Блок	Соответствует одному из 4 принципов
Чистота кода	
Есть ли в ревью избыточный или повторяющийся код?	Не оставлять лишнего
Не используются антипаттерны (Spaggeti Code, God Object и т.п.)	
Используются эффективные структуры данных и алгоритмы	
Есть предложения по улучшению проблемных участков кода	
Есть ли какие-нибудь лучшие практики, шаблоны проектирования или специфические для языка шаблоны, которые могли бы существенно улучшить этот код?	
Учтены шаблоны GRASP	
Общие (Changelist / Merge Request / Pull Request)	
Задача переведена в статус ревью и назначена на Техлида направления	
В задаче в комментарии указан MR или пояснение почему задача переведена в ревью	
MR не содержит меток Draft или WIP	
Есть ли незавершенный код? Если есть, должен ли он быть удален или помечен маркером типа «TODO»?	
MR содержит коммиты, которые относятся только к одной задаче	
MR прошел стадию pre в CI. (для frontend должен пройти build, lint, test для backend - build, unittest)	
MR не содержит конфликтов	
Инфрастуктура	
Учтено влияние перекрытия иерархий конфигурационных файлов на работу на разных стендах	
Проверить в MR версии пакетов - если настроено автоматическое обновление общих (shared) пакетов, то НЕ поднятие версии их не обновит	
Юзабилити и доступность	
Внешний вид (если есть) должен быть хорошим	
Функциональность хорошо сделана с точки зрения пользователей этого кода (user, пользователь-програм	Не усложнять
Код хорошо спроектирован	Заложи гибкость
У предлагаемого решения требуемая доступность (высокая, непрерывная, постоянная).	
API/UI интуитивен для использования	Не усложнять
При использовании RESTful API применены рекомендации по именованию	
Сложность/Читабельность	
Код не переусложнен. Код не более сложен, чем должен быть. Код легок для понимания	Не усложнять
Разработчик не оверинженирит: не нужно писать код, который может понадобиться, а может не понадоби	Не оставлять лишнего
Какие части были вам непонятны и почему?	Не усложнять
Можно ли улучшить читабельность кода, добавив методы меньшего размера?	Не усложнять
Можно ли улучшить читабельность, изменив имена функций/методов/переменных?	Не усложнять
Код находится в правильном файле/папке/пакете/модуле?	
Можно ли изменить структуру каких-либо методов, чтобы получился более интуитивный поток управлени	Не усложнять
Понятен ли поток управления?	Не усложнять
Тестирование	
Код протестирован (запускался) перед отправкой на ревью	
У кода есть тесты. Тесты покрывают код в нужной мере.	
Тесты хорошо спроектированы	Заложи гибкость
Является ли код тестируемым? Например, он НЕ должен содержать слишком много зависимостей или скрывать их, тестовые фреймворки должны иметь возможность использовать методы кода, и т. д.	Упростить зависимости
Юнит-тесты на самом деле проверяют, что код предоставляет требуемую функциональность?	
Все ли массивы проверяются на «выход за границы»?	
Может ли любой тестирующий код быть заменен с использованием существующего API?	Заложи гибкость
Достаточно ли для него автоматизированных тестов (юнит/интеграционных/системных)?	
Уже существующие тесты в достаточной степени покрывают новый код?	
Есть ли какие-либо тест-кейсы, входные данные или пограничные случаи, которые можно протестировать вдобавок?	
Если в MR присутствуют тесты, то они должны быть логичными, не повторяющимися, соответствовать правилам написания тестов.	Не оставлять лишнего
Комментарии и документирование	
Комментарии к коду понятны, полезны и необходимы. Они должны объяснять, ***почему*** так сделано, а не ***как*** это сделано.	Не усложнять
Есть ли комментарии/документация? Раскрывают ли они смысл кода?	
Все ли функции прокомментированы? (полезно комментировать приватные функции, т.к. публичные обычно соответствуют документации или согласованному контракту)	Не усложнять
Есть ли закомментированный код?	Не оставлять лишнего
Есть ли ненужные комментарии?	Не оставлять лишнего
Можно ли донести мысль лучше в каких-то комментариях?	Не усложнять
Сделает ли код понятнее большее количество комментариев?	Не усложнять
Можно ли убрать какие-нибудь комментарии, сделав сам код более читабельным?	Не оставлять лишнего
Использование и функционирование сторонних библиотек документировано?	
Все ли структуры данных и единицы измерения описаны?	
Хорошо ли задокументирован API?	Не усложнять
Стиль кода	
Код соответствует стайл гайдам. Соблюдаются требования и соглашения к написанию кода на проекте	
Соответствует ли код вашему стилю написания кода? Обычно это относится к расположению скобок, названиям переменных и функций, длинам строк, отступам, форматированию и комментариям.	
Наименования (для всего) выбраны хорошо. Дано наглядные имена	Не усложнять
Код следует стандартам оформления кода языка разработки	
Безопасность и защита данных	
Все ли входные данные проверяются (на корректный тип, длину, формат, диапазон) и кодируются?	
Обрабатываются ли ошибки при использовании сторонних утилит?	
Выходные данные проверяются и кодируются (*прим. пер.: например, от XSS*)?	
Обрабатываются ли неверные значения параметров?	
Этот код создаёт дыру в безопасности?	
Авторизация и аутентификация проводятся должным образом?	
Правильно ли обрабатываются и хранятся конфиденциальные пользовательские данные?	
Правильное ли шифрование используется?	
Раскрывает ли это изменение какие-либо секретные данные вроде ключей доступа, паролей или имён пользователей?	
Если код работает с пользовательским вводом, учитывает ли он такие уязвимости, как межсайтовый скриптинг и SQL-инъекции, проводится ли его очистка и проверка?	
Данные, полученные из внешних API или библиотек, проверяются должным образом?	
При использовании OAuth 2.0 учтены рекомендации по Scopes	
При использовании Claims учтены рекомендации	
Реализация	
Отсутствует лишний код	Не оставлять лишнего
Изменения разумны и выглядят хорошо	
Код легко расширяем	Заложи гибкость
Работает ли код? Выполняет ли он свои прямые обязанности, корректна ли логика.	
Решение можно упростить?	Не усложнять
Был ли использован фреймворк/API/библиотека/сервис, который не нужно было использовать?	Не оставлять лишнего
Есть ли фреймворк/API/библиотека/сервис, который не был использован, но мог бы улучшить решение?	
Код находится на верном уровне абстракции?	
Код достаточно разбит на модули?	Заложи гибкость
Можете ли вы придумать другое решение, которое значительно лучше в плане поддерживаемости, читабельности, производительности и безопасности?	
Существует ли похожая функциональность в кодовой базе? Если да, то почему она не была использована повторно?	Не оставлять лишнего
Можно ли избавиться от глобальных переменных или переместить их?	
Учтены все нюансы параллельного программирования (если есть). Если в MR присутствует параллельное программирование, оно должно выполняться безопасно	
Логические ошибки и баги	
Есть ли сценарий, при котором код не будет работать так, как задумано?	
Могут ли какие-то события или введённые данные сломать код?	
Есть ли какое-то необычное поведение или описание пограничных случаев?	
Отсутствуют видимые изъяны в логике и алгоритмах	
Может ли быть удалена часть кода, предназначенного для логирования или отладки?	Не оставлять лишнего
У циклов есть установленная длина и корректные условия завершения?	
Обработка ошибок и логирование	
Правильно ли обрабатываются ошибки?	
Нужно ли добавить/убрать какую-либо отладочную информацию?	
Дружелюбны ли к пользователю сообщения об ошибках?	Не усложнять
Хватает ли логируемых событий и написаны ли они так, чтобы можно было легко провести отладку?	Не усложнять
Производительность	
Повредит ли вносимое изменение производительности системы?	
Есть ли возможность улучшить производительность кода?	
Повлияет ли изменение кода негативно на производительность системы?	
Есть ли какая-нибудь возможность повысить производительность кода?	
Зависимости	
Если это изменение требует изменений вне кода, например в документации или конфигурации, были ли они сделаны?	
Может ли это изменение иметь последствия для других частей системы или обратной совместимости?	Упростить зависимости
Является ли код независимым, насколько это возможно?	Упростить зависимости
Может ли что-то в коде быть заменено библиотечными функциями/функциями из общих пакетов?	
Это изменение вносит нежелательные зависимости времени компиляции или выполнения?	Упростить зависимости
Экспертное мнение	
Стоит ли какому-нибудь специалисту, например по безопасности или юзабилити, посмотреть на код перед коммитом?	
Повлияет ли это изменение на другие команды? Стоит ли им также высказать своё мнение?	
Специфичные для языка/фремворка	
<Тут можно добавить требования, которые зависят от стека технологий на проекте>	

1 Принципы "**Заложи гибкость**": BDUF, DCP, DIP, ISP, Polymorphism

2 Принципы "**Не усложняй**": KISS (Делай проще...), Принцип Эйнштейна (Делай как можно проще), Бритва оккамы, Принцип Калашникова (Избыточная сложность - это уязвимость), Правило наименьшего удивлен

3 Принципы "**Не оставляй лишнего**": DRY (Не повторяйся), YAGNI (Тебе это не понадобится), APO (Избегайте преждевременной оптимизации),

4 Принципы "**Упрости зависимости**": Закон Деметры, Принцип наименьшего знания, SRP (зависимость от акторов), ISP, Римский принцип, Tell-Don't-Ask