

# Ревью кода

Когда читаешь умные слова других,  
в голову приходят собственные умные мысли.

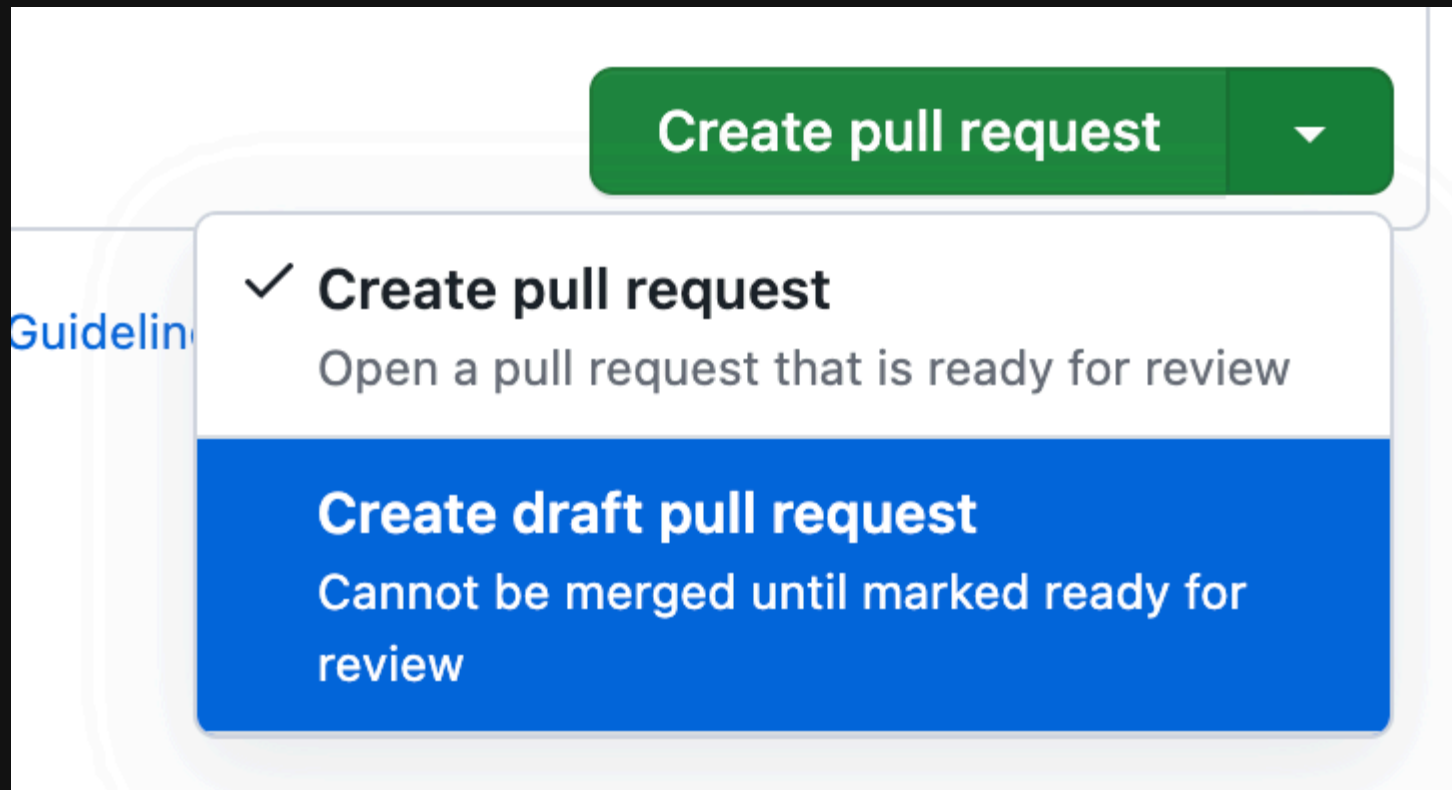
© М. Лашков

# Введение

- Запросы на включение изменений (PR) — важный инструмент для совместной работы и улучшения качества кода.
- PR — это не только технический инструмент, но и средство коммуникации между разработчиками.
- Руководство поможет создавать лучшие PR и проводить эффективные рецензии кода.

Создание PR до готовности кода

Создавайте PR на ранних этапах, даже если код еще не ГОТОВ.



Просматривайте изменения вне IDE

# Документируйте прогресс

## Выполнено

- ☒ Создать UI-компонент для функции поиска
- ☒ Написать unit-тесты для компонента

## Предстоит сделать

- ☐ Добавить компонент в шапку приложения
- ☐ Обновить end-to-end тесты

## Проблемы

- До сих пор не понятно, почему end-to-end тесты иногда падают на шапке приложения

# Создание PR до готовности кода

- Создавайте PR на ранних этапах, даже если код еще не готов.
- Используйте PR в качестве черновика, чтобы:
  - Просматривать изменения вне IDE.
  - Документировать прогресс.

Как сделать PR привлекательным для  
рецензентов



# Делайте небольшие PR

Хотя подходы могут различаться в разных командах, ваш Pull Request (PR) должен быть как можно меньше, но при этом соответствовать следующим критериям:

## 1. Не ломать приложение

Это может казаться очевидным, но ваш PR не должен ничего ломать, даже если проблема будет исправлена в ближайшее время или кем-то другим.

## 2. Включать unit-тесты

- Тесты должны быть частью этого PR, а не выноситься в отдельный
- Необходимость других типов тестов (интеграционных, end-to-end и т.д.) обсуждается отдельно

## 3. Приносить пользу

Требования к полезности PR могут варьироваться:

- В некоторых командах достаточно создать компонент, даже если он пока не используется
- В других - каждый PR должен приносить пользу конечному пользователю (где это возможно)

# Предоставляйте всю необходимую информацию

Название: Добавление функции поиска в шапку приложения

Описание:

Задача в Jira: 1234

Дизайн в Figma: (ссылка)

Добавляет новый раздел справа в шапке приложения (видимый только для авторизованных пользователей), содержащий форму поиска

(скриншот или видео)

Шаги для проверки:

1. Авторизуйтесь тестовым пользователем
2. Введите запрос в строку поиска в правом верхнем углу, например "тест"
3. Нажмите Enter или кнопку поиска
4. Должен произойти переход на страницу поиска (пока пустую), с введённым запросом в URL (параметр query)

Следующие шаги:

- Реализовать страницу поиска (задача Jira 1235)
- Обновить end-to-end тесты (задача Jira 1236) в репозитории e2e-tests

# Как другие выбирают PR для ревью?

## **\*\*Ситуация:\*\***

- Нужно выбрать 1 из 2 PR для ревью
- Оба имеют четкое описание

## **\*\*Alice:\*\***

- Чистый код
- Быстро вносит правки
- 2 PR на прошлой неделе: ревью заняло 15 минут

## **\*\*Bob:\*\***

- В прошлый раз:
  - Ошибка при запуске
  - Несоответствие Figma
  - 12+ комментариев
  - Проваленные e2e-тесты
  - Откат PR

## **\*\*Вывод:\*\***

Разработчики интуитивно выбирают PR с меньшим риском и затратами времени.

## **\*\*Совет:\*\***

Стройте "индекс доверия" – будьте тем разработчиком, чьи PR ревьюят с удовольствием.

# Как сделать PR привлекательным для рецензентов

- Делайте небольшие PR.
- Предоставляйте всю необходимую информацию:
  - Ссылки на задачи.
  - Описания изменений.
  - Шаги для тестирования.
- Заслужите доверие, создавая качественный код и оперативно реагируя на отзывы.

Будьте первым рецензентом своего PR

# Примерьте на себя роль рецензента

## Проблема:

Легко писать код, забывая что его кто-то будет читать.

То, что легко пишется, не всегда легко читается.

## Как делать:

1. Откройте свой PR в GitHub
2. Проверяйте каждое изменение так, будто это чужой код
3. Обратите внимание на:
  - Подсветку кода (она отличается от IDE)
  - Только изменённые строки (а не весь файл)

## Результат:

Вы увидите свой код под другим углом, с новой перспективы.

# Предвосхищайте возможные комментарии и вопросы

Основной принцип:

Заранее продумайте возможные вопросы и замечания ревьюеров. Либо:

- Обоснуйте свои решения в коде/описании PR
- Либо улучшите код до ревью

Пример:

Вы использовали `for` вместо принятого в проекте `.map( )`

Варианты действий:

1. Если это осознанный выбор (например, для производительности):

```
// Используем цикл из-за требований к производительности  
for ( ... ) { ... }
```

# Будьте первым рецензентом своего PR

- Примерьте на себя роль рецензента.
- Предвосхищайте возможные комментарии и вопросы.



# Выбор рецензентов

- Не назначайте слишком много рецензентов.
- Выбирайте рецензентов с соответствующим опытом и знаниями.
- Используйте обязательных и дополнительных рецензентов.

# Работа с комментариями

# Будьте открыты для обратной связи или правильный настрой при получении ревью

Основные принципы:

- Вы просите *оценку кода*, а не одобрение
- Комментарии относятся к коду, а не к вам лично
- Даже если не согласны — благодарите за мнение

Как реагировать на комментарии:

"Спасибо за предложение! Я тоже рассматривал этот вариант, но выбрал текущее решение — мне кажется, так функция читается понятнее, хоть и немного длиннее. Как вам?"

# Работа с комментариями

- Будьте открыты для обратной связи.
- Оперативно вносите изменения.
- Запрашивайте разъяснения, если комментарий непонятен.
- Уважайте чужое мнение, даже если вы с ним не согласны.
- Всегда отвечайте на комментарии.

# Рецензирование PR других разработчиков

- Выделяйте время на рецензирование PR других разработчиков.
- Следите за ходом рецензирования и оперативно реагируйте на комментарии.

# Роль начинающего разработчика в рецензировании кода

- Внимательно изучайте код и перенимайте опыт.
- Задавайте вопросы, если что-то непонятно.
- Предлагайте альтернативные решения.

# Что проверять при рецензировании кода

- Соответствие стандартам кодирования.
- Поиск ошибок и крайних случаев.
- Оценка архитектурных решений.

# Правильный тон при комментировании

- Используйте вопросы вместо утверждений.
- Будьте позитивны и доброжелательны.
- Предлагайте конкретные варианты исправления.



# Важность комментариев

- Указывайте, какие комментарии требуют обязательного исправления.
- Предлагайте альтернативные решения.
- Делитесь своими мыслями и знаниями.
- Хвалите хорошие решения.

# Проверка перед отправкой рецензии

- Проверьте каждую строку кода.
- Оцените общую картину.
- Перечитайте свои комментарии.

# Утверждение PR

- Не затягивайте с утверждением PR после внесения всех необходимых изменений.
- Просмотрите весь PR целиком.

# Конфликты

- Продолжайте обсуждение сложных вопросов за пределами PR.
- Привлекайте третьих лиц.
- Организуйте встречу для обсуждения спорных вопросов.

# Заключение

- Рецензирование кода — это важный инструмент для обмена знаниями и улучшения качества кода.
- PR — это не просто формальность, а возможность для совместной работы и обучения.
- Стремитесь к совершенству в создании PR и рецензировании кода.

# Источники

- Pull requests and code review