

Оглавление

Термины и сокращения.....	5
Введение.....	6
1. Предпроектное исследование.....	7
1.1. Обзор роботов удаленного присутствия	7
1.2. Постановка задачи.....	15
2. Концептуальный этап проектирования.	16
2.1. Разработка диаграммы вариантов использования.	16
2.2. Выбор способа передачи данных.....	20
2.3. Выбор XMPP сервера.....	23
2.4. Выбор языка программирования для разработки приложения робота и пользователя.	25
2.5. Выбор операционной системы для сервера.....	26
2.6. Выбор протокола общения с платами управления.	26
2.7. Выбор графической библиотеки для клиентского приложения.	27
2.8. Диаграмма целей	28
2.9. Диаграмма пакетов приложения vrobot.	29
2.10. Диаграмма пакетов приложения vclient.	31
2.11. Разработка технического задания на систему.	33
2.11.1. Основания для разработки.	33
2.11.2. Назначение разработки	33
2.11.3. Характеристики объекта автоматизации.	33
2.11.4. Требования к системе.	33
2.11.5. Требования к надежности.....	34
2.11.6. Условия эксплуатации.	34
2.11.7. Требования к составу и параметрам технических средств.....	34
2.11.8. Требования к информационной и программной совместимости.	35

2.11.9. Требования к маркировке и упаковке.	35
2.11.10. Требования к транспортированию и хранению.....	35
2.11.11. Требования к программной документации.	35
2.11.12. Стадии и этапы разработки.	35
2.11.13. Порядок контроля и приемки.	35
3. Технический этап проектирования.	36
3.1. Диаграмма классов приложения vrobot.....	36
3.2. Диаграмма классов приложения vclient.	38
3.3. Диаграмма классов приложения vadmin.	40
4. Рабочий этап проектирования.	42
4.1. Диаграмма состояний сессии звонка.	42
4.2. Диаграмма состояний vrobot GUI.	43
4.3. Диаграмма состояний vclient GUI.	44
4.4. Диаграмма деятельности обработки запросов приложением vreport-server.....	45
4.5. Диаграмма состояний приложения vshare-server.	46
4.6. Диаграмма состояний приложения vdatastore-server.	47
4.7. Диаграмма состояний управления движением робота.	48
5. Исследовательская часть.	49
5.1. Постановка задачи.....	49
5.2. Выбор способа оценки времени задержки.....	49
5.3. Построение диаграмм среднего времени задержки передачи данных.....	50
5.4. Итоги исследования.	56
6. Организационно - экономическая часть.....	57
6.1. Введение.....	57
6.2. Организация и планирование процесса разработки системы управления роботом удаленного присутствия.....	57
6.2.1. Расчет трудоёмкости разработки системы управления роботом удаленного присутствия.	59
6.2.2. Расчёт трудоёмкости разработки технического задания.	60

6.2.3. Расчёт трудоёмкости разработки эскизного проекта.....	61
6.2.4. Расчёт трудоёмкости разработки технического проекта.....	62
6.2.5 Расчёт трудоёмкости разработки рабочего проекта.	63
6.2.6 Расчет трудоёмкости внедрения программного продукта.	64
6.2.7 Суммарная трудоемкость	64
6.3. Оценка стоимости разработки и внедрения системы управления роботом удаленного присутствия.....	65
6.3.1. Специальное оборудование (амортизация).....	66
6.3.2. Основная заработная плата.	67
6.3.3. Дополнительная заработная плата.....	67
6.3.4. Отчисления в социальные фонды и страхования от несчастных случаев.	68
6.3.5. Накладные расходы.....	68
6.3.6. Цена разработки и внедрения системы управления роботом удаленного присутствия...69	
6.4. Заключение организационно-экономической части.	70
7. МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА И ТЕХНИКЕ БЕЗОПАСНОСТИ.....	71
7.1. Опасные и вредные факторы.....	71
7.1.1. Опасные факторы:	71
7.1.2. Вредные факторы:	71
7.2. Требования к помещениям для работы с ПЭВМ.	72
7.2.1. Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ.....	73
7.2.2. Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ. ...	74
7.2.3. Требования к освещению на рабочих местах, оборудованных ПЭВМ.....	76
7.2.4. Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ.	77
7.2.5. Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах.	78
7.2.6. Общие требования к организации рабочих мест пользователей ПЭВМ.	79
7.2.7. Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей.	80
7.2.8. Электробезопасность рабочего помещения.....	81

7.2.9. Требования пожарной безопасности.	82
7.2.10. Требования к организации медицинского обслуживания пользователей ПЭВМ.....	83
7.2.11. Требования к проведению государственного санитарно-эпидемиологического надзора и производственного контроля.	83
7.2.12. Расчет виброизоляции для системы кондиционирования.	83
7.3. Утилизация ПЭВМ.....	85
7.3.1. Разборка изделий.....	86
7.3.2. Реализация партий.....	90
8. Заключение.....	101
Список использованных источников.....	102
Приложение А. Исходный код программы vreport-server.	103
Приложение Б. Исходный код программы vshare-server.....	105
Приложение В. Исходный код программы vdatastore-server.	108
Приложение Д. Python скрипт для создания установщика приложения робота для debian family GNU / Linux.	112
Приложение Д. Inno Setup скрипт для создания установщика клиентского приложения для ОС семейства Windows.	116

Термины и сокращения

UML	Universal Modeling Language (универсальный язык моделирования)
XML	eXtensible Markup Language (расширяемый язык разметки)
XMPP	Extensible Messaging and Presence Protocol (расширяемый протокол обмена сообщениями)
ОС	Операционная система
ПП	Программный продукт
ПО	Программное обеспечение
ПЭВМ	Персональная электронно-вычислительная машина
ТЗ	Техническое задание
ЭП	Эскизный проект

Введение

Телеприсутствие — набор технологий, позволяющий пользователю с помощью специальных устройств (например, телеуправляемых роботов), получить впечатление того, что он находится и/или воздействует на место, отличное от его физического местоположения.

Бурное развитие вычислительных средств, беспроводные сети следующего поколения, а также новые программные технологии позволяют создать робот удаленного присутствия. Устройство, позволяющее вам, сидя перед экраном вашего компьютера или мобильного телефона, оказаться в другом месте, видеть и слышать все то же, что и робот, общаться с людьми, перемещаться в пространстве.

1. Предпроектное исследование.

1.1. Обзор роботов удаленного присутствия.

Перед разработкой робота удаленного присутствия и системы управления для него был сделан обзор конкурентов:

Anybots QB.



Творение американской фирмы. Несмотря на ужасный внешний вид, этот робот удаленного присутствия помог компании встать на ноги и сейчас позиционируется как решение для корпоративного сектора, получил распространение в США. Высота робота регулируется и может меняться примерно от 60 до 200 см. Собеседник отображается на маленьком 4.3 дюймовом экране. Большого бреда я не встречал, такое можно было не ставить совсем. Забудем про внешний неэтичный вид и посмотрим на механику. Схема робота аналогична segway. Робот постоянно находится в состоянии балансирования. Вообще, такие схемы очень популярны в США. Там сделать свой segway - это просто первая обязанность любого робототехника. Однако на практике такое решение довольно сомнительно. Это подвергает опасности целостность робота при отключении питания, при столкновениях и прочих нештатных ситуациях.

Проходимость робота почти не увеличивается, динамика резко уменьшается. Такие системы имеют длительный разгон. В состоянии покоя они балансируют, это плохо влияет на картинку - она дрожит. В одной модели робота другого производителя для устранения этого эффекта добавлена статическая опора, которая выдвигается при остановке.

Компания anybots имеет веб-интерфейс управления роботом. Зарегистрировавшись на сайте, можно получить тестовый допуск на управление. Это здорово, то что нужно.

Вес робота - 16 кг.

Заявлена способность в помощи обнаружения препятствий и дверных проемов. Как именно, на сайте не указано.

Чтобы получить такую штуку вам придется выложить более 400 000 рублей. В цену входит сам робот, док. станция, запасной аккумулятор, контейнер, годовая подписка на веб-интерфейс.

Да, не забудьте, что кроме самого робота нужно оплачивать время его использования. Стоимость использования данного робота составляет около 3 долларов в день. Это необходимо для содержания серверной части проекта - сайта, откуда происходит управление.

Главным противоречием в схеме segway является требование смещения центра масс робота вниз - это не позволяет разместить сверху большой монитор, хорошую камеру, доп. питание и прочие вкусности. Я решил проверить управление роботом и оформил заявку. Но мне не дали этого сделать, сославшись на то, что у меня в компьютере есть вирусы и я могу нанести урон их сервису.



Gostai-jazz

Французская компания - создатель робота удаленного присутствия и URBI - еще одной сборки Linux ОС с библиотеками для роботов. Приятный сайт производителя имеет некоторые разделы только на французском языке. Имеет зарядную станцию и автоматическую установку на зарядку, довольно приятный внешний дизайн. Машина трехопорная - не по образцу предыдущей. Тип езды плавный, с

длительными разгонами, не регулируется. Верхняя часть робота сменная, и может быть установлен дисплей. Однако дисплей очень маленький и собеседника на нем почти не видно. Сам робот - менее метра в высоту, компактный, но чтобы с ним говорить - людям в демо роликах приходилось нагибаться. Качество звука не удалось оценить – в демо ролике он отсутствует. Качество видео - среднее.

Робот имеет продвинутый веб - интерфейс управления. Буквально второй раз я увидел управление роботом с помощью МЫШИ. Также доступно управление с клавиатуры. Интерфейс довольно информативный, хотя немного старомодный.

Цена неизвестна. Робот нацелен на корпоративных клиентов. Можно попытаться оценить цену исходя из цен на софт. Компания поставляет пару продуктов, один из которых представляет среду разработки конечных автоматов - серьезно. Так вот, эта IDE и среда разработки GUI (владельцами этого робота тогда станет доступна разработка софта под него) - стоят 3000 евро. Один час тех. поддержки стоит 190 евро. Учитывая то, сколько обычно требуется времени консультаций по роботам (я это очень хорошо знаю) - это нонсенс. Внедрение будет стоить как робот. Оценив доступные материалы, можно предположить, что цена робота составляет порядка 15 000 евро.

На данный момент компания продалась Aldebaran Robotics - это тот, кто разработал робот NAO. Для компании это будет источником нового софта и технологий для роботов в будущей деятельности.

Компания также позиционирует робот как средство охраны. В такой продвинутой версии туда ставят лазерный дальномер. Скажу сразу, версия такого робота будет стоить минимум на 50% дороже.

Характеристики:

- Управление через браузер.
- Встроенная камера.
- Gostai 3D pointer технология контроля положения.
- Linux OS Gostai Urbi (v2.4 and above).
- Бортовые микрофон и динамики.

- скорость Max 4km/h - Max скорость вращения: 405°/s.
- Дискретность перемещения - 1.2 см.
- 5 часов активной работы.
- Док-станция и автоматическое соединение с ней.
- Flash 10 (Firefox 2.0, Internet Explorer 7.0 or Safari 3.0) в качестве браузера.
- 12 ультразвуковых датчиков для объезда препятствий.
- 4 ИК датчика для присоединения к док. станции.
- Телеметрия положения и препятствий (доступно на Jazz Security).
- Высота t: 100,5 cm / Ширина: 41 cm / Длина: 35 cm.
- Вес 8 Kg.
- 5" экран (опция).
- Разрешение видеопотока: 320 x 240 pixels . Улучшенный вариант: 640 x 480.
- 60KB/s at 25 fps - 30KB/s at 12 fps up to 12 FPS – MJPEG.
- Возможно большее разрешение - вплоть до 4 Мп.
- Широкий угол обзора - до 170°.
- Возможна ночная съемка (до 0.001Лк).
- Поток видео и звука в обе стороны.
- Дискуссии с собеседником, есть синтезатор голоса.
- Светодиодная подсветка. Есть ИК для ночной съемки.
- Цвета: серый и белый.
- Место под лазерный дальномер.
- Есть детектор столкновения.

- Предупреждение через SMS или E-mail.
- Запись видео и звука доступна.
- Возможность установки лица.
- Мониторинг опасностей и предупреждений.
- Спереди 2 светодиода, а также камера и микрофон.



VGo.

Новый робот удаленного присутствия и новое достижение для этой технологии. Здесь очень удачно соблюдены критерии, которые я описал в начале статьи - высота, габарит, размер экрана, расположение экрана и камеры. Очень хороший дизайн. Машина является 4х опорной, нет проблем с дрожанием картинки. Размер экрана выбран маленьким, оптимально на 3-5 дюймов больше. Качество звука - среднее, качество картинки хорошее.

Цена неизвестна. Робот нацелен на корпоративный рынок и рынок медицины.

Спецификация:

- Габариты 1300 x 330 x 380 мм.
- 6 часовая батарея обычной емкости.
- 12 часовая батарея повышенной емкости.
- Независимый привод обоих колес (он у всех мобильных роботов независимый).
- До 1.5 м/с скорость (высокая).
- Датчики наличия препятствия и столкновения.
- Док-станция.
- Камера 2 Мп.
- H.264 кодек (будут проблемы с фрагментацией картинки в плохих сетях).
- 8khz - звуковой кодек (качество - на уровне мобильного телефона).
- USB порт для подключения аксессуаров (интересно, каких).
- Встроенный WiFi.
- Дисплей 6 дюймов.
- Технология - beam array mic - микрофона с круговой диаграммой приема.
- Динамики
- 38 состояний индикации светодиодов.
- Встроенный touch pad.
- Звуковой процессор (очень емкое название).
- Приложение для удаленного управления.

Робот требует ширину канала интернет - 768 Кбит, динамической подстройки под ширину канала нет. Это гигантский минус. В обыденной жизни по всему миру это не позволит нормально управлять роботом примерно в 60% случаев.

Для управления роботом требуется производительная машина. Софт собран только под ОС Windows.

В целом, неплохой робот. У компании в клиентах такие гиганты как Cisco, HP.

Beam.



Очень интересный проект калифорнийской фирмы Suitable Tech. Во-первых, в отличие от остальных, здесь устранили проблему малого экрана, однако переборщили с этим. 17 дюймовый монитор - слишком много для таких роботов. Как результат робот имеет проблемы с боковой устойчивостью, из-за этого картинка дрожит при езде. Также решили вторую важную проблему - обзор вниз робота. Дело в том, что на всех предыдущих роботах новичку ездить практически невозможно, он сшибет все косяки и предметы вокруг. Здесь установлена вторая камера, смотрящая вниз, что существенно улучшает управляемость для новичков. Робот имеет док станцию. Двойной канал связи существенно улучшает качество продукта. Предусмотрено даже шифрование канала

данных.

Из рассмотренных роботов вот этот единственный претендент на звание полноценного робота удаленного присутствия. Проект даже имеет веб-сервис, который позволяет управлять роботом удаленно. Данная компания нацелила этот проект на дистрибуцию роботов в других местах. В веб-сервисе существует понятие личного кабинета, где вы можете получить доступ к купленным вами роботам или в целях тестирования.

Роботу требуется канал интернет от 2 до 4 Мбит шириной (очень большой для нас, но не проблема для сетей США).

ПО как приложение для управления собрано под Windows и MAC OS. Управление гибкое - как с мыши, так и с клавиатуры.

Характеристики:

- 17 дюймовый монитор.
- Высота - 1.6 метра (очень удобная для разговора).
- Скорость - до 1.5 м/с (высокая для мобильных роботов).

- 2 видеокамеры.
- 6 микрофонов в beam array mic.
- 2 беспроводных адаптера с двумя диапазонами, каждый для уверенного приема (очень актуально).
- Шифрование канала связи.

Double Robotics.



Колеса для вашего iPad - так звучит девиз этого продукта. К удивлению многих, это самый успешный в плане объема продаж в штуках производитель роботов удаленного присутствия. Компанией были предприняты радикальные шаги по снижению стоимости. В частности, в комплект не входит iPad. И далеко не каждый iPad подойдет для этого робота. Так, первые версии iPad не поддерживаются (нет у них камеры). Отсутствует и зарядная станция. Для зарядки робота просто втыкают в розетку.

В чем же привлекательность такого решения? Казалось бы, вам не только нужно поставить туда ваш iPad, но и купить еще один, чтобы управлять. Областью применения этого робота являются офисы. Так первоначально считали основатели компании. Также я слышал мнение, что этот робот - последнее пристанище вашего старого не нужного iPad и почему бы не превратить его в робота, причем по меркам США, недорого стоящего.

Использование iPad естественно, снижает качество телеприсутствия. Однако, iPad неплохо подходит для этого. В частности, его ставят вертикально, в измерении, где он имеет большой угол обзора. Но робот имеет очень тихий звук, внешних колонок у него нет. Общаться с ним можно только вблизи, в условиях офиса это будет почти невозможно сделать (дело в том, что в офисе высокий уровень постоянного шума, мы сами проводили испытания в такой ситуации и знаем это не понаслышке). Качество микрофона скорее всего не позволит вам общаться дальше 1 метра от робота.

Высота и габариты робота подобраны оптимально. Робот имеет схему segway с выдвигной статической опорой. В неподвижном положении статически устойчив.

На данный момент с доставкой робот обойдется вам примерно 100 000 руб. Еще придется потратить порядка 20 000 руб. на iPad3. Итого 120 000 - очень удачная цена для робота удаленного присутствия. Этот продукт нацелен на обычных пользователей, не на корпоративный сектор. Соответственно, нет вспомогательных функций, наподобие охраны, мониторинга и прочего, как это есть у конкурентов. Однако, это будет довольно просто разработать и, думаю, компания сделает это со временем.

Для видео используется API OpenTok. Это видеозвонки владельцам Mac. Это API очень динамично развивается и, как правило, обладает наилучшими характеристиками связи.

Характеристики:

- время работы - до 8 часов;
- вес - 7 кг;
- зарядка от розетки;
- время зарядки от 2х и более часов;
- Touch интерфейс управления.

1.2. Постановка задачи

Необходимо разработать программное обеспечение, позволяющее человеку оказаться в другом месте с использованием робота, передвигаться в пространстве и общаться с окружающими.

2. Концептуальный этап проектирования.

2.1. Разработка диаграммы вариантов использования.

После анализа роботов удаленного присутствия, были выделены основные варианты использования. Для описания вариантов использования хорошо подходят “Use case” диаграммы [5]. Были выделены роли: администратор робота, доверенный пользователь, веб посетитель.

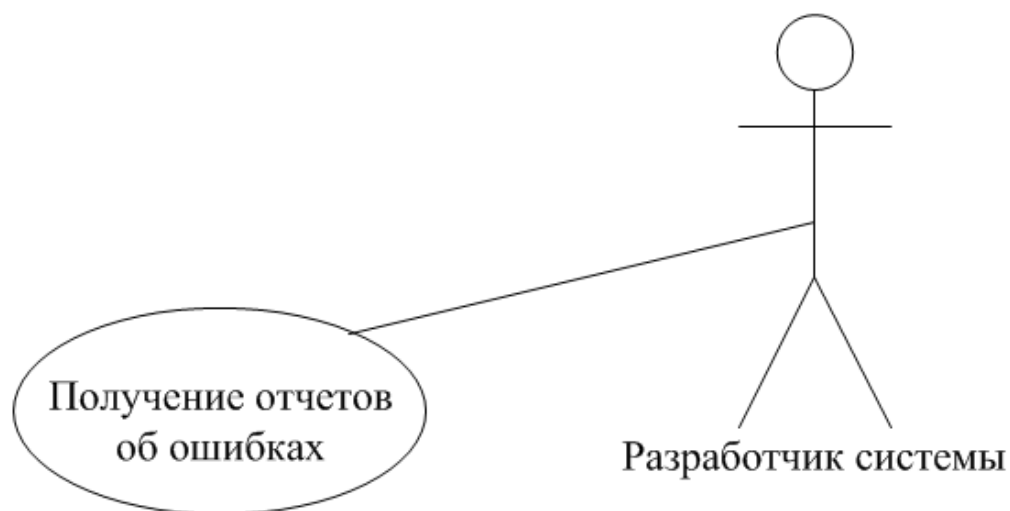


Рисунок 2.1 Диаграмма вариантов использования разработчика системы

Разработчик системы – роль человека, разрабатывающего программное обеспечение. Разработчику системы необходимо получать отчеты об ошибках и сбоях.

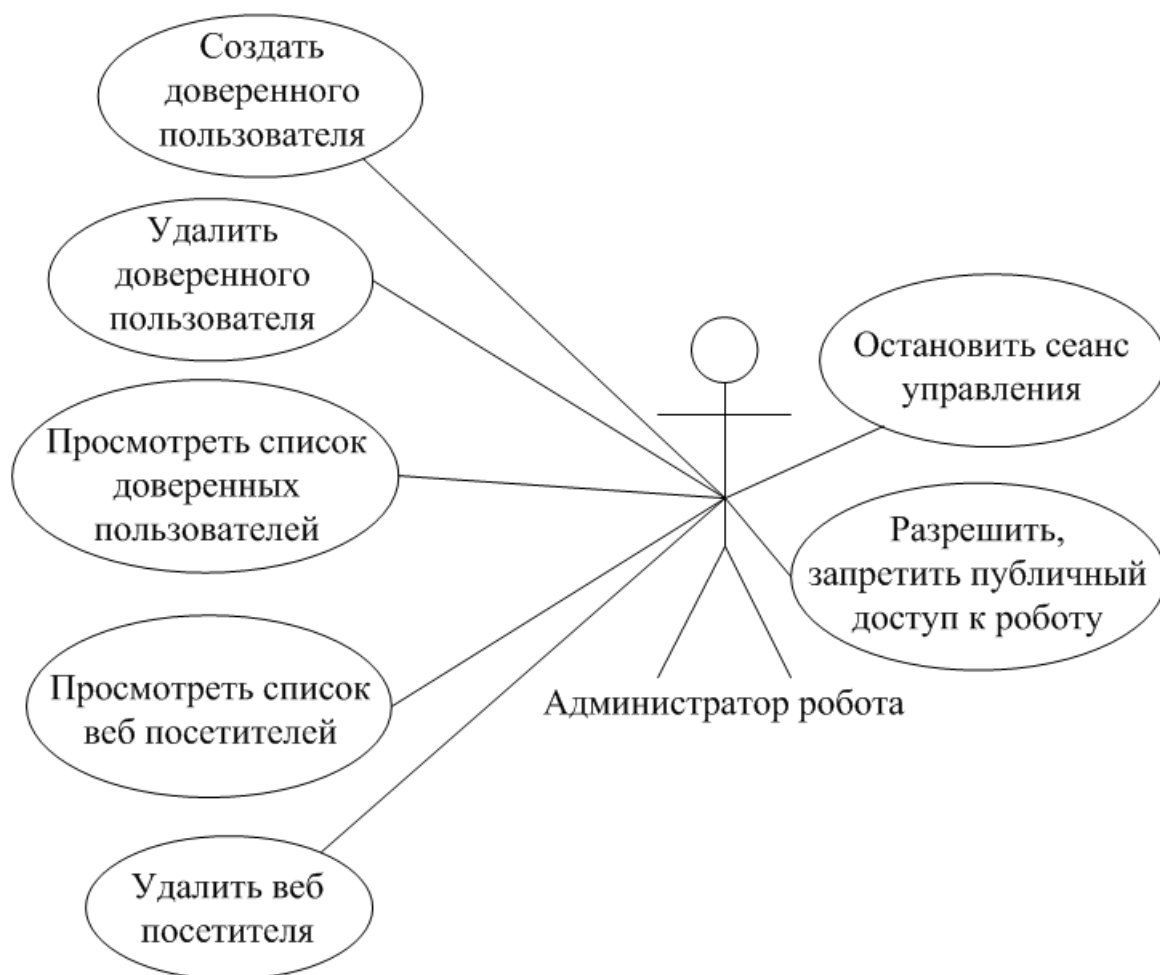


Рисунок 2.2 Диаграмма вариантов использования администратора робота

Администратор робота – роль человека, владеющего средствами администрирования системы и осуществляющего контроль над тем, кто использует робота. Администратор может завершить сеанс пользователя с роботом.

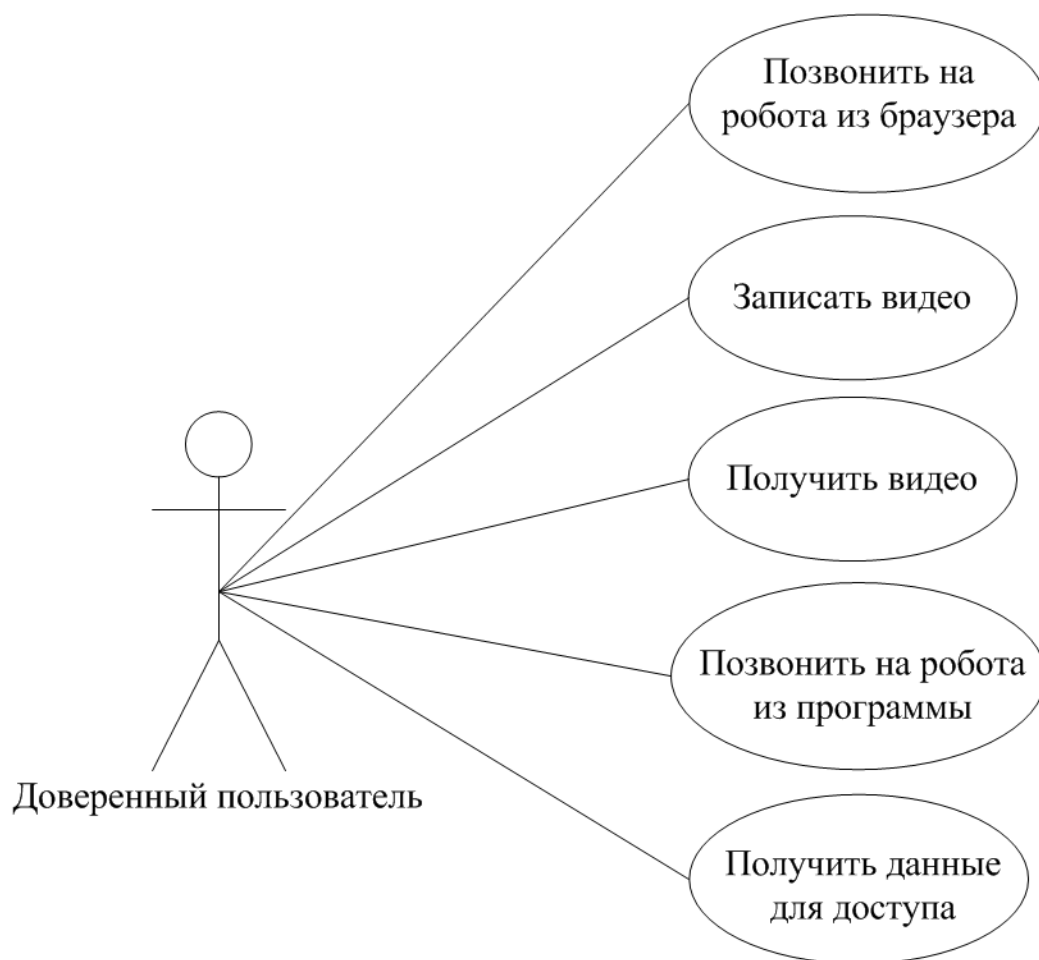


Рисунок 2.3 Диаграмма вариантов использования доверенного пользователя

Доверенный пользователь – роль человека, получающего доступ к роботу для управления. Данный вид пользователей создается администратором на заданный промежуток времени. Пользователь может управлять, записывать видео. Получать записи снятого видео.

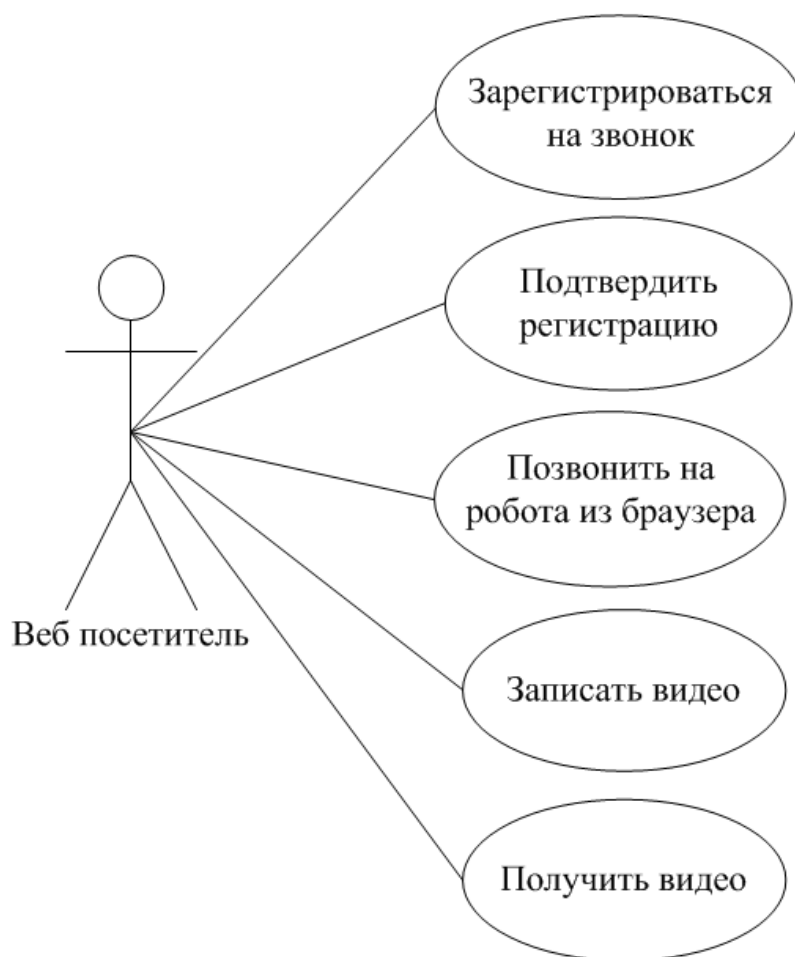


Рисунок 2.4 Диаграмма вариантов использования веб посетителя

Веб посетитель – роль человека, получающего доступ к роботу для управления. От доверенного пользователя, данный пользователь отличается тем, что регистрируется на управление сам, его сеанс управления короче, администратор напрямую не регистрирует пользователей, может только прекратить сессию управления.

2.2. Выбор способа передачи данных.

Простой сервер вещания и просмотра видео на базе ffserver

- передача видео при помощи **ffserver**;
- команды управления передаются через сервер сообщениями фиксированного размера;

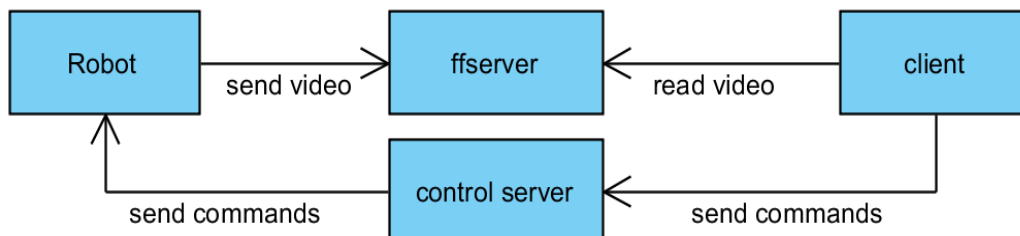


Рисунок 2.5 Структурная схема приложения с ffserver

Недостатки ffmpeg

- проблемы с декодированием h.264 из-за ошибок в исходном коде;
- сложность реализации авторизации роботов на сервере;
- высокая задержка звука и видео из-за буферизации;
- отсутствие подавления эхо в звуковом канале;
- высокая загрузка сети ~1.5 МБ/с.

RTSP сервер

- передачи видео и телеметрии при помощи стандартного протокола RTSP;
- команды управления передаются через сервер сообщениями фиксированного размера;
- механизм авторизации через RTSP server;

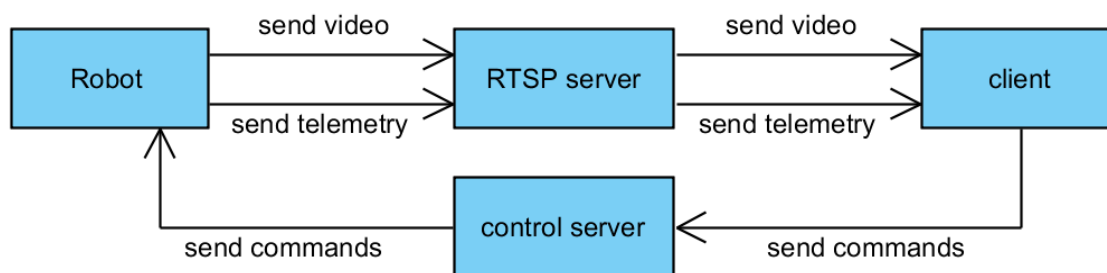


Рисунок 2.6 Структурная схема приложения с RTSP server

Недостатки RTSP сервера

- задержка звука и видео из-за передачи данных через сервер;
- отсутствие подавления эхо в звуковом канале;
- отсутствие шифрования паролей (пароли по сети передаются в открытом виде);
- высокая загрузка сети ~1.5 МБ/с;
- высокая загрузка на сервер.

Общие недостатки клиент-серверной архитектуры

- высоконагруженный сервер для ретрансляции видео;
- задержки видео и звука, связанные с наличием посредника.

"Jingle" - технология установления соединений напрямую

- управление передачей данных осуществляется через стандартный XMPP сервер;
- передача данных напрямую между клиентами в 92 % случаев;
- необходим малонагруженный Relay сервер для передачи данных в 8 % случаев;
- открытый стандарт.

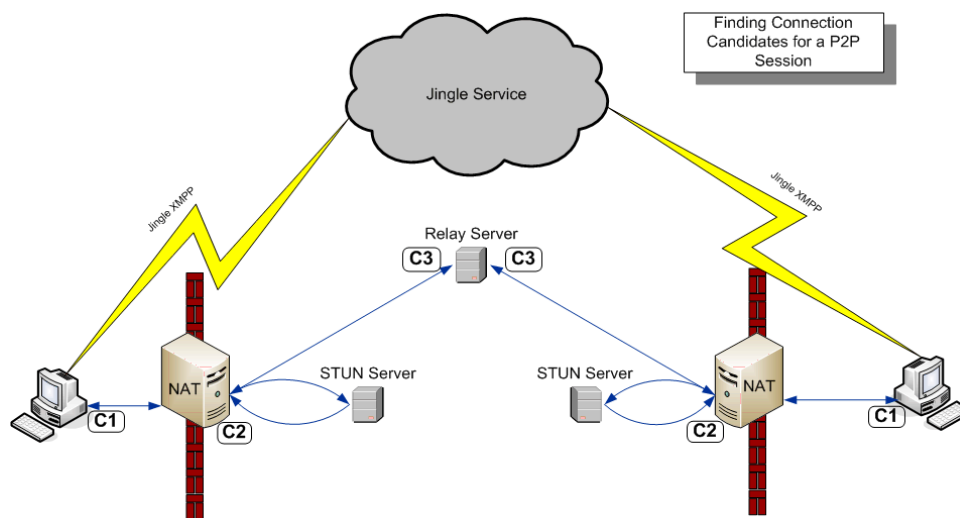


Рисунок 2.7 Схема управления соединениями с использованием Jingle

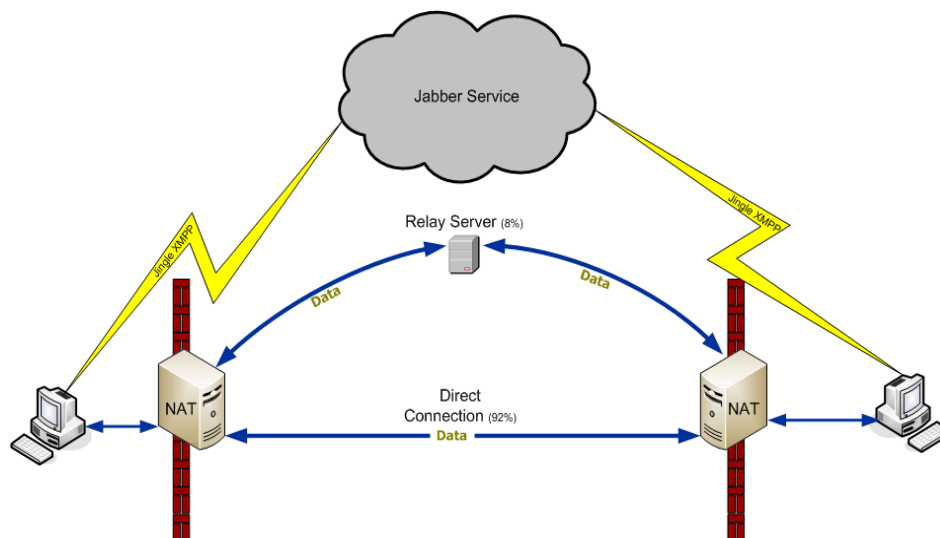


Рисунок 2.8 Схема передачи данных с использованием Jingle

Преимущества использования технологии "Jingle"

- наличие механизма авторизации;
- передача паролей в зашифрованном виде;
- минимальная задержка видео и звука;
- низкая загрузка сервера;
- возможность управления различными типами роботов;
- открытость.

В качестве метода передачи данных был выбран Jingle.

2.3. Выбор XMPP сервера.

XMPP (Extensible Messaging and Presence Protocol — расширяемый протокол обмена сообщениями и информацией о присутствии), ранее известный как Jabber[1] (джа́ббер — «болтовня», «трёп», «тарабарщина») — основанный на XML, открытый, свободный для использования протокол для мгновенного обмена сообщениями и информацией о присутствии (см. список контактов) в режиме, близком к режиму реального времени. Изначально спроектированный легко расширяемым, протокол.

В отличие от коммерческих систем мгновенного обмена сообщениями, таких как AIM, ICQ, WLM и Yahoo, XMPP является децентрализованной, расширяемой и открытой системой. Любой желающий может открыть свой сервер мгновенного обмена сообщениями, зарегистрировать на нём пользователей и взаимодействовать с другими серверами XMPP. На основе протокола XMPP уже открыто множество частных и корпоративных серверов XMPP. Среди них есть достаточно крупные проекты, такие как Facebook, Google Talk, В Контакте, Одноклассники.ru, QIP, LiveJournal, Juick и др.

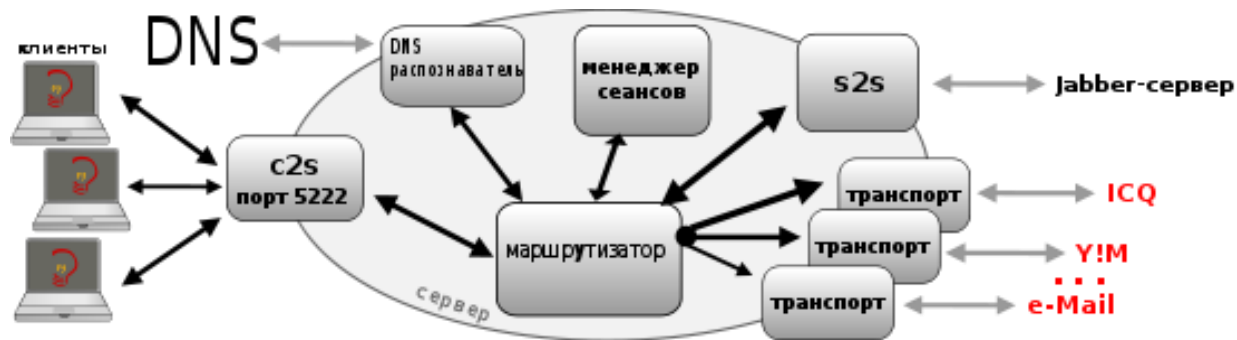


Рисунок 2.9 Схема устройства работы XMPP

Список наиболее распространённых XMPP серверов с сайта The XMPP Standards Foundation <http://xmpp.org/xmpp-software/servers>:

Name	Platform(s)	License	Language	Details
Apache Vysper	Windows / Linux	Apache License Version 2.0	Java	mina.apache.org
ejabberd	Linux / Mac OS X / Solaris / Windows	GPL2	Erlang	process-one.net
iChat Server	Mac OS X	Commercial	-	apple.com
jabberd 2.x	Linux / Solaris / Windows	GPL2	C	jabberd2.org
Jabber XCP	Linux / Solaris / Windows	Commercial	-	cisco.com
Openfire	Linux / Mac OS X / Solaris / Windows	GPL	Java	igniterealtime.org
Open IM	Linux	BSD	Java	openim.techlab.smk.fr
Oracle Communications Instant Messaging Server	Linux / Solaris / Windows	Commercial	-	oracle.com

Таблица 2.1 Список XMPP серверов

Основные критерии выбора сервера:

1. Свободная лицензия. Возможность исправлять программное обеспечение под свои требования.
2. Бесплатность.
3. Удобная расширяемость. Сервер разработан на современном языке программирования – желательно Java.
4. Широкий список поддерживаемых платформ.
5. Активное сообщество.

По данным критериям из списка XMPP серверов подходят: Openfire, Open IM, Apache Vysper.

Из этих трех серверов был выбран наиболее популярный и поддерживающий наибольшее число платформ сервер Openfire.

2.4. Выбор языка программирования для разработки приложения робота и пользователя.

Требования к языку разработки:

1. Поддерживает современные концепции программирования: ООП, шаблоны.
2. Возможность разработки системных приложений.
3. Большое число доступных библиотек.
4. Переносимость на разные операционные системы.
5. Высокая эффективность, производительность.

Из доступных языков был выбран “C++”, потому что он больше подходит для разработки системных приложений.

При разработке ПО на C++ и его отладке активно использовались источники [1] и [2].

2.5. Выбор операционной системы для сервера.

Требования к операционной системе:

1. Низкая стоимость приобретения.
2. Низкая стоимость обслуживания.
3. Низкая стоимость оборудования.
4. Высокая надежность.

Из таких распространённых серверных операционных систем, как GNU/Linux, Windows, Mac OS X Server - GNU/Linux наиболее полно удовлетворяет описанным выше требованиям.

Операционная система GNU/Linux распространяется в виде дистрибутивов, формируемых разными производителями. Требования к дистрибутиву:

1. Низкая стоимость приобретения.
2. Низкая стоимость поддержки.
3. Дружелюбность.

Из доступных серверных дистрибутивов GNU/Linux: Debian, Ubuntu Server, Centos, Red Hat, SUSE Linux Enterprise - Ubuntu Server наиболее полно удовлетворяет описанным выше требованиям.

2.6. Выбор протокола общения с платами управления.

Для управления физическими устройствами исполнения: двигателями постоянного тока, реле, и получения данных с датчиков используются электронные платы. Для взаимодействия этих плат с системой управления необходимо выбрать протокол взаимодействия.

В результате взаимодействия с разработчиком плат был выбран протокол Modbus, так как он позволяет осуществить взаимодействие, прост в использовании, обеспечивает контроль ошибок, для него доступны реализации, как для компьютера, так и для микроконтроллера, на базе которой разработана плата.

Modbus — открытый коммуникационный протокол, основанный на архитектуре «клиент-сервер». Широко применяется в промышленности для организации связи между электронными устройствами. Может использоваться для передачи данных через последовательные линии связи RS-485, RS-422, RS-232, а также сети TCP/IP (Modbus TCP).

2.7. Выбор графической библиотеки для клиентского приложения.

Для разработки графического интерфейса пользователя необходима библиотека. Требования к библиотеке:

1. Бесплатность.
2. Возможность коммерческой поддержки.
3. Поддержка разных операционных систем: Windows, GNU/Linux, Mac OS X.
4. Высокое качество.
5. Открытая лицензия: BSD, LGPL.

Данный список требований удовлетворяют библиотеки: Qt, GTK, FLTK. Из них была выбрана библиотека Qt, потому что для данной библиотеки можно получить коммерческую поддержку от компании разработчика Digia, наличие качественной документации.

2.8. Диаграмма целей

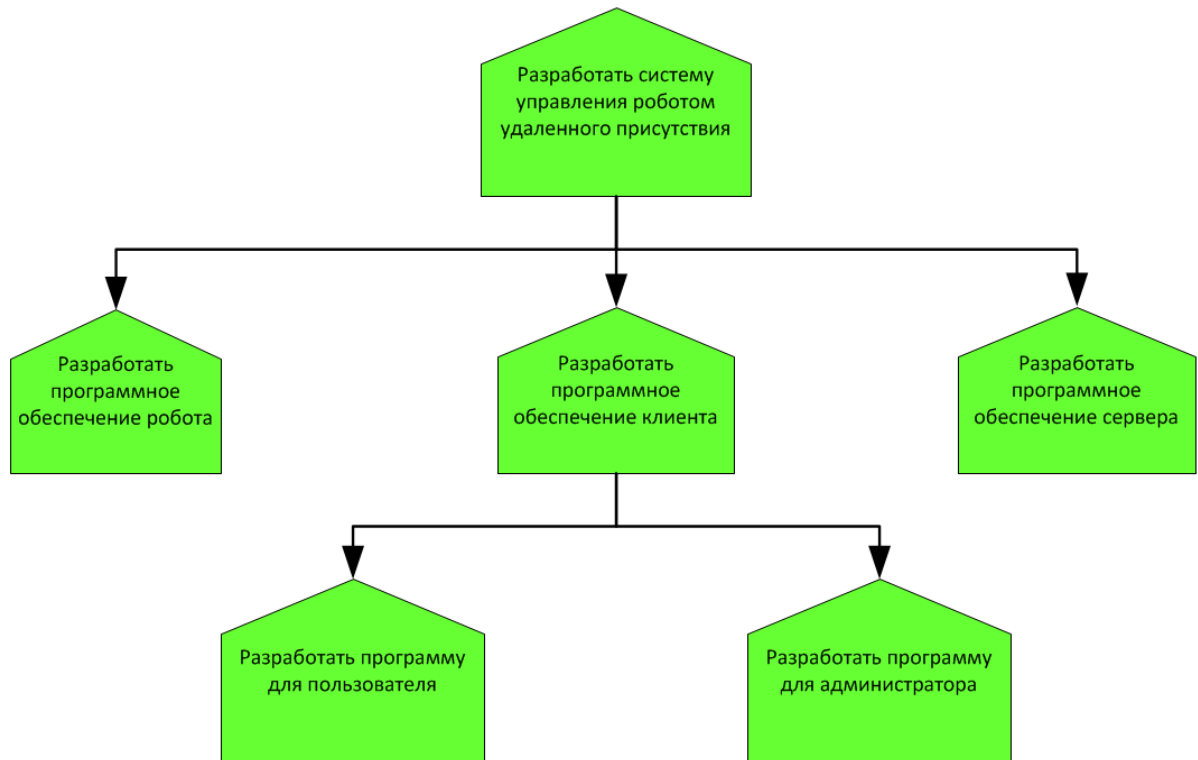


Рисунок 2.10 Диаграмма целей проекта

Были выделены основные цели проекта:

- 1) Разработать программное обеспечение робота
- 2) Разработать программное обеспечение клиента
 - a. Разработать программу для пользователя
 - b. Разработать программу для администратора
- 3) Разработать программное обеспечение сервера

2.9. Диаграмма пакетов приложения vrobot.

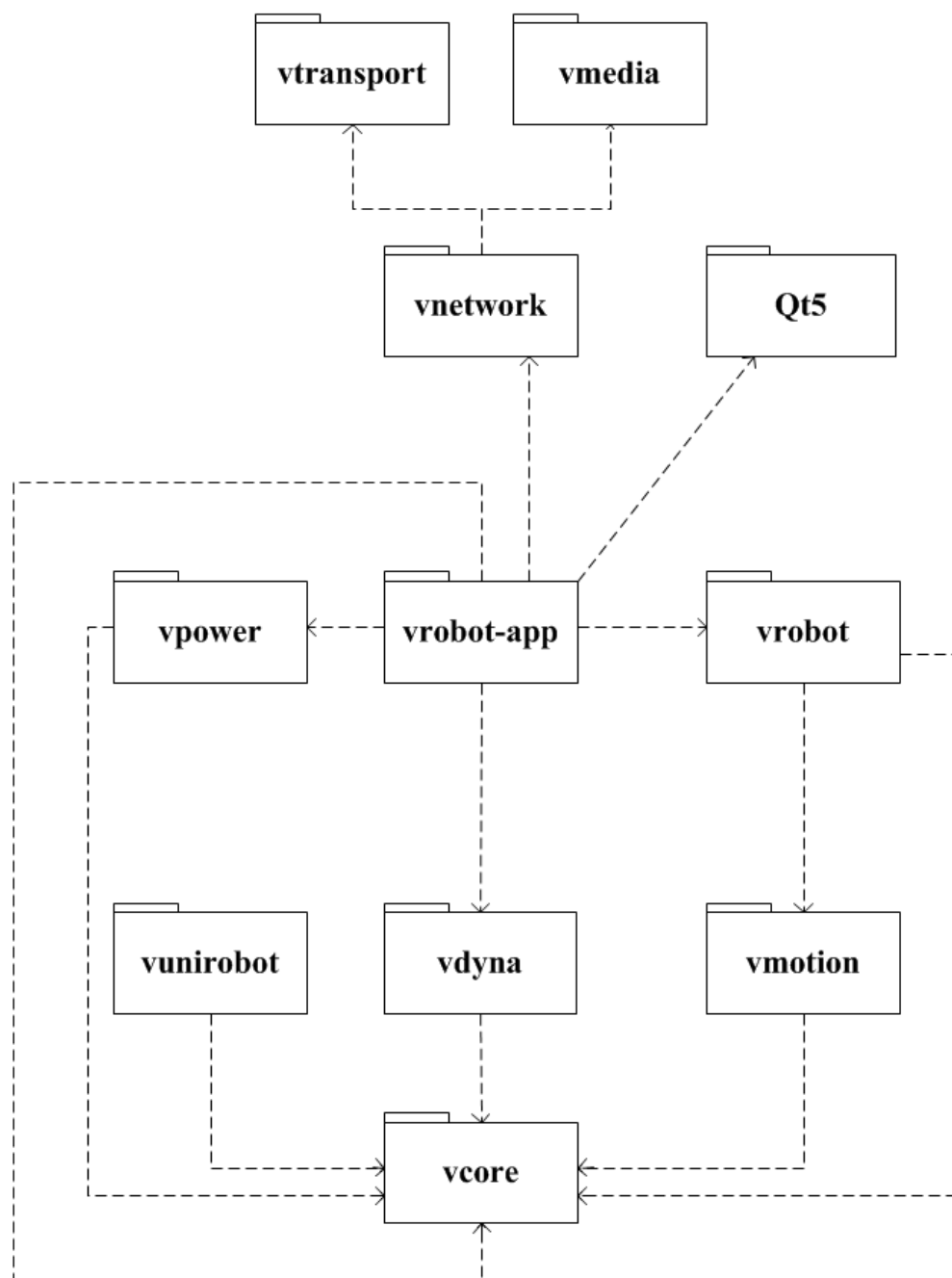


Рисунок 2.11 Диаграмма пакетов приложения vrobot

На концептуальном этапе проектирования была разработана диаграмма пакетов приложения vrobot, был определен список необходимых используемых библиотек:

- vcore – базовая библиотека. Содержит базовые классы для работы с оборудованием и шаблоны проектирования.
- vdyna – библиотека для управления сервоприводами **dynamixel AX-12**.

- vunirobot – библиотека для работы с платой **unirobot**.
- vpower – библиотека для работы с платой **power**.
- vmotion – библиотека с математическими алгоритмами движения робота.
- vnetwork – обобщенная библиотека для сетевого взаимодействия.
- vtransport – библиотека для передачи данных по сети (p2p, direct).
- vmedia – библиотека для работы с мультимедиа: кодирование, декодирование видео, захват данных с камер.
- Qt5 – библиотека графического интерфейса.

vrobot-app – пакет приложения робота.

2.10. Диаграмма пакетов приложения vclient.

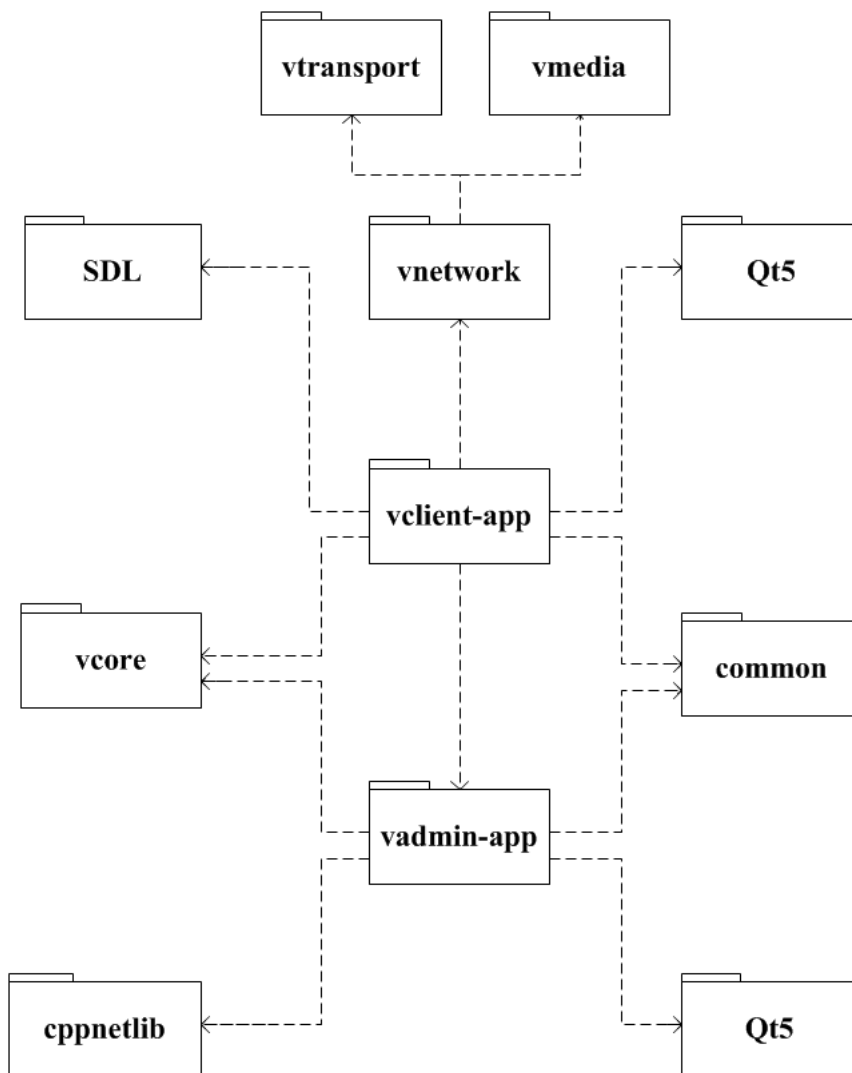


Рисунок 2.12 Диаграмма пакетов приложения vclient

На концептуальном этапе проектирования была разработана диаграмма пакетов приложения vclient, был определен список необходимых используемых библиотек:

- vcore – базовая библиотека. Содержит базовые классы для работы с оборудованием и шаблоны проектирования.
- SDL (Simple DirectMedia Layer) – библиотека для работы с джойстиком.
- cppnetlib – библиотека для отправки GET, POST запросов.
- common – общая для программ из проекта системы удаленного присутствия библиотека вспомогательных классов.

- vnetwork – обобщенная библиотека для сетевого взаимодействия.
- vtransport – библиотека для передачи данных по сети (p2p, direct).
- vmedia – библиотека для работы с мультимедиа: кодирование, декодирование видео, захват данных с камер.
- Qt5 – библиотека графического интерфейса.

vclient-app – пакет приложения управления роботом.

vadmin-app – пакет приложения администрирования.

2.11. Разработка технического задания на систему.

2.11.1. Основания для разработки.

Задание на дипломный проект.

2.11.2. Назначение разработки

Цель разработки системы: предоставить пользователю возможность удаленного присутствия роботом через локальную сеть или сеть интернет.

Основная цель данного курсового проекта – разработать систему управления роботом удаленного присутствия, т.е. написать необходимое программное обеспечение: для робота, для сервера, для клиента – необходимое для удаленного управления.

2.11.3. Характеристики объекта автоматизации.

Робот удаленного присутствия – робот, позволяющий пользователю, например, с помощью специальных устройств (телеуправляемых роботов), получить впечатление того, что он находится и/или воздействует на место, отличное от его физического местоположения.

2.11.4. Требования к системе.

Система управления роботом должна:

1. Передавать аудио и видео от робота клиенту и обратно через локальную сеть или сеть интернет.
2. Передавать команды управления от клиента к роботу через локальную сеть или сеть интернет.
3. Передавать телеметрию от робота к клиенту к роботу через локальную сеть или сеть интернет.
4. Обеспечивать остановку робота в случае разрыва соединения.
5. Сообщать пользователю робота и клиенту о состоянии заряда робота.
6. Воспроизводить полученные аудио и видео на стороне клиента и робота.
7. Обеспечить удобные средства администрирования на сервере.
8. Обеспечить надежность подключения клиента с роботом – доступ по паролю.
9. Обеспечить надежное подключение робота к сети через Wi-Fi.

2.11.5. Требования к надежности.

Основное требование к надежности направлено на поддержание в исправном и работоспособном состоянии серверных ЭВМ и аккуратного обращения с роботом.

2.11.6. Условия эксплуатации.

Система управления роботом должна работать от бортового аккумулятора постоянного тока с напряжением 12В.

Серверные аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью 220В $\pm 10\%$, 50 Гц с защитным заземлением.

2.11.7. Требования к составу и параметрам технических средств.

Серверная часть системы управления должна работать на компьютере с конфигурацией:

- Сетевая карта со скоростью подключения 100 мбит/с.
- Выделенный публичный IP адрес для доступа через интернет.
- Выделенный статический IP адрес для доступа в локальной сети.
- Объем ОЗУ не менее 8192 Мб.
- Объем жесткого диска не менее 320 Гб.
- x86, AMD 64 совместимый процессор.
- Четыре ядерный микропроцессор с тактовой частотой не менее 3000 МГц на ядро.
- openjdk 1.6, sunjdk 1.6.

Система управления робота должна работать на компьютере с конфигурацией:

- Сетевая карта со скоростью подключения 10-100 мбит/с.
- Объем ОЗУ не менее 1024 Мб.
- Устройство хранения данных: eMMC16 Гб.
- armv7 совместимый процессор с низким энергопотреблением.
- Четырех ядерный микропроцессор с тактовой частотой не менее 1000 МГц на ядро.
- Видеоподсистема с аппаратной поддержкой OpenGL ES 2.0.

2.11.8. Требования к информационной и программной совместимости.

Система управления роботом должна работать под управлением различных дистрибутивов GNU/Linux. Серверная часть системы управления должна работать под управлением различных дистрибутивов GNU/Linux. Клиентская часть системы должна работать под управлением различных дистрибутивов GNU/Linux, Mac OSX 10.6 и выше 32/64 bit и OS семейства Windows.

2.11.9. Требования к маркировке и упаковке.

Не предъявляются.

2.11.10. Требования к транспортированию и хранению.

Не предъявляются.

2.11.11. Требования к программной документации.

Описать правила сборки ПО и установки вспомогательных библиотек и программ в README.md файлах.

2.11.12. Стадии и этапы разработки.

1. Изучить существующие решения роботов удаленного присутствия.
2. Концептуальный этап проектирования.
3. Технический этап проектирования.
4. Рабочий этап проектирования.

2.11.13. Порядок контроля и приемки.

Установлены заказчиком в вербальной форме.

- VRobotFacade – класс, реализующий управление и сбор данных для реальных устройств.
- VTestingRobotFacade – класс, реализующий управление и сбор данных с виртуальных устройств.
- VDynamixelThread – класс-поток управления сервоприводами dynamixel в асинхронном режиме.
- VVideoWriter – класс для сохранения видео.
- VVideoWrapper – класс обертка для класса VVideoWriter.
- VDualDeviceManager – класс для создания сдвоенной камеры.
- VDualVideoCapture – класс сдвоенной видеокамеры.
- VRobotClient – класс использующий библиотеки vnetwork и vmedia, является связующим в механизме передачи данных.

Также на диаграмме классов обозначены вспомогательные объекты: enums, constants.

[illegible]

Приложение vclient содержит классы:

- 38

- VInfoWidget – класс виджета информации для вывода сообщений пользователю.
- VEmailDialog – класс диалога для ввода email.
- VBatteryWidget – класс виджета батареи.
- VClientNetworkFacade – класс, использующий библиотеки vnetwork и vmedia, является связующим в механизме передачи данных.
- VRobotController – класс контроллер, обрабатывающий события от пользователя и управляющий роботом через VClientNetworkFacade.
- VClientSettings – класс для сохранения и загрузки настроек пользователя.
- VClientOptionsDialog – класс диалога, для редактирования настроек пользователя.

Также на диаграмме классов обозначены вспомогательные объекты: enums, constants.

3.3. Диаграмма классов приложения vadmin.

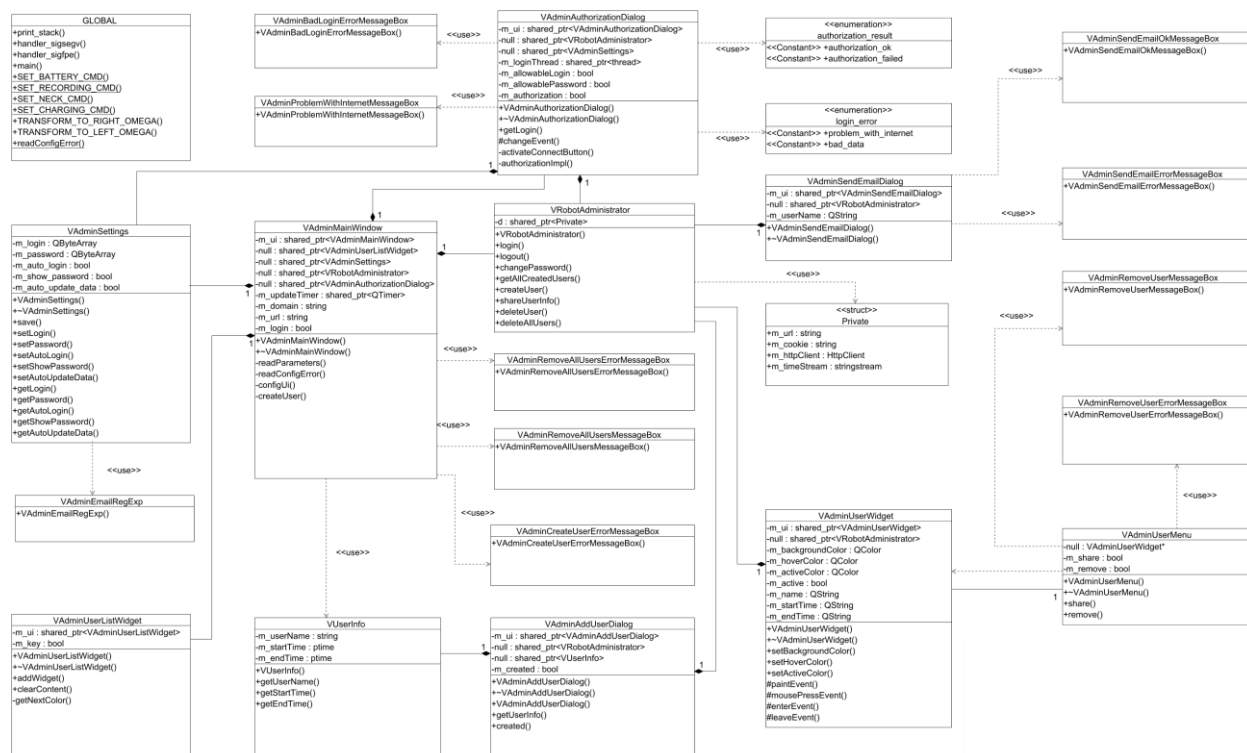


Рисунок 3.3 Диаграмма классов приложения vadmin

Приложение `vadmin` содержит классы:

- VAdminMainWindow – класс главного окна.
- VAdminSettings – класс для хранения настроек.
- VAdminUserListWidget – класс виджета для просмотра и взаимодействия со списком пользователей.
- VAdminAuthorizationDialog – класс диалога авторизации.
- VAdminAddUserDialog – класс диалога, добавления нового пользователя.
- VAdminSendEmailDialog – класс диалога, отправки email сообщения.
- VAdminUserWidget – класс виджета для работы с конкретным пользователем.
- VAdminEmailRegExp – класс регулярного выражения для проверки email.

- VUserInfo – класс информации и пользователя.
- VAdminBadLoginErrorMessageBox – класс сообщения об ошибке для неверного логина.
- VAdminProblemWithInternetMessageBox – класс сообщения об ошибке для проблем с интернетом.
- VRobotAdministrator – класс для отправки GET/POST запросов на сервер и получения информации.
- VAdminRemoveAllUsersErrorMessageBox – класс сообщения об ошибке для удаления всех пользователей.
- VAdminRemoveAllUserOkMessageBox – класс сообщения об успехе для удаления пользователя.
- VAdminCreateUserErrorMessageBox – класс сообщения об ошибке для создания пользователя.
- VAdminSendEmailOkMessageBox – класс сообщения об успехе отправки email сообщения.
- VAdminSendEmailErrorMessageBox – класс сообщения об ошибке отправки email сообщения.
- VRemoveUserErrorMessageBox – класс сообщения об ошибке удаления пользователя.
- VRemoveUserOkMessageBox – класс сообщения об успехе удаления пользователя.
- VAdminUserMenu – класс меню для VAdminUserWidget.

Также на диаграмме классов обозначены вспомогательные объекты: enums, constants.

4. Рабочий этап проектирования.

4.1. Диаграмма состояний сессии звонка.

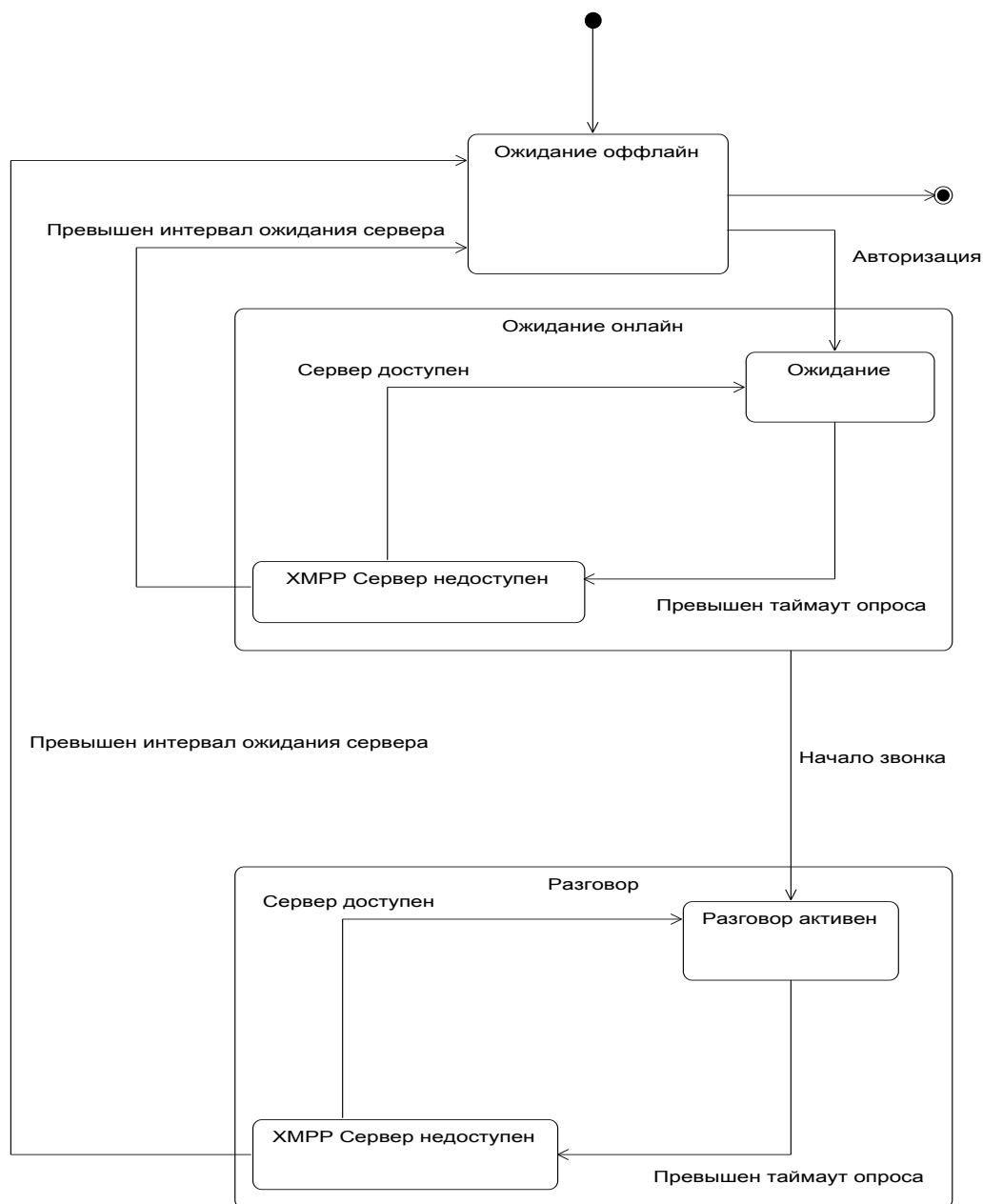


Рисунок 4.1 Диаграмма состояний сессии звонка

Диаграмма на рисунке 4.1 описывает алгоритм взаимодействия между клиентом и роботом с точки зрения процесса общения. Робот или клиент находятся оффлайн, после авторизации на сервере они переходят в состояние онлайн. Из ожидания онлайн клиент инициирует звонок и оба субъекта общения переходят в состояние разговора. Из состояния онлайн или разговора, любой субъект общения в случае долгого отсутствия связи с сервером переходит в состояние оффлайн.

4.2. Диаграмма состояний vrobot GUI.



Рисунок 4.2 Диаграмма состояний графического интерфейса vrobot

Диаграмма на рисунке 4.2 описывает набор состояний графического интерфейса приложения робота. После запуска робот находится в состоянии: диалог сессия общения, из него в случае разблокирования настроек интерфейс переключается в диалог администрирования. Из этого диалога пользователь может настроить интернет: Wi-Fi, Yota, - или обновить программное обеспечение. Также из диалога администрирования можно закрыть приложение или выполнить выключение, перезагрузку операционной системы робота. В любом из состояний настройки робот не принимает звонки.

4.3. Диаграмма состояний vclient GUI.

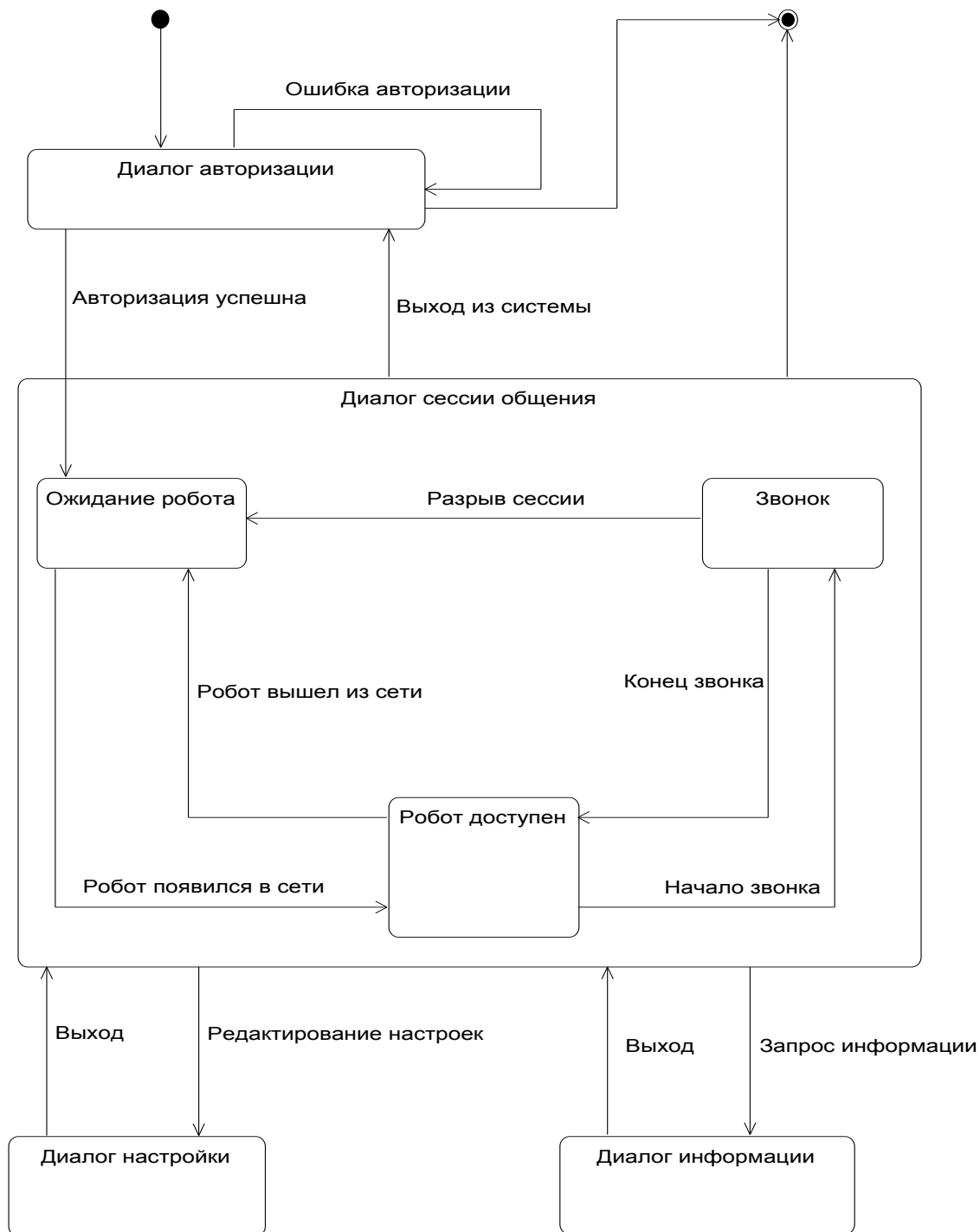


Рисунок 4.3 Диаграмма состояний графического интерфейса vrobot

Диаграмма на рисунке 4.3 описывает набор состояний графического интерфейса приложения клиента. После запуска клиент находится в состоянии ожидания авторизации, из него в случае успешной авторизации интерфейс переключается в диалог сессии общения. Из диалога общения пользователь может перейти в меню настройки, чтобы выбрать какой микрофон и камеру использовать. Также из диалога общения пользователь может запросить справочную информацию о программе.

4.4. Диаграмма деятельности обработки запросов приложением vreport-server.

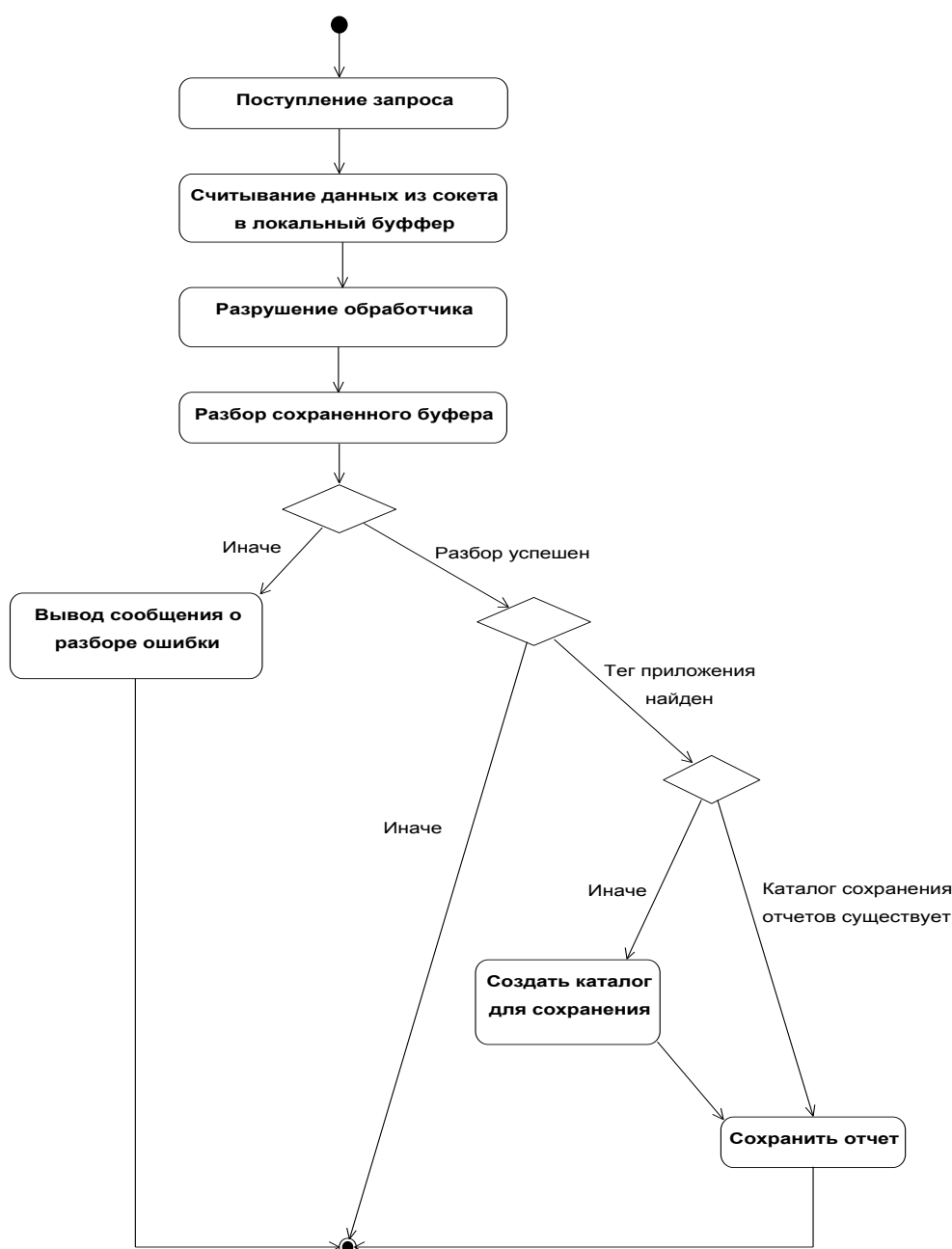


Рисунок 4.4 Диаграмма деятельности обработки запросов приложением vreport-server

Диаграмма на рисунке 4.4 описывает порядок обработки запроса с отчетом об ошибке. Из состояния поступление запроса, программа переходит в состояние считывания данных из буфера. После считывания всех данных программа начинает разрушать объект обработчик. В момент разрушения обработчика происходит XML разбор сохраненного буфера, если разбор успешен, приложение проверяет наличие тега приложения в разобранной структуре и сохраняет отчет в папку с именем приложения. Если тег приложения не найден или XML разбор прошел с ошибкой, программа сохраняет сообщение об ошибке.

4.5. Диаграмма состояний приложения vshare-server.

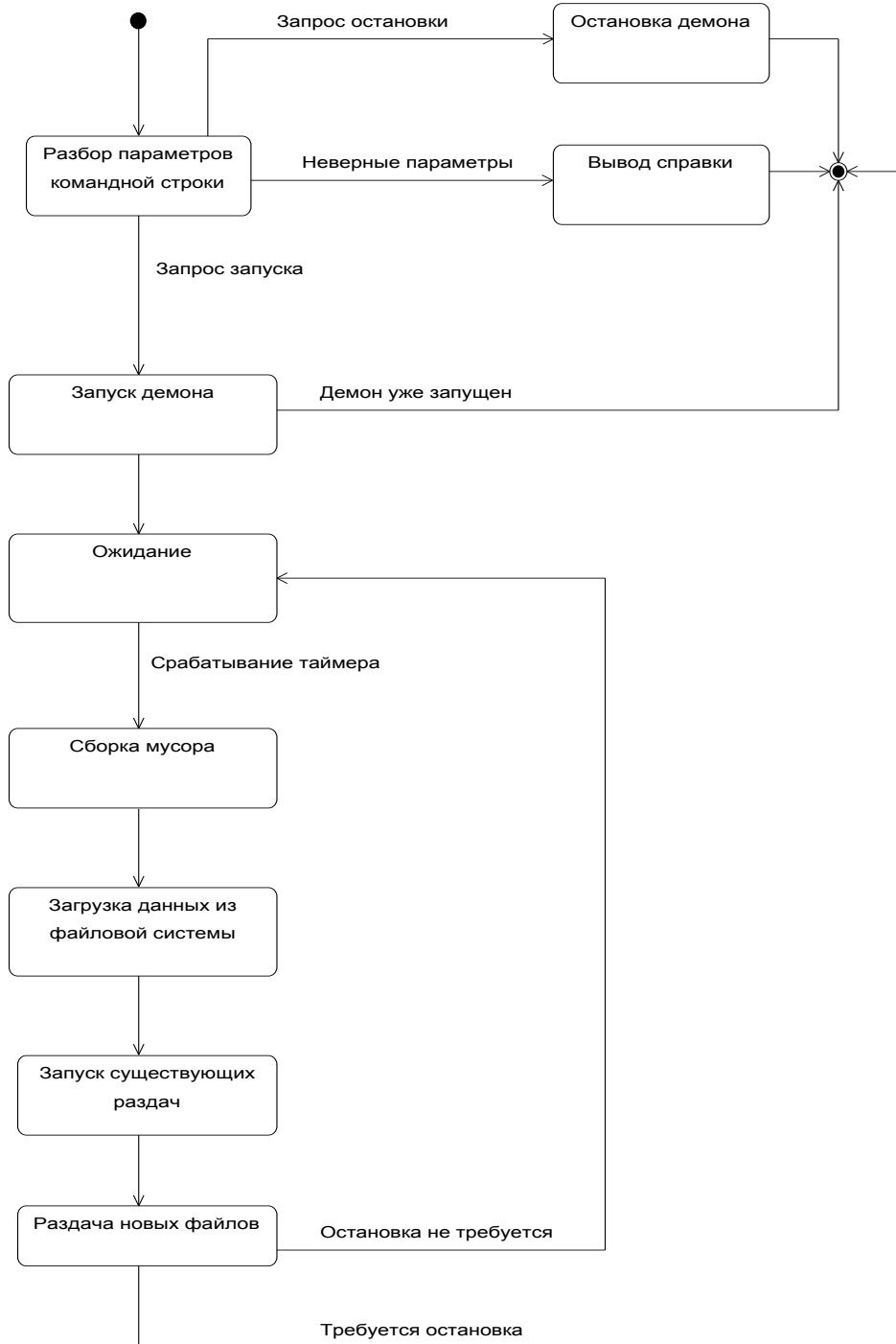


Рисунок 4.5 Диаграмма состояний приложения share-server

Диаграмма на рисунке 4.5 описывает алгоритм работы приложения по раздаче сохраненного видео. После разбора командной строки выполняется остановка или запуск демона. После приложение переходит в состояние ожидания, по срабатыванию таймера начинается цикл операций по порядку до состояния раздачи новых файлов. В ней происходит проверка, если требуется остановка – приложение завершает работу, иначе переходит в состояние ожидания.

4.6. Диаграмма состояний приложения vdatastore-server.

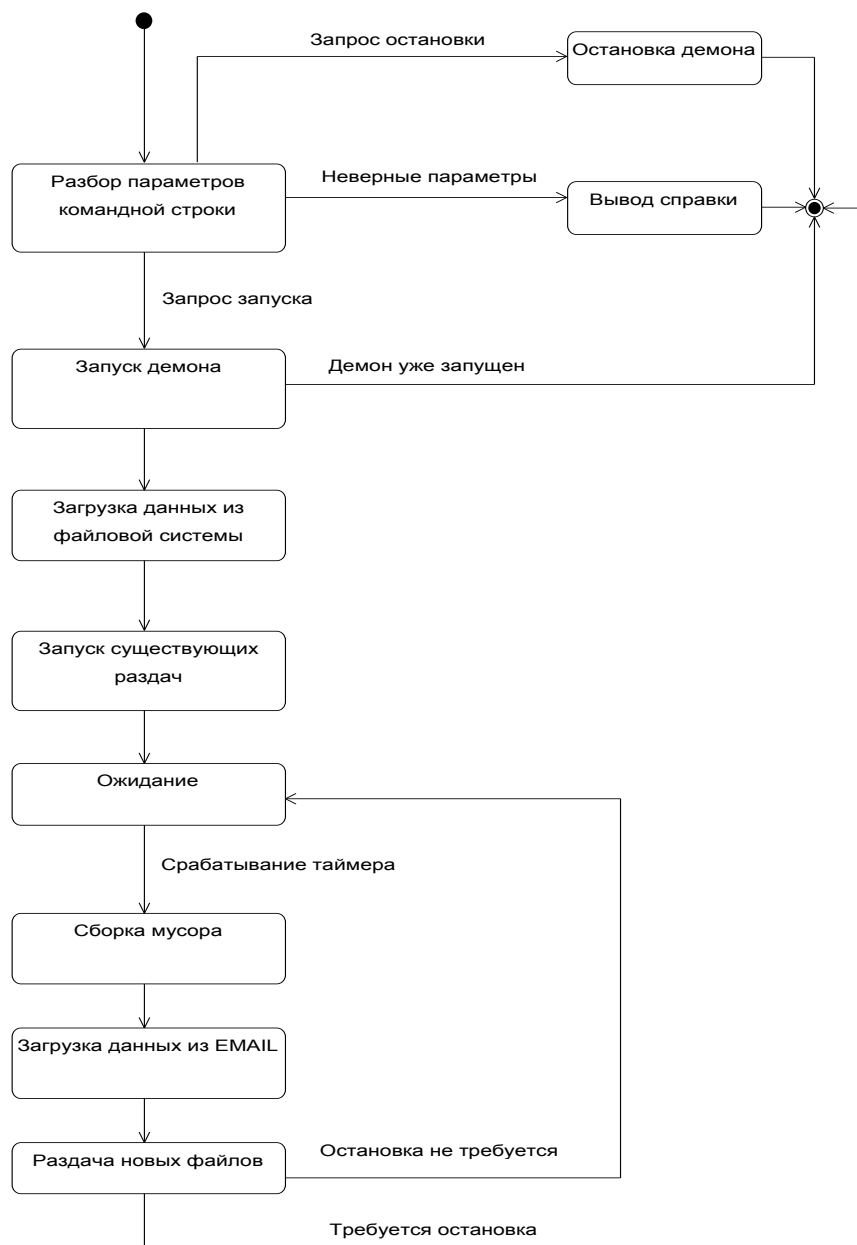


Рисунок 4.6 Диаграмма состояний приложения datastore-server

Диаграмма на рисунке 4.6 описывает алгоритм работы приложения по хранению сохраненного видео. После разбора командной строки выполняется остановка или запуск демона. После запуска происходит загрузка данных из файловой системы, запуск существующих раздач, после приложение переходит в состояние ожидания, по срабатыванию таймера начинается цикл операций по порядку до состояния раздачи новых файлов. В ней происходит проверка, если требуется остановка – приложение завершает работу, иначе переходит в состояние ожидания.

4.7. Диаграмма состояний управления движением робота.

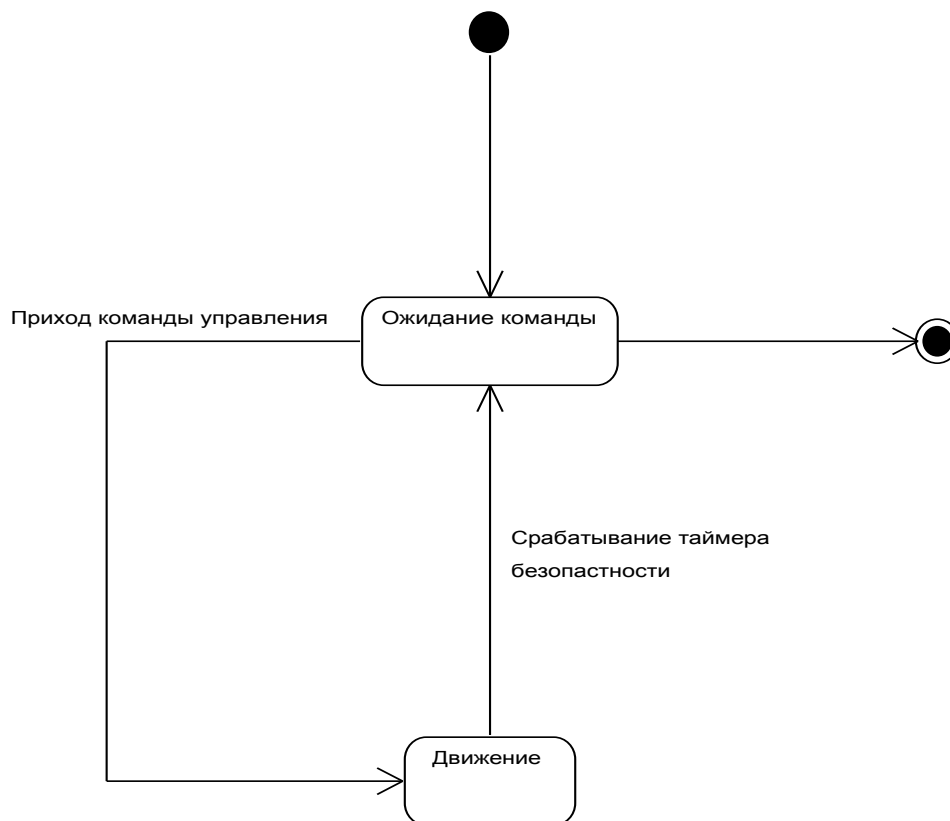


Рисунок 4.7 Диаграмма состояний управления движением робота

Диаграмма на рисунке 4.7 описывает алгоритм системы управления движением робота. Из начального состояния система переходит в состояние ожидание команды, когда приходит команда управления по сети, система управления устанавливает скорость на приводы робота, по срабатыванию таймера безопасности система управления переводит плату управления приводами в состояние сна и робот останавливается. Для того, чтобы робот ехал непрерывно, необходимо чтобы команды управления приходили не реже чем 200 миллисекунд. Данный алгоритм обеспечивает безопасность окружающих его пользователей. Если с сетью произошли неполадки, то робот быстро останавливается.

5. Исследовательская часть.

5.1. Постановка задачи.

Основной задачей дипломного проекта является разработка и реализация архитектуры системы управления роботом удаленного присутствия.

Одной из ответственных частей работы является выбор способа передачи данных: аудио, видео и команды управления. Предварительный обзор способов передачи данных: ffmpeg, RTSP-server, Jingle, - был проведен на этапе концептуального проектирования. В исследовательской части будет проведен анализ среднего времени задержки передачи данных.

5.2. Выбор способа оценки времени задержки.

Передаваемые данные можно разделить на три основные составляющие с их количественной оценкой ширины канала, требуемой для нормальной работы:

- Видео – 1000 кб / сек;
- Аудио – 50 кб / сек;
- Команды управления и телеметрия – 5 кб / сек.

Основной объем передаваемых данных составляет видео. Произведем расчет задержки передачи видео.

Оценку задержки передаваемого видео произведем с использованием следующего эксперимента: перед камерой робота располагаются электронные часы, изображение с робота передается клиенту на экран, клиент и робот находятся в одном помещении; взяв разницу во времени между временем с реальных часов и часов с экрана монитора можно получить задержку времени по передаче видео.

5.3. Построение диаграмм среднего времени задержки передачи данных.

Проведем четыре эксперимента по 10 измерений для каждого способа передачи данных, измерения будем проводить раз в 30 минут в течение рабочего дня, замер времени происходит через три минуты после начала сессии тестирования.

- Эксперимент 1:

Сервер установлен в дата-центре организации “Ашманов и Партнеры”, машина клиента и робота находятся в одной локальной сети (p2p возможен).

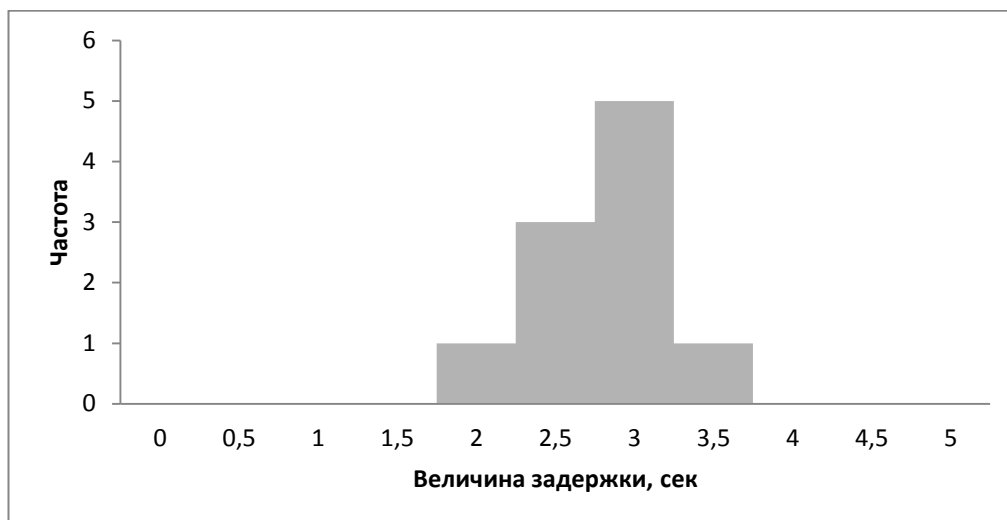


Рисунок 5.1 Гистограмма времени задержки передачи видео через ffserver для эксперимента 1

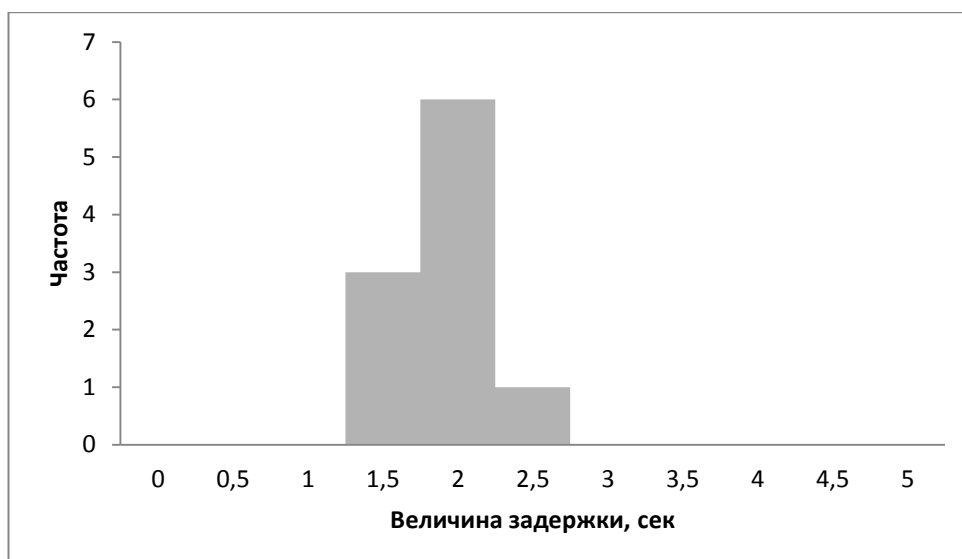


Рисунок 5.2 Гистограмма времени задержки передачи видео через RTSP-server эксперимент 1

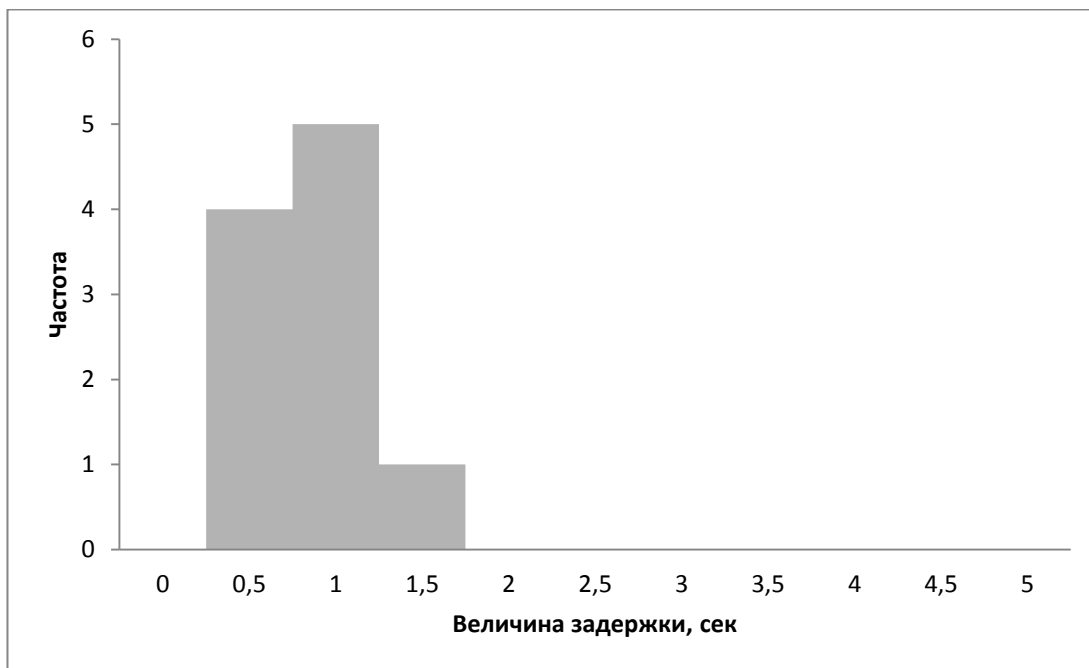


Рисунок 5.3 Гистограмма времени передачи видео через Jingle эксперимент 1

- Эксперимент 2:

Сервер установлен в дата-центре организации “Ашманов и Партнеры”, машина клиента и робота находятся в разных сетях (local internet 1 и local internet 2, с возможностью использовать р2р).

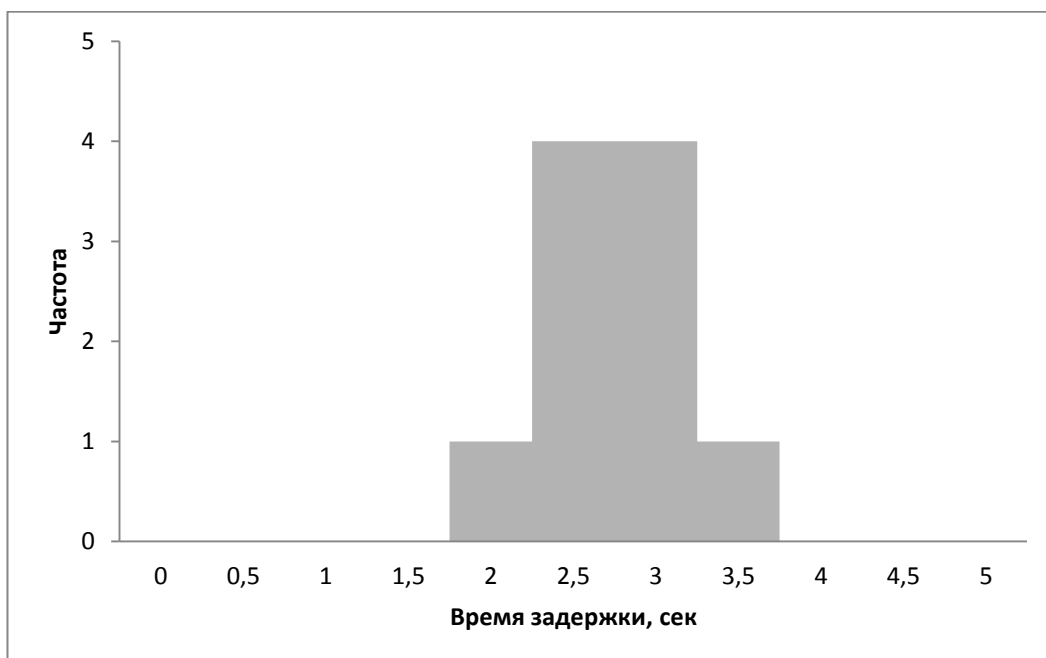


Рисунок 5.4 Гистограмма времени задержки передачи видео через ffmpeg для эксперимента 2

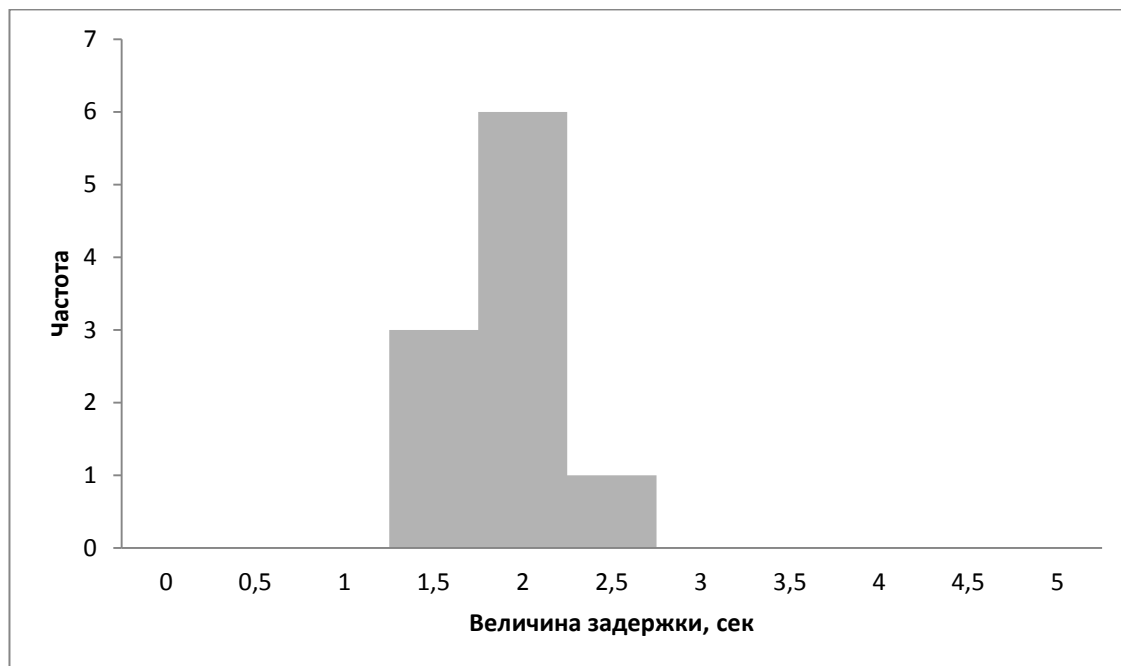


Рисунок 5.5 Гистограмма времени задержки передачи видео через RTSP-server эксперимент 2

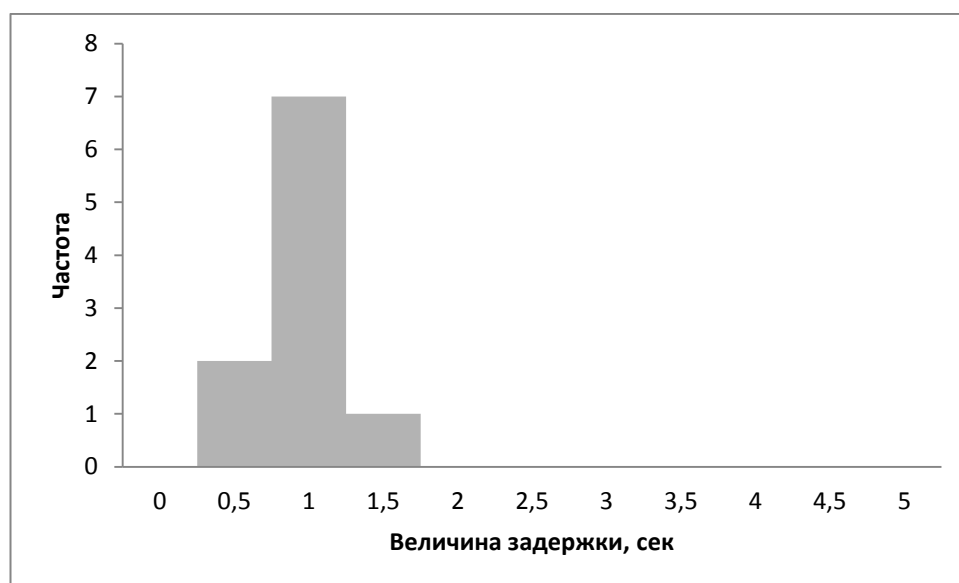


Рисунок 5.6 Гистограмма времени передачи видео через Jingle эксперимент 2

- Эксперимент 3:

Сервер установлен в дата-центре организации “Ашманов и Партнеры”, машина клиента и робота находятся в разных сетях (Yota и local internet, р2р возможен).

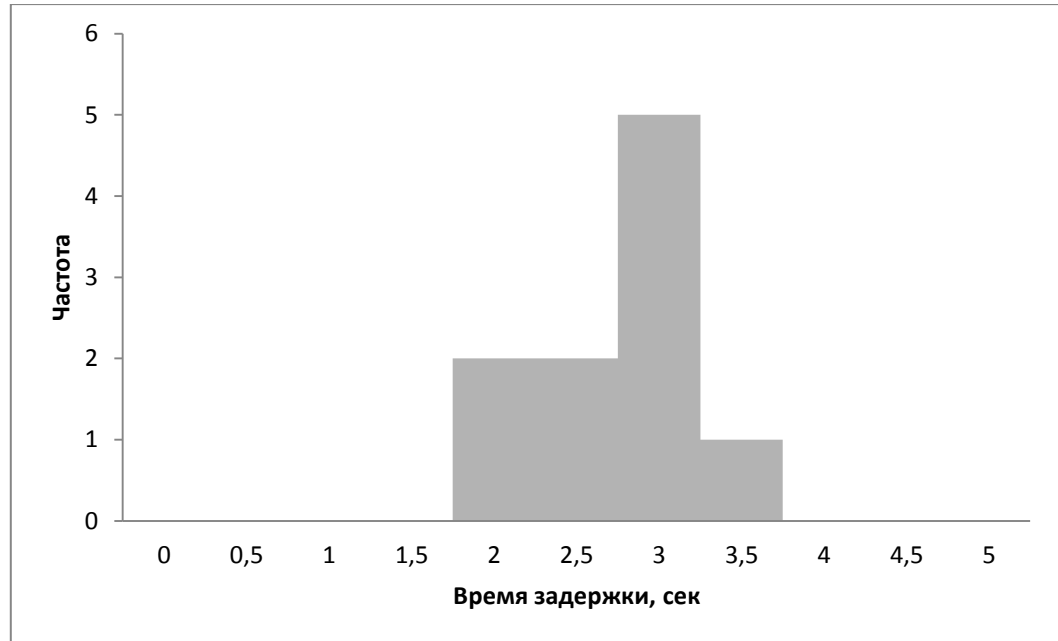


Рисунок 5.7 Гистограмма времени задержки передачи видео через ffserver для эксперимента 3

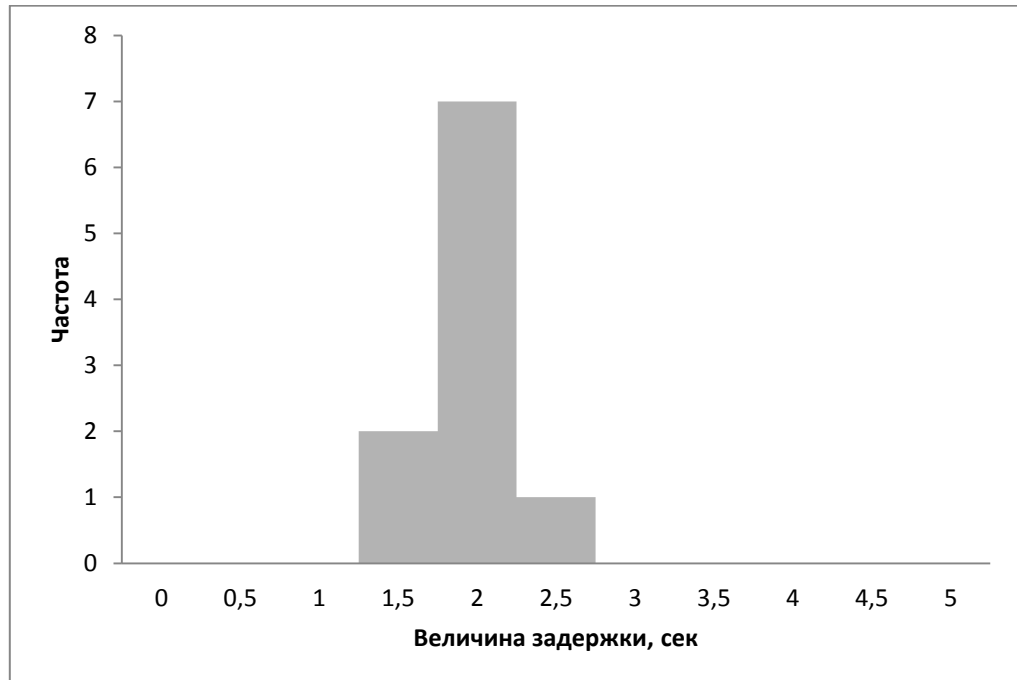


Рисунок 5.8 Гистограмма времени задержки передачи видео через RTSP-server эксперимент 3

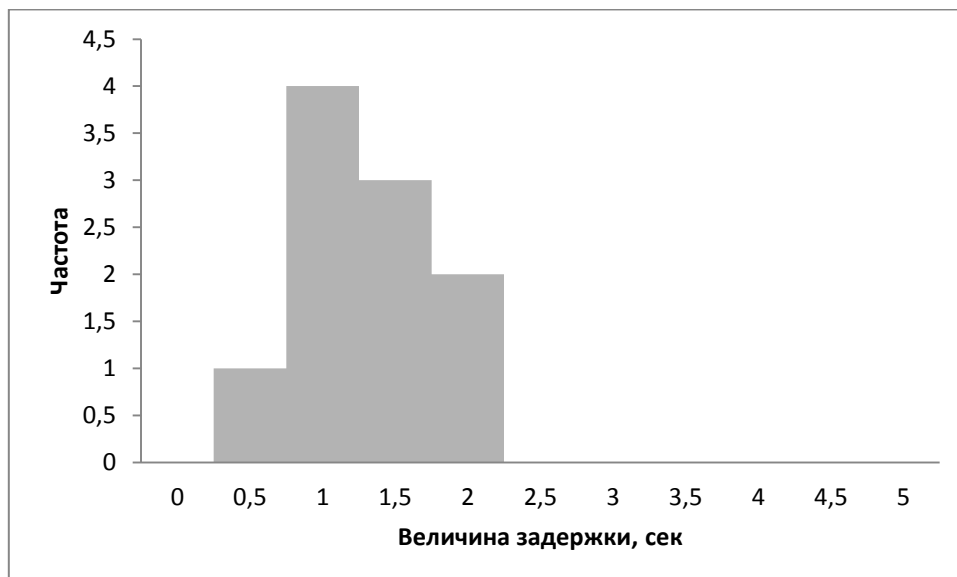


Рисунок 5.9 Гистограмма времени передачи видео через Jingle эксперимент 3

- Эксперимент 4:

Сервер установлен в дата-центре организации “Ашманов и Партнеры”, машина клиента и робота находятся в разных сетях (Yota и local internet, p2p отключен в настройках маршрутизатора).

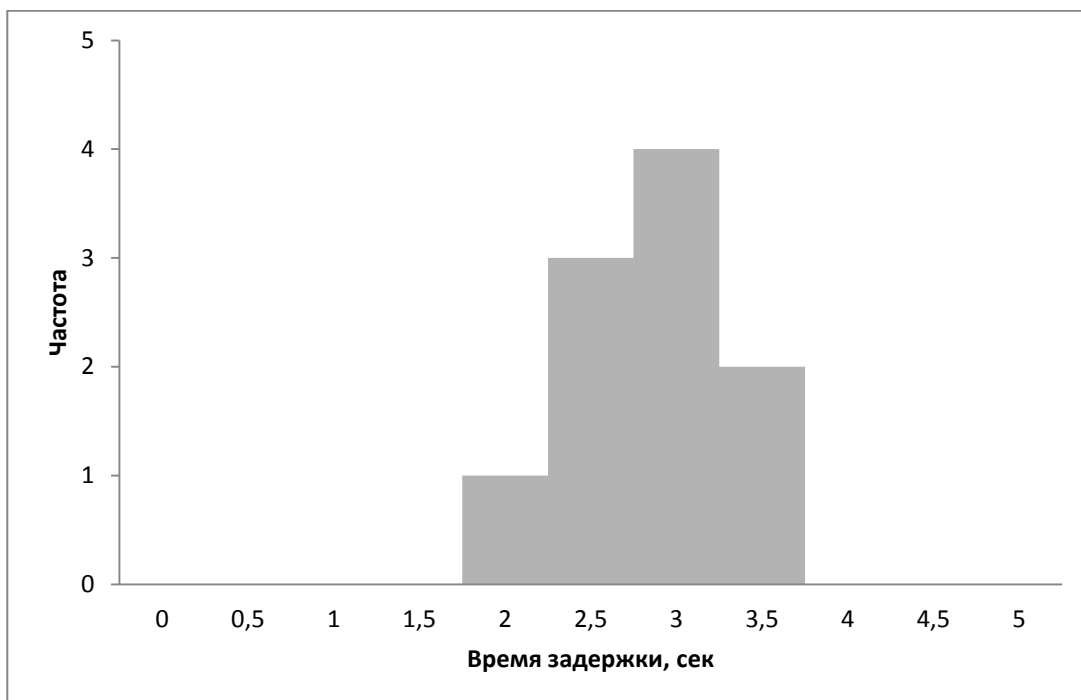


Рисунок 5.10 Гистограмма времени задержки передачи видео через ffmpeg для эксперимента 4

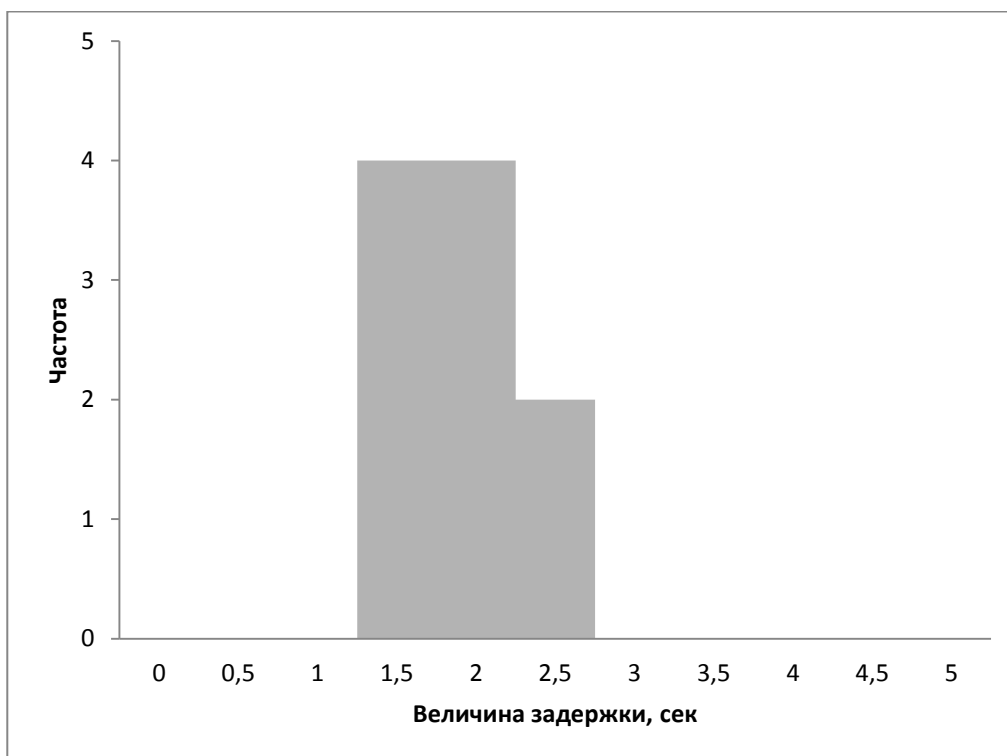


Рисунок 5.11 Гистограмма времени задержки передачи видео через RTSP-server
эксперимент 4

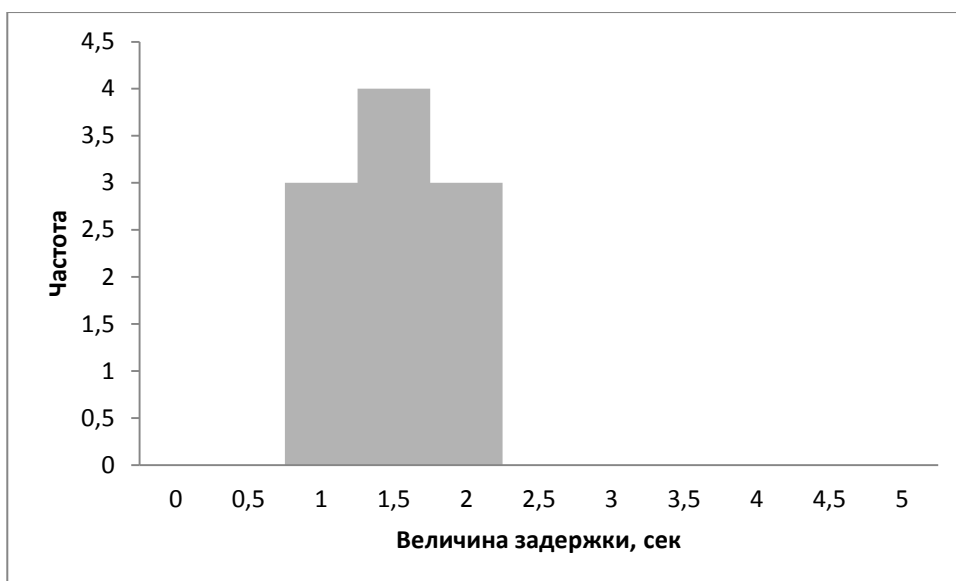


Рисунок 5.12 Гистограмма времени передачи видео через Jingle эксперимент 4

5.4. Итоги исследования.

Анализ диаграмм главы 5.3 показывает, что в тех случаях, когда возможно p2p соединение, Jingle показывает более низкую величину задержки, чем ffserver и RTSP-server. Это объясняется тем, что трафик идет напрямую между клиентами, а не через сервер. В том случае, когда p2p передача данных невозможна, Jingle и RTSP-server показывают аналогичные результаты. Jingle является более прогрессивным средством по передаче видео, чем ffserver и RTSP-server.

6. Организационно - экономическая часть.

6.1. Введение.

Современная инженерная деятельность предполагает не только разработку современных конструкций и технологий, но также и концентрацию усилий специалиста, позволяющую заранее определить возможный рынок реализации разработки, оценить ожидаемую прибыль. Поэтому важной составляющей любого инженерного проекта является раздел, посвященный анализу экономических характеристик и определению экономических параметров, позволяющих сделать вывод о возможности реализации инженерной мысли.

В организационно-экономической части дипломного проекта производится расчет затрат на разработку системы управления роботом удаленного присутствия.

6.2. Организация и планирование процесса разработки системы управления роботом удаленного присутствия.

Организация и планирование процесса разработки программного продукта или программного комплекса при традиционном методе планирования предусматривает выполнение следующих работ:

- формирование состава выполняемых работ и группировка их по стадиям разработки;
- расчет трудоемкости выполнения работ;
- установление профессионального состава и расчет количества исполнителей;
- определение продолжительности выполнения отдельных этапов разработки;
- построение календарного графика выполнения разработки;
- контроль выполнения календарного графика.

Таблица 6.1. Перечень стадий и состава работ.

Стадия разработки программного продукта	Состав выполняемых работ
1	2
Техническое задание	Постановка задач, выбор критериев эффективности. Разработка технико-экономического обоснования разработки. Определение состава пакета прикладных программ, состава и структуры информационной базы. Выбор языков программирования. Предварительный выбор методов выполнения работы. Разработка календарного плана выполнения работ.
Эскизный проект	Предварительная разработка структуры входных и выходных данных. Разработка общего описания алгоритмов реализации решения задач. Разработка пояснительной записки. Консультации разработчиков постановки задач. Согласование и утверждение эскизного проекта.
Технический проект	Разработка алгоритмов решения задач. Разработка пояснительной записки. Согласование и утверждение технического проекта. Разработка структуры программы. Разработка программной документации и передача ее для включения в технический проект. Уточнение структуры, анализ и определение формы представления входных и выходных данных. Выбор конфигурации технических средств.
Рабочий проект	Комплексная отладка задач и сдача в опытную эксплуатацию. Разработка проектной документации. Программирование и отладка программ. Описание контрольного примера. Разработка программной документации. Разработка, согласование программы и методики испытаний. Предварительное проведение всех видов испытаний.
Внедрение	Подготовка и передача программной документации для сопровождения с оформлением соответствующего Акта. Передача программной продукции в фонд алгоритмов и программ. Проверка алгоритмов и программ решения задач, корректировка документации после опытной эксплуатации программного продукта.

Трудоемкость разработки программной продукции зависит от ряда факторов, основными из которых являются следующие: степень новизны разрабатываемого программного комплекса, сложность алгоритма его функционирования, объем используемой информации, вид ее представления и способ обработки, а также уровень используемого алгоритмического языка программирования. Чем выше уровень языка, тем трудоемкость меньше.

По степени новизны разрабатываемый проект относится к группе новизны В - разработка программной продукции, имеющей аналоги.

По степени сложности алгоритма функционирования проект относится к 1 группе сложности - программная продукция, реализующая оптимизационные и моделирующие алгоритмы.

По функциональному назначению – перспективное.

6.2.1. Расчет трудоёмкости разработки системы управления роботом удаленного присутствия.

Расчет трудоёмкости разработки системы управления роботом удаленного присутствия ведётся по формуле:

$$\tau_{nn} = \tau_{mz} + \tau_{en} + \tau_{mn} + \tau_{pn} + \tau_{\varepsilon},$$

где:

τ_{mz} - трудоёмкость разработки технического задания;

τ_{en} - трудоёмкость разработки эскизного проекта системы управления роботом удаленного присутствия;

τ_{mn} - трудоёмкость разработки технического проекта системы управления роботом удаленного присутствия;

τ_{pn} - трудоёмкость разработки рабочего проекта системы управления роботом удаленного присутствия;

τ_{ε} - трудоёмкость внедрения разработанной системы.

6.2.2. Расчёт трудоёмкости разработки технического задания.

$$\tau_{мз} = T_{pz}^3 + T_{pn}^3, \text{ где}$$

T_{pz}^3 - затраты времени разработчика постановки задач на разработку ТЗ;

T_{pn}^3 - затраты времени разработчика программного обеспечения на разработку ТЗ.

$$T_{pz}^3 = t_3 \cdot K_{pz}^3;$$

$$T_{pn}^3 = t_3 \cdot K_{pn}^3, \text{ где}$$

t_3 - норма времени на разработку ТЗ на программный продукт в зависимости от функционального назначения и степени новизны разрабатываемого продукта;

K_{pz}^3 - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задач на стадии ТЗ;

K_{pn}^3 - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ТЗ.

Для разрабатываемой системы управления роботом удаленного присутствия:

$t_3 = 37$ человеко-дней, так как система относится к группе новизны В.

$K_{pz}^3 = 0,65$, так как совместная работа разработчика постановки задач с разработчиком ПО;

$K_{pn}^3 = 0,35$, так как совместная работа разработчика ПО с разработчиком постановки задач.

$$T_{мз} = 37 * (0,35 + 0,65) = 37 \text{ [чел.-дн.]}$$

6.2.3. Расчёт трудоёмкости разработки эскизного проекта.

$$\tau_{\text{эп}} = T_{\text{пз}}^{\text{э}} + T_{\text{пн}}^{\text{э}}, \text{ где}$$

$T_{\text{пз}}^{\text{э}}$ - затраты времени разработчика постановки задач на разработку эскизного проекта;

$T_{\text{пн}}^{\text{э}}$ - затраты времени разработчика программного обеспечения на разработку эскизного проекта.

Эти значения рассчитываются по формулам:

$$T_{\text{пз}}^{\text{э}} = t_{\text{э}} \cdot K_{\text{пз}}^{\text{э}};$$

$$T_{\text{пн}}^{\text{э}} = t_{\text{э}} \cdot K_{\text{пн}}^{\text{э}}, \text{ где}$$

$t_{\text{э}}$ - норма времени на разработку эскизного проекта на программный продукт в зависимости от функционального назначения и степени новизны разрабатываемого продукта;

$K_{\text{пз}}^{\text{э}}$ - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задач на стадии эскизного проекта;

$K_{\text{пн}}^{\text{э}}$ - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии эскизного проекта.

Для разрабатываемой системы:

$t_{\text{э}} = 77$ чел.-дней (группа новизны – В, функциональное назначение – перспективное, технико-экономическое, оперативное планирование);

$K_{\text{пз}}^{\text{э}} = 0,7$, так как совместная работа разработчика постановки задач с разработчиком ПО;

$K_{\text{пн}}^{\text{э}} = 0,3$, так как совместная работа разработчика ПО с разработчиком постановки задач.

$$T_{\text{ЭП}} = 77 * (0,3 + 0,7) = 77 \text{ [чел.-дн.]}$$

6.2.4. Расчёт трудоёмкости разработки технического проекта.

$$\tau_{mn} = (t_{pz}^m + t_{pn}^m) \cdot K_{\epsilon} \cdot K_p, \text{ где}$$

t_{pz}^m - норма времени, затрачиваемого на разработку технического проекта разработчиком постановки задач;

t_{pn}^m - норма времени, затрачиваемого на разработку технического проекта разработчиком программного обеспечения;

K_{ϵ} - коэффициент учёта вида используемой информации;

K_p - коэффициент учёта режима обработки информации.

$$K_{\epsilon} = \frac{K_n n_n + K_{nc} n_{nc} + K_{\bar{o}} n_{\bar{o}}}{n_n + n_{nc} + n_{\bar{o}}}, \text{ где}$$

K_n , K_{nc} , $K_{\bar{o}}$ - значения коэффициентов учёта вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно;

n_n , n_{nc} , $n_{\bar{o}}$ - количество наборов переменной, нормативно-справочной информации и баз данных соответственно.

Для разрабатываемой системы:

$t_{pz}^m = 130$ чел.-дней, $t_{pn}^m = 34$ чел.-дня, так как система решает задачи управления и имеет 7 форм входной информации (информация из БД, информация, получаемая из диалога с пользователем, и информация, получаемая в электронном виде из внешних источников, информация, получаемая по почте, информация, получаемая от оборудования) и 6 форм выходной информации (видео, аудио, текстовые отчеты, управляющее воздействие, электронные письма, документация на систему).

$K_p = 1,26$, так как группа новизны - В, а режим обработки информации – реальный масштаб времени.

Для группы новизны В:

$$K_n = 1,0, K_{nc} = 0,72, K_{\bar{o}} = 2,08,$$

$$n_n = 5, n_{nc} = 1, n_o = 3.$$

$$K_{\epsilon} = (1,0 * 5 + 0,72 * 1 + 2,08 * 3) / (5 + 1 + 3) = 1,33$$

$$T_{mn} = (130 + 34) * 1,33 * 1,26 = 275 \text{ [чел.-дн.]}$$

6.2.5 *Расчёт трудоёмкости разработки рабочего проекта.*

$$\tau_{pn} = K_{\kappa} \cdot K_p \cdot K_{\text{я}} \cdot K_{\text{з}} \cdot K_{\text{иа}} \cdot (t_{pz}^p + t_{pn}^p), \text{ где}$$

t_{pz}^p, t_{pn}^p - норма времени, затраченного на разработку рабочего проекта на алгоритмическом языке высокого уровня разработчиком постановки задач и разработчиком программного обеспечения соответственно;

K_{κ} - коэффициент учета сложности контроля информации;

$K_{\text{я}}$ - коэффициент учета уровня используемого алгоритмического языка программирования;

$K_{\text{з}}$ - коэффициент учета степени использования готовых программных модулей;

$K_{\text{иа}}$ - коэффициент учета вида используемой информации и сложности алгоритма программного продукта.

$$K_{\text{иа}} = \frac{K'_n n_n + K'_{nc} n_{nc} + K'_o n_o}{n_n + n_{nc} + n_o}, \text{ где}$$

K'_n, K'_{nc}, K'_o - значения коэффициентов учета сложности алгоритма программного продукта и вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно.

Для разрабатываемой системы:

$K_{\kappa} = 1,07$, так как степень сложности контроля входной информации - 11, а степень сложности контроля выходной информации – 22;

$$K_p = 1,26$$

$K_я = 1$, так как язык программирования C++ – язык высокого уровня;

$K_з = 0,7$ (использование готовых программных модулей составляет около 40%);

$K_n = 1,0$, $K_{nc} = 0,72$, $K_б = 2,08$,

$t_{pz}^p = 45$ чел.-дней,

$t_{pn}^p = 240$ чел.-дней;

$K_{ua} = 1,33$;

$T_{rp} = 1,07 * 1,00 * 0,7 * 1,26 * 1,33 * (45+240) = 358$ [чел.-дн.]

6.2.6 Расчет трудоемкости внедрения программного продукта.

$$\tau_с = (t_{pz}^с + t_{pn}^с) \cdot K_к \cdot K_p \cdot K_з$$

$t_{pz}^с$, $t_{pn}^с$ - норма времени, затрачиваемого разработчиком постановки задач и разработчиком программного обеспечения соответственно на выполнение процедур внедрения программного продукта.

Для разрабатываемой системы:

$t_{pz}^с = 45$ чел.-дней;

$t_{pn}^с = 51$ чел.-дней;

$T_в = (45 + 51) * 1,07 * 1,26 * 0,7 = 91$ [чел.-дн.]

6.2.7 Суммарная трудоемкость

Суммарная трудоёмкость разработки и внедрения системы:

$T_{пп} = 37 + 77 + 275 + 358 + 91 = 838$ [чел.-дн.]

Планирование и контроль над ходом выполнения проекта по разработке ПП проводят по календарному графику выполнения работ. Проект осуществляет небольшой (2 чел.),

стабильный по составу коллектив исполнителей, поэтому для этих целей можно использовать ленточный график.

Для того, чтобы определить продолжительность всех работ по созданию ПП, рассчитаем продолжительность каждого этапа, исходя из соответствующих трудоемкостей и количества занятых участников на каждом этапе.

Расчет производится по формуле:

$$T_i = \frac{\tau_i + Q}{n_i}$$

где τ_i - трудоемкость i -й работы, чел.-дн.;

Q - трудоемкость дополнительных работ, выполняемых исполнителем, чел.-дн.;

n_i - количество исполнителей, выполняющих i -ю работу, чел.

Так как дополнительные работы на всех этапах отсутствуют, то получаем:

$$T_{мз} = 37 / 2 = 18,5 \text{ дней}$$

$$T_{эп} = 77 / 2 = 38,5 \text{ дней}$$

$$T_{мп} = 275 / 2 = 137,5 \text{ дней}$$

$$T_{рп} = 358 / 2 = 179 \text{ дней}$$

$$T_{внедр} = 91 / 2 = 45,5 \text{ дней}$$

Общее время выполнения проекта:

$$T_{\text{ит}} = 18,5 + 38,5 + 137,5 + 179 + 45,5 = 419 \text{ дней.}$$

6.3. Оценка стоимости разработки и внедрения системы управления роботом удаленного присутствия.

Для определения стоимости работ, необходимо на основании плановых сроков выполнения работ и численности исполнителей, рассчитать общую сумму затрат на разработку программного продукта.

Себестоимость ПП представляет собой стоимостную оценку используемых в процессе производства продукции работ, услуг, природных ресурсов, сырья материалов, топлива, энергии, основных фондов, трудовых ресурсов, а также других затрат на ее производство и реализацию.

Затраты, образующие себестоимость ПП, группируются в соответствии с их экономическим содержанием по следующим элементам:

- Расчёт основной заработной платы.
- Расчёт дополнительной заработной платы.
- Отчисления в социальный фонд.
- Накладные расходы.
- Амортизация оборудования.

6.3.1. Специальное оборудование (амортизация).

В статье учитываются суммарные затраты на приобретение или проектирование специального оборудования:

$$C_{co} = \sum_i \frac{C_{bi} \cdot \alpha_i}{100 \cdot F_d} t_i, \text{ где}$$

C_{bi} - балансовая цена i -го вида оборудования, руб.;

α_i - норма годовых амортизационных отчислений для оборудования i -го вида, %;

t_i - время использования i -го вида оборудования при выполнении разработки, ч.

F_d - действительный годовой фонд рабочего времени сотрудника, ч. (5-ти дневная неделя, 8-и часовой рабочий день – 2080 ч.);

Табл. 6.2. Специальное оборудование и ПО, используемое при разработке

№ п/п	Наименование	Единица измерения	Количество	Цена за единицу	Сумма, руб.
1	ПЭВМ	шт.	2	25000	50000

Затраты на амортизацию оборудования (ПЭВМ):

$$C_{oo} = 50\,000 \cdot 20 \cdot 340,88 \cdot 8 / (100 \cdot 2080) = 13110 \text{ руб.}$$

6.3.2. Основная заработная плата.

В данную статью включается основная заработная плата всех исполнителей, непосредственно занятых разработкой данного программного продукта с учетом их должностных окладов и времени участия. Расчет проводится по формуле:

$$C_{zo} = \sum_i \frac{z_i}{d} \tau_i, \text{ где}$$

z_i - месячный оклад i -го исполнителя, руб;

d - среднее количество рабочих дней в месяце, $d = 21$ день;

τ_i - трудоемкость работ, выполняемых i -м исполнителем, чел.-дн. (значения получены из таблицы).

Расчет затрат на оплату труда для каждого из операторов:

$$C_{zoi} = 10\,000 \cdot 419 / 21 = 199524 \text{ [руб.]}$$

Суммарная заработная плата равна:

$$C_{zo} = 2 * 199524 = 399048 \text{ [руб.]}$$

6.3.3. Дополнительная заработная плата.

В данной статье также учитываются выплаты непосредственным исполнителям за время, не проработанное на производстве, в том числе: оплата очередных отпусков, компенсация за недоиспользованный отпуск, оплата льготных часов подросткам и др. Дополнительная заработная плата рассчитывается по формуле:

$$C_{zd} = C_{zo} \cdot A_d,$$

где A_d - коэффициент отчислений на дополнительную заработную плату.

$$A_d = 0.2$$

$$C_{зд} = 399048 * 0.2 = 79810 \text{ [руб.]}$$

6.3.4. Отчисления в социальные фонды и страхования от несчастных случаев.

Страховые взносы включают в себя отчисления собственно на социальное страхование, а также на обязательное медицинское страхование и в пенсионный фонд, итого 30% от основной заработной платы. Кроме того, к затратам относятся отчисления на страхование от несчастных случаев, что составляет 0,2% от заработной платы (т.к. отрасль информационно-вычислительного обслуживания относится к 1 классу профессионального риска).

Страховые взносы и страхование от несчастного случая составляют:

$$C_{сс} = (399048 + 79810) * (30 + 0,2) / 100 = 144615 \text{ [руб.]}$$

6.3.5. Накладные расходы.

В данную статью входят другие затраты, входящие в состав себестоимости продукции, но не относящиеся к ранее перечисленным элементам затрат. Сюда можно отнести такие виды затрат: ежемесячная арендная плата за помещение, оплата электроэнергии, оплата отопления, оплата телефона и других услуг связи и т.д.

Накладные расходы для компании-разработчика составляют около 200% от суммы общей заработной платы:

$$C_H = \alpha_H \cdot C_{зо}, \text{ где}$$

α_H - коэффициент накладных расходов, $\alpha_H = 2$.

$$C_H = 2 * 399048 = 798096 \text{ [руб.]}$$

Результаты расчетов затрат на разработку программного продукта приведены в таблице:

Табл. 6.3. Смета затрат на разработку системы

№ п/п	Наименование статьи	Сметная себестоимость, руб.
1	Специальное оборудование (амортизация)	13 110
2	Основная заработная плата	399 048
3	Дополнительная заработная плата	79 810
4	Отчисления в фонды	144 615
5	Накладные расходы (включая расходные материалы)	798 096

Итого: 1 434 679 рублей

6.3.6. Цена разработки и внедрения системы управления роботом удаленного присутствия.

Цена разработки и внедрения системы управления роботом удаленного присутствия рассчитывается следующим образом:

$$Ц = K \times C + Пр$$

где C - затраты на разработку ПП.

K - коэффициент учета затрат на изготовление опытного образца ПП как продукции производственно-технического назначения ($K=1,1$).

$Пр$ - нормативная прибыль, рассчитываемая по формуле:

$$Пр = C \cdot \rho_H / 100,$$

где ρ_H - норматив рентабельности, 30 %;

$$Пр = 1\,434\,679 \cdot 30 / 100 = 430\,404 \text{ руб.}$$

$$Ц = 1,1 \cdot 1\,434\,679 + 430\,404 = 2\,008\,551 \text{ руб.}$$

6.4. Заключение организационно-экономической части.

В рамках организационно-экономической части был спланирован календарный график проведения работ по созданию системы управления роботом удаленного присутствия, а также были проведены расчеты по трудозатратам. Были исследованы и рассчитаны следующие статьи затрат: затраты на расходные материалы и специальное оборудование; основная и дополнительная заработная плата исполнителей; отчисления на пенсионное и социальное страхование; накладные расходы.

В результате расчетов было получено общее время выполнения проекта, которое составило 419 дней, получены данные по суммарным затратам на создание системы управления роботом удаленного присутствия, которые составили 1 434 679 рублей.

В результате расчетов была определена цена создания данной системы, которая составила 2 008 551 рублей.

7. МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА И ТЕХНИКЕ БЕЗОПАСНОСТИ.

7.1. Опасные и вредные факторы.

7.1.1. *Опасные факторы:*

- пожарная опасность, обусловленная наличием на рабочем месте мощного источника энергии;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

7.1.2. *Вредные факторы:*

7.1.2.1. *Физические:*

- повышенные уровни электромагнитного излучения;
- повышенный уровень статического электричества;
- повышенные уровни запыленности воздуха рабочей зоны;
- пониженная или повышенная влажность воздуха рабочей зоны;
- пониженная или повышенная подвижность воздуха рабочей зоны;
- повышенный или пониженный уровень освещенности;
- повышенный уровень прямой блескости;
- повышенный уровень отраженной блескости;
- неравномерность распределения яркости в поле зрения;
- повышенная яркость светового изображения;
- повышенный уровень пульсации светового потока;

7.1.2.2. *Психофизиологические:*

- напряжение зрения;
- напряжение внимания;
- интеллектуальные нагрузки;
- эмоциональные нагрузки;
- длительные статические нагрузки;
- монотонность труда;
- большой объем информации, обрабатываемой в единицу времени;
- нерациональная организация рабочего места.

7.1.2.3. Биологические:

- повышенное содержание в воздухе рабочей зоны микроорганизмов.

7.2. Требования к помещениям для работы с ПЭВМ.

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке.

Естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электроннолучевой трубки (ЭЛТ) должна составлять не менее 6 м², в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м². При использовании ПЭВМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м² на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

Для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка - 0,7 - 0,8; для стен - 0,5 - 0,6; для пола - 0,3 - 0,5.

Полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

Не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ.

7.2.1. Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ.

В производственных помещениях, в которых работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений. На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, соответствующем требованиям указанных выше нормативов.

В производственных помещениях, в которых работа с использованием ПЭВМ является основной, величины показателей микроклимата на рабочих местах должны соответствовать действующим санитарным нормам (СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений). При использовании данной подсистемы характер труда оператора ПЭВМ относится к категории 1а, т.к. работы производятся сидя и не требуют физического напряжения, расход энергии составляет до 120 ккал/ч, следовательно, величины показателей микроклимата на рабочих местах в ПЭО должны соответствовать оптимальным величинам показателей для категории работ 1а (см. Таблица 7.1).

Таблица 7.1. Оптимальные величины показателей микроклимата.

Период года	Категория работ	Температура воздуха, ° С не более	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – 1а	22-24	40-60	0,1
Теплый	Легкая – 1а	23-25	40-60	0,1

Уровни положительных и отрицательных аэроионов в воздухе помещений, где расположены ПЭВМ, должны соответствовать действующим санитарно-эпидемиологическим нормативам (см. Таблица 7.2).

Таблица 7.2. Уровни положительных и отрицательных аэроионов.

Уровни	Число ионов в 1 см. куб. воздуха	
	n+	n-
Минимально необходимые	400	600
Оптимальные	1500-3000	3000-5000
Максимально допустимые	50000	50000

Содержание вредных химических веществ в воздухе производственных помещений, в которых работа с использованием ПЭВМ является основной, не должно превышать предельно допустимых концентраций вредных веществ в воздухе рабочей зоны в соответствии с действующими гигиеническими нормативами (Гигиенические нормативы ГН 2.1.6.13 1338-03 Предельно допустимые концентрации (ПДК) загрязняющих веществ в атмосферном воздухе населенных мест). ПДК среднесуточная: окись углерода – 3,0 мг/м³, окислы азота – 0,04 мг/м³.

Данный микроклимат может обеспечиваться системой общеобменной приточно-вытяжной вентиляции. В приточной ветви нужно предусмотреть использование фильтров очистки воздуха, для удаления из воздуха вредных веществ перед его подачей на рабочие места. Для повышения влажности воздуха в помещениях с ВДТ и ПЭВМ следует применять увлажнители воздуха, заправляемые ежедневно дистиллированной или прокипяченной питьевой водой. Также в помещениях, оборудованных ПЭВМ, должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

Для поддержания необходимого уровня ионизации необходимо применять ионизаторы воздуха (например, ионизатор «Люстра Чижевского»).

7.2.2. Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ.

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать предельно допустимых значений, установленных для данных видов работ в соответствии с действующими санитарно-эпидемиологическими нормативами (*СанПиН 2.2.2/2.4.1340-03*

Гигиенические требования к персональным электронно-вычислительным машинам и организации работы) (см. Таблица 7.3).

Таблица 7.3. Допустимые значения уровней шума.

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука в дБА
31,5	63	25	250	500	1000	2000	00	00	
86	71	61	54	49	45	42	40	38	50

Снизить уровень шума в помещениях с ВДТ и ПЭВМ можно использованием подвесных акустических звукопоглощающих панелей с максимальными коэффициентами звукопоглощения в области частот 250-1000 Гц (источниками шума являются операторы ПЭВМ). Дополнительным звукопоглощением служат однотонные занавеси из плотной ткани, гармонирующие с окраской стен и подвешенные в складку на расстоянии 15 - 20 см от ограждения. Ширина занавеси должна быть в 2 раза больше ширины окна.

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ.

В помещениях, в которых работа с ПЭВМ является основной, уровень вибрации на рабочих местах не должен превышать допустимых значений для жилых и общественных зданий в соответствии с действующими санитарно-эпидемиологическими нормативами (Санитарные нормы СН 2.2.4/2.1.8.566-96 Производственная вибрация, вибрация в помещениях жилых и общественных зданий) (см. Таблица 7.4).

Таблица 7.4. Допустимые значения уровней вибрации.

Среднегеометрические частоты октавных полос, Гц	Допустимые значения оси X, Y			
	по виброускорению		по виброскорости	
	$\text{м/с}^2 \times 10^{-3}$	дБ	$\text{м/с} \times 10^{-3}$	дБ
2	10	80	0,79	84
4	11	81	0,45	79
8	14	83	0,28	75
16	28	89	0,28	75
31,5	56	95	0,28	75
63	110	101	0,28	75
Корректированные значения и их уровни	10	80	0,28	75

Для снижения вибрации в помещениях оборудование, аппараты и приборы, являющиеся источниками вибрации, необходимо устанавливать на амортизирующие прокладки. Также могут быть использованы средства индивидуальной защиты.

7.2.3. Требования к освещению на рабочих местах, оборудованных ПЭВМ.

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, и естественный свет падал преимущественно слева.

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м².

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м² и яркость потолка не должна превышать 200 кд/м².

Показатель дискомфорта в помещениях – не более 15.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м², защитный угол светильников должен быть не менее 40 градусов.

Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не

должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования – 10:1.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей.

Применение светильников без рассеивателей и экранирующих решеток не допускается.

При отсутствии светильников с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трехфазной сети.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору.

Коэффициент запаса (K_z) для осветительных установок общего освещения должен приниматься равным 1,4.

Коэффициент пульсации не должен превышать 5%.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

7.2.4. Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ.

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей, представлены ниже (см. Таблица 7.5).

Таблица 7.5.

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах.

Наименование параметров		ВДУ
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

Основным источником электромагнитного излучения на рабочем месте оператора компьютера является монитор. Величина излучения монитора во-многом зависит от его модели, среди наиболее безопасных необходимо выделить мониторы с маркировкой “Low Radiation”, компьютеры с жидкокристаллическим экраном и самые безопасные – мониторы с установленной защитой по методу замкнутого металлического экрана. Рекомендуется использовать подобные видеотерминалы.

В случае использования видеотерминалов с ЭЛТ для уменьшения воздействия электромагнитных полей на человека устанавливаются поглощающие или отражающие экраны перед видеомонитором. Современный качественный защитный экран позволяет уменьшить уровень электромагнитного излучения и электростатического поля в 10-15 раз и более. Кроме того, существуют специальные очки для защиты глаз от электромагнитного излучения.

7.2.5. Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах.

Предельно допустимые значения визуальных параметров ВДТ, контролируемые на рабочих местах, представлены в Таблица 7.6

Таблица 7.6.

N	Параметры	Допустимые значения
1	Яркость белого поля	Не менее 35 кд/кв.м
2	Неравномерность яркости рабочего поля	Не более +-20%
3	Контрастность (для монохромного режима)	Не менее 3:1
4	Временная нестабильность изображения (мелькания)	Не должна фиксироваться
5	Пространственная нестабильность изображения (дрожание)	Не более $2 \times 10(-4L)$, где L - проектное расстояние наблюдения, мм

Для дисплеев на ЭЛТ частота обновления изображения должна быть не менее 75 Гц при всех режимах разрешения экрана, гарантируемых нормативной документацией на конкретный тип дисплея, и не менее 60 Гц для дисплеев на плоских дискретных экранах (жидкокристаллических, плазменных и т.п.).

Обеспечить выполнение этих требований, можно соответствующим образом выбирая компьютерную технику, а также используя различные защитные экраны.

7.2.6. Общие требования к организации рабочих мест пользователей ПЭВМ.

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 - 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

7.2.7. Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей.

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

Конструкция рабочего стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400 - 550 мм и углам наклона вперед до 15 град, и назад до 5 град.;

- высоту опорной поверхности спинки 300 \pm 20 мм, ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах \pm 30 градусов;
- регулировку расстояния спинки от переднего края сиденья в пределах 260 - 400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной - 50 - 70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 \pm 30 мм и внутреннего расстояния между подлокотниками в пределах 350 - 500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

7.2.8. Электробезопасность рабочего помещения.

Работа с разработанным ПО должна проводиться в офисных помещениях, относящихся, согласно ПУЭ (правилам устройства электроустановок), к классу помещений «без повышенной опасности», т.к. в помещении нет условий повышенной и особой опасности.

Питание устройств ПЭВМ осуществляется от однофазной сети переменного тока с частотой 50 Гц, и напряжением 220 В.

В целях обеспечения необходимой электробезопасности при проведении работ в помещениях с ПЭВМ, необходимо выполнять следующие требования:

- для обеспечения работы операторов ПЭВМ необходимо исключить возможность случайного соприкосновения людей с токонесущими частями оборудования. Это достигается путем изоляции токоведущих частей ЭВМ и приборов и размещения их в недоступных зонах;
- не оставлять ЭВМ и другое оборудование под напряжением без наблюдения;
- не подключать разъёмы кабелей ЭВМ при включении напряжения в сети;
- сеть электропитания технических средств должна обеспечивать защитное заземление технических средств, предусмотренное их технической документацией.

7.2.9. Требования пожарной безопасности.

Помещение, где функционирует разработанная подсистема, относится к категории В — пожароопасные помещения, в котором имеются твердые сгораемые вещества, способные только гореть, но не взрываться при контакте с кислородом воздуха. Наиболее вероятной причиной пожара является неисправность электрооборудования и электросетей. При эксплуатации ЭВМ возможны возникновения следующих аварийных ситуаций: короткие замыкания, перегрузки, повышение переходных сопротивлений в электрических контактах, перенапряжение, возникновение токов утечки. При возникновении аварийных ситуаций происходит резкое выделение тепловой энергии, которая может явиться причиной возникновения пожара. Требования к пожаробезопасности зданий и сооружений определяются согласно СНиП 21.01-97.

Для снижения вероятности возникновения пожара необходимо проводить различные профилактические мероприятия:

- организационные — правильная эксплуатация электрооборудования, правильное содержание зданий и помещений;
- технические — соблюдение противопожарных правил и норм, норм при проектировании зданий, при устройстве отопления, вентиляции освещения, правильное размещение оборудования;
- мероприятия режимного характера — запрещение курения в неустановленных местах и т.д.;
- эксплуатационные — своевременные профилактические осмотры и ремонт неисправного электрооборудования.

Для снижения вероятности возникновения и распространения пожара на ранней стадии необходимо:

- установить пожарную сигнализацию с системой оповещения работников, дежурного по объекту и, желательно, автоматическое оповещение противопожарных служб;
- иметь в наличии несколько ручных углекислотных огнетушителей (например, огнетушитель марки ОУ-3);
- помещение должно быть планом эвакуации при пожаре и аптечкой первой помощи.

В помещении должен быть ответственный за пожарную безопасность.

7.2.10. Требования к организации медицинского обслуживания пользователей ПЭВМ.

Лица, работающие с ПЭВМ более 50% рабочего времени (профессионально связанные с эксплуатацией ПЭВМ), должны проходить обязательные предварительные при поступлении на работу и периодические медицинские осмотры в установленном порядке.

Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену) при условии соблюдения гигиенических требований, установленных *СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы*. Трудоустройство беременных женщин следует осуществлять в соответствии с законодательством Российской Федерации.

7.2.11. Требования к проведению государственного санитарно-эпидемиологического надзора и производственного контроля.

Государственный санитарно-эпидемиологический надзор за производством и эксплуатацией ПЭВМ осуществляется в соответствии с *СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы*.

Не допускается реализация и эксплуатация на территории Российской Федерации типов ПЭВМ, не имеющих санитарно-эпидемиологического заключения.

Инструментальный контроль за соблюдением требований *СанПиН 2.2.2/2.4.1340-03* осуществляется в соответствии с действующей нормативной документацией.

Производственный контроль за соблюдением санитарных правил осуществляется производителем и поставщиком ПЭВМ, а также предприятиями и организациями, эксплуатирующими ПЭВМ в установленном порядке, в соответствии с действующими санитарными правилами и другими нормативными документами.

7.2.12. Расчет виброизоляции для системы кондиционирования.

В ходе обследования помещения, где происходит эксплуатация рассматриваемой системы, было обнаружено оборудование, производящее вибрации – кондиционер с частотой вращения вентилятора 720 оборотов в минуту. Проведем расчет жесткости виброизоляторов и их количества.

Оптимальное соотношение частоты вращения вентилятора в кондиционере и его собственной частоты определяется формулой (согласно ГОСТ 12.4.093-80): $\frac{f}{f_0} = 3$

Частота вращения вентилятора, рассчитывается по формуле:

$$f = \frac{n}{60} = \frac{720}{60} = 12 \text{ Гц}$$

Таким образом, собственная частота: $f_0 = \frac{f}{3} = \frac{12}{3} = 4 \text{ Гц}$

С другой стороны, собственная частота рассчитывается по формуле:

$$f_0 = \frac{1}{2 \cdot \pi} \cdot \sqrt{\frac{q_1 \cdot N}{m}}, \text{ где}$$

q_1 - вертикальная жесткость виброизолятора, Н/см

N - число виброизоляторов, штук

m - масса виброизолятора, кг

Выразим и рассчитаем соотношение $\frac{q_1 \cdot N}{m}$:

$$\frac{q_1 \cdot N}{m} = (2 \cdot \pi \cdot f_0)^2 = (2 \cdot 3.1415 \cdot 4)^2 = 631.66$$

Выбираем виброизолятор ДО-38:

Таблица 7.2. Параметры виброизолятора ДО-38

Обозначение	Нагрузка Р, Н		Вертикальная жесткость, Н/см	Высота в свободном состоянии	Осадка пружины под нагрузкой, мм		Число рабочих витков	Масса, кг
	Рабочая (Р _{раб.})	Предельная (Р _{пр.})			Р _{раб.}	Р _{пр.}		
ДО38	122	152	45	72	27	33	5,6	0,3

Для 4 опор-виброизоляторов, согласно таблице 7.2, выражение $\frac{q_1 \cdot N}{m} = \frac{45 \cdot 4}{0,3} = 600$,

$$\text{собственная частота } f_0 = \frac{1}{2 \cdot \pi} \cdot \sqrt{\frac{q_1 \cdot N}{m}} = \frac{1}{2 \cdot 3,14} \cdot \sqrt{600} = 3,89 \text{ Гц}$$

Таким образом, соотношение частоты вращения вентилятора в кондиционере и его собственной частоты будет равно:

$$\frac{f}{f_0} = \frac{12}{3,89} = 3,08$$

Это соотношение оптимально.

Таким образом, для устранения в помещении вибраций, создаваемых кондиционером, необходимо установить его на 4 опоры-виброизоляторы марки ДО-38.

7.3. Утилизация ПЭВМ.

Извлечение драгоценных металлов из вторичного сырья является частью проблемы использования возвратных ресурсов, которая включает в себя следующие аспекты: нормативно-правовой, организационный, сертификационный, технологический, экологический, экономико-финансовый. Проблема использования вторичного сырья,

содержащего драгоценные материалы из компьютеров, периферийного оборудования и иных средств вычислительной техники (СВТ) актуальна в связи с техническим перевооружением отраслей промышленности.

К драгоценным металлам относятся: золото, серебро, платина, палладий, родий, иридий, рутений, осмий, а также любые химические соединения и сплавы каждого из этих металлов. Статья 2 п. 4 "Федерального закона о драгоценных металлах и драгоценных камнях" от 26 марта 1998 года №1463 гласит: "Лом и отходы драгоценных металлов подлежат сбору во всех организациях, в которых образуются указанные лом и отходы. Собранные лом и отходы подлежат обязательному учёту и могут перерабатываться собирающими их организациями для вторичного использования или реализовываться организациям, имеющим лицензии на данный вид деятельности, для дальнейшего производства и аффинажа драгоценных металлов".

Порядок учёта, хранения, транспортировки, инвентаризации, сбор и сдача отходов драгоценных металлов из СВТ, деталей и узлов, содержащих в своём составе драгоценные металлы для предприятия, учреждения и организации (далее - предприятие), независимо от форм собственности, установлен инструкцией Министерства финансов Российской Федерации от 4 августа 1992 года №67. Все виды работ с драгоценными металлами строго регламентированы нормативно-правовыми документами, перечень которых представлен в Приложении 1.

7.3.1. Разборка изделий.

Последовательность разборки определяется типом изделия СВТ, его конструкционными особенностями и комплектацией.

Как правило, процесс разборки должен выполняться в последовательности, обратной процессу сборки изделия.

7.3.1.1. Разборка персональных компьютеров (ПЭВМ), рабочих станций и серверов.

Технологии разборки ПЭВМ, рабочих станций, серверов и информационно-вычислительных систем едины, поскольку состав их модулей стандартный. Он содержит системный блок и комплект периферийных устройств.

Разборку ПЭВМ и составных модулей целесообразно осуществлять по технологической схеме, представленной на рисунке (см. Рисунок 7.1).

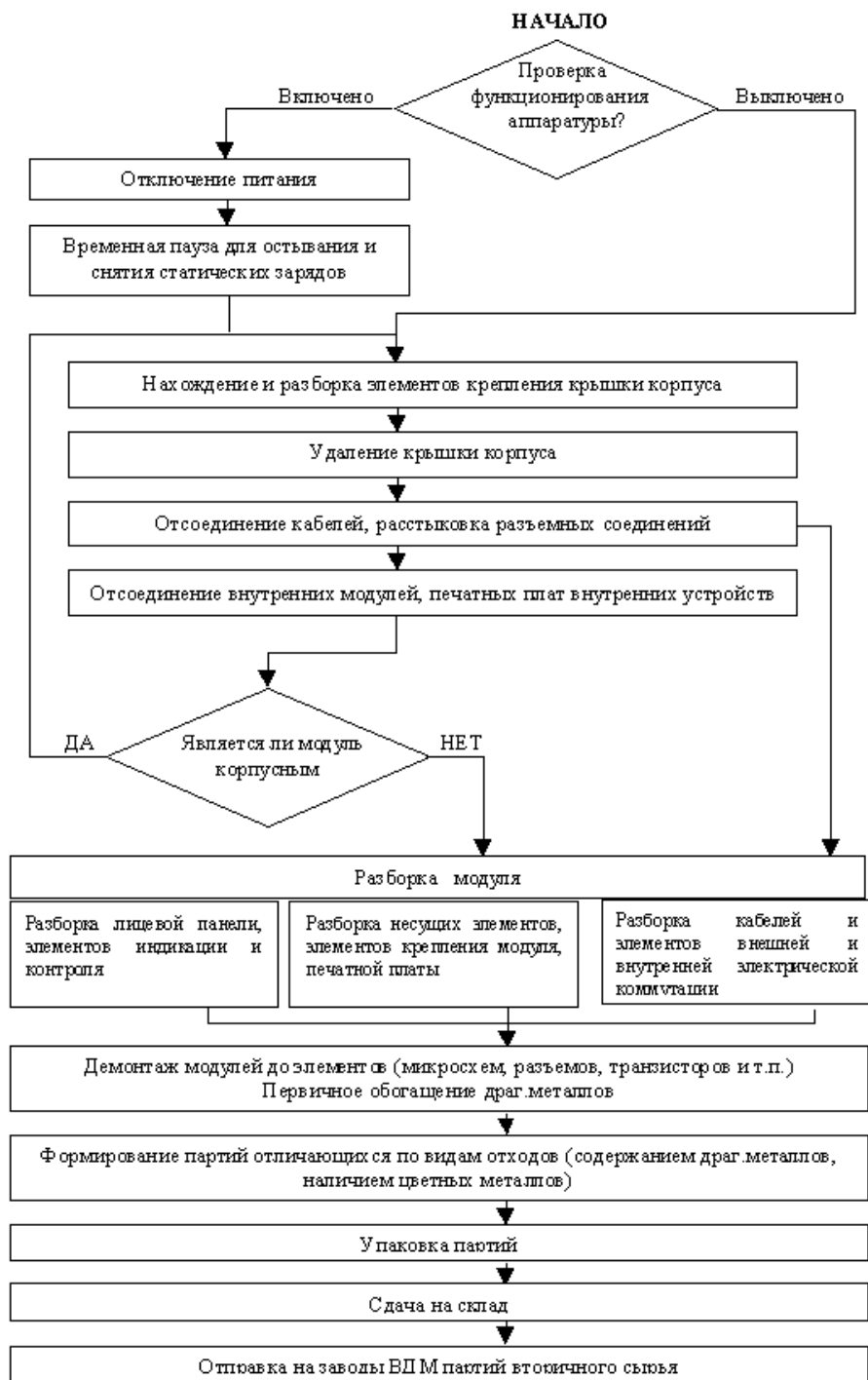


Рисунок 7.1 Технологическая схема разборки ПЭВМ

Порядок разборки системного блока:

Выключить компьютер и отсоединить шнур питания от розетки и системного блока. Отсоединить переходной шнур питания от системного блока к монитору.

Отсоединить от компьютера клавиатуру, монитор, манипулятор "мышь", принтер, сканер и иные внешние устройства.

Найти элементы крепления крышки корпуса (винты, шурупы, пружинные защелки и т.д.). Освободить крышку от элемента крепления.

Снять крышку.

Отсоединить внутренние кабели и плоские шлейфы.

Найти элементы крепления дисководов (НМД, НГМД) в отсеке для дисководов (винты, шурупы, саморезные винты, пружинные защелки и др.). Освободить дисководы и извлечь их из дискового отсека.

Освободить от крепёжных элементов периферийные платы. Извлечь из разъёмов непосредственного контактирования все периферийные платы.

Найти элементы крепления системной платы к корпусу (винты, шурупы). Освободить элементы крепления и извлечь системную плату из корпуса.

Извлечь модули памяти из разъёмов системной платы.

Найти элементы крепления блока питания к корпусу (винты, шурупы, саморезные винты, пружинные защелки и пр.). Освободить элементы крепления и извлечь блок питания.

Разобрать блок питания и извлечь высоковольтные конденсаторы содержащие тантал.

Разобрать ПП и модули памяти до компонентов (микросхем, транзисторов, разъёмов и т.п.).

Произвести сортировку компонентов и сформировать партии электронного лома.

Упаковать партии, составить опись, произвести расчёт (анализ) драгметаллов и передать их на склад.

Провести сортировку цветных и чёрных металлов, пластмасс, сформировать партии и передать их на склад или на переработку.

При оценке содержания драгоценных металлов в партии электронного лома отечественных ПЭВМ необходимо руководствоваться паспортными данными. При оценке ПЭВМ импортного производства необходимо провести ориентировочные расчёты по отечественным аналогам.

7.3.1.2. Обеспечение комплексности технологии разборки.

При разборке изделий СВТ образуются материалы и изделия, которые имеют материальную ценность и подлежат реализации.

Примерный перечень материалов представлен ниже см. 7.10.

Таблица 7.10. Перечень материалов, подлежащих утилизации.

Вид материалов или изделий	Характеристика
Печатные платы, разъемы и соединители, микросхемы	вторичные драгоценные металлы
Электрические провода и кабели, соединители	вторичная медь и её сплавы
Свинец и олово из печатных плат	вторичные припойные пасты (олово и свинец)
Танталовые конденсаторы К-53-1	вторичный тантал
Некоторые корпуса компьютеров, дисковод и т.д.	алюминиевые сплавы
Корпуса стоек, ячеек, шкафов, компьютеров	сталь
Крепежные изделия	болты, гайки, винты
Вентиляторы и электромоторы	по паспорту СВТ
Пластиковая "фракция"	стеклотекстолит, пластмасса разъёмов и соединителей
Экраны компьютеров	стеклофаза, содержащая Pb, Cd, CdS, редкоземельные металлы

Таким образом, можно сделать вывод о целесообразности извлечения вторичных чёрных и цветных металлов, пластмасс, стекла, крепежных изделий, вентиляторов и электромоторов.

7.3.1.3. Извлечение вторичных чёрных металлов.

Отечественная практика показывает, что на 1 г извлекаемого золота приходится около 1 кг лома чёрных металлов. В связи с высокой стоимостью транспортно-погрузочных работ, рекомендуется производить отгрузку предприятиям-покупателям партий лома чёрных металлов весом не менее 10 тонн. Блоки, панели, съёмные кожухи, рамы, каркасы шкафов и стоек стационарных ЭВМ, изготовленные из стального нормализованного профиля или листа, подвергаются сортировке, набираются в партии и реализуются.

Предпочтительно заключение договоров при условии, когда предприятие-покупатель своим транспортом вывозит вторичные металлы с территории предприятия-продавца.

Крепёжные изделия, заготовки стального профиля, листов, вентиляторы, электропускатели, кнопки, электрический кабель направляются на реализацию непосредственно в торговую сеть.

Опыт показывает, что денежные средства от реализации этих изделий не превышают 0,6 % от общей суммы.

7.3.1.4. Извлечение вторичных металлов цветных.

В процессе разборки изделий СВТ образуется лом (содержащий медь) классификация которого должна проводиться по ГОСТ 1639.

В соответствии с ГОСТ 1639 медные шины целесообразно относить к классу А, группам I и II; латунь - к группам IV-VIII; бронзу - к группам XI-XII; отходы кабеля и проводов ПП следует относить к классу Г, группа XIII.

Все виды ломов необходимо сортировать по классам и группам, формировать в партии и реализовывать.

В процессе разборки изделий СВТ алюминий и его сплавы обычно содержатся в типовых конструкциях изделий. По ГОСТ 1639 их следует относить к классам АЗ и Б5.

Все виды отходов необходимо сортировать, формировать в партии и реализовывать.

Свинцово-оловянные припои содержатся в печатных платах и их количество превышает количество золота в десятки раз.

Припои регенерируются при переработке печатных плат.

При разборке СВТ танталовые конденсаторы необходимо складировать отдельно для последующей реализации.

Переработка изделий из пластмасс. Пластмассы следует сортировать по видам.

Переработке подлежат термопласты: поливинилхлорид, полиэтилен, полистирол и т.п.

Стёкла люминесцентных экранов электронно-лучевых трубок следует использовать в производстве керамики и в качестве сырья при производстве новых люминесцентных трубок.

7.3.2. Реализация партий.

На рисунке ниже (см. Рисунок 7.2) представлены основные направления деятельности на этапе "Реализация партий".

Основные действия на этапе "Реализация партий" представляют собой последовательность действий, создающих основу для успешного выполнения процедур завершающего этапа утилизации СВТ.

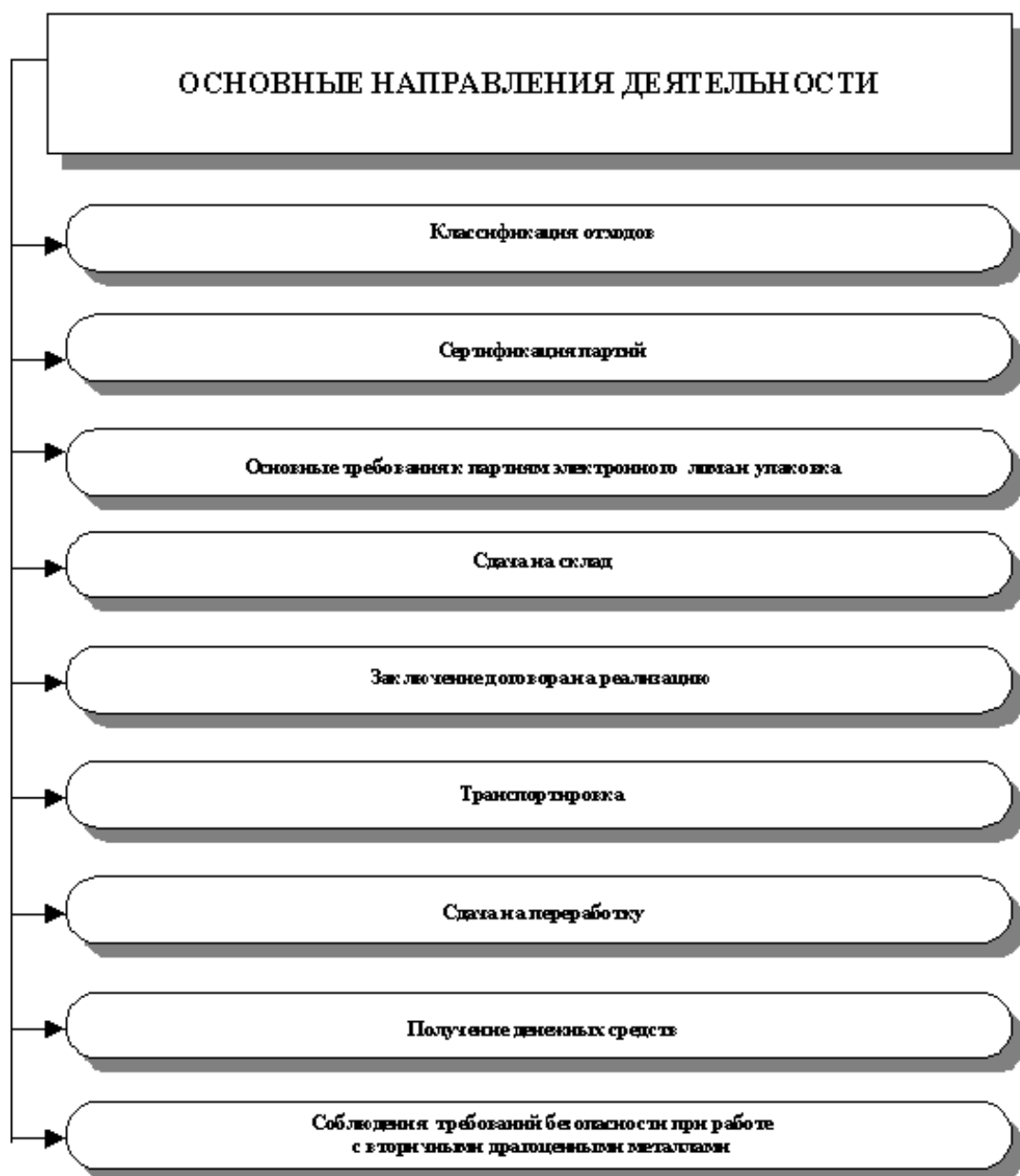


Рисунок 7.2 Основные направления деятельности на этапе “Реализация партии”

7.3.2.1. Классификация отходов.

В настоящее время в России и за рубежом не существует единой классификации вторичного сырья, содержащего драгоценные металлы. Поэтому, возможно разделение вторичного сырья по следующим признакам.

По содержанию драгоценных металлов:

- бедное (менее 1 % золота, 5 % серебра и 1 % металлов платиновой группы);
- богатое (более 1 % золота, 5 % серебра и 1 % металлов платиновой группы).

По составу материала основания:

- на металлической основе;
- на органической (пластиковой) основе;
- на керамической основе;
- на комбинированной основе.

По физическим признакам:

- твёрдые компактные отходы;
- сыпучие (порошки);
- жидкие.

Возможна классификация вторичного сырья в зависимости от типа производства в:

- ювелирной промышленности;
- химической промышленности;
- электронной, электрохимической, оборонной, радиопромышленности (радиолампы, разъёмы, контакты, контактные устройства, платы на органической основе, микросхемы, радиодетали, кабели и провода, лента, высечка, вырубка, аккумуляторы, элементы питания, прочие отходы);
- бытовых отходах (лом бытовой радиоэлектронной аппаратуры, бытовой стеклянный и фарфоровый бой, лом ювелирных украшений и т.д.).

Отходы классифицируются по элементному составу.

При этом электронный лом отличается особым многообразием состава. Например, современный компьютерный лом содержит несколько десятков видов деталей, содержащих благородные, цветные и чёрные металлы.

7.3.2.2. Классификация сырья вторичных драгоценных металлов.

Согласно методике опробования электронного лома и отходов, содержащих драгоценные металлы, разработанной ИАСЦ ГНЦ "ТИРЕДМЕТ", сырьё вторичных драгоценных металлов следует рассортировывать на классы, приведённые в табл.7.11.

Таблица 7.11.

**Классификация электронного лома и отходов, содержащих драгоценные металлы
(по видам)**

Номер класса	Вид сырья
1.	Микросхемы, транзисторы, диоды россыпью
2.	Конденсаторы россыпью
3.	Ножки разъёмов позолоченные и посеребренные
4.	Контакты разделанные
5.	Платы, содержащие элементы классов 1 и 2
6.	Разъёмы с позолоченными и посеребренными ножками
7.	Реле, переключатели, тумблеры
8.	Радиолампы
9.	Платы, содержащие элементы классов 1, 2, 6, 7 и 8
10.	Крупногабаритные детали (волноводы и т.п.)с покрытием из драгметаллами
11.	Элементы питания, аккумуляторы, ампульные батареи
12.	Сыпучие материалы (шихта катализаторов, зола фотоматериалов, шлам иксажный и т.п.)

7.3.2.3. Сертификация партий.

В целях обеспечения строгого учёта, сохранности, сокращения потерь и эффективности использования драгоценных металлов, содержащихся в электронном ломе и отходах, а также для обеспечения единства и требуемой точности измерений при опробовании и проведении анализов химического состава, необходимо руководствоваться нормативными документами утверждёнными Комитетом Российской Федерации по драгоценным металлам и драгоценным камням, Комитетом Российской Федерации по стандартизации, метрологии и сертификации: "Порядком выдачи сертификатов химического состава на партии электронного лома и отходов, содержащих драгоценные металлы", "Временной методикой опробования электронного лома и отходов, содержащих драгоценные металлы".

Указанные документы определяют порядок проведения работ и требования по опробованию и сертификации химического состава партий электронного лома и отходов.

7.3.2.4. Сертификация химического состава электронного лома и отходов, содержащих драгоценные металлы, включает следующие работы:

- оформление и представление заявки в соответствующий орган по сертификации;
- создание комиссии по опробованию;
- опробование, оформление документов по результатам опробования, передача пробы на анализ;
- собственно анализ пробы и оформление количественного химического анализа;
- оформление сертификата.

Выполнение измерений химического состава проб (анализ) электронного лома и отходов, содержащих драгоценные металлы, следует производить методами количественного химического анализа по аттестованным методикам, Приложение 2.

Анализ проб осуществляется аналитическими лабораториями, которые аккредитованы Комитетом Российской Федерации по стандартизации, метрологии и сертификации, а также рекомендованными организациями и органами по сертификации, Приложение 3.

По результатам анализа аккредитованная лаборатория оформляет протокол измерений химического состава пробы электронного лома или отходов по установленной форме.

По результатам процедур опробования и анализа химического состава электронного лома или отходов, орган по сертификации оформляет и выдает заявителю Сертификат химического состава на содержание драгоценных металлов установленной формы.

Сертификат химического состава и комплекс документов по опробованию сертифицируемой партии электронного лома включается в состав сопроводительной документации при передаче партии вторичного сырья от сдатчика заготовителю или переработчику, а также при вывозе за границу для переработки.

При возникновении разногласий, в процессе передачи сертифицированных партий электронного лома и отходов от сдатчика заготовителю или переработчику, производится арбитраж. В этом случае партия не может быть передана переработчику до получения заключения арбитражной лаборатории, аккредитованной Комитетом Российской Федерации по стандартизации, метрологии и сертификации.

7.3.2.5. Основные требования к партиям электронного лома и упаковка.

Утвержденные технические требования к партиям электронного лома, в частности его упаковке, до настоящего времени отсутствуют.

В связи с этим рекомендуется придерживаться следующих правил.

Не допускается смешивание различных классов лома.

В ломе и отходах драгоценных металлов не допускаются посторонние предметы, не относящиеся к естественной засорённости.

Лом и отходы драгоценных металлов должны храниться в специально предназначенной для этого таре: в пакетах из полиэтиленовой плёнки по ГОСТ 10354, в фанерных ящиках по ГОСТ 9396, в металлических ящиках с замками собственного изготовления или мешках из мешочной бумаги по ГОСТ 2226.

Ящики внутри должны быть выложены упаковочной бумагой по ГОСТ 515 или каким-либо плёночным материалом. Пакеты и мешки, предназначенные для хранения и отправки лома и отходов, должны изготавливаться с вывернутыми внутрь двумя боковыми швами без нижнего шва.

Мешки или пакеты, уложенные в деревянные или металлические ящики, допускается заклеивать какой-либо клеевой лентой по ГОСТ 18351.

Опечатывание отходов в таре производится после прошивания мешков и пакетов шпагатом по ГОСТ 17308 или заваривания открытых краев полиэтиленовых пакетов с последующим складыванием краёв "гармошкой" и прошиванием её двумя концами шпагата.

Партия лома и отходов должна состоять из одного или нескольких мест, в каждом из которых вложена одна или несколько позиций различающихся по составу компонентов, конфигураций, габаритным размерам и другими признаками, не меняющих принципиально сущности последующего опробования на перерабатывающих заводах.

Взвешивание и упаковка отходов производится с участием материально ответственных лиц подразделений предприятия сдатчика.

После контрольного взвешивания каждое место должно быть опломбировано или опечатано сургучной печатью сдатчика.

В сопроводительных документах указывается описание пломбы или печати.

7.3.2.6. Сдача на склад.

Передача партии электронного лома из производственного подразделения на склад драгоценных металлов предприятия осуществляется на основе приёмно-передаточного акта, акта изъятия узлов и изделий техники и других нормативных документов.

7.3.2.7. Заключение договора на реализацию.

Между продавцом и покупателем заключается типовый договор, предметом которого является полученная партия электронного лома.

В договоре указываются обязанности сторон, порядок подготовки, отправки лома и отходов драгоценных металлов, порядок приёмки, опробования лома и отходов, порядок расчётов и ответственность сторон.

7.3.2.8. *Транспортировка.*

Транспортирование лома и отходов драгоценных металлов с содержанием золота и металлов платиновой группы более 5 % должно производиться через специальную связь в соответствии с инструкцией Министерства связи Российской Федерации о перевозке ценностей.

Лом и отходы с содержанием золота и металлов платиновой группы менее 5 %, а также отходы серебра отправляются на перерабатывающие заводы почтовыми посылками, багажом по железной дороге или другим видом транспорта с оценочной стоимостью отгружаемого груза.

Сдача на переработку.

Порядок сдачи партии на переработку определяется условиями договора. Типичные условия договора для завода ВДМ следующие.

Вскрытие посылок (мест) производится на заводе приёмной комиссией, которая взвешивает и сверяет фактическое наличие сырья и его качественный состав по каждому виду сырья с данными продавца. По результатам приёмки сырья покупатель высылает продавцу приёмный акт в течение 15 дней от даты поступления сырья.

При доставке сырья транспортом продавца, приёмный акт на количество мест выдаётся на руки уполномоченному представителю продавца в день сдачи сырья.

Представителю продавца необходимо иметь копию описи сдаваемого сырья.

В случае нарушения упаковки или целостности печати материально ответственный работник покупателя в акте на приём отходов отмечает все нарушения.

При расхождении фактически установленных данных при приёмке сырья с данными, значившимися в сопроводительных документах продавца, а также при отсутствии сопроводительных документов, окончательными результатами приёмки является масса брутто, нетто сырья, установленные приёмной комиссией покупателя.

Взвешивание, опробование, пробоотбор и химический анализ проб каждой партии производится в соответствии с нормативно-технической документацией и по технологии, применяемой покупателем.

Однотипные позиции партии подлежат объединению и опробованию по единой технологической схеме.

По результатам опробования сырья на содержание драгоценных металлов, которое производится в течение 60 дней со дня его поступления, покупатель высылает продавцу паспорт с указанием количества драгоценного металла с учетом процента выхода чистого металла в готовую продукцию.

Паспорт подписывается руководителем предприятия-покупателя, главным бухгалтером или их заместителями и скрепляется печатью покупателя.

Порядок получения денежных средств зависит от условия договора. Типичные условия завода ВДМ следующие.

Стоимость поставленного продавцом сырья определяется паспортом покупателя, составленным на основании прејскуранта завода, по мировым ценам на продукцию, получаемую из сырья на день, предшествующий выписке паспорта и пересчитанным в рубли по курсу, установленному Центральным Банком Российской Федерации на день выписки паспорта.

Денежные средства перечисляются на расчётный счет продавца в течение трёх банковских дней со дня получения подтверждающего.

Продавец производит оплату с каждой поставленной партии за опробование.

Все платежи по договору должны производиться в валюте Российской Федерации в безналичной форме.

7.3.2.9. Соблюдение требований безопасности при работе с вторичными драгоценными металлами.

Выполнение работ по разборке списанных СВТ предполагает соблюдение общих правил, изложенных в инструкциях по охране труда для слесаря механо - сборочных работ и лиц, работающих с ручным электроинструментом.

Специальные требования техники безопасности при работе с вторичными драгоценными металлами следующие.

Не допускается сбор, заготовка и переработка радиоактивного лома и отходов драгоценных металлов.

Степень воздействия на организм человека вредных веществ, выделяющихся в процессе заготовки и переработки лома и отходов драгоценных металлов, класс опасности и их предельно-допустимая концентрация (ПДК) в воздухе рабочей зоны установлены по ГОСТ 12.1.005 и ГОСТ 12.1.007, табл.7.12.

Таблица 7.12.

Степень воздействия драгоценных металлов на организм человека

Наименование металла	Характер действия на организм человека	Пути проникновения	Класс опасности	ПДК вредных веществ в воздухе рабочей зоны, мг/м³
Серебро и его соединения	Отходы могут оказывать раздражающее действие на слизистую оболочку носа и дыхательных путей.	Органы дыхания	11	0,5-1
Золото, платина и его соединения	При длительном контакте могут вызывать аллергические дерматиты и экземы.	Открытые участки кожи	-	-
Рутений		Органы дыхания	11	-
Родий	Оказывает раздражающее действие на слизистую оболочку носа и дыхательных путей.	-	-	-

Контроль за содержанием вредных веществ в воздухе рабочей зоны проводится в соответствии с требованиями ГОСТ 12.1.005 и ГОСТ 12.1.007. Анализ проб воздуха проводится в соответствии с нормативно-технической документацией, утверждённой Минздравом Российской Федерации.

Производственные помещения в местах образования вредных веществ и пыли должны быть оборудованы вентиляцией согласно ГОСТ 12.4.021 с обеспечением санитарно-гигиенических требований к воздуху рабочей зоны.

Для снятия статического электричества пылеприёмники, воздуховоды вентиляционных установок должны иметь заземление, выполненное и обозначенное в соответствии с ГОСТ 12.2.007.0, ГОСТ 12.2.007.14 и ГОСТ 21130.

Для предотвращения попадания пыли, твёрдых веществ на слизистую оболочку глаз необходимо пользоваться защитными очками типа ПО-2, ПО-3 согласно ГОСТ 12.4.013.

При работе с пылящими отходами необходимо пользоваться фильтрующими респираторами РУ-60 и РУ-60му по ГОСТ 17269 и респираторами "Лепесток" по ГОСТ 12.4.028.

При этом респираторы должны периодически подвергаться промывке.

Средства индивидуальной защиты работающих с ломом и отходами драгоценных металлов и сплавов должны соответствовать типовым отраслевым нормам бесплатной выдачи рабочим и служащим металлургических производств. Спецодежда должна соответствовать ГОСТ 29057 и ГОСТ 29058.

Помещения в местах выгрузки и загрузки лома и отходов, оказывающих вредное воздействие на организм человека, должны быть оборудованы местными отсосами согласно ГОСТ 12.4.021.

Производственные помещения должны соответствовать требованиям "Санитарных норм проектирования промышленных предприятий СН 245-71".

Метеорологические условия производственных помещений должны соответствовать санитарным нормам проектирования промышленных предприятий по ГОСТ 12.1.005.

Требования безопасности при погрузочно-разгрузочных работах лома и отходов драгоценных металлов и сплавов должны соответствовать ГОСТ 12.3.009.

Требования по обеспечению взрывобезопасности.

Предприятия и организации, заготавливающие и перерабатывающие лом и отходы драгоценных металлов сплавов, должны проверять весь лом и отходы драгоценных металлов на взрывобезопасность.

Из лома необходимо отобрать и удалить взрывоопасные предметы, материалы, в том числе электронно-вакуумные трубки дисплеев.

Замкнутые сосуды, резервуары и другие полые предметы (баллоны, цилиндры, сосуды, электровакуумные изделия и т.д.) разгерметизируются и освобождаются от содержимого (газов или жидкостей).

Разгерметизацию должны производить рабочие, прошедшие специальное обучение, которые перед началом работы инструктируются в установленном порядке о мерах предосторожности.

8. Заключение.

В процессе выполнения дипломного проекта получены следующие результаты:

1. Проанализирована ситуация на рынке роботов удаленного присутствия. Выявлены недостатки и достоинства решений конкурентов.
2. Разработана архитектура системы удаленного присутствия.
3. Разработано все необходимое программное обеспечение: робота, клиента и сервера, включая основные и вспомогательные средства.
4. Введен в эксплуатацию робот удаленного присутствия. Проведена подготовка к введению в строй мелкой серии роботов.

Список использованных источников

1. Бьерн Страуструп. Язык моделирования C++. Специальное издание. Пер. с англ. – М.: ООО «Бином-пресс», 2007 г. – 1104 с.
2. Солтер Н.А., Клепер С.Дж. C++ для профессионалов. М.: Диалектика, 2006 г. , пер. с 2005, 907 с.
3. Макконнелл С. Совершенный код. Мастер класс – М.: Издательско-торговый дом «Русская Редакция» ; СПб.: Питер, 2005. – 896 стр.: ил.
4. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. - Символ-Плюс, 2006, 304 с.
5. Фаулер М. Основы UML, 3-е издание. : СПб: Символ-Плюс, 2002 г. 185 с.
6. Ларман, Крэг. Применение UML и шаблонов проектирования. 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 624 с.
7. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. ГОСТ 19.201-78.
8. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. ГОСТ 19.701-90. Условные обозначения и правила выполнения.

Приложение А. Исходный код программы vreport-server.

```
#!/usr/bin/python
#####
# Copyright (c) 2013 Evgeny Proydakov <lord.tiran@gmail.com>
#####
# -*- coding: utf-8 -*-
#####

import os, sys
import logging
import argparse
import traceback
import time, datetime
import asyncore, socket
import xml.dom.minidom

#####

APP_VERSION = u'0.0.1'
SERVER_PORT = 19750

EXIT_SUCCESS = 0
EXIT_FAILURE = 1

#####

def safe_dir_complete(dir):
    if not dir[len(dir) - 1] == '/':
        dir += '/'
    return dir

#####

class Server(asyncore.dispatcher):

    def __init__(self, address, report_directory):
        self.logger = logging.getLogger('Report Server')
        asyncore.dispatcher.__init__(self)
        self.report_directory = safe_dir_complete(report_directory)

        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.bind(address)
        self.address = self.socket.getsockname()
        self.logger.debug('binding to %s', self.address)
        self.listen(1)

    def handle_accept(self):
        socket, address = self.accept()
        self.logger.debug('handle_accept() -> %s', address)
        QueryHandler(socket, self.report_directory)

    def handle_close(self):
        self.logger.debug('handle_close()')
        self.close()

class QueryHandler(asyncore.dispatcher):

    def __init__(self, socket, report_directory):
        asyncore.dispatcher.__init__(self, socket)
        self.logger = logging.getLogger('Query Handler')
        self.read_buffer = ''
        self.report_directory = safe_dir_complete(report_directory)
        self.logger.debug('Create')
```

```

def __del__(self):
    application_name = u''
    try:
        xml_tree = xml.dom.minidom.parseString( self.read_buffer )
        report_xml_node = xml_tree.getElementsByTagName('report')[0]
        application_name =
report_xml_node.getElementsByTagName('name')[0].childNodes[0].nodeValue
    except:
        traceback.print_exc(file=sys.stdout)
        self.logger.debug('Parsing XML except')

    if len(application_name) > 0:
        out_path = self.report_directory + application_name + u '/'

        if not os.path.exists(out_path):
            os.makedirs(out_path)

        out_file_name = out_path + datetime.datetime.now().strftime(u'%Y-%m-%d_%H-
%M-%S-%f') + u'.xml'
        out_file = open( out_file_name, 'w' )
        out_file.write( self.read_buffer )
        out_file.close()

        self.logger.debug('Create report: %s', out_file_name)

    self.logger.debug('Destroy')

def handle_close(self):
    self.close()

def handle_read(self):
    read_buffer = self.recv(1024)
    self.read_buffer += read_buffer

#####

parser = argparse.ArgumentParser(description = u'vreport server:', version =
APP_VERSION)
parser.add_argument(u'--port', action = 'store', dest = u'port', default =
SERVER_PORT, help = u'server port')
parser.add_argument(u'--dir', action = 'store', dest = u'directory', help = u'report
directory')
parser.add_argument(u'--verbose', action = 'store_true', dest = u'verbose', default =
False, help = u'debug output')

args = parser.parse_args()

if args.directory is None:
    parser.print_help()
    sys.exit(EXIT_SUCCESS)

if not os.path.exists(args.directory):
    print u'Log directory not found:', args.directory
    sys.exit(EXIT_FAILURE)

if args.verbose:
    logging.basicConfig( level=logging.DEBUG, format='%(name)s: %(message)s' )

address = ('', args.port)
s = Server(address, args.directory)
asyncore.loop(timeout=15)

```


Приложение Б. Исходный код программы vshare-server.

```
#!/usr/bin/python
#####
# Copyright (c) 2013 Evgeny Proydakov <lord.tiran@gmail.com>
#####
# -*- coding: utf-8 -*-
#####

import os, sys
import signal
import logging
import argparse
import traceback
import subprocess
import time, datetime

from torrent_server import TorrentServer

#####

APP_VERSION = u'0.0.1'

EXIT_SUCCESS = 0
EXIT_FAILURE = 1

DEFAULT_CYCLE_TIME = 60
THREAD_SLEEP_TIME = 5

#####

g_server = None

#####

def safe_dir_complete(dir):
    if not dir[len(dir) - 1] == '/':
        dir += '/'
    return dir

#####

def handler(signum, frame):
    logger = logging.getLogger('Signal Handler')
    logger.debug('catch')
    global g_server
    if g_server is not None:
        g_server.shutdown()

#####

class ShareServer(TorrentServer):
    def __init__(self, pidfile, datastore, terminal, streaming_iter_time):
        TorrentServer.__init__(self, pidfile, datastore, terminal,
streaming_iter_time)
        self.logger = logging.getLogger('Share Server')
        self.logger.debug('init')
        self.script_directory =
safe_dir_complete(os.path.dirname(os.path.abspath(sys.argv[0])))

    def __del__(self):
        self.logger.debug('destroy')
        TorrentServer.__del__(self)

    def run(self):
        self.logger.debug('run')
```

```

signal.signal(signal.SIGTERM, handler)
signal.signal(signal.SIGINT, handler)

while self.working:
    next_time = self.pull_time + self.pull_iter_time
    if next_time < time.time():
        self.pull_time = time.time()
        self.loop_body()
    time.sleep(self.loop_time)

self.logger.debug('stop')

def loop_body(self):
    self.collect_garbage()
    self.pull_datastore()
    self.load_distributions()
    self.streaming_new_files()

def streaming_new_files(self):
    self.logger.debug('streaming new files')
    for config_name, info in self.info.iteritems():
        torrent_name = info['name'] + self.torrent_exp
        try:
            if self.distributions.get(torrent_name) is None:
                self.generate_torrent(info['name'])
                share = self.share_file(info['email'], torrent_name)
                streaming = False
                if share:
                    streaming = self.start_streaming_file(torrent_name, info,
True)
            if not share or not streaming:
                torrent_file_name = self.datastore + torrent_name
                os.remove(torrent_file_name)
        except:
            traceback.print_exc(file=sys.stdout)

    def share_file(self, email, file):
        file_name = self.datastore + file
        command = 'python ' + self.script_directory + 'share_file.py' + ' --email "' +
email + '" --file "' + file_name + '"'
        proc = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
        proc.wait()
        if proc.returncode == 0:
            self.logger.debug('share file: ' + file)
            return True
        return False

#####

aparser = argparse.ArgumentParser(description = u'share server:', version =
APP_VERSION)
aparser.add_argument(u'--start', action = 'store_true', dest = u'start', help =
u'start daemon')
aparser.add_argument(u'--stop', action = 'store_true', dest = u'stop', help =
u'stop daemon')
aparser.add_argument(u'--restart', action = 'store_true', dest = u'restart', help =
u'restart saemon')
aparser.add_argument(u'--dir', action = 'store', dest = u'datastore', help =
u'datastore directory')
aparser.add_argument(u'--pid', action = 'store', dest = u'pidfile', help =
u'process pid file')
aparser.add_argument(u'--time', action = 'store', dest = u'time', default
= DEFAULT_CYCLE_TIME, help = u'cycle time(default: ' + str(DEFAULT_CYCLE_TIME) + 's
)')
aparser.add_argument(u'--verbose', action = 'store_true', dest = u'verbose', default
= False, help = u'debug output')

```

```

args = aparser.parse_args()

if args.datastore is None or args.pidfile is None:
    aparser.print_help()
    sys.exit(EXIT_SUCCESS)

if args.verbose:
    logging.basicConfig( level=logging.DEBUG, format='%(name)s: %(message)s' )

g_server = ShareServer(args.pidfile, args.datastore, args.verbose, int(args.time))

if len(sys.argv) >= 2:
    if args.start:
        g_server.start()
    elif args.stop:
        g_server.stop()
    elif args.restart:
        g_server.restart()
    else:
        print "Unknown command"
        sys.exit(2)
    sys.exit(0)

else:
    print "usage: %s --start|--stop|--restart [options]" % sys.argv[0]
    sys.exit(2)

```

Приложение В. Исходный код программы vdatastore-server.

```
#!/usr/bin/python
#####
# Copyright (c) 2013 Evgeny Proydakov <lord.tiran@gmail.com>
#####
# -*- coding: utf-8 -*-
#####

import os, sys
import signal
import logging
import argparse
import traceback
import subprocess
import time, datetime
import xml.dom.minidom

import poplib
from email import parser

from torrent_server import TorrentServer

#####

APP_VERSION = u'0.0.1'

EXIT_SUCCESS = 0
EXIT_FAILURE = 1

DEFAULT_CYCLE_TIME = 60

PULL_HOST      = "pop.yandex.ru"
PULL_EMAIL     = "torrent@wicron.com"
PULL_PASSWORD  = "*****"

#####

g_server = None

#####

def safe_dir_complete(dir):
    if not dir[len(dir) - 1] == '/':
        dir += '/'
    return dir

#####

def handler(signum, frame):
    logger = logging.getLogger('Signal Handler')
    logger.debug('catch')
    global g_server
    if g_server is not None:
        g_server.shutdown()

#####

class MailAgent:
    def __init__(self):
        self.logger = logging.getLogger('Mail Agent')
        self.emessages = []

    def connect(self):
        self.logger.debug('connecting ...')
        self.mailserver = poplib.POP3_SSL(PULL_HOST)
        self.mailserver.user(PULL_EMAIL)
```

```

        self.mailserver.pass_(PULL_PASSWORD)
        self.logger.debug('connection completed')

    def disconnect(self):
        self.logger.debug('disconnecting ...')
        self.mailserver.quit()
        self.mailserver = None
        self.logger.debug('disconnection completed')

    def read(self):
        if self.mailserver is None:
            return False

        self.logger.debug('read ...')

        try:
            self.emessages = [self.mailserver.retr(i) for i in range(1,
len(self.mailserver.list()[1]) + 1)]
            self.emessages = ["\n".join(mssg[1]) for mssg in self.emessages]
            self.emessages = [parser.Parser().parsestr(mssg) for mssg in self.emessages]

            for i in range(1, len(self.emessages) + 1):
                self.mailserver.dele(i)

        except:
            traceback.print_exc(file=sys.stdout)
            self.emessages = []

        self.logger.debug('get ' + str(len(self.emessages)) + ' emails')
        return True

#####

class DatastoreServer(TorrentServer):
    def __init__(self, pidfile, datastore, terminal, pull_iter_time):
        TorrentServer.__init__(self, pidfile, datastore, terminal, pull_iter_time)
        self.logger = logging.getLogger('Datastore Server')
        self.agent = MailAgent()
        self.emessages = []

    def __del__(self):
        self.logger.debug('destroy')
        TorrentServer.__del__(self)

    def run(self):
        self.logger.debug('run')

        signal.signal(signal.SIGTERM, handler)
        signal.signal(signal.SIGINT, handler)

        self.pull_datastore()
        self.load_distributions()

        while self.working:
            next_time = self.pull_time + self.pull_iter_time
            if next_time < time.time():
                self.pull_time = time.time()
                self.loop_body()
                time.sleep(self.loop_time)

        self.logger.debug('stop')

    def loop_body(self):
        self.collect_garbage()
        self.pull_email()

        for email in self.emessages:

```

```

        torrent_name = self.save_torrent(email)
    if torrent_name:
        file_name = torrent_name.replace(self.torrent_exp, '')
        info = self.gen_meta_info(file_name)
        if info:
            self.start_streaming_file(torrent_name, info)
        else:
            os.remove(self.datastore + torrent_name)

self.emessages = []

def pull_email(self):
    self.logger.debug('pull_email')

    try:
        self.agent.connect()
        self.agent.read()
        self.agent.disconnect()
    except:
        traceback.print_exc(file=sys.stdout)

    self.emessages = self.agent.emessages

def save_torrent(self, email):
    try:
        payload_len = len(email.get_payload())
        if payload_len > 0:
            attachment = email.get_payload()[0]
            torrent_name = attachment.get_filename()
            open(self.datastore + torrent_name,
'wb').write(attachment.get_payload(decode=True))
            return torrent_name

    except:
        return None

    return None

def gen_meta_info(self, file_name):
    info = {}
    try:
        doc = xml.dom.minidom.Document()

        base = doc.createElement('info')
        doc.appendChild(base)

        name_node = doc.createElement('data')
        name_node_content = doc.createTextNode( file_name )
        name_node.appendChild(name_node_content)
        base.appendChild(name_node)
        info['data'] = file_name

        now = datetime.datetime.utcnow()
        datetime_str = now.strftime(u'%Y-%m-%d_%H-%M-%S')

        datetime_node = doc.createElement('date')
        datetime_node_content = doc.createTextNode( datetime_str )
        datetime_node.appendChild(datetime_node_content)
        base.appendChild(datetime_node)
        info['date'] = datetime_str

        torrent_admin_email = 'torrent@wicon.com'
        email_node = doc.createElement('email')
        email_node_content = doc.createTextNode( torrent_admin_email )
        email_node.appendChild(email_node_content)
        base.appendChild(email_node)

```

```

        info['email'] = torrent_admin_email

        xml_report_text = doc.toxml(encoding='utf-8')
        open(self.datastore + file_name + '.xml', 'wb').write(xml_report_text)

    except:
        traceback.print_exc(file=sys.stdout)
        return None

    return info

#####

aparser = argparse.ArgumentParser(description = u'vtorrent server:', version =
APP_VERSION)
aparser.add_argument(u'--start',    action = 'store_true', dest = u'start',    help =
u'start daemon')
aparser.add_argument(u'--stop',    action = 'store_true', dest = u'stop',    help =
u'stop daemon')
aparser.add_argument(u'--restart', action = 'store_true', dest = u'restart', help =
u'restart saemon')
aparser.add_argument(u'--dir',    action = 'store',    dest = u'datastore', help =
u'datastore directory')
aparser.add_argument(u'--pid',    action = 'store',    dest = u'pidfile', help =
u'process pid file')
aparser.add_argument(u'--time',    action = 'store',    dest = u'time',    default =
DEFAULT_CYCLE_TIME, help = u'cycle time(default: ' + str(DEFAULT_CYCLE_TIME) + 's)')
aparser.add_argument(u'--verbose', action = 'store_true', dest = u'verbose', default =
False, help = u'debug output')

args = aparser.parse_args()

if args.datastore is None or args.pidfile is None:
    aparser.print_help()
    sys.exit(EXIT_SUCCESS)

if args.verbose:
    logging.basicConfig(level=logging.DEBUG, format='%(name)s: %(message)s')

g_server = DatastoreServer(args.pidfile, args.datastore, args.verbose, int(args.time))

if len(sys.argv) >= 2:
    if args.start:
        g_server.start()
    elif args.stop:
        g_server.stop()
    elif args.restart:
        g_server.restart()
    else:
        print "Unknown command"
        sys.exit(2)
    sys.exit(0)

else:
    print "usage: %s --start|--stop|--restart [options]" % sys.argv[0]
    sys.exit(2)

```

Приложение Д. Python скрипт для создания установщика приложения робота для debian family GNU / Linux.

```
#!/usr/bin/python
#####
# Copyright (c) 2013 Evgeny Proydakov <lord.tiran@gmail.com>
#####
# -*- coding: utf-8 -*-
#####

import os
import sys
import glob
import shutil
import argparse
import platform
import subprocess

#####
# constants #
#####

APP_VERSION = u'0.0.1'

LIBRARIES = ['libvcore.so', 'libvdyna.so', 'libvunirobot.so', 'libvpower.so',
             'libvmotion.so', 'libvrobot_cs.so', 'libvrobot_hal.so']

EXECUTABLES = ['vrobot']
SYS_APPS = ['vdyna-ui']

SCRIPTS = ['scripts/watchdog-robot.sh', 'scripts/watchdog-xwikeymap.sh',
            'scripts/watchdog-openvpn.sh', 'scripts/watchdog-reboot.sh',
            'scripts/vrobot-daemon.py', 'scripts/auto_routing_switch']

CONFIGS = ['config/robot.xml']
UDEVS = ['98-wicron.rules']
X11CONFS = ['config/99-calibration.conf']

DEBIAN_VROBOT_CONFIG = 'vrobot-control'
DEBIAN_CONFIG_NAME = 'control'
DEBIAN_CONFIG_DIR = 'DEBIAN'

DEST_BIN_PATH = 'usr/local/bin'
DEST_LIB_PATH = 'usr/local/lib'
DEST_UDEV_PATH = 'etc/udev/rules.d'
DEST_X11_CONF = 'etc/X11/xorg.conf.d'
PKG_PATH = 'pkg-vrobot'

FAKE_FILE_NAME = u'FAKE_FILE'

#####
# functions #
#####

def safe_dir_complete(dir):
    if not dir[len(dir) - 1] == '/':
        dir += '/'
    return dir

#####
# main code #
#####

print u'STARTED SCRIPT:', sys.argv[0], u'\n'
```



```

cpu = u'' + platform.processor()

print u'HOST OS: ', sys.platform
print u'HOST CPU:', platform.processor()

print u''

# parse console options

arg_parser = argparse.ArgumentParser(description = u'wicron vrobot deb bilder:',
version = APP_VERSION)
arg_parser.add_argument(u'-b', action = 'store', dest = u'build_directory', default =
u'', help = u'build directory')
arg_parser.add_argument(u'-a', action = 'store', dest = u'architecture', default =
cpu, help = u'architecture')

args = arg_parser.parse_args()

build_directory = u'' + args.build_directory

if not os.path.exists(build_directory):
    print u'Build directory not found. Stop build deb.'
    sys.exit(1)

build_directory = safe_dir_complete(build_directory)

print u'Found build directory:', build_directory, u'\n'

pkg_path = build_directory + PKG_PATH + u'/'

old_pkg_path = pkg_path
if os.path.exists(old_pkg_path):
    print u'Remove directory:', old_pkg_path
    shutil.rmtree(old_pkg_path)

for files in glob.glob(build_directory + "*.deb"):
    print u'Remove file:', files
    os.remove(files)

print u''

pkg_config_path = pkg_path + DEBIAN_CONFIG_DIR + u'/'
pkg_bin_path = pkg_path + DEST_BIN_PATH + u'/'
pkg_lib_path = pkg_path + DEST_LIB_PATH + u'/'
pkg_udev_path = pkg_path + DEST_UDEV_PATH + u'/'
pkg_x11_conf_path = pkg_path + DEST_X11_CONF + u'/'

dirs = []
dirs.append(pkg_path)
dirs.append(pkg_config_path)
dirs.append(pkg_bin_path)
dirs.append(pkg_lib_path)
dirs.append(pkg_udev_path)
dirs.append(pkg_x11_conf_path)

file = {}
file['src'] = build_directory + DEBIAN_VROBOT_CONFIG
file['dst'] = pkg_config_path + DEBIAN_CONFIG_NAME

files = []
files.append(file)

for e in EXECUTABLES:
    f = {}
    f['src'] = build_directory + e
    f['dst'] = pkg_bin_path + f['src'].split('/')[1]
    files.append(f)

```

```

SYSTEM_PATH = os.environ['PATH']
for s in SYS_APPS:
    f = {}
    f['src'] = FAKE_FILE_NAME
    f['dst'] = pkg_bin_path + s

    system_path = SYSTEM_PATH.split(':')
    for path in system_path:
        if len(path) == 0:
            system_path.remove(path)

    for path in system_path:
        test_path = safe_dir_complete(path) + s
        if os.path.exists(test_path):
            f['src'] = test_path
            break
    if f['src'] == FAKE_FILE_NAME:
        print u'Not found: ' + l + u'.', u'Stop build deb.'
        sys.exit(1)

    files.append(f)

for s in SCRIPTS:
    f = {}
    f['src'] = s
    f['dst'] = pkg_bin_path + f['src'].split('/')[1]
    files.append(f)

for c in CONFIGS:
    f = {}
    f['src'] = c
    f['dst'] = pkg_bin_path + f['src'].split('/')[1]
    files.append(f)

for u in UDEVS:
    f = {}
    f['src'] = build_directory + u
    f['dst'] = pkg_udev_path + u
    files.append(f)

for x in X11CONFS:
    f = {}
    f['src'] = x
    f['dst'] = pkg_x11_conf_path + x.split('/')[1]
    files.append(f)

LD_LIBRARY_PATH = os.environ['LD_LIBRARY_PATH']
for l in LIBRARIES:
    f = {}
    f['src'] = FAKE_FILE_NAME
    f['dst'] = pkg_lib_path + l

    ld_library_path = LD_LIBRARY_PATH.split(':')
    for path in ld_library_path:
        if len(path) == 0:
            ld_library_path.remove(path)

    for path in ld_library_path:
        test_path = safe_dir_complete(path) + l
        if os.path.exists(test_path):
            f['src'] = test_path
            break
    if f['src'] == FAKE_FILE_NAME:
        print u'Not found: ' + l + u'.', u'Stop build deb.'
        sys.exit(1)

```

```

        files.append(f)

for d in dirs:
    print u'Make dir:', d
    os.makedirs(d)

print u''

for f in files:
    if os.path.islink(f['src']):
        print u'Link:', f['src'], u'to:', f['dst']
        linkto = os.readlink(f['src'])
        os.symlink(linkto, f['dst'])
    else:
        print u'Copy:', f['src'], u'to:', f['dst']
        shutil.copy(f['src'], f['dst'])

print u''

command = u'fakeroot dpkg-deb --build ' + pkg_path
exit_code = subprocess.call(command, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.STDOUT)

if not exit_code == 0:
    print u'Create deb error.', u'Command:', command, u'Exit code:', exit_code
    sys.exit(1)

version_file = build_directory + u'git.version.long'
fd = open(version_file, "r")
version_data = fd.readlines()
fd.close()

version = version_data[0]
version = version.replace('\n', '')

in_file = build_directory + PKG_PATH + u'.deb'
out_file = build_directory + 'wicron-vrobot' + '_' + version + '_' + args.architecture
+ '.deb'
shutil.move(in_file, out_file)

print u'Generated:', out_file

```

Приложение Д. Inno Setup скрипт для создания установщика клиентского приложения для ОС семейства Windows.

```
; webot inno setup install script

#define AppName      "${PROJECT_NAME}"
#define AppVersion   "${PROJECT_VERSION_LONG}"
#define AppPublisher "${PROJECT_VENDOR_LONG}"
#define AppURL       "${ORG_WEBSITE}"
#define AppGUID      "34B25717-C3E5-455D-8FCF-01E60BAEB42A"

#define AppVClientName "vclient"
#define AppVClientExeName "vclient.exe"

; SEE THE DOCUMENTATION FOR DETAILS ON CREATING .ISS SCRIPT FILES!

[Setup]
AppName = {#AppName}
AppVersion = {#AppVersion}
AppId = {#AppGUID}
AppPublisher = {#AppPublisher}
AppCopyright = {#AppPublisher}
AppPublisherURL = {#AppURL}
VersionInfoCompany = {#AppPublisher}
DefaultDirName = {pf}\{#AppName}
DefaultGroupName = {#AppName}
UninstallDisplayIcon = {app}\{#AppVClientExeName}
Compression = lzma2
SolidCompression = yes
OutputBaseFilename = ${PROJECT_NAME}-${PROJECT_VERSION_LONG}

[Languages]
Name: ru; MessagesFile: "compiler:Languages\Russian.isl"

[Tasks]
Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}";
Flags: unchecked
```

[Files]

Source: "vclient.exe"; DestDir: "{app}"

Source: "Qt5Core.dll"; DestDir: "{app}"

Source: "Qt5Gui.dll"; DestDir: "{app}"

Source: "Qt5Widgets.dll"; DestDir: "{app}"

Source: "Qt5OpenGL.dll"; DestDir: "{app}"

Source: "qminimal.dll"; DestDir: "{app}/platforms"

Source: "qwindows.dll"; DestDir: "{app}/platforms"

Source: "libGLv2.dll"; DestDir: "{app}"

Source: "libEGL.dll"; DestDir: "{app}"

Source: "icuuc49.dll"; DestDir: "{app}"

Source: "icuin49.dll"; DestDir: "{app}"

Source: "icudt49.dll"; DestDir: "{app}"

Source: "D3DCompiler_43.dll"; DestDir: "{app}"

Source: "vcore.dll"; DestDir: "{app}"

Source: "msvcr100.dll"; DestDir: "{app}"

Source: "msvcrt100.dll"; DestDir: "{app}"

; configs

Source: "client.xml"; DestDir: "{app}"

[Run]

Filename: "{sys}\netsh.exe"; Parameters: "firewall add allowedprogram ""{app}\vclient.exe"" ""Wicron vclient"" ENABLE ALL"; StatusMsg: "Status msg..."; Flags: runhidden; MinVersion: 0,5.01.2600sp2;

[Icons]

Name: "{group}\{#AppVClientName}"; Filename: "{app}\{#AppVClientExeName}"; WorkingDir: "{app}"

Name: "{group}\Uninstall Webot"; Filename: "{uninstallexe}"

Name: "{commondesktop}\{#AppVClientName}"; Filename: "{app}\{#AppVClientExeName}"; Tasks: desktopicon

[UninstallRun]

Filename: "{sys}\netsh.exe"; Parameters: "firewall delete allowedprogram program=""{app}\vclient.exe"""; Flags: runhidden; MinVersion: 0,5.01.2600sp2;