Министерство науки и высшего образования РФ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рыбинский государственный авиационный технический университет

имени П.А. Соловьева»

Факультет радиоэлектроники и информатики

Кафедра МПО ЭВС

**выпускная квалификационная работа**

Разработка системы управления интеллектуальной роботизированной модульной платформой.

на соискание степени бакалавр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по направлению 09.03.04 Программная инженерия

профиль бакалавриата: Разработка программно-информационных систем\_\_\_\_\_\_\_\_\_\_

**Пояснительная записка**

Соискатель, студент группы ИПБ-19 Ювченко Д.А.

*(Код) (Подпись, дата ) (Фамилия И.О.)*

Руководитель к.ф.-м.н., профессор Паламарь И.Н

*(Уч. степень, звание) (Подпись, дата) (Фамилия И.О.)*

Консультант

по экономике к.т.н., доцент Клементьева Н.А.

*(Уч. степень, звание) (Подпись, дата) (Фамилия И.О.)*

Нормоконтролер ст.преп. Задорина Н.А.

*(Уч. степень, звание) (Подпись, дата) (Фамилия И.О.)*

К защите допустить

Зав. кафедрой к.ф.-м.н., профессор Паламарь И.Н.

*(Уч. степень, звание) (Подпись, дата) (Фамилия И.О.)*

ВКР передана в ГЭК «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Секретарь ГЭК ст.преп. Задорина Н.А.

*(Подпись) (Фамилия И.О.)*

Рыбинск 2023

Оглавление

[Введение 11](#_Toc74660749)

[1 Описание проблемы 12](#_Toc74660750)

[1.1 Обзор предметной области 12](#_Toc74660751)

[1.2 Формулировка проблемы 16](#_Toc74660752)

[1.3 Постановка задачи 17](#_Toc74660753)

[2 Обзор аналогов 19](#_Toc74660754)

[2.1 Определение критериев анализа 19](#_Toc74660755)

[2.2 Сравнительная характеристика существующих разработок 20](#_Toc74660756)

[2.3 IBM Rational Functional Tester 21](#_Toc74660757)

[2.4 HP QuickTest Professional 21](#_Toc74660758)

[2.5 Сравнение существующих систем 22](#_Toc74660759)

[3 Программная документация 24](#_Toc74660760)

[3.1 Техническое задание на программное обеспечение 24](#_Toc74660761)

[3.1.1 Общие сведения 24](#_Toc74660762)

[3.1.2 Назначение и цели создания системы 25](#_Toc74660763)

[3.1.3 Требования к системе 25](#_Toc74660764)

[3.2 Пояснительная записка к программному обеспечению 28](#_Toc74660765)

[3.2.1 Анализ предметной области 28](#_Toc74660766)

[3.2.2 Формулировка проблемы 28](#_Toc74660767)

[3.2.3 Обзор аналогов 28](#_Toc74660768)

[3.2.4 Назначение и область применения 28](#_Toc74660769)

[3.2.5 Технические характеристики 28](#_Toc74660770)

[3.3 Описание программы 37](#_Toc74660771)

[3.3.1 Общие сведения 37](#_Toc74660772)

[3.3.2 Функциональное назначение 38](#_Toc74660773)

[3.3.3 Описание логической структуры 38](#_Toc74660774)

[3.3.4 Используемые технические средства 53](#_Toc74660775)

[3.4 Программа и методика испытаний 55](#_Toc74660776)

[3.4.1 Объект испытаний 55](#_Toc74660777)

[3.4.2 Цель испытаний 55](#_Toc74660778)

[3.4.3 Требования к программе 55](#_Toc74660779)

[3.4.4 Описание испытаний 55](#_Toc74660780)

[4 Эксплуатационная документация на программный продукт 59](#_Toc74660781)

[4.1 Руководство системного администратора 59](#_Toc74660782)

[4.1.1 Общие сведения о программе 59](#_Toc74660783)

[4.1.2 Системные требования 59](#_Toc74660784)

[4.1.3 Установка программы 60](#_Toc74660785)

[4.1.4 Файл конфигурации. Составление и правка 60](#_Toc74660786)

[5 Акт испытаний программного продукта 61](#_Toc74660787)

[6 Экономическое обоснование 62](#_Toc74660788)

[6.1 Экономическое обоснование разрабатываемого программного обеспечения 62](#_Toc74660789)

[6.2 Сравнение с аналогами 62](#_Toc74660790)

[6.3 Описание функций, назначения и потенциальных пользователей программного обеспечений 63](#_Toc74660791)

[6.3.1 Назначение программного обеспечения 63](#_Toc74660792)

[6.3.2 Функции программного обеспечения 63](#_Toc74660793)

[6.3.3 Потенциальные пользователи программного обеспечения 64](#_Toc74660794)

[6.4 Расчет затрат на разработку программного обеспечения 64](#_Toc74660795)

[6.5 Оценка результата от использования ПО 69](#_Toc74660796)

[Заключение 73](#_Toc74660797)

[Список литературы 74](#_Toc74660798)

# Введение

В современном мире всё больше и больше идёт процесс автоматизации. Уже на данный момент роботы широко распространены на производствах и получают всё большее распространение на складах.

В современном мире у человека становиться всё меньше и меньше времени на отдых или занятие своими увлечениями. Имея дачный участок человеку требуется много времени тратить на рутинную работу вроде кошения газона и полива деревьев или уборку снега зимой и т.п. А если участок только застраивается, то нередко много времени тратиться на подвоз нужного материала к месту стройки с другого конца участка, или на вывоз строительного мусора.

Безусловно, существуют роботы способные помочь с этим, но в основном, они узкоспециализированные и стоят достаточно много.

Нельзя не забыть, что существуют и профессии с рутинной работой или делами, которые не хотелось бы выполнять вручную, к примеру, это уборка улиц и очищение урн или, к примеру, работу ночного охранника.

# Описание проблемы

## Основная концепция

В основе интеллектуальной роботизированной модульной платформы лежит несколько основных идей:

* Возможность выполнения широкого круга задач, благодаря установке различных модулей;
* Возможность применения платформы как под контролем пользователя, так и в автономном режим;
* Простота в управлении и обслуживании;
* Дешевизна, простота и доступность составляющих платформы.

Предполагается, что благодаря установке различных модулей, возможно будет использовать платформу для решения таких типовых задач как:

* Кошение травы;
* Наблюдение за территорией;
* Мелкое пожаротушение;
* Уборка улиц;
* Развлечения;
* Перевозка грузов;
* Полив растений;
* Опрыскивание территории.

Предполагается, что благодаря комбинации различных модулей, может формироваться уникальный набор возможностей использования платформы.

## Выбор предметных областей

Очевидно, что предполагаемый перечень задач платформы находится в разных предметных областях и каждая из них требует обзора и описания. В следствии большой трудоёмкости и сложности предполагаемой задачи, было выбрано только две основные области, а именно область позиционирования и область наблюдения за территорией.

Область позиционирования была выбрана как основная, поскольку без этого платформа не сможет выполнять другие задачи. Благодаря точному позиционированию становится возможно выполнения множества действий, к примеру, полив растений и кошение газонов. Вся работа платформы строится на её позиционировании.

Область наблюдения за территорией была выбрана как одна из наиболее интересных и предположительно востребованных, а также для демонстрации возможностей модульности платформы.

## Описание предметных областей

Область позиционирования – достаточно обширная, в зависимости от задач её можно рассматривать в разных ключах.

Если упрощённо рассматривать полёты беспилотных летательных аппаратов по маршрутам, для успешного полёта по точкам будет достаточно иметь требуемые координаты и в большинстве случаев, точности получения своих GPS координат будет достаточно для выполнения задачи.

Однако для движения автомобиля одними только GPS координатами не обойтись. В отличии от беспилотных летательных аппаратов, на земле препятствия куда более вероятны, чем в воздухе, в частности люди и другие машины, в следствии чего просто двигаться по GPS координатам не получиться. В целом в зависимости от задач можно реализовать движение по земле только с помощью GPS и каких-либо устройств определения препятствий, однако для некоторых задач этого может оказаться недостаточно. Препятствия бывают разными, в частности можно сказать, что препятствием также могут являться деревья, плохое дорожное покрытие, грязь, вода и тому подобное. Очевидно, что видов препятствий может быть огромное количество, и для беспроблемного движения необходимо учитывать их все. Также нельзя забывать, что для разных задач, препятствия могут быть разимыми, к примеру болото будет непроходимым препятствием для автомобиля, однако для болотохода оно уже не будет таковым считаться.

Наблюдение за территорией — это очень обширная область, в которой можно выделить подобласти, в зависимости от конкретного использования платформы.

Выделим следующие подобласти:

* Наблюдение за территорией объекта (к примеру, завода или дачного участка),
* Наблюдение за конкретными, возможно, специфичными объектами.

К первой относится наблюдение за нарушение периметра, а ко второй можно отнести наблюдение внутри дома за его состоянием (к примеру, не произошло ли пожара, не потекли ли трубы и тому подобное).

Для наблюдения за территорией обычно используются камеры или-же вручную происходит обход территории. В данной области существуют такие проблемы как:

* человеческий фактор (человек может попросту заснуть или отвлечься),
* недостаточность покрытия камер,
* большое время, требуемое на обход территории

Вторая область несколько специфичней и на неё обычно не акцентируют внимание. К проблемам можно отнести:

* Отсутствие наблюдения
* Недостаточное внимание
* Неспособность увидеть (к примеру, утечка газа)

## Постановка задачи

Необходимо разработать систему управления интеллектуальной роботизированной модульной платформой. Данная система должна обеспечить возможность установки и эффективной работы различных модулей.

Ключевые особенности разрабатываемой системы:

* Возможность установки различных модулей на платформу. Система управления должна поддерживать установку различных модулей и необходимого ПО для их работы;
* Удобный интерфейс для работы пользователя. Одним из вариантов общения пользователя с платформой должен быть общение через браузер с помощью сайта, без необходимости дополнительной установки каких-либо программ. Также сайт должен быть адаптирован для мобильных устройств.
* Возможность автономной работы. Должна быть возможность настройки автоматической работы платформы, в частности возможность указания необходимых действия, предоставляемых модулями, для их исполнения;
* Возможность ручного управления платформой и установленными модулями. Помимо возможности отдавать типовые команды на исполнения платформой должна быть возможность выполнения их в ручном режиме;
* Простое добавление новых модулей. Для добавления нового модуля не должна перестраиваться и переписываться вся система;
* Возможность отслеживания действий платформы. Действия платформы должны быть отражены на сейте. Должна быть возможность аварийной остановки выполнения действий;
* Способность комбинирования возможностей модулей. Должна быть возможность выполнять команды различных модулей, в рамках одной заданной пользователем программы действий.

# Обзор аналогов

Нужно отметить что похожих систем удалось найти достаточно малое количество, в основном имеются строго типизированные комплексы, заточенные под определённые задачи. Безусловно, похожие роботизированные комплексы тоже имеются, однако о них достаточно мало информации для полноценного сопоставления.

Чтобы более конкретно сравнить экземпляры необходимо выделить чёткие критерии анализа.

## Определение критериев сравнения

В качестве основных критериев сравнения роботизированных комплексов выбраны следующие:

* Основное назначение. Под основным назначением понимается главная задача робота, то есть к примеру робот снегоуборщик с камерой заточен под уборку снега, а вот использование дистанционного управления и видеонаблюдения с помощью этой функциональности это уже второстепенные возможности;
* Возможности. Под возможностями будем понимать любые полезные задачи, которые робот может выполнять;
* Тип шасси. В данном критерии будет рассматриваться проходимость платформы;
* Условия работы. Под условиями работы будем понимать климатические условия эксплуатации робота, включая температуру и поверхность по которой будет происходить движение.
* Время автономной работы;
* Стоимость.

## Сравнительная характеристика существующих разработок

В данном разделе представлено сравнение разрабатываемой платформы с существующими решениями в данной области по критериям, выбранным и описанным в предыдущем пункте.

Для начала рассмотрим типовых представителей этой сферы, все они достаточно узкоспециализированы.

Робот уборщик Пиксель. Основные назначение – очистка улиц от грязи и влажная убора, однако в перспективе планируется и очистка от снега. Существует возможность самостоятельной установки навесного оборудования для уборки. Шасси полно приводное, однако предназначено для движения только по городу. Время автономной работы составляет около 16 часов. Ограничений по условиям работы не найдено, или не заявлено. Стоимость не указана.

OMI Plow снегоуборщик. Это классический автономный снегоуборщик на гусеничном ходу. Заявлено, что может работать до -50 градусов. Время автономной работы 8 часов. Стоит примерно 180 000 рублей.

Робот-охранник Трал Патруль. Данный робот предназначен для патрулирования территории и обнаружения на ней посторонних людей. Также имеется возможность удалённого управления роботом и задания маршрута. К числу особенностей можно добавить необходимость робота в WI-FI покрытии для общения с пользователем. Нельзя не отметить и повышенную проходимость (в отличии от ранее рассмотренных экземпляров) и возможность автономной работы с записью видео с камер на встроенный накопитель. Стоимость робота в зависимости от версии варьируется от миллиона до полутора миллионов рублей.

Однако рассмотренные выше роботы не совсем подходят для сравнения, поскольку они всё же узконаправленные, в отличии от планируемого комплекса. По этому их следует рассмотреть с более универсальными платформами, а именно с платформой «Автономный ровер» и «Роботехнический комплекс МАРКЕР». Сравнение с ними приведено в таблице 2.1.

Таблица 2.1 – Сравнение с аналогами

|  |  |  |  |
| --- | --- | --- | --- |
| Категория / Продукт | Разрабатываемая платформа | Робототехнический  комплекс МАРКЕР | Автономный ровер DH:FRAME |
| Основное назначение | модульная платформа | Патрулирование, огневое поражение противника | Модульная платформа |
| Наиболее значимые возможности | Возможность установки различных модулей, удобное управление с помощью веб страниц, автономная работа, возможность ручного управления | Противодействие проникновению, противодействие БПЛ, автономное патрулирование, возможность работы с группой, широкий спектр возможностей по распознаванию и навигации | Автономное передвижение, трансляция видео и телеметрии, возможность установки дополнительного бортового и навесного оборудования |
| Тип шасси | Полно приводное, колёсное, 4 x 4 | Есть варианты с колёсной формулой 6 x 6, так-же имеется гусеничное исполнение | Полноприводное, колёсное шасси с формулой 4 x 4 |
| Условия работы | Допускается работа при температуре от + 40, до - 10 | Точно не известно | Нет данных |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Время автономной работы | Предполагаемое время работы 8 часов | Запас хода около 600 – 1000 километров | В зависимости от аккумуляторной батареи, в среднем 20 км. |
| Стоимость | Не известна | Не известна | От 360 000 |

## Робототехнический комплекс «Маркер»

Робототехнический комплекс «Маркер» это проект НПО «Андроидная техника» и фонда перспективных технологий. Он представляет собой модульную робототехническую платформу, которая позволяет выполнять большой спектр задач.

Можно выделить следующие основные особенности:

* Модульное исполнение;
* Автономное управление и обработка информации;
* Наличие системы группового управления;
* Наличие системы технического зрения.

Данный проект на текущем этапе разработки предназначен для решения военных задач. В число основных зада входит патрулирование территории и устранение угроз, как наземных, так и воздушных.

Продвинутые датчики и системы обработки данных позволяют выполнять сложнейшие задачи по патрулированию в сложных метеоусловиях, в условиях отсутствия дорог и под вражеским огнём. Также нельзя забывать про возможности автоматической диагностики и осмотра объектов.

Очевидно, что столь многофункциональная платформа, с применением военных технологий, сложна как в производстве, так и в эксплуатации.

## Автономный ровер

Автономный ровер – многофункциональный ровер способный выполнять большой спектр задач, в зависимости от установленного на него навесного оборудования.

Ровер имеет возможность передвигаться не только по ровной дороге, но и по лёгкому бездорожью.

Также заявлено, что он может выполнять такие функции как:

* перевозки тяжелых грузов;
* буксировки различной техники и прицепов;
* патрулирования охраняемых объектов;
* уборки территории;
* распыления химикатов и обработки посевов;
* обследование труднодоступных мест;
* выполнение работ с установленным оборудованием.

Для навигации ровер использует GPS, карты местности, визуальные маркеры и специальную разметку.

Ровер подходит для посёлков, агрохолдингов, промышленных предприятий, складов, охранения территории и выполнения задач по доставке грузов.

## Сравнение существующих систем

В результате проведённого анализа существующих аналогов было выявлено, что имеющиеся на рынке системы в основном предназначены для конкретных задач, универсальных систем фактически нет. А те, которые имеются имеют свои недостатки. К основным недостатка можно отнести:

* Дороговизна покупки;
* Дороговизна обслуживания;
* Относительная мало функциональность у дешёвых платформ;
* Сложность взаимодействия с платформой;
* Требовательность к обслуживающему персоналу.

# Программная документация

## Техническое задание на программное обеспечение

### Назначение разработки

#### Функциональное назначение

Функциональным назначением программы является управление интеллектуальной модульной платформой.

#### Эксплуатационное назначение

Программа должна использоваться в управлении интеллектуальной модульной платформой для решения поставленных перед платформой задач. Список доступных задач зависит от установленных на платформу модулей. К базовых задачам, стоящим перед платформой, относятся задачи по наблюдению за территорией.

Пользователями платформы могут являться как частные лица, так и компании.

### Требование к программе или программному изделию

#### Требования к функциональным характеристика

##### Требования к составу выполняемых функций

Программа должна обеспечивать возможность выполнения интеллектуальной модульной платформой перечисленных ниже функций:

* Навигация с помощью GPS сигналов
* Патрулирование территории по установленному маршруту
* Самостоятельное патрулирование территории по указанным границам патрулирования
* Идентификация людей с помощью камеры
* Возможность добавление функций при установке новых модулей
* Ручное управление платформой, включая ручное управление движением и работой дополнительных модулей
* Возможность аварийной остановки платформы

##### Требования к организации входных данных

Входными данными, предоставляемыми пользователем являются некоторые управляющие команды, набор которых частично зависит от установленных модулей, а часть задана по умолчанию. Набор команд для выполнения функций определяется на этапе разработки.

Должно быть, несколько способов ввода команд программному продукту.

Первый способ – голосовой ввод непосредственно через микрофон платформы.

Второй способ – ввод команд или выполнение некоторых действий для автоматического формирования команд с помощью браузера.

Третей способ – голосовой ввод команд через браузер.

Так же входными данными являются видеопоток с камеры, данные о расстоянии в миллиметрах до объекта получаемые от лазерного сканера, и GPS координаты, получаемые от GPS модуля.

##### Требования к организации выходных данных

Выходными данными являются действия системы и интеллектуального роботизированного комплекса.

Также выходными данными являются синтезированные слова, которые необходимы для получения обратной связи или сообщения на web странице в браузере, если используется связь с его помощью.

#### Требования к надёжности

##### Требования к обеспечению надёжного (устойчивого) функционирования программы

В случае возникновения ошибок при работе программы и её аварийного завершения программа должна перезапуститься и сообщить пользователю об аварийном завершении работы. Выдав сообщение об ошибке на веб странице и звуковой сигнал.

##### Время восстановления после отказа

Время восстановления после отказа не должно превышать 10 минут.

##### Отказы из-за некорректных действий оператора

Отказы программы не должны допускаться вследствие некорректных действий оператора при взаимодействии с программой. Однако они возможны при взаимодействии оператора непосредственно с интеллектуальной роботизированной модульной платформой при её обслуживании. Во избежание возникновения отказов программы по указанной выше причине производить обслуживание платформы следует только подготовленному персоналу.

#### Условия эксплуатации

##### Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

##### Требования к видам обслуживания

Установка обновлений и программного обеспечения должна производиться при помощи веб страницы.

#### Требования к составу и параметрам технических средств

Программа должна быть развёрнута на устройстве со следующими минимальными характеристиками:

* процессор с тактовой частотой, МГц – 1500;
* оперативную память объемом, Мб - 4048;
* встроенная память объёмом, Гб - 16;
* Видеокамера с разрешением видеозаписи – 720х480.
* Модуль GPS
* Дальномер с минимальной дальностью обнаружения объектов – 10 см и точностью 1 см
* Модуль WIFI

#### Требования к информационной и программной совместимости.

Операционная система должна быть представлена версией Raspbian с версией ядра 6.1

Также должен быть установлен Python версии 3.8

#### Требования к информационным структурам и методам решения

Исходные коды программы должны быть реализованы на языке Python.

Приложение должно включать в себя веб сервер для работы через браузер и основную программу, которая выполняет приходящие её команды.

#### Требования к защите информации

Личные данные пользователя должны быть зашифрованы. Пароли должны храниться в соответствии со стандартом PBKDF2.

#### Специальные требования

Программа должна обеспечивать взаимодействие с пользователем (оператором) с помощью веб страниц браузера и при помощи голосового управления.

### Требование к программной документации

#### Предварительный состав программной документации

В состав программной документации должны входить:

* техническое задание;
* программа и методика испытаний;
* инструкция по эксплуатации;
* ведомость эксплуатационных документов;

### Стадии и этапы разработки

#### Стадии разработки

Разработка должна быть проведена в несколько стадии:

* Изучение проблемы
* Разработка экономического обоснования
* Выбор технологий и инструментальных средств
* Разработка технического задания
* Разработка методики испытаний
* Разработка и отладка программного обеспечения
* Разработка описания программного обеспечения
* Разработка эксплуатационной документации
* Испытания программного обеспечения

#### Этапы разработки

В начале работ над проектом на стадии «Изучение проблемы» должна быть изучена предметная область, в которой предстоит работать интеллектуальной роботизированной платформе, составлены требования к самой платформе и её аппаратной части. После этого должно быть дано примерное описание программного продукта.

Далее на стадии «Разработка экономического обоснования» необходимо провести экономический анализ и убедиться в экономической целесообразности разработки программного комплекса.

На следующей стадии «Выбор технологий и инструментальных средств» должны быть выбрани основные технологи и библиотеки используемы при разработке.

На стадии «Разработка технического задания» должно быть создано и согласовано техническое задание.

На следующей стадии «Разработка методики испытаний» необходимо разработать и согласовать методику испытаний программного продукта.

На стадии «Разработка и отладка программного обеспечения» необходимо непосредственно реализовать и отладить программный продукт.

На стадии «Разработка описания программного продукта» необходимо получить формальное описание разрабатываемого продукта.

На стадии «Разработка эксплуатационной документации» необходимо проработать документацию необходимую для качественного обучения пользователя работе с комплексом.

На последней стадии «Испытания программного обеспечения» необходимо провести тесты в соответствии с методикой тестирования.

#### Этапы разработки

На стадии «Изучение проблемы» должны быть произведены следующие работы:

* Изучение предметной области
* Составление и формализация описания предметной области
* Составление перечня основных функций программного продукта
* Составление перечня дополнительных, возможных функций программного продукта

На стадии «Разработка экономического обоснования» необходимо проанализировать возможный эффект от внедрения системы и рассчитать примерные затраты на разработку, чтобы убедиться, что разработки системы экономически оправдана.

На стадии «Выбор технологий и инструментальных средств» должны быть проанализированы возможные основные технологии и выбраны необходимые для разработки комплекса. Также должны быть определены языки программирования и инструментарий разработки.

На стадии «Разработка технического задания» должен быть реализованы следующие работы:

* Сформулирована постановка задачи;
* Определены требования к техническим средствам;
* Определены требования к программе;
* Определены сроки разработки
* Определены стадии и этапы разработки программы
* Определён набор документации на программу;
* Выбраны языки программирования;

На стадии «Разработка методики испытаний» должна быть составлены и согласованы наборы модульных, интеграционных и системных тестов.

На стадии «Разработка и отладка программного обеспечения» предполагается выполнение ниже перечисленных работ:

* Проработка архитектуры разрабатываемой системы;
* Разработка, кодирование и демонстрация прототипа программы
* Разработка, кодирование и отладка основной программы

На стадии «Разработка описания программного продукта» должно быть разработано, составлено в формализованном виде описание продукта, включающее описание основных алгоритмов работы программы, входных и выходных данных, а также требования к ним.

На стадии «Разработка эксплуатационной документации» должны быть разработана и согласована основная документация необходимая для эксплуатации программного продукта.

На стадии «Испытания программного обеспечения» должны быть проведены тесты в соответствии с методикой тестирования разработанной на одноимённом этапе.

### Порядок контроля и приёмки

#### Виды испытаний

Приемосдаточные испытания могут проводиться на ОС linux, при моделировании подключений к нему необходимых датчиков для работы системы или на прототипе интеллектуальной модульной системы. Испытания должны проводиться в сроки 1.06.2023 - 10.06.2023

Приёмо-сдаточные испытания должны проводиться по методике, разработанной на стадии «разработка методики испытания»

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

##### Проверка навигации с помощью GPS

1. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
2. В соответствии с инструкцией выполнить ввод GPS координат и выполнить команду
3. Убедиться, что роботизированная платформа двигается к указанным GPS координатам или если используется эмуляция, то убедиться, что система выдаёт корректные команды на движение.

##### Проверка патрулирования территории по установленному маршруту

1. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
2. В соответствии с инструкцией по эксплуатации ввести маршрут патрулирования
3. Убедиться, что роботизированная платформа двигается по указанному маршруту или если используется эмуляция, то убедиться, что система выдаёт корректные команды на движение.

##### Проверка самостоятельного патрулирования территории по указанным границам

1. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
2. В соответствии с инструкцией по эксплуатации ввести границы патрулирования
3. Убедиться, что роботизированная платформа двигается внутри границ и оценить покрытие территории патрулирования или если используется эмуляция, то убедиться, что система выдаёт корректные команды на движение.

##### Проверка идентификации людей с помощью камеры

1. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
2. В соответствии с инструкцией по эксплуатации добавить себя в базу знаний платформы о людях.
3. Запустить какой-либо режим патрулирования
4. Поставить перед платформой в момент патрулирования неизвестного человека
5. Убедиться, что платформа в соответствии с инструкцией не идентифицировала человека
6. Поставить перед платформой в момент патрулирования известного человека
7. Убедиться, что платформа в соответствии с инструкцией идентифицировала человека

##### Проверка возможности добавления функций при установке новых модулей

1. Установить поддерживаемый модуль в соответствии с инструкцией по эксплуатации
2. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
3. В соответствии с инструкцией по эксплуатации убедиться, что появилась возможность управления дополнительными модулями
4. Выполнить некоторые команды по управления новым модулем
5. Убедиться, что модуль активен и работает или если используется эмуляция, то убедиться, что система корректно взаимодействует с модулем

##### Проверка ручного управления платформой, включая ручное управление движение и работой дополнительных модулей.

1. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
2. В соответствии с инструкцией по эксплуатации выполнить ручное задание команд платформе на движение
3. Убедиться, что платформа движется или если используется эмуляция, то убедиться, что система выдаёт корректные команды.
4. В соответствии с инструкцией по эксплуатации выполнить остальные доступные команды ручного управления
5. Убедиться в исполнении платформой команд или если используется эмуляция, то убедиться, что система выдаёт корректные команды.

##### Проверка возможности аварийной остановки платформы

1. В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой
2. В соответствии с инструкцией по эксплуатации начать выполнение команд платформой
3. В момент выполнения, в соответствии с инструкцией по эксплуатации выполнить аварийную остановку платформы
4. Убедиться, что платформа перестала выполнять свои действия

#### Общие требования к приемке работы

На основании протокола испытаний исполнитель совместно с заказчиком подписывает акт приемки-сдачи программы в эксплуатацию.

##### Проверка навигации с помощью GPS

- Роботизированная платформа двигается к указанным GPS координатам или если используется эмуляция, то система выдаёт корректные команды на движение

##### Проверка патрулирования территории по установленному маршруту

- Роботизированная платформа двигается по указанному маршруту или если используется эмуляция, то система выдаёт корректные команды на движение.

##### Проверка самостоятельного патрулирования территории по указанным границам

- Роботизированная платформа двигается внутри границ территории патрулирования или если используется эмуляция, то система выдаёт корректные команды на движение.

##### Проверка идентификации людей с помощью камеры

- Роботизированная платформа корректно идентифицирует людей с помощью камеры

##### Проверка возможности добавления функций при установке новых модулей

- Модуль активен и работает или если используется эмуляция, то система корректно взаимодействует с модулем

##### Проверка ручного управления платформой, включая ручное управление движением и работой дополнительных модулей

- Платформа корректно исполняет команды или если используется эмуляция, то система выдаёт корректные команды.

##### Проверка возможности аварийной остановки платформы

- Платформа перестала выполнять свои действия.

## Пояснительная записка к программному обеспечению

### Анализ предметной области

Анализ предметной области приведен в разделе 1.3.

### Формулировка проблемы

Формулировка проблемы выведена в разделе 2.5.

### Обзор аналогов

Обзор аналогов приведён в разделе 2.5.

### Назначение и область применения

Назначение платформы приведено в подразделе 3.1.1 технического задания, а область применения описана в разделе 1.1.

### Технические характеристики

#### Постановка задачи на разработку программы

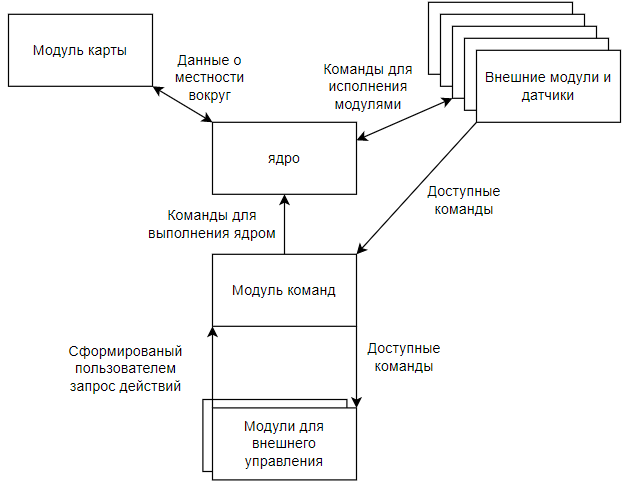
Постановка задачи представлена в разделе 1.4.

#### Описание алгоритма и функционирования программы

Всю программу можно разбить на несколько основных модулей:

* Модуль карты
* Ядро
* Внешние модули и датчики
* Модуль команд
* Модули для внешнего управления

Схема работы приложения приведена на рисунке 3.1.

Рисунок 3.1 Схема работы системы

Внешние модули и датчики могут общаться с навесным оборудованием, в том числе и датчиками. Предполагается, что они хранят в себе информацию и алгоритмы действия для решения определённых задач, в соответствии со своим назначением. Список доступных команд и некоторую дополнительную информацию они передают в модуль команд, а тот в свою очередь передаёт полный список команд в модули внешнего управления, где они становятся доступными для использования пользователем.

Модуль команд хранит информацию о доступных командах, условиях их выполнения и не только. Он служит промежуточным звеном между внешними модулями, модулями для внешнего управления и ядром. Это позволяет упростить добавление новых внешних модулей и взаимодействие с ними.

Внешние модули представляю из себя два основных модуля, а именно модуль для работы веб сервера и модуля голосового управления. В частности, благодаря им осуществляется взаимодействия пользователя и системы. Внешние модули управления формируют команду для исполнения ядром.

Через ядро проходят все команды перед их исполнениями внешними модулями. Именно ядро отслеживает необходимые условия для выполнения той или иной команды. Ядро взаимодействует с модулем карты для определения положения платформы, поиска необходимых объектов на карте и построения маршрутов. Также благодаря тому, что через ядро проходят все команды, оно может добавлять на карту некоторую информацию, которая необходима для работы модулей.

После достижения необходимых условий для выполнения действий внешними модулями, ядро выдаёт команды на действия непосредственно внешним модулям.

#### Описание и обоснование выбора состава технических и программных средств

Разработка самой системы автоматизированного тестирования выполнялась на языке программирования Python 3.8. поскольку он достаточно прост в освоении и имеет множество библиотек для решения различных задач (и как следствие разработка дополнительных модулей будет несколько проще).

Для создания веб приложения для взаимодействия пользователя с платформой используется Фреймворк Flask, благодаря простоте использования и функциональности.

#### Ожидаемые техника-экономические показатели

Представлены в пункте 6.5

## Описание программы

### Общие сведения

#### Обозначение и наименование программы

Краткое наименование программы: IRMP.

Краткое наименование программы на русском: ИРМП.

Полное наименование программы: IRMP Managed System.

Полное наименование программы на русском: Система управление интеллектуальной модульной платформой.

#### Программное обеспечение, необходимое для функционирования программы

Для обеспечения работы программы необходима система Respbian с версией ядра 6.1 и Python версии 3.8.

Также для обеспечения работы, во время настройки системы согласно инструкции по эксплуатации необходим доступ в интернет, для автоматической загрузки библиотек, необходимых для работы системы и внешних модулей.

#### Языки программирования, на которых написана программа

Основная программа написана на языке Python. Для работы с web страницами также использовался JavaScript, HTML, CSS и язык шаблонизатора jinja2.

### Функциональное назначение

#### Классы решаемых задач

Классы решаемых задач определяются модулями, установленными на платформу. Основной предполагаемый перечень задач, решаемых платформой указан в пункте 1.1. Непосредственно система управления интеллектуальной роботизированной модульной платформой отвечает за взаимодействие с внешними модулями и предоставления взаимодействия пользователя с ними.

#### Назначение программы

Разрабатываемый программный продукт позволит интеллектуальной модульной платформе выполнять поставленные ей задачи.

### Описание логической структуры

#### Алгоритм программы

Алгоритм работы программы приводится в пояснительной записке раздел «3.2.5.2 Описание алгоритма и функционирования программы».

#### Разработанные модули

##### Модуль «additionalModules»

Этот модуль, содержит в себе набор модулей, которые выполняют различные функции, в частности работают с установленными на платформу модулями и управляют системами движения.

Также сюда добавляются модули сторонних разработчиков.

###### Класс «HelperForConnectionWithExternalModules»

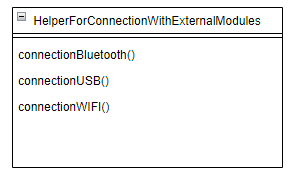


Рисунок .1. – HelperForConnectionWithExternalModules

Содержит следующие методы:

* connectionBluetooth(name, key) – подключается к устройству по Bluetooth;
* connectionUSB(name) – подключается к устройству по usb;
* connectionWIFI(nameWIFI, key) – подключается к устройству по WIFI.

###### Модуль «patrolling»

Внешний модуль для работы с патрулированием территории.

Файл «main»

Основной файл модуля, который необходим для того, чтобы система поняла, что существует модуль patrolling и подключила его

Файл «PatrolBlueperint»

Предоставляет собой endpoind для Flask, чтобы предоставить возможности модуля на web странице управления роботом.

##### Модуль «core»

Основной модуль, через него проходят основные потоки данных. Он также отвечает за управление внешними модулями. Именно он говорит, когда можно выполнять то или иное действие, которое хочет выполнить внешний модуль.

###### Класс «core»

Основной класс модуля core. Здесь происходит обработка данных, проходящих через него.

##### Модуль «db»

Модуль для работы с базой данных SQLite

###### Класс «DataBaseHelper»

Класс для упрощения работы с БД.

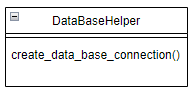


Рисунок 3.2 – DataBaseHelper

Содержит следующие методы:

Create\_data\_base\_connection()

###### Класс «DataBaseLauncher»

Запускает ядро БД и генерирует метаданные

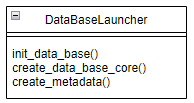


Рисунок 3.3 – DataBaseLauncher

Содержит следующие методы:

* + - init\_data\_base() – вызывает остальные методы инициализации БД
    - create\_data\_base\_core() – создаёт «ядро» для работы sqlAlchemy
    - create\_metadata() – создаёт таблицы в БД

###### Класс «DataBaseObject»

Представляет собой базовую сущность в БД. Необходим для ORM.

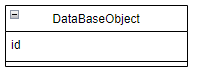


Рисунок 3.4 – DataBaseObject

Содержит следующие поля:

* + - id – Представляет собой id объектов

##### Модуль «externalControllers»

Модуль для управления пользователем системой

###### Модуль «webServer»

Представляет собой веб сервер flask и классы для его работы, а также классы для работы с ним.

##### Модуль «instruction»

Через него проходит работа с командами от внешних модулей в модули внешнего управления и ядро.

###### Класс «instructionController»

Отвечает за работу с инструкциями.

##### Модуль «map»

Модуль для работы с картой.

###### Класс «mapController»

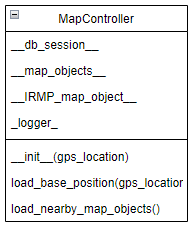


Рисунок 3.5 – MapController

Содержит следующие поля:

* + \_\_db\_session\_\_ – Представляет собой сессию для работы с БД;
  + \_\_map\_objects\_\_ – Представляет собой набор Point (точек на карте;
  + \_\_IRMP\_map\_object\_\_ – объект IRMP на карте;
  + \_logger\_ – класс для логирования;

Содержит следующие методы:

* + \_\_init\_\_(gps\_location) – инициализирует класс, обновляет текущие координаты платформы;
  + load\_base\_positon(gps\_location) – устанавливает начальную позицию платформы;
  + load\_nearby\_map\_objects() – получает все объекты на карте в квадрате 100 на 100 метров.

###### Класс «mapHelper»

Класс с набором методов, для работы с картами

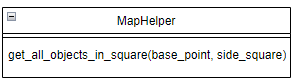


Рисунок 3.6 – MapHelper

Содержит следующие методы:

* + get\_all\_objects\_in\_square(base\_point, side\_square) – получает все объекты на карте в квадрате определённого размера.

###### Класс «AbstractMapObject»

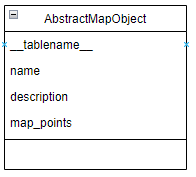


Рисунок 3.7 – AbstractMapObject

Содержит следующие поля:

* + \_\_tablename\_\_ – имя таблицы;
  + name – имя объекта;
  + description – описание объекта;
  + map\_points – точки объекта на карте.

###### Класс «MapObject»

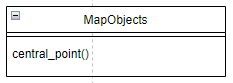


Рисунок 3.8 – MapObjects

Содержит следующие методы:

* central\_point() – вычисляет центральную точку объекта

###### Класс «MapPoint»

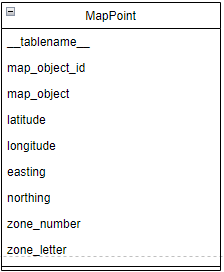


Рисунок 3.9 – MapPoint

Содержит следующие поля:

* \_\_tablename\_\_ – имя таблици
* map\_object\_id – id объекта обратной ссылки
* map\_object – обратная ссылка
* latitude – широта по GPS
* longitude: – долгота по GPS
* easting – Ориентация на восток UTM
* northing – Ориентация на север UTM
* zone\_number – Номер UTM зоны
* zone\_letter – Номер UTM зоны буквенный

##### Файл «Launcher»

Осуществляет запуск программы

## Программа и методика испытаний

### Объект испытаний

Объектом испытаний является система управления интеллектуальной модульной платформой.

#### Наименование работ

Полное наименование работ: выполнение работ по созданию системы управления интеллектуальной роботизированной модульной платформой (далее — Система).

Сокращенное наименование работ: работы по созданию Системы*.*

#### Комплектность испытательной системы

При испытании Системы проверяются:

* Комплекс программных и технических средств, приведенных в п. 4
* Состав и качество документации, указанной в п.6.1

### Цель приемочных испытаний

Целью проведения приемочных испытаний является проверка соответствия созданной системы управления интеллектуальной модульной платформы на выполнение работ по созданию системы управления интеллектуальной модульной платформы.

Задачами приемочных испытаний является последовательное выполнение этапов проверки согласно «Программе и методике приемочных испытаний».

### Требования к программе и программной документации

#### Перечень руководящих документов, на основании которых проводятся приёмочные испытания

При проведении приемочных испытаний соблюдаются требования, изложенные в техническом задании на разработку системы управления интеллектуальной роботизированной платформой.

#### Место и продолжительность приёмочных испытаний

Место проведения приемочные испытания Системы выбирается исходя из установленных модулей на платформе и выбираются непосредственно перед проведением испытаний. Сроки окончания испытаний определяютсявыполнением всех предусмотренных данным документом испытаний.

#### Организации, участвующие в приёмочных испытаниях

В проведении испытаний Системы участвуют следующие организации:

Заказчик: Ювченко Давид Андреевич;

Исполнитель: Ювченко Давид Андреевич.

#### Перечень ранее проводимых испытания

Предварительные испытания и опытная эксплуатация системы не были проведены.

#### Требования к программной документации

Перечень предъявляемых на приемочные испытания документов

* техническое задание;
* программа и методика испытаний;
* инструкция по эксплуатации;

#### Требования к программе

Таблица 3.1 содержит перечень пунктов Технического задания, на соответствие которым проведены приемочные испытания.

Таблица 3.1 — Перечень пунктов Технического задания, по которым проведены испытания

|  |  |  |
| --- | --- | --- |
| № | Наименование проверки | Пункт ТЗ |
| 1 | Комплектность и качество документации | 3.1.3.1 |
| 2 | *Требования к надёжности* | 3.1.2.2 |
| 2.2 | Проверка автоматического перезапуска в случаи возникновения ошибок | 3.1.2.2.2 |

|  |  |  |
| --- | --- | --- |
| № | Наименование проверки | Пункт ТЗ |
| 2.2 | Проверка времени перезапуска | 3.1.2.2.2 |
| 2.3 | *Проверка организации входных данных* | 3.1.2.3.1.2 |
| 2.3.1 | Проверка голосового ввода через микрофон платформы | 3.1.2.3.1.2 |
| 2.3.2 | Выполнение команд через браузер | 3.1.2.3.1.2 |
| *2.3.3* | Проверка голосового ввода через браузер | 3.1.2.3.1.2 |
| 2.5 | *Проверка организации выходных данных* | 3.1.2.3.1.3 |
| 2.5.1 | Проверка получения обратной связи в браузере | 3.1.2.3.1.3 |
| 3. | *Функциональные требования* | 3.1.2.3.1.1 |
| 3.1 | Проверка навигации с помощью GPS | 3.1.2.3.1.1 |
| 3.2 | Проверка патрулирования территории по установленному маршруту | 3.1.2.3.1.1 |
| 3.3 | Проверка патрулирования территории внутри границ | 3.1.2.3.1.1 |
| 3.4 | Проверка идентификации людей с помощью камеры | 3.1.2.3.1.1 |
| 3.5 | Проверка добавления новых модулей | 3.1.2.3.1.1 |
| 3.6 | Проверка ручного управления платформой | 3.1.2.3.1.1 |
| 3.7 | Проверка аварийной остановки | 3.1.2.3.1.1 |

### Методы приёмочных испытаний

#### Перечень этапов приемочных испытаний и проверок, а также количественные и качественные характеристики, подлежащие оценке

Приемочные испытания включают проверку выполнения следующих требований Технического задания:

* комплектности и качества документации Системы;
* соответствия состава и функций Системы, заявленных в ТЗ;
* соответствия программных, технических и других видов обеспечения, заявленных в ТЗ.

#### Последовательность проведения и режима приемочных испытаний

Испытания Системы проводятся путем выполнения проверок, в соответствии с последовательностью, приведенной в настоящем разделе.

Испытания проводятся в стандартном режиме функционирования Системы.

##### Методика проверки комплектности и качества документации

Проверка комплектности и качества документации проводится путем последовательного выполнения (в указанном порядке) следующих частных проверок:

* проверка комплектности представленных на испытания документов, указанных в таблице (см. Таблица 5.);
* проверка содержания и оформления представленных на испытания документов, указанных в таблице (см. Таблица 5.);

Проверка комплектности документов выполняется визуально путем сверки их состава, фактически представленного на испытания, с их составом, определенным в документах:

* Техническое задание на выполнение работ по созданию Системы.

Проверка содержания и оформления представленных на испытания документов выполняется визуально путем:

* контроля соблюдения в этих документах требований к содержанию документов (в части состава разделов и состава представленной в них информации);
* контроля соответствия содержания документов представленным на испытания программным средствам, а также пригодность эксплуатационных документов для эксплуатации Системы.

Последовательность проверки комплектности и качества документации на Систему приведена в таблице (см. Таблица 3.2).

Таблица 3.2 — Проверка комплектности и качества документации

|  |  |  |  |
| --- | --- | --- | --- |
| № проверки | Испытываемая функция | Порядок проверки | Требуемые результаты |
| Проверка комплектности и качества документации | | | |
| 1 | Провести проверку соответствия комплектности представленной документации на Систему согласно комплекту документов, указанному в ТЗ в разделе 6.1 | Выполнение проверки комплектности и полноты документации в составе:   * техническое задание; * программа и методика испытаний; * инструкция по эксплуатации; * ведомость эксплуатационных документов. | Исполнителем выполнены обязательства по разработке документации на Систему в полном объеме |

##### Методика проверки функциональных и нефункциональных требований к Системе

Контроль работоспособности Системы осуществляется при помощи визуального контроля отображения на экране элементов пользовательского интерфейса, в случаи использования браузера для взаимодействия, и ответной реакцией (выполнением каких-либо операций, патрулирование, движение и т.д.) платформы.

Последовательность проведения приемочных испытаний приведена в тестах ниже.

Тест №1. Наличие визуальных компонентов HTML, javascript, CSS, jQuery.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Команды консоли | Выходные данные (результат действия) |
| 1.1 | Открыть в браузере страницу проекта в режиме просмотра кода страницы . |  | Наличие визуальных компонентов HTML, javascript, CSS, jQuery в коде веб-страницы проекта. |

Тест №2. Проверка навигации с помощью GPS

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 2.1 | В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  В соответствии с инструкцией выполнить ввод GPS координат и выполнить команду |  | Убедиться, что роботизированная платформа двигается к указанным GPS координатам или если используется эмуляция, то убедиться что система выдаёт корректные команды на движение. |

Тест №3. Проверка патрулирования территории по установленному маршруту

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 3.1 | В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  В соответствии с инструкцией по эксплуатации ввести маршрут патрулирования |  | Убедиться, что роботизированная платформа двигается по указанному маршруту или если используется эмуляция, то убедиться что система выдаёт корректные команды на движение. |

Тест № 4. Проверка самостоятельного патрулирования территории по указанным границам

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 4.1 | В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  В соответствии с инструкцией по эксплуатации ввести границы патрулирования |  | Убедиться, что роботизированная платформа двигается внутри границ и оценить покрытие территории патрулирования или если используется эмуляция, то убедиться что система выдаёт корректные команды на движение. |

Тест № 5. Проверка идентификации людей с помощью камеры

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 5.1 | В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  В соответствии с инструкцией по эксплуатации добавить себя в базу знаний платформы о людях.  Запустить какой-либо режим патрулирования  Поставить перед платформой в момент патрулирования неизвестного человека  Зафиксировать результат, убрать человека, продолжить патрулирование  Поставить перед платформой в момент патрулирования известного человека |  | Убедиться, что платформа в соответствии с инструкцией не идентифицировала неизвестного ей человека и успешно идентифицировала известного |

Тест № 6. Проверка возможности добавления функций при установке новых модулей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 6 | Установить поддерживаемый модуль в соответствии с инструкцией по эксплуатации  В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  Выполнить некоторые команды по управления новым модулем |  | Убедиться, что модуль активен и работает или если используется эмуляция, то убедиться что система корректно взаимодействует с модулем |

Тест № 7. Проверка ручного управления платформой, включая ручное управление движением и работой дополнительных модулей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 7 | В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  В соответствии с инструкцией по эксплуатации выполнить ручное задание команд платформе на движение  В соответствии с инструкцией по эксплуатации выполнить остальные доступные команды ручного управления |  | Убедиться, что платформа движется или если используется эмуляция, то убедиться что система выдаёт корректные команды. |

Тест № 8. Проверка возможности аварийной остановки платформы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действия пользователей Системы | Входные данные | Выходные данные (результат действия) |
| 8 | В соответствии с инструкцией по эксплуатации войти в браузер и соединиться с роботизированной платформой  В соответствии с инструкцией по эксплуатации начать выполнение команд платформой  В момент выполнения, в соответствии с инструкцией по эксплуатации выполнить аварийную остановку платформы |  | Убедиться, что платформа перестала выполнять свои действия |

##### Методика проверки требований к видам обеспечения

При проверке требований к видам обеспечения проверяются:

* требования к программному и техническому обеспечению Системы.

Проверка считается выполненной успешно, если соблюдены следующие условия:

* программное и техническое обеспечение установлено и успешно запущено;

##### Перечень работ, проводимых после завершения испытаний, требования к ним, объем и порядок проведения

После завершения испытаний должны быть проведены следующие работы:

* Оформлен документ «Протокол приемочных испытаний»;
* Устранены неисправности функционирования Системы, если такие были обнаружены в процессе проведения приемочных испытаний и зафиксированы в документе «Протокол приемочных испытаний»;
* Оформлен документ «Акт сдачи-приемки работ» по этапу.

### Условия и порядок проведения приемочных испытаний

#### Условия проведения испытаний

Приемочные испытания проводятся в условиях, максимально приближенных к реальным условиям функционирования Системы, силами рабочей группы специалистов заказчика при участии специалистов Исполнителя.

В период проведения испытаний должны выполняться правила техники безопасности в соответствии с действующими у заказчика правилами и положениями.

Считается, что приемочные испытания прошли с положительным результатом, если описанные в Методике испытаний, в тестовых сценариях шаги приводят к ожидаемым результатам (ожидаемой реакции Системы).

#### Условия начала и завершения этапов приемочных испытаний

До начала испытаний:

* На испытательном оборудовании производится установка и настройка программных средств в соответствии с инструкцией по эксплуатации Системы.
* Рабочая группа убеждается в готовности Системы к приемочным испытаниям.

Возникающие в процессе приемочных испытаний сбои и отказы технических средств регистрируются в Протоколе приемочных испытаний.

Приемочные испытания проводятся по утверждённой «Программе и методике приемочных испытаний» в сроки, установленные в соответствии с календарным планом.

Приемочные испытания считаются завершенными после выполнения всех тестовых проверок, фиксирования результатов в Протоколе приемочных испытаний и его подписания.

#### Имеющиеся ограничения в условиях проведения приемочных испытаний

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Имеющиеся ограничения в условиях проведения испытаний заранее оговариваются организациями, участвующими в испытаниях, до начала испытаний и соблюдаются в течение всего времени, необходимого для проведения испытаний.

Перерывы в работоспособности программных и/или аппаратных средств, входящих в испытательную Систему, не являются основанием для признания испытаний неуспешными.

#### Требования к составу технических и программных средств

При эксплуатации Системы входящее в его состав системное программное и аппаратное обеспечение должно соответствовать рекомендациям производителя.

Должны соблюдаться правила эксплуатации компонентов Системы, а также производиться своевременная установка обновлений программного обеспечения, рекомендованных производителями.

#### Меры по обеспечению безопасности приемочных испытаний

В целях обеспечения мер безопасности при использовании технических средств следует соблюдать правила техники безопасности, предусмотренные при работе с электрооборудованием.

#### Требования к персоналу, проводящему приемочные испытания, и порядок его допуска к приемочным испытаниям

Рабочая группа, проводящая приемочные испытания, состоит из персонала Исполнителя и персонала Заказчика. Персонал Исполнителя должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке и настройке Системы. Персонал Заказчика должен ознакомиться с настоящей «Программой и методикой приемочных испытаний» и с инструкцией по эксплуатации Системы.

#### Отчетность

Результаты проведения приемочных испытаний фиксируются в Протоколе приемочных испытаний, который подписывает Заказчик и Исполнитель. Перечень выявленных в ходе приемочных испытаний замечаний может быть оформлен в качестве приложения к протоколу. Как недостатки реализации оформляются исключительно выявленные отклонения от ТЗ. Прочие недостатки могут быть внесены в протокол как желательные доработки. Наличие желательных доработок не влияет на признание результатов испытания успешными и на процесс передачи Системы в промышленную эксплуатацию.

В случае значительного отклонения Системы от требований, предъявляемых на испытаниях, сроки проведения испытаний могут быть перенесены в пределах сроков выполнения в соответствии с Календарным планом выполнения работ.

# Эксплуатационная документация на программный продукт

## Руководство пользователя

### Общие сведения о программе

Полное наименование программы: Система управления интеллектуальной роботизированной модульной платформой.

Краткое название программы: система управления.

Система управления отвечает за работу всей платформы. Именно она принимает основные решения о дальнейших действиях платформы. Она отвечает за управление работой установленных модулей, общение с пользователем и движение платформы.

### Установка программы

Программа поставляется совместно с платформой. И непосредственно установку программы выполнять не нужно.

### Первый запуск

После включения платформы происходит автоматический запуск системы.

При первом запуску платформа начнёт раздавать Wi-Fi сигнал. Необходимо будет к нему подключиться. После чего зайти в браузер и ввести адрес: 127.0.0.1. После должен будет открыться сайт. На сайте необходимо будет выполнить регистрацию.

Кнопка для входа и регистрации находится в верхнем правом углу экрана. При нажатии на ней появится окно входа или регистрации. Нам нужно окно регистрации. Необходимо будет заполнить поля и ввести ключ платформы, который выдаётся при покупке платформы.

После успешной регистрации нужно выбрать кнопку «настройка» на панели сверху. На открывшейся странице нужно ввести имя WI-FI сети, подключённой к интернету и ввести пароль. После чего нажать кнопку подключиться. При успешном подключении платформа издаст звуковой сигнал и сообщит о новом адресе страницы. Далее необходимо будет также подключиться к сети, к которой подключили систему и ввести новый адрес.

После подключения к интернету необходимо будет нажать кнопку «обновить систему и модули». После чего произойдёт загрузка и обновление системы. После успешной загрузки нужно нажать кнопку перезагрузки.

После этого необходимо будет, примерно, через пять минут заново подключиться к сайту и если никаких ошибок выведено не будет, то система готова к работе.

### Выбор модулей

Открыв сайт мы должны перейти к настройкам. В настройках будут отображаться все доступные модули, которые поддерживает система. Нам необходимо выбрать те, которые установлены на неё. Что-бы узнать, какие модули установлены у вас необходимо будет либо обратиться к документам, которые выдаются при покупке, либо нажать на кнопку «провести поиск установленных модулей». Далее установленные модули автоматически активируются.

Доступные модули будут отображаться в выпадающем списке, который появляется при нажатии на кнопку «модули». Для изучения возможностей модуля рекомендуется обратиться к инструкции к этому модулю.

### Просмотр информации о платформе.

Для просмотра информации о местоположении платформы необходимо зайти на сайт, и выбрать «Управление» / «Просмотр карты». Откроется карта, на которой будет отображено положение платформы и некоторая дополнительная информация, которая зависит от установленных модулей.

### Аварийная остановка

Для аварийной остановки платформы необходимо либо зайти на сайти и нажать кнопку «аварийная остановка», которая находится в низу экрана, либо произнести «Стоп» или «Остановись» находясь у платформы. В случаи голосовой остановки, платформа произнесёт: «Подтвердите остановку», после чего нужно будет сказать «да».

### Модуль «Патрулирование».

Это стандартный модуль, поставляемы с платформой. Он позволяет запустить патрулирование определённой территории. Робот будет патрулировать указанную территорию и искать людей. При обнаружении человека, он попробует его распознать по лицу, если у него не получится это сделать, то он отправит сообщение на сайти и воспроизведёт предупреждение звуком.

Для выбора маршрута патрулирования необходимо зайти в «Управление» / «Управление маршрутами» и нажать на кнопку «Создать новый маршрут». Далее необходимо выбрать способ задания маршрута: «Патрулирование по точкам» или «Патрулирование в пределах территории»

После нажать на необходимые точки на карте и нажать на кнопку «запустить патрулирование».

Для того, чтобы добавить человека в список, «одобреных» людей, необходимо открыть модуль «патрулирование» и нажать на кнопку «добавить человека». После чего необходимо следовать инструкции на экране. Для удаления людей необходимо нажать кнопку «удалить человека» и выбрать его из появившегося списка.

## Руководство разработчика модулей.

# Акт испытаний программного продукта

Объектом испытаний является система автотестирования пользовательских интерфейсов программной линейки продуктов НПО Криста Испытания программного продукта проводили:

* Разработчик Бубенцов С. А.
* Руководитель дипломного проекта Смирнов Н. В.

При этом проверялось соответствие программного продукта техническим требованиям.

Испытания проводились при соблюдении, указанных в техническом задании, требованиях к аппаратуре. Результаты испытаний зафиксированы в протоколах. Испытания проводились по методике испытаний (раздел 3.4).

Проведя тестирование по всем пунктам методики испытаний, можно сказать, что программный продукт работает корректно и удовлетворяет заданным требованиям. Работу можно считать завершенной с положительным результатом.

Студент-дипломник\_\_\_\_\_\_\_\_\_\_Ювченко Д.А.

Руководитель  
дипломного проекта Паламарь И.Н.

# Экономическое обоснование

## Экономическое обоснование разрабатываемого программного обеспечения

С экономической точки зрения внедрение системы автоматизированного тестирования позволит получить высвобождение ресурсов для отдела тестирования, так как автоматические тесты могут прогоняться либо совсем без участия человека, либо с минимальным вмешательством, требуемым для настройки запусков и окружений тестов. Также выигрыш может быть получен за счет увеличения скорости прогона автоматизированных тестов. Плюсом является также высокий уровень повторяемости, то есть обеспечения сравнимых прогонов, исключающих человеческий фактор. Методика расчета экономической эффективности будет включать в себя:

* описание функций, назначения и потенциальных пользователей ПО;
* определение длительности работ и календарного плана. Будет проведена оценка времени, необходимого на проектирование, разработку и внедрение системы;
* сравнение с аналогами;
* расчет расходов на заработную плату работников;
* расчет расходов на сопутствующие материалы;
* расчет расходов на оборудование;
* определение полной себестоимости разработки системы;
* оценка результата от использования ПО.

Таким образом, путем расчетов можно будет убедиться в экономической выгоде, получаемой от проекта разработки, и на основе этих данных принять управленческое решение о возможности разработки и внедрения системы.

## Сравнение с аналогами

На сегодняшний день, для автоматизированного тестирования, существует множество инструментов, но их основным минусом является стоимость. Для примера будут рассмотрены несколько программных продуктов:

IBM RFT — платный инструмент автоматического, функционального и регрессионного тестирования. Это программное обеспечение предоставляет возможности автоматического тестирования для функционального, регрессионного, графического интерфейса и тестирования на основе данных. Цена составляет от 3400долларов в месяц, что для российского ITрынка является достаточно большой, даже для крупных компаний.

Система HP UFT обладает схожим функционалом с IBM RFT, но также является платной, стоимость составляет от 600 долларов в месяц.

Помимо условной бесплатности, разрабатываемая система обладает такими преимуществами, как наличие открытого исходного кода, и осуществление поддержки внутренним персоналом компании, что увеличит оперативность в решении проблем и в осуществлении необходимых доработок.

## Описание функций, назначения и потенциальных пользователей программного обеспечений

### Назначение программного обеспечения

Система автотестирования предназначена для автоматизации тестирования веб-интерфейса программных продуктов НПО «Криста».

### Функции программного обеспечения

Система обеспечивает возможность выполнения следующих функций:

* подключение к тестируемой системе – необходимо обеспечить

возможность тестирования уже развёрнутого и готового к использованию программного продукта. В данном случае, подключение должно осуществляться, путём ввода URL-адреса системы, которую необходимо протестировать;

* выполнение тестовых сценариев – предоставляется возможность

выполнения разработанных тестов как на локальном браузере, так и на удалённом. Также имеется возможность выполнения разработанных тестовых сценариев по расписанию;

* создание отчётов – в систему интегрирован инструмент построения

графических отчётов автотестов, упрощающий их анализ.

### Потенциальные пользователи программного обеспечения

Пользователями ПО будут являться разработчики, пишущие тестовые сценарии для проверки функциональности тестируемой системы, тестировщики, и менеджеры, анализирующие результаты тестирования.

## Расчет затрат на разработку программного обеспечения

В таблице 6.1 представлены необходимые работы и их длительность.

Таблица 6.1 – Расчет длительности работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование работ | Длительность работ, дней | | |
| Минимум | Максимум | Ожидаемая |
| 1 | Изучение функциональных требований, присланных заказчиком | 1 | 2 | 2 |
| 2 | Изучение литературы | 2 | 4 | 3 |
| 3 | Разработка алгоритмов | 3 | 10 | 8 | |
| 4 | Разработка ПО | 30 | 50 | 40 | |
| 5 | Отладка ПО | 10 | 20 | 15 | |
| 6 | Тестирование | 5 | 10 | 7 | |

Продолжение таблицы 6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | Экономическое обоснование | 1 | 2 | 1 |
| 8 | Оформление пояснительной записки | 3 | 9 | 6 |
|  | Итого | 55 | 107 | 82 |

Исходя из данных о работах и их длительности, необходимо рассчитать затраты на разработку ПО. В таблице 2 приводится расчет затрат на основную заработную плату.

Расчет величины основной заработной платы *Зо*, руб. участников команды производится по формуле:

*Зо* = , (6.1)

где n – количество исполнителей, занятых разработкой конкретного ПО;

– часовая заработная плата i-го исполнителя, руб;

– трудоемкость работ, выполняемых i-м исполнителем, ч.

Трудоемкость рассчитывается по формуле:

, (6.2)

где – ожидаемая продолжительность работ;

и – соответственно наименьшая и наибольшая по мнению эксперта длительность работы.

Для разработки любой системы или подсистемы необходимо несколько человек, так как один разработчик со всем справиться не сможет. Поэтому в данном случае в команду входят один программист, один аналитик и один тестировщик.

На примере программиста покажем, как рассчитывается его заработная плата. Трудоемкость работы программиста, равна следующему значению:

= (3 \* 45 + 2 \* 84) / 5 = 60,6 д. = 484,8 ч.

Поскольку месячная заработная плата программиста составляет 80000 рублей, то часовая зарплата составит:

= 80000 / (22⋅8) = 455 руб./ч.

Таблица 6.2. – Расчет затрат на основную заработную плату

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Участник команды | Выполняемые работы | Месячная з/п, р. | Часовая з/п, р. | Трудоем-кость работ, часов | Основная з/п,р. |
| 1 | Программист | 2, 3, 4, 5 | 80000 | 455 | 484,8 | 220 584 |
| 2 | Аналитик | 1, 7, 8 | 55000 | 313 | 65,6 | 20 532 |
| 3 | Тестировщик | 6 | 35000 | 200 | 56 | 11 200 |
| Итого затраты на основную заработную плату разработчиков | | | | | 252 316 | |

Затраты на дополнительную заработную плату разработчика включают выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяются по формуле:

(6.3)

Рассчитаем затраты на дополнительную заработную плату, где Зо – затраты на основную заработную плату с учетом премии; Нд- норматив дополнительной заработной платы, равный 10%.

Социальные отчисления (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле:

, (6.4)

Рассчитаем затраты на социальные отчисления, где – норматив отчислений на социальные нужды (согласно действующему законодательству) равный 30%.

Рассчитаем затраты на эксплуатацию оборудования:

*,*

где – издержки на заработную плату обслуживающего персонала, руб./год;

– годовые издержки на амортизацию, руб./год;

– годовые издержки на электроэнергию, потребляемую ЭВМ, руб./год;

– годовые издержки на вспомогательные материалы, руб./год;

– затраты на текущий ремонт компьютера, руб./год;

– прочие и накладные расходы, руб./год;

в данной формуле принимаем равной нулю.

Затраты на амортизацию определяются по формуле:

*Zам = Cба\* Нам*, (6.5)

где *Cба* – балансовая стоимость компьютера, р.;

*Нам* – норма амортизации, %.

Для вычислительной техники норма амортизации допускается в размере 25 %. *Cба* в данном случае равно 40 000 руб.

Затраты на амортизацию составят:

*Zам* = 40000 \* 0.25 = 10000 рублей / год.

Затраты на электроэнергию, потребляемой за год, определяется по формуле:

*Zэт = Pсп\* Tэф\* Cэт*. \* *A*, (6.6)

где Сэт – стоимость 1 кВт/ч электроэнергии, равная 3,84 руб.;

Тэф – действительный годовой фонд времени работы ЭВМ, час/год;

Pcn – суммарная потребляемая мощность ЭВМ, равная 0,5 кВт;

*A* – коэффициент интенсивного использования мощности машины.

Действительный годовой фонд времени ЭВМ равняется числу рабочих часов для участников команды, за вычетом времени на профилактику и ремонт ЭВМ. Время профилактики: ежемесячная – пять часов, ежегодная – 7 суток. Следовательно, *Тгф* = 260 × 8 – (7 \* 8 + 5 \* 12) = 2 080 – 116 = 1 964 ч. Так как фактическое время работы ЭВМ совпадает с плановым временем работы участников команды, следовательно, коэффициент интенсивного использования мощности машины принимается равным единице.

Таким образом, затраты на электроэнергию за период разработки составят

*Zэт* = 3,84 × 1964 × 0,5 × 1 = 3770,88 руб./год.

Затраты на ремонт ПК, 5% от стоимости ПК

*Zтр* = 40000 × 0.05 = 2000 рублей

Затраты на необходимые материалы, 1% от стоимости ПК

*Zвм* = 40000 × 0.01 = 400 рублей

Прочие затраты на ПК, 5% от стоимости ПК

*Zпр* = 40000 × 0.05 = 2000 рублей

Полные затраты на эксплуатацию составят:

*Zзэ* = 10 000 + 3770,88 + 400 + 2000 + 2000 = 18 170,88 руб./год.

Цену машино-часа можно определить по формуле:

*Смч = Zзэ / Тгф*,

где *Z*зэ – полные затраты на эксплуатацию ЭВМ в течение года;

Тгф – действительный годовой фонд времени работы ЭВМ, час/год.

*Смч* = 18 170,88 / 1 964 ≈ 9,5 руб./ч.

Для расчета фактического времени отладки программы используется формула:

*tфв* = *t*бс + *t*д + *t*о, (6.7)

где tбс – затраты труда на составление программы по готовой блок-схеме, ч;

tд – затраты труда на подготовку документации задачи, ч;

tо – затраты труда на отладку программы на ЭВМ при комплексной отладке задачи, ч.

*tфв=* 484,8 + 65,6 + 56 = 606,4 ч

Тогда затраты на машинное время, составят:

*Zомв* = *Смч* × *t*фв = 9,5 × 606,4 = 5 760,8 руб.

В качестве прочих затрат будем считать плату за отопление, электричество, бумаги и другие внеплановые траты. Размер установим, как 25% от затрат на разработку.

Общие затраты на разработку ПО приведены в таблице 6.3.

Таблица 6.3. – Общие затраты на разработку ПО

|  |  |
| --- | --- |
| Статья затрат | Сумма, рублей |
| Основная заработная плата команды разработчиков(Зо) | 252 316 |
| Дополнительная заработная плата команды разработчиков(Зд) | 25231 |
| Социальные отчисления(Зсоц) | 83 264,34 |
| Машинное время (Zомв) | 5 760,8 |
| Прочие затраты | 91 643 |
| Общая сумма затрат на разработку | 458 215,14 |

Итоговая сумма разработки системы автотестирования составляет 458 215,14 рублей.

## Оценка результата от использования ПО

В результате разработки ПО, заказчик получил продукт, полностью удовлетворяющий его запросам. Использование разработанной системы автотестирования позволяет снизить затраты на поддержку, повысить надежность написанных тестов, а также упросить их написание и чтение.

Благодаря разработанной системе по автоматизации тестирования, время пользовательского интерфейса уменьшилось. Прямым последствием является то, что уменьшилась нагрузка на разработчиков, разрабатывающих автоматизированные тесты. Так же применение разработанной системы для написания автоматизированных тестов позволило упростить анализ отчётов, создаваемых по результатам выполнения тестовых сценариев. Это позволило снизить время реакции на найденные ошибки.

Повышение производительности труда можно рассчитать по формуле:

(6.8)

где Fj – время, которое планировалось пользователем для выполнения работы j-го вида до внедрения программы, ч;

ΔTi - количество часов, которые экономит пользователь.

Если сравнить сколько минут выполняется работа до автоматизации и сколько позже, то можно увидеть сколько экономиться времени и насколько повышается производительность труда.

В таблице 6.4 приведено повышение производительности труда при различных видах работ пользователей.

Таблица 6.4 – Повышение производительности выполнения работ

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работ | До авто­матизации, мин (*Fj)* | Экономия времени, мин. | Повышение про­изводительности труда *Рi*, % |
| Проведение тестирования пользовательского интерфейса | 120 | 100 | 500 |
| Составление отчёта по результатам тестирования | 20 | 15 | 100 |

Из данной таблицы 6.4 видно, что самые трудоемкие виды работ: проведение тестирования пользовательского интерфейса, составление отчёта по результатам тестирования. Автоматизация данных видов работ позволяет сэкономить большое количество времени, и значительно повышает производительность труда.

Среднемесячная зарплата тестировщика компании составляет 35000 рублей, тогда часовая зарплата составит 200 руб./ч. В день он успевает сделать примерно 3 задачи, связанных с проверкой интерфейсных компонентов систем и на каждую задачу у него уходит примерно два часа времени. Следовательно, за выполнение именно этой части работы, его заработок составляет: (200 руб./ч 6 ч) 22 дня = 26 400 руб.

В отделе тестирования системы «Исполнение бюджета» работают 5 тестировщиков. Если предположить, что они будут пользоваться разработанной системой, то экономия, связанная с повышением производительности труда, будет составлять:

руб.

Годовая экономия *Эр* (руб./год) складывается из экономии эксплуатационных расходов и экономии в связи с повышением производительности труда пользователя:

*Эр* = (*Р1* – *Р2*) + Δ*Рп,* (6.9)

где *Р*1 и *Р*2 – соответственно эксплуатационные расходы до и после внедрения разрабатываемой программы;

Δ*Р*п – экономия от повышения производительности труда дополнительных пользователей.

Если эксплуатационные расходы на тестирование пользовательского интерфейса проекта компании составляют примерно 5 000 руб. за один проект в месяц, а таких крупных проектов в месяц 3, то за год ожидается экономия в ((5 000 руб. 3) 12) / 6 = 30000 руб.

Следовательно, годовая экономия *Эр* = 30 000 + 792 000= 822 000 руб./год.

Критерием эффективности создания и внедрения новых средств автоматизации является ожидаемый экономический эффект. Он определяется по формуле:

*Э* = *Эр* – *Ен*×*Кп,,* (6.10)

где Эр – годовая экономия, р.;

Ен – нормативный коэффициент (Eн = 0,15);

Кп – капитальные затраты на проектирование и внедрение, включая первоначальную стоимость программы, р.

Выполним расчет ожидаемого экономического эффекта:

*Э* = 822 000 – 0,15458 215,14 = 753 267,729 руб.

По данному ожидаемому экономическому эффекту видно, что внедрение системы автотестирования благоприятно скажется для предприятия. Сократит количество расходов и при этом цена от внедрения подсистемы окупится в будущем, за счёт экономии средств.

До внедрения данной системы, в ряде проектов, тестирование пользовательского интерфейса, выполнялось вручную. Из-за этого время тестирования значительно увеличивалось, также страдала точность результатов из-за «человеческого фактора».

После автоматизации тестирования пользовательского интерфейса время, требуемое на выполнение работ, значительно сократилось. Прямым последствием является то, что уменьшилась нагрузка на тестировщиков.

Также внедрение разработанной системы, позволило снизить время реакции на найденные ошибки.

# Заключение

В ходе выполнения выпускной квалификационной работы была разработана система автотестирования пользовательских интерфейсов программной линейки продуктов НПО «Криста». Так же была выполнена разработка следующих документов:

* техническое задание (подраздел 3.1);
* пояснительная записка (подраздел 3.2);
* описание программы (подраздел 3.3);
* методика и программа тестирования (подраздел 3.4).

Также, в работе:

* определены критерии для сравнения аналогичных систем автотестирования (подраздел 2.1);
* было произведено сравнение существующих систем (подраздел 2.2).

Система автотестирования предназначается для разработчиков, тестировщиков, менеджеров прикладных проектов НПО «Криста». Она представляет собой инструмент, позволяющий разрабатывать тесты пользовательских интерфейсов.

С экономической точки зрения для обоснования целесообразности разработки и использования системы автотестирования были произведены расчеты эффективности системы, продолжительности и трудоемкости работ при разработке системы, заработной платы разработчика (раздел 6).

Разработанная система успешно прошла ряд испытаний, описанных в методике испытаний, на соответствие требованиям, заявленным в техническом задании. Об этом свидетельствует акт испытаний (раздел 5).

# Список литературы

1. Бек, К. Шаблоны реализации корпоративных приложений / К. Бек. –

М. : Вильямс, 2017. – 176 с.

1. Канер, С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ. / С. Канер, Д. Фолк, Е. К. Нгуен. – К. : ДиаСофт, 2001. – 544 с.
2. Шаблоны корпоративных приложений / М. Фаулер [и др.]. – М. : Вильямс, 2016. – 544 с.
3. PageObject. Martin Fowler [Электронный ресурс]. – Режим доступа : https://martinfowler.com/bliki/PageObject.html.
4. Apache Cayenne [Электронный ресурс]. – Режим доступа: URL: https://cayenne.apache.org
5. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001 . — 368 с.: ил. (Серия «Библиотека программиста»)
6. Metanit.com [Электронный ресурс]. – Режим доступа: URL: https://metanit.com/java/
7. Allure.qatools.ru [Электронный ресурс]. – Режим доступа: URL: http://allure.qatools.ru
8. Proglib.io [Электронный ресурс]. – Режим доступа: URL: http:// https://proglib.io/p/docker/
9. HP QuickTest Professional [Электронный ресурс] — Режим доступа: URL: https://ru.wikipedia.org/wiki/HP\_QuickTest\_Professional
10. IBM Rational Functional Tester [Электронный ресурс] — Режим доступа: URL: https://en.wikipedia.org/wiki/Rational\_Functional\_Tester
11. jUnit [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/JUnit
12. Selenium [Электронный ресурс] — Режим доступа: URL: selenium2.ru
13. Автоматизированное тестирование [Электронный ресурс] — Режим доступа: URL: https://en.wikipedia.org/wiki/Test\_automation
14. Приемочное тестирование или Приемо-сдаточное испытание (Acceptance Testing) [Электронный ресурс] — Режим доступа: URL: http://www.protesting.ru/testing/levels/acceptance.html