# 

# 

# 

# Общий регламент работы АО “Т-Информ”

## Распорядок рабочего дня

1. График работы рабочего дня: 10:00 - 19:00.
2. Перерыв на обед для приема пищи: 14:00 - 15:00 (время обеда может быть перенесено на другое время по согласованию с непосредственным руководителем, но продолжительность должна быть не более 1 часа).
3. В остальное время перекусы и кофе-брейк не запрещаются.
4. Каждый рабочий час сотрудник может сделать пятиминутный перерыв.
5. Запрещается устанавливать софт, не имеющий отношения к работе.
6. Все рабочее время должно быть посвящено работе над задачами. На рабочем месте в рабочее время запрещается заниматься посторонними делами.

## Корпоративные ресурсы

В нашей компании созданы корпоративные ресурсы для обеспечения удобного и комфортного пребывания на рабочем месте:

1. <https://my.corp.local> - внутренний сайт компании, на котором можно найти много полезной информации: новости, дни рождения и другая информация о коллегах, геймификация, соц. сети компании, вне офисная деятельность, ссылки на другие корпоративные ресурсы и так далее.
2. <https://webmail.corp.local> - корпоративная электронная почта для обмена внутренней информацией между коллегами. Также здесь приходят рассылки о различных мероприятиях, корпоративах, учебных курсах, незаполненных отчетах и много другой полезной информации.
3. <https://reporter.corp.local> - внутренняя система учета рабочего времени и прохождения ежегодной аттестации сотрудника. Более подробная инструкция по работе с этой системой будет описана в соответствующем разделе регламента.
4. <https://br.srvdev.ru> - система управления процессом разработки, в которой ведется вся работа от постановки до выполнения задач. Более подробная инструкция по работе с этой системой будет описана в соответствующем разделе регламента.
5. <https://gitlab.srvdev.ru> - корпоративный гитлаб для внутренней разработки проектов.
6. \\fs01.corp.local - хранилище архивов: прошедших обучений, фото и видео с корпоративов, книги и многое-многое другое.
7. Корпоративный VPN - VPN-соединение, необходимое для доступа к корпоративным ресурсам. Более подробная инструкция по настройке VPN будет описана в соответствующем разделе регламента.

Доступ к корпоративным ресурсам выдается непосредственным руководителем в виде логина и пароля.

## Инструкция по настройке VPN-подключения по протоколу SSTP для операционных систем MS Windows (10, 11)

1. В меню «Пуск» выбрать пункт «Параметры» (Рисунок 1)

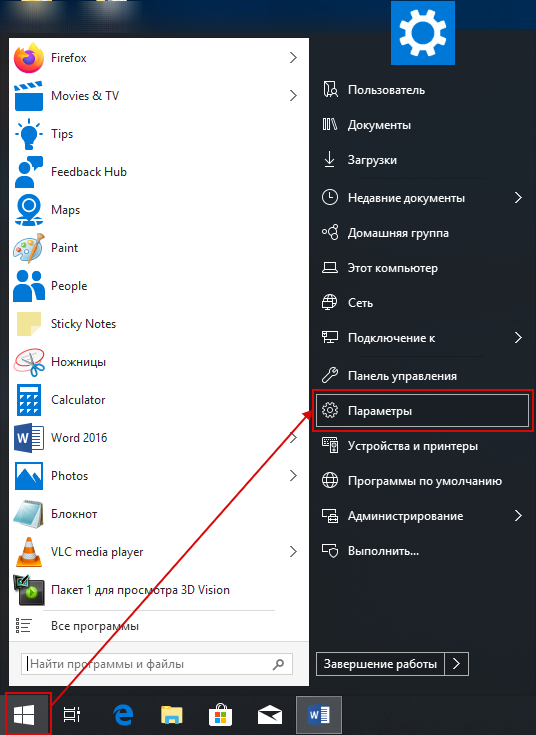


Рис. 1

2. В окне «Параметры Windows» выбрать раздел «Сеть и Интернет» (Рисунок 2)

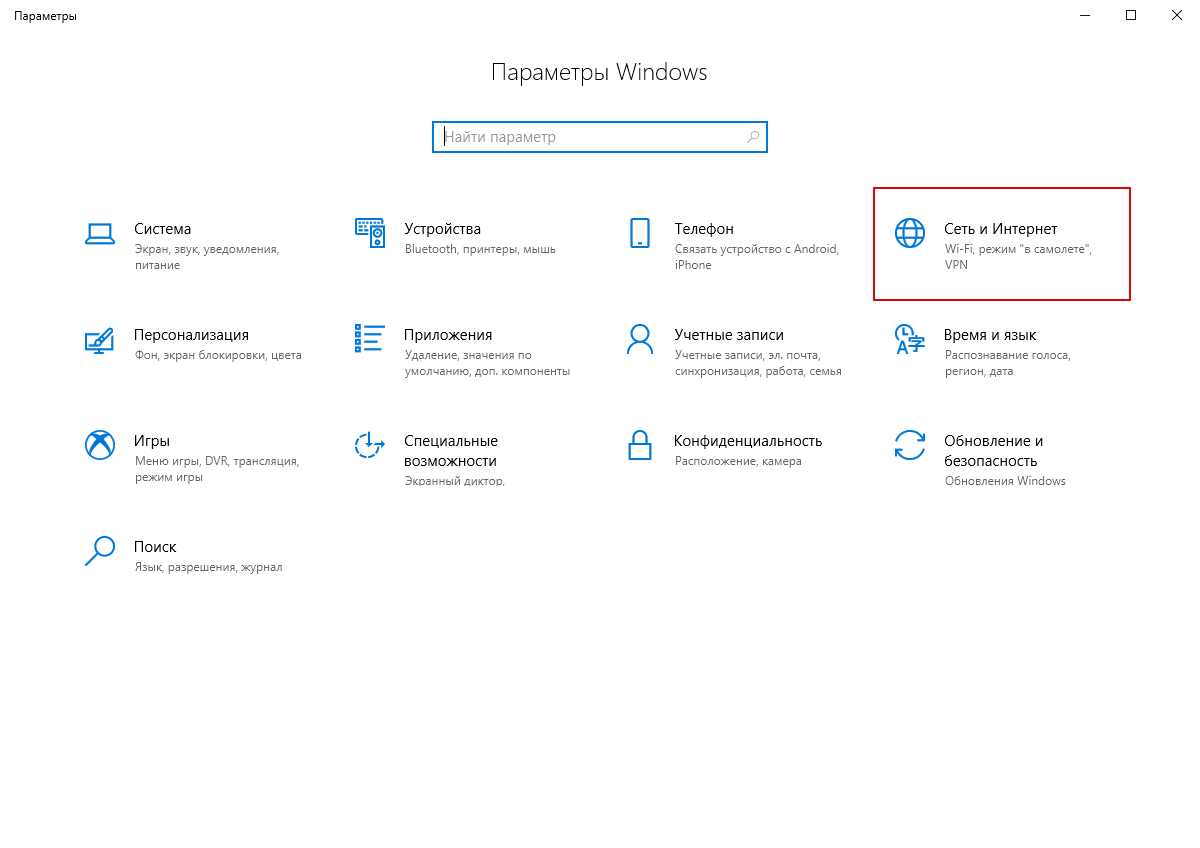


Рис. 2

3. В левой части окна выбрать пункт «VPN», нажать «Добавить VPN-подключение» (Рисунок 3)

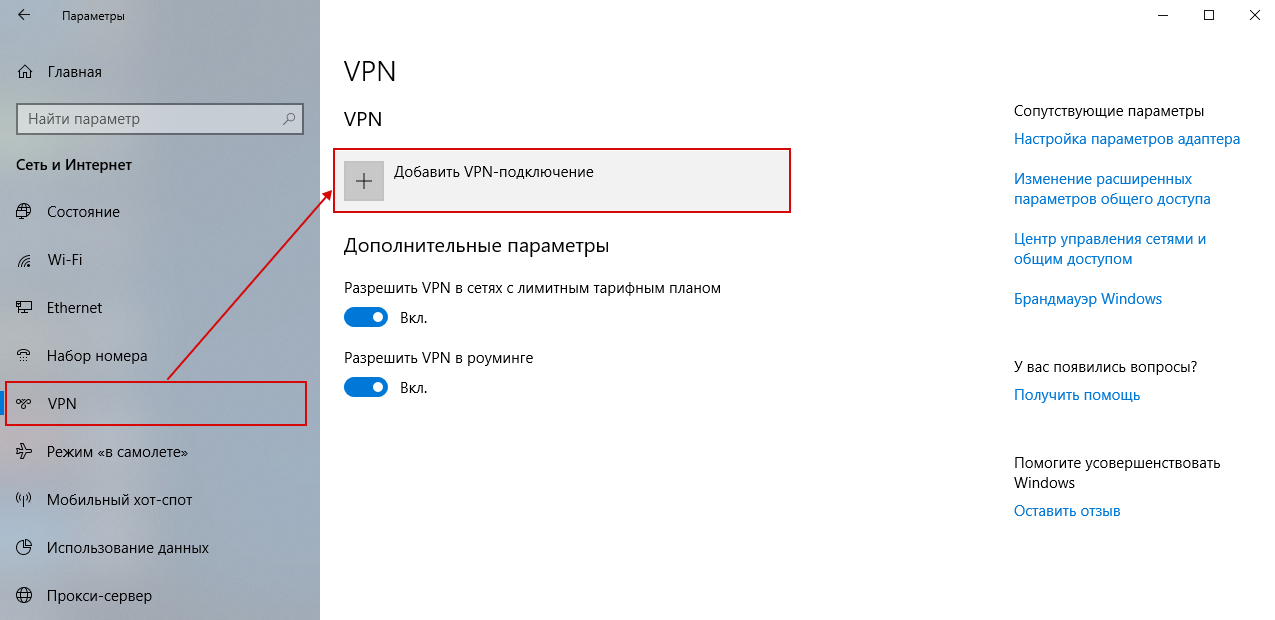


Рис. 3

4. В окне «Добавить VPN-подключение» заполнить поля, нажать кнопки «Сохранить» (Рисунок 4)

⦁ В поле «Поставщик услуг VPN» выбрать значение «Windows (встроенные)»;

⦁ В текстовом поле «Имя подключение» указать значение «vpn.corp»;

⦁ В текстовом поле «Имя и адрес сервера» указать значение «sstpt.srvdev.ru»;

⦁ В поле «Тип VPN» выбрать значение «Протокол SSTP»;

⦁ В поле «Тип данных для входа» выбрать значение «Имя пользователя и пароль»;

⦁ Текстовые поля «Имя пользователя» и «Пароль» заполнить параметрами, которые были переданы по электронной почте.

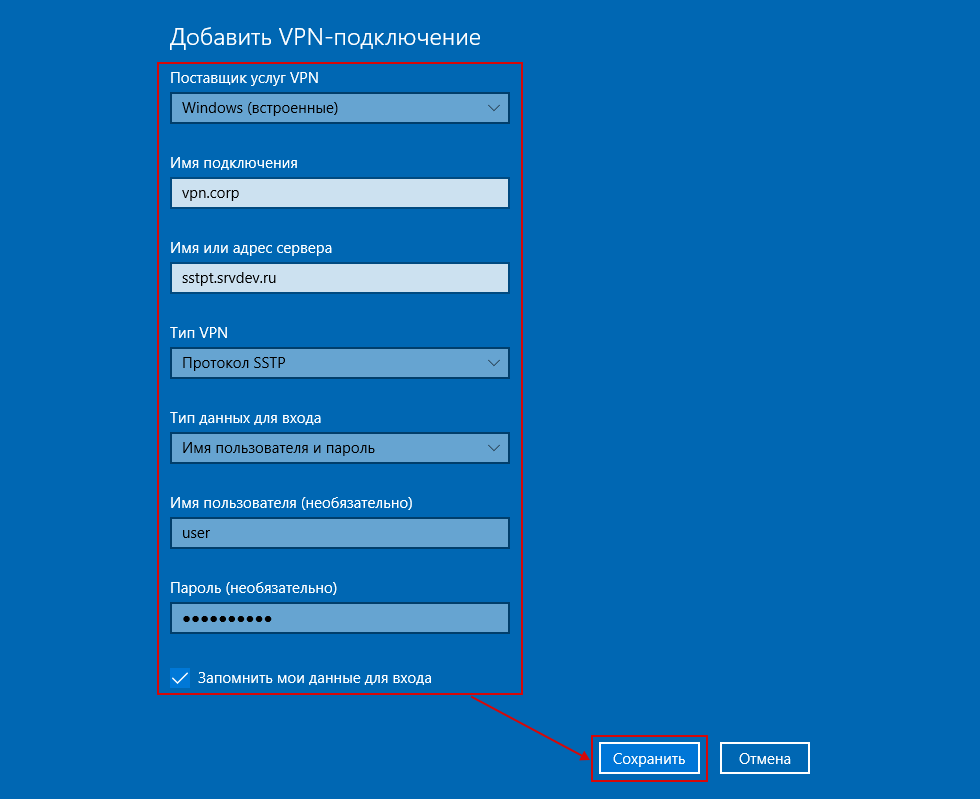


Рис. 4

5. Для работы сети Интернет необходимо произвести настройку. В блоке «Сопутствующие параметры» выбрать «Настройка параметров адаптера» (Рисунок 5)

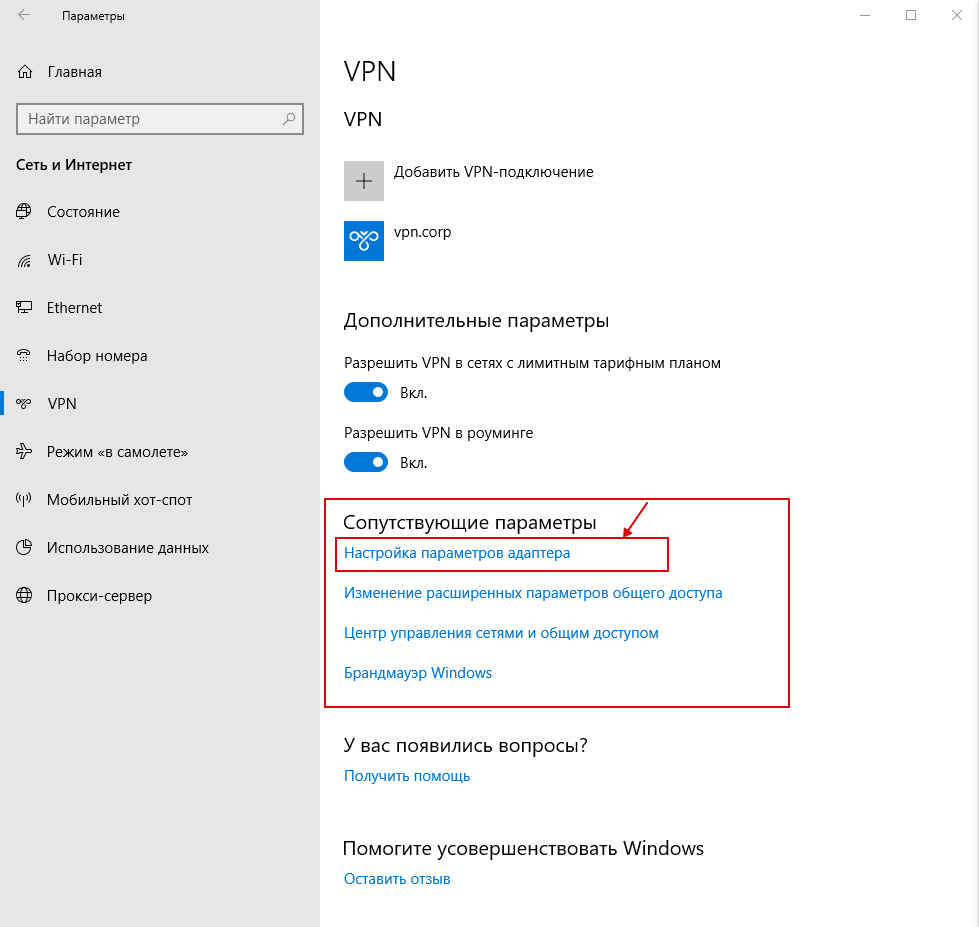


Рис. 5

6. В открывшемся окне «Сетевые подключения» правой кнопкой мыши кликнуть на ярлыке «vpn.corp», выбрать «Свойства» (Рисунок 6)

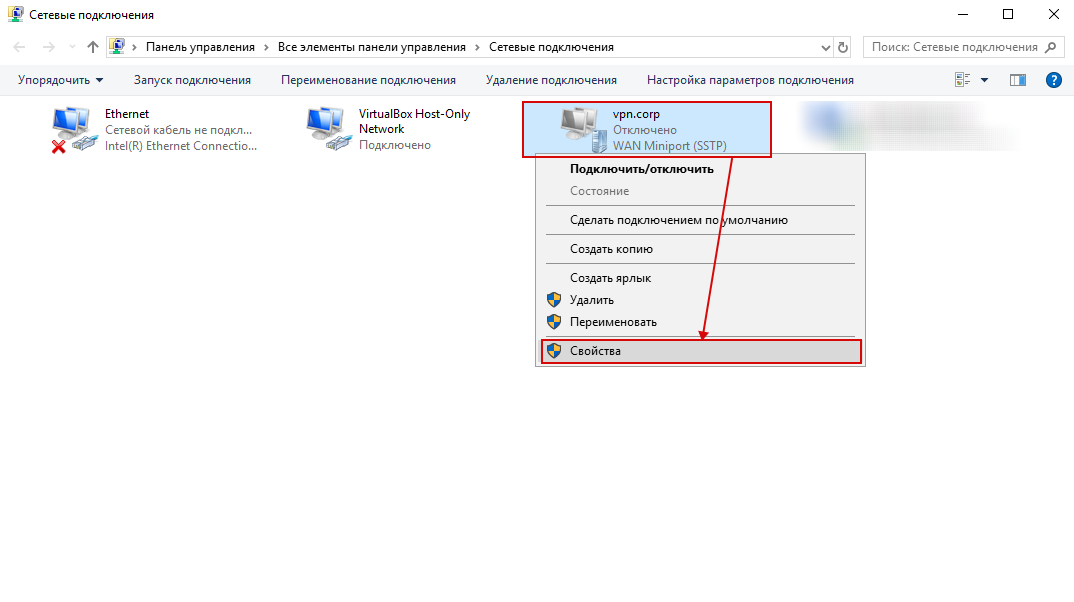


Рис. 6

7. В окне «vpn.corp: свойства» выбрать вкладку «Сеть». При использовании курсора и левой кнопки мыши, в списке «Компоненты, используемые этим подключением» выбрать значение «IP версии 4 (TCP / IPv4)». Нажать кнопку «Свойства» (Рисунок 7)

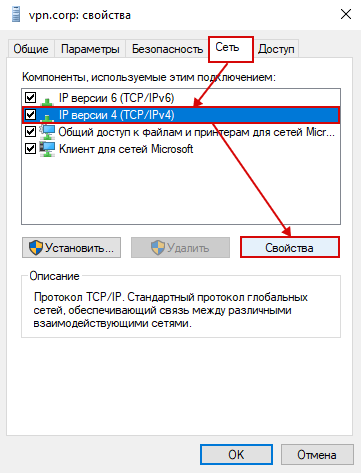


Рис. 7

8. В окне «Свойства: IP версии 4 (TCP/IPv4)» нажать кнопку «Дополнительно» (Рисунок 8)

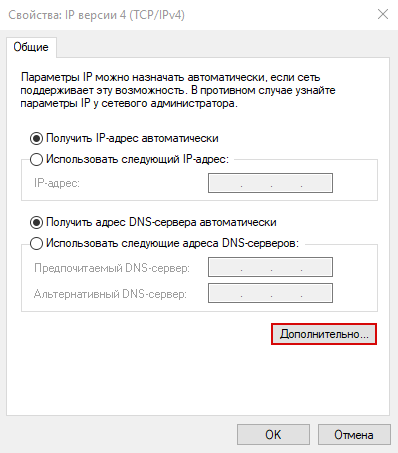


Рис. 8

9. В открывшемся окне «Дополнительные параметры TCP/IP» на вкладке «Параметры IP» снять чек-бокс (флаг) «Использовать основной шлюз для удалённой сети», нажать кнопку «ОК» (Рисунок 9)

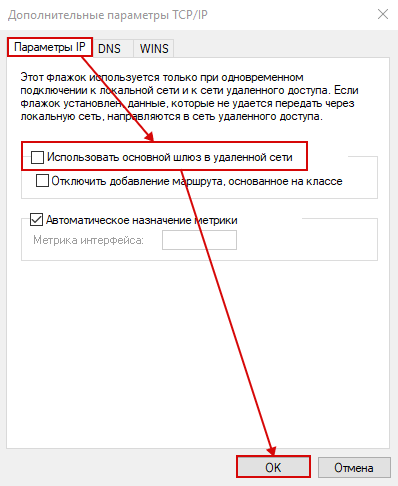


Рис. 9

10. Закрыть окно «Свойства: IP версии 4 (TCP/IPv4)» кнопкой «ОК», чтобы настройки вступили в силу (Рисунок 10)

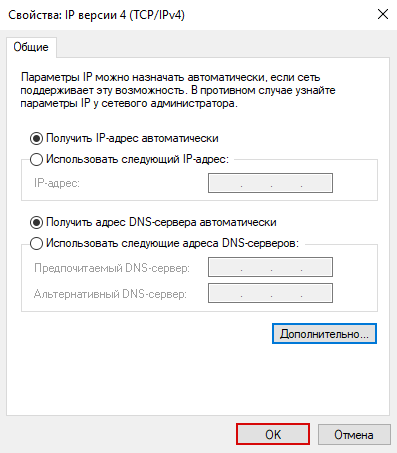


Рис. 10

11. Закрыть окно «vpn.corp: свойства» с использованием кнопки «ОК» (Рисунок 11)

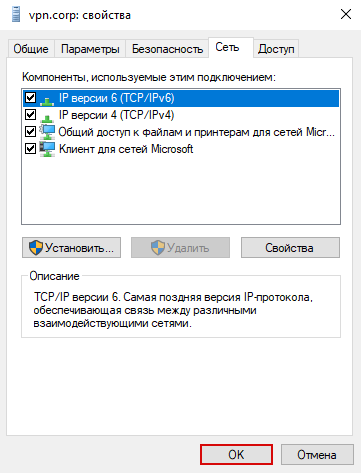


Рис. 11

12. Для удобства использования, ярлык подключения можно разместить в удобном месте. Для этого необходимо в окне «Сетевые подключения» правой кнопкой мыши кликнуть на ярлыке «vpn.corp», выбрать «Создать ярлык» (Рисунок 12)

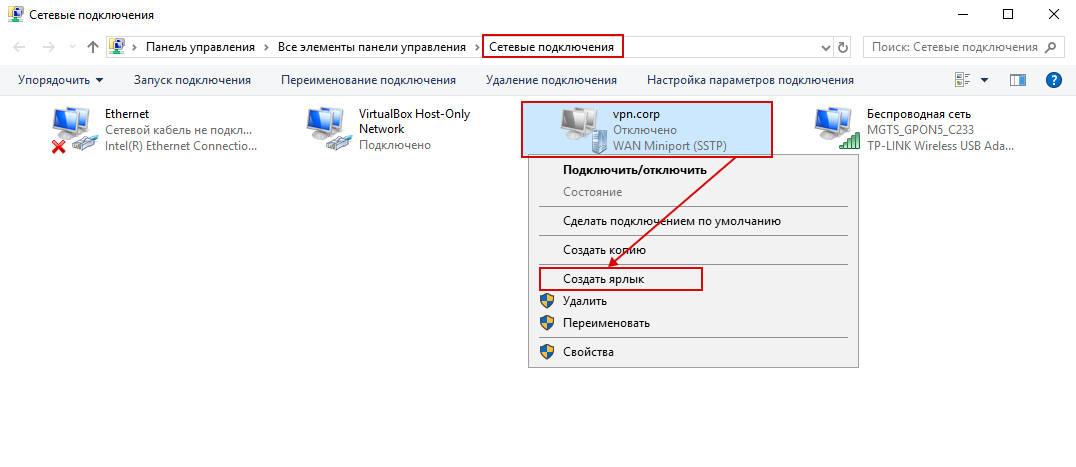


Рис. 12

**Важно**: После выполнения настройки VPN необходимо перезагрузить компьютер!

## Инструкция по работе системы учета рабочего времени (<https://reporter.corp.local>)

### Заполнение ежедневного отчета

В конце каждого рабочего дня сотрудник создает одну или несколько

записей, в которых указывает: Проект, Вид работы, Количество потраченных на работу часов, Краткий отчет о выполненной работе (1-2 предложения) желательно с указанием номеров задач из системы постановки задач. В сумме сотрудник должен отчитаться за 8 часов рабочего времени ежедневно.

Для заполнения ежедневного отчета необходимо:

1. Нажать кнопку “+” напротив необходимо дня или кнопку “Добавить отчет” (Рисунок 1).

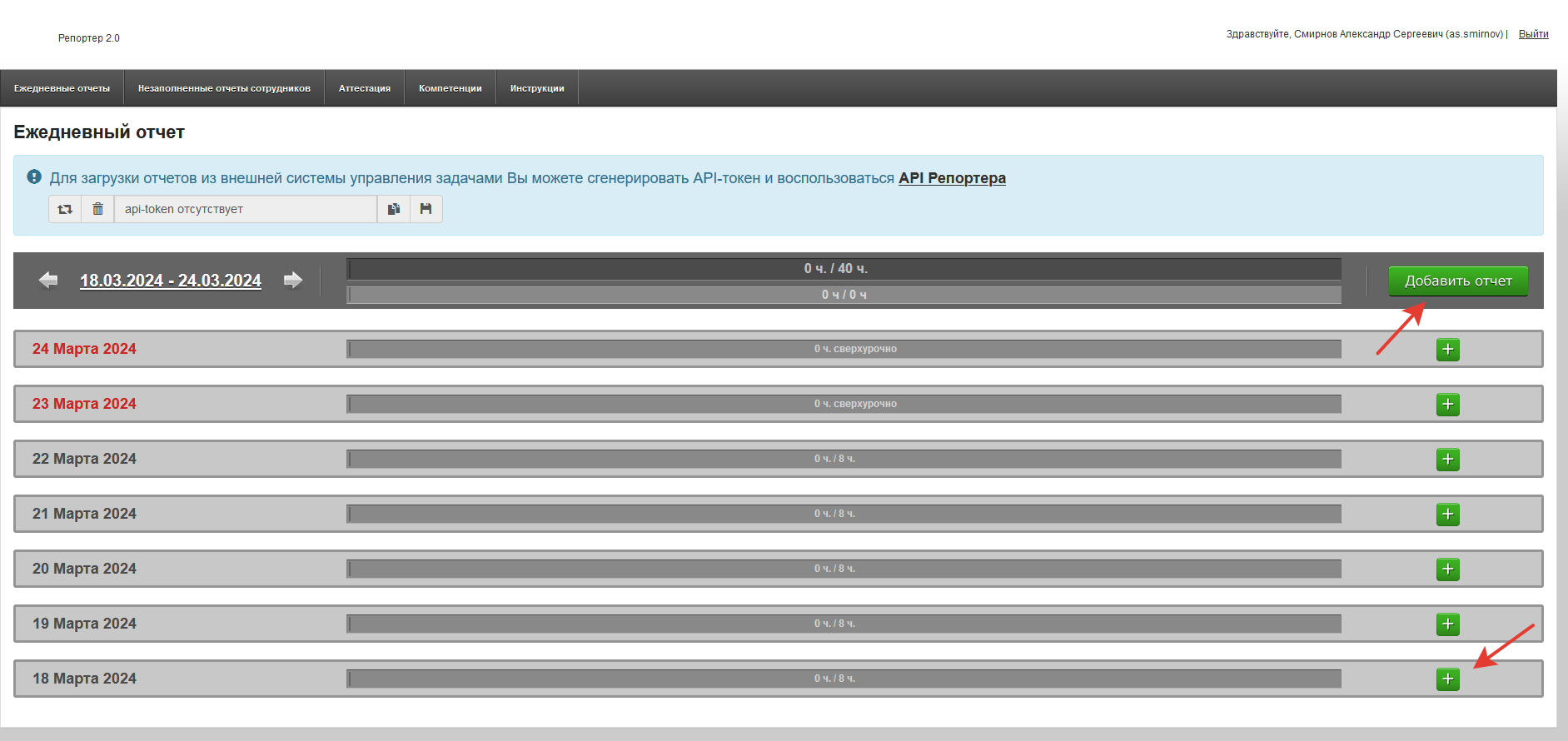


Рис. 1

1. В открывшейся форме выбрать (Рисунок 2):
   1. проект из списка, по которому выполнялась задача (если нужного проекта в списке нет, необходимо добавить новый проект по его названию, который можно уточнить у непосредственного руководителя)
   2. тип задачи: Производство - Разработка программного кода
   3. время выполнения задачи
   4. текстовое описание, что было сделано с указанием номера задачи из системы постановки задач
2. Нажать кнопку “Добавить отчет” (Рисунок 2).

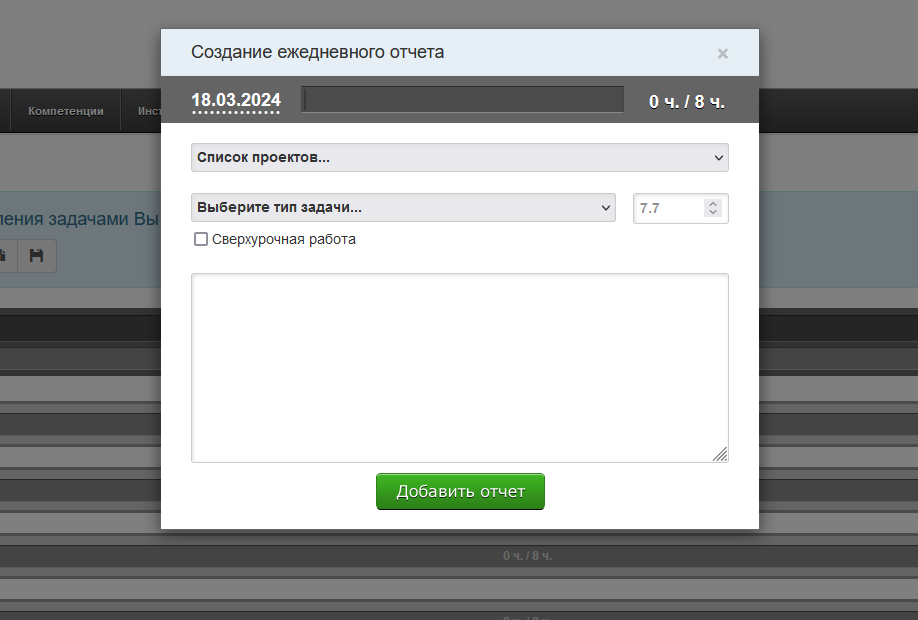


Рис. 2

### Заполнение сверхурочного времени

В случае, если по производственной необходимости приходится потратить более 8 часов за один рабочий день, отчет на дополнительные часы работы заполняется таким же образом с единственным отличием - сверхурочная работа помечается соответствующей галкой (Рисунок 2).

### Заполнение времени во время отпуска

Для списания времени на отпуск необходимо на форме выбрать (Рисунок 3):

1. Проект - Административный проект
2. Отсутствие - Отпуск
3. Время 8 часов
4. При помощи календаря, который появляется при нажатии на дату, выбрать период отсутствия
5. В поле комментария написать: Ежегодный оплачиваемый отпуск

**Важно:** Заполнить таким образом отчет необходимо в последний рабочий день перед отпуском!

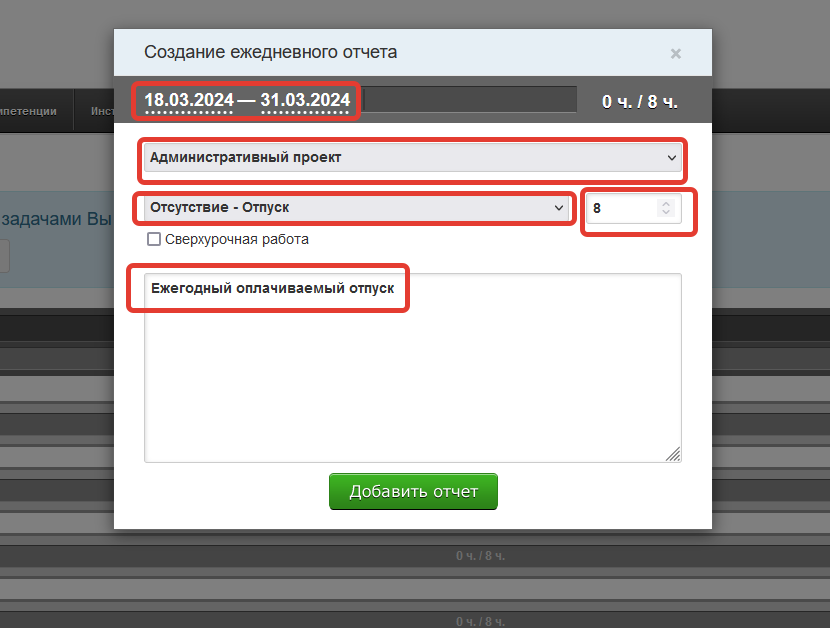


Рис. 3

### Заполнение времени во время больничного

Заполняется точно таким же образом, как и при отпуске, за исключением двух пунктов (Рисунок 4):

1. Отсутствие - Больничный
2. Комментарий - Больничный

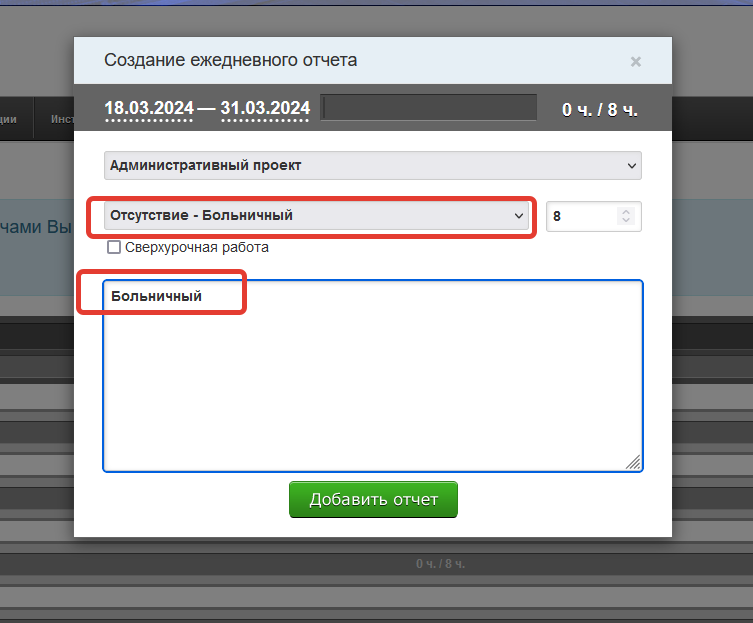


Рис. 4

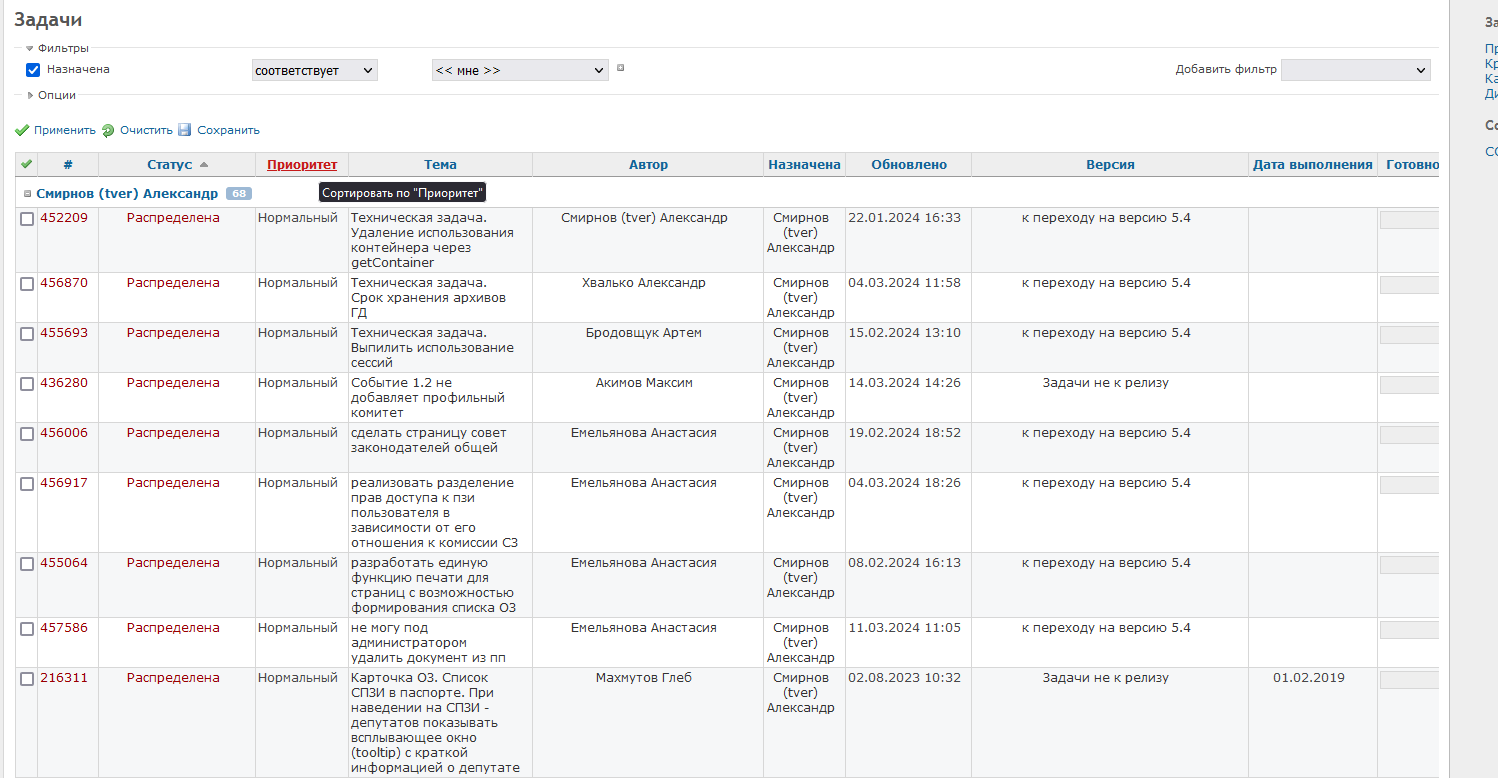
### Прохождение аттестации

*Нет возможности сделать скрины для заполнения данного раздела - дополню в будущем*

## Система управление процессом разработки (<https://br.srvdev.ru>)

### Общие сведения

В разделе “Задачи” каждого проекта находится перечень работ, который необходимо выполнить. При помощи фильтров, сортировки и других настроек можно настроить страницу под свои нужды. Порядок выполнения задач определяется версией релиза (от ближайшего к будущему) и приоритетом (от немедленного к низкому).



Иногда в разделе “Wiki” разработчики ведут некоторую документацию по настройкам проекта, информацию о предстоящих и прошедших релизах и другую техническую информацию.

В случае отсутствия задач необходимо об этом сообщить непосредственному руководителю и менеджеру проекта.

### Выполнение задачи

1. Приступая к работе над задачей, необходимо поставить ей статус “Выполняется”. Этот факт свидетельствует о том, что сотрудник приступил к задаче, и с этого момента считается время на её выполнение. В конкретный момент времени только одна задача должна быть со статусом “Выполняется”.
2. В случае возникновения вопросов по задаче к её постановщику:
   1. Необходимо обратиться в чат проекта (в мессенджере), выделив постановщика задачи соответствующим тегом. Это делается для того, чтобы ведущий разработчик и менеджер проекта были в курсе этого вопроса. Также, есть вероятность, что другие коллеги смогут оперативнее ответить на вопрос, что ускорит выполнение задания.
   2. Допускается обсуждение каких-то незначительных или уточняющих моментов по задаче в личном общении (в мессенджере), но все принятые решения должны быть зафиксированы в постановке задаче или в комментариях к ней.
   3. Если нет контактов для общения с постановщиком или ответ на возникший вопрос не получен в течение 5-10 минут, необходимо описать свой вопрос в комментариях к задаче и перевести её на постановщика со статусом “Приостановлена”. При этом обязательно следует указать потраченное время на выполнение задачи - это нужно для того, чтобы потраченное время на задачу было учтено, а также для учета скорости умения сотрудника разбираться с новыми поставленными задачами.
3. В случае остановки работы по задаче по какой-либо причине, необходимо поставить у задачи статус “Приостановлена” с обязательным списанием потраченного времени и комментарием с описанием причины остановки.
4. Когда задача выполнена, необходимо поставить статус “Сделана”, списать время и перевести её на постановщика для проверки.

### Учет времени работы над задачей

Во время выполнения задачи необходимо вести учет времени. Время считается от момента выставления статуса “Выполняется” и до одного из следующих событий:

1. Задача выполнена.
2. Окончание рабочего дня, если выполнение задачи откладывается на следующий рабочий день.
3. Приостановка работы над задачей по какой-либо причине.

При списании времени обязательно надо указать комментарий, что было сделано за это время. Это делается для учета скорости выполнения задачи сотрудником.

Суммарное время, списанное на задачи за день в системе управления процессом разработки, должно совпадать с временем, списанным за день на проекте в системе учета рабочего времени.

### Работа в режиме многозадачности

С увеличением вовлеченности в деятельность компании у сотрудника появляются новые проекты, между которыми приходится распределять своё рабочее время. Даже в рамках одного проекта могут возникать ситуации, когда приходится выполнять несколько задач одновременно.

Для качественного выполнения задач в каждый конкретный момент времени необходимо выполнять только одну задачу, а также следовать следующему регламенту:

1. Сотрудник не должен переключаться на выполнение новой задачи без уведомления менеджера текущего проекта (даже в случае большей приоритетности и уговоров менеджера с другого проекта сделать “по-быстрому”).
2. Согласование переключения с проекта на проект производится менеджерами проектов. Только после их согласования между собой сотрудник может переключиться с текущей задачи на другую.
3. В случае затруднительной ситуации необходимо обратиться к непосредственному руководителю.

Нарушение данного регламента не допускается!

# 

# 

# 

# Технический регламент работы АО “Т-Информ”

## Разработка на локальном компьютере (Docker)

### Общие сведения

Все проекты нашей компании переведены на разработку на локальном компьютере. Такая система позволяет работать в случае отсутствия или сбоя интернета, а также работать с разных устройств.

Ниже приведена инструкция по настройке локального рабочего места на основе Windows 10 (минимальная сборка 2004) или Windows 11. Для других операционных систем данную настройку следует выполнять самостоятельно.

### Настройка WSL

WSL - подсистема Windows для Linux, которая позволяет устанавливать дистрибутив Linux в качестве виртуальной машины.

Для установки WSL необходимо открыть командную строку в режиме администратора и выполнить команду:  
wsl --install

После завершения установки необходимо создать учетную запись администратора в системе Linux.

Далее необходимо убедиться, что была установлена версия WSL 2. Для этого в командной строке необходимо выполнить:

wsl -l -v

Если была установлена первая версия, то необходимо обновить её до версии 2 командой:

wsl –set-version Ubuntu-20.04 2

где Ubuntu-20.04 - версия устанавливаемого дистрибутива Linux.

В корне пользовательской директории Windows (C:\Users\{текущий пользователь}) необходимо создать файл .wslconfig со следующим содержимым для более оптимизированной работы WSL:

[wsl2]

memory=5GB

swap=0

localhostForwarding=true

В адресной строке окна Windows можно получить доступ к файлам Linux, введя адрес:

\\wsl$

В случае возникновения проблем или потери актуальности данной инструкции следует воспользоваться [официальной документацией по установке WSL](https://learn.microsoft.com/ru-ru/windows/wsl/install).

### Установка Docker Desktop

Затем следует установить приложение для управления докер-контейнером из-под Windows: <https://www.docker.com/products/docker-desktop>

### Создание SSH-ключа

Для получения доступа к проекту в гитлабе и возможности скачать его по SSH, необходимо сгенерировать ключ.

Пуск - Ubuntu

В открывшейся консоли переходим в корневую папку пользователя:  
**cd ~**

Проверяем содержимое папки командой:  
**ls -la**

Если в списке каталогов нет папки .ssh, то создаем её командой:  
**mkdir .ssh**

Переходим в эту папку и генерируем ключ:

**cd ./.ssh  
ssh-keygen -t rsa**

Нажимаем Enter, система предложит ввести кодовую фразу для дополнительной защиты ключа (этот шаг можно пропустить и оставить поле пустым). После того, как ключ будет создан выполняем команду:

**cat ./id\_rsa.pub**

На экран будет выведен ключ, который необходимо скопировать и вставить в специальное поле в профиле корпоративного гитлаба после авторизации:  
<https://gitlab.srvdev.ru/-/profile/keys>

### Клонирование проекта с корпоративного гитлаба

Создаем папку под проекты в домашнем каталоге, а также папку для конкретного проекта (name\_project может быть любым понятным для вас названием), после чего выполняем клонирование:

**cd ~**

**mkdir projects (если не была ранее создана)**

**cd projects**

**mkdir name\_project**

**cd name\_project**

**git clone git@gitlab.srvdev.ru:duma/name\_project.git .**   
(точка в конце последней команды обязательна)

### Запуск локального сервера

Обычно в корне проекта лежит папка с названием Docker, а в ней папка ComposeDev, в которую и надо перейти для запуска локального сервера:

**cd ~/projects/name\_project/Docker/ComposeDev**

Далее следует запустить команды:  
**docker-compose build**

**docker-compose up -d**

В результате вы должны увидеть, что все контейнеры запустились. Ту же информацию можно увидеть в приложении Docker Desktop.

### Добавление и настройка проекта в PHPStrom

Заходим в PHPStorm и выбираем в меню:

File -> New Project…

В появившемся окне указываем расположение папки name\_project в Ubuntu и нажимаем OK и Create.

После добавления проекта PHPStorm (в зависимости от версии) может предложить подключиться к локальному докеру. Если этого не произошло, то нужно подключиться вручную: для этого в настройках File -> Settings -> Build, deploy находим Docker и добавляем новое подключение с дефолтными настройками.

После добавления подключения к докеру в нижней части экрана появится дополнительная вкладка Services, на которой можно увидеть все докер-контейнеры всех подключенных приложений.

### Доступ к приложению через браузер

В файл C:\Windows\System32\drivers\etc\hosts добавляем:

127.0.0.1 name\_project.local

Теперь в браузере приложение доступно по адресу name\_project.local

### Настройка локальной базы данных

Для настройки локальной базы данных необходимо запросить текущий дамп БД у команды разработки проекта и развернуть его в используемой на проекте СУБД.

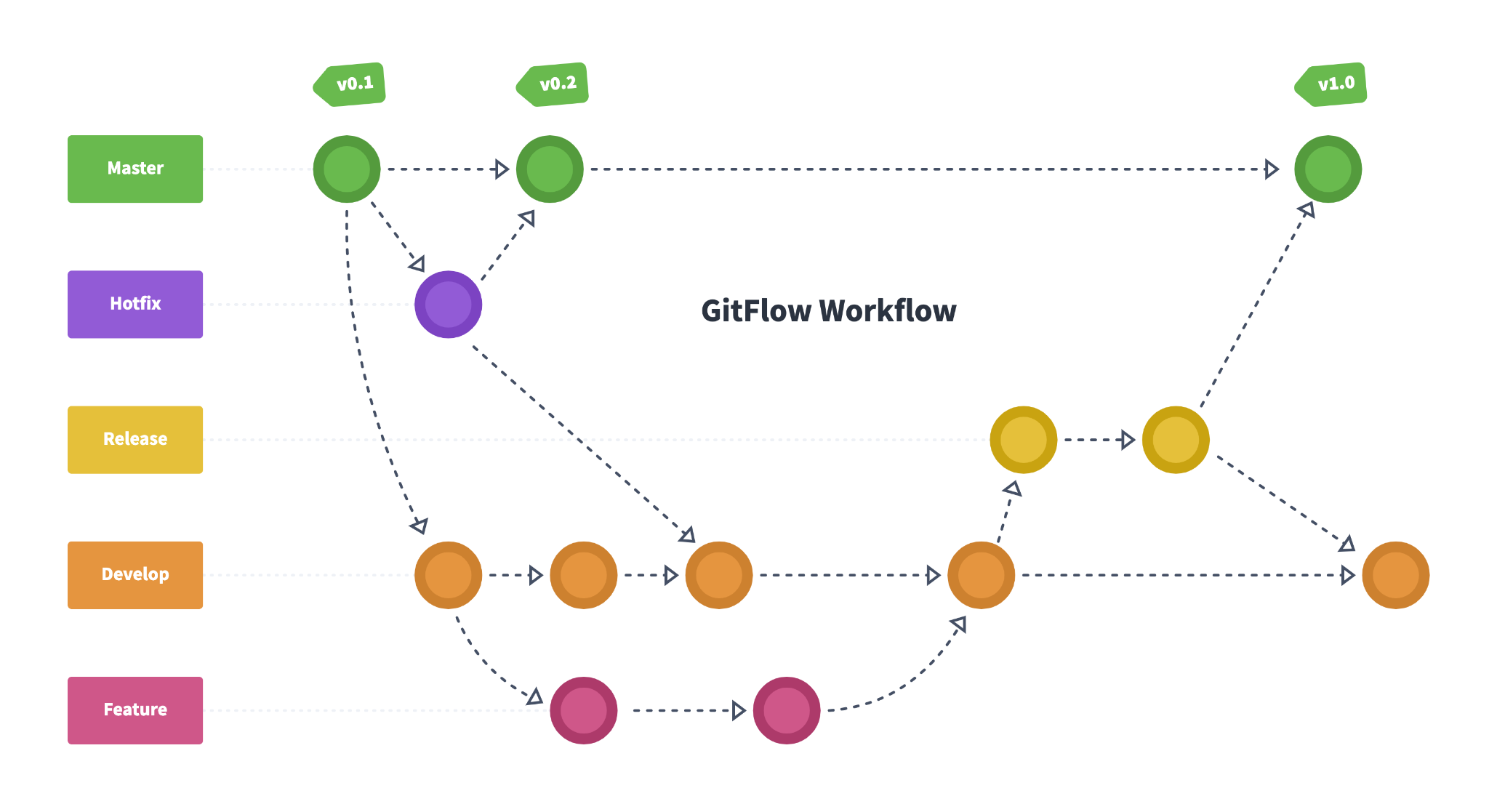
## Система контроля версий Git (обсуждаемо)

### Жизненный цикл приложения Git flow

Обычно на наших проектах используется система Git flow, при которой:

1. Каждая новая задача выполняется в отдельной ветке с названием соответствующим номеру задачи.
2. После выполнения задача отправляется в ветку develop для проверки.
3. По итогам проверки задача отправляется в ветку release.
4. Во время обновления прода изменения из ветки release отправляются в master.

Графически это можно изобразить следующим образом



### Правила работы с Git-репозиторием проекта

Для того, чтобы поддерживать такую систему необходимо соблюдать ряд несложных правил:

1. Разработчики при выполнении новой задачи действуют следующим образом:
   1. переходят в локальный release
   2. пулятся из удаленного release
   3. создают новую ветку с названием - номер задачи (как минимум)
2. После выполнения задачи разработчики:
   1. комитят свои изменения
   2. пушат в удаленную ветку с таким же названием, как и локально
   3. создают мердж-реквест (МР) в ветку develop
   4. галочку на удаление ветки при этом надо снять
3. Если при слиянии с develop образуется конфликт, действуем следующим образом:
   1. из ветки с задачей создается новая ветка с названием, например, 111111\_conflict
   2. пулятся изменения с develop
   3. решается конфликт
   4. комитятся и пушатся изменения в удаленную ветку с таким же названием
   5. создается новый МР в девелоп
   6. старый МР закрывается
   7. галочку на удаление ветки с постфиксом conflict следует оставлять выбранной
4. Далее этот МР проверяет и принимает ведущий разработчик.
5. По окончании сборки проверяются изменения разработчиком и аналитиком/менеджером на сайте.
6. Если проверка прошла с ошибками, то возвращаемся к пункту 2.
7. если проверка была успешной, задача переносится в релиз из ветки с номером задачи (**не из ветки с постфиксом conflict**):
   1. создается МР в ветку release
   2. МР принимается ведущим разработчиком
   3. По окончании сборки изменения проверяются на релизе
8. При возникновении конфликта действуем аналогично пункту 3, только develop меняем на release.

**ВАЖНО**: нельзя пулиться из ветки develop в текущие ветки с номером задачи! Если вы это сделаете, а потом отправите в release, то все непроверенные задачи уйдут в релиз, а потом на прод и никто этого не узнает.

Примечание: на некоторых проектах названия веток могут отличаться, а также где-то могут быть свои особенности по слиянию кода. Эту и другую полезную информацию вам расскажет ведущий разработчик проекта.