ЛЕКЦИЯ 3

Инженерия требований

CSE842 Capstone Project
ОП 6B06102 Computer Science



План лекции

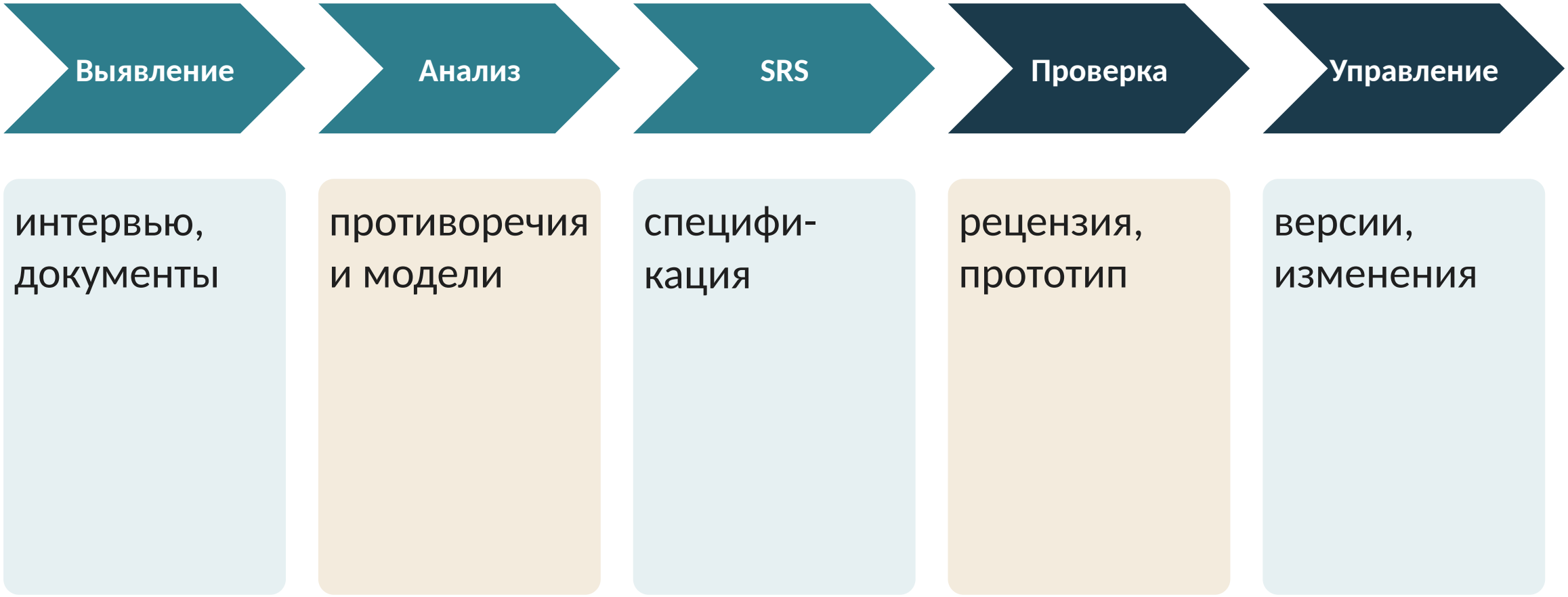
1. Уровни
2. Процесс
3. Выявление
4. FR и истории
5. NFR
6. Качество
7. MoSCoW
8. Gherkin

Уровни требований по Вигерсу



Каждое FR ведёт к бизнес-цели

Процесс инженерии требований



Базовая линия меняется только через запрос на изменение

Методы выявления требований

Интервью

3–5 на роль

Наблюдение

неявные правила

Анкеты

частота проблем

Документы

регламенты, журналы

Прототип

макет интерфейса

Воркшоп

общие ожидания

User story и use case



User story

- Как студент, я хочу видеть свободные интервалы,
- чтобы не ходить в лабораторию
- + критерии приёмки



Use case

- предусловие: вход
- основной поток
- альтернатива: занято
- исключение: ремонт

Измеримые формулировки NFR

Плохо	Хорошо
быстрая	p95 ≤ 300 мс при 50 RPS, нагрузочный тест
надёжная	0 двойных броней на 1000 параллельных заявок
удобная	4 из 5 новичков оформляют бронь < 2 мин
безопасная	bcrypt, HTTPS, чужая бронь → 403

Хорошее требование



Необходимое

связано с целью



Однозначное

без «быстро», «удобно»



Проверяемое

есть тест



Выполнимое

в ограничениях



Атомарное

одна мысль



Трассируемое

ID и источник

Проверка измеримости кодом

l03_requirements.py

```python
import re
reqs = [
 ("FR-1", "M", "Студент бронирует прибор на интервал"),
 ("FR-2", "M", "Система отклоняет пересекающуюся бронь"),
 ("FR-3", "S", "Лаборант получает уведомление о брони"),
 ("NFR-1", "M", "Ответ API p95 не более 300 мс при 50 RPS"),
 ("NFR-2", "S", "Интерфейс должен быть удобным"),
]
unit = re.compile(r"\d+\s*(мс|с|%|RPS|шт)")
for rid, prio, text in reqs:
    if rid.startswith("NFR") and not unit.search(text):
        print(f"{rid}: неизмеримо -> переписать")
must = sum(p == "M" for _, p, _ in reqs)
print(f"Must: {must} из {len(reqs)}")
```

```
$ python l03_requirements.py
NFR-2: неизмеримо -> переписать
Must: 3 из 5
```

- NFR без единиц — переписать
- Must: 3 из 5
- Must ≤ 60% трудоёмкости

Листинг 1: регулярное выражение ищет число и единицу

Приоритизация MoSCoW



Must

бронь, проверка конфликтов



Should

уведомление лаборанта



Could

экспорт в iCal



Won't

мобильное приложение

Критерий приёмки Given–When–Then

l03_booking.feature

```
Функция: Бронирование оборудования
  Сценарий: отказ при пересечении интервалов
    Дано прибор "Осциллограф-3" забронирован
      на 10:00-12:00 12.03.2027
    Когда студент бронирует его на 11:00-13:00
    Тогда система возвращает код 409
    И бронь студента не сохраняется
    И пользователь видит ближайший свободный слот
```

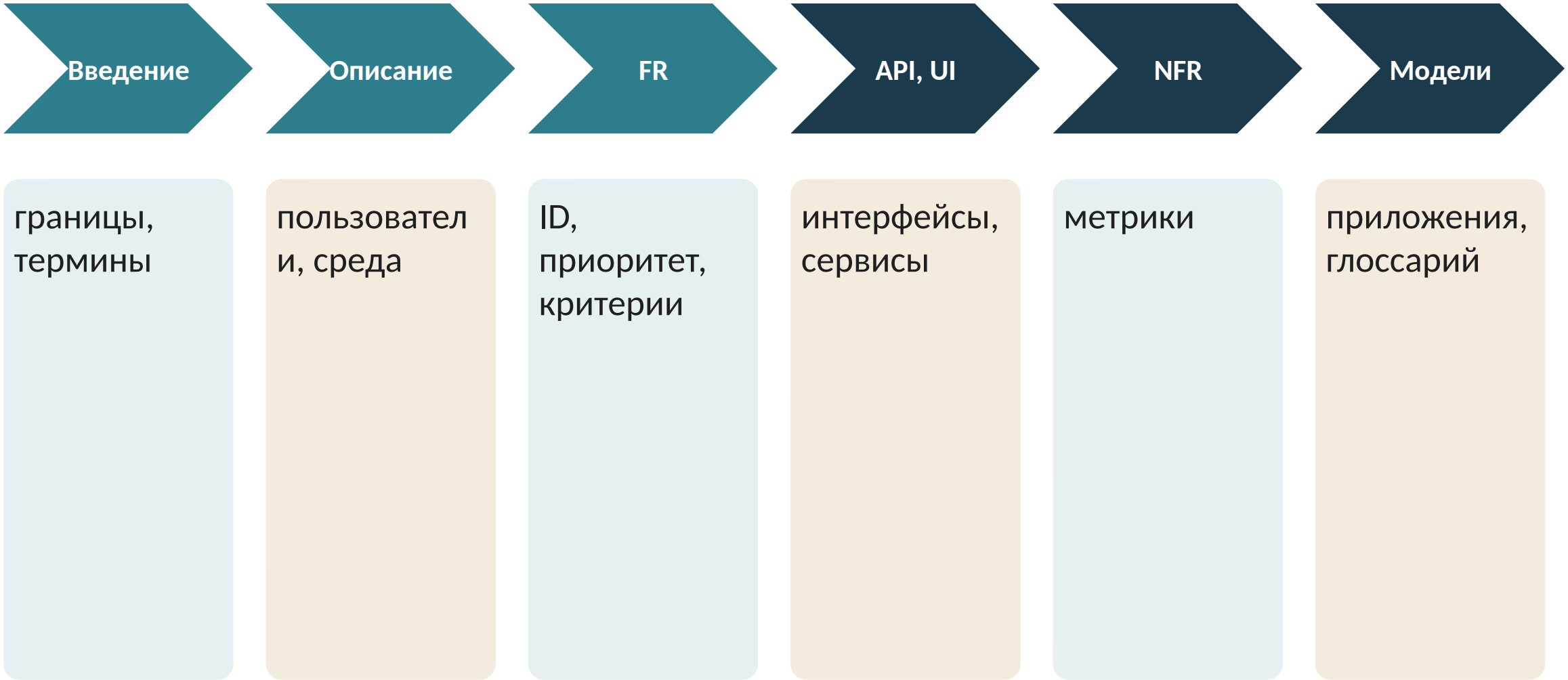
- Дано — состояние
- Когда — действие
- Тогда — результат
- становится тестом

Критерии приёмки FR-2: классы

Сценарий	Новая бронь	Ожидание
основной	13:00–14:00	201
пересечение	11:00–13:00	409
граница	12:00–14:00	201
ошибка	конец раньше начала	422

Существующая бронь: 10:00–12:00

Структура спецификации SRS



Фрагмент реестра требований

ID	Приор.	Требование	Проверка
FR-1	M	студент создаёт бронь	авто-тест
FR-2	M	пересечение отклоняется, 409	авто-тест
FR-3	S	уведомление лаборанту ≤ 1 мин	интеграция
NFR-1	M	p95 ≤ 300 мс при 50 RPS	нагрузка

Проверка и изменения требований



Рецензия

чек-лист: ID, источник, критерий



Прототип

показать пользователям



Изменения

запрос, версия, трассировка

Требования в Markdown — история изменений видна в Git

Вопросы для самопроверки

1. Уровни требований
2. FR, NFR, ограничения
3. Методы выявления
4. Измеримое NFR
5. Хорошее требование
6. MoSCoW
7. Given–When–Then
8. Разделы SRS



Неделя 3: спецификация требований (3 балла)