

Введение в тестирование

Урок 6

Создание баг-репортов. Баг-трекинг-овые системы. Отчёты о тестировании





Оглавление

На этом уроке	3
Что такое баг?	3
Причины появления дефектов	4
Жизненный цикл дефекта	5
Отчёт о дефекте	6
Атрибуты отчёта о дефекте	7
Название отчёта	8
Шаги по воспроизведению	9
Фактический и ожидаемый результаты	9
Серьёзность и приоритет	10
Правила хорошего баг-репорта	10
Багтрекинг-система	11
Отчёт о результатах тестирования	11
Содержание отчёта о тестировании	12
Контрольные вопросы	13
Дополнительные материалы	13



На этом уроке

1. Узнаем, что такое баг-репорт.
2. Рассмотрим жизненный цикл баг-репорта.
3. Разберём атрибуты баг-репорта.
4. Научимся составлять баг-репорты.
5. Поговорим о баг-трекингowych системах.
6. Узнаем о ретесте багов.
7. Поймём, как составлять отчёт о тестировании.

Что такое баг?

На предыдущих уроках мы научились составлять тестовую документацию и проводить тестирование. Узнали, что у качественного ПО есть два признака:

1. Соответствует требованиям — спецификации.
2. Удовлетворяет потребность пользователя.

Но иногда одного или обоих признаков у ПО нет. Например, пользователь хочет оплатить заказ в интернет-магазине картой, но получает сообщение об ошибке. Он столкнулся с дефектом, то есть с отклоняющимся поведением программы.

Любые отклонения поведения системы нужно документировать. Документирование позволяет вести учёт дефектов, оценивать качество продукта и передавать разработчикам информацию об обнаруженных проблемах.

При работе с ПО пользователь или тестировщик ожидает определённого поведения системы. Дефект (баг) — это отклонение фактического результата от ожидаемого.

Например, вы открываете сайт mail.ru — ожидаете увидеть главную страницу, но получаете ошибку: страница не загрузилась. Ответ с ошибкой противоречит ожиданиям, следовательно, это дефект.

По легенде, термин «баг» впервые был применён 9 сентября 1947 года, когда учёные Гарвардского университета, тестировавшие вычислительную машину Mark II Aiken Relay Calculator, нашли мотылька, застрявшего между контактами электромеханического реле. Именно тогда Грейс Хоппер произнесла этот термин. Насекомое достали и вклеили в технический дневник, сопроводив надписью: «First actual case of bug being found» (англ. «первый реальный случай, когда жук был найден»).



Как определить, что поведение, которое мы видим, некорректное? Нужно опираться на требования. Но не у всех проектов есть документация: требования, спецификации и технические задания. Поэтому тестировщику приходится опираться:

- на предыдущий опыт тестирования подобных приложений;
- на критерии качества, работающие на рынке;
- на логику и здравый смысл.

Причины появления дефектов

Проблемы в коммуникациях между членами команды. Если команда не работает слаженно, то разработка тормозится или идёт с ошибками. Разработчик не понял требований, задал вопрос аналитику, но не получил ответ или получил его слишком поздно — вероятно, в системе появились дефекты. Тестировщик обнаружил отклоняющееся поведение, но не нашёл документов с описанием «как надо», не дождался ответа разработчика или заказчика — есть риск получить инцидент.

Сложность программного обеспечения. Чем сложнее программа, тем больше вероятность появления ошибок. Сложная логика работы, множество взаимодействующих систем, нетривиальные алгоритмы — всё это элементы повышенного риска появления дефектов.

Изменение требований. ПО подстраивается под меняющийся мир и потребности пользователя. Изменения требований приводят к модификации кода, и чем позже изменились требования, тем больше работ выполняется для трансформации системы. Правки влекут за собой новые дефекты или возврат уже исправленных.

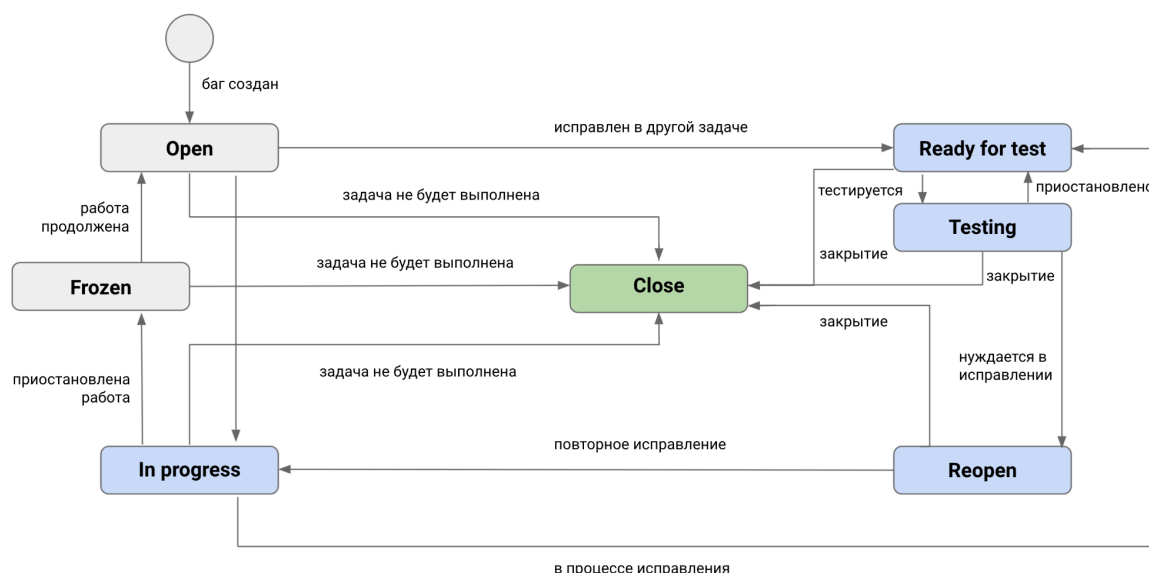
Ошибки программистов. Разработчики должны писать и документировать код по установленным правилам. Но нехватка времени вынуждает их работать быстро, поэтому срабатывает фактор невнимательности и забывчивости.

Ошибки тестировщиков. Тестировщики не замечают ошибки из-за недостатка опыта, упускают проблемы из-за сложности программного обеспечения и невозможности исчерпывающего тестирования.



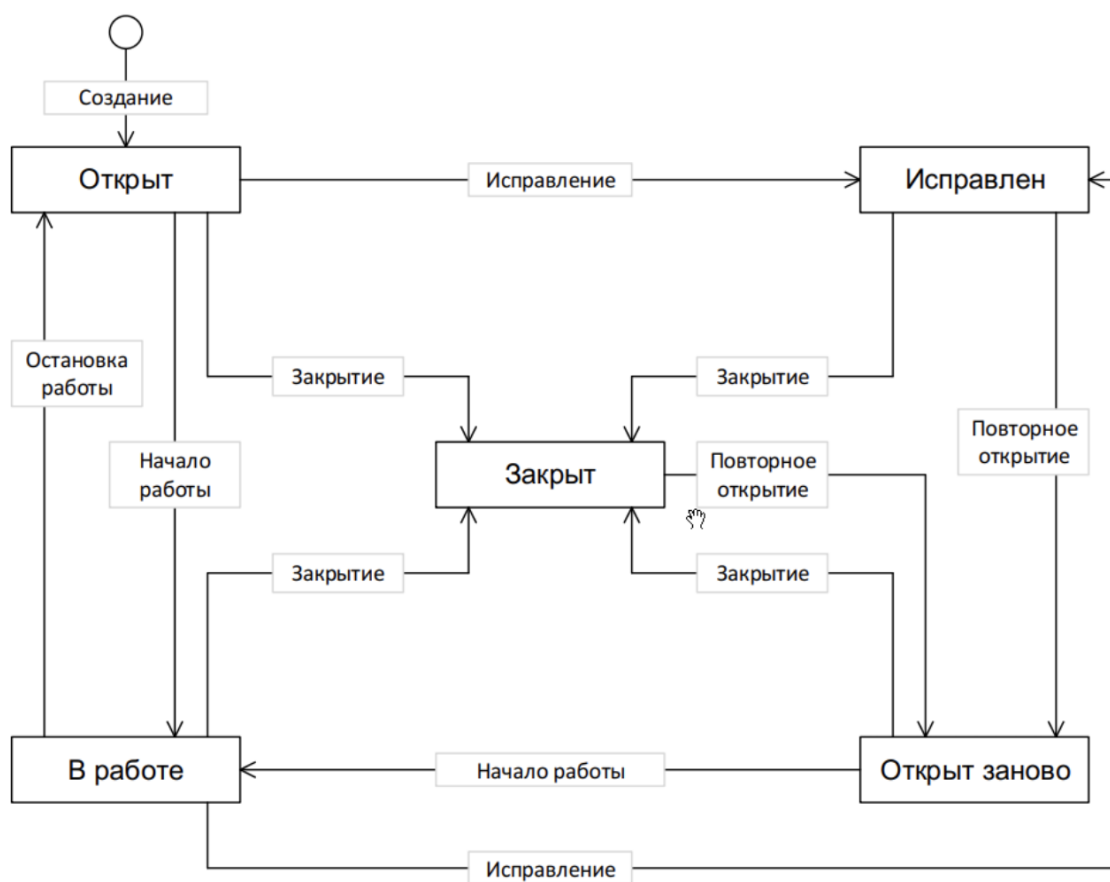
Жизненный цикл дефекта

Тестировщик создаёт новый дефект в системе, затем его чинит разработчик, и, если работоспособность ПО восстановилась, тестировщик закрывает дефект. Эти этапы составляют **жизненный цикл дефекта**.



Этапы жизненного цикла дефекта:

- **Open** — специалист создал баг-репорт.
- **In progress** — баг назначен на разработчика, разработчик начал его исправлять.
- **Ready for test** — разработчик всё поправил, задача готова к тестированию.
- **Testing** — тестировщик взял задачу в тестирование.
- **Reopen** — тестировщик проверил, задача нуждается в доработках.
- **Close** — задача работает, как описано в ТЗ.
- **Frozen** — разработка задачи приостановлена из-за изменений в продукте.



Отчёт о дефекте

Чтобы сообщить о дефекте, тестировщик описывает его установленным образом. В результате получается отчёт о дефекте, или **баг-репорт** (bug report). Для краткости его называют «баг».

Отчёт о дефекте — это документ, который описывает шаги воспроизведения дефекта, фактический и ожидаемый результат, серьёзность дефекта и приоритет устранения.

Отчёты о дефектах — инструменты для сбора статистики на проекте. Они помогают определить:

- в каких областях приложения;
- при каких условиях концентрируются дефекты.

Важная функция отчётов о дефектах — приоритизация проблем. Если дефектов много, и времени на все не хватает, разработчику нужно понимать, какие из них хуже всего влияют на работу приложения, и что исправить в первую очередь.



Отчёт о дефекте предоставляет важные подробности для понимания сути случившегося, анализирует причины возникновения проблемы и даёт рекомендации для исправления.

Атрибуты отчёта о дефекте

Ниже приведён пример отчёта о дефекте. Давайте разберёмся с теми атрибутами, которые нужно соблюдать при его составлении.

№	Название		Предусловие	Шаги воспроизведения	Фактический результат	Ожидаемый результат
2930	Пропадают заголовки на главной странице при скролле		Открыть главную страницу и отключить интернет	1. Проскроллить главную страницу	Пропадают заголовки на главной странице при скролле	Заголовки не пропадают на главной странице при скролле
	Severity	Critical				
	Priority	medium	Окружение			
			macOS Big Sur, Chrome 99.0.4844.84			
	Status	new				
	Assign	Ivan Ivanov				

- Уникальный идентификатор (ID)** — присваивается автоматически.
- Тема** (заголовок) — кратко сформулированная суть дефекта по правилу «Что? Где? Когда?».
- Подробное описание** — более широкое описание сути дефекта (может быть, может не быть).
- Предусловие** — описывается подготовка системы для воспроизведения дефекта.
- Шаги для воспроизведения** — последовательное описание действий, которые привели к выявлению дефекта. Описываются максимально подробно с указанием конкретных вводимых значений.
- Фактический результат** — указывается, что работает не так, в каком месте продукта и при каких условиях.



7. **Ожидаемый результат** — указывается, как именно должна работать система, по мнению тестировщика, основанному на документации.
8. **Вложения** — скриншоты, видео или лог-файлы.
9. **Серьёзность дефекта** — влияние на работоспособность приложения.
10. **Приоритет дефекта** — влияние на очередность выполнения задачи или устранения дефекта. Чем выше приоритет, тем быстрее нужно исправить дефект.
11. **Статус** — текущее состояние дефекта: «открыт», «в работе», «исправлен» и так далее.

Окружение:

1. **Компонент или среда** — на какой платформе или операционной системе дефект воспроизводится: iOS, Android, Windows, Mac.
2. **Версия** — на каком этапе разработки программного продукта обнаружился дефект.
3. **Назначение** — кто ответственен за починку.
4. **Номер сборки**, в которой обнаружился дефект.

Название отчёта

Название должно содержать ответы на вопросы:

1. Что происходит или не происходит согласно спецификации или представлению тестировщика о нормальной работе продукта?
2. В каком месте интерфейса пользователя или архитектуры программного продукта находится проблема?
3. В какой момент работы программного продукта, при наступлении какого события или при каких условиях проявляется проблема?

Это называется правилом «Что? Где? Когда?»

Название содержит предельно краткую, но достаточную для понимания сути проблемы информацию о дефекте. Оптимальная длина — до 10 слов и 80 символов. В названии не употребляются слова «некорректно», «неправильно», «ошибка» и так далее. Тестировщик описывает, что именно некорректно, и в чём именно ошибка.



✗ Неудачное название:

«Возникает ошибка при загрузке большого файла».

Какая ошибка? Насколько большого файла?

✓ Удачное название:

«Код ошибки DB1001 при загрузке файла 10 Гб».

Шаги по воспроизведению

Шаги по воспроизведению — это руководство к действию для тех, кто будет решать проблему. Составляются по правилам:

1. Вернуться к началу. Первый шаг — указание, по какой ссылке перейти, какое окно открыть и так далее.
2. Шаг отвечает на вопрос «Что сделать?».
 - нажать кнопку «Найти»;
 - ввести валидный email и пароль;
 - заполнить требуемые поля валидными данными.
3. Количество шагов — от 2 до 8. Если больше, нужно подумать, какие шаги лишние.
4. Последний шаг — указание, на что обратить внимание:
 - осмотреть текст в выпадающем списке (подменю);
 - обратить внимание на Profile-форму.

Фактический и ожидаемый результаты

Фактический и ожидаемый результаты рекомендуется описывать так:

1. Один дефект — один фактический и ожидаемый результат.
2. В ожидаемом результате дать ссылку на документацию.

Фактический результат. Элементы подменю «Курсы» отображаются за границами выпадающего списка в главном меню после наведения курсора на блок «Курсы».

Ожидаемый результат. Элементы подменю «Курсы» находятся в пределах выпадающего списка в главном меню после наведения курсора на блок «Курсы».



Серьёзность и приоритет

Серьёзность показывает степень ущерба, который наносится проекту дефектом.

1. **Блокирующий** (Blocker) — функция не работает.
2. **Критический** (Critical) — функция работает, но с ограничениями.
3. **Значительный** (Major) — функция работает, но неправильно.
4. **Незначительный** (Minor) — функция работает, но неудобно.
5. **Тривиальный** (Trivial) — грамматические ошибки в пользовательской документации.

Рассмотрим статусы серьёзности на примере автомобиля:

1. Блокирующий — автомобиль не едет, так как закончился бензин.
2. Критический — сломалась дверь у водительского места слева, приходится заходить через правую.
3. Значительный — при попытке включить ближний свет включается дальний.
4. Незначительный — машина с правым рулем в России.
5. Тривиальный — закончился полис ОСАГО.

Приоритет (срочность) показывает, как быстро нужно устранить дефект.

Срочность — высокая (High), средняя (Medium), низкая (Low). Порядок исправления ошибок по их приоритетам:

1. High — высокий, дефект нуждается в срочном устранении.
2. Medium — средний, дефект нуждается в устранении, но это не срочно.
3. Low — низкая срочность исправления дефекта.

Правила хорошего баг-репорта

1. Следовать правилу «Что? Где? Когда?».
2. Одна ошибка — один отчёт о дефекте.
3. Краткость — сестра баг-репорта.
4. Писать техническим языком с применением терминологии, принятой на проекте.
5. Прикреплять дополнительные файлы: логи, скриншоты, видео.
6. Прикреплять ссылки к требованиям, чтобы избежать споров.
7. Избегать дубликатов дефектов — прежде чем сохранить отчёт о дефекте, надо проверить в баг-трекере, есть ли уже такой дефект.



8. Указывать версию ПО и тестовый стенд (окружение), на котором обнаружился дефект.
9. Воспроизводить дефект, следуя собственным шагам.

Багтрекинговая система

Система отслеживания ошибок (от англ. bug tracking system) — прикладная программа, созданная, чтобы разработчикам ПО (программистам, тестировщикам и другим) было проще:

- учитывать и контролировать ошибки и неполадки, найденные в программах, а также пожелания пользователей;
- следить за устранением этих ошибок и выполнением или невыполнением пожеланий.

Багтрекинговые системы используются для создания отчётов о дефектах и для управления ими. Они также применяются при создании задач для команды разработки.

Отчёт о результатах тестирования

Метрики и критерии описываются в плановой документации проекта. В процессе выполнения или завершения тестирования требуется оценить эффективность проделанной работы и соответствие метрик, критериев и прочих характеристик плану. Для этого используются различные виды тестовой отчётности.

Отчётность — сбор и распространение информации о результатах работы, включая текущий статус, оценку прогресса и прогноз развития ситуации.

Отчёт о результатах тестирования — документ, обобщающий результаты работ по тестированию и содержащий информацию, достаточную для соотнесения текущей ситуации с тест-планом и для принятия необходимых управленческих решений.

Отчёт о результатах тестирования может формироваться каждый день, в конце каждой итерации, каждую неделю и так далее. При полном завершении тестирования формируется отчёт о его результатах. Он представляется в разных видах: в текстовом, табличном и графическом.



Содержание отчёта о тестировании

Краткое описание — вводный раздел, в котором кратко описывается содержание отчёта, перечисляются основные числовые показатели хода тестирования, обозначаются главные выводы и даются рекомендации для дальнейшей работы. Этой части отчёта достаточно для общей оценки ситуации заинтересованными лицами (менеджерами, руководителями команд).

Команда тестировщиков — список участников проектной команды, задействованных в тестировании, с указанием их должностей и ролей в отчётный период.

Описание процесса тестирования — вся выполненная тестировщиками работа: количество написанных и пройденных тест-кейсов, число найденных дефектов, добавленные и протестированные функции.

Расписание — календарные сроки, в которые выполнены работы.

Статус активностей тестирования и прогресс по сравнению с планом тестирования — график burndown, на котором видно соотношение запланированных и проведённых работ.

Факторы, препятствующие прогрессу — критичные дефекты, недоработанная функциональность, пробелы в требованиях.

Статистика по новым дефектам — количество дефектов, обнаруженных за отчётный период, с указанием степени важности, срочности, места обнаружения и прочего.

Список новых дефектов — список дефектов, обнаруженных в отчётном периоде, с указанием их основных атрибутов.

Статистика по всем дефектам — график, который отражает динамику обнаружения дефектов на протяжении всего процесса тестирования. В итоговом отчёте по результатам тестирования указываются все обнаруженные дефекты за весь период тестирования, распределённые по выбранной классификации, например, по степени важности, времени обнаружения.

Качество объекта тестирования — экспертное заключение об уровне качества продукта.



Рекомендации — выводы по результатам тестирования и рекомендации по повышению его эффективности в будущем.

Приложения — графики, таблицы, диаграммы, демонстрирующие числовые характеристики процесса тестирования.

Отчёт о тестировании представляется в разных видах: в текстовом, табличном и графическом.

Контрольные вопросы

1. Что такое баг?
2. Что такое баг-репорт?
3. Как должно составляться название отчета о дефекте?
4. Что описываем в шагах воспроизведения?
5. Что описываем в фактическом результате? В ожидаемом результате?
6. Что такое баг-трекинг-система?
7. Из каких частей состоит отчет о результатах тестирования?

Дополнительные материалы

1. [Распространенные ошибки при составлении баг-репортов](#)
2. [Ты ж тестировщик или как правильно составлять Bug report](#)
3. [Примеры хороших заголовков баг-репортов](#)
4. [Багред - проверить название баг-репорта](#)