**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет управления

Кафедра Информатики и информационных технологий

Специальность: 09.03.03 Прикладная информатика

Корпоративные информационные системы управления

**Отчет по практической работе №1**

|  |  |
| --- | --- |
| по дисциплине: | Программная инженерия |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **АВТОР** |
|  |  | Обучающийся 2 курса группы Ик-722 |
|  |  | заочной формы обучения |
|  |  |  |
|  |  | Красильников И.В. |
|  |  |  |
|  |  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Нижний Новгород, 2025г.

**Цель работы:**

Освоить принципы и инструменты распределённого контроля версий на примере Git, а также платформы удалённого хостинга Git‑репозиториев (GitHub, GitLab, GitFlick): изучить их назначение, возможности и отличия; выработать навыки работы с ключевыми командами Git; зарегистрироваться на одном из сервисов и применить полученные знания, ведя отчёты по лабораторным работам в собственном репозитории.

**Инструменты:**

* Git CLI (командная строка) или GUI‑клиент (GitKraken, Sourcetree, GitHub Desktop).
* Учётная запись на GitHub или GitLab (Web‑интерфейс).
* Текстовый редактор/IDE с поддержкой Git‑интеграции (VS Code, IntelliJ IDEA, WebStorm).

**Документация:**

* Официальный справочник Git (git-scm.com).
* Документация GitHub (docs.github.com) и GitLab (docs.gitlab.com).
* Руководство по российской платформе GitFlick (официальный сайт или внутренний портал).

**Исходные данные:**

Набор лабораторных работ (исходные файлы / задания), которые будут храниться в репозитории.

Доступ к интернету для регистрации и работы с удалёнными репозиториями.

**Назначение, возможности, особенности**

**Git**

**Назначение:** распределённая система контроля версий (DVCS), позволяющая отслеживать историю изменений файлов и работать в команде без постоянного доступа к центральному серверу.

**Возможности:**

* локальные коммиты и ветвление;
* быстрые операции с ветками (создание, переключение, слияние);
* сохранение полного журнала изменений;
* поддержка ремоут‑репозиториев (SSH, HTTPS).

**Особенности:**

* каждый клон — полноценная копия всей истории;
* продуманная модель ветвления и слияния;
* оптимизирован для скорости и небольшого размера репозиториев.

**GitHub**

**Назначение:** облачный хостинг Git‑репозиториев с веб‑интерфейсом и социальными функциями.

**Возможности:**

* pull requests (запросы на слияние) и code review;
* встроенный issue‑tracker и Wiki;
* GitHub Actions для CI/CD;
* Pages для простого деплоя статических сайтов;
* социальная сеть (звёздочки, форки, профили разработчиков).

**Особенности:**

* бесплатные публичные и приватные репозитории (с ограничениями по команде);
* интеграции со сторонними сервисами (Slack, Jira, Travis CI).

**GitLab**

**Назначение:** платформа DevOps «всё в одном» с встроенным Git‑хостингом и инструментами CI/CD.

**Возможности:**

* встроенный Git‑репозиторий + CI/CD‑конвейеры (GitLab CI);
* управление проектами (issues, milestones, boards);
* контейнерный реестр (Container Registry);
* мониторинг и управление инцидентами (DevSecOps‑функции).

**Особенности:**

* доступен как SaaS (gitlab.com), так и для самхостинга (Community & Enterprise Edition);
* гибкая система прав доступа и групп;
* поддержка Auto DevOps и встроенного Kubernetes.

**Сравнение: Git vs GitHub vs GitLab**

| **Характеристика** | **Git** | **GitHub** | **GitLab** |
| --- | --- | --- | --- |
| Тип | CLI‑программа (DVCS) | Облачный хостинг репозиториев | Платформа DevOps (SaaS и self‑hosted) |
| Хранение репозиториев | Локально + любые удалённые серверы | На серверах GitHub | На серверах GitLab или собственных серверах |
| Ветвление и слияние | Очень быстрое, на уровне локали | Через pull requests, code review | Через merge requests, code review |
| CI/CD | Нет (нужны сторонние инструменты) | GitHub Actions (встроенно) | GitLab CI (полноценный встроенный конвейер) |
| Управление проектами | Отсутствует | Issues, Projects, Wiki | Issues, Milestones, Boards, Roadmaps |
| Самохостинг | Да | Нет (только GitHub Enterprise) | Да (Community и Enterprise Edition) |
| Безопасность и DevSecOps | Зависит от внешних плагинов | Dependabot, CodeQL | Встроенное сканирование безопасности |
| Бесплатный уровень | полностью бесплатен | публичные и маленькие приватные проекты | большинство функций в бесплатной CE |

**Российская система GitFlick**

**Назначение:** отечественная альтернатива GitHub/GitLab для корпоративных и образовательных нужд с учётом требований российского законодательства.

**Возможности и особенности:**

* хранение данных на серверах внутри России (соответствие ФЗ‑152 «О персональных данных»);
* базовый функционал Git‑хостинга: репозитории, ветки, MR/PR, issue‑трекер;
* интеграция с отечественными CI‑серверами (например, TeamCity, Jenkins с локальными инстансами);
* поддержка SSO (LDAP/Active Directory), двухфакторной аутентификации;
* российская локализация интерфейса и техподдержка.

**Основные команды Git: названия и назначение**

| **Команда** | **Описание** |
| --- | --- |
| git init | Инициализация нового локального репозитория |
| git clone <url> | Клонирование удалённого репозитория на локальную машину |
| git status | Показ статуса файлов (изменённые, добавленные, удалённые) |
| git add <файл/путь> | Добавление изменений в индекс (staging area) |
| git commit -m "сообщение" | Фиксация (сохранение) изменений из индекса в локальную историю |
| git log | Просмотр истории коммитов |
| git diff | Показ различий между рабочей копией и индексом или между коммитами |
| git branch | Список веток; git branch <имя> — создание новой ветки |
| git checkout <ветка/файл> | Переключение ветки или восстановление файла из другого коммита |
| git merge <ветка> | Слияние указанной ветки в текущую |
| git pull | git fetch + git merge: получить изменения из удалённого репозитория и слить |
| git push | Отправка локальных коммитов в удалённый репозиторий |
| git fetch | Получение изменений из удалённого репозитория без автоматического слияния |
| git remote add <имя> <url> | Добавление ссылки на удалённый репозиторий под заданным именем |
| git tag <метка> | Создание «метки» (релиза) на текущем коммите |
| git revert <коммит> | Создание нового коммита, отменяющего указанный |
| `git reset [--soft | --hard]` |
| git stash | Временное «спрятать» незакоммиченные изменения, чтобы очистить рабочую область |

**Регистрация на GitHub**



