Работа над ошибками в коммитах

Оформление сообщений к коммитам

Оформление сообщений к коммитам в Git может подчиняться определённым правилам, которые могут быть продиктованы культурой команды или техническими ограничениями.

В выводе команды git log --oneline умещается максимум 72 первых символа сообщения, поэтому многие правила включают пункт: «Сообщение не должно быть длиннее 72 символов».

Все люди разные и у всех есть предпочтения - в том числе, как формулировать сообщения коммитов.

Без единообразия коммитов нет и эффективной работы в Git.

В корпоративном стиле в начале сообщения обычно указывают Jira-ID, а после - текст сообщения.

$ git commit -m "LGS-239: Дополнить список пасхалок новыми числами"

Conventional Commits предлагает такой формат коммита:<type>: <сообщение>.

Для сообщений на русском языке часто рекомендуют использовать инфинитивы, а для сообщений на английском рекомендуется использовать повелительное наклонение. Например:

Добавить тесты для PipkaService,

Исправить ошибку #123.

Use library mega\_lib\_300,

Fix exit button и так далее.

Как исправить коммит

Иногда в только что выполненном коммите нужно что-то поменять: например, добавить ещё пару файлов или заменить сообщение на более информативное.

В таком случае можно внести правки в уже сделанный коммит с помощью опции --amend (от англ. *amend —* «исправить», «дополнить») у команды commit: git commit --amend. Опция --amend работает только с последним коммитом (HEAD). Для исправления более ранних коммитов есть другие команды.

**Дополнить коммит новыми файлами — git commit --amend --no-edit**

Дополните последний коммит забытым файлом common.css с помощью опции --amend.

$ git add забытый\_файл

*# добавили файл в список на коммит*

*# но вместо команды commit -m '...'*

*# будет:*

$ git commit --amend --no-edit

$ git log --oneline

*# коммит в истории (но у него новый хеш)*

С опцией --amend команда commit не создаст новый коммит, а дополнит последний, просто добавив в него забытый\_файл. При этом хеш последнего коммита изменится.

Обратите внимание на опцию --no-edit. Она сообщает команде commit, что сообщение коммита нужно оставить как было.

Точно так же можно добавить не новый файл, а дополнительные изменения в уже добавленном в коммит файле.

$ git add измененный\_файл *# добавили в список на коммит*

$ git commit --amend --no-edit

**Что же выбрать: новый коммит или --amend?**

В нашем примере вместо изменения последнего коммита можно было также выполнить новый коммит с забытым либо измененным файлом. Кажется, что так проще, но добавить изменения в уже существующий коммит может быть правильнее.

Например, через месяц кто-то захочет просмотреть историю изменений. Намного проще понять, что изменилось, если оба файла находятся в одном коммите. Иначе коммит со второй порцией изменений придётся искать.

**Изменить сообщение коммита — git commit --amend -m "Новое сообщение"**

Может быть и так, что добавлять новые файлы в коммит не нужно, зато понадобилось изменить сообщение.

Допустим, хочется заменить сообщение. Сделать это можно через commit --amend с флагом -m.

$ git commit --amend -m "Новое сообщение"

$ git log --oneline

Хеш коммита снова поменялся, потому что изменились сообщение и время коммита. При этом файлы в коммите остались те же.