Тестирование ПО. Урок 4

Артефакты тестирования, требования

1

Требования

Требования

Требование (requirement) — описание того, какие функции и с соблюдением каких условий должно выполнять приложение в процессе решения полезной для пользователя задачи.

Бывают

- Продуктовые требования (PRD - Product Requirement Document)
- Системные требования (SRS - System Requirement Specification)
- Функциональные требования (FRS - Functional Requirement Specification)
- Технические требования (TR/TS - Technical Requirements/ Technical Specification)

Все они отличаются степенью детализации, то есть каждый следующий документ содержит всё больше деталей реализации.

Требования.Ожидание/Реальность

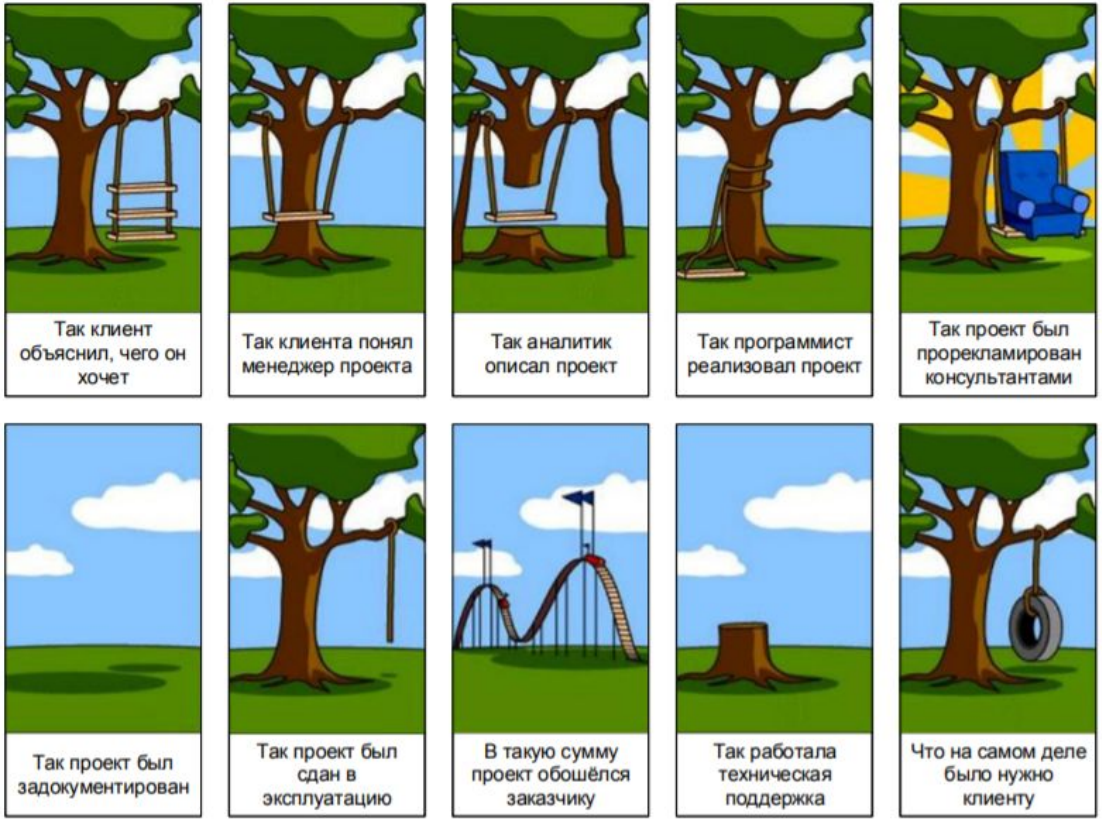


Рисунок 2.2.b — Типичный проект с плохими требованиями

Требования. Уровни

Рисунок 2.2.e — Уровни и типы требований

Требования. Свойства и качества

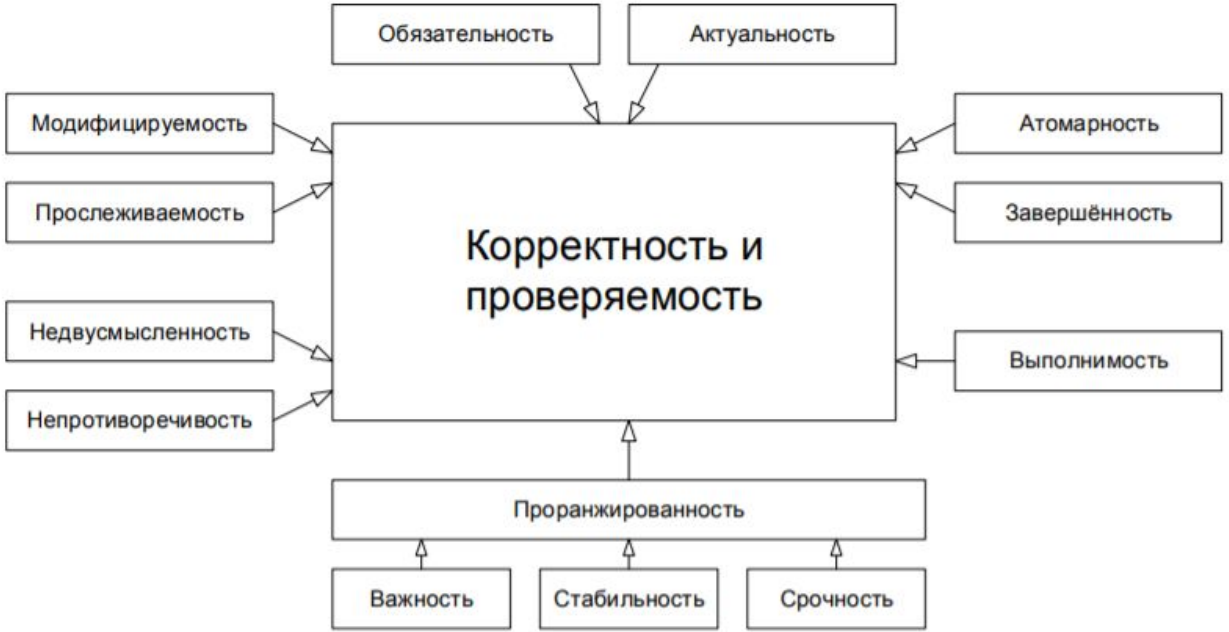


Рисунок 2.2.f — Свойства качественного требования

2

Артефакты тестирования

Артефакты тестирования

- **Тестовая стратегия, тест план** (Test Strategy, Test plan)
- **Тест кейсы** (Test Case)
- **Баг репорты** (Bug reports, defects, issues, etc.)
- **Тест репорт** (Test report)
- **Матрица соответствия требованиям** (Traceability matrix)
- **Чек-листы** (Checklist)
- **Автоматические тесты** (Automation test scenarios)

2.1

Тест план и тестовая стратегия

Тест план. Вопросы, на которые должен отвечать

Тест план. Шаблон

1. Test subject
<should be information about what we are going to test in next format>

2. Feature to be tested: <feature name>
Requirements: <links to the requirements>
Jira tasks: <link to Jira epic>

3. Features Not to Be Tested:
<what and why we are not going to test>

4. Target release
<release when feature should be ready>

5. Resources
<list of QA team that is going to take part in testing>

6. Environments
<where we're going to test>

7. Test Approach
<how we are going to test feature. Which types/levels of testing. Are we going to automate, cover on unit test level, etc.>

8. Testing Tools
<which instruments will be used>

9. Risk Analysis & Dependency
<any risks, people, resource dependency that can hold on the project>

10. Exit criteria
<criteria of finishing testing >

Тест план и Тестовая стратегия

Тестовая стратегия (Test Strategy) – описывает общие для организации, независимые от проекта методы тестирования, объемы тестирования, подходы и тд.

Тест план (Test plan) – документ, описывающий объем (scope) тестирования и тестовых активностей (техники, правила, методы, уровни и тд.):

- Мастер тест план (master test plan) – описывает применение тестовой стратегии организации на конкретном проекте
- План тестирования уровня(или фазы) – описывает активности по тестированию определенного уровня или отдельной части программы

2.2

Тест кейс и тестовые системы

Тест кейс

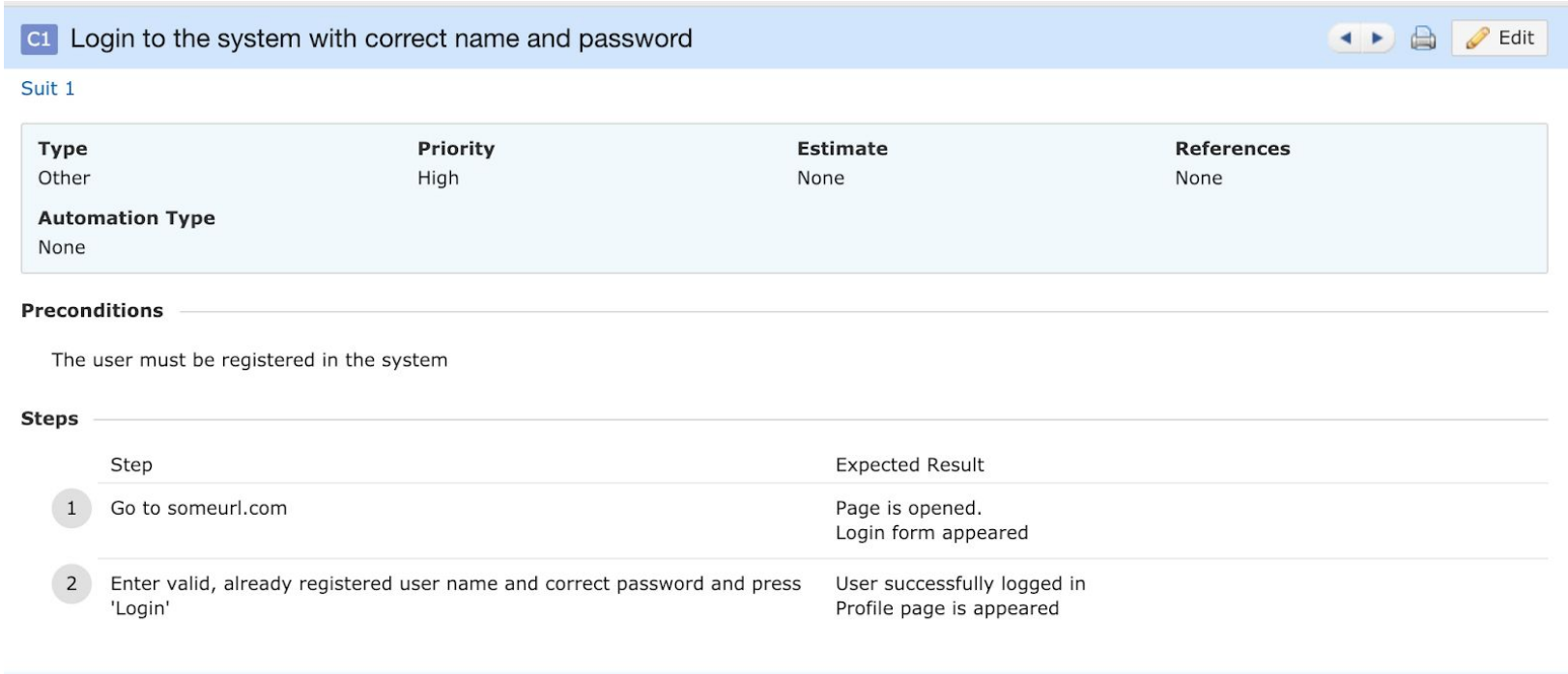
Тестовый случай (Test Case) - это пошаговое описание необходимых действий для получения определенного результата

Характеристики хорошего теста

Хороший тест должен удовлетворять следующим критериям:

- Существует обоснованная вероятность выявления тестом ошибки
- Набор тестов не должен быть избыточным
- Тест должен быть наилучшим в своей категории
- Он не должен быть слишком простым или слишком сложным

Тест кейс. Test Rail



C1 Login to the system with correct name and password

Edit

Suit 1

Type	Priority	Estimate	References
Other	High	None	None

Automation Type
None

Preconditions

The user must be registered in the system

Steps

	Step	Expected Result
1	Go to someurl.com	Page is opened. Login form appeared
2	Enter valid, already registered user name and correct password and press 'Login'	User successfully logged in Profile page is appeared

Тест кейс. SQA Mate

TEST-1: First test

Keywords: `P0`

Description

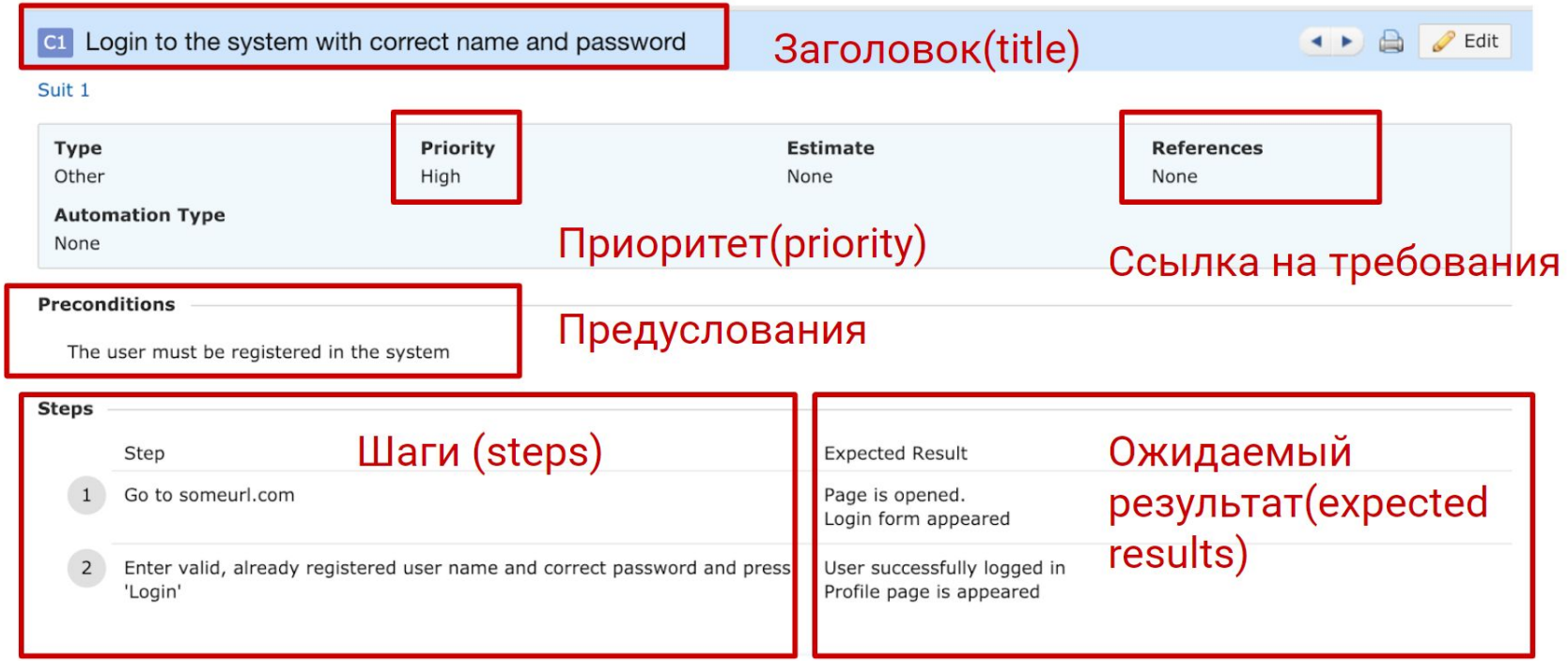
This is the first test to show markup

Preconditions

Some test preconditions. For example - You have registered account, etc.

	Steps	Expected results
1.	Do something in Step 1	Expected results 1
2.	Do something in Step 2	Expected results 2

Тест кейс. Разбор полей



Тестовая система (Test management tool)

Система управления тестами (test management system) - это система, которая позволяет создавать, хранить, структурировать тест кейсы. В ней есть возможность управлять прогонами - выполнением тест кейсов, строить отчеты по результатам прогонов.

Основные функции:

- Тестовый редактор и просмотрщик (создание, удаление, перемещение, редактирование тест кейсов)
- Тестовые запуски (создание запуска, редактирование, удаление)
- Управление отчетами (создание отчетов, возможность обмена)

Тестовая система (Test management tool)

- **Test Link** (http://testlink.org)
- **Test Rail** (https://brunoyamlearn.testrail.io/index.php?/cases/view/1)
- **Hiptest**
- **SQA Mate** (https://sqamate.com/d?q=3&s=13949)
- другие

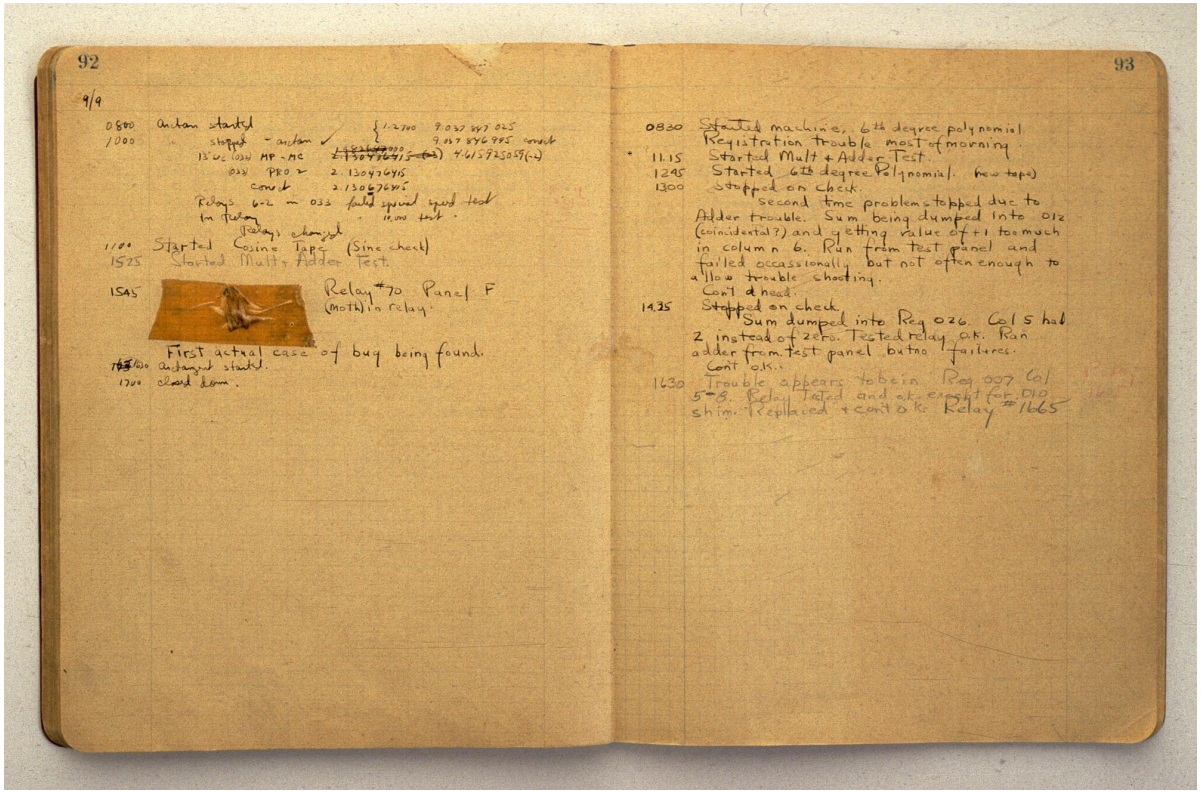
2.3

Баг репорт и баг трекинг системы

Что такое bug?

Bug (Defect) – это отчёт об ошибке, в котором описано некорректное, несоответствующее требованиям поведение программного обеспечения

Первый баг репорт



92

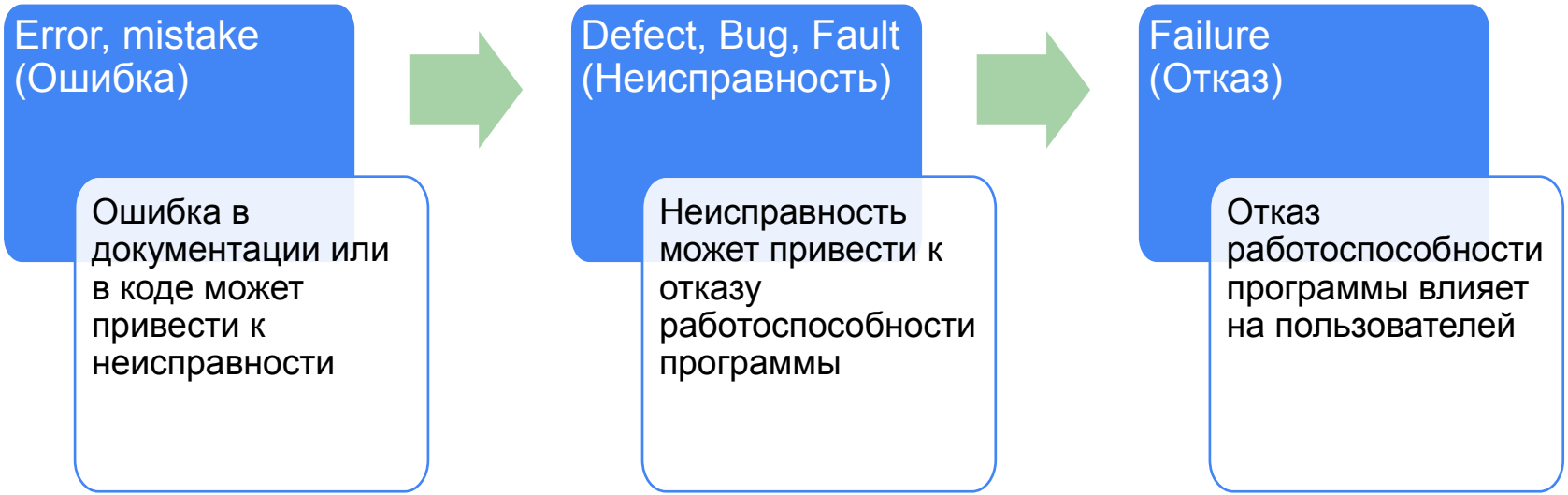
9/9

0800 Arctan started
1000 " stopped - arctan ✓
1300 MP - MC
033 PRO 2
correct
Relays 6-2 in 033 failed special speed test
in relay
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1700 closed down.

93

0830 Started machine, 6th degree polynomial
Registration trouble most of morning.
11.15 Started Mult + Adder Test.
1245 Started 6th degree Polynomial. (new tape)
1300 Stopped on check.
second time problem stopped due to
Adder trouble. Sum being dumped into 012
(coincidental?) and getting value of +1 too much
in column 6. Run from test panel and
failed occassionally but not often enough to
allow trouble shooting.
Cont'd next.
1435 Sum dumped into Reg 026. Col 5 had
2 instead of zero. Tested relay o.k. Ran
adder from test panel but no failures.
Cont o.k.
1630 Trouble appears to be in Reg 007 Col
5-9. Relay Tested and ok except for DID
shim. Replaced + cont o.k. Relay #1665

Bug, Defect, Failure, Mistake, Fault, Error



Структура баг репорт

Баг репорт должен содержать:

- Заголовок (title)
- Версию
- Приоритет/Важность (Priority/Severity)
- Короткое описание проблемы (Problem)
- Предусловия (Preconditions)
- Шаги воспроизведения (Steps to reproduce)
- Ожидаемый результат (Expected results)
- Фактический результат (Actual results)
- Техническую информацию (Technical info)
- Логи (Technical logs)
- Ссылка на тест кейс (Test case link)
- И любую другую информацию, которая поможет баг починить, воспроизвести и в будущем проверить

Priority/Severity

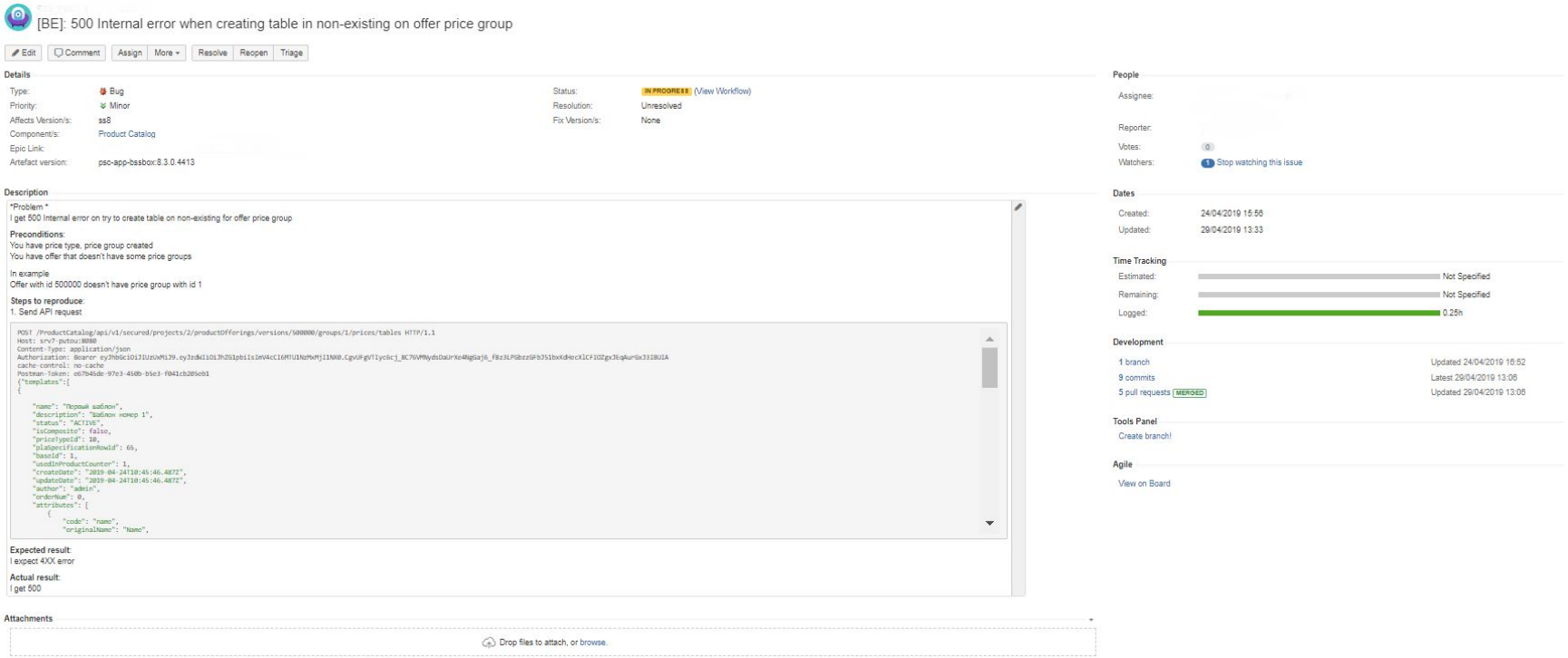
Серьезность (Severity) - это атрибут, характеризующий влияние дефекта на работоспособность приложения

Приоритет (Priority) - это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Можно сказать, что это инструмент менеджера по планированию работ. Чем выше приоритет, тем быстрее нужно исправить дефект

Определите Severity/Priority

- На главной веб странице сделана опечатка
- Приложение падает, если пользователь введет отрицательное число в поле возраст
- Приложение падает, когда пользователь вводит данные в поле, а потом их редактирует
- В логах приложения пишется сообщение с кодом ошибка, в то время когда в реальности ошибки нет(неправильный лог левел)
- Приложение перестает писать логи
- В интернет магазине нельзя добавить в корзину определенную группу товаров
- В интернет магазине невозможно произвести оплату

Структура баг репорт



[BE]: 500 Internal error when creating table in non-existing on offer price group

Edit | Comment | Assign | More | Resolve | Reopen | Triage

Details

Type:	Bug	Status:	IN PROGRESS (View Workflow)
Priority:	Minor	Resolution:	Unresolved
Affects Version/s:	ss8	Fix Version/s:	None
Component/s:	Product Catalog		
Epic Link:			
Artefact version:	psc-app-bssbox:8.3.0.4413		

Description

*Problem *
I get 500 Internal error on try to create table on non-existing for offer price group

Preconditions:
You have price type, price group created
You have offer that doesn't have some price groups

In example
Offer with id 500000 doesn't have price group with id 1

Steps to reproduce:
1. Send API request

```
POST /ProductCatalog/api/v1/secured/projects/2/productOfferings/versions/500000/groups/1/prices/tables HTTP/1.1
Host: srv7-putou:8080
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzUxMi19.eyJzdWIiOiJhZG1pbiIsImV4cCI6NTU1NzMxMjI1N08.CgvUFgVTiyc6cj_8C76VMNydsDaUrXe4NgGaj6_F8z3LPGbzzGFbJ51bxKdHecXlCFIOZgxJEqAur0xJ3IBUIA
cache-control: no-cache
Postman-Token: c67b45de-97e3-450b-b5e3-f041cb205cb1
{"templates":[
{

    "name": "Первый шаблон",
    "description": "Шаблон номер 1",
    "status": "ACTIVE",
    "isComposite": false,
    "priceTypeId": 10,
    "plaSpecificationRowId": 65,
    "baseId": 1,
    "usedInProductCounter": 1,
    "createDate": "2019-04-24T10:45:46.487Z",
    "updateDate": "2019-04-24T10:45:46.487Z",
    "author": "admin",
    "orderNum": 0,
    "attributes": [
      {
          "code": "name",
          "originalName": "Name",
```

Expected result:
I expect 4XX error

Actual result:
I get 500

Attachments

Drop files to attach, or browse.

People

Assignee:
Reporter:
Votes: 0
Watchers: 1 Stop watching this issue

Dates

Created: 24/04/2019 15:56
Updated: 29/04/2019 13:33

Time Tracking

Estimated: Not Specified
Remaining: Not Specified
Logged: 0.25h

Development

1 branch — Updated 24/04/2019 16:52
9 commits — Latest 29/04/2019 13:06
5 pull requests MERGED — Updated 29/04/2019 13:06

Tools Panel

Create branch!

Agile

View on Board

Баг трекинговые системы (Bug tracking systems)

Система управления багами (Bug tracking system) – система, позволяющая управлять багами – создавать, редактировать, удалять, отслеживать их состояние, оценивать качество продукта

Инструменты:

- Jira
- BugZilla
- Redmine
- другие

2.4

Тест репорт

Структура тест репорта

Testing scope - какие тесты и на каких конфигурациях были сделаны
Versions - на какой версии\ях продукта
Environment - в каком окружении
Link to executions - ссылки на отчеты в тестовых системах или на прогнанные автотесты

- Manual
- Automation

Known issues - проблемы, которые были зафиксированы, но было принято решение выпускать продукт с ними
Approvals

2.5

Матрица соответствия требованиям

Traceability matrix

Traceability matrix (или Матрица соответствия требованиям) - матрица (таблица) соответствия тестов требованиям, их покрытию.

Матрица соответствия требований используется QA-инженерами для валидации покрытия требований по продукту тестами.

Цель «Traceability Matrix» состоит в том, чтобы выяснить:
какие требования «покрыты» тестами, а какие нет, есть ли избыточность тестов (когда одно функциональное требование покрыто большим количеством тестов) или недостаточность (когда требование не покрыто тестами).

Traceability matrix. Пример

Requirements Traceability Matrix

Requirements Traceability Matrix				Root Folder: Contract processing	Requirement	Agree on	Check	Create contact	Determine	See customer off	Send contact	Sign contact	Determine net price	Inform customer	Send original	Contract processing	Check if	Develop proposal	Explain contact	Quotation	Sales order
					#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		Total		Req		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Root Folder: Modeling				Covered		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Test	#	Test			Relate	4	4	4	4	4	4	4	2	2	2	1	0	0	0	0	0
Contact processing - path 2	1	1	X		10	X	X	X					X	X							
Contact processing - path 1	2	1	X		8				X	X	X	X									
Agree on	3	1	X		2	X															
Check	4	1	X		2		X														
Create contact	5	1	X		2			X													
Determine	6	1	X		2				X												
See customer off	7	1	X		2					X											
Send contact	8	1	X		2						X										
Send original	9	1	X		2										X						
Sign contact	10	1	X		2							X									
Contact processing	11	1	X		1											X					

2.6

Чеклисты

Чеклисты

Чеклист (checklist), **или контрольный список -** это документ, описывающий что должно быть протестировано. При этом чек-лист может быть абсолютно разного уровня детализации. На сколько детальным будет чек-лист зависит от требований к отчётности, уровня знания продукта сотрудниками и сложности продукта

Что должно быть в чек-листе?

- Список проверок (с требуемой степенью детализации)
- Статус проверок:
- Сборка, на которой проводилось тестирование
- Тестовое окружение (если применимо)
- Кто проводил тестирование
- Результат проверки

2.7

Автоматические тесты

Автоматические тесты

Автоматические тесты – это описанные программным кодом шаги и ожидаемые результаты этих шагов, которые выполняются в автоматическом режиме.

Рисунок 3.1.а — Соотношение времени разработки и выполнения тест-кейсов в ручном и автоматизированном тестировании

Автоматическое (automation) и ручное (manual) тестирование

- Ручное и автоматическое тестирование - единый процесс
- Нет смысла автоматизировать, если нет порядка в тестовой базе
- Тестировщик всегда должен оставаться тестировщиком
- Баг в тестах никто не увидит - Баг в продукте – да

Советы на будущее

Советы

- Оптимизируйте тест дизайн (сокращение за счёт техник тест дизайна, за счёт области использования программы)
- Изучайте систему, которую тестируете (её область применения, аналоги, особенности)
- Изучайте технологии (протоколы, операционные системы, системы развертывания и тд.)
- Советуйтесь с разработчиками (спрашивайте реализацию, слабые места, показывайте свои тесты)
- Используйте автоматизацию в ручном тестировании (скрипты для заполнения, подготовки данных и др)
- Автоматизируйте важное
- Не гоняйте все авто(ручные) тесты. Чтобы этого достичь необходимо:
    - Приоритизироать тесты, делить их на области, функции
    - Выносить one-time тесты отдельно от всего

Вопросы?

Ресурсы

1. ISTQB Foundations,
   https://www.istqb.org/downloads/send/51-ctfl2018/208-ctfl-2018-syllabus.html
   2. IEEE 610.12-1990,
   http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit..
3. QA fundamental terminology,
   http://softwaretestingfundamentals.com/
4. Святослав Куликов, Тестирование ПО. Базовый курс,
   https://svyatoslav.biz/software_testing_book_download/
5. Вики словарь тестировщика
   http://wiki.software-testing.ru

Спасибо!