

Лекция 6.2.
Отчеты о дефектах



!!!

Ошибка (error³⁰⁹, mistake) — действие человека, приводящее к некорректным результатам.

!!!

Сбой (interruption³¹¹) или **отказ** (failure³¹²) — отклонение поведения системы от ожидаемого.

В ГОСТ 27.002-89 даны хорошие и краткие определения сбоя и отказа:

Сбой — самоустраняющийся отказ или однократный отказ, устраняемый незначительным вмешательством оператора.

Отказ — событие, заключающееся в нарушении работоспособного состояния объекта.

!!!

Дефект — отклонение (deviation³¹⁴) фактического результата (actual result³¹⁵) от ожиданий наблюдателя (expected result³¹⁶), сформированных на основе требований, спецификаций, иной документации или опыта и здравого смысла.

Отчет о дефекте

Отчёт о дефекте (defect report) — документ, который:

- описывает обнаруженный дефект,
- определяет его приоритет,
- способствует его устранению.

Создаётся тестировщиком сразу после обнаружения ошибки.

Основные цели отчета о дефекте

1. Информирование

- Уведомить команду и заинтересованных лиц о проблеме.
- Чётко описать суть дефекта.

2. Приоритизация

- Оценить серьёзность и срочность исправления.
- Определить влияние на проект.

3. Содействие исправлению

- Предоставить всю необходимую информацию для воспроизведения и анализа.
- При необходимости — предложить гипотезу причины или рекомендации.

Качество отчета о дефекте

«Хорошо написанный отчёт о дефекте — половина решения проблемы для программиста»

- Плохой отчёт → путаница, потеря времени, повторные запросы.
- Отличный отчёт → быстрое и точное исправление.

Ключевые качества отчёта:

- ✓ полнота
- ✓ корректность
- ✓ логичность
- ✓ аккуратность
- ✓ воспроизводимость

«Сверхцель» отчета о дефекте

Быстрое исправление ошибки + предотвращение её повторного возникновения

- Отчёт — не просто фиксация проблемы, а инструмент повышения качества ПО.
- Инвестиции в качественное документирование дефектов окупаются снижением стоимости исправлений и улучшением надёжности системы.

Пример плохого отчета о дефекте

Заголовок: «Не работает кнопка»

Описание: «Нажал — ничего не произошло. Исправьте, пожалуйста.»

Шаги воспроизведения: Не указаны

Ожидаемый результат: Не указан

Фактический результат: «Не работает»

Окружение: Не указано

× Почему это плохой отчёт?

- Нет конкретики: какая кнопка? Где?
- Отсутствуют шаги для воспроизведения.
- Непонятно, что должно было произойти.
- Невозможно воспроизвести или исправить без дополнительных уточнений.
- Тратит время разработчиков на выяснение деталей.

Пример хорошего отчета о дефекте

Заголовок: «Кнопка "Сохранить профиль" не реагирует при вводе email без домена»

Описание:

При попытке сохранить профиль с email-адресом, не содержащим домен (например, `user@`), форма не отправляется и не отображается сообщение об ошибке.

Шаги воспроизведения:

1. Перейти в раздел «Мой профиль».
2. В поле Email ввести: `user@`
3. Нажать кнопку «Сохранить профиль».

Ожидаемый результат:

Отображается сообщение об ошибке: «Некорректный email-адрес».

Фактический результат:

Кнопка не реагирует, форма не отправляется, сообщение об ошибке отсутствует.

Окружение:

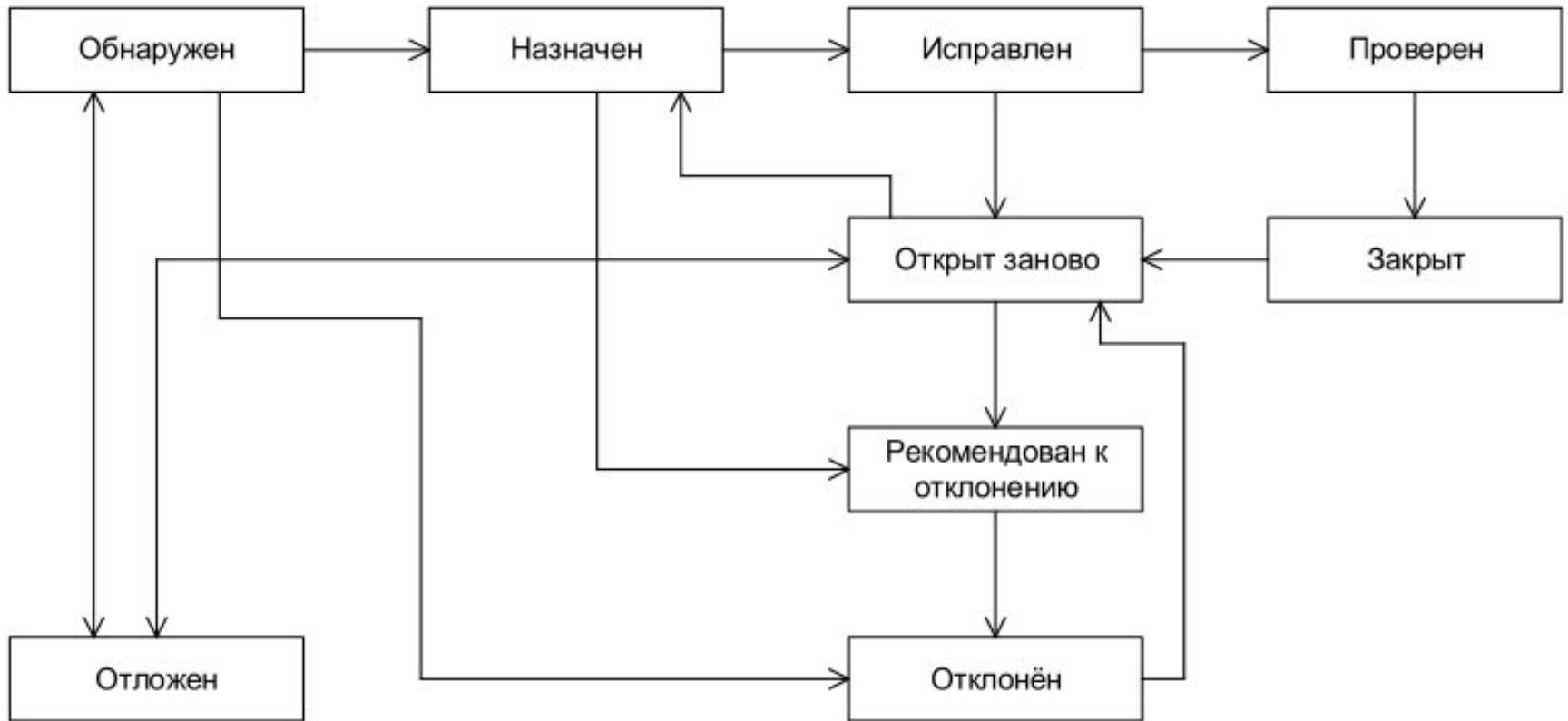
- Браузер: Chrome 120
- ОС: Windows 11
- Версия ПО: v2.4.1

Пример хорошего отчета о дефекте

✓ Почему это хороший отчёт?

- Чёткий, информативный заголовок.
- Полные и воспроизводимые шаги.
- Ясно указаны ожидаемое и фактическое поведение.
- Указано окружение — помогает локализовать проблему.
- Разработчик может сразу приступить к исправлению.

Жизненный цикл отчета о дефекте



Атрибуты отчета о дефекте

Идентификатор

Краткое описание

Подробное описание

Шаги по воспроизведению

19

Бесконечный цикл обработки входного файла с атрибутом «только для чтения»

Если у входного файла выставлен атрибут «только для чтения», после обработки приложению не удастся переместить его в каталог-приёмник: создаётся копия файла, но оригинал не удаляется, и приложение снова и снова обрабатывает этот файл и безуспешно пытается переместить его в каталог-приёмник.

Ожидаемый результат: после обработки файл перемещён из каталога-источника в каталог-приёмник.

Фактический результат: обработанный файл копируется в каталог-приёмник, но его оригинал остаётся в каталоге-источнике.

Требование: [ДС-2.1](#).

1. Поместить в каталог-источник файл допустимого типа и размера.
2. Установить данному файлу атрибут «только для чтения».
3. Запустить приложение.

Дефект: обработанный файл появляется в каталоге-приёмнике, но не удаляется из каталога-источника, файл в каталоге-приёмнике непрерывно обновляется (видно по значению времени последнего изменения).

<p>Всегда</p>	<p>Средняя</p>	<p>Обычная</p>	<p>Некорректная операция</p>	<p>Нет</p>	<p>Если заказчик не планирует использовать установку атрибута «только для чтения» файлам в каталоге-источнике для достижения неких своих целей, можно просто снимать этот атрибут и спокойно перемещать файл.</p>	<p>-</p>
---------------	----------------	----------------	------------------------------	------------	---	----------

Воспроизводимость

Важность

Срочность

Симптом

Возможность обойти

Комментарий

Приложения

Ключевые атрибуты отчета о дефекте

1. Идентификатор (Identifier)

- Уникальный номер или код, однозначно определяющий отчёт.
- Позволяет:
 - ссылаться на дефект в коммуникации,
 - отслеживать его статус,
 - избегать дублирования.
- Может включать префиксы/суффиксы (например, `AUTH-102`, `UI-45`), указывающие на модуль или тип дефекта.

2. Краткое описание (Summary)

Должно отвечать на три ключевых вопроса:

- Что произошло?
- Где это произошло?
- При каких условиях?

✓ Пример:

«Отсутствует логотип на странице приветствия, если пользователь — администратор»

Качественное краткое описание

Рекомендуемый алгоритм:

1. **Полностью поймите суть дефекта** — не пишите отчёт, пока не уверены.
2. **Напишите полное описание** без ограничений по длине.
3. **Удалите лишнее**, оставьте только существенное.
4. **Выделите ключевые элементы**: что, где, при каких условиях.
5. **Сформулируйте грамматически правильное предложение**.
6. **Сократите, если нужно** — используйте общепринятые сокращения или аббревиатуры.

Требования к хорошему summary:

- Лаконичность + информативность
- Уникальность (не похоже на другие отчёты)
- Грамматическая корректность
- Полное отображение на экране (без скролла)



Хороший summary экономит время всей команды!

Подробное описание (Description)

- Это «тело» отчёта — развёрнутое объяснение проблемы.
- Должно содержать:
 - Контекст возникновения дефекта.
 - Чёткое описание фактического поведения.
 - Ожидаемое поведение системы.
- Цель: дать разработчику полную картину без необходимости уточнять детали.

Шаги по воспроизведению (Steps to Reproduce)

- Нумерованный, пошаговый сценарий, который любой член команды может выполнить.
- Пример:
 1. Авторизуйтесь под пользователем `test_user`.
 2. Перейдите в раздел «Настройки».
 3. В поле «Номер телефона» введите `+7`.
 4. Нажмите кнопку «Сохранить».
- **Ключевое правило:** каждый шаг должен быть однозначным и повторяемым [[1], [7]].

Воспроизводимость (Reproducibility)

- Указывает, насколько стабильно проявляется дефект.
- Обычно фиксируется в процентах или категориях:
 - **Всегда (100%)** — воспроизводится при каждом выполнении шагов.
 - **Иногда** — воспроизводится нестабильно (например, при высокой нагрузке).
 - **Невозможно воспроизвести** — наблюдается единожды.
- Эта информация помогает разработчикам оценить сложность диагностики.

Важность (Severity & Priority)

- **Серьёзность (Severity)** — техническое влияние на систему:
 - *Критический* (система падает), *Высокий* (основная функция недоступна), *Низкий* (опечатка).
- **Приоритет (Priority)** — бизнес-важность исправления:
 - *Срочно*, *Высокий*, *Средний*, *Низкий*.
- **Важно:** серьёзность и приоритет — это разные вещи! Например, опечатка на главной странице может иметь низкую серьёзность, но высокий приоритет [[4], [7]].

Обходной путь (Workaround)

- Описание способа временно избежать проблемы без её исправления.
- Пример: *«Пользователь может сохранить данные, если сначала очистит кэш браузера».*
- **Зачем это нужно?**
 - Помогает поддержке работать с пользователями.
 - Снижает срочность исправления, если есть надёжный обход.
 - Может дать подсказку разработчику о причине дефекта.

Другие важные атрибуты

- **Окружение:** ОС, браузер, версия ПО, устройство.
- **Вложения:** Скриншоты, логи, видео — «картинка стоит тысячи слов».
- **Дата обнаружения и автор отчёта** — для отслеживания ответственности и истории.

Хороший отчёт — это не просто жалоба, а готовый план действий для разработчика [[1], [7]].

Алгоритм создания эффективного отчета о дефекте

1. **Обнаружить дефект** 😊
2. **Понять суть проблемы** — чётко определите, что именно работает неправильно.
3. **Воспроизвести дефект** — убедитесь, что проблема повторяется стабильно.
4. **Проверить наличие дубликата** — найдите в системе управления дефектами (Jira, Bugzilla и др.), не сообщал ли кто-то об этом ранее.
5. **Сформулировать суть проблемы по шаблону:**
«Что сделали → Что получили → Что ожидали получить».
6. **Заполнить поля отчёта**, начиная с подробного описания, шагов воспроизведения и окружения.
7. **Перечитать отчёт** — исправьте неточности, добавьте недостающие детали.
8. **Ещё раз перечитать отчёт** — ведь в пункте 6 вы точно что-то упустили! 😊

1. Обнаружить дефект

Что делать:

- Будьте внимательны во время тестирования.
- Не игнорируйте «странные» или «незначительные» аномалии.

Пример:

Вы тестируете форму регистрации. После ввода данных и нажатия «Зарегистрироваться» страница перезагружается, но аккаунт не создаётся.

Важно: Это отправная точка всего процесса. Без обнаружения — нет отчёта!

2. Понять суть проблемы

Что делать:

- Не спешите писать отчёт!
- Задайте себе вопросы: «*Что именно пошло не так?*», «*Какая функция нарушена?*».
- Отделите симптом от причины (если возможно).

Пример (продолжение):

Проблема не в кнопке, а в том, что данные формы не отправляются на сервер. Суть: «**Форма регистрации не отправляет данные**».

Почему это важно: Непонятая проблема приведёт к неправильному описанию и потере времени www.oreateai.com.

3. Воспроизвести дефект

Что делать:

- Повторите последовательность действий минимум 2-3 раза.
- Убедитесь, что дефект воспроизводится стабильно.

Пример:

1. Открыть страницу регистрации.
2. Заполнить все поля корректными данными.
3. Нажать «Зарегистрироваться».

Результат: Страница перезагружается, пользователь не создан.

(Повторено 3 раза).

Цель: Подтвердить, что это не случайный сбой, а настоящий дефект

4. Проверить наличие дубликата

Что делать:

- Перед созданием нового отчёта **обязательно** выполните поиск в системе управления дефектами (Jira, Bugzilla и т.д.).
- Используйте ключевые слова из вашего краткого описания.

Пример поиска:

Запрос: регистрация форма не отправляется

Почему это критично:

- Избегает засорения системы дубликатами.
- Экономит время всей команды.
- Позволяет объединить информацию, если дефект уже известен

5. Сформулировать суть по шаблону

Что делать:

- Используйте универсальный шаблон для ясности:
| «Что сделали → Что получили → Что ожидали»

Пример:

Что сделали: Заполнили форму регистрации и нажали кнопку «Зарегистрироваться».

Что получили: Страница перезагрузилась, новый пользователь не создан, сообщение об успехе не показано.

Что ожидали: Пользователь должен быть создан, и должна отобразиться страница «Добро пожаловать».

Эта формулировка станет основой для **Подробного описания** и **Краткого заголовка** medium.com.

6. Заполнить поля отчета

Что делать:

- Начните с **подробного описания**, используя формулировку из Шага 4.
- Последовательно заполните все обязательные поля:
 - Идентификатор (система обычно генерирует его сама)
 - Краткое описание (Summary)
 - Шаги воспроизведения
 - Ожидаемый / Фактический результат
 - Окружение, Вложения и т.д.

Совет: Лучше перестраховаться и добавить чуть больше деталей, чем их не хватит www.testdevlab.com.

7. Перечитать, исправить

Что делать:

- **Шаг 6:** Прочитайте отчёт от начала до конца как будто вы разработчик, который видит его впервые.
 - Есть ли всё для воспроизведения?
 - Понятны ли шаги?
 - Нет ли опечаток или двусмысленностей?
- **Шаг 7:** Прочитайте ещё раз через 5 минут. Гарантированно найдёте то, что упустили в первый раз!

Пример упущенной детали:

В шагах было написано «заполнить все поля», но не указано, что поле «Телефон» было оставлено пустым, что и вызвало ошибку.

Типичные ошибки в отчете о дефекте

Оформление и формулировки

1. Плохие краткие описания (Summary)


- **Примеры:** «Неожиданное прерывание», «В приложении красный шрифт».
- **Проблема:** Не отвечают на вопросы *«Что? Где? При каких условиях?»*.
- **Результат:** Разработчик не понимает суть проблемы без дополнительных усилий.

2. Идентичные Summary и Description

- **Ошибка:** Копирование одного и того же текста в оба поля.
- **Правило:** Краткое описание — для быстрого понимания, подробное — для глубокого анализа.

3. Отсутствие ключевой информации

- Что должно быть в описании:
 - Фактический результат
 - Ожидаемый результат
 - Ссылка на требование (если применимо)

 «Приложение показывает содержимое OpenXML-файлов» — это не дефект, если не сказано, что оно не должно этого делать.

4. Игнорирование кавычек и грамматики

- **Пример:** «запись исчезает при наведении мыши» vs. «поле “Запись” исчезает...»
- **Последствия:** Искажение смысла, путаница.

5. Неграмотные формулировки

- **Примеры:**
 - «Поиск не работает нажатием кнопки Enter»
 - «The application doesn't work with the from keyboard...»
- **Совет:** Перечитывайте текст перед отправкой!

6. Лишние шаги в воспроизведении

- **Плохо:** «1. Запустить приложение. 2. Открыть меню...»
- **Хорошо:** Начинайте с момента, непосредственно связанного с дефектом.

7. Плохие скриншоты

- **Ошибки:**
 - Скриншот всего экрана
 - Проблема не выделена
 - Изображения размещены на сторонних сервисах (нарушение конфиденциальности!)
- **Правило:** Используйте `Alt+PrintScreen`, выделяйте проблему, прикрепляйте файлы **только через систему управления дефектами.**

8. Откладывание написания отчёта

- **Последствия:** Забываются важные детали, команда получает информацию с задержкой.
- **Решение:** Пишите отчёт сразу после обнаружения.

9. Выдуманные дефекты

- **Ошибка:** Описание поведения как ошибки, хотя оно соответствует требованиям.
- **Фраза-индикатор:** «Not a bug».
- **Правило:** Всегда сверяйтесь с требованиями!

10. Неверная оценка важности

- **Пример:** Критический дефект (крах приложения) описан как «опечатка в сообщении об ошибке».
- **Совет:** Оценивайте влияние на пользователя, а не на внешний вид.

11. Техническая безграмотность

- **Пример:** «Количество найденных файлов не соответствует реальной глубине вложенности каталога».
- **Итог:** Такой отчёт подрывает доверие к тестировщику.

12. Лишняя информация в шагах


- **Пример:** Указание точного имени файла, размера, диска (`J:\Data\song1.mp3`).
- **Правило:** Указывайте только то, что критично для воспроизведения.

13. Недостающая информация в шагах

- **Пример:** Не указано, что путь содержит два пробела подряд, что и вызывает ошибку.
- **Совет:** Проводите исследование, чтобы найти ключевую деталь.

14. Игнорирование «последовательных дефектов»

- **Ситуация:** Один дефект маскирует другой.
- **Пример:**
 - Дефект 1: Файл повреждается при передаче.
 - Дефект 2: Приложение падает при обработке повреждённого файла.
- **Правило:** Описывайте **оба дефекта**, даже если один зависит от другого.

 **Главный принцип:** Ваш отчёт должен позволить разработчику воспроизвести и понять проблему с первого раза.