



Лекция №2. Теория тестирования



1. Что такое требования (спецификации/спеки)
2. Что такое баг
3. Приоритеты, критичность багов
4. Уровни тестирования ПО
5. Критерии входа/выхода
6. 7 базовых принципов тестирования
7. Тест-кейсы/тест-планы/сценарии тестирования/чек листы/чит-листы
8. Причины появления багов
9. Тестовое покрытие
10. Этапы тестирования:
 - подготовка;
 - исполнение;
 - итоги.
11. ISTQB-сертификация

Что такое требование (спецификации/спеки)

Требования — совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации.

Бизнес-требования — определяют назначение ПО.

Пользовательские требования — определяют набор пользовательских задач, которые должно решать программа.

Функциональные требования — определяют действия, которые система способна выполнять

Что такое баг

Баг (дефект) — ошибка в программе или в системе, из-за которой ПО выдает неожиданное поведение и непредсказуемый результат.

Приоритеты, критичность багов

5

По уровню критичности баги бывают:

- Блокирующие (Blocker)
- Критические (Critical)
- Значительные (Major)
- Незначительный (Minor)
- Тривиальные (Trivial)

По приоритету:

- Высокий (High)
- Средний (Medium)
- Низкий (Low)

Уровни тестирования ПО

6

1. Модульное (компонентное) тестирование (Unit)
2. Интеграционное тестирование (Integration)
3. Системное тестирование (System)
4. Операционное тестирование (Release)
5. Приемочное (Acceptance)

Уровни тестирования ПО

7

- 1. Модульное (компонентное) тестирование (Unit)**
2. Интеграционное тестирование (Integration)
3. Системное тестирование (System)
4. Операционное тестирование (Release)
5. Приемочное (Acceptance)

Модульное (компонентное) тестирование (Unit) — проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по отдельности (модули программ, объекты, классы, Функции и т.д)

Уровни тестирования ПО

9

1. Модульное (компонентное) тестирование (Unit)
- 2. Интеграционное тестирование (Integration)**
3. Системное тестирование (System)
4. Операционное тестирование (Release)
5. Приемочное (Acceptance)

Интеграционное тестирование (Integration) — проверяется взаимодействие между компонентами системы после проведения модульного тестирования

Уровни тестирования ПО

11

1. Модульное (компонентное) тестирование (Unit)
2. Интеграционное тестирование (Integration)
- 3. Системное тестирование (System)**
4. Операционное тестирование (Release)
5. Приемочное (Acceptance)

Системное тестирование (System) — проверка как функциональных, так и не функциональных требований в системе в целом. При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных пользовательского уровня, несовместимость с окружением, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство использования, т.д.

Уровни тестирования ПО

13

1. Модульное (компонентное) тестирование (Unit)
2. Интеграционное тестирование (Integration)
3. Системное тестирование (System)
- 4. Операционное тестирование (Release)**
5. Приемочное (Acceptance)

Операционное тестирование (Release) — проводится с целью убедиться, что система выполняет свою роль в среде эксплуатации согласно бизнес-модели.

Уровни тестирования ПО

15

1. Модульное (компонентное) тестирование (Unit)
2. Интеграционное тестирование (Integration)
3. Системное тестирование (System)
4. Операционное тестирование (Release)
- 5. Приемочное (Acceptance)**

Приемочное (Acceptance) — формальный процесс тестирования, который проверяет соответствие системы техническому заданию.

Критерии входа/выхода

17

Критерий входа описывает, что нужно сделать перед началом тестирования.

Критерий выхода описывает то, что считается необходимым для завершения испытаний.

Принципы тестирования

18

1. Тестирование демонстрирует наличие дефектов (Testing shows presence of defects).
2. Исчерпывающее тестирование недостижимо (Exhaustive testing is impossible).
3. Раннее тестирование (Early testing).
4. Скопление дефектов (Defects clustering).
5. Эффект пестицида (Pesticide paradox).
6. Тестирование зависит от контекста (Testing is concept depending).
7. Заблуждение об отсутствии ошибок (Absence-of-errors fallacy).

Тест-кейсы/тест-планы/сценарии тестирования/чек листы/чит-листы

Тест-кейс — последовательность действий, направленная на проверку какого-либо функционала, описывающая как прийти к фактическому результату.

Тест-комплект (сценарий тестирования) — совокупность тест-кейсов, которые могут быть выполнены, чтобы протестировать определенный функционал в необходимой степени.

TC ID/Priority	CCPG0002	1
IDEA: Оплата может быть произведена картой MasterCard SETUP and ADDITIONAL INFO: Эккаунт: testuser1/paSSwOrd Данные карты: Номер: 3333-7112-4444-7844 Окончание действия: 12/08 CVV2: 676		
Revision History		
Created on: 11/17/2003 by О.Тарасов	Новый тест-кейс	
Modified on: 11/26/2003 by И. Новикова	Шаги были упрощены, чтобы сделать тест-кейс более удобным для поддержки	
Modified on: 01/17/2003 by И. Новикова	Изменение шагов и второй ожидаемый результат с целью удостоверения в снятии денег со счета	
Execution part		
PROCEDURE	EXPECTED RESULT	
1. Запиши баланс счета карты 2. Открой www.main.testshop.rs 3. Войди в систему. 4. Найди любой товар. 5. Добавь товар в корзину. 6. Произведи <i>оплату картой</i> из секции SETUP and ADDITIONAL INFO (!!! запиши полную сумму заказа:). 7. Запиши номер заказа 8. Запроси базу данных с SQL1.	> "20"	
9. Запиши баланс счета карты	> Шаг 1 - Шаг 6	

Тест-кейсы/тест-планы/сценарии тестирования/чек листы/чит-листы

Тест-план — документ, описывающий весь объем работ по Тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

Что надо тестировать?

Что будем тестировать?

Как будем тестировать?

Когда будем тестировать?

Критерии начала/окончания тестирования.

Тест-кейсы/тест-планы/сценарии тестирования/чек листы/чит-листы

22

Чек-лист — документ, описывающий, что должно быть протестировано.

Чит-лист — список повторяющихся проверок, который можно переиспользовать.

Пример чек-листа

23

3	ID	Назначение	Шаги	Ожидаемый результат	Актуальный результат
4					
5	1	Проверка корректности орфографии формы.	Визуальный осмотр корректности орфографии надписей.	Корректные надписи	Провален
6	2	Проверка tab_order элементов формы.	1. Устанавливаем курсор в любой элемент формы. 2. Пробуем перемещаться по элементам формы при помощи клавиши TAB и комбинации клавиш Shift+TAB	Корректное перемещение по элементам формы (слева на право, сверху вниз)	Пройден
7	3	Проверка возможности регистрации с корректными данными, соответствующими спецификации.	1. Установка значений в поля: email = lol@mail.ru; Логин = 123swiss; Пароль = folkmetal. 2. Активация кнопки "Регистрация"	Сообщение об успешной регистрации.	Пройден
8	4	Проверка возможности регистрации нового пользователя с идентичными предыдущему данными.	1. Установка значений в поля: email = lol@mail.ru; Логин = 123swiss; Пароль = folkmetal. 2. Активация кнопки "Регистрация"	Сообщение о невозможности регистрации одного пользователя дважды.	Провален
9	5	Проверка возможности регистрации с пустыми полями формы	Активация кнопки "Регистрация"	Сообщение о необходимости заполнения пустых полей	Пройден

Пример чит-листа

24

МОЙ ЛЮБИМЫЙ ПРОЕКТ > ЧИТ-ЛИСТЫ > ПРОСМОТР "ПОЛЕ EMAIL"



Мой любимый проект ▼

Поле Email

Проверяется поле Email

?) ТИП ЧИТ-ЛИСТА

Перечень проверок

Проверка поля Email

НОМЕР	ПРОВЕРКА	ОЖИДАЕМЫЙ РЕЗУЛЬТАТ
chit 1 	Валидный email засчитывается	Письмо приходит
chit 2 	Email с дефисом в локальной части email	Письмо отправляется
chit 3 	Email с дефисом в доменной части email	Письмо отправляется
chit 4 	Пустое поле email	Сообщение о незаполненном поле email
chit 5 	Превышение длины email (>320 символов)	Сообщение о некорректном email
chit 6 	Два символа в email	Сообщение о некорректном email
chit 7 	Отсутствие @ в email	Сообщение о некорректном email
chit 8 	Email без доменной части	Сообщение о некорректном email
chit 9 	Email без локальной части	Сообщение о некорректном email

Причины появления багов

25

1. непонимание или отсутствие информации
2. Сложность программного обеспечения
3. Изменение требований
4. Плохо документированный код
5. Средства разработки ПО

Тестовое покрытие — плотность покрытия тестами требований, Либо исполняемого кода.

Покрытие требований:

$$Tcov = (Lcov/Ltotal) * 100\%$$

Tcov — тестовое покрытие

Lcov — количество требований, проверяемых тест-кейсами

Ltotal — общее количество требований

Покрытие кода:

$$Tcov = (Ltc/Lcode) * 100\%$$

Tcov — тестовое покрытие

Ltc — количество строк кода, покрытых тестами

Lcode — общее количество строк кода

1. Анализ
2. Разработка стратегии тестирования и планирование процедур
Контроля качества
3. Работа с требованиями
4. Создание тестовой документации
5. Тестирование прототипа
6. Основное тестирование
7. Стабилизация
8. Эксплуатация

Этапы тестирования (упрощенно)

28

1. Подготовка (тестирование требований, подготовка тест-кейсов/чек-листов, настройка тестового окружения...)
2. Исполнение (тестирование новых функциональностей/регрессионное тестирование)
3. Итоги (фиксация постановок в багтрекере)



[RC] Шапка не отображается при повторном открытии меню-бургера в мобильных браузерах

Edit Comment Assign More Reopen Issue Release Issue

Details

Type:	Bug	Status:	CLOSED (View Workflow)
Priority:	Major	Resolution:	Fixed
Labels:	None		
Источник требования:	Команда Russia.Study		
Спринт:	RISE Sprint 60		

Description

1) Где

Мобильный: android chrome, iPhone (safari)

<https://rise-rc.naumen.ru>, при переходе по информационным страницам и открытии меню-бургера, как под гостем, так и под кандидатом

2) Как воспроизвести

1. Перейти на любую информационную страницу при помощи меню-бургера (например, "образование в России" - "уровни образования")
2. Повторно открыть меню-бургер

Не отображается шапка, не отображается крестик, закрывающий меню



3) Как должно быть

Вернуть шапку (см на рабочем)