автор Екатерина Лаптюхова

БАГИ

Лайфхаки при оформлении дефектов.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

СОДЕРЖАНИЕ

Для начала хочу дать несколько понятий, которые будут использоваться далее.

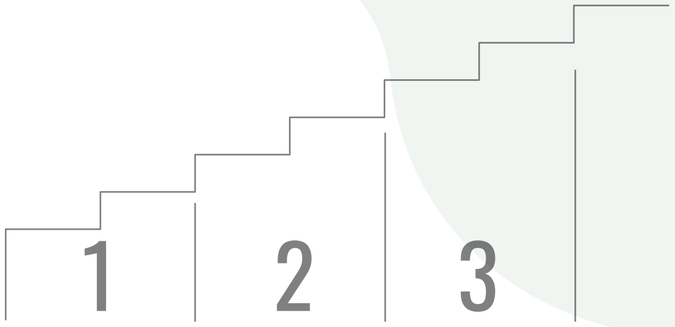
Высокоуровневая архитектура и вытекающие модули (и подмодули).

На всех проектах перед стартом тестирования (во время ознакомления с продуктом) я рекомендую составлять высокоуровневую архитектуру продукта. По сути, это чек-лист, состоящий из названий модулей и подмодулей. И далее использовать этот чек-лист всей командой, чтобы эти названия были едины при заведении дефектов и написании тестов.

Стейдж, стенд, окружение (оно же enviroment).

В данном тексте под всеми этими словами я буду иметь в виду стенд, где происходит тестирование. Если же речь будет идти о девелоперском/демо/предпрод стенде, будет добавлено “дев”/ “демо”/ “предпрод”.

OP и AP - ожидаемый и актуальный результаты.



01
НАЗВАНИЕ

СОДЕРЖАНИЕ

Дефект начинается с названия.

Название должно быть максимально простым, емким, понятным. Чтобы любой сотрудник, будь то разработчик или другой тестировщик, мог по названию определить в чем заключается суть дефекта.

Зачастую, когда название у дефекта длинное, необходимо его открыть, чтобы увидеть полное название, при этом ответ на вопрос «где?» находится в конце.

Есть пара способов облегчить себе жизнь, при составлении красивого названия:

1. Вынесение модулей/подмодулей.

Выделенные модули можно указывать в начале названия с вынесением в скобки, чтобы не дописывать в конце ответ на вопрос «где?».

[Регистрация][Пароль] Отсутствие ошибки при использовании пробелов.

По примеру видно, что можно выделять не только модуль, но и подмодуль/ поле, для которого дефект актуален.

Почему удобнее использовать такое вынесение?

Просто сравните два вариант:

1. [Регистрация][Пароль] Отсутствие ошибки при использовании пробелов.
2. Отсутствие ошибки при использовании пробелов в поле Пароль в форме Регистрации.

Первый вариант короче, легко читается и воспринимается, четко описывает проблему.

Также есть и дополнительные плюсы вынесения модуля в названии:

- вы и ваша команда быстрее сможете находить тикеты (тут важно, чтобы такое правило было прописано для всех),
- быстрый поиск всех багов, относящихся к одному модулю,
- из вышесказанного вытекает сокращение количества дубликатов,
- разработчику не надо будет лезть внутрь каждого бага, чтобы понять к какой части ПО он относится.

автор
Екатерина Лаптюхова

СОДЕРЖАНИЕ

2. Если не знаете с чего начать название, поставьте существительное или отглагольное существительное на первое место.

«Отображение, наличие, отсутствие, несоответствие..»

Дефект – это всегда результат какого-то действия, что великолепно отображается через отглагольное существительное.

Пример работы во время обучения:

- Текст блока «Основные задачи» близко расположен к блоку «Область»

- Отсутствие отступа между блоками ...

Я хотела найти побольше примеров работ “До/ После”, но суть этого лайфхака в том, что мы разбираем его на семинаре и он настолько быстро ложится на практику, что за полчаса поиска, я не нашла подходящих примеров.

И здесь есть свои плюшки:

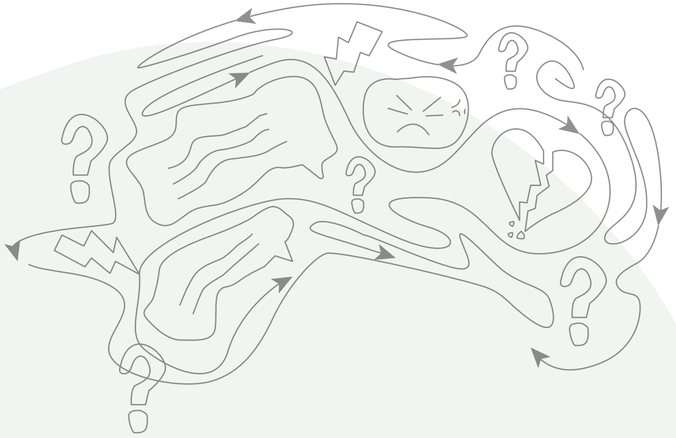
- когда понимаете с чего начинать, остальные слова подбираются в разы быстрее и выливаются в четкое и лаконичное название,

- через отглагольное существительное вы описываете суть дефекта, т.е.: чего-то не хватает, что-то отсутствует, или наоборот есть что-то лишнее, и тем самым вы не даете себе возможности забыть о сути проблемы в названии.

На курсах и в работе на проектах я встречаю ситуации, когда название не отображает сути проблемы:

- Форматирование текста на странице.

- Унифицированный подход к использованию символа «№» и сокращения для слов *— здесь вообще два бага в названии, об этом будет дальше.*



Стоит добавить только одно слово и становится все понятно:

- отсутствие форматирования текста на странице

- отсутствие использования унифицированного ...

СОДЕРЖАНИЕ

А теперь про распространенные ошибки в названиях.

На обучении есть два противоположных фаворита:

Перебор - Пробовать запихнуть несколько багов в название одного.

Недобор - Не давать вообще никакой инфы о том, где это, при каких условиях.

Перебор.

Начнем с первого: попытки отобразить в названии несколько дефектов.

Зачастую, это еще через союз «и».

- [Авторизация] Поле “Логин” смещено влево и поле “Пароль” шире поля Логин

- Отсутствует выравнивание по левой границе элементов меню. Блоки с изображением имеют разную ширину.

Если есть несколько дефектов относительно одного элемента, можно и желательно дать более общее название, а уже в шагах все четко описать.

- [Авторизация] Отсутствие форматирования полей

И внутри данного бага можно перечислять все проблемы, связанные с полями блока Авторизация.

- Отсутствие форматирования элементов меню.

То же самое, списком внутри бага указываем все проблемы форматирования.

Недобор.

Здесь все максимально просто, настолько просто, что даже непонятно.

- Отсутствие выравнивания по левой границе.

А теперь представьте себе сайт с кучей информации, разными блоками (например, Алиэкспрес) и такое название бага.

Название бага должно давать достаточно информации, чтобы без просмотра описания шагов и OP было понятно, что пошло не так.

Конечно, эти рекомендации подходят не под все дефекты и есть случаи, когда тестеры не в силах точно локализовать проблему, а в связи с этим и дать соответствующее краткое и емкое название.

В таких ситуациях больший акцент идет на выделение модуля и общее название бага, а внутри оставляем максимально подробное описание, шаги, AP, OP.

Но в абсолютном большинстве эти рекомендации применимы и приносят пользу командам.

автор
Екатерина Лаптюхова

02 ОКРУЖЕНИЕ/ ENVIRONMENT

Зачастую забываемое поле.

Порой, есть только одно окружение, на котором проводится тестирование, или же наоборот, слишком много и лень перечислять.

В любом случае имеет место договоренность с командой.

Если вы проводите кроссбраузерное/ кроссплатформенное тестирование, стоит договориться на берегу, что вы пишете в поле “Окружение” и когда.

Например:

1. Оставляем поле пустым, если дефект воспроизводится во всех браузерах.

2. Заполняем поле только тем браузером, где воспроизводится бага.

Также, если тестирование проходит на разных стендах (Qa и предпрод, например), необходимо указывать на каком стенде найден дефект.

Здесь можно выделить еще одну рекомендацию, относящуюся к названию и стенду: если дефект воспроизводится в специфическом браузере или на одной оси, эту информацию тоже можно продублировать в названии в скобках, чтобы разработчики с наименьшей вероятностью пропустили важность окружения.

С другой стороны, при просмотре дефектов ПМом/лидом мы чутка уменьшим количество стресса в их жизни:

- Ошибка при оплате товара незарегистрированным пользователем

- [Edge] ошибка при оплате товара незарегистрированным пользователем

!!! *Для сайта, где пользователи Edge составляют <1%, добавление браузера в название имеет критическую для остановки сердца важность.*

03

ЛЕЙБЛЫ

1 2 3

СОДЕРЖАНИЕ

На мой взгляд недооцененное поле.
Но тут главное не переборщить.

Из любимого, какие разделения по лейблам мы используем на проектах:

Модули/подмодули.

Может показаться дублированием информации из названия, но как показывает практика, кто чем привык пользоваться и куда смотреть и по лейблам поиск порой проще.

Окружение, если дефект специфичен.

Это все про тот же Edge, или, например: iOs, Android, Safari..

Регрессионные баги.

Во время каждого регресса мы создаем лейбл с номером версии, например: v.1.0_regress.

Если в регрессионном тестировании есть несколько раундов, то будут разделяться: v.1.0_regress_Round1, v.1.0_regress_Round2.

Последнее мы добавляем по нескольким причинам:

- регрессионные баги имеют наивысший приоритет, если они major и выше и должны быть пофикшены до релиза.

- удобно собирать статистику.

- удобно искать и фильтровать, не используя настроек даты и т.д.

Из дополнительных лейблов:

- во время плэнинга и разбора задач, можем добавлять лейбл ‘questions’, если остались неотвеченные вопросы внутри задачи и необходимо внимание БА или ПО,

- ‘TECH’ когда задача сугубо техническая и не подлежит тестированию.

Правильной системой лейблов можно сильно упростить себе жизнь.

Обо всем необходимо договариваться со всеми командами, в данном случае и с разработчиками, и с БА, и с ПО. Т.к. Questions и questions это два разных лейбла.

Необходимо проговорить ваш подход, вынести название всех лейблов в доступное место, расшарить всей команде и получать удовольствие от простоты их использования.

автор
Екатерина Лаптюхова

04
ОПИСАНИЕ

Здесь мы уже расписываем максимально подробно предусловие, шаги, OP и AP.

Будет нормальным и полезным продублировать информацию об окружении (тестовый стенд + ось/ браузер, если баг зависит от этого).

Я не буду подробно расписывать как описывать шаги, но расскажу про основные моменты, о которых чаще всего забывают и на что стоит обратить внимание:

1. Шаг - должен отвечать на вопрос “что сделать?”, на то он и шаг;

2. OP и AP - это уже результат, а значит действие завершилось и что-то произошло.

В более редких случаях здесь тоже будет действие.

Например, почувствуйте разницу:

- Отобразился попап с информацией.

- Отображается попап с информацией.

В первом случае я понимаю, что по выполнению шага я должна увидеть попап, который быстро отобразился.

Во втором случае результатом шага будет сам процесс отображения, т.е. он будет длительным, что-то в себя включать.

3. Шаги должны быть последовательными.

Если тестер превосходно знает систему, это может стать причиной пропуска шагов, т к он начинает думать, что всем все очевидно.

Информацию необходимо давать так, чтобы вновь прибывший разработчик смог взять в работу тикет и воспроизвести этот баг без дополнительных вопросов, или же любой тестировщик, вчера пришедший на проект, смог протестировать фикс без вопросов “а где это? А как это?”.

4. OP и AP должны быть информативными.

OP и AP должны быть максимально информативными и очевидными. Иногда встречаю ситуацию, когда по описанию бага не понятно вообще в чем же дефект и чего необходимо ожидать от системы.

OP и AP можно и нужно подкреплять скринами/видео, но это не отменяет необходимости описывать результат словами.

(Можно сделать исключения, о которых будет последний раздел).

Здесь еще хочется напомнить, что OP в багах не пишется на каждый шаг (не путаем с тест-кейсом), если только у нас не несколько дефектов на пути одного сценария.

5. Нумерация.

При необходимости, не забываем нумеровать OP и AP, в таком случае, все будет исправлено с большей вероятностью, чем если все предоставить сплошным текстом.

автор
Екатерина Лаптюхова

СОДЕРЖАНИЕ

05 ВЛОЖЕНИЯ

Казалось бы, тут все сильно просто, но увы.

На что стоит обратить внимание:

- Скриншоты должны быть информативные, а это означает, что по ним должно быть понятно в какой части страницы дефект.

- Не надо делать скриншот только одного поля, которое расположено непонятно где, так же, как и не следует скринить всю страницу ради фавиконки.

- Достаточное количество указаний на скрине (стрелочка, подчеркивание, выделение области).

Бывает, что кто-то чувствует себя нереализованным художником и это выплескивается на скриншоты и выделение элемента до такой степени, что самого элемента не видно.

Поверьте, одной стрелки достаточно. Не 7, не 5, даже не трех из разных сторон, а одной.

Тут еще из любимого, когда пишут баг “отсутствие подчеркивания у выбранного элемента”, а на скрине сами подчеркивают линией, где отсутствует подчеркивание :)).

Если у вас случай, что в одном баг-репорте вы описываете несколько UI дефектов, пронумеруйте их на скрине относительно нумерации AP и OP.

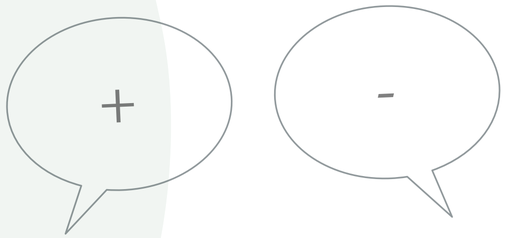
В целом, если в баг-репорт добавляете несколько вложений, то не забудьте дать им названия (для простоты обычно используют нумерацию) и указать в OP и AP где именно применимы эти скрины.

06 ИСКЛЮЧЕНИЯ

Принято считать, что один баг-репорт должен содержать описание одного дефекта. Но все же есть ситуации, когда багов ну очень много и они расположены в одном месте.

Например, редизайн продукта и на одной станице у вас скопились: и не те цвета, и поехавшая верстка, и отсутствие выделения, и присутствие выделения, где не нужно и т.д. и т.п.

В таких ситуация я рекомендую заводить один дефект вида “[] Дизайн не соответствует макету” и на скриншоте через нумерацию обозначить дефекты. В качестве OP мы прикладываем страницу макета.



ИСПОЛЬЗОВАНИЕ ЭТИХ РЕКОМЕНДАЦИЙ ПОМОЖЕТ СОКРАТИТЬ КОЛИЧЕСТВО ВОПРОСОВ К ВАМ ОТ ДРУГИХ КОМАНД И УСКОРИТЬ ПРОЦЕСС ЗАВЕДЕНИЯ БАГОВ.

Екатерина Лаптюхова