Семинар №1

Локальный репозиторий git

Общие сведения

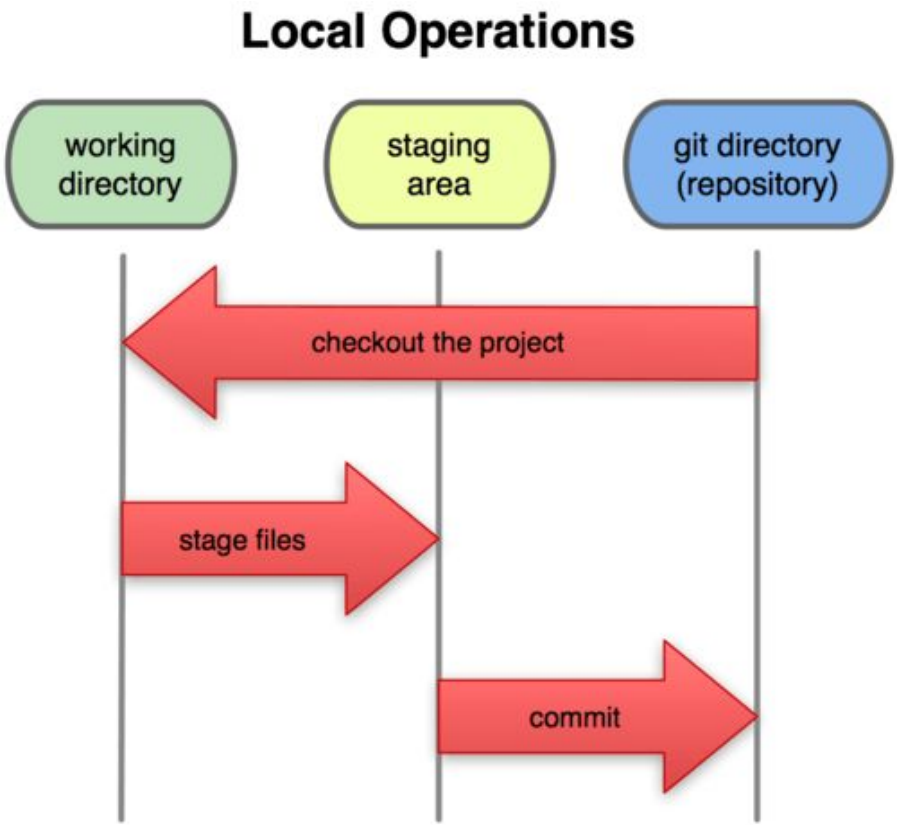
- Git (гит) - распределенная система управления версиями (файлов, в первую очередь текстовых).
- Состоит из набора программ, для работы с ними нужен только терминал.
- Не использует никаких баз данных (в классической интерпретации), вся информация о репозитории хранится в файлах в директории .git
- В гите повсеместно используются хеши вычисленные по алгоритму SHA-1. Они вычисляются для самих данных, для комитов и т.д. Пример: `24b9da6552252987aa493b52f8696cd6d3b00373`
- Любое действие можно выполнить в терминале, но есть и множество графических оболочек (TortoiseGit, Git Extensions, …)

Состояния репозитория

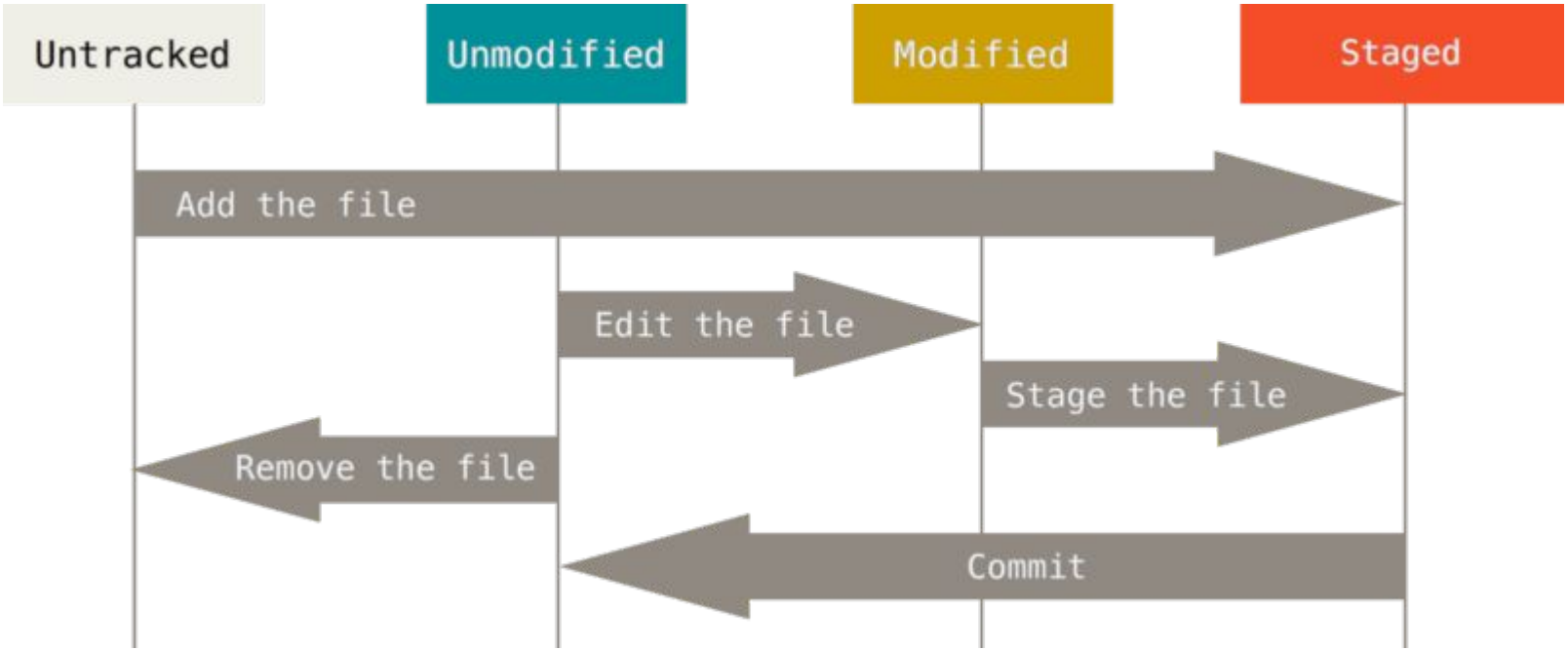
.git каталог - история всех файлов в репе + метаданные, самая важная часть

Working directory - текущая редактируемая версия репозитория, это все файлы в каталоге репы кроме .git

Staging area - файлы помеченные для фиксации изменения в виде комита



Жизненный цикл файлов в репозитории



Создание репозитория

Команда: **git init**

Создает в текущей директории .git директорию с метаданными.

После создания репы надо ее настроить через **git config**:

- Просмотр текущих настроек: **git config -l**
- Установить значение для параметра “user.email”: **git config user.email a@b.cd**
- Если значение с пробелами, то надо брать в двойные кавычки: **git config user.name “Vasily Kornienko”**

Просмотр состояния репозитория

Команда: **git status**

Показывает какие файлы в каких состояниях находятся

Показывает подсказки, что можно сделать с каждым из файлов

Добавление файлов

Команда: **git add**

Добавление происходит в два этапа:

- Сначала помечаем файл/директорию для фиксации (добавляем в staging area): **git add lab01.c** или **git add lab01** (где lab01 - директория с файлами). Помечаем все файлы и директории для фиксации: **git add .**
- Фиксируем изменения (сохраняем текущую версию файлов в репозитории): **git commit “soobshenie luche pisat latinitsey or in English”** Сообщение должно лаконично описывать суть изменений в данной версии файлов.

.gitignore

Обычно не все файлы кладутся под версионный контроль.

Первичные файлы держим в репе (исходные коды).

Вторичные файлы не кладем в репу (результаты компиляции).

Все что надо исключить из репы описываем в файле **.gitignore** (НЕ .gitignore.txt!!!)

```
# исключаем файл по расширению
*.exe
*.pyc
# исключаем по имени директории
**/__pycache__/**
```

Редактирование файлов

Файлы которые находятся под версионным контролем (уже были добавлены в репу ранее) можно изменить.

- После изменения файл переходит в состояние changed (working directory)
- Помечаем файл для фиксации: **git add file.ext** (файл перешел в состояние staged)
- Фиксируем изменения в репозитории: **git commit “message”** (файл перешел в состояние Unmodified).

Удаление файлов

Команда: **git rm**

Удаляем файл из репозитория и из файловой системы: **git rm file.ext**

Не забываем закоммитить.

Удаляем файл только из репозитория: **git rm --cached file.ext**

Не забываем закоммитить.

Может понадобиться если забыли добавить файлы в .gitignore

Просмотр внесенных изменений

Команда: **git diff**

Показывает разницу между файлами в состоянии changed и unmodified

Если надо посмотреть разницу между staged и unmodified: **git diff --cached**

Разница между коммитами: **git diff commithash1 commithash2**

Разница между версиями одного файла в разным комитах:
git diff commithash1 commithash2 file.ext

Просмотр изменений в конкретном коммите: **git show commithash**

Просмотр истории коммитов

Команда: **git log**

Список коммитов в обратном хронологическом порядке: **git log**

Поиск коммита по фразе в сообщении коммита (case sensitive):

git log --grep=”soobshenie luche pisat”

Просмотр истории коммитов затрагивающих заданный файл:

git log file.ext

Откатываем изменения

Убираем ненужный файл из подготовленных к фиксации (переносим из состояния staged в modified: **git reset file.ext**

Отменяем изменения для файла (из состояния staged или modified в состояние unmodified): **git checkout file.ext**

Отказываем изменения заданного коммита: **git revert commithash**
Сразу создается еще один коммит! Комментарий вводится в редакторе vi(m)!
Для выхода из режима редактирования: Esc => :wq