ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7
ПОДГОТОВКА ПЛАНА РЕАЛИЗАЦИИ И ПРОВЕРКИ

Дисциплина: CSE842 «Capstone Project». **Образовательная программа:** 6B06102 «Computer Science», 4 курс, 8 семестр. **Связь с лекцией:** лекция 7 «Проектирование проверок и планирование интеграции». **Результаты обучения:** PO2, PO4, PO6. **Объём:** 3 часа практики, 7 часов самостоятельной работы, включая СРСП. **Максимальный балл:** 2.

Цель занятия: составить план реализации и проверки: стратегию тестирования, тестовые случаи, матрицу трассируемости и порядок интеграции.

Задачи:

1. Определить стратегию проверок: уровни, инструменты, среды, критерии входа и выхода.
2. Спроектировать тестовые случаи для требований Must (классы эквивалентности, граничные значения).
3. Составить матрицу «требование — компонент — тест».
4. Определить порядок интеграции, заглушки и интеграционные тесты.
5. Составить календарный план недель 9–15.

Программное обеспечение и инструменты: Markdown или электронные таблицы, Python (проверка матрицы), трекер задач.

1. Краткие теоретические сведения

V-модель связывает этап проектирования с уровнем проверки. Пирамида тестов: много модульных, меньше интеграционных, мало сквозных. Тестовые случаи выбираются разбиением на классы эквивалентности и анализом граничных значений. Матрица трассируемости показывает требования без тестов и компоненты без требований. Рекомендуемая стратегия интеграции — «ходячий скелет».

Проверка матрицы:

```
matrix = {"FR-1": ("Service", ["t_ok"]),
          "FR-3": ("Notifier", [])}
gaps = [r for r, (_, t) in matrix.items() if not t]
print("без тестов:", gaps)
```

Таблица 1 — Шаблон матрицы трассируемости

Требование	Компонент	Тесты	Уровень
FR-2	BookingService	T2–T7 (классы и границы)	модульный
FR-3	Notifier + outbox	IT-4 с имитацией SMTP	интеграционный
NFR-1	API + БД	PERF-1 (Locust, 50 RPS)	системный

2. Порядок выполнения работы

1. Опишите стратегию проверок и критерии входа/выхода испытаний.
2. Для каждого требования Must спроектируйте тесты: класс, граница, ошибка; используйте сценарии варианта (таблица 2).
3. Составьте матрицу трассируемости и проверьте её скриптом на пропуски.
4. Определите порядок интеграции подсистем, заглушки и интеграционный тест для каждого шага.
5. Настройте заготовку CI (пустой набор тестов, линтер).
6. Составьте календарный план недель 9–15 с ответственными и создайте задачи в трекере.

3. Индивидуальные задания

Номер варианта совпадает с номером темы проекта, выбранной на практическом занятии № 1, и сохраняется до конца семестра. Команда (до 4 человек) выполняет задание своего варианта; распределение работы между участниками указывается в отчёте.

Таблица 2 — Индивидуальные задания: проверки и интеграция

№	Граничные сценарии	Ошибочные сценарии	Заглушка при интеграции
1	слот, начинающийся сейчас; последний слот дня	двойное нажатие «Записаться»; отмена чужой записи	SMS-шлюз
2	токен на 29-й и 31-й секунде; последний студент группы	повторное сканирование; чужой токен	API расписания
3	значение ровно на пороге; пустой интервал	отрицательная концентрация; дубликат измерения	физический датчик (эмулятор)
4	решение ровно на лимите времени; пустой файл	fork-бомба; попытка сетевого запроса	GitHub API
5	покупка последней единицы; корзина из 1 товара	два заказа на последнюю единицу; отрицательное количество	платёжный шлюз
6	возврат в последний день срока; лимит 5 книг	выдача уже выданного экземпляра; неизвестный штрихкод	Open Library API
7	вопрос из 1 слова; 4000 символов	эмодзи и стикеры; вопрос на казахском	Telegram Bot API

№	Граничные сценарии	Ошибочные сценарии	Заглушка при интеграции
8	тень на месте; частично заехавший автомобиль	камера отключена; пустой кадр	видеопоток (записанные файлы)
9	отгрузка ровно остатка; пустая ячейка	повторная отправка операции; отгрузка сверх остатка	учётная система
10	курьер ровно на границе зоны; последний свободный курьер	два диспетчера назначают одновременно; GPS-скачок	ресторанная касса
11	окно на границе суток; пустое окно	повреждённая строка журнала; неизвестный шаблон	продуктивный сервер (воспроизведение журнала)
12	чек с суммой 0; очень длинный чек	размытое фото; повторный чек	облачный OCR
13	согласование последним участником маршрута; пустое поле шаблона	изменение после подписи; повторное согласование	NCALayer
14	оплата на 10-й минуте блокировки; последний слот	повторный webhook; отказ оплаты	платёжный шлюз
15	два правила на одно событие; датчик на границе расписания	устройство офлайн; повтор события	физические реле
16	пропуск истекает сегодня; номер с похожими символами 0/O	грязный номер; два автомобиля в кадре	шлагбаум
17	переход на летнее время; ровно порог перерасхода	счётчик сбросился в 0; пропуск 6 ч	счётчики Modbus
18	ответ за 1 с до конца; последний вопрос	две вкладки с одной попыткой; разрыв сети	LMS Moodle
19	отправление ровно в момент запроса; последняя пересадка дня	одинаковые точки; повреждённый фид	поставщик GTFS

№	Граничные сценарии	Ошибочные сценарии	Заглушка при интеграции
20	слот вплотную к паре; изменение во время записи	удаление события в Google; истёк токен	Google Calendar API

4. Содержание отчёта

1. Стратегия проверок.
2. Список тестовых случаев для требований Must.
3. Матрица трассируемости и результат проверки покрытия.
4. Порядок интеграции с заглушками.
5. Календарный план недель 9–15.

5. Форма сдачи

Артефакты занятия размещаются в репозитории проекта (каталог docs/ или указанный в задании), в трекере задач создаются и закрываются соответствующие задачи. Преподавателю сдаётся ссылка на версию (коммит или тег) и краткий отчёт.

6. Критерии оценивания

Максимальный балл за работу — **2** (календарный план силлабуса). Баллы распределяются по трём критериям силлабуса.

Таблица 3 — Критерии оценивания текущего задания

Критерий	Доля	Что проверяется
Полнота и соответствие требованиям задания	40%	все пункты задания и индивидуального варианта выполнены
Техническая корректность	40%	решения обоснованы, расчёты и код верны, терминология точна
Проверяемость материалов	20%	артефакты в репозитории, ссылки, команды запуска, данные

Работа сдаётся через канал, указанный преподавателем; в командном проекте каждый участник указывает свой вклад. Использование генеративного ИИ раскрывается по правилам университета.

7. Контрольные вопросы

1. Чем верификация отличается от валидации?
2. Как применены классы эквивалентности в вашем проекте?
3. Какие требования пока не покрыты тестами?
4. Почему выбран такой порядок интеграции?
5. Каковы критерии выхода испытаний?

Литература

1. Wiegers K., Beatty J. Software Requirements. — 3rd ed. — Microsoft Press, 2013.
2. Ammann P., Offutt J. Introduction to Software Testing. — 2nd ed. — Cambridge University Press, 2016.
3. NASA. Systems Engineering Handbook. NASA/SP-2016-6105 Rev2.