**Порядок управления исходным кодом**

**Фиксация (Commit)**

Каждое логическое изменение в коде оформляется отдельным коммитом с информативным сообщением.  
Следуется правилу: **один коммит — одно законченное изменение**.

**Правила фиксации:**

* Использовать короткие, понятные сообщения коммитов.
* Не включать в коммиты временные файлы, логи, результаты сборки.

**Ветвление (Branching)**

Для управления разработкой используется **модель ветвления по задачам (task-based branching)**.  
Каждая задача из списка проекта реализуется в **отдельной ветке**, что позволяет независимо разрабатывать и тестировать функциональность.

**Основные ветки проекта:**

|  |  |
| --- | --- |
| **Ветка** | **Назначение** |
| **main** | Содержит только стабильные релизные версии проекта |
| **<task number>\_<task name>** | Временные ветки под конкретные задачи |

Правила ветвления:

* Каждая ветка соответствует одной задаче из плана разработки.
* После завершения ветка объединяется (merge) в main и удаляется, чтобы не засорять историю.
* Ветки именуются коротко и понятно

**Контроль качества исходного кода**

Контроль качества кода проводится перед слиянием:

1. Проверка сборки и тестов (проект должен компилироваться и работать в соответствии с задачей).
2. Просмотр изменений в Pull Request (самостоятельное или внешнее ревью).
3. CI-проверка (GitHub Actions) — автоматически выполняет сборку и анализ при каждом push.

**Релиз:**

Релизы выполняются непосредственно из ветки main. После завершения набора задач и успешного тестирования создаётся тег версии (v1.0, v1.1 и т.д.).  
Тег фиксирует стабильное состояние проекта и используется для публикации релиза на GitHub.