Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

ПО МДК 04.02 «Обеспечение качества функционирования компьютерных систем»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-3-20  Ощепкова Анна Владимировна | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_ К.С. Образцова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года |

Москва 2023

# Практическая работа №1

## «Работа с Git»

Цель работы: Продемонстрировать конспект двух лекций. Продемонстрировать конфигурацию Git. Создать локальный репозиторий. Продемонстрировать работу основных команд для работы с файлами в репозитории. Создать несколько различных веток. Описать работу с указателями HEAD

Ход работы:

1. Конспект двух лекций

Waterfall

- Каскадные модели разработки

- Гибкие модели разработки

- SCRUM

* В конце спринта имеется готовый продукт
* Гибкое изменение требований
* Быстрое реагирование на изменение трендов
* Бюджет не фиксирован

Основные этапы:

* Анализ
* Разработка
* Сборка
* Тестирование
* Развертывание
* Релиз
* Сопровождение
* Мониторинг

Понятие DevOps

**DevOps:**

Бесперебойная поставка ПО с помощью непрерывной интеграции рабочих процессов

Снижение ошибок, связанных с человеческим фактором

Ускорение этапов разработки

Развитие DevOps - DevSecOps

Интегрирует безопасность в процессы разработки и автоматизации

Инфраструктура как код

Методология, которая предлагает использовать архитектурный подход к управлению инфраструктурными ресурсами, а именно автоматизировать создание и управление инфраструктурой компонентами с помощью программного кода

Основная идея заключается в том, что вся инфраструктура, такая как виртуальные машины, сетевые настройки, хранилища данных и другие ресурсы, описывается в виде кода, который может быть управляем исходными файлами, хранящимися в системе контроля версий.

CI/CD

Методология разработки программного обеспечения, целью которой является автоматизация и ускорение процесса разработки, тестирования и доставки программного продукта

Цель CI/CD состоит в том, чтобы обеспечить быстрое время от идеи до развертывания новых функций или исправлений ошибок, с учетом меньшей вероятности ошибок и конфликтов, а также поддержания высокого уровня качества кода и процесса разработки.

Состав

* Непрерывная интеграция(continuous Integration)
* Непрерывная доставка(Continuous Delivery)
* Непрерывное развертывание(Continuous Deployment)

Преимущества внедрения DevOps в разработке

1. Ускорение процесса разработки.

DevOps позволяет автоматизировать и упростить различные этапы разработки, такие как сборка, тестирование и развертывание. Это позволяет сократить время, затрачиваемое на каждый из этих этапов, значительно увеличить производительность команды разработки.

2. Улучшение качества ПО

DevOps способствует автоматизации тестирования и контроля качества, что позволяет быстрее выявлять и устранять ошибки и повышать стабильность и надежность разрабатываемого продукта.

3. Снижение рисков.

DevOps позволяет осуществлять постоянный мониторинг и контроль процесса разработки. Это позволяет оперативно реагировать на возникающие проблемы и снижает риски, связанные с качеством, безопасностью и доступностью приложения.

4. Более быстрая доставка нового функционала.

DevOps способствует автоматизации процесса развертывания и доставки нового функционала пользователям. Благодаря этому, команда разработки может выпускать обновления и новые версии приложения значительно быстрее и с меньшими затратами.

5. Более гибкий и масштабируемый процесс разработки.

DevOps позволяет легко внедрять изменения и масштабировать процесс разработки в соответствии с потребностями пользователей или рынка. Это позволяет более гибко реагировать на изменения условий и быстро адаптироваться к новым требованиям.

6. Снижение затрат на разработку и эксплуатацию.

DevOps позволяет оптимизировать процессы разработки и эксплуатации, что сокращает затраты на труд и ресурсы. Это позволяет сделать разработку ПО более эффективной и экономичной.

7. Более высокая удовлетворенность пользователей.

Благодаря улучшенному качеству ПО, более быстрой доставке нового функционала и более оперативной поддержке, пользователи получают более удовлетворительный опыт использования приложения. Это повышает удовлетворенность пользователей и способствует удержанию и привлечению новых клиентов.

Система контроля версий

Система контроля версий – это специальное программное обеспечение, которое используется для управления изменениями в файловой системе, отслеживания и контроля версий документов или кода программы. Она позволяет разработчикам в команде проекта работать над файлами вместе, отслеживать изменения, управлять их версиями и восстанавливаться к предыдущим состояниям проекта.

Преимущества:

* + История изменений
  + Контроль версий
  + Коллективная работа
  + Ветвление и слияние
  + Отслеживание ошибок

Распределенная vs централизованная

Архитектура:

Централизованная система контроля версий имеет длинную центральную базу данных, где хранятся все файлы и изменения. В распределенной системе каждый пользователь имеет свою копию репозитория, включая историю изменений.

Работа в офлайн-режиме:

Распределенная система позволяет работать в оффлайн режиме, так как каждый пользователь имеет полную копию репозитория. В централизованной системе операции происходят через удаленное соединение, что может быть медленнее при большом количестве пользователей.

Управление правами происходит на уровне сервера

Snapshod-based

Система обхода версий с механизмом снимков (например, Git, Mercurial) хранят все версии файлов и директорий в форме полных копий (снимков) состояния проекта в определенный момент времени. Как вы делает коммит, система фиксирует все измененные файлы и создает новый снимок, который включает в себя полные копии всех измененных файлов. Это позволяет быстро переключиться между разными версиями проекта и осуществлять обход старых коммитов.

Delta-based

Система контроля версий со списком изменений (например Subversion) хранят только разницу (delta) между последующими версиями фалов. Когда вы делаете коммит, система созраняет только изменения, сделанные в файлах, относительно предыдущего коммита, в виде списка изменений. На самом деле система часто хранит несколько версий файла, чтобы быстрее обрабатывать запросы, основанные на списке изменений. При переключении на предыдущие версии проекта, СКВ применяет все изменения последовательно, чтобы восстановить запрошенную версию проекта.

Система контроля версий

* + Git
  + Perforce
  + Mercurial
  + Subversion

1. Продемонстрировать конфигурацию Git

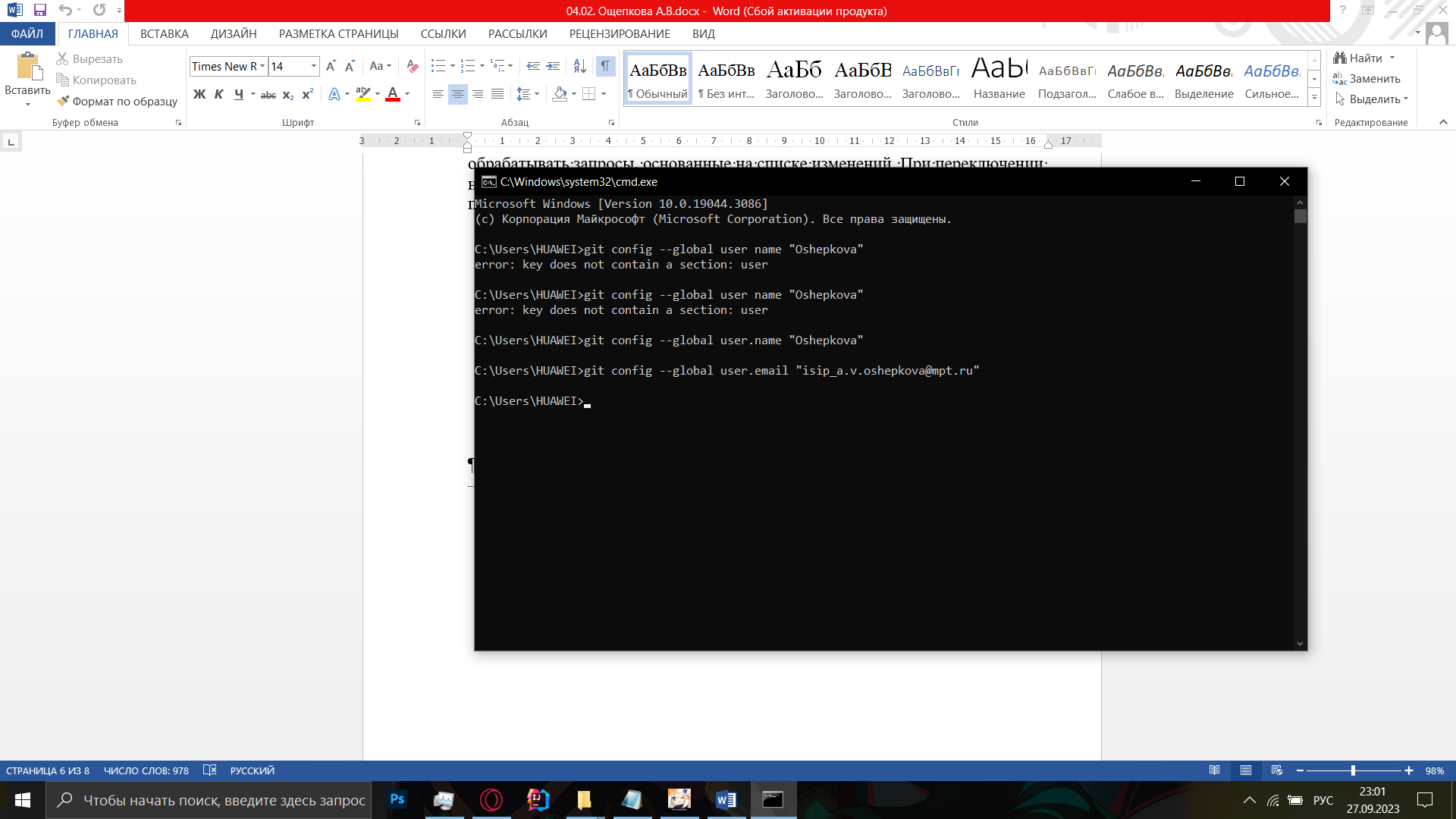


Рисунок . Конфигурация

1. Создание локального репозитория
2. Продемонстрировать работу основных команд для работы с файлами в репозитории
3. Создание нескольких различных веток
4. Описать работу с указателями HEAD

Вывод:

# Практическая работа №2

## «Работа Git-хостингами»

Цель работы: продемонстрировать Конспект лекции. Описать создание аккаунта на любом Git-хостинге. Кратко описать функционал различных вкладок. Продемонстрировать работу с удаленным репозиторием. Основные команды для работы с удаленным репозиторием ( + обязательно сделать ограничение на определенную ветку и попробовать залить сразу несколько веток, описать результат)

Ход работы:

1. Конспект лекции

Git-хостинги - это платформы, которые предоставляют инфраструктуру для хранения и управления репозиториями Git. Они позволяют разработчикам работать с Git репозиториями удаленно, управлять версиями, отслеживать изменения и сотрудничать с другими разработчиками.

Git-хостинги:

* GitHub
* GitLab

GitLab - это хостинг-платформа для хранения и управления кодом. Она предоставляет функциональность, аналогичную GitHub, но с открытым исходным кодом. GitLab может быть развернут на собственных серверах, что позволяет организациям иметь полный контроль над своими репозиториями. Он также предлагает широкий набор инструментов для управления проектами.

1. Описать создание аккаунта на любом Git-хостинге. Кратко описать функционал различных вкладок.
2. Работа с удаленным репозиторием

Вывод: