

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

УДК 004.413.2:004.774

На правах рукописи

ТОЛСТЫХ
Максим Андреевич

**СИСТЕМА КОНТРОЛЯ ВЫПОЛНЕНИЯ ЗАДАЧ НА ОСНОВЕ
WEB-ТЕХНОЛОГИЙ**

АВТОРЕФЕРАТ

диссертации на соискание степени
магистра техники и технологии

по специальности 1-39 80 03 «Электронные системы и технологии»
(профилизация «Компьютерные технологии проектирования
электронных систем»)

Минск 2021

Работа выполнена на кафедре проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

Научный руководитель: **ГРЕМЕНОК Валерий Феликсович**,
Доктор физ.-мат. наук, профессор кафедры проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

Рецензент: **ГРИГОРЬЕВА Наталья Александровна**,
кандидат экономических наук, доцент кафедры «Строительные материалы и технология строительства» БНТУ

Защита диссертации состоится «24» июня 2021 г. года в 9⁰⁰ часов на заседании Государственной экзаменационной комиссии по защите магистерских диссертаций в учреждении образования «Белорусский государственный университет информатики и радиоэлектроники» по адресу: 220013, Минск, ул. П.Бровки, 6, копр. 1, ауд. 415, тел. 293-20-80, E-mail: kafpiks@bsuir.by

С диссертацией можно ознакомиться в библиотеке учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

ВВЕДЕНИЕ

Вне зависимости от рода и вида деятельности человека результат его труда напрямую зависит от продуктивности. Высокая работоспособность позволяет эффективнее и быстрее достигать поставленные цели.

Продуктивность человека – это количество и качество результатов его интеллектуального и физического труда. Иметь возможность поддерживать её на высоком уровне значит эффективно использовать свое время для достижения поставленных целей.

Чем выше продуктивность, тем более осмысленным и качественным ощущается труд. Это положительно сказывается на самореализации так как раскрываются таланты и реализуются навыки человека.

В современном мире, когда зачастую приходится делать много дел за краткий промежуток времени, тяжело качественно сфокусироваться на выполнении задач. Так же стоит помнить про такое явление, как информационный шум, мешающий концентрации на выполняемой работе. Подобная проблема существовала и в рамках офисов, но особо остро стала в период массового перевода сотрудников на удаленную работу. Её особенностью являются размытые границы между рабочим и личным пространством, что негативно сказывается на продуктивности.

Для решения проблемы продуктивности и концентрации придумано множество техник, методик и подходов. Какие-то из них основаны на простой идее вроде ведения списка задач, а другие на чем-то вроде методик организации потока этих задач. Зачастую подходы комбинируются с целью максимизации эффекта. Тем не менее

В силу того, что для каждого человека методики работают по-разному, есть необходимость в поиске наиболее оптимального сочетания подходов к решению задач продуктивности.

Из-за ускорений темпов разработки программного обеспечения нецелесообразно разрабатывать различные нативные версии для разнообразных платформ. В такой ситуации хорошим выбором будет использование *web*-технологий. В дальнейшем это позволит легко мигрировать на мобильные и настольные платформы с помощью технологий, подобных *React Native*.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

В настоящее время перед человеком стоит проблема переизбытка информации, которая негативно отражается на его продуктивности. Современные программные средства предлагают решение этой проблемы, применяя различные методы. Как и во многих сферах жизни, важно подобрать наиболее актуальный и рабочий вариант персонально для каждого.

Особо остро эта проблема стала в связи с массовым переводом сотрудников компаний на удаленную работу.

В связи с вышеизложенным, актуальным является разработка программного средства, основанного на технике помodoro с использованием *web*-технологий.

Степень разработанности проблемы

В процессе работы над магистерской диссертацией был проведен обзор основных решений на рынке и подходов, которые лежат в основе данных программных средств. Недостатком рассмотренных средств является то, что они преимущественно соединяют с себе лишь одну основную идею для решения задач продуктивности.

Цель и задачи исследования

Целью магистерской диссертации является разработка собственного программного средства, которое объединит в себе несколько подходов к решению задач продуктивности.

Для выполнения поставленной цели в работе были сформулированы следующие задачи:

1. Проанализировать уже существующие аналоги, представленные на рынке, и принципы, лежащие в их основе;
2. Применить принцип помodoro в создании программного средства на основе *web*-технологий;
3. Исследовать полученное решение путем нагрузочного тестирования и выделить способы дальнейших улучшений.

Объектом исследования являются системы контроля выполнения задач.

Предметом исследования эффективность системы контроля задач на основе техники помodoro и современных *web*-технологий.

Область исследования

Содержание диссертации соответствует образовательному стандарту высшего образования второй ступени (магистратуры) ОСВО 1-39 80 03-2019 специальности 1-39 80 03 Электронные системы и технологии (профилизация «Компьютерные технологии проектирования электронных систем»)

Теоретическая и методологическая основа исследования

В основу диссертации легли представленные на рынке программные средства и методы, лежащие в их основе.

Информационная база исследования сформирована на основе данных, полученных из открытых источников в сети интернет, научных изданий и публикаций, которые описывают методы и средства решения рассматриваемого объекта.

Научная новизна

Научная новизна диссертационной работы заключается в применении принципа помодоро к *web*-сервису контроля выполнения задач.

Теоретическая значимость диссертации заключается в том, что в ней описан поход к созданию современного сервиса контроля выполнения задач с применением принципа помодоро.

Практическая значимость диссертации состоит в том, что разработанный сервис использует принцип помодоро в своей основе, а значит сможет удовлетворить запросы пользователей, которые искали именно такого решения проблем продуктивности.

Основные положения, выносимые на защиту

1. Произведен обзор современных систем контроля выполнения задач на основе *web*-технологий и принципов, лежащих в их основе, что позволило предложить альтернативное подход к решению задач продуктивности;
2. Проанализированы технологии, лежащие в основе разработанного сервиса, с описанием их преимуществ и недостатков;
3. Разработано программное средство контроля выполнения задач. Рассмотрены технические аспекты разработки и произведено тестирование сервиса при помощи инструмента *Lighthouse*.

Апробация и внедрение результатов исследования

Результаты исследования были представлены на международных конференциях «Научные исследования XXI века» (г. Воронеж, Российская Федерация, 2020 год), «Внедрение передового опыта и практическое применение результатов инновационных исследований» (г. Волгоград, Российская Федерация, 2020 год), «Концепция, теория и методика фундаментальных и прикладных научных исследований» (г. Екатеринбург, Российская Федерация, 2021 год), «Прорывные научные исследования как двигатель науки» (г. Новосибирск, Российская Федерация, 2021 год), «Наука в современном обществе: закономерности и тенденции развития» (г. Омск, Российская Федерация, 2021 год).

Публикации

Изложенные в диссертации положения и выводы опубликованы в 9 печатных работах на международных конференциях. Общий объем публикаций по теме составляет 22 страницы.

Структура и объём работы

Диссертация состоит из введения, общей характеристики работы, трех глав с краткими выводами по каждой главе, заключения, библиографического списка и приложений.

В первой главе приведен обзор современных программных решения для контроля выполнения задача на основе *web*-технологий. Выделены недостатки и преимущества существующих решений.

Во второй главе представлен обзор основных технологий и методологий, на основе которых будет разработано программное средство. Описаны особенности и технические аспекты работы технологий.

В третьей главе будут описан процесс разработки и проведено тестирование готового решения. Сделаны выводы касательно результатов тестирования и предложены пути улучшения приложения.

В приложении представлен исходный код одного из главных модулей приложения.

Общий объем диссертационной работы составляет 102 страниц. Из них 36 страниц основного текста, 25 иллюстраций на 20 страницах, 5 таблиц на 2 страницах, библиографический список из 50 наименований на 4 страницах, список собственных публикаций соискателя из 9 наименований на 2 странице, 3 приложений на 57 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введение** рассмотрено современное состояние проблемы контроля выполнения задач и продуктивности.

В **общей характеристике работы** показана актуальность проводимых исследований, степень разработанности проблемы, сформулированы цель и задачи диссертации, обозначена область исследований, научная значимость исследований, а также апробация работы.

В **первой главе** приведено обзор современного состояние проблемы систем контроля выполнения задач на основе *web*-технологий.

Проанализированы основные и наиболее популярные подходы к решению задач продуктивности и эффективности выполнения задач.

Из анализа следует, что в текущий момент на рынке систем контроля

выполнения задач представлено достаточно много различных решений. Все они основаны на различных подходах и решают задачу немного отлично от все остальных приложений. Тем не менее, не смотря на достаточно крупный рынок программного обеспечения, разнообразие принципов и подходов позволяют подобрать пользователю что-то максимально подходящее под его требования. С целью понять особенности каждого приложения, были проанализированы. Выделены основные положительные и отрицательные стороны этих решений.

При проведении анализа рынка было выявлено, что существует определенная ниша в данном виде программного обеспечения, которая ещё не была занята никаким из конкурентов.

Во второй главе представлены основные идеи и технологии, лежащие в основе разрабатываемого программного обеспечения.

Главной идеей является использование принципа «*Pomodoro*», который был придуман предпринимателем Франческо Чирилло. Принцип был назван именно так в честь кухонного таймера, который в то время был у автора принципа, которым он пользовался для отслеживания эффективности выполнения домашних задач для университета.

Основа метода очень проста. При наличии большого задания его необходимо разбить на множество мелких временных интервалов. Между ними стоит запланировать небольшие паузы. Техника помогает мозгу эффективно фокусироваться на небольшие промежутки времени, при этом избегая переутомления.

Данная система работает циклами. Задачи делаются короткими перебежками, что позволяет не терять внимание из-за слишком длительной рабочей сессии.

Кратко технику можно представить в виде пяти шагов:

- поставить задачу;
- запустить таймер на 25 минут;
- работать над задачей в течении интервала, не отвлекаясь на побочные активности;
- сделать перерыв после выполненной сессии;
- каждые четыре сессии сделать перерыв подольше.

Обычно перерыв побольше означает 20-30 минут отдыха. Это время уходит на отдых и подготовку к следующему спринту.

Во время выполнения задач важно не отвлекаться на посторонние дела. Если всё же не получилось и что-то отвлекло от основной задачи – необходимо завести таймер заново. Если побочное дело терпит, то можно выписать его на лист бумаги и вернуться после того, как сессия будет закончена.

В основе техники помodoro лежит гибкость. Её нужно использовать для того, чтобы подстроить технику под себя. Четыре сессии под 25 минут и перерывы в 5 минут являются лишь основной, от которой стоит отталкиваться. Необходимо учитывать, что не все виды деятельности подходят под такие короткие рабочие интервалы, поэтому их можно увеличить, сделав меньше общее количество подходов до большого перерыва.

В основе *frontend* части приложения лежит фреймворк *React*. Это инструмент создания пользовательских интерфейсов, задачей которого является обеспечение вывода на экран интерфейса для взаимодействия с пользователем.

Основной является разбиение интерфейса на небольшие переиспользуемые компоненты.

Плюсами использования данного инструмента является:

- технология *virtual DOM* для повышения производительности приложения за счет уменьшения количества необоснованной перерисовки интерфейса;

- использование изоморфного подхода для использования кода на стороне клиента или сервера;

- простой перенос компонентов *web*-сервиса на мобильные устройства.

В основе *backend* части приложения лежит платформа *NodeJS*.

Изначально язык *JavaScript* создавался компанией *Mozilla* в качестве простого скриптового языка для оживления интернет-страниц, но со временем разработчики начали экспериментировать с применением данного языка.

На данный момент *NodeJS* является программной платформой, которая использует язык *JavaScript* для создания серверной части приложений. В отличие от таких языков, как *PHP*, платформе *NodeJS* не нужна обвязка в виде стороннего сервера, хотя и желательная для целей кеширования и перенаправления запросов.

Основой этого инструмента является движок *V8*, изначально разработанный *Google* для интерпретации кода *JavaScript*.

В качестве главной особенности можно выделить событийно-ориентированный подход к параллелизму, а не классический параллелизм. По этой причине данная платформа часто используется в приложениях, где присутствует интенсивный ввод и вывод. Например, чатов или программ для коммуникации.

В качестве базы данных используется *MongoDB*. Это документоориентированная СУБД с открытым кодом. Для хранения данных используется формат, подобный *JSON*. В качестве особенностей можно выделить высокую доступность и масштабируемость.

Основными особенностями являются:

- принадлежность к NoSQL типу баз данных;
- нет необходимости описывать схемы таблиц, как это делается в реляционных базах данных;
- формат хранения данных называется *BSON*. Можно расшифровать как бинарный *JSON* формат;
- между коллекциями отсутствуют сложные связи как в реляционных базах.

Основные преимущества данной базы данных:

- документоориентированность. Сохранения данных в формате документов делает базу гибкой и легко адаптируемой к изменяющимся требованиям бизнеса;
- специальные запросы, в том числе с помощью регулярных выражений;
- опциональная индексация для увеличения скорости выполнения запросов;
- балансирование нагрузки, основанная на горизонтальном принципе;
- простая развертка в облаке и широкая доступность многими языками программирования.

Выбранный технологический стек позволяет создавать современные, безопасные и высокопроизводительные *web*-приложения.

В третьей главе произведен краткий обзор приложения *Figma*, которое использовалось для прототипирования интерфейсов. Это кроссплатформенный сервис, который позволяет быстро создать макеты интерфейсов для последующей разработки.

Центральной частью интерфейса является таймер и система контроля интервалов. Под ней расположен блок со списком задач, который позволяет наблюдать за общим прогрессом выполнения дел.

Основным принципом, лежащим в основе архитектуры на уровне кода, является *SOLID*.

Принципы *SOLID* – это стандарт программирования, который все разработчики должны хорошо понимать, чтобы избегать создания плохой архитектуры. Этот стандарт широко используется в ООП. Если применять его правильно, он делает код более расширяемым, логичным и читабельным. Когда разработчик создаёт приложение, руководствуясь плохой архитектурой, код получается негибким, даже небольшие изменения в нём могут привести к багам. Поэтому нужно следовать принципам *SOLID*.

Оценим скорость загрузки приложения в зависимости от различных типов соединения. Данные по скорости загрузки сведены в таблицу 1.

Таблица 1 – Скорость загрузки в зависимости от типа соединения

Тип соединения	Скорость, Мб/с	Время загрузки, с
<i>LAN</i>	30	1.11
Быстрое 3G	20	2.64
Медленное 3G	8	5.8

Мобильное соединение значительно уступает подключению через *LAN*. По этой причине важно особо тщательно относиться к оптимизации размера пакетов, отправляемых сервером.

Сравним полученные результаты с данными конкурентов. За основу возьмем подключение с помощью мобильных устройств с медленным соединением. Результаты сравнения сведены в таблицу 2.

Таблица 2 – Скорости загрузки с конкурентами

Название сервиса	Время загрузки, с
<i>Todoist</i>	10.2
<i>Trello</i>	7.2
<i>Habitica</i>	15

Можно наблюдать, что *Habitica* показало худшие результаты по причине большого количества графических элементов, загружающихся при первом подключении. Два остальных сервиса незначительно хуже разработанного программного средства.

Todoist и *Trello* показали результаты несколько худшие результаты из-за отсутствия должной оптимизации и сжатия файлов, отправляемых с сервера. Для улучшения ситуации можно было бы использовать более качественные алгоритмы сжатия отправляемых данных. Так же можно обратить внимание на способы оптимизации кода *Javascript*. Особенность его в том, что в два раза больший объем занимает у браузера не только в два раза больше времени на получение самих данных, но и в несколько раз большее время для распаковки и интерпретации полученного кода.

Для комплексной оценки качества разработанного программного продукта использовался *Lighthouse*. Это автоматизированный инструмент, который можно запустить на любой *web*-страницу и получить отчет о том, насколько хорошо работает страница.

На рисунке 2 представлены категории, по которым производилось тестирование приложения. Выбраны все основные пункты:

- производительность;
- прогрессивные веб-приложения
- лучшие практики;

- доступность приложения;
- *SEO* оптимизация.

Все эти категории являются очень важными для современных сайтов так как поисковые системы при анализе сайта обращают внимание не только на то, какая информация представлена, но и то, как именно всё реализовано с технической стороны.

Результаты проведенного теста представлены на рисунке 1.

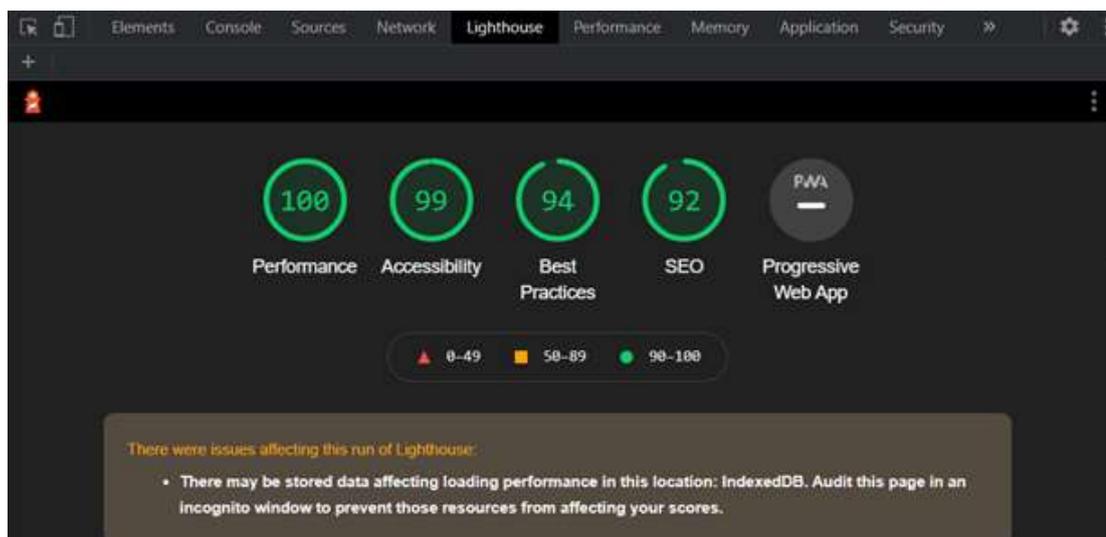


Рисунок 1 – Результаты тестирования приложения

Как видно по вышеизложенным данным, созданное приложение соответствует всем современным стандартам разработки *web*-приложений. Поддерживаются как оптимизации для качественной выдачи в поисковых системах, так и высокий уровень производительности и доступности приложения. Это значит, что сервис не будет терять пользователей по причине недостаточной качественности. В качестве дальнейших улучшений можно обратить внимание на те советы, которые были выданы тестовой системой *Lighthouse*. Они помогут сделать приложение ещё более отзывчивым и быстрым.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

1. Рассмотрены уже существующие аналоги, представленные на рынке. Дано их краткое описание и рассмотрены положительные и отрицательные стороны.
2. Разработано собственное программное средство с применением современных технологий и подходов к разработке высокоскоростных и нагрузоустойчивых *web*-приложений.
3. Проведен обзор полученного решения и его подробное тестирование

с помощью инструмента *Lighthouse* для *web*-приложений.

Рекомендации по практическому использованию результатов

Полученные результаты были внедрены в собственный процесс обучения и работы, где доказали свою эффективность. Рекомендуется обратить внимание на советы по улучшению, которые были получены при анализе страницы с помощью инструмента тестирования страниц *Lighthouse*.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1. Толстых, М.А. Оценка перспективности фреймворков для создания *WEB*-приложений / М. А. Толстых // Научные исследования XXI века. – 2020. №1. – С. 79-82.

2. Толстых, М.А. Концепция *PWA* в современной *WEB* разработке / М. А. Толстых // Научные исследования XXI века. – 2020. №2. – С. 78-81.

3. Толстых, М.А. Построение дизайн системы компонентов *React*. / М. А. Толстых // Внедрение передового опыта и практическое применение результатов инновационных исследований: материалы Международной научно-практической конференции, Волгоград, 20 мая 2020 г. / [редкол: Сукиасян А. А. (отв. ред.) и др.]. – Уфа: Аэтерна, 2020. – С. 41–43.

4. Толстых, М.А. Применение инструмента *Lighthouse* для анализа качественного состояния *web*-приложения. / М. А. Толстых // Концепция, теория и методика фундаментальных и прикладных научных исследований: материалы Международной научно-практической конференции, Екатеринбург, 5 февраля 2021 г. – Уфа: Аэтерна, 2021. – С. 67-69.

5. Толстых, М.А. Использование *HTTP 3* в современных *web*-приложениях. / М. А. Толстых // Концепция, теория и методика фундаментальных и прикладных научных исследований: материалы Международной научно-практической конференции, Екатеринбург, 5 февраля 2021 г. – Уфа: Аэтерна, 2021. – С. 69-72.

6. Толстых, М.А. Хранилище данных для *Web* приложений. / М. А. Толстых // Прорывные научные исследования как двигатель науки: сборник статей международной научно-практической конференции (Новосибирск, 05.03.2021 г.) / Уфа, 2021. – С. 72–74.

7. Толстых, М.А. Адаптивная загрузка *Web*-ресурсов. / М. А. Толстых // Прорывные научные исследования как двигатель науки: сборник статей международной научно-практической конференции, Новосибирск, 5 марта 2021 г. / Научно-издательский центр Аэтерна; редкол.: Г. Ш. Абидова [и др.]. – Уфа, 2021. – С. 72–74.

8. Толстых, М. А. Раннее установление соединения как способ сократить длительность *Web*-запроса / М. А. Толстых // Наука в современном обществе: закономерности и тенденции развития: сборник статей международной научно-практической конференции, Омск, 10 марта 2021 г. / Научно-издательский центр Аэтерна; редкол.: Г. Ш. Абидова [и др.]. – Уфа, 2021. – С.

23–24.

9. Климов, К. О. Приложение на Reactjs для обработки данных интернета вещей с использованием сервиса AWS = App on Reactjs for processing data on the internet of things using the AWS service / К. О. Климов, М. А. Толстых // Электронные системы и технологии: сборник материалов 57-й научной конференции аспирантов, магистрантов и студентов БГУИР, Минск, 19-23 апреля 2021 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Д. В. Лихаческий [и др.]. – Минск, 2021. – С. 610–611.

РЭЗІЮМЭ

Талстых Максім Андрэевіч

Сістэма кантролю выканання задач на аснове web-тэхналогій

Ключавыя словы: *web-сервіс, javascript, react, nodejs, mongodb.*

Мэта працы: стварэнне сістэмы кантролю выканання задач з унікальным падыходам у сваёй аснове, які зможа задаволіць свабодную нішу на рынку праграмага забеспячэння падобнага роду.

Атрыманыя вынікі і іх навізна: выкананы аналіз асноўных метадаў, якія ляжаць у аснове рашэння задач прадуктыўнасці. Прааналізаваны асноўныя праграмныя прадукты на аснове web-сэрвісаў, прадстаўленых на рынку. Вылучаны перавагі і недахопы дадзеных рашэнняў. У ходзе аналізу падобных прадуктаў вылучана канцэпцыя для распрацоўкі прыкладання, якая магла б перакрыць недахопы ўжо існуючых прыкладанняў і знайсці сваё месца на рынку. Распрацаваны web-сэрвіс і пратэставаны дапамогай сістэмы кантролю якасці Lighthouse. Па выніках атрыманы рэкамендацыі, якія стануць асновай для будучых паляпшэнняў прыкладання.

Ступень выкарыстання: вынікі ўкаранёны ва ўласны навучальны і рабочы працэс. Па выніках ўласнага тэставання выяўленыя тэхнічныя недахопы ў кодзе праграмы, якія будуць выпраўляць паралельна ўкараненню новых функцый.

Вобласць ужывання: навучальны працэс, працоўны працэс, любыя сферы дзейнасці чалавека, звязаныя з доўгім выкананнем складаных задач.

РЕЗЮМЕ

Толстых Максим Андреевич

Система контроля выполнения задач на основе *web*-технологий

Ключевые слова: *web*-сервис, *javascript*, *react*, *nodejs*, *mongodb*.

Цель работы: создание системы контроля выполнения задач с уникальным подходом в своей основе, который сможет удовлетворить свободную нишу на рынке программного обеспечения подобного рода.

Полученные результаты и их новизна: выполнен анализ основных методов, лежащих в основе решения задач продуктивности. Проанализированы основные программные продукты на основе *web*-сервисов, представленных на рынке. Выделены преимущества и недостатки данных решений. В ходе анализа схожих продуктов выдвинута концепция для разработки приложения, которая могла бы перекрыть недостатки уже существующих приложений и найти свое место на рынке. Разработан *web*-сервис и протестирован помощью системы контроля качества *Lighthouse*. По результатам получены рекомендации, которые станут основой для будущих улучшений приложения.

Степень использования: результаты внедрены в собственный учебный и рабочий процесс. По результатам собственного тестирования выявлены технические недостатки в коде программы, которые будут исправлять параллельно внедрению новых функций.

Область применения: учебный процесс, рабочий процесс, любые сферы деятельности человека, связанные с длительным выполнением сложных задач.

SUMMARY

Talstykh Maksim Andreevich

Web-based task execution control system

Keywords: *web-service, javascript, react, nodejs, mongodb.*

The object of study: creation of a task execution control system with a unique approach at its core, which can satisfy a free niche in the market for software of this kind.

The results and novelty: the analysis of the main methods underlying the solution of productivity problems has been carried out. The main software products based on web services on the market are analyzed. The advantages and disadvantages of these solutions are highlighted. During the analysis of similar products, a concept was put forward for developing an application that could overcome the shortcomings of existing applications and find its place in the market. A web service was developed and tested using the Lighthouse quality control system. Based on the results, recommendations were obtained that will become the basis for future improvements to the application.

Degree of use: the results are embedded in our own educational and workflow. Based on the results of our own testing, we identified technical flaws in the program code, which will be corrected in parallel with the introduction of new functions.

Sphere of application: educational process, work process, any areas of human activity associated with the long-term performance of complex tasks.