

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных систем и сетей
Кафедра Информатики

К защите допустить:

Заведующая кафедрой
Информатики

_____ Н. А. Волорова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту
на тему:

**СЕРВИС ДЛЯ ИЗВЛЕЧЕНИЯ ХАРАКТЕРИСТИК
МУЗЫКАЛЬНОЙ КОМПОЗИЦИИ С ПРИМЕНЕНИЕМ
МАШИННОГО ОБУЧЕНИЯ**

БГУИР ДП 1-40 04 01 01 097 ПЗ

Студент

Д. В. Шматков

Руководитель

П. С. Саттарова

Консультанты:

*от кафедры Информатики
по экономической части*

П. С. Саттарова

К. Р. Литвинович

Нормоконтролёр

Н. Н. Бабенко

Рецензент

Минск 2017

РЕФЕРАТ

Пояснительная записка 32 с., 0 рис., 5 табл., 30 формул, 4 источников.
ПРОГРАММНОЕ СРЕДСТВО, ВЕБ-СЕРВИС, УНИВЕРСИТЕТ, АНАЛИЗ
МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ, МАШИННОЕ ОБУЧЕНИЕ,
ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Цель настоящего дипломного проекта состоит в разработке программной системы, предназначенной для автоматизации извлечения характеристик музыкальных произведений.

В процессе анализа предметной области были выделены основные характеристики музыкальных композиций и способы их извлечения, которые в настоящее время практически не охвачены автоматизацией. Было проведено их исследование и моделирование. Кроме того, рассмотрены существующие средства, разрозненно разрабатываемые и применяемые сотрудниками университетов и отдельными компаниями (так называемые частичные аналоги). Выработаны функциональные и нефункциональные требования.

Была разработана архитектура программной системы, для каждой ее составной части было проведено разграничение реализуемых задач, проектирование, уточнение используемых технологий и собственно разработка. Были выбраны наиболее современные средства разработки, широко применяемые в индустрии.

Полученные в ходе технико-экономического обоснования результаты о прибыли для разработчика, пользователя, уровень рентабельности, а также экономический эффект доказывают целесообразность разработки проекта.

СОДЕРЖАНИЕ

Введение	7
0.1 Анализ предметной области	8
1 Обзор существующих аналогов	11
1.1 The Echo Nest	11
1.2 Niland	11
1.3 Music Technology Group	12
1.4 DeepMind	13
2 Требования к проектируемому программному средству	14
3 Проектирование программного средства	15
4 Используемые технологии	16
5 Руководство по установке и использованию	17
6 Техничко-экономическое обоснование разработки и внедрения программного средства	18
6.1 Характеристика программного средства	18
6.2 Определение объема и трудоемкости ПС	18
6.3 Расчет сметы затрат	21
6.4 Оценка экономической эффективности применения ПС у пользователя	25
Заключение	29
Список использованных источников	30
Приложение А Фрагменты исходного кода	31

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения и сокращения.

API – Application Programming Interface (программный интерфейс приложения) [1].

ВВЕДЕНИЕ

Подъем сервисов электронного распространения медиаинформации дал беспрецедентный доступ пользователям к записям музыкальных композиций. Сейчас такие сервисы, как iTunes, Google Music, Яндекс Музыка, Spotify и другие, предоставляют мгновенный доступ к миллионам записей. Перед пользователями возникает проблема выбора следующей композиции для прослушивания. Для решения этой проблемы, а так же для того, чтобы облегчить ориентацию в этом большом количестве информации, сервисы электронного распространения записей предоставляют пользователям системы рекомендации контента.

Современные системы рекомендации контента обычно используют статистику прослушивания композиций пользователями, чтобы сделать рекомендации более точными. Однако на такие системы часто оказывают влияние смещения популярности в больших масштабах, из-за чего системы не могут рекомендовать композиции, которые менее популярны. Так же это затрудняет создание индивидуальных рекомендаций для пользователей, исходя из их предпочтений. Поэтому возникает необходимость извлекать информацию из композиций, а так же классифицировать сами композиции. На сегодняшний день извлечение характеристик и классификация музыкальных композиций является интересной и соревновательной задачей, которой занимаются различные компании и исследователи. На данный момент сильное развитие получают способы, основанные на машинном обучении.

Целью данного дипломного проекта является создание сервис для извлечения музыкальных характеристик, который будет пригоден для дальнейшей интеграцией с другими системами. В ходе работы предстоит выполнить следующие задачи:

- а) Выбрать способ извлечения базовых характеристик музыкальных композиций (таких как спектр, ритм, продолжительность);
- б) Построить модель машинного обучения для извлечения высокоуровневых характеристик;
- в) Собрать и упорядочить данные для обучения модели;
- г) Построить эффективную архитектуру сервиса;
- д) Построить API для интеграции с другими системами.

Реализация сервиса для извлечения характеристик музыкальных композиций позволит упростить задачу классификации контента для сервисов электронного распространения записей, а так же создать систему для создания более точных и индивидуальных рекомендаций для пользователей.

0.1 Анализ предметной области

Извлечение музыкальной информации - небольшая, но интересная и быстрорастущая область исследований, которая охватывает большой диапазон продуктов, используемых по всему миру. Эта область охватывает несколько дисциплин научных знаний: музыковедение, психологию, обработку сигналов, машинное обучение и комбинации этих дисциплин. Задача, которую будет решать разрабатываемый сервис, относится к задачам извлечения музыкальной информации.

Несмотря на то, что извлечение музыкальной информации является еще небольшой областью исследований, для извлечения характеристик музыкальных приложений применяется достаточное количество методов. Эти методы различаются как на основании применяемых подходов, так и на основании уровней извлекаемых характеристик. На основании характеристик выделяют методы, которые извлекают низкоуровневые характеристики (спектр, ритм), и методы которые извлекают высокоуровневые характеристики (жанр, настроение). В основном, нас будут интересовать методы, которые извлекают высокоуровневые характеристики.

Для извлечения высокоуровневых характеристик выделяют два класса методов: основанные на математических функциях, в основе которых лежит машинное обучение. В музыке появляются новые направления, а старые могут изменяться, поэтому наиболее перспективными являются методы, которые способны подстраиваться под переменчивую природу музыки. Под такие условия в большей степени подходят методы, в основе которых лежит машинное обучение.

Машинное обучение - класс методов, характерной чертой которых является не прямое решение задачи, а решение, которое строится в процессе обучения при применении решения к множеству сходных задач. В последнее время этот класс методов получил активное развитие и поддержку со стороны крупных компаний и университетов.

Цифровой аудиосигнал можно представить в виде изображения звуковой волны.

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК ЗВУКОВОЙ ВОЛНЫ!!!

Цифровой аудиосигнал представляет собой множество точек, которые расставлены на одинаковом расстоянии друг от друга. Расстояние между точками называется частотой дискретизации. Чем меньше интервал (выше частота дискретизации), тем шире частотный диапазон, который можно закодировать таким образом.

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК ЗВУКОВОЙ ВОЛНЫ В БОЛЬШОМ

РАЗРЕШЕНИИ!!!

Амплитуда колебаний зависит от времени звука и коррелирует с громкостью звука. А частота колебаний напрямую связана с высотой звука. Для того, чтобы получить информацию о частоте колебаний, необходимо применить преобразование Фурье. Преобразование Фурье позволяет разложить периодическую функцию в сумму гармонических с разными частотами. Коэффициенты гармонических функций при сложении будут давать нам те частоты, которые мы хотим получить.

При применении преобразования Фурье ко всей звуковой дорожке мы получим "смазанный" во времени спектр. Для того, чтобы получить спектр, не теряя временной составляющей, необходимо применить к сигналу оконное преобразование Фурье. Оконное преобразование Фурье отличается от обычного тем, что мы делим наш сигнал на короткие отрезки (окна) и применяем к каждому преобразование Фурье. По-сути мы получаем набор спектров - отдельно для каждого отрезка. В результате мы получим картинку, которой можно описать звуковую дорожку.

КАРТИНКА СПЕКТРАЛЬНОЙ ХАРАКТЕРИСТИКИ!!!!

Жанр композиции является высокоуровневой характеристикой. Для определения жанра необходимо разобраться, как человек воспринимает звук.

Ухо человека устроено так: звуковые волны смещают барабанную перепонку при взаимодействии с ней. Вибрации передаются во внутреннее ухо и считываются им. Смещение барабанной перепонки зависит от звукового давления, при том зависимость является не линейной, а логарифмической. Для измерения громкости принято использовать относительную шкалу - уровень звукового давления (измеряется в децибелах). Так же воспринимаемая громкость зависит и от частоты звука. Для оценки громкости звука используется логарифмическая единица измерения - фон. В шкале фонов, в отличие от децибелов, значения громкости связаны с чувствительностью на разных частотах человеческого уха. Частота 1000 Гц является чистым тоном, и уровень фона для неё численно равен уровню в децибелах. Для остальных частот используют поправки, которые представляют собой стандартизированное семейство кривых, называемых изофонами (ISO 226).

РИСУНОК КОНТУР РАВНЫХ ГРОМКОСТЕЙ!!!!

Аналогично с громкостью, частота так же воспринимается человеческим ухом нелинейно. Для измерения воспринимаемой частоты человеческим ухом используется мел-шкала. Шкала основана на статистической обработке субъективного восприятия звука на больших данных. За 1000 мел взят звук с частотой 1000 Гц при уровне громкости 40 фон. За 0 мел взят звук частотой 20 Гц при уровне громкости 40 фон.

РИСУНОК С МЕЛ-ШКАЛОЙ!!!!

1 ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ

Существует ряд сервисов, которые решают задачу извлечения характеристик музыкальных композиций и достаточно близкие к этой задаче. Рассмотрим наиболее популярные и известные из них.

1.1 The Echo Nest

The Echo Nest - компания из Соммервила, которая занимается разработкой сервиса для анализа музыкальных композиций и составлении рекомендаций. Их продукт собирает информацию используя акустический и текстовый анализ. Текстовый анализ состоит в том, что любое упоминание музыкальной композиции, которое найдено в интернете, проходит через системы The Echo Nest. Эти системы настроены на то, чтобы извлекать ключевые слова и термины. Этим данным потом присваивается собственный вес, который говорит про их важность. Акустический анализ начинается с того, что композиция разбивается на небольшие кусочки. Затем для каждого сегмента определяются громкость, тембр и другие характеристики. Далее полученные данные объединяются и анализируются. Анализ проводится с помощью методов машинного обучения, что позволяет понять композицию на высоком уровне. Объединение акустического и текстового анализа позволило создать мощную технологию, которая сделала компанию мировым лидером в алгоритмах анализа музыки. В 2014 году компанию купил мировой гигант стримминга музыки - Spotify.

1.2 Niland

Niland - компания из Парижа, которая занимается поиском и рекомендациями музыкальных композиций. Их продукт анализирует композицию, используя акустический анализ. Решаемые задачи разделяются на 2 типа: оценка схожести и классификация. Для оценки схожести используется подход music2vec. То есть преобразование композиции в вектор из значений характеристик. Для классификации композиций используется преобразование музыки в вектор для дальнейшей обработки с помощью алгоритмов машинного обучения. Преобразование музыки в вектор включает в себя:

- а) Вычисление спектрограммы, т.е. вычисление интенсивности различных частот в различные моменты времени;
- б) Извлечение кратковременных особенностей, т.е. свойств сигнала, которые имеют высокую частоту обновления;

в) Моделирование статического распределения кратковременных особенностей и объединение их в вектор.

В результате получается вектор большой размерности, порядка тысячи элементов, который впоследствии используется для классификации композиций и извлечения характеристик высокого уровня.

1.3 Music Technology Group

Music Technology Group - исследовательская группа с факультета информации и коммуникационных технологий университета Pompeu Fabra, Барселоны. Специализируется на вычислительных задачах в области музыки и звука. Свои исследования группа основывает на знаниях из других областей, таких как обработка сигналов, машинное обучение и взаимодействие человека с компьютером. Темы исследований, которыми группа занимается в данный момент:

а) Обработка аудиосигналов, т.е. спектральное моделирование для синтеза и трансформации звука;

б) Описание звука и музыки, т.е. семантический анализ и классификация аудио;

в) Продвинутое взаимодействие с музыкой, т.е. разработка интерфейсов для создания и изучения музыки;

г) Звуковые и музыкальные сообщества, т.е. технологии социальных сетей для музыкальных и звуковых приложений.

Группа создала библиотеку для определения степени похожести и классификации музыки для описания музыки с помощью высокоуровневых характеристик - Gaia. Обработываются данные в несколько этапов: подготовка данных, обучение на подготовленных данных, проверка результатов. Подготовка данных включает в себя:

а) удаление метаданных;

б) извлечение тональных характеристик;

в) нормализацию выделенных характеристик;

г) преобразование низкоуровневых характеристик в нормально распределенные величины.

Для обучения используется метод опорных векторов с полиномиальным ядром (однородным). В качестве метрики точности используется n-кратная кроссвалидация.

1.4 DeepMind

DeepMind - компания, которая занимается искусственным интеллектом. Ранее она была известна под названием Google DeepMind. Компания занимается решением проблем, связанных с интеллектом, т.е. занимается разработкой обучающихся алгоритмов общего назначения. На данный момент компания сосредоточила свои усилия на разработке интеллекта, который способен играть в компьютерные игры - от стратегических до аркад. Сандер Дилеман, ученый из этой компании, стал соавтором статьи ([ССЫЛКА НА СТАТЬЮ В ЛИТЕРАТУРЕ](#)), в которой утверждается, что глубинное обучение способно гораздо лучше справляться с рекомендациями, чем колаборативная фильтрация. В качестве инструмента использовалась сверточная нейронная сеть с 7-8 слоями. Для визуализации данных использовался алгоритм t-SNE. Алгоритм определял инструменты, аккорды, и даже гармонии и прогрессии.

2 ТРЕБОВАНИЯ К ПРОЕКТИРУЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ

По результатам изучения предметной области, анализа литературных источников и обзора существующих систем-аналогов сформулируем требования к проектируемому программному средству.

Сервис должен уметь извлекать информацию из распространенных форматов аудиозаписей.

Сервис должен выдавать высокоуровневые характеристики (жанр) и низкоуровневые характеристики (длина, частота ударов в минуту и тп).

Давать доступ к функциональности выше по API.

Так как в сервисе реализованы сложные математические вычисления, то они должны иметь достаточный уровень оптимальности, чтобы обрабатывать данные за разумное время с использованием разумного количества ресурсов. Так же уровень оптимизации влияет на системные требования данного продукта.

Иметь модульную архитектуру для удобного изменения алгоритмов обработки информации на более актуальные.

Являются модулем, который просто встраивается в сторонние системы.

Иметь адаптивный формат данных для более удобной интеграции с системой пользователя. – пока не точно!!!

Обрабатывать звуковую информацию на уровне человеческого слуха.
(в плане характеристик звука)

УПОРЯДОЧИТЬ!!!!

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Для модульности - упаковать в контейнер - Docker

Для того, чтобы писать модели - Keras

Для того, чтобы модели исполнялись оптимизированно - TensorFlow

В качестве веб-сервера, чтобы было ничего лишнего - Flask

Для того, чтобы обеспечить общение между моделью и сервером - RedisMQ

Модель упаковать, чтобы все время висела в памяти - Daemon

Чтобы все запускать единомоментно - Supervisor

Чтобы считать спектры оптимальным образом - Essentia

Чтобы обрабатывать большое число запросов в секунду - Gunicorn

Чтобы задавать количество workers, default формат для общения - Environment variables???

СДЕЛАТЬ КРАСИВО И БОЛЬШЕ ВОДЫ!!!!

ВОЗМОЖНО ПЕРЕНЕСТИ ТО, ЧТО НИЖЕ К ПРОЕКТИРОВАНИЮ!!!!

Для того, чтобы уменьшить количество обрабатываемой информации было решено использовать мел-частотные кепстральные коэффициенты. Достоинствами этого метода являются:

а) Используется спектр сигнала, что позволяет учесть волновую природу звука;

б) Спектр проецируется на мел-шкалу, что позволяет учесть восприятие частот человеческим ухом;

в) Позволяет сжать количество информации количеством вычисляемых коэффициентов.

Ниже приведен алгоритм вычисления мел-частотных кепстральных коэффициентов:

а) Применить к сигналу оконное преобразование Фурье;

б) Для каждого полученного спектра:

1) Расположить спектр на мел-шкале (в результате получим мел-частотный спектр);

2) Возвести полученные коэффициенты в квадрат и прологорифмировать их;

3) Применить к результату преобразование Фурье.

4 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

Все, что в разделе выше, правда описание написанных там штук

5 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ

+20 - 30 СТРАНИЦ ДОПИСАТЬ К ЗАПИСКЕ

6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И ВНЕДРЕНИЯ ПРОГРАММНОГО СРЕДСТВА

6.1 Характеристика программного средства

...

Разработки проектов программных средств связана со значительными затратами ресурсов. В связи с этим создание и реализация каждого проекта программного обеспечения нуждается в соответствующем технико-экономическом обосновании [2], которое и описывается в данном разделе.

6.2 Определение объема и трудоемкости ПС

Целесообразность создания ПС требует проведения предварительной экономической оценки. Экономический эффект у разработчика ПС зависит от объема инвестиций в разработку проекта, цены на готовый продукт и количества проданных копий, и проявляется в виде роста чистой прибыли.

Оценка стоимости создания ПС со стороны разработчика предполагает составление сметы затрат, вычисление цены и прибыли от реализации разрабатываемого программного средства.

Исходные данные, которые будут использоваться при расчете сметы затрат, представлены в таблице 6.1.

Перед определением сметы затрат на разработку программного средства необходимо определить его объём. Однако, на стадии ТЭО нет возможности рассчитать точные объёмы функций, вместо этого с помощью применения действующих нормативов рассчитываются прогнозные оценки.

В качестве метрики измерения объема программных средств используется строка их исходного кода (LOC – lines of code). Данная метрика широко распространена, поскольку она непосредственно связана с конечным продуктом, может применяться на всём протяжении проекта и, кроме того, может использоваться для сопоставления размеров программного обеспечения. Далее под строкой исходного кода будем понимать количество исполняемых операторов.

Расчет объема функций программного средства и общего объема приведен в таблице 6.2.

Исходя из определенной 3-ей категории сложности и общего объема ПС $V_0 = 28\,940$, нормативная трудоемкость $T_n = 520$ чел./д. [2, приложение 3]. Перед определением общей трудоемкости разработки необходимо определить несколько коэффициентов.

Таблица 6.1 – Исходные данные

Наименование показателя	Условное обозначение	Значение
Категория сложности		3
Дополнительный коэффициент сложности	$\sum_{i=1}^n K_i$	0,13
Степень охвата функций стандартными модулями	K_T	0,7
Коэффициент новизны	K_H	0,9
Количество дней в году	D_r	365
Количество праздничных дней в году	$D_{п}$	9
Количество выходных дней в году	$D_{в}$	103
Количество дней отпуска	D_o	21
Количество разработчиков	$Ч_p$	3
Тарифная ставка первого разряда, руб.	$T_{ч}^1$	265
Среднемесячная норма рабочего времени, ч.	Φ_p	168,3
Продолжительность рабочей смены, ч.	$T_{ч}$	8
Коэффициент премирования	K	1,3
Норматив дополнительной заработной платы	H_d	20%
Норматив отчислений в ФСЗН	$H_{сз}$	34%
Норматив отчислений по обязательному страхованию	$H_{ос}$	0,6%
Норматив расходов по статье «Материалы»	$H_{мз}$	5%
Норматив расходов по статье «Машинное время»	$H_{мв}$	15%
Понижающий коэффициент к статье «Машинное время»		0,5
Стоимость машино-часа, руб.	$Ц_m$	0,8
Норматив расходов по статье «Научные командировки»	$H_{рнк}$	15%
Норматив расходов по статье «Прочие затраты»	$H_{пз}$	20%
Норматив расходов по статье «Накладные расходы»	$H_{нр}$	50%
Уровень рентабельности	$У_{рп}$	10%
Ставка НДС	$H_{дс}$	20%
Норматив затрат на освоение ПО	H_o	10%
Норматив затрат на сопровождение ПО	H_c	20%

Коэффициент сложности, который учитывает дополнительные затраты труда, связанные с обеспечением интерактивного доступа и хранения, и

Таблица 6.2 – Перечень и объём функций программного модуля

№ функции	Наименование (содержание)	Объём функции, LoC
101	Организация ввода информации	100
102	Контроль, предварительная обработка и ввод информации	500
109	Организация ввода/вывода информации в интерактивном режиме	190
111	Управление вводом/выводом	2600
204	Обработка наборов и записей базы данных	1900
207	Манипулирование данными	8000
208	Организация поиска и поиск в БД	7500
304	Обслуживание файлов	500
305	Обработка файлов	800
309	Формирование файла	1000
506	Обработка ошибочных и сбойных ситуаций	500
507	Обеспечение интерфейса между компонентами	750
601	Отладка прикладных программ в интерактивном режиме	4300
707	Графический вывод результатов	300
	Общий объем	28 940

поиска данных в сложных структурах [2, приложение 4, таблица П.4.2]

$$K_c = 1 + \sum_{i=1}^n K_i = 1 + 0,06 + 0,07 = 1,13, \quad (6.1)$$

где K_i — коэффициент, соответствующий степени повышения сложности за счет конкретной характеристики;

n — количество учитываемых характеристик.

Коэффициент K_t , учитывающий степень использования при разработке стандартных модулей, для разрабатываемого приложения, в котором степень охвата планируется на уровне около 50%, примем равным 0,7 [2, приложение 4, таблица П.4.5].

Коэффициент новизны разрабатываемого программного средства K_n примем равным 0,9, так как разрабатываемое программное средство принадлежит определенному параметрическому ряду существующих программных средств [2, приложение 4, таблица П.4.4].

Исходя из выбранных коэффициентов, общая трудоемкость разработ-

ки $T_o = T_n \cdot K_c \cdot K_t \cdot K_n = 520 \cdot 1,13 \cdot 0,7 \cdot 0,9 = 370$ чел./д.

Для расчета срока разработки проекта примем число разработчиков $Ч_p = 3$. Исходя из комментария к постановлению Министерства труда и социальной защиты Республики Беларусь от 05.10.16 №54 «Об установлении расчетной нормы рабочего времени на 2017 год» [3], эффективный фонд времени работы одного человека составит

$$\Phi_{эф} = D_r - D_n - D_v - D_o = 365 - 9 - 103 - 21 = 232 \text{ д.}, \quad (6.2)$$

где D_r — количество дней в году;

D_n — количество праздничных дней в году;

D_v — количество выходных дней в году;

D_o — количество дней отпуска.

Тогда трудоемкость разработки проекта

$$T_p = \frac{T_o}{Ч_p \cdot \Phi_{эф}} = \frac{370}{3 \cdot 232} = 0,53 \text{ г.} = 194 \text{ д.} \quad (6.3)$$

Исходя из того, что разработкой будет заниматься 3 человека, можно запланировать фонд рабочего времени для каждого исполнителя

$$\Phi_{ин} = \frac{T_p}{Ч_p} = \frac{194}{3} \approx 65 \text{ д.} \quad (6.4)$$

6.3 Расчет сметы затрат

Основной статьей расходов на создание ПО является заработная плата разработчиков проекта. Информация об исполнителях перечислена в таблице 6.3. Кроме того, в таблице приведены данные об их тарифных разрядах, приведены разрядные коэффициенты, а также по формулам 6.5 и 6.6 рассчитаны месячный и часовой оклады.

$$T_m = T_m^1 \cdot T_k, \quad (6.5)$$

$$T_{ч} = \frac{T_m}{\Phi_p}, \quad (6.6)$$

где T_m — месячный оклад;
 T_m^1 — тарифная ставка 1-го разряда (положим ее равной 265 руб.);
 T_k — тарифный коэффициент;
 $T_{\text{ч}}$ — часовой оклад;
 Φ_p — среднемесячная норма рабочего времени (в 2017 г. составляет 168,3 ч. [3]).

Таблица 6.3 – Работники, занятые в проекте

Исполнители	Разряд	Тарифный коэффициент	Месячный оклад, руб.	Часовой оклад, руб.
Руководитель проекта	17	3,98	1054,70	6,27
Ведущий инженер-программист	15	3,48	922,20	5,48
Инженер-программист II категории	11	2,65	702,25	4,17

Тогда основная заработная плата исполнителей составит

$$Z_o = \sum_{i=1}^n T_{\text{чи}} \cdot T_{\text{ч}} \cdot \Phi_{\text{пи}} \cdot K =$$

$$= (6,27 + 5,48 + 4,17) \cdot 8 \cdot 65 \cdot 1,3 = 10\,761,92 \text{ руб.}, \quad (6.7)$$

где $T_{\text{чи}}$ — часовая тарифная ставка i -го исполнителя, руб.;
 $T_{\text{ч}}$ — количество часов работы в день;
 $\Phi_{\text{пи}}$ — плановый фонд рабочего времени i -го исполнителя, д.;
 K — коэффициент премирования (принятый равным 1,3).

Дополнительная заработная плата включает выплаты, предусмотренные законодательством о труде: оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей, и определяется по нормативу, установленному в организации, в процентах к основной заработной плате. Приняв данный норматив $H_d = 20\%$, рассчитаем дополнительные выплаты

$$Z_d = \frac{Z_o \cdot H_d}{100\%} = \frac{10\,761,92 \cdot 20\%}{100\%} = 2152,38 \text{ руб.} \quad (6.8)$$

Отчисления в фонд социальной защиты населения и в фонд обязательного страхования определяются в соответствии с действующим законодательством по нормативу в процентном отношении к фонду основной и до-

полнительной зарплат по следующим формулам

$$\begin{aligned} Z_{сз} &= \frac{(Z_o + Z_d) \cdot H_{сз}}{100\%}, \\ Z_{ос} &= \frac{(Z_o + Z_d) \cdot H_{ос}}{100\%}. \end{aligned} \quad (6.9)$$

В настоящее время нормы отчислений в ФСЗН $H_{сз} = 34\%$ и в фонд обязательного страхования $H_{ос} = 0,6\%$. Исходя из этого, размеры отчислений

$$\begin{aligned} Z_{сз} &= \frac{(10\,761,92 + 2152,38) \cdot 34\%}{100\%} = 4390,86 \text{ руб.}, \\ Z_{ос} &= \frac{(10\,761,92 + 2152,38) \cdot 0,6\%}{100\%} = 77,49 \text{ руб.} \end{aligned} \quad (6.10)$$

Расходы по статье «Материалы» отражают траты на магнитные носители, бумагу, красящие материалы, необходимые для разработки ПО определяются по нормативу к фонду основной заработной платы разработчиков. Исходя из принятого норматива $H_{мз} = 5\%$ определим величину расходов

$$M = \frac{Z_o \cdot H_{мз}}{100\%} = \frac{10\,761,92 \cdot 5\%}{100\%} = 538,10 \text{ руб.} \quad (6.11)$$

Расходы по статье «Машинное время» включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам на 100 строк исходного кода. Норматив зависит от характера решаемых задач и типа ПК; для текущего проекта примем $H_{мв} = 15\%$ [2, приложение 6]. Примем величину стоимости машино-часа $C_m = 0,8$ руб. Тогда, применяя понижающий коэффициент 0,5, получим величину расходов

$$P_m = C_m \cdot \frac{V_o}{100} \cdot H_{мв} = 0,8 \cdot \frac{28\,940}{100} \cdot 15\% \cdot 0,5 = 1736,40 \text{ руб.} \quad (6.12)$$

Расходы по статье «Научные командировки» определяются по нормативу в процентах к основной заработной плате. Принимая норматив равным $H_{рнк} = 15\%$ получим величину расходов

$$P_{рнк} = \frac{Z_o \cdot H_{рнк}}{100\%} = \frac{10\,761,92 \cdot 15\%}{100\%} = 1614,29 \text{ руб.} \quad (6.13)$$

Расходы по статье «Прочие затраты» включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу в процентах к основной зара-

ботной плате. Принимая норматив равным $H_{пз} = 20\%$ получим величину расходов

$$П_з = \frac{З_o \cdot H_{пз}}{100\%} = \frac{10\,761,92 \cdot 20\%}{100\%} = 2152,38 \text{ руб.} \quad (6.14)$$

Затраты по статье «Накладные расходы», связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды, относятся к конкретному ПО по нормативу в процентном отношении к основной заработной плате исполнителей. Принимая норматив равным $H_{нр} = 50\%$ получим величину расходов

$$P_n = \frac{З_o \cdot H_{нр}}{100\%} = \frac{10\,761,92 \cdot 50\%}{100\%} = 5380,96 \text{ руб.} \quad (6.15)$$

Общая сумма расходов по смете определяется как сумма вышерассчитанных показателей

$$C_{п} = З_o + З_d + З_{сз} + З_{ос} + M + P_m + P_{нк} + П_з + P_n = 28\,804,78 \text{ руб.} \quad (6.16)$$

Рентабельность определяется из результатов анализа рыночных условий и переговоров с потребителями ПО. Исходя из принятого уровня рентабельности $Y_{рп} = 10\%$, прибыль от реализации ПО составит

$$П_o = \frac{C_{п} \cdot Y_{рп}}{100\%} = \frac{28\,804,78 \cdot 10\%}{100\%} = 2880,48 \text{ руб.} \quad (6.17)$$

На основании расчета прибыли и уровня себестоимости рассчитаем прогнозируемую цену программного средства без учета налогов

$$Ц_{п} = C_{п} + П_o = 28\,804,78 + 2880,48 = 31\,685,26 \text{ руб.} \quad (6.18)$$

Далее рассчитаем налог на добавленную стоимость

$$НДС = \frac{Ц_{п} \cdot H_{дс}}{100\%} = \frac{31\,685,26 \cdot 20\%}{100\%} = 6337,05 \text{ руб.} \quad (6.19)$$

НДС включается в прогнозируемую отпускную цену

$$Ц_o = Ц_{п} + НДС = 31\,685,26 + 6337,05 = 38\,022,31 \text{ руб.} \quad (6.20)$$

Организация-разработчик участвует в освоении и внедрении ПО и несет соответствующие затраты, которые определяются по нормативу $H_o = 10\%$

от себестоимости ПО в расчете на три месяца

$$P_o = \frac{C_n \cdot H_o}{100\%} = \frac{28\,804,78 \cdot 10\%}{100\%} = 2880,48 \text{ руб.} \quad (6.21)$$

Кроме того, организация-разработчик осуществляет сопровождение ПО,¹⁾ которое также оплачивается заказчиком. Расчет осуществляется в соответствии с нормативом $H_c = 20\%$ от себестоимости ПО

$$P_c = \frac{C_n \cdot H_c}{100\%} = \frac{28\,804,78 \cdot 20\%}{100\%} = 5760,96 \text{ руб.} \quad (6.22)$$

Экономическим эффектом разработчика будет являться сумма прибыли с вычетом налога на прибыль

$$P_{\text{ч}} = P_o - \frac{P_o \cdot H_{\text{п}}}{100\%} = 2880,48 - \frac{2880,48 \cdot 18\%}{100\%} = 2361,99 \text{ руб.} \quad (6.23)$$

6.4 Оценка экономической эффективности применения ПС у пользователя

В результате применения нового ПО пользователь может понести значительные капитальные затраты на приобретение и освоение ПО, доукомплектованием ЭВМ новыми техническими средствами и пополнение оборотных средств. Однако, если приобретенное ПО будет в достаточной степени эффективнее базового, то дополнительные капитальные затраты быстро окупятся.

Для определения экономического эффекта от использования нового ПО у потребителя необходимо сравнить расходы по всем основным статьям сметы затрат на эксплуатацию нового ПО с расходами по соответствующим статьям базового варианта. При этом за базовый вариант примем ручной вариант. Исходные данные для расчета приведены в таблице 6.4.

Общие капитальные вложения заказчика (потребителя) вычисляются следующим образом

$$\begin{aligned} K_o &= K_{\text{пр}} + K_{\text{ос}} + K_c = 38\,022,31 + 2880,48 + 5760,96 = \\ &= 46\,663,75 \text{ руб.} \end{aligned} \quad (6.24)$$

где $K_{\text{пр}}$ — затраты пользователя на приобретение ПО по отпускной цене;
 $K_{\text{ос}}$ — затраты пользователя на освоение;
 K_c — затраты пользователя на оплату услуг по сопровождению.

¹⁾На февраль 2017 г. [4]

Таблица 6.4 – Исходные данные

Наименование показателя	Условное обозначение	Значение в базовом варианте	Значение в новом варианте
Затраты пользователя на приобретение ПО	$K_{пр}$	—	38 022,31 руб.
Затраты пользователя на освоение	$K_{ос}$	—	2880,48 руб.
Затраты пользователя на сопровождение	K_c	—	5760,96 руб.
Трудоемкость на задачу, чел./ч.	T_{c1}, T_{c2}	2	0,1
Средняя зарплата ¹⁾ , руб.	$Z_{см}$	716,5	716,5
Количество выполняемых задач	A_1, A_2	1460	1460
Время простоя сервиса, мин. в день	P_1, P_2	50	10
Стоимость одного часа простоя, руб.	$C_{п}$	79,8	79,8

В качестве типичного примера использования разрабатываемого ПС предполагается сценарий ее использования ежедневно 4 раза в день, при этом предполагается снижение трудоемкости с $T_{c1} = 2$ чел./ч. до $T_{c2} = 0,1$ чел./ч.. Приняв среднемесячную заработную плату работника $Z_{см} = 716,5$ руб., рассчитаем экономию затрат на заработную плату в расчете на одну задачу $C_{зе}$ и за год C_3

$$C_{зе} = \frac{Z_{см} \cdot (T_{c1} - T_{c2})}{\Phi_p} = \frac{716,5 \cdot (2 - 0,1)}{168,3} = 8,09 \text{ руб.}, \quad (6.25)$$

$$C_3 = C_{зе} \cdot A_2 = 8,09 \cdot 1460 = 11\,811,4 \text{ руб.}, \quad (6.26)$$

где Φ_p — среднемесячная норма рабочего времени, ч.

Экономия с учетом начислений на зарплату

$$C_n = C_3 \cdot \frac{100\% + K}{100\%} = 11\,811,4 \cdot \frac{100\% + 1,3}{100\%} = 11\,964,95 \text{ руб.}, \quad (6.27)$$

где K — норматив начислений на зарплату, руб.

Экономия за счет сокращения простоев сервиса

$$C_c = \frac{(\Pi_1 - \Pi_2) \cdot D_{\text{рг}} \cdot C_{\text{п}}}{60} = \frac{(50 - 10) \cdot 300 \cdot 79,8}{60} = 15\,960,00, \text{ руб.}, \quad (6.28)$$

где $D_{\text{рг}}$ — плановый фонд работы сервиса, д.

Тогда общая годовая экономия текущих затрат, связанных с использованием нового ПО

$$C_o = C_{\text{н}} + C_c = 11\,964,95 + 15\,960,00 = 27\,924,95 \text{ руб.} \quad (6.29)$$

Внедрение нового ПО позволит пользователю сэкономить на текущих затратах, то есть получить на эту сумму дополнительную прибыль. Для пользователя в качестве экономического эффекта выступает лишь чистая прибыль. Принимая размер ставки налога на прибыль $H_{\text{п}} = 18\%$ получим

$$\Delta\Pi_{\text{ч}} = C_o - \frac{C_o \cdot H_{\text{п}}}{100\%} = 27\,924,95 - \frac{27\,924,95 \cdot 18\%}{100\%} = 22\,898,46 \text{ руб.} \quad (6.30)$$

В процессе использования нового ПО чистая прибыль в конечном итоге возмещает капитальные затраты. Однако полученные при этом суммы прибыли и затрат по годам приводят к единому времени – расчетному году (за расчетный год принят 2017-й год) путем умножения результатов и затрат за каждый год на коэффициент дисконтирования α . При расчете используются следующие коэффициенты: 2017 г. – 1, 2018 г. – 0,8696, 2019 г. – 0,7561, 2020 г. – 0,6575. Все рассчитанные данные экономического эффекта сводятся в таблицу 6.5.

В результате технико-экономического обоснования применения программного средства были получены следующие значения показателей эффективности:

- чистая прибыль разработчика составит $\Pi_{\text{ч}} = 2361,99$ руб.;
- затраты заказчика окупятся уже на четвертом году использования;
- экономическая эффективность для заказчика, выраженная в виде чистого дисконтированного дохода, составит 5618,02 руб. за четыре года использования данного ПС; более высокий прирост прибыли заказчик получит по истечению данного срока.

Полученные результаты свидетельствуют об эффективности разработки и внедрения проектируемого программного средства.

Таблица 6.5 – Расчет экономического эффекта от использования нового ПО

Наименование показателя	2017 г.	2018 г.	2019 г.	2020 г.
<i>Результаты</i>				
Коэффициент приведения	1	0,8696	0,7561	0,6575
Прирост прибыли за счет экономии затрат (P_q), руб.	—	22 898,46	22 898,46	22 898,46
То же с учетом фактора времени, руб.	—	19 912,50	17 313,53	15 055,74
<i>Затраты</i>				
Приобретение ПО ($K_{пр}$), руб.	38 022,31	—	—	—
Освоение ПО ($K_{ос}$), руб.	2880,48	—	—	—
Сопровождение ПО (K_c), руб.	5760,96	—	—	—
Всего затрат (K_o), руб.	46 663,75	—	—	—
То же с учетом фактора времени, руб.	46 663,75	—	—	—
<i>Экономический эффект</i>				
Превышение результата над затратами, руб.	–46 663,75	19 912,5	17 313,53	15 055,74
То же с нарастающим итогом, руб.	–46 663,75	–26 751,25	–9437,72	5618,02

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] ISTQB glossary – Specification [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://glossary.astqb.org/search/specification>. — Дата доступа: 25.03.17.

[2] Палицын, В.А. Техничко-экономическое обоснование дипломных проектов: Метод. пособие для студ. всех спец. БГУИР. В 4-х ч. Ч. 4: Проекты программного обеспечения / В.А. Палицын. — Мн : БГУИР, 2006. — 76 с.

[3] Пещенко, Е.А. Производственный календарь на 2017 год [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://www.mintrud.gov.by/system/extensions/spaw/uploads/files/Kommetarij-2017-RV.pdf>. — Дата доступа: 06.04.17.

[4] Белстат. О начисленной средней заработной плате работников в феврале 2017 г. [Электронный ресурс]. — Электронные данные. — Режим доступа: http://www.belstat.gov.by/ofitsialnaya-statistika/solialnaya-sfera/trud/operativnaya-informatsiya_8/o-nachislennoi-srednei-zarabotnoi-plate-rabotnikov/o-nachislennoy-sredney-zarabotnoy-plate-rabotnikov-v-fevrале-2017-nbsp-g/. — Дата доступа: 07.04.17.

ПРИЛОЖЕНИЕ А

(обязательное)

Фрагменты исходного кода

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="./node_modules/semantic-ui-css/semantic.min.css"/>
</head>
<body>
  <div id="root"></div>
  <script src="./node_modules/react/dist/react.js"></script>
  <script src="./node_modules/react-dom/dist/react-dom.js"></script>

  <script src="./dist/bundle.js"></script>
</body>
</html>
```