



Баги



Основы тестирования. ЖИЗНЕННЫЙ ЦИКЛ БАГА



Жизненный цикл дефекта: ОСНОВНЫЕ СТАДИИ ВЕРИФИКАЦИИ ОШИБОК

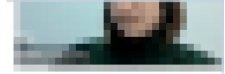


Как правильно оформить баг-репорт

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/849dcb11-9937-40d0-8a83-54585572ebc5/%D0%91%D0%B0%D0%B3-%D1%80%D0%B5%D0%BF%D0%BE%D1%80%D1%82_\(bug_report\)_%D1%87%D1%82%D0%BE_%D1%8D%D1%82%D0%BE_%D1%82%D0%B0%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B_%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2_%D0%BE%D0%B1_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B5.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/849dcb11-9937-40d0-8a83-54585572ebc5/%D0%91%D0%B0%D0%B3-%D1%80%D0%B5%D0%BF%D0%BE%D1%80%D1%82_(bug_report)_%D1%87%D1%82%D0%BE_%D1%8D%D1%82%D0%BE_%D1%82%D0%B0%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B_%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2_%D0%BE%D0%B1_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B5.pdf)

Чтобы признать ошибку дефектом, необходимо, чтобы её смог воспроизвести и разработчик (в некоторых компаниях — аналитик и разработчик). Перед оформлением дефекта нужно его локализовать, оценить масштаб, исследовать окружение (устройство, ОС, версию), проверить на других версиях. Важно выяснить причину (хотя бы предположительно). Всё это нужно добавить в описание дефекта и приложить к нему подтверждающие документы (логи, скрин, скринкаст, рекордер действий).

Обязательные поля



Часть 1

Автор: Ваше имя

Продукт: В каком продукте вы нашли эту ошибку.

Версия: Версия продукта с ошибкой, если таковая имеется.

Компонент: Основные подмодули продукта.

Операционная система:

Укажите все операционные системы, в которых вы обнаружили ошибку.

Приоритет/Серьезность ошибки:

Типы Серьезности ошибки:

- Блокировщик (Blocker): дальнейшая работа по тестированию невозможна.
- Критическая (Critical): сбой приложения, потеря данных.
- Major: серьезная потеря функциональности.
- Minor: незначительная потеря функциональности.
- Незначительная (Trivial): некоторые улучшения пользовательского интерфейса.
- Улучшение (Enhancement): запрос новой функции или некоторого улучшения существующей.

Обязательные поля



Часть 2

Статус ошибки:

Когда вы регистрируете ошибку в любой системе отслеживания ошибок, то по умолчанию статус ошибки будет «Новый».

Позднее ошибка проходит через различные этапы, такие как «Исправлено», «Проверено», «Повторно открыто», «Не исправлено» и т. д.

Назначить разработчику:

Если вы знаете, какой разработчик отвечает за тот конкретный модуль, в котором произошла ошибка, вы можете указать адрес электронной почты этого разработчика. В противном случае оставьте это поле пустым, так как это присвоит полю авторства ошибки значение владельца модуля, если менеджер не назначит ошибку разработчику. Или же назначать на главного тестировщика продукта (проекта)

Окружение:

URL страницы, на которой произошла ошибка.

Описание:

Используйте следующие поля для поля описания:

- Воспроизводимые шаги: ясно упомяните шаги для воспроизведения ошибки.
- Ожидаемый результат: как приложение должно вести себя на вышеуказанных этапах.
- Фактический результат: каков фактический результат выполнения вышеупомянутых шагов, то есть поведение ошибки.

Лайфхаки по написанию багов



- Сначала глагол.
- Принцип «Что-Где-Когда».
- Обезличенность.
- Простые конструкции.
- Без лишних слов.
- Сократить очевидное.
- Упростить описание сложного действия.
- По пунктам.
- Однозначность.
- Перечитать.



Жизненный цикл дефекта: ОСНОВНЫЕ СТАДИИ ВЕРИФИКАЦИИ ОШИБОК



Чтобы отлично понимать статус багов, уметь их быстро и правильно описывать, оформлять и закрывать в системах по отслеживанию ошибок, **необходимо знать их жизненный цикл (Defect Lifecycle).**

Жизненный цикл дефекта – это стадии, которые проходит ошибка с начала своего существования и до ее полного разрешения. Чтобы было проще воспринимать, жизненный цикл рисуют схематично, где отображаются все статусы и действия, которые эти статусы и сменяют.

Каждый этап работы с дефектом обозначается как **«Статус»**. Такой статус на любом этапе показывает, что нужно сделать с ошибкой и кто сейчас с ней работает. Когда один из разработчиков заканчивает работу с ошибкой, статус ошибки меняется и она переходит «под власть» следующего, который продолжает работу с ней.

Чаще всего, **«жизнь» дефекта тесно связана с системой управления, выбранной для него.** В основном, в такой системе все статусы и действия настроены автоматически, но есть и такие системы, где можно настраивать жизненный цикл дефекта под определенный проект, если это нужно.

В системе по отслеживанию ошибок администратор может добавлять посетителей, которые могут просматривать и корректировать дефекты, исходя из их состояния, менять их состояние или полностью удалять.



Статус дефекта

СТАТУСЫ В ЖИЗНЕННОМ ЦИКЛЕ ДЕФЕКТА И ИХ ОБОЗНАЧЕНИЕ

«New» (новый) – описание дефекта записывается в систему управления впервые.

«Assigned» (назначен) – отчет об ошибке назначен на определенного пользователя.

«Open» (открыт) – пользователь начинает работу с отчетом (анализ и редактирование).

«Fixed» (исправлен) – пользователь внес нужные исправления в код и протестировал их самостоятельно. Отчет со статусом **«Исправлен»** снова возвращается тестировщику.

«Pending retest» (Тестирование в режиме ожидания) – разработчик исправил баг, предоставил новый код для тестирования.

«Retesting» (повторное тестирование) – тестировщик повторно проверяет код, измененный разработчиком, с целью посмотреть, исправлена ли ошибка.

«Verified» (проверен) – если дефект исправлен, тестировщик ставит данный статус.

«Reopened» (переоткрыт) – если ошибка снова появляется, тестировщик опять перенаправляет ее на разработчика. Дефект проходит те же стадии.

«Closed» (закрыт) – такой статус ставится тестировщиком, если тот уверен, что дефект исправлен и он больше не появляется.

«Duplicate» (дубликат) – когда дефект встречается дважды или есть два дефекта, появившихся вследствие одной проблемы, один из них получает этот статус.

«Rejected» (отклонен) – если с точки зрения разработчика, ошибка не является весомой и она не требует рассмотрения и исправления, он ее отклоняет.

«Deferred» (отсрочен) – в режиме ожидания, что ошибку с таким статусом исправят в других версиях. В основном, дефект получает такой статус по нескольким причинам: низкий приоритет ошибки, недостаток времени, дефект не приведет к значительным сбоям в программе.

«**Not a bug**» (не является багом) – назначается в том случае, когда функциональные возможности программы меняться не будут. К примеру, заказчик хочет сменить габариты клавиш или цвет продукта — это не ошибка, а просто необходимость внесения поправок в дизайн программы.

СТАДИИ ЖИЗНЕННОГО ЦИКЛА ОШИБКИ

1. Тестировщик обнаруживает дефект.

2. Тестировщик пишет отчет об ошибке в систему управления дефектами (статус **New (новый)**) и перенаправляет его на разработчика (статус **Assigned (назначен)**).

3. Разработчик изучает ошибку, ее возможности воспроизведения и по полученным результатам соотносит ее к одному из статусов:

- **Duplicate (дубликат)** – подобный дефект уже существует в системе по отслеживанию ошибок;
- **Rejected (отклонен)** – ошибка не требует внесения корректив, поскольку ее влияние на продукт незначительное;
- **Deferred (отсрочен)** – корректировку данной ошибки можно осуществить в другой версии программы;
- **Not a bug (не баг)** – дефект не есть ошибкой, поэтому вносит коррективы не требуется;
- **Open (открыт)** – дефект в процессе исправления;
- **Fixed (исправлен)** – код изменен и протестирован разработчиком.

4. Тестировщик повторно проверяет ошибку (статус «**Retesting**» (повторное тестирование)).

5. Если дефект исправлен, тестировщик его закрывает (статусы «**Verified**» (проверен), затем «**Closed**» (закрыт)).

6. Если дефект проявляется и дальше, он опять передается на редактирование разработчику (статусы «**Reopened**» (переоткрыт), «**Assigned**» (назначен)) и вновь проходит через каждую стадию цикла.

На основе статусов ошибок в системе управления дефектами, можно создавать отчеты, по которым уже оценивать работу как программиста, так и тестировщика при выполнении им возложенных услуг по тестированию программного обеспечения.