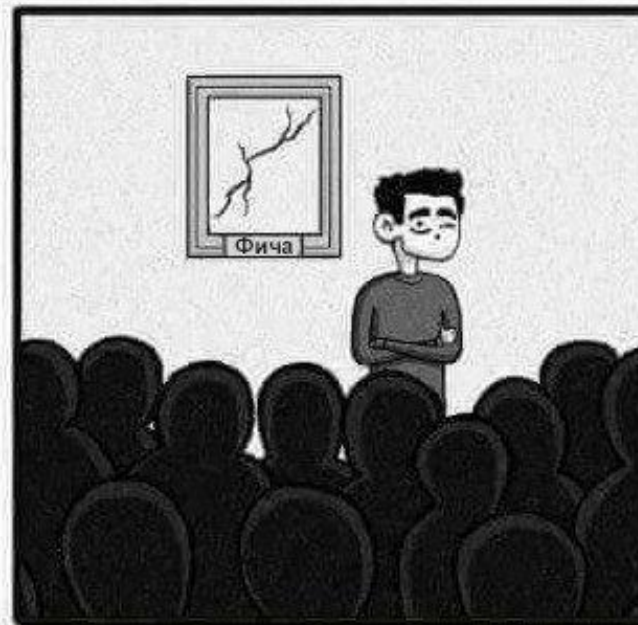
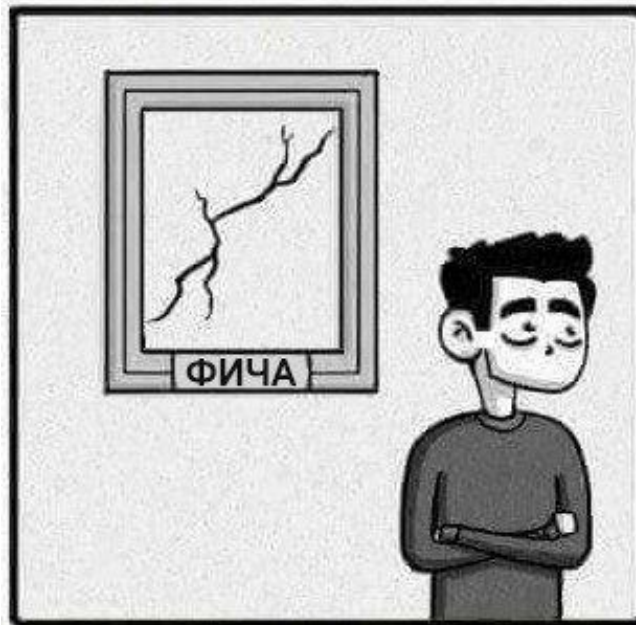
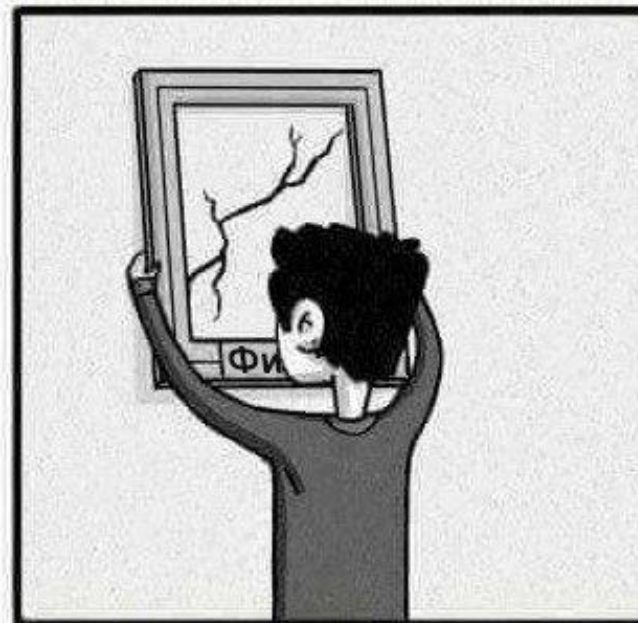
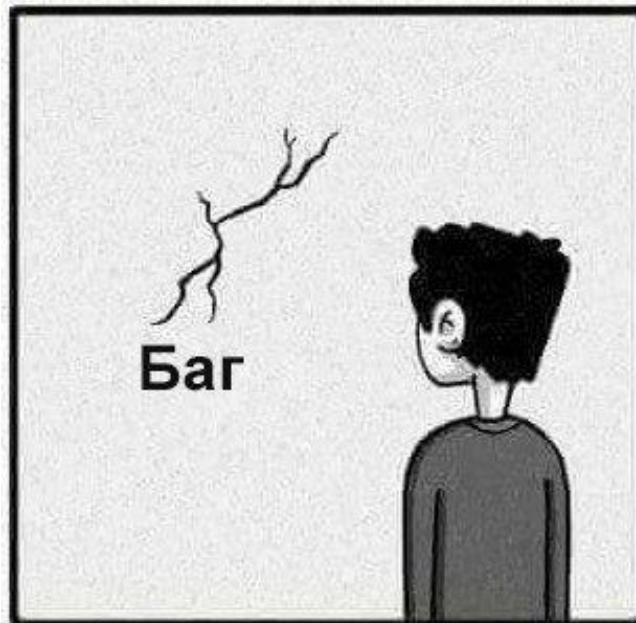


Классификация ошибок
Жизненный цикл бага
Отчеты об ошибках

Это не баг, а фича

?



Что такое баг?



Баг (*bug*) — это отклонение фактического результата (*actual result*) от ожидаемого результата (*expected result*).

Баг живет лишь при **одновременном** выполнении всех трех условий:

1. **Известен** фактический результат;
2. **Известен** ожидаемый результат;
3. **Известно**, что результат из пункта 1 не равен результату из пункта 2.

Источники ожидаемого результата

- ✓ **Спецификация ! ! !**
- ✓ жизненный опыт
- ✓ здравый смысл
- ✓ устоявшиеся стандарты
- ✓ статистические данные
- ✓ авторитетное мнение и др.

Статистические данные:

Время ожидания открытия веб-страницы, после которого пользователь уходит

!!! В большинстве случаев баг — это отклонение от спецификации

Здравый смысл

Пример: по спецификации пользователь может загрузить лишь одну фотографию за раз. А что, если у него таких фотографий 200? Будет он счастлив? Что делаем? Пишем *Feature Request* ("запрос об улучшении").

УСТОЯВШИЕСЯ СТАНДАРТЫ

Как правило, после регистрации, пользователь должен получить е-мейл с подтверждением. Если спек не упоминает о таком е-мейле, вы можете потребовать дополнить его на основании сложившейся практики.

Пример выявления багов

Пункт 19.а спецификации #8724 "О регистрации нового пользователя" устанавливает:
«Поле "Имя" должно быть обязательным. Страница с ошибкой должна быть показана, если пользователь посылает регистрационную форму без заполнения указанного поля».

Функциональный баг

В общем все просто:

- тестировщик идет на страничку с регистрационной формой;
- кликает линк "Регистрация";
- заполняет все обязательные поля, **кроме поля "Имя"**;
- нажимает на кнопку "Зарегистрироваться".

Если ошибка не показана и регистрация подтверждается, то нужно рапортовать баг.

Баг в спецификации

! НО. Непонятно, каким должно быть сообщение об ошибке !

Программист может написать любое на своё усмотрение.

Дефектом называют ошибку в компьютерной программе, возникающую в процессе проектирования или написания кода программы. Ошибка заключается в некорректном поведении программы.

Ошибка, дефект (баг) и отказ в контексте жизненного цикла дефекта

Ошибка (error) — например когда разработчик видит отличие между тем как приложение себя ведет и тем как должно себя вести, в процессе разработки

Дефект (баг) — случается когда тестировщик обнаруживает несоответствие между реальным и предполагаемым поведением приложения в процессе тестирования

Отказ (failure) — несоответствие реального и предполагаемого поведения обнаруживается уже в процессе пользования приложением конечным пользователем, клиентом, или тестировщиком на этапе приемочного тестирования.

Типы багов

В терминологии тестирования программного обеспечения существует множество типов багов, которые можно классифицировать по разным критериям. Вот некоторые из типов багов:

- ✓ Воспроизводимый — баг, который может быть воспроизведен тестировщиком в тестовом окружении.
- ✓ Не воспроизводимый — баг, который не может быть воспроизведен тестировщиком в тестовом окружении.
- ✓ Фатальный — баг, который вызывает крах или аварийное завершение программы.
- ✓ Не фатальный — баг, который не вызывает крах программы, но может приводить к некорректному поведению или неправильным результатам.
- ✓ Функциональный — баг, связанный с неправильной работой функциональности программы.
- ✓ Нефункциональный — баг, связанный с аспектами производительности, надежности, безопасности и т.д. программы.
- ✓ Интерфейсный — баг, связанный с интерфейсом пользователя.
- ✓ Синтаксический — баг, связанный с нарушением правил синтаксиса языка программирования.
- ✓ Логический — баг, связанный с неправильной логикой работы программы.
- ✓ Спецификационный — баг, связанный с нарушением требований к программе, указанных в спецификации.
- ✓ Неоднозначный — баг, который может иметь несколько причин и неоднозначный результат.

Вид бага	Что означает	Пример
Функциональный	Баг, при котором действия пользователя не совпадают с их результатом	Пользователь нажимает на кнопку «Личный кабинет», а попадает в раздел «Корзина»
Визуальный	Баг, из-за которого деформируется интерфейс приложения или веб-страницы	Кнопка и её надпись разъехались в разные стороны
Логический	Баг, при котором приложение или сайт нарушают логику	Сайт разрешает установить слишком старую дату рождения, например 1700 год. Или, наоборот, дату из будущего
Баг нагрузки	Баг, при котором приложение или сайт не справляются с большим количеством входящего трафика	Приложение социальной сети не работает или зависает из-за наплыва пользователей
Баг производительности	Баг, который делает приложение «прожорливым» до памяти, заряда и т. д.	Во время сессий в мобильной игре смартфон разряжается гораздо быстрее, чем в других играх

СЕРЬЕЗНОСТЬ БАГА

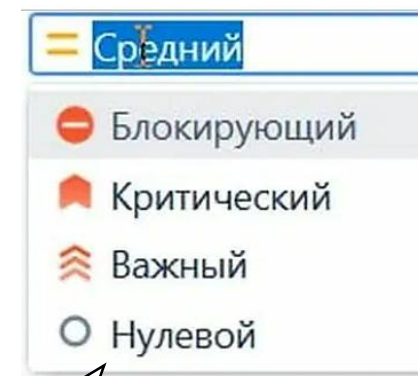
Серьезность (Severity) – это то, насколько баг влияет на систему

- ✓ Блокирующий (Blocker);
- ✓ Критичный (Critical);
- ✓ Значительный (Major);
- ✓ Незначительный (Minor);
- ✓ Тривиальный (Trivial).

ПРИОРИТЕТ БАГА

Приоритет (Priority) – это то, насколько срочно надо исправить баг

- ✓ Высокий (High);
- ✓ Средний (Medium);
- ✓ Низкий (Low).



Могут быть и другие приоритеты

*У бага могут не совпадать уровень приоритетности и серьезности. Например, замечена опечатка на главной странице Google.com. **Приоритет исправления такого бага будет высокий**, потому что его заметит много людей и пострадает репутация компании. А **серьезность бага – незначительная**, потому что это всего лишь опечатка, не влияющая на работу сайта.*

Примеры по уровням важности

Серьёзность бага	Как влияет на систему
S1 — Блокирующая (Blocker).	Приложение не запускается ни на одном устройстве или выдаёт ошибку при загрузке. <i>Пример: приложение не открывается, поэтому невозможно сделать покупку.</i>
S2 — Критическая (Critical).	Часть функционала не работает, но это не блокирует процесс. <i>Пример: Скидка по купону не работает, но товар всё равно можно купить без скидки.</i>
S3 — Значительная (Major).	Часть логики работает некорректно, но пользователь может решить проблему другим способом. <i>Пример: поиск в приложении не выдаёт нужный товар, но его можно найти, если искать по брендам.</i>
S4 — Незначительная (Minor).	Не нарушает логику приложения. <i>Пример: кнопка «Купить» уехала за экран и видна только наполовину.</i>
S5 — Тривиальная (Trivial).	Не относится к логике приложения и не влияет на общее качество продукта. <i>Пример: малозаметная опечатка в меню.</i>

Жизненный цикл дефекта(бага) – это цикл, через который проходит дефект за весь период своего существования, включая его различные состояния.

Он начинается с момента обнаружения тестировщиком нового бага, и заканчивается, когда тестировщик закрывает баг, гарантируя, что он не будет воспроизведен снова.

Некоторые этапы цикла могут зависеть от компании и принятых в ней процедур, методик тестирования и инструментов, а главным образом от используемого баг-трекера (поэтому схемы в разных источниках отличаются).

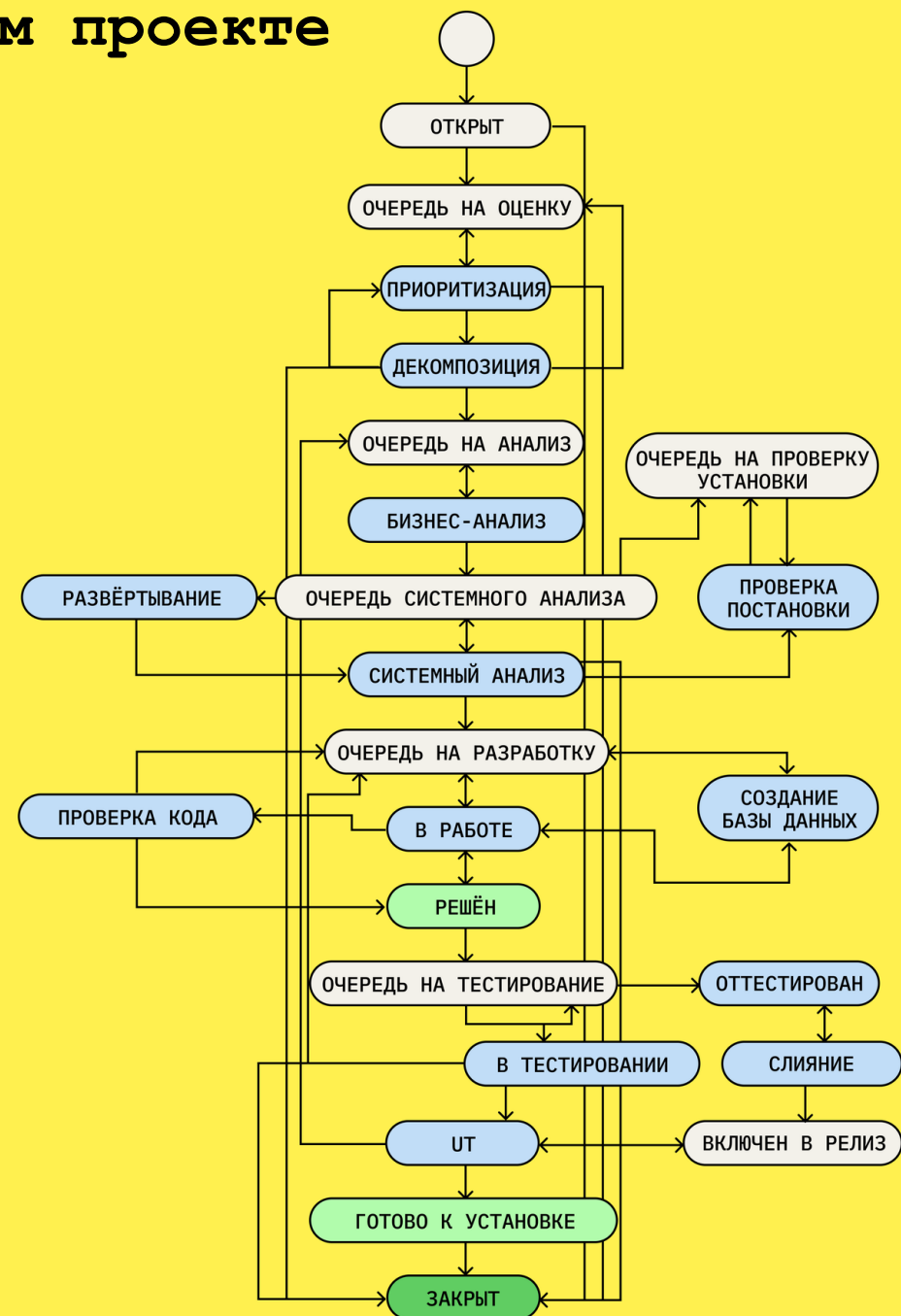
Жизненный цикл бага



Пример жизненного цикла бага на реальном проекте

Допустим, в приложении магазина обнаружили дефект: долго подгружаются товары в каталог. Это может быть связано с тем, что приложение некорректно интегрируется с базой данных товаров. В итоге в каталоге отображаются товары, которых фактически нет, а тех, что в наличии, пользователь не видит. Вот как может выглядеть путь этого бага в Jira:

- Тестировщик описал, в чём проблема, и присвоил задаче статус — новый баг.
 - Задачу в работу берёт аналитик, чтобы уточнить, какие условия закладывали в ТЗ для продукта. Баг получает новый статус — анализ. На проекте может не быть аналитика или задачу не нужно уточнять — тогда её сразу берёт в работу разработчик.
 - Аналитик добавил уточнения по задаче и передал разработчику. Новый статус — в разработке.
 - Тестировщик проводит ретест — проверяет, исправили баг или нет. Если проблему решили, он закрывает задачу, если нет — возвращает задачу разработчику. Она снова получит статус «В разработке».
 - Отработанную задачу тестировщик передаёт во внедрение. После этого приложение либо обновят сразу, либо подождут до релиза: тогда обновлённую функциональность добавят в приложение вместе с другими отработанными задачами.
 - Готово. Теперь пользователь видит только актуальные товары.
- Если дефект повторится, то баг реинкарнирует: его заводят как новую задачу, и он проходит тот же жизненный цикл.



Список статусов бага

Новый. Первый статус в цикле, New, или Обнаружен, означает, что дефект обнаружен тестировщиком, зарегистрирован, и по нему создан баг-репорт, на основе которого разработчик потом будет искать и устранять дефект.

Назначен. Когда новый дефект подтвержден и принят в обработку, получает статус Назначен (Assigned). Как правило назначается ответственный за устранение этого бага (поэтому статус еще может называться «Назначен НА кого-то»).

Открыт. Open-статус означает, что дефект приняли разработчики и начали процесс устранения. На этом этапе возможен переход в статус «Отклонен» или «Отложен», то есть разработчик может «не принять» этот дефект — отклонить или отложить.

Устранен. Или Решен/Исправлен (Fixed, Resolved). Разработчики поработали с кодом, внесли нужные правки, поместили статусом «Исправлен», и возвращают тестировщикам для повторной проверки.

Ожидает повторного тестирования. В статусе Pending Retest дефект ожидает, когда тестировщики повторно проверят его, убедившись что все ОК, код теперь исправлен.

Повторно тестируется. Retest: тестировщик еще раз проверяет этот дефект, и убедившись что он устранен разработчиками, верифицирует это и закрывает дефект, а в противном случае переоткрывает.

Повторно открыт. Если повторное тестирование не смогло устранить баг, обнаруживается снова, ему присваивается статус Reopen. Баг открывается опять и еще раз проходит по циклу.

Проверен. Тестировщик еще раз проверяет (верифицирует) баг, повторно исправленный разработчиком, и если теперь он не проявляется, присваивается статус Verified.

Закрит. Разработчики и тестировщики общими усилиями нашли и устранили дефект, он больше не появляется, и можно присвоить статус Closed.

Отклонен. Разработчик отклонил этот дефект, присвоив статус Rejected, потому что дефект или является дубликатом (уже внесен в систему кем-то из коллег), дефект не воспроизводится, или вообще не считается дефектом.

Отложен. Некоторые дефекты могут счесть не очень важными, не приоритетными, следовательно их можно отложить на потом и устранить в следующих релизах, присвоив статус Deferred и исключив из цикла сейчас.

Дубликат. Этот дефект уже зарегистрирован другим тестировщиком, или суть дефекта та же, тогда присваивается статус Duplicate.

Не дефект. Баг может и есть, но ни на что не влияет, ни в чем не ухудшает функциональность и юзабельность — отмечается статусом Not a Defect.

Не воспроизводится. По какой-то причине баг не удалось воспроизвести, будь то проблемы с платформой, окружением, тестовыми данными, порядком действий и т.п. Присвоен статус Non Reproducible.

Невозможно устранить. Бывают ситуации, когда баг устранить невозможно по какой-то причине: недостатки технологии, стоимость, нехватка времени, недостаточная квалификация или просто лень. Он переводится в статус Can't be fixed.

ОТЧЕТ ОБ ОШИБКЕ (Bug report)

Документ, описывающий и приоритизирующий обнаруженный дефект, а также содействующий его устранению.

ЦЕЛЬ ОТЧЕТА ОБ ОШИБКЕ

- ✓ **Предоставить информацию о проблеме:**
уведомить проектную команду и иных заинтересованных лиц о наличии проблемы, описать суть проблемы;
- ✓ **Приоритизировать проблему:**
определить степень опасности и желаемые сроки её устранения;
- ✓ **Содействовать устранению проблемы:**
предоставить необходимые подробности для понимания сути случившегося, а также анализ причин возникновения проблемы и рекомендации по исправлению ситуации.

РЕЗОЛЮЦИИ ОТЧЕТА ОБ ОШИБКЕ

- | | |
|---|---|
| ✓ Новый; | ✓ Невозможно воспроизвести (Can't reproduce); |
| ✓ Назначен; | ✓ Не будет исправлен (Won't fix); |
| ✓ Открыт; | ✓ Невозможно исправить (Can't fix); |
| ✓ Исправлен; | ✓ Переоткрыт; |
| ✓ Повторное тестирование; | ✓ Рекомендован к отклонению; |
| ✓ Проверен; | ✓ Отклонен (Deferred); |
| ✓ Закрыт; | ✓ Отсрочен. |
| ✓ Работает, как задумано (As Designed); | |
| ✓ Дубликат (Duplicate); | |

АТТРИБУТЫ ОТЧЕТА ОБ ОШИБКЕ

- ✓ Идентификатор (ID);
- ✓ Краткое описание (Summary, Title)
- ✓ Подробное описание (Description):
 - Шаги для воспроизведения (Steps to reproduce);
 - Актуальный результат (Actual result);
 - Ожидаемый результат (Expected result);
- ✓ Воспроизводимость (Repeatability);
- ✓ Важность (Severity);
- ✓ Срочность (Priority);

АТТРИБУТЫ ОТЧЕТА ОБ ОШИБКЕ

- ✓ Возможность обойти (Workaround);
- ✓ Комментарий (Comment, Note);
- ✓ Среда для воспроизведения бага (Environment);
- ✓ Компонент (Component);
- ✓ Тег (Tag, Label);
- ✓ Ответственный (Assignee);
- ✓ Приложение к багу (Attachment);
- ✓ Срок исправления бага (Due Date);
- ✓ Версия билда, где был обнаружен баг (Affects Version);
- ✓ Версия, в которой должен быть исправлен баг (Fix Version);
- ✓ Создатель отчета об ошибке (Reporter);
- ✓ Оценка трудозатрат на исправление бага (Original Estimate).

Название (заголовок) баг-репорта

- Название не должно быть слишком длинным
- Прочитав название должно быть сразу понятно в чем дело
- Принцип “Что? Где? Когда?”

<https://www.youtube.com/watch?v=FRSiAl-ydB0>

Плохой пример:

Если открыть вкладку `сrm`, потом выбрать напоминания, потом мышкой нажать на чек-бокс любого напоминания, то тогда не появится корзина.

Хороший пример:

На вкладке `сrm` в разделе напоминаний не появляется иконка удаления при проставлении чек-бокса у любого напоминания.

Что?

Где?

Когда?

Пример:

В заголовке письма не отображаются символы при отправке письма.

В заголовке письма не отображаются символы при отправке письма.

Что?

Где?

Когда?

✓ Так лучше:

В заголовке письма не отображаются русские символы при отправке письма статуса Daily Stat.

Четкое описание позволит избежать неприятных ситуаций

- Лишняя коммуникация
- Время
- Недовольство сторон

Шаги

Описываем все шаги с упоминанием всех вводимых данных (пароли, итоги промежуточных результатов и т.д.)

- Оптимально не более 7 шагов
- Минимально возможные шаги, выкидываем лишнее
- Добавляем пример, на котором воспроизводится баг, если это необходимо (ссылка, файл, фотография и т.д., именно те, с которыми вы поймали баг)

Плохой пример

Шаги

1. Открыть браузер
2. Открыть jivo.ru
3. Войти в систему
4. Вести данные нашего админа
5. Теперь щелкнуть напоминания
6. В напоминаниях создать новую напоминалку
7. Нужно заполнить все нужные поля
8. Сохраняем

Хороший пример

1. Войти в jivo.ru : (логин: oleg@test.ru, пароль: 123456)
2. Перейти в Управление --> CRM --> Напоминания
4. Нажать на кнопку "Создать напоминание"
5. Заполнить поле "Описание" и "Дата" (Например: Test, 04.04.2023)
6. Нажать на кнопку "Сохранить"

Результат/Фактический результат

- Указываем кратко, что произошло и в каком состоянии находится система
- Прикладываем скриншоты, видео, логи (при грамотно составленном баге разработчику достаточно хорошего названия баг-репорта и скриншота/видео/логов)

1. Войти в jivo.ru : (логин: oleg@test.ru, пароль: 123456)
2. Перейти в Управление --> CRM --> Напоминания
4. Нажать на кнопку "Создать напоминание"
5. Заполнить поле "Описание" и "Дата" (Например: Test, 04.04.2023)
6. Нажать на кнопку "Сохранить"

Плохой пример

Результат: Кажется, напоминание не сохранилось и не создано, но вообще должно было

Результат: Сохранение не работает корректно

Хороший пример:

Появилось уведомление об успешном создании напоминания, но напоминание не появилось в списке.

1. Войти в jivo.ru : (логин: oleg@test.ru, пароль: 123456)
2. Перейти в Управление --> CRM --> Напоминания
4. Нажать на кнопку "Создать напоминание"
5. Заполнить поле "Описание" и "Дата" (Например: Test, 04.04.2023)
6. Нажать на кнопку "Сохранить"

Фактический результат:

Появилось уведомление об успешном создании напоминания, но напоминание не появилось в списке.

Ожидаемый результат
пример

Ожидаемый результат

Плохой пример:

Напоминание должно быть создано

✓ **Хороший пример:**

Появилось уведомление об успешном создании напоминания,
созданное напоминание появилось в списке.

Ожидаемый результат

- В ожидаемом результате указываем по факту, что должно произойти
- Прикладываем скриншот ожидаемого результата.
- Помимо этого прикладываем доказательство, что результат должен быть такой, а не какой то другой



Что служит доказательством:

- ✓ Спецификация
- ✓ Макеты
- ✓ Ссылка на документацию
- ✓ Здравый смысл

СВОЙСТВА КАЧЕСТВЕННЫХ ОТЧЕТОВ

- ✓ Тщательное заполнение всех полей точной и корректной информацией;
- ✓ Отсутствие «размытых» формулировок (*«иногда неправильно работает», «UI сломан», «кнопка Export не работает»*);
- ✓ Вся информация, особенно Ожидаемый результат, корректна;
- ✓ Баг заведен на функциональность, которая разработана (*неразработанный функционал за ошибку не считается*);
- ✓ Отсутствует жаргонная лексика (*«Фигово подцепились картинки»*);
- ✓ Отсутствие критики кого-то из участников команды (*«Ну каким болваном надо быть, чтобы такое сделать?!»*);

СВОЙСТВА КАЧЕСТВЕННЫХ ОТЧЕТОВ

- ✓ Важность и приоритет бага корректно выставлены;
- ✓ К отчету прикреплены все возможные документы, которые облегчат понимание причины дефекта;
- ✓ Отсутствуют грамматические и семантические ошибки («Not keyboard in parameters accepting values»);
- ✓ Использован правильный технический язык;
- ✓ Краткое описание бага должно следовать правилу WWW (what where when);
- ✓ В отчете должен быть описан только 1 дефект;
- ✓ Оформление соответствует соглашению о названиях, принятом на проекте.

Типичные проблемы при описании бага

- ✓ Ошибочные описания и формулировки;
- ✓ Идентичные краткое и подробное описание;
- ✓ Отсутствие фактического и ожидаемого результатов;
- ✓ Лишние пункты при воспроизведении;
- ✓ Копии экрана целиком, без отмеченной проблемы;
- ✓ Выдуманные ошибки;
- ✓ Заниженные (завышенные) срочность и важность;
- ✓ Концентрация на мелочах;
- ✓ Указание в шагах воспроизведения неважной информации;
- ✓ Отсутствие в шагах воспроизведения важной информации;
- ✓ Игнорирование последовательных багов.

Плохо сформулировано

Когда мы как будто бы хотим убрать папку, где что-то внутри есть, оно не спрашивает, хотим ли мы.

- Актуальный результат (Actual result)
- Ожидаемый результат (Expected result)

Хорошо сформулировано

Не производится запрос подтверждения при удалении непустого подкаталога в каталоге SOURCE_DIR.

Act: производится удаление непустого подкаталога (со всем его содержимым) в каталоге SOURCE_DIR без запроса подтверждения.

Exp: в случае, если в каталоге SOURCE_DIR приложение обнаруживает непустой подкаталог, оно прекращает работу с выводом сообщения «Non-empty subfolder [имя подкаталога] in SOURCE_DIR folder detected. Remove it manually or restart application with --force_file_operations key to remove automatically.»

Req: UR.56.BF.4.c.

Рекомендации

1. Перечитать баг-репорт сразу после его написания

2. Один баг - одна задача

НО! Баги, относящиеся к одной группе, оформляют в одном баг-репорте

3. Не дублируем баги

4. Исключить переход на личности

5. Локализуем баг

Пример

В форме Авторизации найдено 4 бага:

1. не проходит авторизация
2. едет верстка в поле логин
3. едет верстка в поле пароль
4. едет верстка в кнопке

Оформляем 2 баг-репорта

- баг-репорт на функцию авторизации
- баг-репорт на верстку в полях и кнопке

Локализация

Тестировщик сталкивается с чем-то непонятным в работе системы и начинает разбираться, что произошло. Это называется **локализацией**. Её цель — убедиться, что обнаружили именно дефект. Для этого тестировщик смотрит проектную документацию, ставит эксперименты и узнаёт, в каких ситуациях воспроизводится дефект и можно ли его как-то обойти.

В результате локализации может быть два вывода:

- ✓ Это не баг, или проблема не на стороне разработчиков. Например, внутренний пользователь чего-то не знает по системе и его нужно обучить. Или у пользователя приложения застряли деньги, а проблема на стороне банка.
- ✓ Это баг программы, и его нужно завести в баг-трекинг-системе (оформить баг-репорт).



Не баг, а фича!



ИНСТРУМЕНТ ДЛЯ ОТЧЕТОВ О БАГАХ

Система для создания, хранения и работы с отчетами об ошибках.

Набор функций:

- ✓ Создание отчётов о дефектах, управление их жизненным циклом с учётом контроля версий, прав доступа и разрешённых переходов из состояния в состояние;
- ✓ Сбор, анализ и предоставление статистики;
- ✓ Рассылка уведомлений, напоминаний и иных артефактов соответствующим сотрудникам;
- ✓ Организация взаимосвязей между отчётами о дефектах, тест-кейсами, требованиями и анализ таких связей с возможностью формирования рекомендаций;
- ✓ Подготовка информации для включения в отчёт о результатах тестирования;
- ✓ Интеграция с системами управления проектами.

MANTIS

MantisHub

mantisbt

vboctor

vboctor

administrator

Issue #

Filters

Advanced Filters

Create Permalink

Viewing Issues

1 - 50 / 2042

Print Reports

CSV Export

Excel Export

Graph

First

Prev

1

2

3

4

5

6

7

8

9

10

11

...

Next

Last

		P	ID	Category	Severity	Status	Updated	Summary
<input type="checkbox"/>			0017964	documentation	text	new	2014-12-13	Documentation - minor typographical error.
<input type="checkbox"/>			0017963	security	major	new	2014-12-13	Error en la Autenticacion
<input type="checkbox"/>			0017918	ui	1 tweak	assigned (syncguru)	2014-12-13	"More Events" Hyperlink in Timeline is not working
<input type="checkbox"/>			0017828	code cleanup	minor	new	2014-12-13	Replace ' ' usages with appropriate constants
<input type="checkbox"/>			0017777	api soap	minor	new	2014-12-13	Add test case for mc_issue_add() and mc_issue_note_add() reporter override
<input type="checkbox"/>			0017460	email	major	new	2014-12-13	Implement a global and generic way of sending e-mail notifications
<input type="checkbox"/>			0016953	email	2 minor	confirmed	2014-12-13	Incorrect usage of \$s_word_separator
<input type="checkbox"/>			0003030	db mysql	4 feature	confirmed	2014-12-13	back slash in search string not escaped
<input type="checkbox"/>			0017910	filters	1 minor	confirmed	2014-12-13	MatchType / Hightlight changed expands all filter fields.
<input type="checkbox"/>			0017852	filters	minor	confirmed	2014-12-13	Tags is showing on its own row in filter box

- Написан на PHP;
- Поддерживает Linux, Windows, Mac;
- Есть time-tracking;
- Разграничение по ролям;
- Отчетность;
- Интеграция с TestLink, Git.

BUGZILLA

Bugzilla – Bug List

Home | New | Browse | Search | Search [?] | Reports | My Requests | Preferences | Help | Log out eileen@softwareadvice.com

Fri May 20 2016 08:59:45 PDT
[I'll be back](#)

[Show Search Description](#)

22 bugs found.

ID	Product	Comp	Status ▲	Resolution	Summary	Changed	
33762	FoodRepl	Salt	CONF	---	testing 1 testing 2	19:27:10	
33750	FoodRepl	Salt II	CONF	---	AQ Test bug	Thu 11:06	
33754	FoodRepl	Salt II	CONF	---	AQ Test bug	Thu 14:36	Windows XP
33755	FoodRepl	Salt II	CONF	---	AQ Test bug	Thu 14:41	Windows XP
33756	FoodRepl	Salt II	CONF	---	AQ Test bug	15:00:36	Windows XP
33751	FoodRepl	renamed	CONF	---	test	Thu 11:28	Mac OS X 10.
33753	FoodRepl	renamed	CONF	---	Google Map showing INCORRECT result for 'Trinethra Indian Supermarket' in Cedar Blvd	Thu 12:51	Windows NT
33763	FoodRepl	renamed	CONF	---	test Summary update asdf all all fdhaki all dakldsai dosa	19:34:38	Windows NT
33765	FoodRepl	renamed	CONF	---	On Costco website search functionality is not showing proper result with the item "Chestnuts"	20:19:48	Windows NT
33757	Sam's Wi	Widget G	CONF	---	On costco website search functionality is not showing proper result with item watermelon	17:19:25	Windows NT
33758	Sam's Wi	Widget G	CONF	---	On Shalimar restaurant website online order link is not working	17:33:14	Windows NT
33764	Sam's Wi	Widget G	CONF	---	Fatal Error Inserir Figura	20:12:18	Windows NT
33770	Sam's Wi	Widget G	CONF	---	test1	03:15:45	Windows NT
33752	FoodRepl	SpiceDis	CONF	---	In COSTCO website search functionality is not showing any result with "Madeleines"	Thu 12:32	Windows NT
33761	FoodRepl	SpiceDis	CONF	---	test	19:25:42	Windows Serv
33767	MyOwnBad	Comp1	CONF	---	myComment	00:37:45	All
33759	WorldCon	Economic	CONF	---	Changed Summary	18:14:35	Windows NT
33760	WorldCon	Economic	CONF	---	This is a test issue.	18:14:45	Windows NT
33768	WorldCon	Economic	CONF	---	Changed Summary	00:39:50	Windows NT
33769	WorldCon	Economic	CONF	---	This is a test issue.	00:40:02	Windows NT
33749	FoodRepl	Salt	IN_P	---	Unable to add product	Thu 11:32	Windows XP
33766	MyOwnBad	Comp1	RESO	FIXE	Bug via API key	00:35:57	All

22 bugs found.

Long Format

Time Summary

Change Columns

Change Several Bugs at Once

- Написан на Perl;
- Поддерживает Linux, Windows, Mac;
- Есть time-tracking;
- Отчетность.

REDMINE

HomeMy pageProjectsAdministrationHelp

Logged as admin - My accountSign out

My project

Search: Jump to a project... ▼

OverviewActivityRoadmapIssuesNewsDocumentsWikiFilesRepositorySettings

Issues

Filters

☒ Status

open ▼

Add filter: ▼

✓ Apply ↺ Clear 💾 Save

#	Tracker	Status	Priority	Subject	Assigned to	Updated
<input type="checkbox"/> 127	Bug	New	Normal	Ticket with attachments		12/22/2007 12:11 PM
<input type="checkbox"/> 116	Bug	New	Low	Keep playing audio when rw/ff and preserve pitch.	John Smith	12/17/2007 09:56 PM
<input type="checkbox"/> 88	Feature	Assigned	Low	HTTP Challenge-MD5 authentication		12/22/2007 04:33 PM
<input type="checkbox"/> 83	Feature	Assigned	Low	Export the parameters of an input	John Smith	12/17/2007 09:56 PM
<input type="checkbox"/> 82	Feature	New	Low	Formatted text rendering support	Dave Loper	12/17/2007 06:58 PM
<input type="checkbox"/> 81	Feature	New	Normal	DVTS support		12/17/2007 06:58 PM
<input checked="" type="checkbox"/> 79	Feature	New	Low	QuickTime capturing		12/17/2007 06:58 PM
<input type="checkbox"/> 78	Feature	New	Low	Full H323 videoconferencing		12/17/2007 06:58 PM
<input type="checkbox"/> 77	Feature	Assigned	Low	Closed captions / Teletext support		12/17/2007 06:58 PM
<input type="checkbox"/> 74	Feature	New	Low	Progressive download playing		
<input type="checkbox"/> 73	Feature	New	Low	Dshow tuning support		
<input type="checkbox"/> 72	Feature	New	Low	V4L tuning support		
<input type="checkbox"/> 70	Feature	New	Low	Electric Program Guide		
<input type="checkbox"/> 69	Bug	New	Low	SDL vout cleaning		
<input type="checkbox"/> 65	Feature	New	Low	Protocol rollover support		
<input type="checkbox"/> 64	Feature	New	Normal	Improve ZLM functionality		12/22/2007 04:33 PM
<input type="checkbox"/> 63	Feature	New	Low	Gstreamer and Helix integration		12/17/2007 06:58 PM
<input type="checkbox"/> 62	Feature	New	Low	Gnutella servlet		12/17/2007 06:58 PM
<input type="checkbox"/> 59	Feature	New	Low	Finalization of Pocket PC port		12/17/2007 06:58 PM
<input type="checkbox"/> 58	Bug	Assigned	Low	Re-write of the AppleScript bindings		12/22/2007 04:33 PM
<input type="checkbox"/> 57	Feature	New	Low	MacOS X SVCD support	Dave Loper	12/17/2007 06:58 PM
<input type="checkbox"/> 51	Bug	New	Low	Better Mozilla plugin control		12/17/2007 06:58 PM

New issue

Tracker: ▼

Issues

View all issues

Summary

Change log

Custom queries

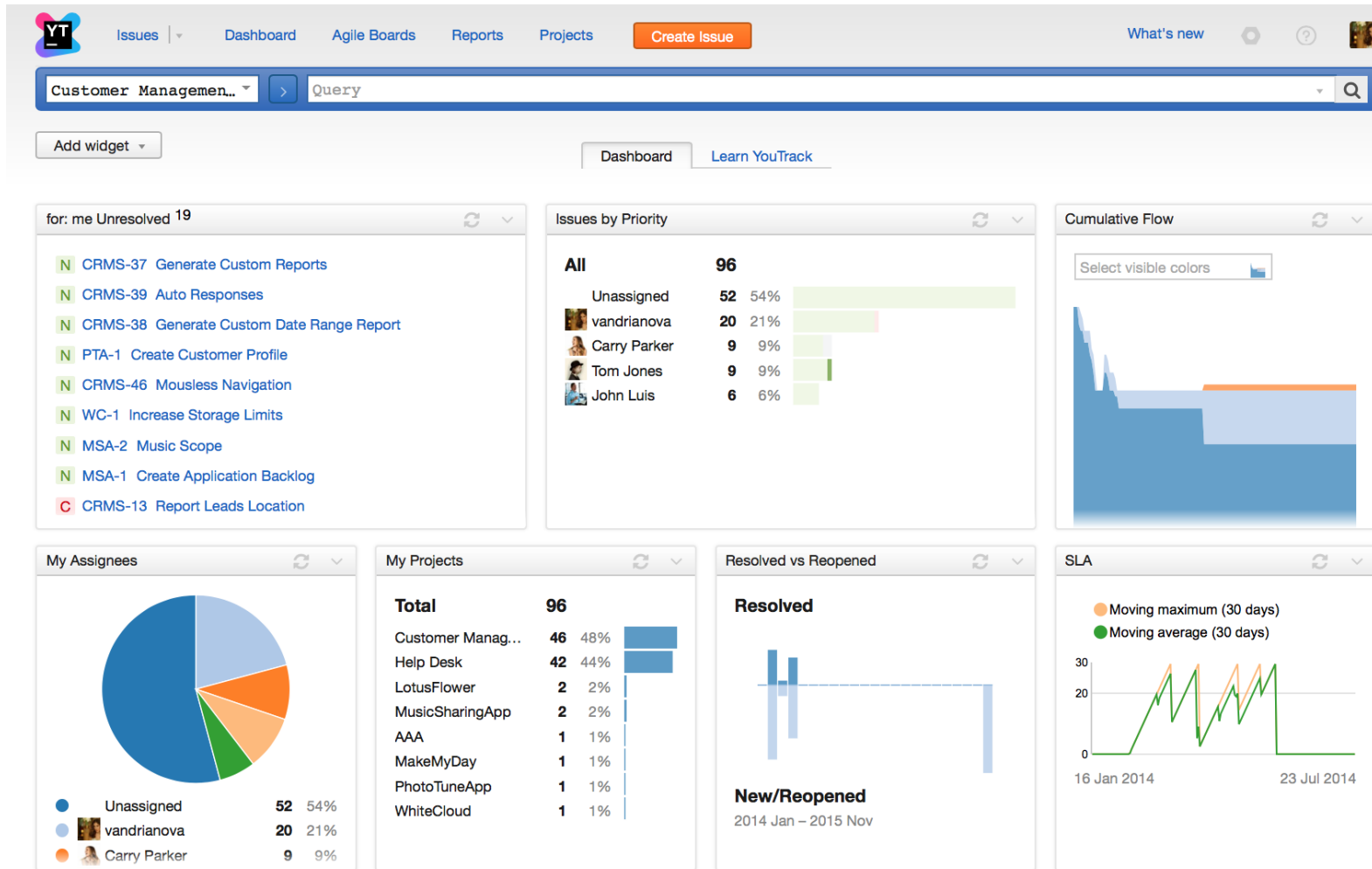
Assigned to me

Due this week

Late features

- Написан на Ruby on Rails;
- Поддерживает Linux, Unix, Mac, Windows;
- Есть time-tracking;
- Разграничение по ролям;
- Интеграция с Git.

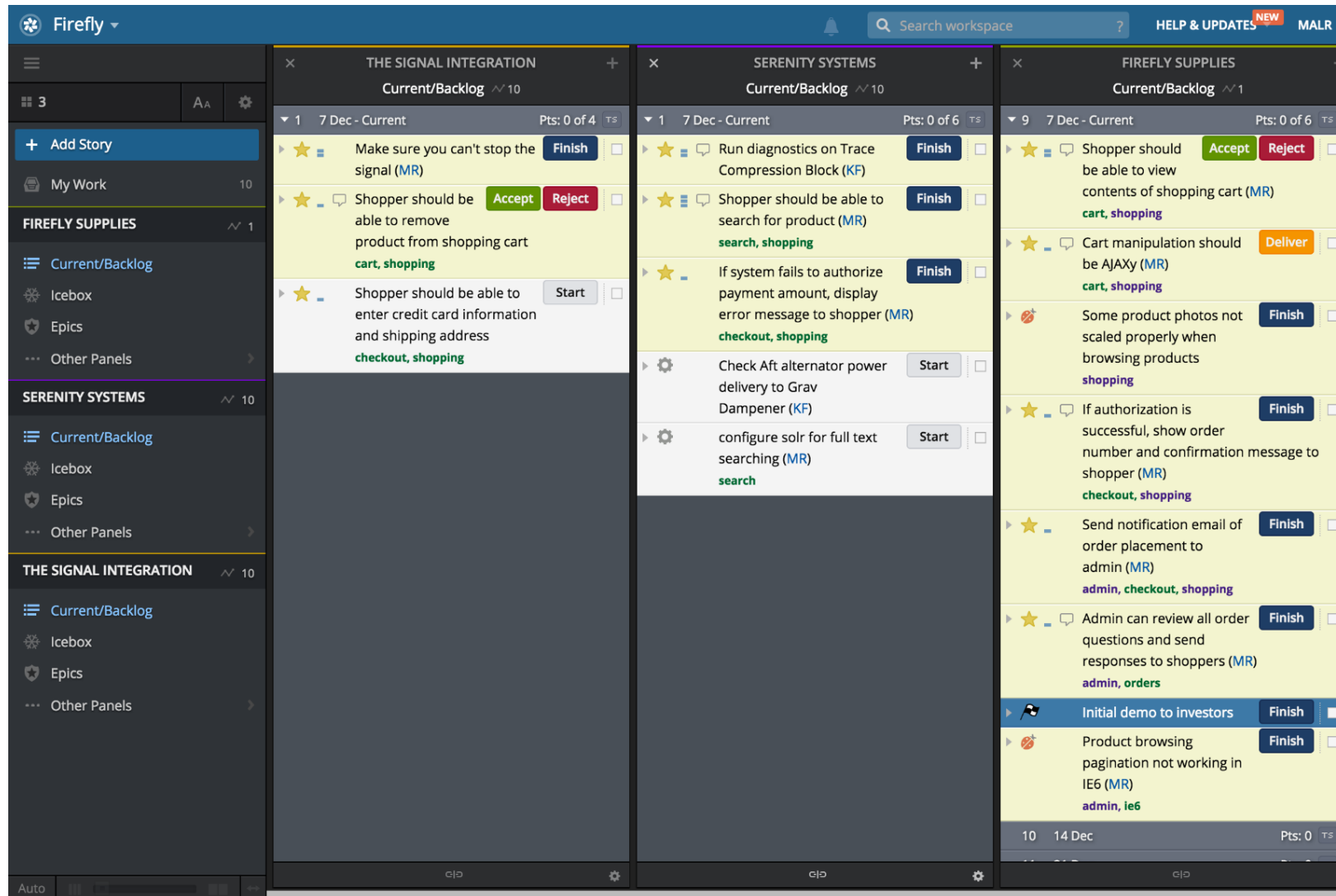
YOUTRACK



- Размещается в облаке;
- Настраивается под Scrum, Kanban;
- Совмещен с управлением проектом;
- Разграничение по ролям;
- Отчетность;
- Есть time-tracking;
- Есть «умный» поиск;
- Есть уведомления пользователей;
- Есть интеграция с тест-кейс системами.

Бесплатный до 10 пользователей

PIVOTAL TRACKER



- Размещается в облаке;
- Настраивается под Scrum, Kanban;
- Совмещен с управлением проектом;
- Отчетность;
- Есть time-tracking;
- Есть «умный» поиск;
- Есть уведомления пользователей;
- Есть интеграция с тест-кейс системами.

Бесплатный до 3 пользователей

ОТЧЕТЫ О РЕЗУЛЬТАТАХ ТЕСТИРОВАНИЯ

ОТЧЕТ О РЕЗУЛЬТАТАХ ТЕСТИРОВАНИЯ

Официальный документ, описывающий результаты, полученные в ходе проведения определенных видов/уровней тестирования.

ЦЕЛЬ НАПИСАНИЯ ОТЧЕТОВ

- ✓ Предоставить статистику по количеству проверок и результатам их выполнения;
- ✓ Предоставить статистику по количеству и серьезности найденных дефектов в приложении;
- ✓ Повысить осведомленность команды о качестве текущей версии приложения;
- ✓ Обратить внимание команды и владельцев бизнеса на области, которые нуждаются в исправлении дефектов
- ✓ Дать рекомендации относительно выпуска или наоборот задержки выпуска версии в релиз.

СТРУКТУРА ОТЧЕТА

1. Уровень/вид проведенного тестирования;
2. Количество пройденных проверок (test cases/test scenarios/checkpoints):
 - Общее количество;
 - Количество успешно пройденных (passed);
 - Количество неудачных (failed);
 - Количество заблокированных проверок (blocked);
3. Количество найденных багов:
 - Общее количество
 - Количество critical багов
 - Количество major багов
 - Количество minor багов
 - Количество trivial багов
4. Области / компоненты, которые наиболее повреждены багами (больше 75% проверок failed)
5. Проблемы, с которыми столкнулись в процессе выполнения задания.

Integration test of AgilQuest with Microsoft Exchange Server (build 1.2.33)

Test cases :

- Total = 120
- Passed = 23
- Failed = 70
- Blocked = 27

Bugs that were found during this round of integration test:

- Total = 75
- Critical bugs = 2
- Major bugs = 24
- Minor bugs = 35
- Trivial bugs = 14

Please, note that the following scenarios are fully not working and require urgent fixes:

- Sync of recurring meetings;
- Sync of meetings for multiple invitees.

During testing we ran into a number of test cases that require update. They will be updated after regression testing of the AgilQuest system.

У БАГОВ ТОЖЕ ЕСТЬ ЧУВСТВА

НАЙДЯ БАГ:
СООБЩИТЕ О НЕМ

БАГИ НЕ ЛЮБЯТ
БЫТЬ ЗАБЫТЫМИ



НАЙДЯ БАГ:
ИЗУЧИТЕ ЕГО

БАГИ ЛЮБЯТ
БЫТЬ ПОНЯТЫМИ



У нее 3
пятнышка

НАЙДЯ БАГ:
СДЕЛАЙТЕ СНИМОК

БАГИ ЛЮБЯТ ХРАНИТЬ
ПАМЯТЬ О ВСТРЕЧЕ



НАЙДЯ БАГ:
УЗНАЙТЕ ЕГО ДРУЗЕЙ

БАГИ ОБЩИТЕЛЬНЫ



НАЙДЯ БАГ:
СООБЩИТЕ О НЕМ БЫСТРО

ИНАЧЕ БАГИ ОБЖИВУТСЯ И
ПОСТРОЯТ СЕБЕ ДОМА



НАЙДЯ БАГ:
БУДЬТЕ ЧЕСТНЫ

БАГИ НЕ ЛЮБЯТ
СПЛЕТЕН



НАЙДЯ БАГ:
ЗАПОМНИТЕ КАК ВЫ
ВСТРЕТИЛИ ЕГО

БАГИ РОМАНТИЧНЫ



НАЙДЯ БАГ:
НЕ ИГНОРИРУЙТЕ ЕГО

БАГИ МОГУТ УКУСИТЬ,
ЕСЛИ ИХ НЕ ЦЕНИТЬ

