

В процессе работы разработчики исправляют баги и реализуют новый функционал. Разные баги и функции имеют разную важность для проекта, разный приоритет (то есть порядок выполнения).

Но у разных людей – разные взгляды на то, что более важно, а что – менее. То, что тестировщик или разработчик сочтёт неважным, может оказаться существенным для заказчика. И наоборот.

Ну а ситуации, когда тестировщик утверждает, что ошибку обязательно нужно исправить, а разработчик с этим никак не соглашается, знакомы каждому из нас. На практике из-за этого часто возникают споры и непонимание.

Кроме того. В отсутствие какого-либо регулирующего документа, при определении важности и приоритета нам остаётся полагаться только на собственное представление о продуктах. Учитывая масштаб и долгую историю продуктов, нет никакой гарантии, что оно

а) будет правильным

б) совпадёт у разных людей.

По этим причинам создается регламент определения критичности и приоритетности дефектов. Пример такого регламента приведен ниже.

Регламент определения критичности и приоритетности дефектов

1 Цель документа

«Регламент определения критичности и приоритетности дефектов» вводится для упорядочивания работы над задачами и ошибками, унификации и упрощения таких задач, как определение очерёдности, в которой должны выполняться задания, постановка критичности тестировщиками, предотвращения споров о важности и приоритетности дефектов и задач.

2 Правила определения критичности

Severity. Критичность ошибки. Определяется тестировщиком при регистрации ошибки. Указывает на серьёзность её последствий для работы системы, удовлетворённость пользователей, финансовые показатели.

Ошибки в backend получают более низкий приоритет, чем ошибки во frontend. Дефекты, для которых существует не приводящий к ошибке обходной путь выполнения, получают более низкий приоритет, чем не имеющие такого пути.

Для определения критичности используется следующий перечень функционала системы:

Базовый функционал

Функция

Пояснение

Реклама

Показ внешних баннеров и тизеров

Добавление контента в общем случае

Backend сайта. Ошибкой затрагивается большинство типов контента.

Чтение контента в общем случае

Frontend сайта. Все браузеры и настройки. Ошибкой затрагивается не менее 25% страниц или л

Продажи

Совершение покупки в интернет-магазине

Дополнительный функционал

Функция

Пояснение

Добавление контента в частном случае

Backend сайта. Ошибкой затрагиваются отдельные типы контента

Чтение контента в частном случае

Frontend сайта. Для конкретных браузеров и настроек. Ошибкой затрагивается не менее 25% с

Прочие функции backend сайта

Любые другие функции

Прочие функции frontend сайта

Любые другие функции

Для дефектов поле «severity» является обязательным, для задач – опциональным. По умолчанию всем новым записям присваивается значение «Minor».

Critical. Критичные дефекты, делающие невозможным работу тестируемого функционала либо нарушающие нормальную работу критичных для бизнес-модели функций. Критичными признаются **стабильно** воспроизводящиеся дефекты,

делающие
вание базовых функций.

НЕВОЗМОЖНЫМ использо

Major. Крупные дефекты, без решения которых запланированный функционал нельзя считать реализованным и работоспособным. Крупными признаются дефекты, **осложняющие** использование базовых функций либо делающие **невозможным** использование дополнительных функций.

Minor. Мелкие дефекты, влияющие на функционал системы в незначительной степени либо имеющие очевидные обходные пути. Мелкими признаются дефекты, **осложняющие** использование дополнительных функций.

Trivial. Тривиальные дефекты, не влияющие на функционал системы, но ухудшающие впечатление пользователя о ней. Тривиальными признаются дефекты в пользовательском интерфейсе, например, опечатки, грамматические, орфографические, пунктуационные ошибки в текстах, малозаметные ошибки в отображении страниц, зависящие от браузера и настроек пользователя и т.п.

3 Правила определения приоритета

Priority. Приоритет ошибки или задачи определяется менеджером проекта при передаче в разработку. Указывает на то, в каком порядке должны делаться задачи, какие должны быть сделаны в первую очередь, а какие позже.

Поле «priority» является обязательным, как для дефектов, так и для задач. По умолчанию всем новым записям присваивается значение «Normal». Регистрирующий запись тестировщик это значение не меняет. При необходимости повысить или понизить приоритет менеджер проекта меняет значение перед передачей в разработку.

Top. Наивысший приоритет. Присваивается ошибкам, которые должны быть исправлены безотлагательно. Этот приоритет может назначаться только ошибкам, но не задачам.

Предназначен для экстренных ситуаций, крайне негативно влияющих на бизнес компании и требующих немедленного исправления. Исправление и публикация таких ошибок выполняются в первую очередь, до реализации любых новых задач.

Обычно наивысший приоритет присваивается критичным функциональным дефектам, но возможны исключения. Например, ситуация когда из-за тривиальной ошибки в пути к файлам на главной странице портала никогда не загружаются логотип и кнопки меню, также должна быть исправлена безотлагательно.

Для исправления таких ошибок может назначаться внеплановая публикация обновлений, в том числе в дни и часы, в которые плановые публикации не допускаются.

High. Высокий приоритет. Присваивается ошибкам и задачам, которые должны быть исправлены в первую очередь. В каждом проекте в любой момент времени может быть не более 10 открытых ошибок или задач высокого приоритета. В случае, если появляется более приоритетная задача, менеджер проекта может присвоить ей приоритет “High” только после того, как понизит приоритет другой задач.

Normal. Обычный приоритет. Присваивается ошибкам и задачам, которые следует исправить во вторую очередь, в рабочем порядке. Этот приоритет выбран по умолчанию. Большинство ошибок и задач должны получать именно такой приоритет.

Low. Низкий приоритет. Присваивается ошибкам и задачам, исправление которых не обязательно и производится в последнюю очередь при наличии времени.