

## Общее описание

В рамках трека «Coding Agents» необходимо разработать автоматизированную агентную систему, поддерживающую полный цикл разработки программного обеспечения (SDLC) внутри GitHub.

Система должна имитировать работу разработчика и ревьюера: анализировать задачи, вносить изменения в код, создавать Pull Request, запускать CI/CD, анализировать результаты проверок и принимать решение о завершении работы либо повторе цикла разработки.

Все действия должны выполняться исключительно с использованием механизмов GitHub: Issues, Pull Requests и GitHub Actions.

## Основной сценарий работы системы

1. Пользователь создаёт Issue с текстовым описанием задачи.
2. CLI Code Agent получает описание Issue, анализирует требования и:
  - вносит необходимые изменения в код репозитория,
  - фиксирует изменения,
  - создаёт Pull Request.
3. После создания или обновления Pull Request автоматически запускается CI/CD pipeline.
4. В рамках CI/CD запускается AI Reviewer Agent, который:
  - анализирует изменения кода,
  - проверяет результаты CI jobs,
  - сравнивает реализацию с требованиями Issue.
5. Результаты анализа публикуются в Pull Request в виде:
  - автоматического комментария,
  - summary проверки,
  - code review.
6. Если выявлены ошибки или несоответствия, Code Agent запускает новый цикл исправлений.
7. Процесс повторяется до получения корректного результата либо достижения лимита итераций.

## Ожидаемый результат

По итогам выполнения задания должна быть реализована система, включающая:

- GitHub Actions workflow, автоматически запускающийся при создании Issue и обновлении Pull Request.
- Code Agent, способный читать Issue, изменять код и создавать Pull Request.
- AI Reviewer Agent, выполняющий автоматический анализ кода и CI/CD.
- Поддержку нескольких итераций правок в рамках одного Issue.
- Финальный Pull Request с корректным, проверенным и рабочим решением.

# Технические требования

Решение должно быть реализовано с использованием Python версии 3.11 или выше.

Допускается использование следующих инструментов и библиотек:

- LLM: GPT-4o-mini, YandexGPT или эквивалентные модели.
- Работа с GitHub: GitPython или PyGithub.
- Инструменты качества кода: ruff, black, mypy, pytest.
- CI/CD: GitHub Actions.

Code Agent должен быть реализован в виде CLI-инструмента.

## Дополнительные требования:

1. В репозитории должен быть **Dockerfile** для сборки проекта в контейнер.
2. Все зависимости и код должны быть в репозитории  
Автоматически прорастать в контейнер
3. Весь сервис должен подниматься по команде:  
**docker-compose up -d**  
Либо должна быть инструкция как запустить сервис локально

# Архитектура решения

## Code Agent

Code Agent должен обеспечивать:

- получение и парсинг текста Issue;
- анализ требований задачи;
- генерацию и модификацию кода;
- создание Pull Request;
- повторный запуск цикла при наличии замечаний от Reviewer Agent.

## AI Reviewer Agent

AI Reviewer Agent должен:

- запускаться в рамках GitHub Actions;
- анализировать изменения в Pull Request;
- учитывать результаты CI jobs;
- сравнивать полученный результат с описанием Issue;
- публиковать результаты проверки непосредственно в Pull Request.

# План реализации (рекомендуемый)

Рекомендуемый план реализации включает:

- создание workflow skeleton и настройку триггеров Issue и Pull Request;
- сбор контекста Pull Request, включая diff, файлы и результаты CI;
- разработку промптов и генерацию отчётов;
- публикацию результатов анализа в Pull Request в виде комментариев, summary и артефактов.

## Валидация и тестирование

Решение должно быть проверено на реальном GitHub-репозитории с заранее подготовленными Issues.

В рамках валидации необходимо продемонстрировать:

- корректное выполнение задач из Issues;
- отсутствие бесконечных циклов разработки;
- корректную обработку ошибок;
- приемлемую скорость выполнения pipeline;
- полноту и корректность реализации требований;
- процент успешно выполненных задач;
- количество завершённых циклов разработки;
- базовую проверку кода на уязвимости;
- отсутствие галлюцинаций модели;
- соответствие используемых библиотек их актуальным версиям;
- разумное количество итераций правок.

## Критерии оценки

При оценке решения учитываются:

- корректность работы SDLC pipeline;
- полнота автоматизации процесса разработки;
- качество автоматического code review;
- стабильность и предсказуемость CI/CD;
- отсутствие бесконечных циклов;
- читаемость и структура кода;
- адаптация тестов в репозитории под изменения кода;
- воспроизводимость и документированность решения.

Доп-баллы за разворачивание решения в облаке:

- Вариант через бесплатный период на <https://cloud.ru>
- Вариант через бесплатный период на <https://console.yandex.cloud>

Если решение не развернуто публично, оно должно запускаться по одной команде через Docker.

## **Предоставление решения**

Участник должен предоставить:

- Ссылку на GitHub-репозиторий с реализованным решением;
- Отчет по работе агента и системы SDLC
- Работающий GitHub Actions workflow;
- Примеры Issues и Pull Requests, демонстрирующие работу системы.

Репозиторий должен быть доступен для проверки.