**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Пермский государственный национальный исследовательский университет»**

Механико-математический факультет

Кафедра информационных технологий

**ОТЧЕТ**

по выполнению практической работы   
на тему: «Создание инфраструктуры предприятия»

**Выполнили:**

Студенты 4-го курса механико-математического факультета группы ФИТ-1,2-2017  
Ешмаев В.О.

Миллер А.Д.

Захаров П.С.

Пермь 2021

**Оглавление**

[1 Постановка задачи 3](#_Toc70409590)

[2 Проектирование инфраструктуры 3](#_Toc70409591)

[2.1 Ведение контактов между участниками 3](#_Toc70409592)

[2.2 Контроль версий 11](#_Toc70409593)

[2.3 Общее файловое пространство на устройствах 16](#_Toc70409594)

[2.4 Резервное копирование файлов 22](#_Toc70409595)

[3 Репозиторий и ведение журнала в формате markdown 22](#_Toc70409596)

[4 Исследование на уязвимости 22](#_Toc70409597)

[5 Комплекс мер по защищенности 22](#_Toc70409598)

[Заключение 22](#_Toc70409599)

# 1 Постановка задачи

Виртуальная сеть малого предприятия, "стартапа", состоящего из студентов. Денег - нет, офиса - нет, есть оборудование (ноутбуки, телефоны...) и Сеть. Предприятие занимается дизайном стикеров для соцсетей и по-мелочи торгует разным через соц площадки. Необходимо обеспечить единое файловое пространство на компьютерах участников, резервное копирование данных, обмен контактами, ведение проектов (контроль версий).

Задачи:

1. Спроектировать инфраструктуру в соответствии с темой;
   1. общее файловое пространство на устройствах,
   2. резервное копирование данных,
   3. обмен контактов между участниками,
   4. контроль версий.
2. Создать репозиторий, журнал и вести его по ходу выполнения работы (формат markdown);
3. Провести исследование системы на наличие уязвимостей;
4. Предложить комплекс мер по доведению защищенности инфраструктуры до одного из уровней по стандарту.

# 2 Проектирование инфраструктуры

## 2.1 Ведение контактов между участниками

Синхронизация контактов может пригодиться в том случае, когда имеется несколько гаджетов, независимо на Android или iOS. Именно с её помощью можно без проблем перекинуть все имеющеюся контакты с одного устройства на другое. Утилиты, которые справляются с этой задачей, широко распространены. Их много, а поэтому найти удобную и любимую не составит труда. Мы рассмотрим парочку вариантов, они могли бы помочь осуществить задуманное.

**Outlook**

Это не специализированная программа для синхронизации. Это утилита, которая играет роль менеджера и легко работает как почтовый клиент. Но помимо написания писем, программа может работать с календарем, планировать задачи, использовать записную книжку, менеджер контактов. Также есть и работа с документами.

Когда вы добавляете в программу учетную запись IMAP, можно настроить синхронизацию папок с подпиской. Папки с подпиской отображаются в списке, поскольку регулярно синхронизируются с почтовым сервером. Если есть такие, которые вы нечасто используете, от них можно отписаться. Это нужно для того, чтобы процесс обновления был быстрее.

Вообще, эта программа универсальна. Она позволяет синхронизировать разного рода данные. Можно с одного почтового ящика обновлять письма на другом. Можно использовать контакты с Outlook на смартфоне. Можно переправлять данные с календаря, планировщика и многое другое (рис. 1).

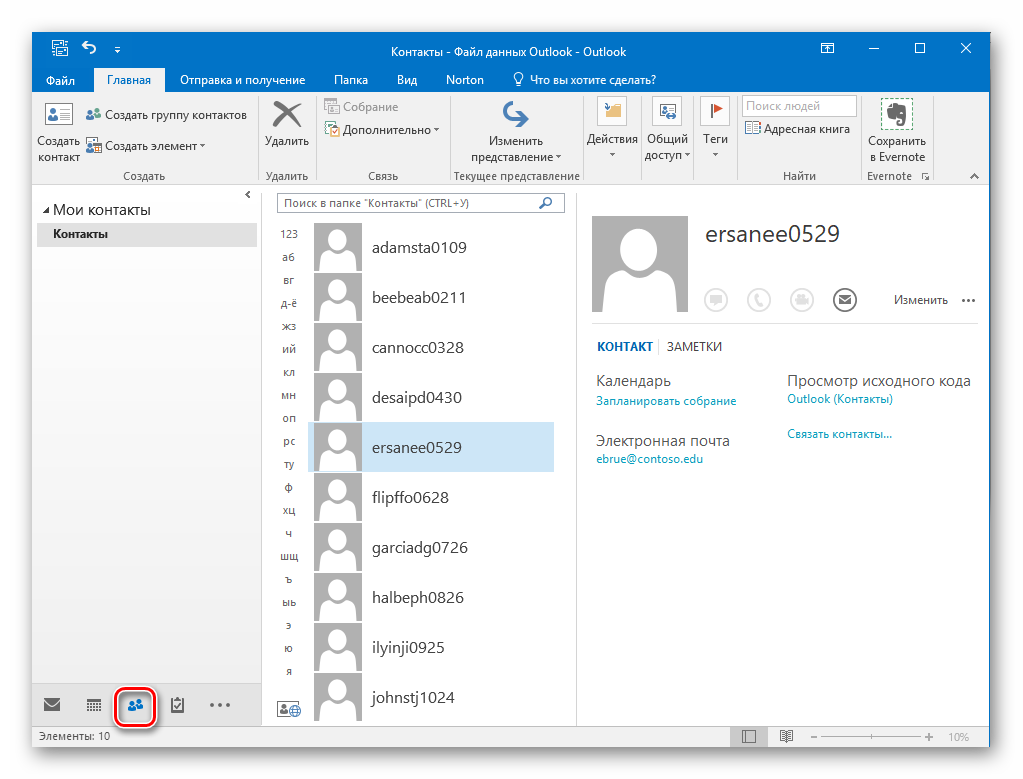


Рисунок 1 - Синхронизация контактов через Outlook

**Google**

Профиль Google за последние десятилетия стал гораздо большим, чем просто адрес электронной почты. Он используется для доступа к более сотни сервисов, созданных Гугл, и для тысяч иных программ и приложений сторонних разработчиков. С помощью функций дублирования информации в собственное облако, можно сохранить и переносить информацию с одного устройства на другое. Синхронизация контактов с Google проходит подобным способом.

Синхронизация контактов работает по схеме:

* анализ нужных данных на устройстве,
* копирование информации в выделяемое для этого аккаунта облако,
* предоставление доступа к сведениям в веб-версии или для других устройств, после авторизации через этот аккаунт,
* копирование сведений из облака в память нового устройства, после включения синхронизации.

Синхронизация работает в двух направлениях: копирует данные с устройства в облако, и копирует сведения с облака на смартфон. Таким способом пополняется база сведений о номерах, которую удобно использовать владельцу профиля как на смартфоне и других устройствах, так и через веб-сервис (рис. 2).

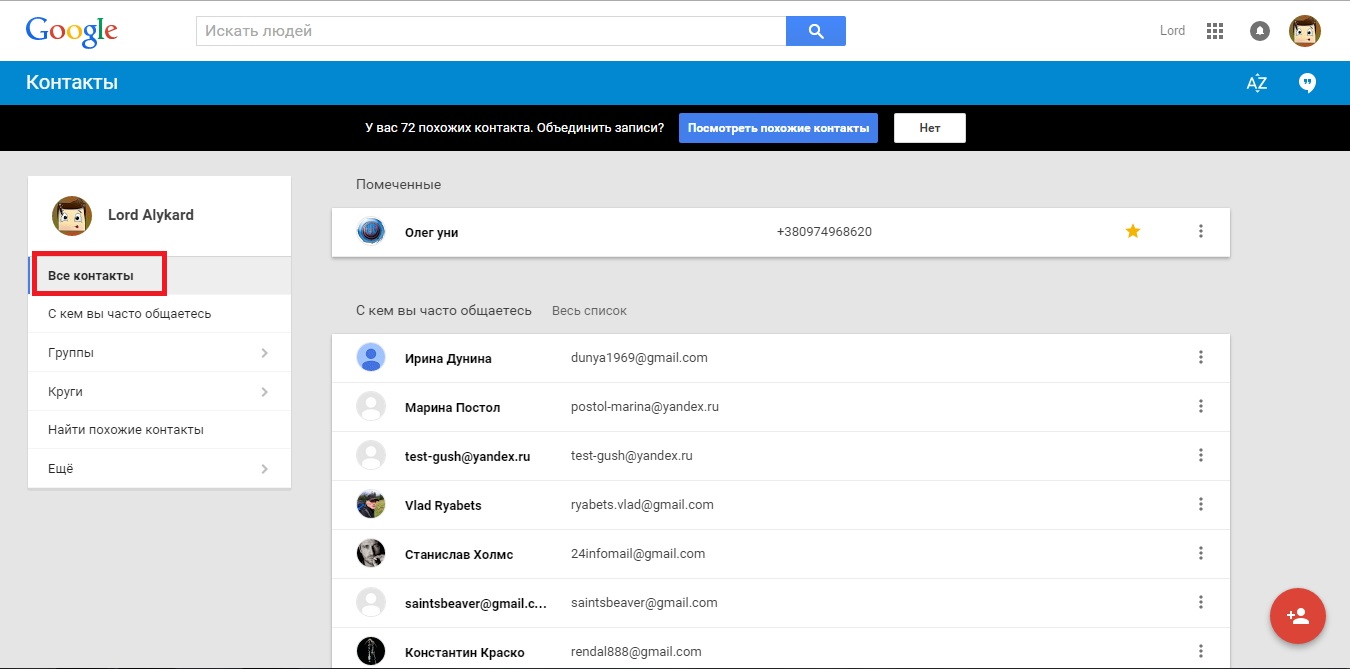


Рисунок 2 - Синхронизация контактов через Google

**Yandex**

Для управления контактами в Почте используется адресная книга. Если вы состоите в организации, в вашей адресной книге есть группы *Личные контакты* и *Общие контакты*. В общих собраны все адреса всех сотрудников вашей организации. Адресная книга формируется автоматически из адресов, на которые вы когда-либо писали. Для импорта и экспорта контактов между адресными книгами Яндекс.Почты и других почтовых сервисов и программ используются файлы формата vCard.

Для промежуточного хранения данных используется сервер Яндекса, куда информация передается по защищенному каналу, а пароли — в зашифрованном виде. Данные на сервере защищены системой авторизации, которая используется на сервисах Яндекса. Синхронизация обеспечивает доступ к данным со всех ваших устройств и восстановление данных, если устройство потерялось или сломалось.

Можно синхронизировать контакты с адресной книгой смартфона (рис. 3). Контакты со смартфона появятся в Яндекс.Почте в группе *Контакты* с телефона при соединении с интернетом (группа создается автоматически при первой синхронизации).

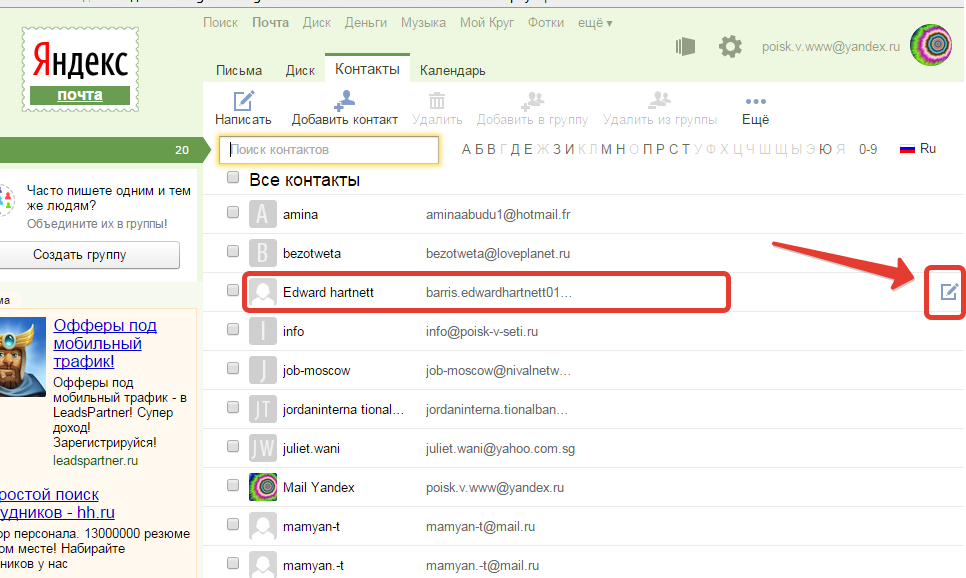


Рисунок 3 - Контакты в Yandex

Контакты синхронизируются по протоколу CardDAV. CardDAV — это интернет-протокол, который позволяет синхронизировать с мобильным устройством адресные книги. Учетную запись CardDAV можно настроить стандартными средствами смартфона или с помощью стороннего приложения.

**Наш выбор:** Google

**Обоснование выбора:** настройка достаточно простая, синхронизация возможна между всеми устройствами в сети, независимо от их версии ПО и т.д. За пользование сервисом нам не придется платить, а кол-во контактов довольно высоко (25000). Сам сервис легковесный и не требовательный к устройствам, а на Android чаще всего предустановлен, что также является преимуществом.

**Настройка синхронизации контактов**

На Android:

1. Найти и выбрать раздел «Аккаунты», «Учетные записи» или «Синхронизация» (название зависит от версии ПО и производителя);
2. На экране отобразятся все аккаунты пользователя. Среди них нужно выбрать Google-аккаунт. Если его нет, нужно нажать на кнопку «Google» и добавить профиль, введя логин и пароль;
3. Выбрав аккаунт, нужно найти в списке данных для синхронизации пункт «Контакты» и перетащить ползунок напротив него, чтобы он стал активным (рис. 4).

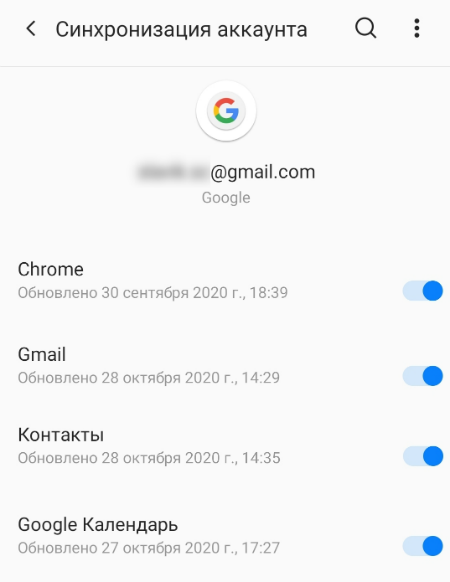


Рисунок 4 - Включение синхронизации на Android

На iOS:

1. Переходим в настройки;
2. После открытия пункта под названием «Учетные записи» в списке нажимаем на подпункт «Добавить учетную запись»;
3. В открывшемся перечне находим Google;
4. Вводим требуемую информацию в соответствующие графы;
5. Выполняем процедуру синхронизации всего, включив синхронизацию контактов (рис. 5).

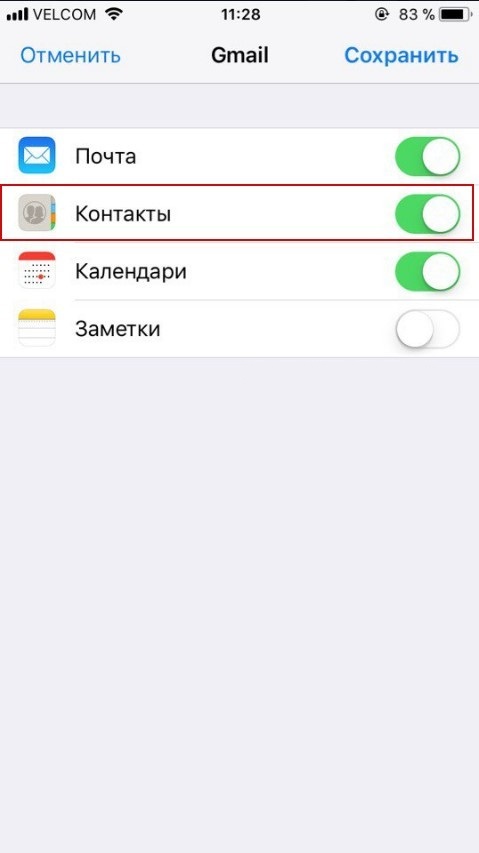


Рисунок 5 - Синхронизация контактов на iOS

На компьютере:

1. В браузерную версию *Контактов* войти можно уже привычным нам образом — при помощи меню *Приложения Google* (рис. 6).

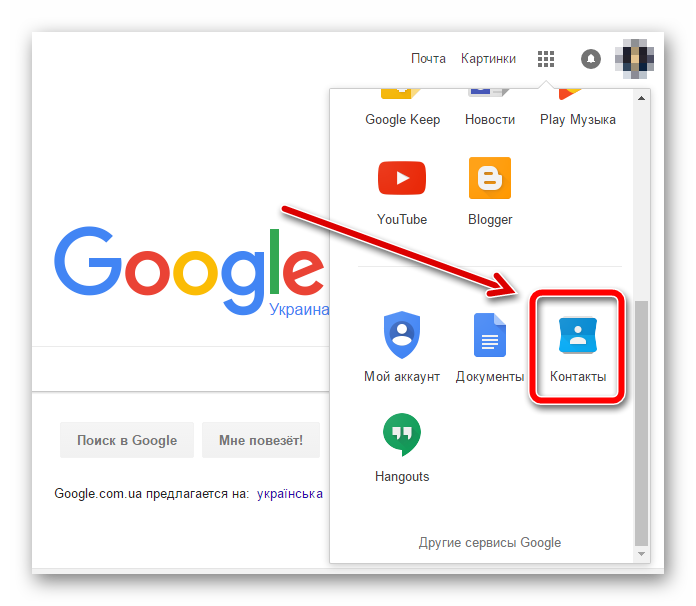


Рисунок 6 - Контакты на ПК

1. Сервис предлагает все то же, что и соответствующее приложение на вашем смартфоне: работа с имеющимися контактами, добавление новых, а также их полноценный импорт и экспорт. Интерфейс полностью идентичен стандартному на Android (рис. 7).

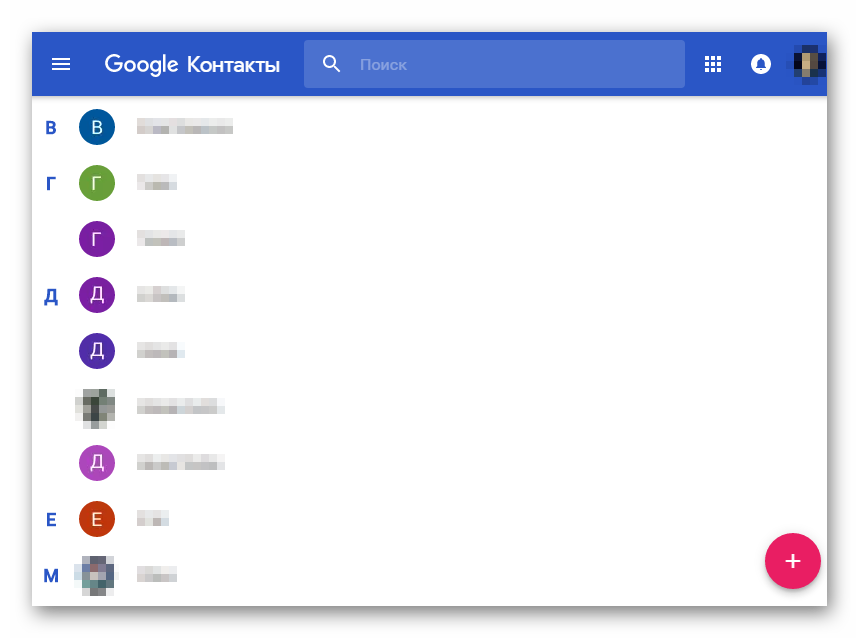


Рисунок 7 - Google контакты веб-интерфейс

## 2.2 Контроль версий

Система контроля версий является прежде всего инструментам, а инструмент призван решать некоторый класс задач. Итак, система контроля версий – это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определенной версии. Мы хотим гибко управлять некоторым набором файлом, откатываться до определенных версий в случае необходимости. Можно отменить те или иные изменения файла, откатить его удаление, посмотреть кто что-то поменял. Как правило системы контроля версий применяются для хранения исходного кода, но это необязательно. Они могут применяться для хранения файлов совершенно любого типа, как в нашем случае.

**Concurrent Versions System (CVS)**

Данная система существует с 80-х годов, она очень популярна как среди разработчиков коммерческих продуктов, так и среди разработчиков открытого программного обеспечения. CVS выпущен под лицензией GNU, система позволяет пользователям проверить код, над которым они будут работать, и записать изменения. Первоначально CVS разрешал конфликты между двумя программистами, позволяя работать над и обновлять только последнюю версию кода. Это была первая попытка создать систему контроля версий, причём в ней пользователь должен был быстро публиковать изменения, чтобы быть уверенным, что его изменения будут приняты. В противном случае обновление кода от другого пользователя могло не учесть его изменений.

Сервер CVS работает на Unix-подобных системах с мультиплатформенным клиентским программным обеспечением. Считается, что это самая зрелая система контроля версий, потому что она совершенствовалась в течение длительного времени.

**Плюсы:**

* Используется в течение многих лет и считается «зрелой» технологией.

**Минусы:**

* Перемещение или переименование файлов не включает обновление версии;
* Угрозы безопасности при использовании символических ссылок на файлы;
* Операции, связанные с ветвлениями, требовательны к ресурсам.

**Apache Subversion (SVN)**

SVN был создан как альтернатива CVS, которая исправила бы некоторые ошибки в системе CVS, сохраняя при этом обратную совместимость с ней. Как и CVS, SVN является свободным и открытым ПО с той лишь разницей, что он распространяется по лицензии Apache, а не GNU. Чтобы предотвратить повреждение базы данных, SVN использует концепцию, называемую атомарными операциями. Либо все внесенные изменения применяются, либо не применяются, что означает, что никакие частичные изменения не нарушат целостности. Многие разработчики переключились на SVN, поскольку эта система была новее, использовала лучшие функции CVS и улучшала их.

Критика SVN и CVS включает в себя более медленную скорость и отсутствие распределённого контроля версий. Распределённый контроль версий использует одноранговую (peer-to-peer) модель, а не централизованный сервер для хранения изменений кода. В то время как модель с одноранговым доступом будет работать лучше для глобальных проектов с открытым исходным кодом, она может быть не идеальной в других ситуациях. Недостатком выделенного серверного подхода является то, что когда сервер не работает, ни один пользователь не может получить доступ к коду.

**Плюсы:**

* Включает атомарные операции;
* Более «дешёвые» операции «ветвления»;
* Широкий выбор плагинов для интегрированных сред разработки.

**Минусы:**

* Все ещё содержит ошибки, связанные с переименованием файлов и каталогов.

**Git**

Git использует радикально новый подход, который сильно отличается от CVS и SVN. Git разрабатывался с целью создания более быстрой и распределенной системы контроля версий, которая бы решала проблемы CVS. Он разработан в первую очередь для Linux и демонстрирует наивысшую степень быстродействия именно там. Он также поддерживает другие Unix-подобные системы и Windows.

Git оснащён широким набором инструментов, помогающих пользователям ориентироваться в истории изменений файлов. Каждый репозиторий содержит всё дерево изменений, которое может быть полезно при разработке без подключения к интернету.

**Плюсы:**

* Увеличение скорости работы;
* «Дешёвые» операции «ветвления»;
* Полное дерево истории доступно в автономном режиме.

**Минусы:**

* Не оптимально для одиночных разработчиков;
* Ограниченная поддержка Windows по сравнению с Linux.

**Mercurial**

Разработка Mercurial началась почти одновременно с Git, и, так же как и Git, она является распределённой системой контроля версий. Для разработки ядра Linux требовалось внедрение системы контроля версий. Конкурентами за это право были Git и Mercurial, которые начали разрабатываться недавно. В итоге был выбран Git, что сделало Mercurial изначально менее популярной системой. Тем не менее, это не означает, что она не используется. Многие крупные компании используют её, в том числе OpenOffice.org. Она, в первую очередь, отличается от Git тем, что Mercurial в основном реализуется на Python, а не на C. Mercurial разделяет некоторые особенности с SVN, поэтому порог вхождения для тех, кто уже знаком с SVN, будет ниже.

**Плюсы:**

* Освоить легче, чем Git;
* Подробная документация.

**Минусы:**

* Нельзя «слить» две основные ветки;
* Медленнее, чем Git.

**Сравнение**

Для сравнительного анализа систем контроля версий воспользуемся экспертным методом оценивания, результаты которого сведены в таблицу 1.

Таблица 1 – Сравнительный анализ систем контроля версий

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Название** | **Скорость** | **Тип системы** | **Функциональные возможности** | **Документация** | **Платформы** | **Цена** | **Итого** |
| CVS | 1 б. | 0 б. | 1 б. | 1 б. | 3 б. | 1 б. | 7 б. |
| SVN | 2 б. | 0 б. | 2 б. | 2 б. | 3 б. | 1 б. | 10 б. |
| Git | 5 б. | 1 б. | 3 б. | 5 б. | 3 б. | 1 б. | 18 б. |
| Mercurial | 3 б. | 1 б. | 3 б. | 3 б. | 3 б. | 1 б. | 14 б. |

Скорость системы оцениваем по 5-балльной шкале. Согласно тестам производительности систем контроля версий, Git является наиболее быстрой

из них. Затем идет Mercurial, а за ним SVN и CVS.

По-нашему мнению, распределённая система имеет ряд преимуществ перед централизованной. Во-первых, не требует постоянного подключения к сети. Во-вторых, более надежная с точки зрения хранения данных. При централизованном подходе в случае нарушения целостности данных на сервере данные будут потеряны, чего не случится при распределённом подходе, так как пользователи сохранят локальную копию репозитория. В третьих, удобство работы с несколькими репозиториями в проекте при распределенном подходе. И потому мы оцениваем системы с этим типом более высоким баллом.

В функциональных возможностях мы оцениваем поддержку возможностей :

* копирования файлов и папок;
* переноса файлов и папок;
* отслеживания изменений в файле построчно.

CVS позволяет только отслеживать изменения в файле и, соответственно, получает 1 балл. SVN также построчно отслеживает изменения, а ещё позволяет переносить файлы и папки. Git и Mercurial поддерживают все три возможности.

Документацию мы оценивали с точки зрения новичка, то есть человека, у которого никогда не было опыта работы с данной системой. Чем очевиднее способ получения доступа к документации, чем более понятным языком она написана, тем больше баллов получает система. Наиболее удобными оказались Git и Mercurial. На их официальных сайтах прямо на главных страницах есть ссылки на документации. Сами документации написаны чётко и ясно. Что касается CVS и SVN, то их документации ориентированы на специалистов. Человеку без опыта в них трудно разобраться.

Кроссплатформенность является характерной чертой для всех систем, поэтому все получают максимальный балл.

Цена. Все системы бесплатные и получают по одному баллу.

В итоге **Git** набирает наибольшее количество баллов. И мы будем использовать эту систему контроля версий.

Для работы с выбранной системой мы будем использовать github. Сам проект будет доступен по ссылке <https://github.com/figa59rus/startup>, там же мы будем вести журнал по ходу написания работы (для этого создан файл changelog.md).

## 2.3 Общее файловое пространство на устройствах

Для общего пространства на устройствах можно использовать облачные хранилища данных, при использовании которых можно будет получить доступ к заранее созданным синхронизируемым папкам на разных устройствах. Такой функционал позволит на одном устройстве в папке производить изменения, они будут автоматически синхронизироваться с облаком, а облако будет синхронизировать уже с другими устройствами. Нам нужно рассмотреть несколько популярных сервисов для работы с облачным хранилищем и проверить какие из них поддерживают нужный нам функционал.

**One Drive**

Эта программа является предустановленной для windows 10, и автоматически позволяет синхронизировать рабочий стол windows с облачным хранилищем размером в 5 ГБ, так же можно выбрать дополнительные папки для синхронизации.

**Плюсы**:

* интеграция с Windows 10;
* можно онлайн создавать документы, таблицы или презентации;
* для настольных операционных систем – функция «Файлы по запросу»;
* надежная защита пользовательских данных — популярный алгоритм AES-256, протоколы HTTPS, TLS.

**Минусы**:

* отсутствует приложение для Linux;
* малый объем памяти — 5 ГБ.

**Яндекс.Диск**

Один из представителей облачных технологи позволяет сохранять на облаке до 10 ГБ данных, на бесплатной версии. Однако при установке, нельзя выбрать определённую папку для синхронизации, только те что уже есть в облаке. Из плюсов можно отметить скорость синхронизации.

**Плюсы**:

* поддержка веб-версий приложений Microsoft Office 365;
* бесплатное хранилище — до 10 Гб;
* приложения для Linux и iOS.

**Минусы**:

* реклама на бесплатной версии;
* неудобный интерфейс обмена контентом;
* ограниченное количество загрузок.

**Google Drive**

Самый популярный в мире поставщик облачного хранилища. Позволяет удобно работать с документами и таблицами. На бесплатном доступе выдается 15 ГБ хранилища. В плюс ещё пойдет удобный интерфейс самой программы синхронизации.

**Плюсы**:

* понятное управление — сначала папки, затем файлы;
* сервис является файловой системой для сервисов Google;
* большой объем бесплатного хранения контента — 15ГБ;
* разделение подписки между несколькими пользователями.

**Минусы**:

* при создании файла нельзя указать папку назначения.

**Сравнение**

Для сравнительного анализа систем облачного хранилища воспользуемся экспертным методом оценивания, результаты которого сведены в таблицу 2.

Таблица 2 – Сравнительный анализ систем облачного хранилища

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название** | **Скорость** | **Функциональные возможности** | **Объем хранилища** | **Платформы** | **Итого** |
| One Drive | 5 б. | 5 б. | 1 б. | 3 б. | 12 б. |
| Яндекс.Диск | 4 б. | 4 б. | 3 б. | 5 б. | 16 б. |
| Google Drive | 3 б. | 5 б. | 5 б. | 5 б. | 18 б. |

Скорость системы оцениваем по 5-балльной шкале. Согласно тестам скорости синхронизации, One Drive является наиболее быстрой из них. Затем идет Яндекс.Диск, а за ним Google Drive.

По функциональным возможностям оценивалась возможность просмотра документов на сервисе и скорость их открытия. Так как One Drive и Google Drive используют свои технологии просмотра и редактирования документов, то они в этом выигрывают, а Яндекс.Диск использует технологии One Drive, то в скорости он проигрывает.

Доступ на разных платформах это тоже важный аспект. Google Drive и Яндекс.Диск доступны на всех известных на текущий момент платформах, в то время, как One Drive не доступен на системах Linux.

По сравнению видно, что Google Drive выигрывает в сравнении, поэтому наш выбор пал на именно этот сервис хранения данных.

**Настройка Google Drive на устройствах**

Для настройки использования на разных устройствах нам будет нужно войти в Google аккаунт. Вводим логин и пароль для входа на устройстве.

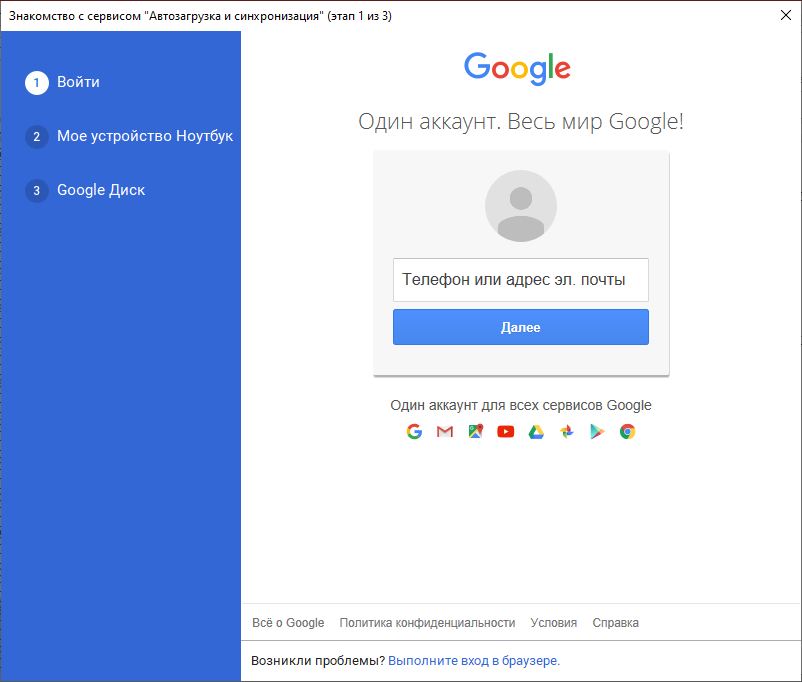


Рис 8. Вход в приложение на устройстве.

Далее нужно будет выбрать папки синхронизации, нам не нужно синхронизировать папки по умолчанию, поэтому их отключаем и добавляем папку вручную, нажав кнопку «Выбрать папку» (рис. 9). А потом нужно будет решить, синхронизировать ли всё что есть на диске с этим устройством. Так как нам это не нужно, то мы просто отключаем эту возможность (рис. 10). И нажимаем кнопку «Начать».

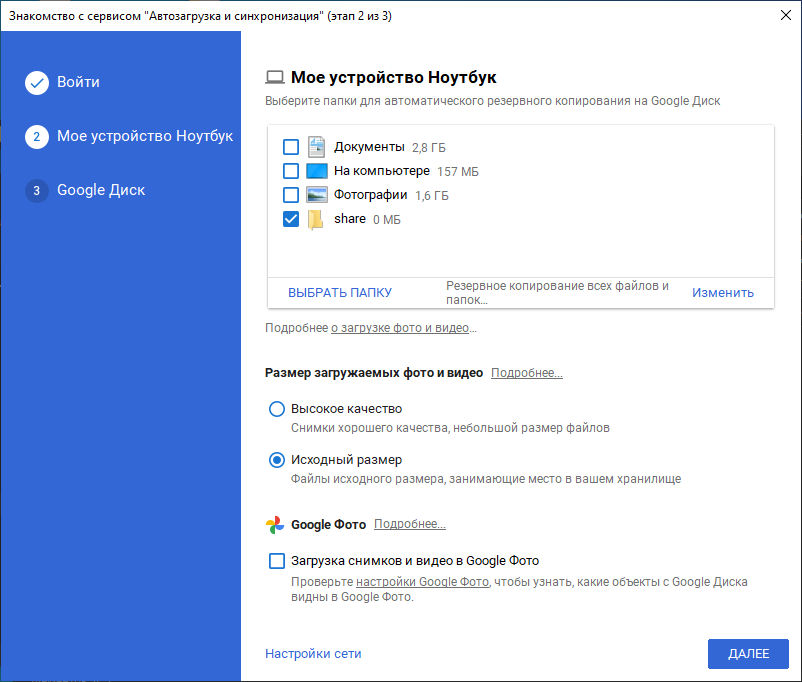


Рис 9. Добавление папки на устройстве.

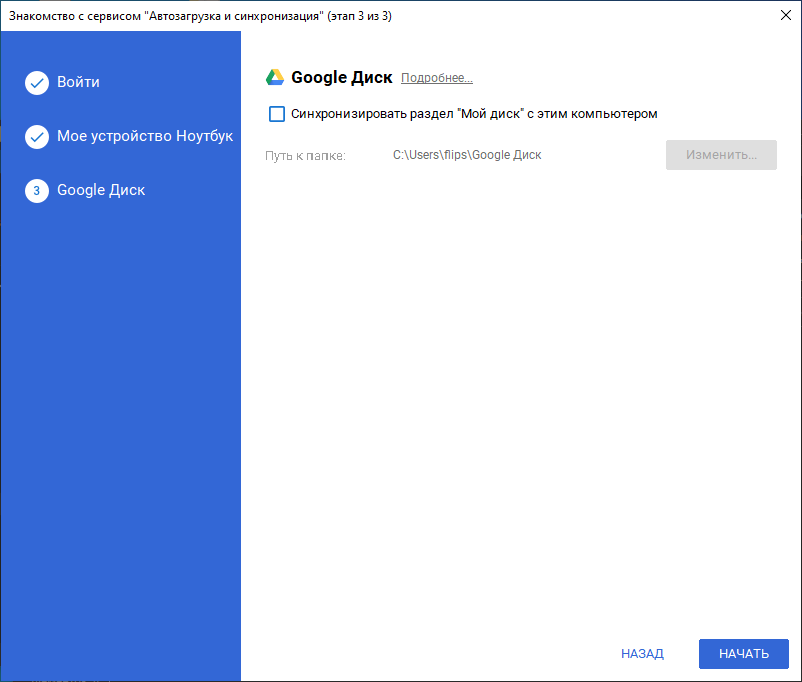


Рис 10. Синхронизация всего диска с устройством.

На этом настройка на ПК завершена, теперь что бы получить доступ к это папке на другом устройстве скачиваем приложение Google Drive, также входим в аккаунт и открываем доступные компьютеры. В списке видим именно те устройства на которых у нас уже настроена синхронизация, и мы можем получить доступ к файлам компьютера, изменять или добавлять новые, и все изменения будут отражены на основном компьютере (рис. 11).

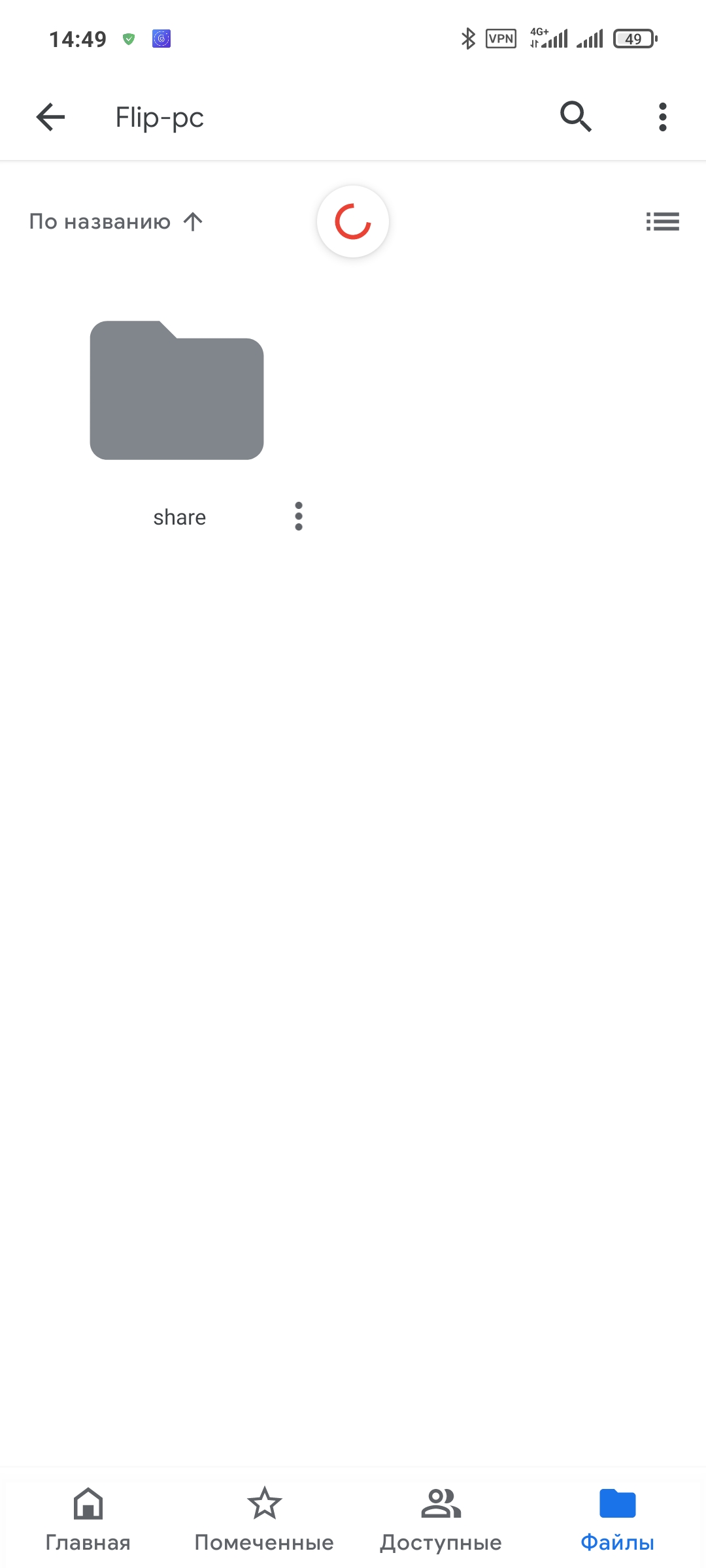
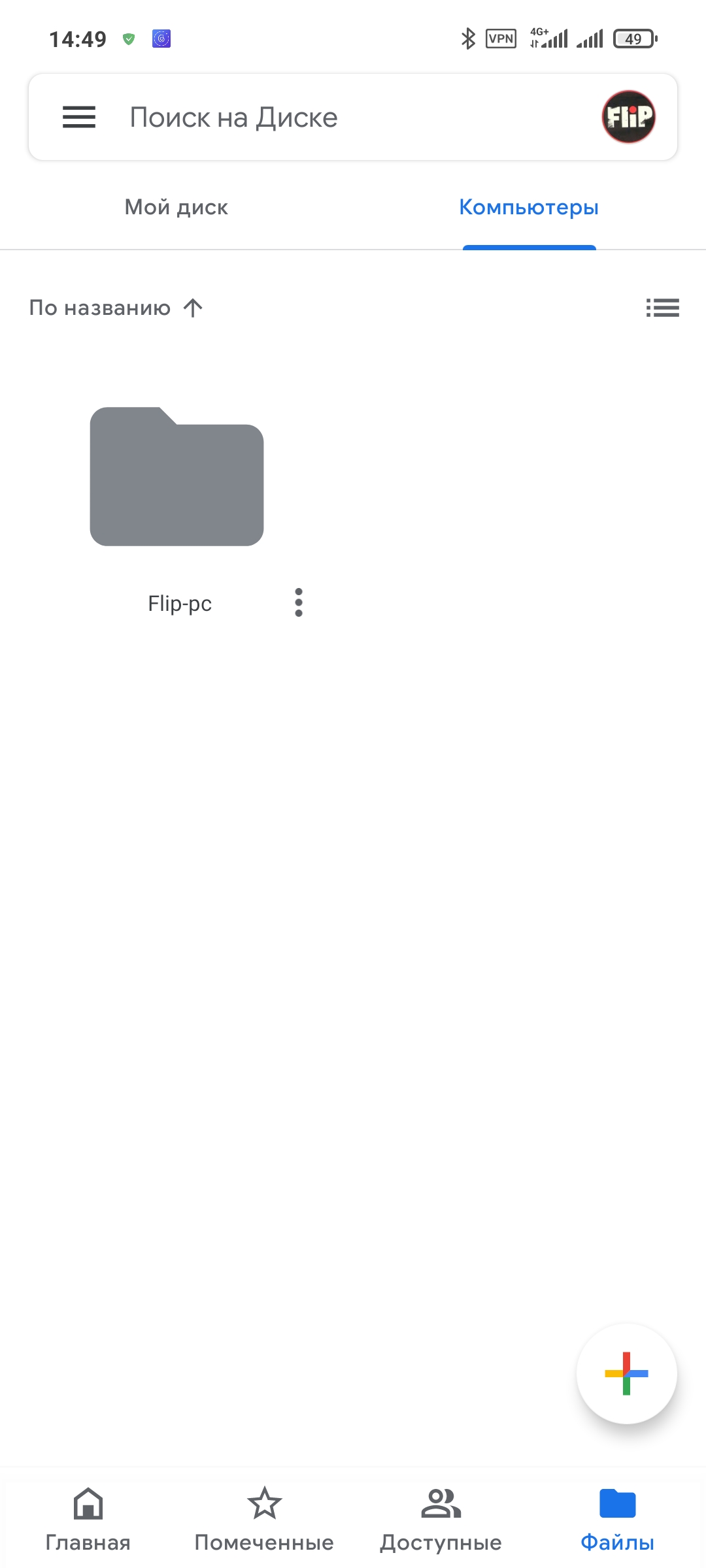


Рис 11. Доступные устройства и его синхронизируемые папки.

## 2.4 Резервное копирование файлов

Так как при облачном файловом хранилище файлы сохраняются в облаке, то при утрате или ошибочном изменении файлов, мы всегда можем отследить кто и когда менял файлы и восстановить их из истории облака. Следовательно, резервное копирование файлов у нас уже реализовано.

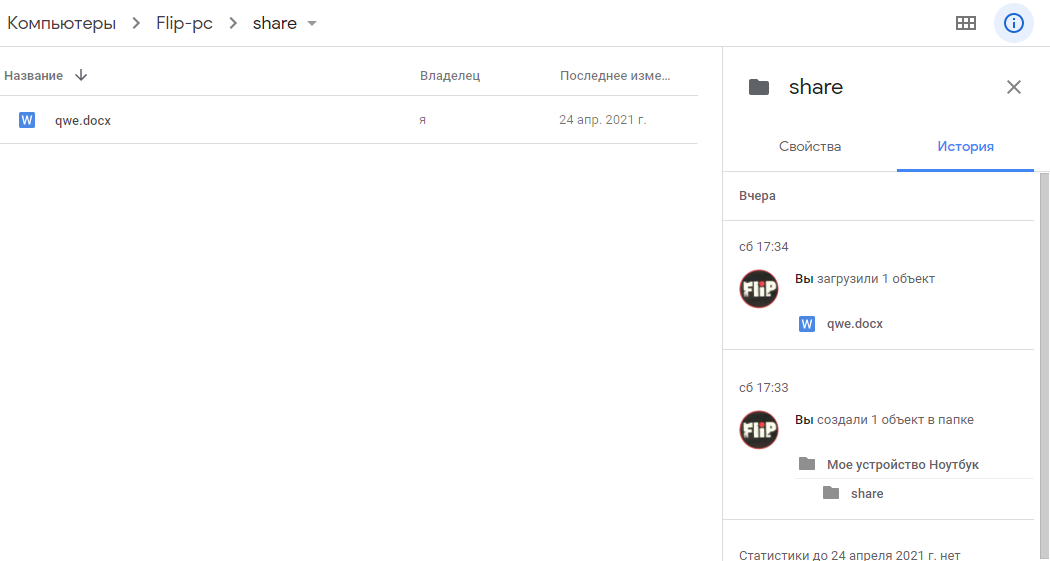


Рис 11. История взаимодействия с облаком.

# 3 Репозиторий и ведение журнала в формате markdown

Во второй главе было проведено исследование систем контроля версий и сделано их сравнение, в результате которого был выбран git.

Был создан репозиторий на GitHub. GitHub – веб-сервис хостинга проектов с использованием системы контроля версий git. Пользователи могут создавать неограниченное число репозиториев, для каждого из которых предоставляется wiki, система issue tracking-а, есть возможность проводить code review и многое другое. На данный момент является самым популярным сервисом среди своих конкурентов в этой области.

Репозиторий можно найти по ссылке <https://github.com/figa59rus/startup>.

Создан файл README, который содержит подробное описание проекта.

Также создан файл changelog для журнализации изменений проекта, в котором мы протоколировали все изменения, которые были внесены нами в проект.

Для файлов README и changelog используется язык разметки Markdown, созданный для обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций (например, HTML). Главной особенностью данного языка является максимально простой синтаксис, который служит для упрощения написания и чтения кода разметки, что, в свою очередь, позволяет легко его корректировать.

# 4 Исследование на уязвимости

# 5 Комплекс мер по защищенности

Для создания комплекса мер по защищенности, как правило, руководство предприятия или организации должно разработать и внедрить концепцию по обеспечению информационной безопасности. Этот документ является основополагающим для разработки внутренних регламентов и системы защитных мер. Разработку политики безопасности часто поручают приглашенным экспертам по защите информации, способным провести аудит как информационной системы, так и организационной структуры компании и ее бизнес-процессов и разработать актуальный комплекс мер по защите информации. У нас же, по очевидным причинам, таких возможностей нет, поэтому будем пытаться написать комплекс мер своими силами.

# Заключение