ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 12
ОТЛАДКА И ПОВТОРНАЯ ПРОВЕРКА

Дисциплина: CSE842 «Capstone Project». **Образовательная программа:** 6B06102 «Computer Science», 4 курс, 8 семестр. **Связь с лекцией:** лекция 12 «Диагностика и устранение неполадок». **Результаты обучения:** РО6. **Объём:** 3 часа практики, 7 часов самостоятельной работы, включая СРСП. **Максимальный балл:** 3.

Цель занятия: воспроизвести дефекты, найденные при интеграции и испытаниях, найти их причины, исправить и подтвердить исправления регрессионными тестами.

Задачи:

1. Провести триаж дефектов: серьёзность, приоритет, ответственный.
2. Воспроизвести и минимизировать каждый дефект, написать тест на дефект.
3. Диагностировать причину (журналы, отладчик, git bisect) и исправить её.
4. Выполнить регрессионное тестирование и вести журнал дефектов.
5. Провести анализ первопричины для самого серьёзного дефекта.

Программное обеспечение и инструменты: logging, pdb или отладчик IDE, git bisect, unittest/pytest, CI.

1. Краткие теоретические сведения

Цикл диагностики: симптом → воспроизведение → гипотеза → эксперимент → исправление → регрессия. Дефект исправляется в порядке: тест падает → исправить причину → тест проходит → весь набор → коммит со ссылкой на дефект → рецензия. git bisect находит коммит с дефектом за $\log_2(N)$ проверок.

Журналирование с контекстом:

```
log = logging.getLogger("booking")
log.info("booking.created id=%s req=%s",
         booking_id, request_id)
```

Таблица 1 — Шаблон журнала дефектов

ID	Симптом	Серьёзн.	Причина и исправление	Регрессия
BUG-14	смежная бронь отклоняется	высокая	<= заменено на <, коммит 5e1c	31/31 PASS
BUG-17	письмо уходит дважды	средняя	проверка id в outbox	33/33 PASS

2. Порядок выполнения работы

1. Проведите триаж всех открытых дефектов; определите, какие исправляются до РК2.

2. Для каждого дефекта напишите шаги воспроизведения и минимальный пример; оформите тест, который падает.
3. Добавьте журналирование с идентификатором запроса в ключевые точки сценария.
4. Найдите причину учебного дефекта варианта (таблица 2); при необходимости используйте git bisect.
5. Исправьте причину, убедитесь, что тест на дефект и весь набор проходят.
6. Ведите журнал дефектов с результатами регрессионных прогонов после каждого исправления.
7. Проведите анализ «пять почему» для самого серьёзного дефекта и предложите улучшение процесса.

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: учебный дефект для диагностики

№	Дефект (воспроизвести и исправить)	Регрессионная проверка
1	время слота хранится в местном поясе, сравнивается с UTC	тест на дефект + весь набор; сценарий: слот, начинающийся сейчас
2	часы телефона спешат на 40 с — токен отклоняется	тест на дефект + весь набор; сценарий: токен на 29-й и 31-й секунде
3	дубликаты измерений после переподключения (QoS 1)	тест на дефект + весь набор; сценарий: значение ровно на пороге
4	зомби-контейнеры после тайм-аута заполняют диск	тест на дефект + весь набор; сценарий: решение ровно на лимите времени
5	остаток уменьшается до оплаты и не возвращается при отказе	тест на дефект + весь набор; сценарий: покупка последней единицы
6	поиск не находит «алгоритмы» по запросу «алгоритм»	тест на дефект + весь набор; сценарий: возврат в последний день срока
7	бот отвечает дважды из-за повторной доставки webhook	тест на дефект + весь набор; сценарий: вопрос из 1 слова
8	при смене дня на ночь все места «заняты»	тест на дефект + весь набор; сценарий: тень на месте

№	Дефект (воспроизвести и исправить)	Регрессионная проверка
9	операция применяется дважды после повторной синхронизации	тест на дефект + весь набор; сценарий: отгрузка ровно остатка
10	маркер курьера «прыгает» назад из-за неупорядоченных сообщений	тест на дефект + весь набор; сценарий: курьер ровно на границе зоны
11	оповещение дублируется при перезапуске детектора	тест на дефект + весь набор; сценарий: окно на границе суток
12	разделитель тысяч распознаётся как десятичная точка	тест на дефект + весь набор; сценарий: чек с суммой 0
13	кириллица в PDF заменяется квадратами	тест на дефект + весь набор; сценарий: согласование последним участником маршрута
14	истёкшая блокировка не снимается — слот «завис»	тест на дефект + весь набор; сценарий: оплата на 10-й минуте блокировки
15	правило зацикливается: свет включает датчик, датчик — свет	тест на дефект + весь набор; сценарий: два правила на одно событие
16	буква О и цифра 0 путаются — отказ для действующего пропуска	тест на дефект + весь набор; сценарий: пропуск истекает сегодня
17	сброс счётчика даёт огромное отрицательное потребление	тест на дефект + весь набор; сценарий: переход на летнее время
18	серверные и клиентские часы расходятся — ответы отклоняются раньше времени	тест на дефект + весь набор; сценарий: ответ за 1 с до конца
19	пересадка с 0 минут на ту же остановку допускается	тест на дефект + весь набор; сценарий: отправление ровно в момент запроса
20	при смене часового пояса консультации сдвигаются на 5 ч	тест на дефект + весь набор; сценарий: слот вплотную к паре

4. Содержание отчёта

1. Журнал дефектов с причинами, исправлениями и результатами регрессии.

2. Тесты на дефекты в репозитории.
3. Описание диагностики учебного дефекта (шаги, гипотезы, инструменты).
4. Анализ первопричины.

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 — **3** (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. Какие поля содержит отчёт о дефекте?
2. Как вы минимизировали пример воспроизведения?
3. Как работает git bisect?
4. Почему тест на дефект пишется до исправления?
5. Что показал анализ «пять почему»?

Литература

1. Ammann P., Offutt J. Introduction to Software Testing. — 2nd ed. — Cambridge University Press, 2016.
2. Adkins H. et al. Building Secure and Reliable Systems. — O’Reilly, 2020.
3. Beyer B. et al. Site Reliability Engineering. Ch. 12: Effective Troubleshooting. — O’Reilly, 2016.