptiness of View.

2.5.4 Distinctive features of MVVM are distribution – View in MVVM has more tasks, than View in MVP, because MVVM updates its state from ViewModel and second one only transfers all the events to Presenter without updating itself. Regarding testability – ViewModel knows nothing about View, what allows testing it without any obstacles. View also can be tested, but as it depends on UIKit - testing it is not advisable.

2.5.5 Viper surpasses all presented architectural patterns in distribution, testability and simplicity of new modules adoption. The main features of Viper are distribution – Viper has excellent ability for task distribution, testability – better distribution gives better testability. Also usability ease – small amount of code in Viper's blocks allows to easily switch from one task to another, thereby increasing the useful work of programmer.

So far, we can make a conclusion that for the best task distribution and coverage of application by the tests of application, Viper is the most suitable architectural template.

ПРИЛОЖЕНИЕ Б

Алгоритм работы приложения

