



Тестирование программного обеспечения с Testopia

Greg Hendricks

gregaryh@gmail.com

Перевод:

Андрей Сатарин

asatarin@yandex.ru

Виталий Филиппов

vitalif@mail.ru

Стас Фомин

stanislav.fomin@gmail.com

<http://mozilla.org/projects/testopia>

Оглавление

Введение 5

Testopia – Система управления тестами 5

Testopia и Bugzilla 5

The Mozilla Foundation и проект Testopia 5

Установка 6

Первый взгляд на требования 6

То, что Вам необходимо знать 6

Получение и установка 7

Обновление 8

Конфигурация 8

Использование Testopia 8

Зачем нужно тестирование? 8

Для какого тестирования вам поможет Testopia? 10

 Тестирование черного ящика 10

 Прозрачный ящик и автоматизированное тестирование 10

Начало работы 11

Крупным планом 11

Планы тестирования (Test Plans) 11

Тесты (Test Cases) 12

Прогоны тестов (Test Runs) 12

Окружения запуска тестов (Test Run Environments) 12

Сборки (Builds) 12

Прогоны отдельных тестов (Test Case-Runs) 12

Тестирование в Testopia 13

Панель продукта 13

Добавление категорий и сборок 13

Начните с плана 13

- Просмотр плана тестирования 14

- Файловые вложения 14

- Редактирование полей плана 14

- Просмотр истории плана 14

Создание тестов 15

- По шагам: Создание тестов 15

- Добавление и удаление компонентов и тегов 15

- Просмотр результатов запуска тестов 16

- Вложение файлов 16

- Связывание с багами 16

- Редактирование полей теста 16

- Зависимости тестов 16

Создание окружений тестирования 16

- Управление окружениями 17

- Категории (Categories) 17

- Элементы (Elements) 17

- Свойства (Properties) 17

- Значения свойств (Property Values) 18

- Создание окружения 18

Создание прогонов тестов 18

- По шагам – Создание прогона тестов 18

- Просмотр прогона тестов 19

- Добавление тестов 19

- Правка параметров прогонов тестов 20

Запуск тестов 20

Выборка тестов прогона по фильтрам 20

Сортировка списка тестов 21

Запись успешных и неуспешных результатов тестов 21

Добавление заметок 21

Прикрепление багов 21

Переназначение тестов 21

Изменение сборки или окружения для отдельного теста 22

Удаление прогонов тестов 22

Изменение нескольких тестов сразу 22

Окончание тестирования 22

Навигация в Testopia 22

Поиск 23

Сортировка результатов поиска 24

Страничная навигация 24

Пакетная обработка 24

Сохранение поисковых запросов 24

Отчёты 24

Отчёты по планам тестирования 25

Отчёты по прогрессу (Build Coverage Reports) 25

Рейтинг багов (Top Bugs) 25

Баги, выявленные в плане (Bugs Found In This Plan) 25

Печатный отчёт по прогрессу в % (Printable Percentage Report) 25

Отчёты по тестам 25

Оценка времени / реальное время (Estimated vs Actual Time) 25

Развёртка по статусам (Historic Status Breakdown) 25

Клонирование 26

Клонирование планов тестирования 26

Клонирование тестов 26

Клонирование прогона 27

Удаление 27

Импорт и экспорт 28

Импорт тестов 28

Testopia и безопасность 28

Группа “Testers” 28

Списки контроля доступа к планам тестирования 28

Регулярное выражения для выбора пользователей 29

Явное включение 29

Права доступа 29

Чтение 29

Запись 30

Удаление 30

Управление 30

Теги (метки) 30

Добавление тегов к объекту 30

Просмотр тегов 31

Testopia и автоматизированные скрипты тестирования 31

Описания полей 31

Планы тестирования 31

Тесты 32

Прогоны тестов 34

Прогоны отдельных тестов 35

Глоссарий 37

Дополнительная информация 40

Списки рассылки 40

Комната IRC 40

Wiki 40

Страница проекта 40

Отправка багов 40

Список литературы 41

Введение

Тестирование, особенно тестирование программного обеспечения, штука скучная и мучительная. Особенно, если учесть, что в жизненном цикле проекта приходится делать множество итераций тестирования многоуровневых требований. Учет результатов выполнения (прогона) тестов важен и заказчику, и разработчику. Заказчикам важно знать, насколько выполнены их требования, а разработчикам — видеть, не упустили ли они какой баг. Тестирование — неотъемлемая часть процесса разработки ПО, ибо «Софт должен быть предсказуем и вменяем, не пугая пользователей сюрпризами» (The Art of Software Testing by Glenford J. Myers, Chapter 1).

Testopia – Система управления тестами

Управление тестами — это процесс регистрации результатов прогонов наборов тестов для заданного набора программно-аппаратных сред и для различных версий ПО. Чтобы составлять и выполнять сценарии тестирования, требуется выделенная оргструктура. Testopia обеспечивает центральный репозиторий тестов и систему управления тестами, пригодную для совместной работы распределенных тестировщиков. Testopia должна удовлетворять потребности тестировщиков ПО команд и организаций любого размера.

Более того — хоть Testopia предназначена главным образом для тестирования программного обеспечения, ее можно использовать для ведения тестирования чего угодно. Поскольку Testopia является системой с открытым исходным кодом (open-source), её легко адаптировать под любую принятую модель тестирования.

Testopia и Bugzilla

Разработанная организацией Mozilla система учета ошибок Bugzilla является одной из наиболее популярных систем учета проблем. Bugzilla — «удобное в использовании и администрировании, выгодное решение с богатым функционалом, на равных конкурирующая с коммерческими альтернативами» («Open-Source Bug Tracking with Bugzilla», <http://www.linuxjournal.com/article/7216>).

Тесты (тест-кейсы) тесно связаны с дефектами ПО. Как только найден какой-либо дефект, должны быть написаны тесты, проверяющие, что этот дефект больше не возникнет в последующих версиях. Поэтому Testopia разработана как расширение к Bugzilla, что дает единое управление проектом и в отслеживании дефектов и в ведении тестов.

The Mozilla Foundation и проект Testopia

Testopia выросла из системы Bugzilla Testrunner. Она была создана Maciej Maczynski в 2001 году, а потом была передана сначала Ed Fuentetaja, а потом, в 2006 году, Greg Hendricks. Название “Testopia” она получила в марте 2006 года, перед выпуском версии 1.0 в мае того же года. Testopia была принята в качестве проекта Mozilla и сейчас находится по адресу <http://mozilla.org/projects/testopia>.

С мая 2006 года Testopia получала всё большую и большую популярность среди тестировщиков как в мире открытого программного обеспечения, так и в других организациях. В недавней записи в своём блоге Rosie Sherry, уважаемый эксперт в тестировании ПО, упоминает, что «вероятно, это [Testopia] наиболее универсальная система управления тестированием, и будучи установленной и настроенной, она предоставляет всё, что нужно от системы управления тестами» (Sherry, 2006).

Установка

Первый взгляд на требования

Рекомендованная версия Bugzilla и где её взять

Bugzilla 3.2.x <http://bugzilla.org/downloads>

СУБД

MySQL 5.0 <http://mysql.com>

Дополнительные модули Perl

JSON 1.14* <http://search.cpan.org/~makamaka/JSON-1.14/>

Text::Diff <http://cpan.org>

Библиотеки

Ext JS toolkit 2.0.1 <http://extjs.com>

Браузер

Firefox 2.0+ <http://getfirefox.com>

* Версия 2.01 модуля – несовместима! См. [bar 409309](#)

То, что Вам необходимо знать

Так как наша разработка зашла далеко вперёд, мы решили попробовать держаться наравне с последней стабильной версией Bugzilla (на данный момент — 3.2.2). Это даёт нам стабильную кодовую базу для работы, потому что вообще-то разработка плагинов/расширений для любых программ похожа на стрельбу по движущейся цели. Так что наше решение даёт нам возможность сфокусироваться на добавлении новых возможностей, а не на переносе старых в новые версии Bugzilla. С другой стороны, это же означает, что наиболее важные возможности не будут появляться в новых версиях до тех пор, пока у нас не появится времени на их перенос. Поэтому мы будем более чем рады всем желающим помочь нам в данном направлении.

Хотя Bugzilla официально поддерживает и СУБД MySQL, и PostgreSQL, Testopia полностью тестировалась лишь с СУБД MySQL. Опять-таки, мы будем рады любым патчам для улучшения совместимости Testopia с PostgreSQL.

Аналогично, Testopia полностью тестировалась только с браузерами семейства Firefox – и это не потому, что другие браузеры нас не волнуют, а потому, что ни времени, ни ресурсов для организации полноценной поддержки и тестирования Testopia в других браузерах у нас нет. Между тем, теоретически в современных браузерах Testopia должна работать нормально; если это всё-таки окажется не так – присылайте, пожалуйста, сообщения об ошибках на bugzilla.mozilla.org.

Testopia использует некоторые передовые AJAX-технологии, предоставляемые библиотекой Ext Javascript. Наша философия заключается в том, чтобы перенести большую часть обработки данных на клиента, а не завязывать на сервер. Библиотека Ext добавляет веса клиентскому приложению, но зато делает более быстрым взаимодействие с сервером. Заодно это повышает и нагрузку на клиентскую машину, что делает необходимыми чуть большую процессорную мощность и больший объём оперативной памяти – работа с Testopia на машине с Pentium 2 и 128 Мб ОЗУ, вероятно, будет не очень приятным экспериментом.

Получение и установка

Testopia – расширение к Bugzilla. Поэтому, очевидно, у Вас в первую очередь должна быть установлена Bugzilla – дистрибутивы и инструкции по её установке можно найти на сайте <http://bugzilla.org>. Обычно всё, что вам нужно – скачать архив, распаковать его, запустить checksetup.pl и следовать инструкциям.

Как только Вы успешно установите Bugzilla, для установки Testopia Вам нужно будет предпринять следующие дополнительные шаги:

1. Скачать архив Testopia с <ftp://ftp.mozilla.org>;
2. Скопировать его в директорию установки Bugzilla;
3. Распаковать:

```
tar xzvf testopia-<release>.tar.gz
```

4. Применить патч:

```
patch -p0 -i testopia/patch-3.0
```

or

```
patch -p0 -i testopia/patch-3.0.x
```

or

```
patch -p0 -i testopia/patch-3.2
```

5. Запустить checksetup:

```
./checksetup.pl
```

Всё это подразумевает установку Testopia на свежую версию Bugzilla на Linux/UNIX системах. Если вам необходимы более детальные инструкции по установке, или инструкции по установке Testopia под другими операционными системами, например, Windows – смотрите файл README в директории testopia скачанного архива.

После установки Вы должны увидеть ссылки на действия Testopia в нижней части страницы Bugzilla после входа в оную. Кстати, добавьте всех пользователей, которые должны иметь доступ к тестам, в группу Testers.

Обновление

Для обновления уже установленной Testopia Вы можете точно так же следовать вышеописанным инструкциям, предварительно отменив патч, который применяли ранее. Также Вы можете обновить Testopia до последней CVS-версии, запустив скрипт `cvs_update.sh` в директории `testopia/scripts`. Более детальные инструкции смотрите в файле `README`.

Обновление установленного Testrunner (версии 0.7 и ниже) не поддерживается.

Конфигурация

Testopia добавляет некоторое количество параметров на страницу Bugzilla «Parameters». Они находятся на вкладке Testopia. Кстати, если вы устанавливаете Testopia на только что установленную Bugzilla, не забудьте задать параметр **urlbase**. Ниже приведён список параметров Testopia с описаниями:

allow-test-deletion

Если эта опция включена, пользователи Testopia могут удалять объекты тестирования, включая планы тестирования и сами тесты; права на удаление поддерживаются в списках контроля доступа планов тестирования.

testopia-allow-group-member-deletes

Если эта опция включена, члены группы Testers смогут удалять объекты тестирования; эта настройка имеет больший приоритет, чем индивидуальные.

testopia-default-plan-testers-regexp

Регулярное выражение для выбора пользователей, которым по умолчанию даётся доступ к новым планам тестирования.

default-test-case-status

Статус новых тестов по умолчанию.

testopia-max-allowed-plan-testers

Ограничение количества пользователей, соответствующих регулярному выражению контроля доступа плана тестирования.

Использование Testopia

Зачем нужно тестирование?

Следующий пример показывает, почему тестирование ПО так важно:

Взято с <http://techiecorner.blogspot.com> .

В марте 1992 человек, живущий в г. Ньютаун, рядом с Бостоном штат

Массачусетс, получил счет по своей еще не использованной кредитной карте, в счете указывалась сумма в \$0.00. Он проигнорировал его и выкинул.

В апреле он получил еще один счет такой счет и тоже его выкинул.

В следующем месяце, кредитная компания послала ему очень недоброжелательное послание в котором сообщалось, что они аннулируют его кредитную карту если он не вышлет им \$0.00 с ответом. Он позвонил им и разъяснил ситуацию, ему сообщили что это компьютерная ошибка и сказали, что они с ней разберутся.

В следующем месяце наш герой решил, что пора разобраться с беспокойной картой, полагая что если на его счете будут покупки, то это положит конец смешному затруднению. Тем не менее, в первом магазине, в котором он воспользовался своей кредитной картой для оплаты покупки, он обнаружил, что его кредитная карта была аннулирована.

Он позвонил в кредитную компанию, где снова извинились за компьютерную ошибку и сказали что они с этим разберутся. На следующий день он получил счет на \$0.00, в котором говорилось, что теперь его долг просрочен.

Подразумевая, что он говорил с представителем кредитной компании только вчера, а счет был еще одной ошибкой, он проигнорировал его, веря, что компания сдержит обещание и разберется с проблемой.

В следующем месяце он получил счет на \$0.00, в котором сообщалось что у него есть 10 дней чтобы погасить долг или компания примет меры для его взимания.

Окончательно сдавшись, он решил играть с компанией в их собственную игру и отправил им чек на \$0.00. Компьютер надлежащим образом обработал его счет и вернул сообщение о том что теперь он ничего не должен кредитной компании.

Неделю спустя, из банка позвонили и спросили его, зачем он послал чек на \$0.00. После длинного объяснения в банке ответили, что чек на \$0.00 привел к ошибке в их программе обработки чеков. Теперь банк не мог обрабатывать чеки всех клиентов, потому что чек на \$0.00 привел к серьезному сбою в их компьютере.

На следующий месяц, человек получил письмо от кредитной компании с претензией что его чек не принят и что теперь он должен им \$0.00 и до тех пор пока он не пошлет к ним чек они будут предпринимать шаги по взиманию долга.

Несмотря на то, что в реальной жизни таких ситуаций не был, этот довольно забавный случай показывает влияние, которое может оказывать плохо протестированное ПО. В случае кредитной компании и банка эта кажущаяся ничтожной проблема могла стоить обоим организациям существенных сумм из-за потерь транзакций и простоя, не говоря уже о неудовлетворенности клиентов.

Во многих случаях ошибки в ПО приводят лишь к небольшим неудобствам. В других случаях они могут иметь серьезные последствия. Подумайте о системе жизнеобеспечения на шаттлах или системе управления полетами на авиалайнере. ПО стало настолько неотъемлемой частью нашей жизни, что во многих случаях мы не осознаем, насколько сильно от него зависим, до тех пор, пока что-то не пойдет не так.

Для какого тестирования вам поможет Testopia?

Существует две наиболее общих разновидности тестирования ПО: тестирование черного ящика (black box) и тестирование белого (прозрачного) ящика (white box).

Тестирование черного ящика - это стратегия, в которой тестирование основывается исключительно на требованиях и спецификациях. В противоположность своему дополнению, тестированию белого ящика, тестирование черного ящика не требует знания о внутренних путях, структуре или реализации тестируемого ПО (Copeland, 2004, Sec I).

Тестирование черного ящика

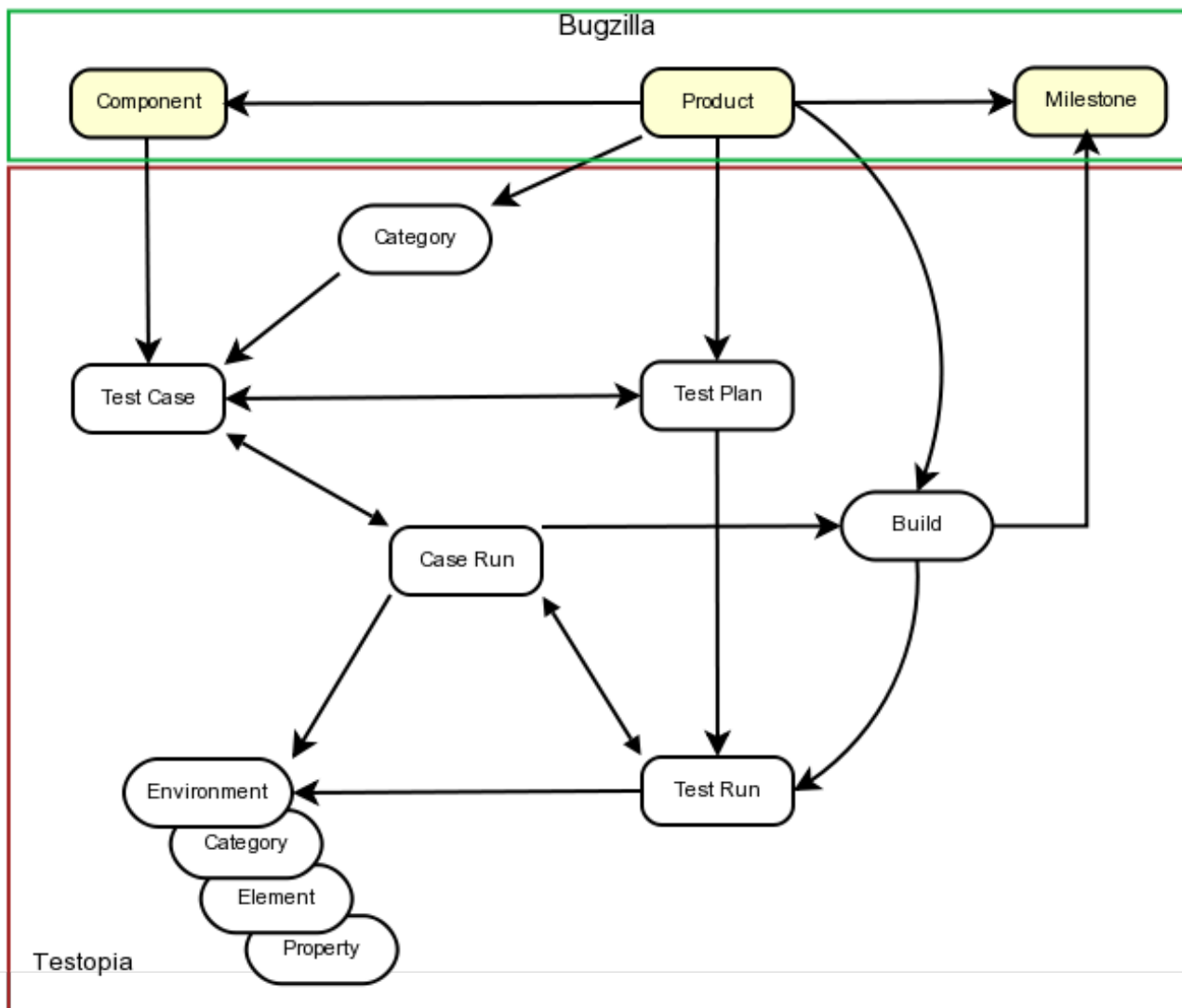
Testopia была создана в основном именно для данного типа тестирования. Требования преобразуются прямо в тесты, которые применяются тестировщиками к разрабатываемому ПО. Если требование не выполняется - тест проваливается, если оно выполняется - тест проходит.

Прозрачный ящик и автоматизированное тестирование

Хотя Testopia не создавалась для работы с тестами белого ящика (которые часто автоматизируются), она может предоставить репозиторий для результатов тестов. Каждый тест может иметь связанный скрипт, и логи тестирования могут быть присоединены к тесту, чтобы сохранять и отображать результаты тестирования.

Начало работы

Крупным планом



Поскольку Testopia является расширением к Bugzilla, она зависит от многих сущностей, которые Bugzilla использует для отслеживания багов. Чтобы использовать Testopia, вам сначала придётся задать продукты и версии, используя редактор продуктов Bugzilla; также Testopia использует функциональность верх (target milestone) Bugzilla — это необязательное поле может быть включено в параметрах.

Testopia оперирует несколькими различными типами взаимозависимых объектов. Все вместе они и позволяют проводить тестирование с помощью Testopia. Давайте по очереди рассмотрим эти объекты.

Планы тестирования (Test Plans)

На вершине иерархии в Testopia находятся планы тестирования. Прежде чем вы сможете сделать что-то в Testopia, вам нужно создать хотя бы один план тестирования. Планы тестирования связаны с одним продуктом в Bugzilla, но вы можете иметь несколько планов для одного продукта. Ваш план тестирования будет служить точкой хранения для всех связанных тестов и прогонов тестов и пультом управления всего вашего процесса тестирования; также он будет служить для определения тех пользователей, которые будут иметь доступ к изменению тестов.

Тесты (Test Cases)

Тесты — сердце процесса тестирования. Тесты разъясняют, какие шаги делать при тестировании и какие результаты ожидать. Если после отдельного запуска теста по шагам ожидаемых результатов не видно, тест проваливается. В Testopia тест может быть связан с многими планами тестирования, поэтому надо соблюдать осторожность и проверять, что изменение теста не мешает тестированию в другом — не вашем — плане тестирования. Это не должно быть слишком сложно, потому что в Testopia список связанных планов тестирования отображается рядом с каждым тестом.

Тесты могут быть разделены между категориями. На страницы плана тестирования вы можете определить столько категорий для вашего продукта, сколько вам нужно. Тем не менее, не надо путать их с компонентами: каждый продукт в Bugzilla разделен на компоненты и каждому тесту вы можете сопоставлять несколько компонентов, но только одну категорию.

Прогон тестов (Test Runs)

Прогон тестов — средоточие результатов тестирования. После определения набора тестов для отдельного прогона тестов вы будете готовы произвести запуск и пройти по этим тестам (хотя для этого потребуется создать ещё пару объектов — см. ниже). Каждый прогон связан с одним планом тестирования может содержать в себе любое число тестов из этого плана.

Окружения запуска тестов (Test Run Environments)

Если тесты в тестировании - это «что», то окружение это «где». Никакие тесты не происходят в вакууме; то, **где** вы запускаете свои тесты, так же важно, как и **как** вы их запускаете. ПО часто создается для работы на специальном оборудовании или в специальных условиях — эти условия и записываются в окружении тестирования. Окружения обычно сопоставляются напрямую запускам тестов, но также, как мы увидим далее, могут неявно применяться и к самим тестам.

Сборки (Builds)

Разработка ПО обычно является итеративным процессом. Разработчики пишут код, который компилируется и включается в систему; потом, когда поступают баги и запросы на улучшение функциональности, разработчики переписывают части кода снова для того, чтобы исправить или улучшить продукт. В Testopia каждая итерация называется «сборкой». Сборки часто связываются с вехами проекта. Это выражается в соотношении сборок и вех Bugzilla; однако, независимо от того, используете ли вы вехи в Bugzilla, вам нужно определить минимум одну сборку для вашего проекта, прежде чем вы сможете начать тестирование.

Прогон отдельных тестов (Test Case-Runs)

Запуск теста является записью об успехах (или неудачах) конкретного теста в конкретном запуске конкретной сборки в конкретном окружении. Когда вы создаете прогон тестов, в нём создаются записи о каждом отдельном тесте. По умолчанию им сопоставляется сборка и окружение, сопоставленные всему прогону, но Вы можете изменить эти атрибуты для прогона каждого отдельного теста, по сути, создавая новый прогон отдельного теста для каждой их комбинации. Желательно делать именно так в ситуации,

когда большинство тестов имеют общее окружение, но некоторые из них могут требовать специальных условий. Также это помогает в итеративном тестировании, поскольку каждый тест может быть изменен в уже существующем прогоне без необходимости создания новых прогонов для проваленных тестов.

Тестирование в Testopia

Панель продукта

В Testopia 2.0, панель продукта — это центральная панель управления Testopia. Отсюда вы можете видеть ваши планы по данному продукту, смотреть и сохранять отчеты, манипулировать сборками и категориями. Панель продукта содержит закладки с планами, тестами, прогонами, сборками, окружениями и категориями. Эти закладки содержат списки (сортируемые). Отдельные элементы списков могут быть отредактированы нажатием на иконку редактирования в панели инструментов над ними, или при помощи двойного клика в поле. Внесённые изменения сохраняются немедленно.

Добавление категорий и сборок

Категории используются для классификации ваших тестов. Каждый продукт имеет категорию по умолчанию; использование категорий для классификации тестов остается на ваше усмотрение. Поскольку также для классификации тестов можно использовать компоненты продуктов, может оказаться, что этого Вам будет достаточно.

Прежде, чем вы сможете начать запускать тесты, вам нужно определить как минимум одну сборку. Для создания сборки или категории нажмите ссылку «Add» которая находится над каждым из этих полей на соответствующей закладке панели продукта. Значения полей формы достаточно очевидны. Тем не менее, в случае создания сборки, у вас есть дополнительная возможность — скрыть сборку (из спискаборок при регистрации нового прогона тестов), сняв галку с поля «Active».

Чтобы отредактировать сборку, кликните дважды на поле в списке. Так вы можете редактировать или удалять отдельные сборки или категории.

Начните с плана

Любое тестирование должно начинаться с плана. В большинстве случаев он имеет вид документа, определяющего цель, структуру и методы, которые будут использованы при тестировании. Существует шаблон планов тестирования, созданный институтом IEEE - IEEE Standard 829-1998 for Software Test Documentation. Но независимо от того, решите вы использовать формальный шаблон (такой как IEEE) или нет, наличие грубого наброска того, что является целью тестирования, поможет тем, кто будет проводить тестирование знать, что вы ждёте от него.

Чтобы создать новый план тестирования в Testopia:

1. Нажмите на ссылку «New Plan» внизу Bugzilla;
2. Введите название этого плана тестирования;
3. Выберите продукт из выпадающего списка продуктов;

4. Выберите тип данного плана (описание типов внизу);
5. Выберите версию продукта. Это будет версия по умолчанию для новых прогонов;
6. Напечатайте или скопируйте текст документа плана в редактор документа;
7. Нажмите на кнопку «Add».

Выполнив эти пункты, Вы перейдете к просмотру только что созданного плана тестирования.

Просмотр плана тестирования

Сверху вашего нового плана вы увидите обзорный раздел, подробно описывающий информацию, введенную при создании плана; в дополнение к этому вы также увидите имя автора плана, время создания плана и версию документа, который вы просматриваете.

Файловые вложения

При или после создания плана вы можете загружать вложения к нему. Для создания вложения просто нажмите кнопку «Обзор» и укажите файл, который вы хотите вложить, впишите описание в поле «Description» и нажмите кнопку «attach».

Вложения обрабатываются также, как и в Bugzilla. Предел размера вложения определяется параметром «*max-attachment-size*», также как и в Bugzilla.

Для редактирования вложения нажмите ссылку «edit» - это позволит вам поменять описание, имя файла, а также mime-тип вложения. Если вложение имеет тип, пригодный для просмотра в браузере, Вы можете просмотреть его, нажав на ссылку «ID».

Редактирование полей плана

В обзорном разделе редактируемые поля имеют иконку карандаша рядом с ними или являются выпадающими списками. Внесённые изменения сохраняются немедленно.

Просмотр истории плана

Любые изменения, сделанные в плане, записываются. Testopia сохраняет полную историю изменений, чтобы в изменения условий тестирования в течении цикла разработки Вы всегда могли знать, что использовалось для указанного набора тестов в прошлом. Также история нужна, чтобы пресечь случайные изменения в вашем плане тестирования — в каждом изменении записывается, когда оно произошло и кто произвел его. Итак, чтобы посмотреть историю, перейдите на закладку «History».

Изменения в документа плана обрабатываются немного не так. Для каждого изменения хранится полная версия, но между любыми версиями документа можно смотреть различия. Для этого нужно выбрать, какие версии вы хотите сравнить и нажать кнопку «View» на странице истории.

Создание тестов

Когда у вас есть план («родина» тестов), Вы можете начать их создавать. Нажмите кнопку «New Case» на странице плана или ссылку «New Case» внизу страницы. Также Вы можете создать тест, сделав правый клик на плане в списке и выбрав пункт меню «Add Test Case».

По шагам: Создание тестов

1. Нажмите «Create New Test Case» на странице плана или «New Case» внизу страницы;
2. Введите краткое описание вашего теста в поле «Summary»;
3. Выберите категорию;
4. Добавьте тестировщика по умолчанию или выберите компонент, для назначения теста ответственному за этот компонент в Bugzilla (QA Contact);
5. Введите шаги тестирования в поле «Action»;
6. Запишите ожидаемые результаты в поле «Expected Results»;
7. Нажмите кнопку «Add».

Поскольку каждый тест можно привязывать к нескольким планам тестирования, Вы можете сразу выбрать, к каким планам следует привязать созданный тест.

Сначала Вы должны написать короткую аннотацию вашего теста в поле «Summary». Обратите внимание на опции для других полей — их описание может быть найдено в конце данного документа. Единственное обязательное поле здесь — аннотация (Summary), но вы можете указать и тестировщика по умолчанию, и требование, а выбор компонента с закладки компонентов автоматически установит тестировщика по умолчанию согласно ответственному за этот компонент в Bugzilla (QA contact).

Хоть на данном шаге это и необязательно, вы должны описать в поле «Action» шаги, а поле «Expected Results» - ожидаемые результаты. Чтобы сделать это, нажмите ссылку «Edit Document» под соответствующим полем — отобразится WYSIWYG редактор, в который Вы сможете ввести или скопировать свою информацию. За информацией о том, как писать хорошие тесты, мы предлагаем обратиться к статье D.L. Runnel [How to Write Better Test Cases](#).

После создания Вы попадаете к своему тесту. Также, наверху, Вы увидите ссылку, позволяющую сразу создать еще один тест в данном плане.

Добавление и удаление компонентов и тегов

Итак, Вам предоставляется обзор, содержащий детальную информацию о тесте, после чего у вас появляется возможность добавить дополнительные компоненты к тесту. Так как тестирование иногда проходит по нескольким продуктам, Вы можете добавить к тесту компонент любого продукта Bugzilla. Вы также можете удалить компонент, выбрав его и нажав кнопку «Remove».

Как и с планами, Вы можете помечать свои тесты тегами. Это будет описано ниже.

Просмотр результатов запуска тестов

При переходе на закладку «Case Run History» будет показана история теста во всех его прогонах и их статус (прошли тесты или были провалены).

Вложение файлов

Как и в планах тестирования, Вы можете совершенно аналогично прикладывать файлы к вашим тестам.

Связывание с багами

В противоположность планам тестирования у вас есть возможность привязать баги Bugzilla к тестам. Каждый тест может иметь несколько привязанных багов. Для того чтобы привязать баг введите номер бага в поле и нажмите кнопку «Add». Будет показана таблица привязанных багов.

Редактирование полей теста

Как и с планами тестирования, Вы можете изменять любые значения, введенные при создании теста. История этих изменений также сохраняется и Вы можете увидеть ее, нажав на кнопку «History» сверху теста.

Зависимости тестов

Тесты имеют понятие зависимости. Часто при тестировании набора тестов порядок, в котором Вы тестируете определяется тестами, проведенными до данного теста. Также часто случается, что провал одного теста делает совершенно невозможным успешное проведение других. Вы можете вводить эти зависимости при помощи полей «blocks» и «depends on». Если тест потенциально блокирует выполнение другого теста, вам следует ввести идентификатор блокируемого теста в поле «blocks»; если данный тест требует успешного выполнения другого теста, вы можете ввести идентификатор блокирующего теста в поле «depends on». Если тест, блокирующий другой тест проваливается, и оба они включены в один запуск, заблокированный тест автоматически получит статус «BLOCKED».

Создание окружений тестирования

Как сказано выше, окружение в процессе тестирования - это **«где»**. Например, окружение может включать информацию об операционной системе и аппаратной платформе, на которой проводится тест. Окружения могут быть широкими или узкими настолько, насколько это Вам необходимо. Наиболее простое окружение состоит из операционной системы и платформы, выбираемой из соответствующих списков Bugzilla. Однако оно может быть гораздо более сложным, включая набор приложений и других продуктов — например, это может быть браузер (или другой клиентский пакет).

Создание окружения в Testopia требует двух шагов — первый включает набор переменных, используемых в окружении, а второй - создание самого окружения из сочетания выборов возможных вариантов этих переменных.

Управление окружениями

Когда вы первый раз устанавливаете Testopia, Вы должны установить множество переменных окружения, которые будут использованы для конструирования окружений. При редактировании окружения переменные окружения упорядочиваются в древовидной иерархии объектов. Существует четыре основных уровня этой иерархии: категории, элементы, свойства и значения свойств.

Категории (Categories)

Категории окружений похожи на категории тестов тем, что они предоставляют механизм сортировки элементов окружений. Каждая категория связана с одним продуктом или специальным сегментом, отмеченным «--ALL--» и обозначающим все продукты, т.е. содержащим все категории, не специфичные для каких-то продуктов.

При установке Testopia Вы увидите, что сегмент «--ALL--» содержит две категории «OS» и «Platform». Раскрыв их, Вы увидите что там есть элементы представляющие каждую операционную систему и платформу определенную в Bugzilla. Этот список генерируется при установке Testopia, а потом поддерживается независимо от списков Bugzilla.

Чтобы создать категорию, кликните правой кнопкой мыши на продукте или сегменте «--ALL--» и выберите «Add Category». Вы можете нажать на только что созданную категорию в дереве — при этом появится форма, которая позволит вам редактировать имя категории и продукт, с которыми она связана.

Элементы (Elements)

Определив категорию для своего продукта или секции «--ALL--», Вы можете добавлять элементы к этой категории. Элементы — это то, из чего состоят ваши окружения — «варианты» полей категорий. Для создания элемента, кликните правой кнопкой мыши на категории, в которой хотите его создать, и выберите пункт меню «create element».

При этом будет создан элемент под названием «New Element», который можно редактировать, нажимая на него в дереве или нажимая правую кнопку и выбирая «Edit».

Элементы могут включать другие элементы. Для того, чтобы создать подэлемент, нажмите правой кнопкой на элемент и выберите «Add Element». Вы можете редактировать элемент-потомок так же, как и элемент-родитель. Вы можете создавать столько уровней вложенности элементов, сколько Вам необходимо для отражения сложности вашего окружения.

Свойства (Properties)

Свойства описывают элементы. Вы можете добавлять свойства к элементам, нажимая правую кнопку мыши на элементе и выбирая «Add Property». Вы можете добавлять сколько угодно свойств к каждому элементу, но свойства не могут быть вложенными.

Чтобы отредактировать свойство, нажмите на него в дереве или кликните правой кнопкой мыши и выберите «Edit».

Значения свойств (Property Values)

Определив свойства элемента, Вам нужно составить список значений, которые может принимать данное свойство в вашем окружении. Для добавления значения свойства кликните правой кнопкой мыши на свойстве и выберите «Add Value». Редактировать значения свойств Вы можете, кликая правой кнопкой мыши и выбирая «Edit», или нажимая на одно из значений внутри вашего свойства.

При помощи специальной формы Вы также можете переупорядочить список значений или изменить их названия. Внесённые изменения сохраняются при нажатии кнопки «Save Changes».

Создание окружения

После того как вы создали элементы, которые будут использованы в вашем окружении, Вы можете создавать и сами окружения.

С закладки окружений (environments) панели продукта нажмите кнопку «Add» в панели инструментов.

Вас спросят о названии и продукте окружения; продукт используется только для классификации и не ограничивает элементы, которые вы можете помещать в окружение.

После создания, для вызова редактора окружения Вы попадёте к редактору окружения. Здесь Вы увидите два дерева — одно представляет ваше окружение, а второе содержит список элементов, из которых Вы можете выбирать. Окружение может состоять из элементов, которые Вы определили ранее — для добавления элемента найдите его в списке и просто перетащите в дерево окружения; порядок элементов не имеет значения. Вы можете брать элементы с любого уровня иерархии, но перенос элемента с потомками перенесет и потомков.

После выбора элементов окружения Вы можете выбирать, какими будут значения свойств этих элементов в окружении. Раскройте элемент и его свойства, и просто кликните на нужное Вам значение. Чтобы отметить ваш выбор, рядом с элементом будет показана звездочка.

Чтобы убрать элемент, кликните на нем правой кнопкой мыши и выберите «Remove».

Внесённые в дерево изменения сохраняются немедленно.

Создание прогонов тестов

По шагам – Создание прогона тестов

1. Нажмите на ссылку [Create a New Test Run](#) на странице плана тестирования или на ссылку [New Run](#) внизу страницы.
2. Выберите тесты для включения в прогон. Вы можете использовать фильтр для выбора тестов и ссылки на страницы для выбора большего или меньшего количества тестов.
3. Введите описание (summary) для создаваемого прогона тестов.

4. Выберите сборку из списка **Build** или введите имя новой сборки для её создания.
5. Выберите окружение тестирования из списка **Environment**.
6. Нажмите на кнопку **Add** (Добавить).

Итак, после создания окружения тестирования, а также одного или нескольких тестов, вы готовы начать тестирование, и начинается оно с создания прогона тестов. Наиболее просто сделать это со страницы плана тестирования: нужно нажать на ссылку [Create a New Test Run](#) (Создать новый прогон тестов) в секции Test Runs (Прогоны тестов). Также вы можете создать прогон, нажав на ссылку [New Run](#) внизу страницы, и введя номер вашего плана тестирования.

Вам будет показан список тестов из вашего плана тестирования, имеющих статус CONFIRMED. Вы можете выбрать как все (нажимая на ссылку «select all»), так и лишь какую-то их часть. Также для ограничения выбора тестами, удовлетворяющими некоторым критериям, Вы можете использовать фильтры. Если у вас очень много тестов, вы также можете использовать страничную навигацию для просмотра за раз более, чем 25 тестов — это важно по той причине, что выбраны будут только видимые на экране тесты, рядом с ID которых установлен флажок. А если Вы не знаете, какие тесты хотите выбрать, или не хотите этого делать именно сейчас, имейте ввиду, что Вы можете добавить тесты в прогон и потом.

Далее, Вам нужно задать описание прогона тестов и выбрать сборку; если окажется, что создать сборку перед созданием прогона Вы забыли, Вы можете просто ввести имя сборки в поле **New** и она будет создана вместе с прогоном. Эту сборку Вы также можете поменять и потом, например, чтобы сопоставить ей веху (milestone) Bugzilla. Наконец, остаётся лишь выбрать окружение тестирования – для этого нужно ввести имя окружения в поле **Environment**. При вводе Вам будет показан список подходящих окружений выбранного продукта; выбирать окружение можно и из этого списка.

После заполнения всех полей, просто нажмите кнопку **Add** – и прогон будет создан.

Просмотр прогона тестов

Аналогично другим объектам, сверху прогона тестов Вы увидите секцию «Overview» (Обзор). Информация, отображаемая там, аналогична таковой в планах тестирования и отдельных тестах, за одним только исключением: полосой прогресса, которая показывает долю количества завершённых тестов с цветовой маркировкой по статусам.

Ниже этой секции находится журнал прогонов отдельных тестов – данная таблица содержит тесты (которые вы и будете тестировать), по строке на каждый тест. Также рядом с каждой строкой таблицы можно заметить стрелки, нажатие на которые откроет форму прогона тестов. Аналогично, рядом находятся стрелки **Filter** и **Report** – фильтры можно использовать точно так же, как и при выборе тестов для включения в прогон; единственное различие – здесь фильтры предлагают больше вариантов для ограничения списка видимых прогонов отдельных тестов.

Добавление тестов

Как уже было упомянуто выше, Вы всегда можете добавлять тесты в прогон – для этого нажмите кнопку **Add Cases** под таблицей прогонов отдельных тестов, и Вам снова будет показан список тестов, аналогичный тому, в котором Вы выбирали тесты при создании прогона; уже добавленные тесты будут исключены из него.

Правка параметров прогонов тестов

Так же как и с тестами, и с планами тестирования, Вы можете всегда изменять значения любых полей прогонов тестов через форму внизу страницы. Не забудьте, что изменение сборки и/или окружения тестирования повлияет не на тесты, уже добавленные в прогон, а только на вновь добавляемые тесты. Все внесённые изменения сохраняются в истории изменений, которая поддерживается совершенно аналогично истории планов тестирования и тестов. Посмотреть историю можно, нажав на кнопку **History**.

Запуск тестов

Как и всякий тестировщик, Вы, скорее всего, будете проводить большую часть времени за запуском тестов и записью результатов на странице прогона тестов. Когда вы первый раз открываете страницу прогона, отдельные тесты отображаются в таблице как “прогоны отдельных тестов”. Ваш же прогон может включать только те тесты, которые нужны для специальной задачи тестирования, или специального тестировщика. Или же ваш прогон может включать более широкий набор тестов, отражая полный цикл тестирования. А в каждый конкретный день Вы можете захотеть поработать только с тестами наивысших приоритетов, оставляя тесты с более низкими приоритетами на потом. Или же – вы можете захотеть запустить только регрессионные тесты на наборе ранее проваленных тестов... Всё это становится возможным благодаря фильтрам.

Выборка тестов прогона по фильтрам

Разверните меню **Filter**, нажав на треугольник рядом с ним. Вы увидите некоторое количество опций, доступных для фильтрации тестов:


- Статус (Status);
- Категория (Category);
- Сборка (Build);
- Окружение (Environments);
- Приоритет (Priorities);
- Компонент (Components);
- Ответственный (Assignee);
- Метки (Tags);
- Описание (Case summary).


Testopia может сохранять фильтры, чтобы при следующем просмотре прогона Вы снова могли применить сохранённый фильтр. Для сохранения фильтра введите его название и нажмите кнопку **Сохранить** (save). Чтобы очистить фильтр и вернуться к полному списку, нажмите кнопку **Сброс** (reset). Сохранённые фильтры доступны справа, на вкладке «Фильтры» (Filters).

Сортировка списка тестов

Вы можете задавать порядок сортировки списка тестов, кликая на любой из заголовков колонок таблицы; также Вы можете сортировать тесты по описанию, нажимая на любой из заголовков полей **Summary**. Если же Вы хотите создать собственный, новый, порядок сортировки списка тестов, введите номера в поля **Index** и нажмите **Change**. Сортировка по этому полю будет аналогична сортировке по полям с заданными номерами по возрастанию.

Запись успешных и неуспешных результатов тестов

Теперь, когда Вы готовы начать тестирование, разверните первый тест в своём списке и прочитайте списки действий (**Action**) и ожидаемых результатов (**Expected Results**); после этого Вы можете провести тест. Если получены ожидаемые результаты, Вы можете нажать на зелёный флажок, записывая таким образом факт успешного выполнения теста и переводя его в состояние **PASSED**: .

Если же были получены некорректные результаты, или произошла ошибка, Вы можете записать факт неуспешного выполнения теста, нажимая на красный крестик и переводя тест в состояние **FAILED**: .

За объяснениями остальных состояний тестов обратитесь в конец данного документа.

Добавление заметок

Изменяя статус, Вы каждый раз добавляете строчку к полю «Заметки» (Notes); в строчку также включается информация о времени и авторе изменения. Также вы можете добавлять дополнительные замечания, набирая их в поле Добавить замечание (**Add Note**) и нажимая на кнопку **Append Note**.

Прикрепление багов

Вы можете захотеть сопоставить тесту баг – например, в случае, если тест проваливается. Вы можете прикреплять существующие баги или создавать новые. Чтобы прикрепить существующий баг, введите его номер в поле и нажмите Прикрепить баг (**Attach Bug**). После этого баг будет отображён в поле «Выявленные баги» (**Bugs Detected**). Если же Вы хотите создать новый баг, нажмите кнопку **New** — после этого Вы попадёте на страницу ввода бага с уже заполненной в форме информацией о тесте. Соответственно, Вы можете дополнить эту информацию и создать баг. Вся эта активность происходит в новом окне, так что Вы не теряете своего местоположения в процессе тестирования; чтобы вернуться к Вашему прогону тестов, просто закройте окно бага.

Переназначение тестов

Поле «Ответственный» (Assignee) используется, чтобы помогать тестировщикам отслеживать свои тесты. Поменять ответственного за конкретный тест можно, введя его Bugzilla-логин в поле Assignee и нажав **Assign**. Между тем, работа с тестами не ограничивается ответственными за них. Любой пользователь, имеющий права редактирования запуска конкретного теста, может менять его статус. Если тест закрыт (имеет статус PASSED, FAILED или BLOCKED), имя пользователя, который перевёл его в это состояние, выводится в поле **Tested By**.

Изменение сборки или окружения для отдельного теста

Testopia разрабатывалась гибкой. Режимы и методологии тестирования сильно различаются в разных случаях. Чтобы удовлетворять всем этим различиям, прогоны тестов были разработаны с расчётом на максимальную гибкость процесса выполнения тестов. Некоторые группы людей могут создавать один прогон на каждую сборку и окружение, но другие захотят иметь единый прогон, который будет отражать релиз целиком. В этом случае они могут изменять сборку или окружение в каждом конкретном тесте прогона. При этом либо создаётся новый прогон отдельного теста, либо выбирается подходящий, если он уже существует; к нему добавляется заметка, отражающая факт создания, со временем и автором изменения. Если создаётся новая запись, её статус устанавливается в значение IDLE, после чего Вы вновь готовы тестировать — уже с новой сборкой или окружением. Для каждого конкретного прогона, отображаются в первую очередь прогоны тестов, изменённые последними. В сущности, таблица прогонов тестов представляет собой двумерный вид трёхмерного объекта. Изменение сборки или окружения переносит соответствующую запись на передний план. Об этом необходимо помнить при просмотре отчётов по прогонам отдельных тестов, т. к. то, что Вы видите на странице прогона — это только один срез из доступной информации.

Удаление прогонов тестов

Иногда бывает, что Вы случайно добавляете тест в прогон, который не принадлежит вам, или имеет неправильные выбранные сборку и окружение. В этих случаях самое простое — удалить такой тест, удалив его прогон. Чтобы это сделать, во-первых, это должен разрешить администратор в параметрах, а во-вторых, Вы должны иметь соответствующие права. Чтобы удалить отдельный прогон или несколько прогонов, Вы должны их выбрать из списка и нажать на кнопку Удалить (**Delete**) в панели управления. Перед удалением система попросит у Вас подтверждения этого действия.

Изменение нескольких тестов сразу

Вы можете изменить одновременно несколько прогонов отдельных тестов, выбирая их из списка и либо кликая правой кнопкой и выбирая пункт из контекстного меню, либо используя панель инструментов над списком тестов. Таким образом Вы можете изменять статус, прикреплять баги, изменять сборку и/или окружение, и просто удалять выбранные тесты.

Окончание тестирования

Как только все тесты прогона будут завершены, предполагается, что Вы установите статус прогона в значение «Остановлен» (STOPPED); это нужно для предотвращения лишних попыток изменения тестов в прогоне.

Навигация в Testopia

Навигация по страницам Testopia возможна с использованием ссылок внизу страницы Bugzilla. Вы можете использовать поле быстрого поиска (QuickSearch) для быстрого перехода к любому тесту, прогону, плану или окружению; для этого просто введите аббревиатуру типа объекта и ID объекта или часть имени/описания после неё. Ниже приведён список типов объектов:

<i>Объект</i>	<i>Префикс</i>
Тест (TEST CASE)	(Для тестов – необязателен) <ul style="list-style-type: none"> • case • TC • c
План тестирования (TEST PLAN)	<ul style="list-style-type: none"> • plan • TP • p
Прогон тестов (TEST RUN)	<ul style="list-style-type: none"> • run • TR • r
Окружение тестирования (TEST RUN ENVIRONMENT)	<ul style="list-style-type: none"> • env • TE • e
Прогон отдельного теста (TEST CASE-RUN)	<ul style="list-style-type: none"> • caserun • TCR • cr
Тег (TAGS)	<ul style="list-style-type: none"> • tag

Например, для перехода к прогону тестов №45 Вы можете набрать *r 45*, или *tr 45*, или *run 45*.

Для поиска окружений со словом 'Linux' в имени вы можете набрать *e:linux* или *TE Linux*.

Поиск

Возможно Bugzilla по поиску весьма и весьма широки, поэтому Testopia использует систему поиска Bugzilla для поиска тестов, планов тестирования, прогонов, окружений и прогонов отдельных тестов. Каждый из этих объектов имеет собственный набор параметров, по которым можно осуществлять поиск. Чтобы произвести поиск, нажмите на ссылку [Search](#) внизу страницы, а потом выберите вкладку, соответствующую необходимому объекту. Вы увидите форму, в который сможете задать параметры

запроса и нажать кнопку Отправить (**Submit**), что приведёт вас на страницу со списком найденных объектов.

Сортировка результатов поиска

Результаты поиска разбиваются на страницы и могут быть по-разному упорядочены. Чтобы изменить порядок сортировки, нажмите на заголовок колонки, по значению которой Вы хотите упорядочить список, и Вы увидите список, отсортированный по-новому. Чтобы теперь отсортировать список по тому же полю, но в обратном порядке, нажмите на заголовок колонки повторно.

Страничная навигация

По умолчанию Testopia отображает одновременно только 25 записей, но Вы всегда можете изменить количество отображаемых записей, выбирая значение из списка «Размер страницы». Вы также можете использовать ссылки ниже списка или поле «Перейти на страницу» (**Jump To Page**) для перехода на другие страницы списка. Наконец, Вы можете выбрать опцию «Показать все» (View all), если записей не слишком много, и Вы хотите видеть их все сразу.

Пакетная обработка

Со страниц результатов поиска тестов, прогонов и планов Вы можете перейти к изменению сразу группы объектов – для этого выберите нужные объекты в списке и нажмите «править» (edit). Например, в случае с тестами, таким образом Вы можете изменять почти любые их поля, добавлять тесты в прогоны, или связывать с наборами планов. Также Вы можете экспортировать выбранный набор в CSV или XML-формат (эта возможность будет подробно рассмотрена далее).

Сохранение поисковых запросов

Как и Bugzilla, Testopia позволяет сохранять параметры поискового запроса для их повторного использования в будущем – чтобы сохранить поиск, просто введите имя для запроса и нажмите «сохранить запрос» (save search). Хранимые запросы выводятся ниже других ссылок Testopia внизу страницы и на вкладке «Запросы» (Searches). Уже сохранённые запросы всегда можно удалять, выбирая их из списка и нажимая кнопку «Удалить запрос» (Delete saved search).

Отчёты

Как и в случае с поиском, Testopia использует модифицированную версию системы отчётов Bugzilla, позволяя генерировать простые табличные отчёты по наборам параметров выбора. Такие отчёты могут создаваться из наборов тестов, прогонов или планов тестирования. Чтобы создать отчёт, выберите соответствующий пункт из меню Reports на странице панели продукта, выберите необходимые поля для горизонтальной и вертикальной осей отчёта и задайте параметры поиска. Обычно отчёт отображается в табличной форме, но Вы также можете выбирать линейный график или гистограмму, или выводить отчёт в файл формата CSV (comma separated values – значения, разделённые запятыми), который можно открывать в редакторах электронных таблиц.

В дополнение к этим – обычным – отчётам, некоторые объекты могут иметь связанные с ними отчёты – эти отчёты доступны прямо со страниц объектов.

Отчёты по планам тестирования

Отчёты по прогрессу (Build Coverage Reports)

Нажимая правой кнопкой на выбранных прогонах или сборках (на соответствующих вкладках), можно перейти к меню отчётов, из которого можно просмотреть отчёт по тестам, соответствующим выбранным прогонам или сборкам. Аналогично поиску, настройки отчётов могут быть сохранены. Нажмите на иконку инструмента над отчётом, и выберите «сохранить» (save).

Рейтинг багов (Top Bugs)

Данный отчёт показывает баги, выявленные тестами внутри плана тестирования, отсортированные по количеству соответствующих им тестов в порядке убывания; другими словами, отчёт показывает те баги, которые являются причиной провала наибольшего числа тестов.

Баги, выявленные в плане (Bugs Found In This Plan)

Данный отчёт включает в себя все баги, выявленные тестами плана.

Печатный отчёт по прогрессу в % (Printable Percentage Report)

Данный отчёт показывает процент успешно и неуспешно завершённых тестов по прогонам, сборкам или окружениям.

Отчёты по тестам

Оценка времени / реальное время (Estimated vs Actual Time)

Каждому тесту может быть сопоставлена оценка времени, а в сочетании с прогонами тестов её можно использовать для выявления реального затраченного времени. Чтобы это сделать, прогоны отдельных тестов в начале тестирования нужно переводить в состояние RUNNING — при этом Testopia замечает время начала тестирования данного теста в данном прогоне. Когда тест завершается (успешно или неуспешно), разница во времени принимается за реальное время его выполнения. Далее при построения отчёта для каждого теста учитывается среднее по всем его прогонам.

Развётка по статусам (Historic Status Breakdown)

Данная круговая диаграмма отражает процент завершения всех прогонов теста, разделённых по статусам.

Клонирование

Часто удобным подходом является копирование информации о тесте для её использования в новой итерации разработки. Testopia позволяет клонировать тесты, прогоны тестов, и даже планы тестирования целиком, что даёт возможность более эффективно двигаться от старого релиза продукта к новому. Для клонирования объекта достаточно нажать на нём кнопку Клонировать (**Clone**).

Клонирование планов тестирования

Для клонирования плана тестирования:

1. Перейдите к плану тестирования, который хотите клонировать, и нажмите кнопку **Клонировать (Clone)**;
2. Введите имя для нового плана тестирования (или оставьте значение по умолчанию);
3. Выберите продукт и версию продукта для клонирования в неё;
4. Выберите опции клонирования;
5. Нажмите **Клонировать (Clone)**.

При клонировании плана тестирования Вы можете выбрать – клонировать ли вложения, документ плана, теги, клонировать ли списки контроля доступа. Вы также можете выбрать либо опцию копирования всех тестов плана, либо создания ссылок на существующие тесты. При копировании тестов будет создан полностью новый набор тестов, с новыми ID, а создание ссылок даст Вам возможность распространения всех будущих изменений существующих тестов на новый план.

Если Вы всё-таки выберете опцию копирования тестов, Вам также будет предложено выбрать опцию установки Вас в качестве автора всех новых тестов.

Клонирование тестов

Чтобы клонировать тест:

1. Перейдите к тесту, который хотите клонировать, и нажмите кнопку **Клонировать (Clone)**;
2. Выберите способ клонирования – установите флажок **Копировать (Copy)**, если хотите копировать тест, и снимите его, если хотите создать ссылки на тест;
3. Если Вы выбрали копирование, удостоверьтесь в том, что копируете тест во все уже связанные планы тестирования:
 1. Введите ID всех планов тестирования, в которые Вы также хотите копировать тест;
 2. Выберите опции клонирования;
4. Если Вы выбрали создание ссылок, введите ID всех планов тестирования, с которыми Вы хотите связать тест;

Аналогично планам тестирования, у Вас есть опции установки себя в качестве автора копий, копирования вложений, меток, компонентов, и описания (документа плана).

Клонирование прогона

Для клонирования прогона тестов:

1. Перейдите к прогону, который Вы хотите клонировать, и нажмите кнопку **Клонировать (Clone)**;
2. Введите описание копии;

3. Выберите из продукта, в который клонируете, план тестирования;
4. Выберите версию и сборку продукта;
5. Выберите нужные опции клонирования;
6. Нажмите кнопку **Клонировать (Clone)**.

В дополнение к опциям копирования меток и установки себя в качестве ответственного за копию прогона, Вы также можете копировать тесты плана, в т.ч. не все, а только фильтрованные по статусу (например, копировать только проваленные тесты).

Удаление

Вы можете удалять и тесты, планы тестирования, и прогоны. Однако, чтобы делать это, нужно, чтобы был включён параметр `allow-test-deletion`, а Вы должны иметь соответствующие права. Администраторы Bugzilla (члены группы *admin*) могут производить удаление любых объектов Testopia вне зависимости от настроек параметров.

Чтобы удалить объект, нажмите кнопку **Удалить (Delete)** на странице этого объекта. Вам будет предложено подтверждение операции, также предупреждающее о других объектах, которые будут затронуты операцией удаления. Чтобы подтвердить, нажмите кнопку **Удалить данный объект (Delete This Object)**.

Удаление в Testopia безвозвратно. Поэтому, если Вы просто хотите скрыть объект, а не удалять его, используйте следующие способы:

Тесты

Чтобы скрыть тест из результатов любых поисков, поместите его в состояние DISABLED.

Планы тестирования

Чтобы скрыть план тестирования из результатов любых поисков, архивируйте его, нажав на кнопку **Архивировать (Archive)** на странице плана.

Прогоны тестов

Для сокрытия прогонов помещайте их состояние STOPPED.

Импорт и экспорт

Тесты могут быть экспортированы из Testopia в форматы XML или CSV (Comma Separated Values). Чтобы экспортировать тесты, нажмите на кнопку соответствующего формата (XML/CSV) в области **Экспорт (Export)** внизу страницы плана тестирования, теста или страницы результатов поиска тестов.

Экспорт в CSV позволяет Вам просматривать тесты в табличном процессоре, в котором Вы, вероятно, сможете генерировать отчёты собственного формата или нетривиально обрабатывать данные.

Экспорт в XML позволяет Вам переносить тесты в другие экземпляры Testopia или в специализированные СУБД.

Импорт тестов

Вы можете импортировать тесты из других экземпляров Testopia или других систем управления тестами с помощью скрипта `tr_importxml.pl`. Чтобы быть пригодным для импорта, Ваш файл с экспортированными тестами должен соответствовать определению типа документа (DTD) Testopia, находящемуся в файле `testopia.dtd`. Преобразовать XML в нужный формат можно, используя, например, XSLT. XSLT находится за границами данного документа; информацию о них легко найти в сети.

Чтобы импортировать тесты, запустите скрипт `tr_importxml.pl` с единственным аргументом – путём к нужному XML-файлу. Дополнительную информацию об импорте Вы можете прочитать, перейдя по следующей ссылке:

<http://wiki.mozilla.org/Testopia:Documentation:XMLImport>

Testopia и безопасность

Как и Bugzilla, Testopia даёт возможность ограничивать доступ к данным тестов. Так как данные тестирования, как правило, гораздо более глубоки, чем отчёты о багах, Testopia требует входа в систему перед просмотром любых объектов. В Testopia есть несколько уровней безопасности, которые должен пройти пользователь для получения необходимого доступа – первый – это группы Bugzilla, а второй и главный – списки контроля доступа плана тестирования.

Группа “Testers”

При установке Testopia создаёт группу под названием 'Testers'. Члены данной группы имеют доступ на просмотр и изменения любых планов тестирования и других объектов, таких, как тесты и прогоны. Членство в данной группе также требуется для создания новых планов тестирования, клонирования существующих, и работы с окружениями. Если же включён параметр **testopia-allow-group-member-deletes**, члены данной группы также смогут удалять любые объекты Testopia. Членство в данной группе имеет приоритет перед проверкой любых списков контроля доступа к отдельным планам тестирования.

Списки контроля доступа к планам тестирования

В дополнение к группе Testers каждый план тестирования поддерживает отдельный список контроля доступа, используемый для разрешения или запрещения доступа к планам на основе доменов e-mail адресов пользователей, или явном включении. Каждый план тестирования имеет свой отдельный список контроля доступа. Пользователям, не включённым в группу Testers, требуется разрешение доступа через списки контроля доступа (по регулярному выражению или явному выбору) для того, чтобы работать с планом тестирования, или любыми связанными с ним объектами, т. е. тестами или прогонами. Чтобы изменить список контроля доступа, перейдите к плану тестирования и выберите вкладку Разрешения (Permissions).

Регулярное выражения для выбора пользователей

Пользователям с логинами (т. е. e-mail адресами), соответствующими заданному регулярному выражению, могут быть даны разрешения на доступ к отдельным планам тестирования. Регулярное выражение должно создаваться аккуратно, так, чтобы предотвратить доступ случайных пользователей к планам тестирования. Например,

чтобы дать доступ к плану тестирования всем пользователям с адресами на acme.com, нужно написать следующее регулярное выражение:

```
^.*@acme\.com$
```

Чтобы дать доступ всем пользователям с адресами на acme.com и foo.org:

```
^.*@(acme\.com|foo\.org)$
```

Чтобы дать доступ всем пользователям вообще:

```
.*
```

Пустое регулярное выражение не имеет такого же эффекта, т.к. если поле имеет пустое значение, система принимает, что доступ к плану должен даваться только по явным включениям.

После задания регулярного выражения нужно задать уровень доступа.

Явное включение

Если Вы не хотите давать доступ целой группе пользователей, то Вы можете добавить разрешения для конкретного пользователя, введя его логин в соответствующее поле и нажав кнопку **Добавить пользователя (Add User)**. Это наиболее точный способ выбора пользователей для установки разрешений; однако помните, что если пользователь также будет соответствовать регулярному выражению, он получит большие права из двух (заданных явно и полученных через регулярное выражение).

Права доступа

Пользователям в списках контроля доступа планов тестирования могут быть даны следующие права: право чтения, право записи, право удаления, и право управления планами тестирования и связанными с ними объектами.

Чтение

Даёт возможность просматривать план тестирования и все тесты и прогоны, с ним связанные. Тесты, связанные с несколькими планами, будут видны пользователям всех этих планов.

Запись

Подразумевает право на чтение. Также даёт права на модификацию плана и всех связанных тестов и прогонов. Тесты, связанные с более чем одним планом, пользователи менять не могут, если только не имеют доступ на запись ко всем этим планам.

Удаление

Подразумевает права на чтение и запись. Также даёт права удалять план тестирования и все связанные тесты и прогоны. Тесты, связанные с более чем

одним планом, нельзя удалить, если у Вас нет прав на удаление всех этих планов.

Управление

Подразумевает чтение, запись и удаление. Также даёт права менять списки контроля доступа плана тестирования.

Теги (метки)

Testopia использует механизм проставления тегов (меток) для помощи пользователям в систематизации объектов тестирования. В отличие от стандартного метода размещения объектов в иерархии виртуальных каталогов, теги дают возможность устанавливать связи типа «многие ко многим», т. е. позволяют одному тегу быть связанными с несколькими объектами, а одному объекту быть связанным со многими тегами.

Теги – это новаторский подход к систематизации объектов. Они стали очень популярны вместе с большим числом появившихся инструментов Веб 2.0. Теги похожи на ключевые слова Bugzilla за тем исключением, что их не должен создавать администратор. Тегирование объекта создаёт теги автоматически в случае их отсутствия, и они потом могут быть использованы для тегирования любых других объектов.

Ключевая идея здесь заключается в том, что каждый пользователь может систематизировать все объекты так, как хочется ему, не разрушая при этом чужую систематизацию. Все тесты, планы и прогоны могут быть тегированы, и хотя теги явно не связаны с продуктами, Testopia использует продвинутую систему типирования, чтобы автоматически предлагать выбирать теги на основе их отношений к другим объектам данного продукта.

Теги помнят, кто их создаёт и к какому объекту изначально приписывает, таким образом позволяя пользователю управлять тегами, которыми создают именно они.

Добавление тегов к объекту

Чтобы добавить тег к тесту, прогону или плану, начните набирать название тега в поле **Тег (Tag)** на вкладке тегов справа страницы. Если текст, который Вы вводите, является началом названия одного или нескольких тегов, уже связанных с объектами текущего плана, Вам будет предложена подсказка, из которой Вы сможете выбрать подходящие варианты. Вы можете либо выбрать вариант из подсказки, либо просто продолжить вводить имя тега, а потом нажать кнопку **Add**, чтобы добавить выбранный тег к объекту.

Просмотр тегов

Чтобы просмотреть список своих тегов, нажмите на ссылку Tags вверху области тегов на странице любого объекта. С открывшейся страницы Вы также можете просматривать теги по продуктам, или по пользователям.

Testopia и автоматизированные скрипты тестирования

Testopia реализует XML RPC веб-сервис, использующий протокол SOAP для управления объектами Testopia из внешних приложений. Документация по сервису доступна в Wiki проекта по адресу <http://wiki.mozilla.org/Testopia:Documentation:XMLRPC>

Описания полей

Планы тестирования

Имя (Name)

Краткое описание плана тестирования; уникальным быть не должно.

Продукт (Product)

Продукт Bugzilla, с которым связан план.

Версия продукта (Product Version)

Версия связанного продукта Bugzilla. Используется по умолчанию для новых прогонов.

Тип (Type)

Тип тестирования данного плана. Варианты – системный тест, интеграционный тест, юнит-тест, функциональный тест, приёмочный тест. Администраторы Bugzilla могут редактировать данный список вариантов на странице администрирования Testopia.

Архивный или нет (Archived)

Если данный флаг установлен, значит, план был архивирован и по умолчанию не отображается в результатах поиска. Чтобы архивировать план, нажмите кнопку **Archive**. Чтобы восстановить план из архива, нажмите кнопку **Unarchive**.

Документ плана (Plan Document)

Документ плана – это просто детальное описание особенностей и методов тестирования, используемых в данном плане.

Теги (Tags)

См. секцию «Теги» выше.

Тесты

Заголовок (Summary)

Краткое описание теста.

Тестирующий по умолчанию (Default Tester)

Пользователь, задаваемый по умолчанию в качестве тестирующего данного теста при создании нового прогона, включающего тест.

Псевдоним (Alias)

Уникальная строка, которая может использоваться для идентификации данного

теста вместо ID.

Требование (Requirement)

Номер или URL к документу с описанием требования, для тестирования которого предназначен данный тест.

Статус (Status)

Статус теста определяет, должен ли он быть включён в новые прогоны.

- **PROPOSED** – Над данным тестом ведётся работа, и он ещё не был проверен на аккуратность; таким образом, он не готов к включению в прогоны.
- **CONFIRMED** – Тест прошёл проверку и готов к включению в новые прогоны. В новые прогоны включаются *только* тесты, имеющие данный статус.
- **DISABLED** – Данный тест более неприменим.

Приоритет (Priority)

Данное поле отражает уровень тестирования. Тесты с большим приоритетом должны запускаться первыми.

Категория (Category)

Категория продукта, к которой принадлежит данный тест. У каждого продукта есть категория по умолчанию; дополнительные категории могут быть добавлены и использованы для классификации тестов в любое время. Так как категория относится не к плану, а к продукту, все планы продукта будут иметь одинаковый список возможных категорий тестов.

Оценка времени (Estimated Time)

Оценка требуемого на прогон теста времени (в формате ЧЧ:ММ:СС).

Добавлять в прогоны (Add To Runs)

Список ID прогонов, в которые нужно включить данный тест, если он ещё не включён.

Автоматизированный или нет (Automatic)

Тесты могут быть ручными или автоматизированными. Автоматизированные тесты запускаются скриптами, а ручные – тестировщиками.

Скрипт (Script)

Если тест автоматизирован, Вы можете ввести имя скрипта тестирования в данное поле. ВНИМАНИЕ! TESTORIA НЕ ЗАПУСКАЕТ СКРИПТЫ ЗА ВАС. Вы должны сами запускать свой скрипт, получать результаты и передавать их в Testopia. Для автоматического сохранения результатов можно использовать, например, интерфейс XMLRPC.

Аргументы (Arguments)

Если скрипту тестирования необходимо передать какие-либо аргументы, Вы можете хранить их в данном поле. Повторим ещё раз, что TESTOPIA НЕ ЗАПУСКАЕТ СКРИПТЫ ЗА ВАС. Данное поле используется только для хранения информации; скрипты могут получать значение данного поля через интерфейс XMLRPC.

Связанные баги (Attach Bugs)

Вы можете связывать баги с тестами. Данное поле – список ID связанных багов, разделённых запятыми.

Зависит от (Depends on)

Если данный тест для успешного выполнения требует вначале успешного выполнения других тестов, список их ID (разделённый запятыми) может быть введён в данное поле.

Блокирует (Blocks)

Если перед выполнением других тестов требуется вначале успешное выполнение данного теста, то список ID таких тестов (разделённый запятыми) может быть введён в данное поле.

Порядок настройки (Set Up)

Данное поле определяет список шагов, нужных для настройки среды тестирования перед выполнением теста.

Порядок сброса (Break Down)

Данное поле определяет список шагов, требуемых для сброса среды тестирования после выполнения теста, для её очистки от временных результатов теста и подготовки к следующему прогону.

Действия (Action)

Данное поле определяет список действий, требуемых для проведения теста.

Ожидаемые результаты (Expected Results)

Данное поле определяет ожидаемые после проведения теста результаты.

Компонент (Component)

Вы можете связывать тест с компонентами Bugzilla. Тестирование может производиться поверх разных компонентов разных продуктов, поэтому Вы можете добавлять в данное поле несколько компонентов любых продуктов. Выбор продукта из выпадающего списка показывает доступные компоненты; для добавления нужно нажать кнопку **Add**.

Теги (Tags)

См. секцию «Теги» выше.

Прогоны тестов

Версия продукта (Product Version)

Версия продукта Bugzilla, используемая в данном прогоне.

Версия плана (Plan Version)

Версия документа плана, с которой связан прогон.

Ответственный (Manager)

Ответственный за данный прогон пользователь.

Сборка (Build)

Сборка продукта, применяемая по умолчанию к новым прогонам отдельных тестов, добавляемых к прогону.

Состояние (Status)

Определяет активность данного прогона.

- **RUNNING** – Прогон активен; тесты в прогоне могут изменяться.
- **STOPPED** – Прогон остановлен; тесты в таком прогоне не могут быть изменены.

Окружение (Environment)

Окружение тестирования, используемое в данном тесте. Тесты, добавляемые в прогон, по умолчанию получают данное окружение. Окружения создаются через вкладку Environments на панели продукта перед созданием прогонов.

Описание (Summary)

Краткое описание прогона.

Заметки (Notes)

Место любых заметок и наблюдений о данном прогоне.

Прогоны отдельных тестов

Состояние (Status)

Статус прогона теста определяет успешность его выполнения.



- **IDLE**: Статус по умолчанию; тест ещё не проверялся.



- **PASSED**: Тест пройден успешно.



- **FAILED**: Тест завален – получены неожиданные результаты или неизвестное исключение.



- **RUNNING**: Идёт тестирование.



- **PAUSED**: Данный статус используется, чтобы отслеживать проблемы с самим тестом (а не его завершением).



- **BLOCKED**: Одна из зависимостей теста завалена.

Индекс (Index)

Индекс для сортировки, задаваемый пользователем (целое число). Можно использовать, чтобы сортировать тесты в прогоне в любом порядке.

Новая заметка (Add Notes)

Место для ввода новых замечаний к прогону теста.

Заметки (Notes)

Данное поле содержит в себе все заметки, события изменения состояний, и записи прогонов.

Изменение статуса бага (Update bug status)

Если данный флаг установлен, то связанные с тестом баги автоматически помещаются в состояние RESOLVED FIXED при успешном проходе теста и в состояние REOPENED при завале теста.

Связанные баги (Attach Bugs)

Данное поле позволяет связывать баги с прогоном теста. Добавьте разделённый запятыми список ID багов, если хотите связать с прогоном новые баги.

Ответственный (Assignee)

Пользователь, ответственный за запуск теста.

Тестирующий (Tested By)

Пользователь, в действительности запустивший тест, т.е. Переведший его в одно из состояний PASSED, FAILED, или BLOCKED.

Дата закрытия (Close Date)

Время закрытия теста (т.е., его перевода в состояние PASSED, FAILED, или BLOCKED)

Сборка (Build)

Сборка продукта, используемая в данном прогоне.

Окружение (Environment)

Окружение тестирования, используемое в данном прогоне.

Глоссарий

<i>Термин</i>	<i>Объект(ы)</i>	<i>Описание</i>
Действие (Action)	Тест	Список действий, требуемых для проведения теста.
Псевдоним (Alias)	Тест	Уникальная строка, которая может использоваться для идентификации данного теста вместо ID.
Архивный (Archived)	План тестирования	Планы могут быть архивированы и скрыты из результатов поиска по умолчанию.
Аргументы (Arguments)	Тест	Список параметров, которые нужно передать автоматическому скрипту тестирования.
Ответственный (Assignee)		Пользователь, ответственный за изменение статуса теста в прогоне.
Блокирует (Blocks)		Тест, который блокирует другой тест.
Заблокирован (BLOCKED)		Состояние прогона теста, говорящее о том, что одна из его зависимостей провалена.
Сборка (Build)		В тестировании ПО, строка, определяющая результаты периода разработки.
Категория (Category)		Подразделение продукта, используемое для классификации тестов.
Клонирование (Clone)		Точная копия данных между двумя объектами. В Testopia Вы можете клонировать планы тестирования, прогоны и тесты.
Компонент (Component)		Компонент Bugzilla, подразделение продукта.
Подтверждён (CONFIRMED)		Состояние теста, говорящее о том, что тест готов к включению в прогоны.
Тестировщик по умолчанию (Default Tester)		Тестировщик для прогонов данного теста по умолчанию.
Зависимость (Dependency)		Тесты могут зависеть от других тестов. Если первый тест зависит от второго, говорят, что второй тест блокирует первый. Тест, зависящий от другого, должен запускаться после его успешного прогона.
Отключён (DISABLED)		Состояние теста, говорящее о том, что он более не используется в тестировании. Аналог архивирования планов.
Ожидаемые результаты (Expected Results)		Ожидаемые результаты для успешного прогона теста после проведения заданных действий.
Окружение (Environment)		Список окружающих прогон тестов условий.

Провален (FAILED)		Состояние прогона теста, говорящее о том, что данный тест был провален в данном прогоне.
Не задействован (IDLE)		Состояние прогона теста, говорящее о том, что данный тест ещё не был проверен в данном прогоне.
Ответственный (Manager)		Ответственный за данный прогон.
Веха (Milestone)		Объект Bugzilla, подразделение продукта, в котором должны быть исправлены/реализованы заданные баги/возможности. Сборки Testopia могут быть связаны с вехами.
Пройден (PASSED)		Состояние прогона теста, говорящее о том, что данный тест успешно удовлетворил всем необходимым условиям тестирования в данном прогоне.
Приостановлен (PAUSED)		Состояние прогона теста, говорящее о том, что проведение данного теста в данном прогоне было приостановлено. Используется, например, для тестов производительности, требующих длительного времени прогона.
Документ плана (Plan Document)		Описание плана тестирования, говорящее о том, для чего и как был создан план. Может быть очень подробным, в зависимости от уровня безопасности.
Версия плана (Plan Version)		Версия документа плана, используемого в данном прогоне.
Приоритет (Priority)		Приоритет Bugzilla. Каждому тесту можно назначать приоритет, аналогично приоритету багов.
Запланирован (PROPOSED)		Состояние прогона теста, говорящее о том, что тест ещё не был проверен и не готов для включения в прогоны.
Требование (Requirement)		Поле теста, в которое можно вводить информацию о пользовательском требовании (обычно ID или ссылка на страницу описания).
Запущен (RUNNING)		Состояние прогона теста, говорящее о том, что данный тест в данном прогоне находится на стадии исполнения.
Запущен (Running)		Состояние прогона тестов, говорящее о том, что тесты в прогоне могут быть изменены, т. е., что в данном прогоне сейчас ведётся тестирование.
Скрипт (Script)		Путь к внешнему автоматизированному скрипту тестирования для данного теста. Testopia не запускает этот скрипт, поле носит чисто информативный характер.
Остановлен (Stopped)		Состояние прогона тестов, говорящее о том, что тестирование в данном прогоне остановлено, и тесты внутри него не могут быть обновлены.

Тег/метка (Tag)		Строка, задаваемая пользователем для классификации планов тестирования, прогонов и тестов.
Тест (Test Case)		Список условий и ожидаемых результатов, требуемых для успешного функционирования некоторой функции объекта. Тесты связываются с одним или более планами тестирования и любым количеством прогонов.
Прогон теста (Test Case-run)		Объединение теста и прогона. Каждый раз, когда тест включается в новый прогон, создаётся прогон отдельного теста, и появляется запись в таблице прогонов, отражающая состояние теста в прогоне (был он успешно пройден или провален). Каждый прогон теста связывается с ровно одной сборкой.
План тестирования (Test Plan)		Главный объект в Testopia. Включает в себя все остальные объекты.
Прогон тестов (Test Run)		Единица деятельности в Testopia. Каждый прогон связывается с отдельным планом и окружением тестирования. Он содержит список тестов, которые нужно проверить, и отображает результаты в таблице прогонов отдельных тестов.
Тестирующий (Tested By)		Пользователь, произведший тест и присвоивший состояние данному прогону теста.
Тип (Type)		Тип плана тестирования. Например, Системный, Юнит-тест, Интеграционный, и т. д. Каждому плану соответствует ровно один тип.

Получение помощи

Существует некоторое количество ресурсов, на которых Вы можете получить дополнительную информацию о Testopia. В первую очередь проверьте раздел часто задаваемых вопросов (FAQ) на Wiki проекта. Если же Вашей проблемы там не окажется, Вы можете задать свой вопрос в списках рассылки или в IRC. Пожалуйста, не пишите вопросы самим разработчикам напрямую. Они, как и Вы – люди занятые, и не могут быть всё время доступны. Между тем, если Вы зададите свой вопрос в открытом форуме, то Вам, скорее всего, ответит кто-нибудь другой, и Вы получите решение проблемы быстрее.

Списки рассылки

- support-webtools@lists.mozilla.org
- dev-apps-webtools@lists.mozilla.org

Комната IRC

<irc://irc.mozilla.org/testopia>

Wiki

<http://wiki.mozilla.org/Testopia>

Страница проекта

<http://mozilla.org/projects/testopia>

Отправка багов

Testopia находится в состоянии разработки. Таким образом, в ней есть различные баги (ошибки), которые должны быть исправлены. Если Вы найдёте баг или у Вас появится идея о том, как сделать Testopia лучше, перейдите по адресу:

http://bugzilla.mozilla.org/enter_bug.cgi?product=Testopia

И введите описание своей идеи или бага в Testopia. Обязательно включите в отчёт шаги, требуемые для повторения проблемы, и браузер, который Вы используете для работы с Testopia.

Список литературы

Copeland, L. (2004). A Practitioner's Guide to Software Test Design
Norwood, MA: Artech House

IEEE Std 829-1998 IEEE Standard for Software Test Documentation (n.d).
Retrieved February 28, 2007 from http://standards.ieee.org/reading/ieee/std_public/description/se/829-1998_desc.html

Lohmeyer, J. (2004, April 27). Open-Source Bug Tracking with Bugzilla. Linux Journal.
Retrieved February 27, 2007, from <http://www.linuxjournal.com/article/7216>

Meyers, G.J. (2004). The Art of Software Testing, Second Edition
Hoboken, NJ: John Wiley & Sons.

Runnels, D.L. (1999). How to Write Better Test Cases
Retrieved September, 2006 from
<http://www.stickyminds.com/getfile.asp?ot=XML&id=2136&fn=XDD2136filelistfilename1%2Epdf>

Sherry, R.(2006, August).Testopia
Retrieved 2 December 2007 from
<http://blog.drivenqa.com/2006/08/the-hunt-for-a-.html>

Why Software Testing is Important ??? - Good one! (June 6, 2006).
Retrieved January 23, 2007 from
<http://techiecorner.blogspot.com/2006/06/why-softwaretesting-is-important-good.html>