

Министерство образования Республики Беларусь
Учреждение образования

«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет Компьютерного Проектирования
Кафедра инженерной психологии и эргономики

ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ ИНФОРМАЦИОННЫХ СИСТЕМ
ОТЧЁТ
по практической работе №1

Студент гр.110901

Самусевич А.Г.

Проверил

Давыдович К.И.

Минск 2023

1. Система контроля версий (СКВ) – это инструмент, который используется разработчиками для управления изменениями в исходном коде и других ресурсах проекта. Он позволяет отслеживать историю изменений, возвращаться к предыдущим версиям, объединять изменения, работать в команде и многое другое. СКВ делает процесс разработки более управляемым и безопасным.

2. Локальные СКВ – это системы, которые управляют версиями файлов и историей изменений только на одной локальной машине. Примером локальной СКВ является «RCS» (Revision Control System). Они просты в использовании, но не подходят для совместной разработки.

3. Централизованные СКВ – это системы, в которых существует центральный сервер, на котором хранится вся история исходного кода и куда разработчики отправляют свои изменения. Примером централизованных СКВ является «Subversion» (SVN). Они подходят для совместной разработки, но могут иметь единую точку отказа.

4. Распределенные СКВ – это системы, в которых каждый разработчик имеет полную копию репозитория, включая всю историю исходного кода. Примером распределенных СКВ является «Git». Они обеспечивают более гибкую и надежную работу в команде, так как каждый разработчик имеет собственную копию репозитория.

5. Git – это распределенная система контроля версий, разработанная Линусом Торвальдсом. Основные отличия Git от других СКВ:

- каждый разработчик имеет полную копию репозитория, что делает работу в команде более гибкой и надежной;
- Git работает быстро, даже с большими репозиториями;
- создание, слияние и управление ветками легко и эффективно;
- история изменений хранится локально, что позволяет работать оффлайн;
- все данные в Git хранятся целостно и защищены от повреждений.

Теперь рассмотрим команды Git:

1. Git init: команда `git init` создает новый репозиторий Git в текущем каталоге, начиная с текущего каталога.

2. Git add: команда `git add` добавляет файлы в индекс, подготавливая их к фиксации (`commit`).

3. Git commit: команда `git commit` фиксирует изменения, добавленные в индекс, и создает новую версию репозитория.

4. Git status: команда `git status` отображает текущий статус рабочего каталога и индекса.

5. Git reset: команда `git reset` используется для отмены изменений в рабочем каталоге и индексе, позволяя вернуться к предыдущему состоянию.

6. Git log: команда `git log` выводит историю коммитов в репозитории, отображая авторов, даты и сообщения коммитов.

7. Git branch: команда `git branch` используется для создания, просмотра и управления ветками в репозитории.