ЛЕКЦИЯ 9

Реализация функционального прототипа

CSE842 Capstone Project
ОП 6B06102 Computer Science



План лекции

1. Прототип
2. Скелет
3. Репозиторий
4. Код
5. Хранилище
6. Качество
7. Git
8. Запуск

Виды прототипов



Одноразовый

- проверить идею
- эксперимент недели 6
- код выбрасывается

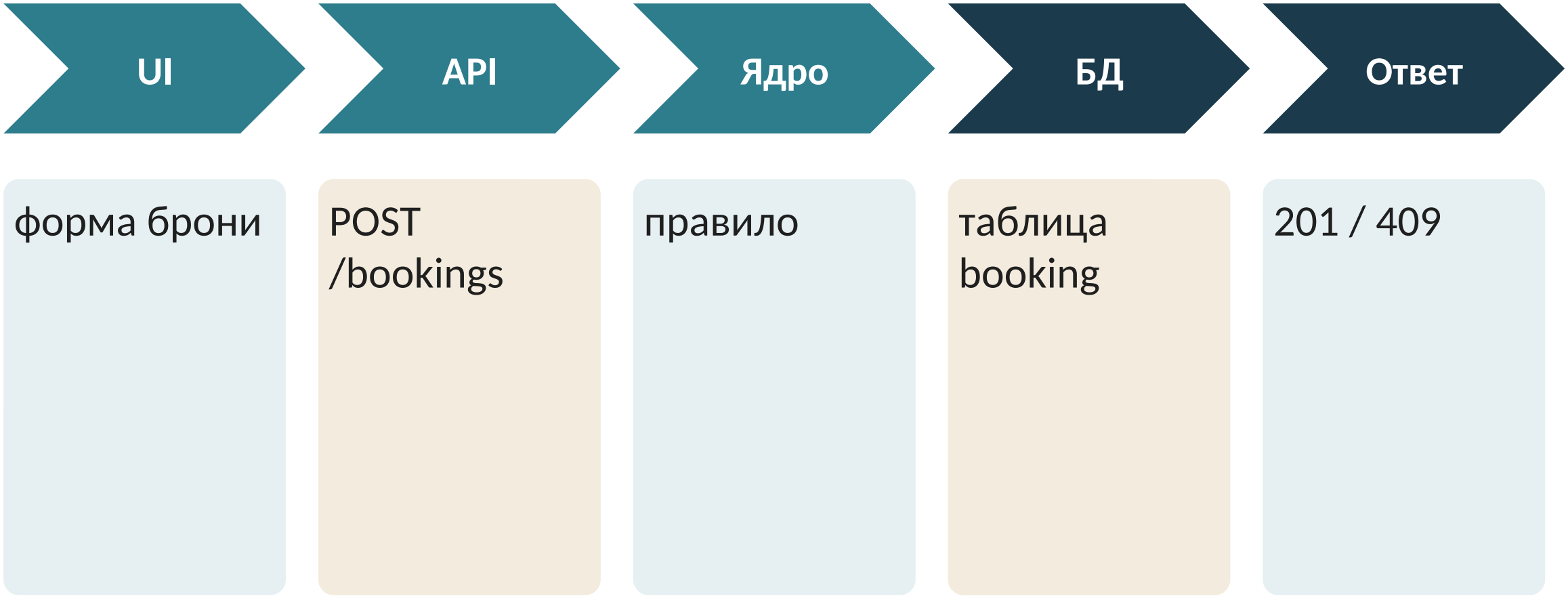


Эволюционный

- растёт до системы
- недели 9–15
- качество с начала

MVP проверяет рынок, прототип курса — инженерные решения

«Ходячий скелет»

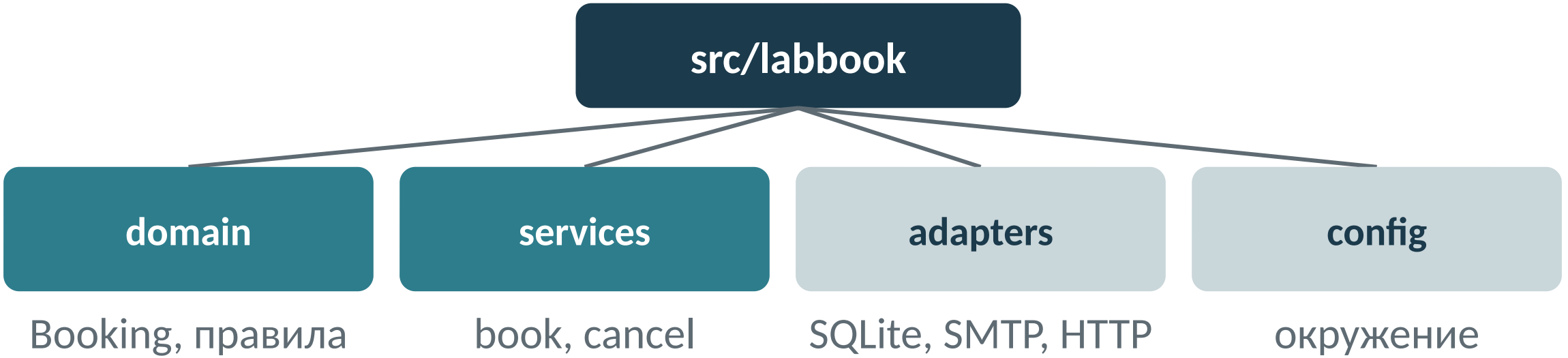


Тонкий вертикальный срез через все слои — первая неделя

Структура репозитория

Путь	Содержимое
src/labbook/	код по модулям архитектуры
tests/	модульные, интеграционные, e2e
docs/	SRS, ADR, контракты, отчёт
experiments/	код и данные экспериментов
scripts/	инициализация БД, демо-данные
.env.example	конфигурация без секретов

Код повторяет архитектуру



Ядро не импортирует sqlite3

Вертикальный срез на Python

l09_prototype.py

```python
import sqlite3
class Repo:
    def __init__(self, path=":memory:"):
        self.db = sqlite3.connect(path)
        self.db.executescript("""CREATE TABLE booking(
          id INTEGER PRIMARY KEY, eq INT, usr INT,
          s TEXT, e TEXT, CHECK (s < e));""")
    def overlaps(self, eq, s, e):
        q = "SELECT 1 FROM booking WHERE eq=? AND s<? AND ?<e"
        return self.db.execute(q, (eq, e, s)).fetchone()
    def add(self, eq, usr, s, e):
        cur = self.db.execute("INSERT INTO booking(eq,usr,s,e)"
                              " VALUES (?,?,?,?)", (eq, usr, s, e))
        return cur.lastrowid

def book(repo, eq, usr, s, e):
    if repo.overlaps(eq, s, e):
        return {"status": 409, "error": "SLOT_TAKEN"}
    return {"status": 201, "id": repo.add(eq, usr, s, e)}

r = Repo()
print(book(r, 3, 101, "2027-03-12T10:00", "2027-03-12T12:00"))
print(book(r, 3, 102, "2027-03-12T11:00", "2027-03-12T13:00"))
```

```
$ python l09_prototype.py
{'status': 201, 'id': 1}
{'status': 409, 'error': 'SLOT_TAKEN'}
```

- CHECK (s < e)
- параметры ?
- BEGIN IMMEDIATE в рабочей версии

201, затем 409 SLOT_TAKEN

Качество кода

PEP 8

стиль и типы

Форматтер

black, ruff

Линтер

ruff, flake8

Типы

mypy

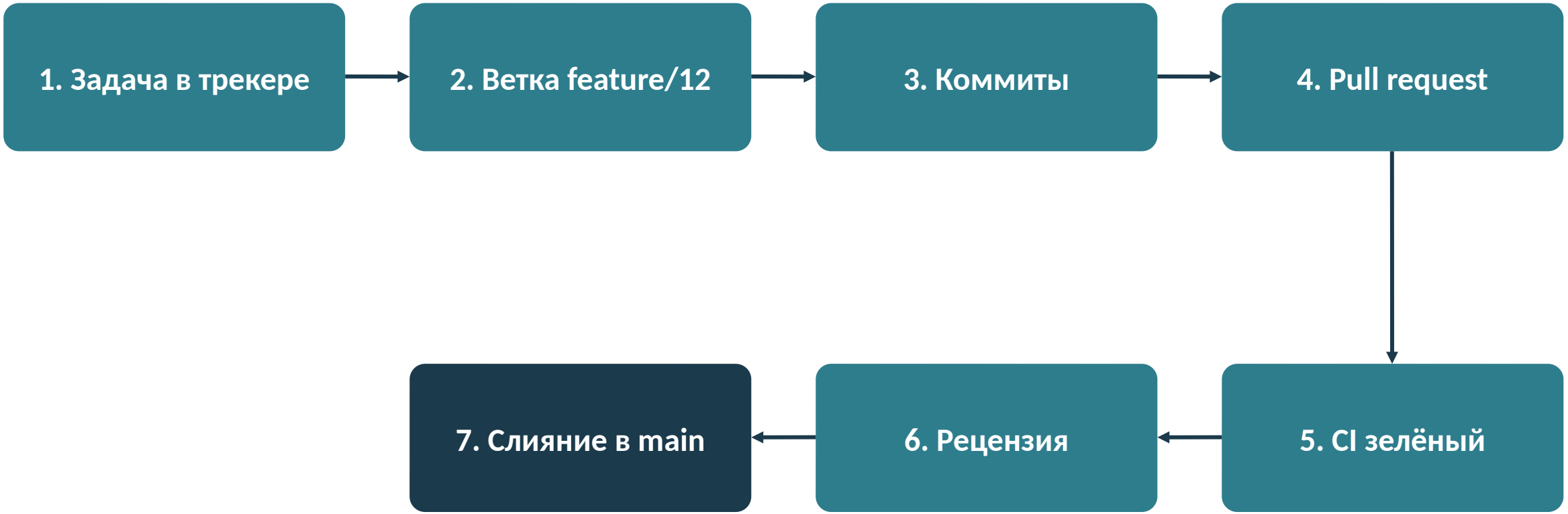
Рецензия

до 400 строк

pre-commit

и CI

GitHub flow в команде



История Git подтверждает индивидуальный вклад

Definition of Done

Условие	Проверка
код по критериям приёмки	ссылка на FR в PR
тесты проходят	зелёный CI
линтер без замечаний	шаг lint
рецензия пройдена	одобрение PR
документация обновлена	изменения docs/

Конфигурация и секреты



Окружение

переменные, а не код



Секреты

не в репозитории; утёк — отозвать



Зависимости

версии ==, venv, лицензии

Методология «двенадцати факторов»

Интерфейс прототипа



Путь

3–4 шага сценария



Сообщения

язык пользователя, не коды



Состояния

загрузка, пусто, ошибка



Доступность

подписи, контраст

Инструкция запуска в README

l14_README.md

```
# LabBook — бронирование оборудования лаборатории
## Быстрый запуск (5 минут)
    git clone https://gitlab.example.edu/team7/labbook
    cd labbook && git checkout v1.0
    python -m venv .venv && . .venv/bin/activate
    pip install -r requirements.txt
    python scripts/init_db.py --demo
    python -m labbook.api        # http://localhost:8000
## Проверка
    python -m unittest -v        # 31 тест
    python scripts/e2e_smoke.py  # сквозной сценарий
```

- команды по порядку
- демо-данные
- адрес и учётные записи
- запуск тестов

«Тест новичка»

0

подсказок от команды

≤ 15 мин

до работающей системы

100%

шагов по README



Участник другой команды запускает прототип на чистой машине

Результат недели 9

Сценарий

упрощённый, работает

Must ядра

реализованы

Тесты

модульные в CI

README

проверен новичком

Тег

v0.9

Вклад

у каждого коммиты

Вопросы для самопроверки

1. Виды прототипов
2. Ходячий скелет
3. Репозиторий
4. Порты в коде
5. SQL-параметры
6. Качество кода
7. Секреты
8. Инструкция



Неделя 9: прототип и инструкция (4 балла)