**Шпаргалка. Командная работа в Git**

Модуль подошёл к концу, а это значит, что пришло время подвести итоги и запастись шпаргалкой!

**Feature branch workflow**

Самый популярный подход к работе с Git в команде — это feature branch workflow. В нём создают ветку для каждой задачи (например, для новой функциональности или исправления бага), а когда всё готово, вливают новую ветку в main.

Важные этапы этого процесса — пул-реквест и ревью изменений. Пул-реквест — это интерфейс, в котором можно обсудить изменения. Ревью — просмотр изменений другими участниками и один из способов проверить качество таких изменений.

Если вы уже участник проекта (или collaborator в терминах GitHub), можно клонировать репозиторий напрямую. А если нет, нужно предварительно сделать «форк». Также для участников доступна кнопка **Merge** после ревью, а для неучастников — нет.

**Конфликты слияния**

Когда один и тот же файл меняется в нескольких ветках, при их слиянии может произойти конфликт. Пугаться конфликтов не нужно, это нормальная часть работы с системами контроля версий. IDE, вроде VSCode или Intellij IDEA, помогут «склеить» файл из двух конфликтующих версий.

**Алгоритм-шпаргалка для создания PR**

1. Склонировать репозиторий.
   1. Если вы не участник проекта, предварительно сделать «форк» исходного репозитория.
   2. На странице репозитория или «форка» нажать кнопки: **Code → SSH → скопировать ссылку**.
   3. Выполнить команду git clone <ссылка на репозиторий>.
2. Создать ветку для вашей задачи: git checkout -b my-task-branch-name.
3. Добавить и «закоммитить» изменения, которые вы хотите внести в проект.
4. «Запушить» ветку: git push --set-upstream origin HEAD или git push -u origin my-task-branch-name.
   1. GitHub (с помощью Git) выведет ссылку на создание PR. По ней нужно перейти.
   2. PR можно также создать через интерфейс GitHub.
5. Сообщить о пул-реквесте ревьюеру.
   1. Иногда ревьюеры назначаются автоматически, тогда сообщать не нужно.
6. Обсуждать с ревьюером предлагаемые изменения и вносить правки, пока эти изменения не будут одобрены (пока не будет получен «апрув»).

6.1. Если кто-то добавил конфликтующие изменения в main, пока ваш PR был на ревью, нужно разрешить конфликт:

* 1. Обновить main: git checkout main && git pull.
  2. Влить main в свою ветку: git checkout my-task-branch-name && git merge main.
  3. Разрешить конфликты слияния с помощью IDE или вручную.
  4. Создать коммит слияния: git commit --no-edit или git commit -m 'merge main'.
  5. Сделать git push своей ветки.

1. Нажать кнопку **Merge** или подождать, пока её нажмёт кто-то ещё.
2. Ещё раз обновить main, чтобы «подтянуть» ваши изменения в основную ветку локального репозитория: git checkout main && git pull.
3. Вы великолепны! Можете начинать снова со второго пункта.

**Алгоритм-шпаргалка для разрешения конфликтов слияния**

1. Открыть проект в IDE (VS Code, IDEA или другие).
2. Открыть файл, в котором есть конфликт.
3. Выбрать, какие части файла нужно взять из одной ветки, а какие — из другой.
4. Когда конфликты разрешены, сделать коммит: git commit --no-edit или git commit -m 'merge branch <название ветки>'.