ЛЕКЦИЯ 2

Анализ существующих решений и техническая осуществимость

CSE842 Capstone Project
ОП 6B06102 Computer Science



План лекции

1. Аналоги
2. Источники
3. Критерии
4. Альтернативы
5. Матрица
6. Анализ весов
7. Осуществимость
8. Риски

Три вопроса до проектирования



Решено ли раньше?

продукты, код, статьи



Чем не устраивает?

ниша проекта



Что доказано?

проверенные подходы

NASA: Decision Analysis — альтернативы, критерии, обоснованный выбор

Источники поиска аналогов

Продукты

SaaS, магазины

Открытый код

GitHub, лицензии

Публикации

IEEE, ACM, Scholar

Стандарты

SWEBOK, ISO, OWASP

Интервью

как сейчас решают

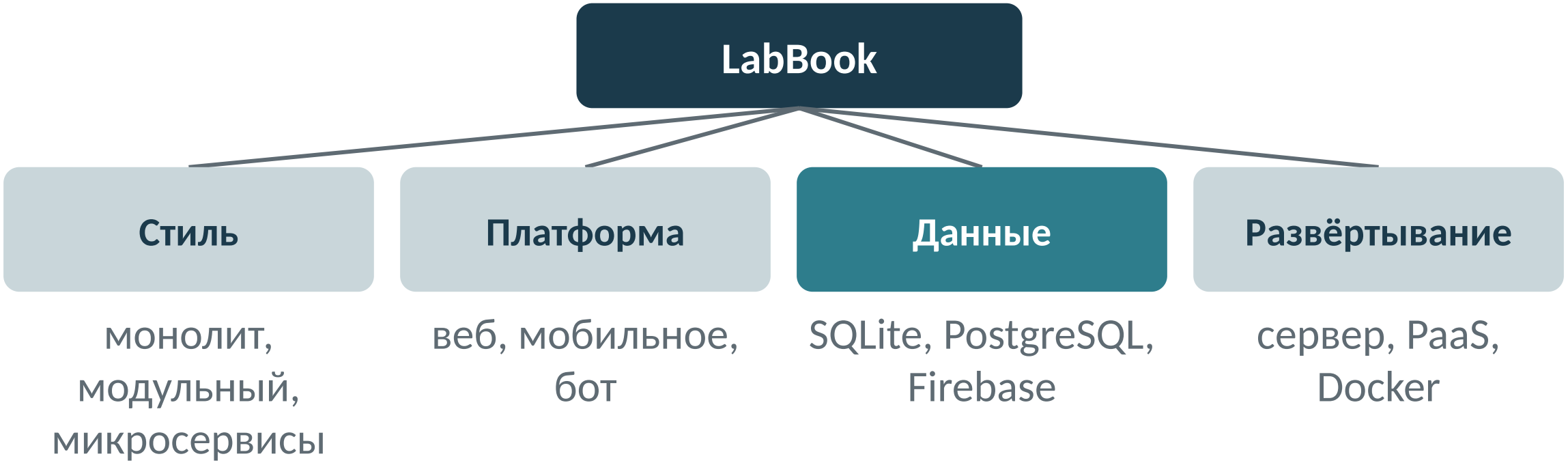
Дата проверки

аналоги меняются

Сравнение аналогов LabBook

Критерий	Журнал	Calendar	Skedda	LabBook
Проверка пересечений	нет	частично	да	да
Роль лаборанта	да	нет	частично	да
Уведомления	нет	да	да	да
Стоимость	0	0	платно	0
Данные в вузе	да	нет	нет	да

Пространство технических решений



Не менее 3 альтернатив, включая самую простую

Матрица альтернатив: веса

0,30	0,25	0,20	0,25
срок освоения	производитель-ность	стоимость	зрелость экосистемы

$S(a) = \Sigma\, w_i \cdot x_i(a)$, $\Sigma\, w_i = 1$; веса согласуются до выставления оценок

Расчёт взвешенной матрицы

l02_matrix.py

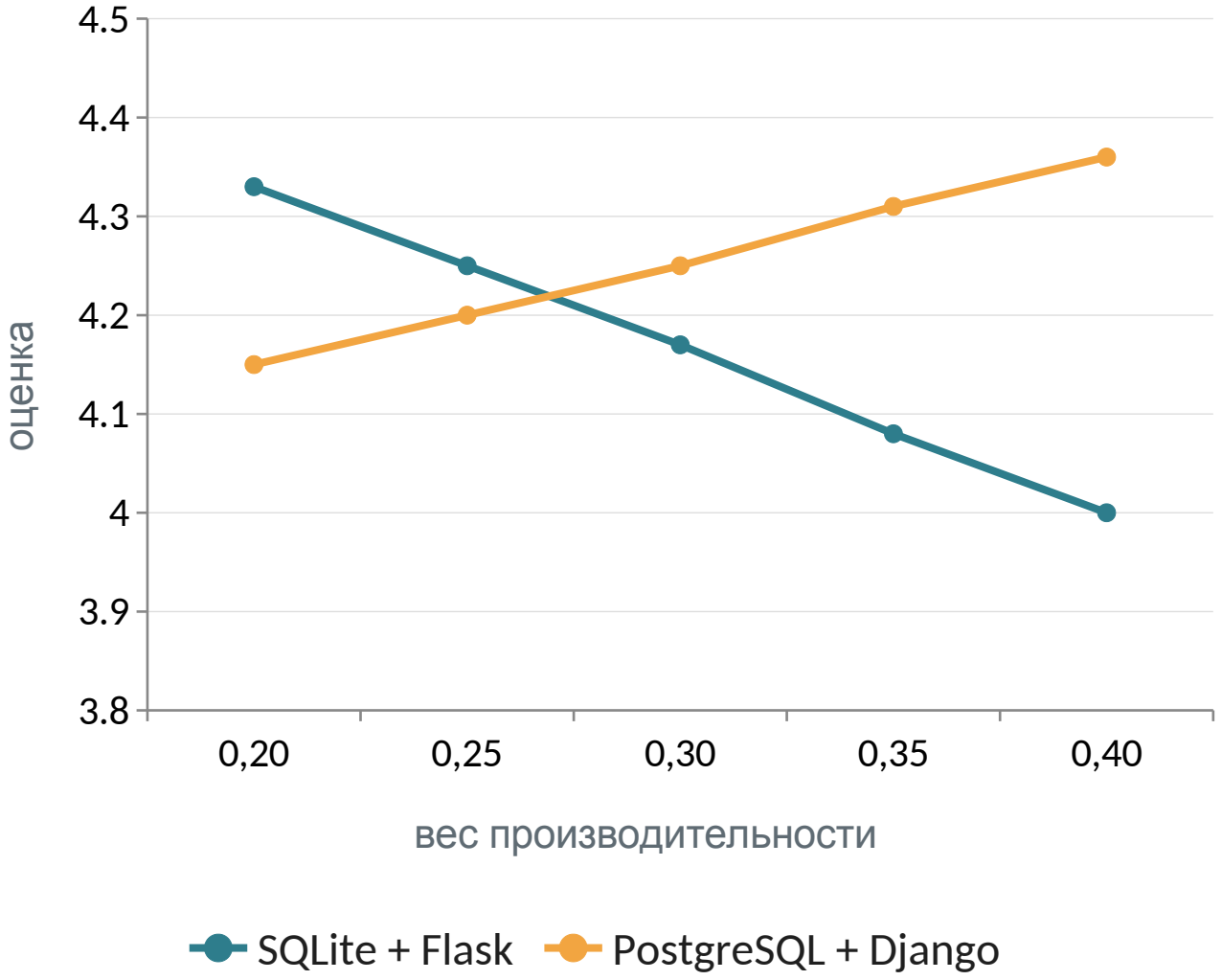
```
crit = {"срок освоения": 0.30, "производительность": 0.25,
        "стоимость": 0.20, "зрелость экосистемы": 0.25}
alts = {  # оценки 1..5 по каждому критерию
    "SQLite + Flask":   [5, 3, 5, 4],
    "PostgreSQL + Django": [3, 5, 4, 5],
    "Firebase":         [4, 4, 2, 4]}
w = list(crit.values())
score = {a: sum(x * k for x, k in zip(v, w))
         for a, v in alts.items()}
for a, s in sorted(score.items(), key=lambda t: -t[1]):
    print(f"{a:20s} {s:.2f}")
# чувствительность: вес производительности 0.40
w2 = [0.20, 0.40, 0.15, 0.25]
best = max(alts, key=lambda a: sum(
    x * k for x, k in zip(alts[a], w2)))
print("при смене весов лучший:", best)
```

```
$ python l02_matrix.py
SQLite + Flask       4.25
PostgreSQL + Django  4.20
Firebase             3.60
при смене весов лучший: PostgreSQL + Django
```

- SQLite + Flask: 4,25
- PostgreSQL + Django: 4,20
- Firebase: 3,60
- разница 0,05 — мала

Лидер неустойчив — нужен анализ чувствительности

Анализ чувствительности

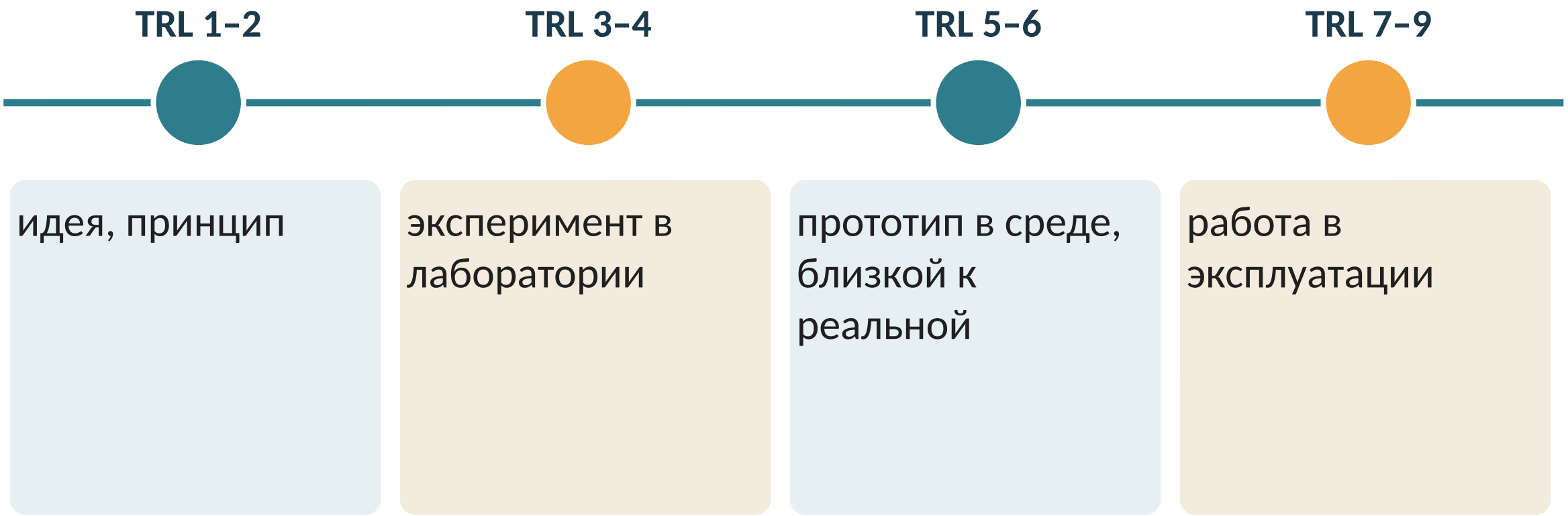


- порог ≈ 0,27
- близко к базовому весу 0,25
- решение: SQLite + порт репозитория
- проверить экспериментом

Пять видов осуществимости

Вид	Вопрос	Проверка
Техническая	умеем ли?	пробный запуск Flask + SQLite
Экономическая	хватит ли бюджета?	бесплатный хостинг
Временная	успеем ли?	оценка задач, буфер 20%
Операционная	будут ли пользоваться?	интервью с лаборантом
Юридическая	законно ли?	ПДн, лицензии библиотек

Технологическая готовность TRL



Студенческий прототип — обычно TRL 4–6

Приоритизация рисков

Вероятность

Наблюдать
вероятно, но слабо: письма в спам

Действовать сейчас
двойная бронь при параллельных запросах (15)

Принять
маловероятно и слабо

План на случай
уход участника команды (8)

Влияние на цель проекта

Стратегии реагирования на риск

Избежать

выбрать другую технологию

Снизить

proof of concept, обучение

Передать

готовый сервис

Принять

резерв времени

Самый рискованный пункт — в эксперимент недели 6

Создавать или брать готовое



Создаём сами

- ядро сквозного сценария
- бронь и конфликты
- новизна проекта



Берём готовое

- аутентификация, почта
- UI-библиотеки
- с указанием лицензии

MIT, Apache 2.0, BSD — свободно; GPL — обязательства

Результат недели 2

Аналоги	Альтернативы	Матрица	Оценка	Риски
3–5 и ниша	≥ 3 на подсистему	веса и расчёт	5 видов осуществимости	реестр и план

Вопросы для самопроверки

1. Зачем анализ аналогов
2. Источники поиска
3. Матрица альтернатив
4. Чувствительность
5. Виды осуществимости
6. TRL и PoC
7. Приоритет риска



Неделя 2: матрица альтернатив (2 балла)