## 1. Я хочу указывать причину падения теста

**User:**

Тестировщик, анализирующий результаты автотестов

**User needs:**

При анализе отличать причины сбоев в автотестах, понимать с чем связан сбой

Расширение возможностей анализа автотестов поможет составить конкуренцию продуктам типа allure, заменить репортеры автотестов на Test IT.

**What should be done:** автотестов - с продуктом, самим тестом или инфраструктурой

**Business value:**

После получения результатов тестового запуска все результаты этого запуска имеют значение поля Причина дефекта: needs investigation/требуют исследования (Без анализа)

При рассмотрении таких результатов можно выставить категорию проблемы

Классы проблем бывают 3 типов:

Инфраструктурные сбои,

Дефекты продукта,

Дефекты теста

(Без анализа. До этого статуса можно откатиться)

Для каждого из класса проблем есть соответствующая категория проблем:

Класс инфраструктурные сбои - категория "Инфраструктурный сбой",

Класс дефекты продукта - категория "Дефект продукта",

Класс дефекты автотеста - категория "Дефект автотеста".

Разбиение на классы предполагается для того чтобы добавить возможность расширения категорий для каждого класса проблем. (Отдельная US TES-7327)

\*UC:\*

1) В разделе запуски могу открыть тест-ран на просмотр с полученными результатами

2) (В таблице) В результатах без анализа указана "Причина падения" - "Требует исследования" (выделены цветом или значком)

3) При открытии результата автотеста требующего исследования могу видеть детальную информацию по автотесту, а также имею возможность сменить его причину падения.

4) Выбираю причину падения из доступных категорий проблем (по умолчанию 3 категории, но в будущем можно будет кастомизировать) указать ссылку на дефект и комментарий.

5) После того как причина падения для результата выбрана он не отображается как требующий исследования.

**Definition of Done:**

1) У результатов автотестов появилась дополнительная возможность указать причину (как из UI, так и при отправке результата)

2) При открытии результата автотестов из истории есть возможность установить категорию ошибки, добавить комментарий и ссылку на дефект

3) У результатов автотестов по умолчанию отображается что результат не проанализирован

4) По умолчанию можно выбрать одну из трех категорий проблем для результата Инфраструктурный сбой, Дефект продукта, Дефект теста

5) Можно изменить причину падения для уже проанализированных тестов

Доп. комменты:

Аттачи будут в отдельной вкладке (как на макете), на первой вкладке не будет кнопки.

Права которые нужны для анализа результата - Test Execution - редактирование (поддерживаем это на фронте)

Если у пользователя недостаточно прав, то будет форма read only

## 2. Хочу переносить тест-кейсы с одного инстанса на другой

Как пользователь, управляющий тестированием на проектах и использующий разные инстансы Test IT, я хочу переносить тест-кейсы с одного инстанса на другой.

**Пользователь:**

Администратор Test IT, координатор/руководитель проектов, Тест-менеджер.

**Проблема пользователя:**

Существует два инстанса Test IT не синхронизированные между собой. Есть потребность частично или полностью перенести тесты одного и того же проекта с одного инстанса на другой, с возможностью просмотра что было изменено.

**Бизнес ценность:**

Возможность импорта/экспорта тестов в собственном формате, возможность актуализации тестов с помощью импорта/экспорта.

**What should be done:**

1) Реализовать возможность выполнения импорта и экспорта тестов в проекте, с сохранением структуры секций, а также содержащих названия, описания, системные и пользовательские атрибуты, пред-, пост- условия тестов и секций, и общие шаги.

2) Реализовать возможность при экспорте проекта выбирать какие секции и тесты конкретно экспортировать

3) Реализовать возможность актуализации тестов с помощью импорта, с отображением изменений в журнале изменения тестов.

4) Не поломать историю прохождения для тестов, которые актуализированы с помощью импорта.

**Use case 1:**

1) Воспользоваться API для экспорта тестов на инстансе 1, указать проект целиком, или конкретные секции или конкретные тесты для выгрузки

2) Получить выгрузку

3) Воспользоваться API для импорта тестов на инстансе 2, отправить полученную выгрузку

4) На инстансе 2 появился проект с  секциями и тестами, которые были экспортированы, с сохранением структуры секций. В тест-кейсах содержатся, названия, описания  системные и пользовательские атрибуты, пред-, пост- условия тестов и секций,ссылки.и общие шаги.

**Use case 2:**

1) Воспользоваться API для экспорта тестов на инстансе 1, указать проект целиком, или конкретные секции или конкретные тесты для выгрузки

2) Получить выгрузку

3) Воспользоваться API для импорта тестов на инстансе 2, отправить полученную выгрузку

4) На инстансе 2 появился проект с  секциями и тестами, которые были экспортированы

5) Изменить тесты на инстансе 1

6) Воспользоваться API для экспорта тестов на инстансе 1

7) Воспользоваться API для импорта тестов на инстансе 2, отправить новую полученную выгрузку

8) На инстансе 2 обновились те тесты, которые были изменены на инстансе 1, запись об изменении посредством импорта отображается в журнале изменений.

**Use case 3:**

1) Воспользоваться API для экспорта тестов на инстансе 1, указать проект целиком, или конкретные секции или конкретные тесты для выгрузки

2) Получить выгрузку

3) Воспользоваться API для импорта тестов на инстансе 2, отправить полученную выгрузку

4) На инстансе 2 появился проект с секциями и тестами, которые были экспортированы, с сохранением структуры секций. В тест-кейсах содержатся, названия, описания  системные и пользовательские атрибуты, пред-, пост- условия тестов и секций, ссылки и общие шаги.

5) Создать тест-планы с тестами на инстансе 2, пройти их.

6) Изменить тесты на инстансе 1

7) Воспользоваться API для экспорта тестов на инстансе 1

8) Воспользоваться API для импорта тестов на инстансе 2, отправить новую полученную выгрузку

9) На инстансе 2 обновились те тесты, которые были изменены на инстансе 1, запись об изменении посредством импорта отображается в журнале изменений. *История прохождения не перетерлась, история прохождения у этих тестов соответствует той что была на инстансе 2*

 \*Definition of “Done”\*:

1) Есть возможность экспортировать тесты с помощью API

2) При экспорте тестов есть возможность выбрать проект целиком, конкретные секции проекта или конкретные тесты для выгрузки

3) При экспорте тестов пользователь получает выгрузку в JSON формате в файле. Файл имеет следующее название: Test IT - {имя\_проекта} - {date}

4) Есть возможность импортировать тесты с помощью API

5) При импорте тестов сохраняется структура секций в проекте, названия, описания тестов, системные и пользовательские атрибуты,  пред-, пост- условия тестов и секций, и общие шаги, ссылки.

6) При повторном импорте тестов они сопоставляются по ID (даже если он удалён) и обновляются на новую версию, с записью в журнале изменений.

7) При импорте тестов сохраняется история прохождения, на тех версиях на которых были пройдены тесты в рамках данного инстанса (история прохождения не переносится и не перезатирается при импорте/экспорте)

8) Для импорта используем файл с JSON'ом, который был получен при экспорте

9) Перенос осуществляется с одной версии в ту же версию с которой сделана выгрузка, импорт/экспорт поддерживается начиная с Aquila

## 3. Хочу чтобы бага из результата автоматически линковалась со связанными issue

**User personas**:

QA

**User problem**:

При прикреплении баги к результату по тест-кейсу, связанному с US, пользователю приходится вручную линковать этот баг к US

**Business value**:

Расширить возможности интеграции с Jira

**What should be done**:

1. При заведении бага в Jira из тест-поинта, необходимо автоматически прикреплять эту ссылку на Jira-баг и к тест-кейсу и к связанной issue.

**Use case 1**:

1. Ставлю тест кейсу статус Failed
2. Нажимаю на кнопку Save and Create Bug
3. Ссылка автоматически добавляется и в результат тест-кейса, и в связанную Jira-сторю.

**Use case 2**:

1. Ставлю тест кейсу статус Failed
2. Добавляем ссылку на Jira с типом дефект
3. Сохраняем результат
4. Ссылка добавилась в связанную с кейсом issue

**Definition of “Done”**:

1. Автоматическое прикрепление данных при добавлении ссылки
2. Ссылки прикрепленные в ходе выполнения тест-кейса также линкуются с issue в Jira и в юзер сторю слинкованную с тест-кейсом

## 4. Необходимо установить сквозную интеграцию таска в Jira из Test IT

**User personas**:  
QA  
**User problem**:  
Необходима сквозная интеграция между US и тест-кейсом  
**Business value**:   
Повышение - UX  
**What should be done**:

1. Настроить сквозную интеграцию Jira - Test IT, для автоматических добавлений обратных ссылок

**Page layout**:

**Use case 1**:

1. В тест-кейс добавляю ссылку на таску в jira(issue);
2. Открываю таск(issue) в jira;
3. В issue присутствует ссылка на дубликат тест-кейса в Jira
4. В issue присутствует ссылка на тест-кейс в Test IT

**Definition of “Done”**:

Автоматическое добавление обратных ссылок в Jira реализовано.

Учесть возможность включения/отключения.

## 5. Bar chart  "Distribution of launch results"

**User:**

Testing automation engineer.

**User needs:**

to keep track all of Test Runs in one place.

**What should be done:**

Implement bar chart with the ratio of results by test-runs, which are filtered and displayed in the list with percent stacked bar.

The components of the percentage stacked bar are the same data that is displayed in the chart bar.

**DoD:**

1) In the Autotests section you can move to launches.

2) In the tests section there is a bar chart with the ratio of results by test-runs.

3) The bar chart displays the titles of test runs and their results, according to the filter.

4) If list is a large, scrolling is available.

5) Clicking on the bar chart opens launch card

## 6. List of launches in the autotests section

User: Testing automation engineer

**User needs:**

to keep track all of Test Runs in one place.

**What should be done:**

The user must provide a sorted list with rich filtering and grouping capabilities of all project Test Runs. Filtering should include the following items:

* period of the creation the Test-Run;
* status of the Test-Run;

Test-Run primary information should contain:

* title;
* status;
* number of autotests involved in launching; (According to the layout)
* percentage of autotest results sorted by status (Failed/Blocked/Skipped/Passed) (According to the layout)

**DoD:**

1) Opportunity to switch to launches in the autotests section.

2) A table with Tests-Run appeared in the launches section.

3) Opportunity to filter by period, status

4) The table provides the title, status, number of autotests in the execution, the ratio of results. (According to the layout)

5) Test runs sorted by start time from latest to earliest

**Дополнительно:**

Было принято решение не выводить на график с тест-ранами результаты Pending

## 7. I want the changes to be highlighted in the WI change log.

**User personas**:

Testing engineer, Test designer

**User problem**:

When viewing the WI change log it is not obvious what changes were made.

**Business value**:

Improving the usability of the WI change log.

**What should be done**:

1. Currently 2 arrays are shown: how it was and how it became. You need to additionally highlight the difference.

|  |  |
| --- | --- |
| Was | Became |
| Интеллект ментально преобразует естественный закон внешнего мира. Бабувизм транспонирует естественный гений. Сомнение, как следует из вышесказанного, осмысленно создает даосизм, изменяя привычную реальность. Единственной космической субстанцией ~~Гумбольдт~~ считал материю, наделенную внутренней активностью, несмотря на это гедонизм ментально заполняет трансцендентальный гравитационный **парадокс**. Такие дела | Интеллект ментально преобразует естественный закон внешнего мира. Бабувизм транспонирует естественный гений. Сомнение, как следует из вышесказанного, осмысленно создает даосизм, изменяя привычную реальность. Единственной космической субстанцией Менделеев считал материю, наделенную внутренней активностью, несмотря на это гедонизм ментально заполняет трансцендентальный гравитационный парадокс. **Такие дела**. |

**Definition of “Done”**:

1. The change log highlights the differences between "Was" and " Became".(must compare fields:title, description, steps, preconditionSteps, postconditionSteps)
2. Deleted items are highlighted in **red** and **crossed out**.
3. The added elements are highlighted in **green**.
4. Changes in formatting are not displayed in the change log.

## 8. I want to open specific test run details from the test runs list

**User personas**:

Testing automation engineer

**User problem**:

Users Would like to view test-run Analytics.

**Business value**:

Convenient test Analytics will help us compete with solutions for automation engineer.(for example: Allure)

**What should be done**:

1. When opened from the execution list, open the test-run view.
2. Inside the test-run, the results of autotests for namespaces and classes are presented
3. A summary statistics of the test run and results
4. The results table contains columns: Autotest, configuration, result, message (comment), duration, links.

**Definition of “Done”**:

1. When opening from the list of launches, opens launch-card(TestRun).
2. Launch-card contains: summary statistics of the autotest execution and results table of the autotests
3. The table contains columns: Autotest, configuration, result, message (comment), duration, links.
4. Autotests in the table are grouped namespaces and classes.
5. By column - configuration, result, filtering is available.