

ОГБПОУ «ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ»

УТВЕРЖДАЮ

Заместитель директора по УМР

\_\_\_\_\_/Е.А. Родзик

«\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Методические рекомендации по выполнению  
лабораторно-практической работы №3  
учебной дисциплины

*МДК 05.03 Тестирование ИС*

*Тема: «Анализ и обеспечение обработки исключительных ситуаций»*

г. Томск – 2021 г

РАССМОТРЕНО

на заседании ПЦК

«Информационные системы и  
программирование»

протокол № \_\_\_\_\_

от « \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Председатель ПЦК

\_\_\_\_\_ / Фунтиков М.Н.

Рекомендации разработаны на  
основе Федерального  
государственного образовательного  
стандарта (далее – ФГОС) по  
специальностям среднего  
профессионального образования  
(далее СПО) 09.02.07 –  
«Информационные системы и  
программирование» и в соответствии  
с примерной основной  
образовательной программой.

Организация-разработчик:

ОГБПОУ «Томский техникум информационных технологий»

Разработчики:

\_\_\_\_\_ / Грушевский Ю.В.

## **СОДЕРЖАНИЕ**

1. Общая характеристика лабораторно-практической работы №1	4
2. Краткие теоретические сведения	6
3. Задания для выполнения	12
4. Индивидуальные задания	13
5. Контрольные вопросы	14
6. Рекомендованная литература	15
ПРИЛОЖЕНИЕ А	16
ПРИЛОЖЕНИЕ Б	17

**1. ОБЩАЯ ХАРАКТЕРИСТИКА**  
**ЛАБОРАТОРНО-ПРАКТИЧЕСКОЙ РАБОТЫ №3**  
**Тема: «Анализ и обеспечение обработки исключительных ситуаций»**

**Цель работы:** формирование умений составлять отчеты по найденным дефектам программного обеспечения.

**Проверяемые компетенции:**

ПК 5.5, 5.7; ОК 1 – ОК 11.

**Инструкция по выполнению:**

- 1) Ознакомьтесь с теоретическим материалом, необходимым для выполнения практической работы;
- 2) Выполните предложенные задания для выполнения;
- 3) Выполните предложенные индивидуальные задания в соответствии с вариантом;
- 4) Ответьте на контрольные вопросы;
- 5) Оформите отчет по практической (шаблон представлен в приложении);
- 6) Отправьте отчет на проверку.

**Время выполнения заданий:** 2 часа.

**Критерии оценки:**

№	Критерий	Баллы
1	Отчет соответствует предложенному шаблону	1
2	Баг-репорты индивидуального задания составлены корректно (Штраф 4 балла за каждый ошибочный баг-репорт)	12
3	Представлены ответы на контрольные вопросы (штраф 1 балл за каждый неправильный ответ)	3
4	Оформление отчета соответствует требованиям (шрифт, поля, отступы, интервалы, оформление рисунков, автоматическое оглавление)	2
5	Своевременность выполнения задания	2
Итого		20

**Перевод в пятибалльную систему оценивания:**

«отлично»: > 17 баллов

«хорошо»: 15-17 баллов

«удовлетворительно»: 11-14 баллов

«неудовлетворительно»: < 10 баллов

## 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Баг(дефект)**— расхождение ожидаемого и фактического результата.

**Ожидаемый результат** — поведение системы, описанное в требованиях.

**Фактический результат** — поведение системы, наблюдаемое в процессе тестирования.

**Ошибка** — действие человека, приводящее к некорректным результатам. Этот термин очень часто используют как наиболее универсальный, описывающий любые проблемы («ошибка человека», «ошибка в коде», «ошибка в документации», «ошибка выполнения операции», «ошибка передачи данных», «ошибочный результат» и т.п.). Ошибки, сделанные программистом, известны как «Ошибка», что ,в свою очередь, приводит к ошибке в программе.

**Дефект** — это отклонение от требований спецификаций или ожиданий пользователей. Другими словами, дефект является ошибкой в кодировании или логике, что приводит к сбою программы или созданию неправильного / неожиданного результата. Это могут быть аппаратные средства, программное обеспечение, сеть, производительность, формат или функциональность. Недостаток в компоненте или системе, способный привести к ситуации сбоя или отказа.

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

**Отказ** — событие, заключающееся в нарушении работоспособного состояния объекта. Это неспособность системы или компонента выполнять требуемые функции в рамках определенных требований к производительности. Неисправность возникает при выполнении ошибки.

**Отчёт о дефекте** - это документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Отчёт о дефекте (и сам дефект вместе с ним) проходит определённые стадии жизненного цикла:

- **Обнаружен (submitted)** — начальное состояние отчёта (иногда называется «Новый» (new)), в котором он находится сразу после создания. Некоторые средства также позволяют сначала создавать черновик (draft) и лишь потом публиковать отчёт.
- **Назначен (assigned)** — в это состояние отчёт переходит с момента, когда кто-то из проектной команды назначается ответственным за исправление дефекта. Назначение ответственного может производиться решением лидера команды разработки, коллегиально, по добровольному принципу, иным принятым в команде способом или выполняется автоматически на основе определённых правил.
- **Исправлен (fixed)** — в это состояние отчёт переводит ответственный за исправление дефекта член команды после выполнения соответствующих действий по исправлению.
- **Проверен (verified)** — в это состояние отчёт переводит тестировщик, удостоверившись, что дефект на самом деле был устранён. Как правило, такую проверку выполняет тестировщик, изначально написавший отчёт о дефекте.
- **Закрыт (closed)** — состояние отчёта, означающее, что по данному дефекту не планируется никаких дальнейших действий. Здесь есть некоторые расхождения в жизненном цикле, принятом в разных инструментальных средствах управления отчётами о дефектах.
- **Открыт заново (reopened)** — в это состояние (как правило, из состояния «Исправлен») отчёт переводит тестировщик, удостоверившись, что дефект по-прежнему воспроизводится на билде, в котором он уже должен быть исправлен.
- **Рекомендован к отклонению (to be declined)** — в это состояние отчёт о дефекте может быть переведён из множества других состояний, чтобы вынести на рассмотрение вопрос об отклонении отчёта по той или иной причине. Если рекомендация является обоснованной, то отчёт переводится в состояние «Отклонён».
- **Отклонён (declined)** — в это состояние отчёт переводится в случаях, подробно описанных в пункте «Закрыт»: если средство управления отчётами о дефектах предполагает использование этого состояния вместо состояния «Закрыт» для тех или иных резолюций по отчёту.
- **Отложен (deferred)** — в это состояние отчёт переводится в случае, если исправление дефекта в ближайшее время является

нерациональным или не представляется возможным, однако есть основания полагать, что в обозримом будущем ситуация исправится (выйдет новая версия библиотеки, вернётся из отпуска специалист по необходимой технологии, изменятся требования заказчика и т.д.)

### **Атрибуты отчёта о дефекте**

В зависимости от инструментального средства управления отчётами о дефектах внешний вид их записи может немного отличаться, могут быть добавлены или убраны отдельные поля, но концепция остаётся неизменной.

- **Идентификатор (identifier)** представляет собой уникальное значение, позволяющее однозначно отличить один отчёт о дефекте от другого и используемое во всевозможных ссылках. В общем случае идентификатор отчёта о дефекте может представлять собой просто уникальный номер, но (если позволяет инструментальное средство управления отчётами) может быть и куда сложнее: включать префиксы, суффиксы и иные осмысленные компоненты, позволяющие быстро определить суть дефекта и часть приложения (или требований), к которой он относится.
- **Краткое описание (summary)** должно в предельно лаконичной форме давать исчерпывающий ответ на вопросы «Что произошло?», «Где это произошло?», «При каких условиях это произошло?». Например: «Отсутствует логотип на странице приветствия, если пользователь является администратором». Что произошло? Отсутствует логотип. Где это произошло? На странице приветствия. При каких условиях это произошло? Если пользователь является администратором.
- **Подробное описание (description)** представляет в развёрнутом виде необходимую информацию о дефекте, а также (обязательно!) описание фактического результата, ожидаемого результата и ссылку на требование (если это возможно). В отличие от краткого описания, которое, как правило, является одним предложением, здесь можно и нужно давать подробную информацию. Если одна и та же проблема (вызванная одним источником) проявляется в



нескольких местах приложения, можно в подробном описании перечислить эти места.

- **Шаги по воспроизведению (steps to reproduce, STR)** описывают действия, которые необходимо выполнить для воспроизведения дефекта. Это поле похоже на шаги тест-кейса, за исключением одного важного отличия: здесь действия прописываются максимально подробно, с указанием конкретных вводимых значений и самых мелких деталей, т.к. отсутствие этой информации в сложных случаях может привести к невозможности воспроизведения дефекта.
- **Воспроизводимость (reproducibility)** показывает, при каждом ли прохождении по шагам воспроизведения дефекта удаётся вызвать его проявление. Это поле принимает всего два значения: всегда (always) или иногда (sometimes). Можно сказать, что воспроизводимость «иногда» означает, что тестировщик не нашёл настоящую причину возникновения дефекта. Тестировщику нужно потратить много времени на то, чтобы удостовериться в наличии дефекта (т.к. однократный сбой в работе приложения мог быть вызван огромным количеством посторонних причин). Разработчику тоже нужно потратить время, чтобы добиться проявления дефекта и убедиться в его наличии. После внесения исправлений в приложение разработчик фактически должен полагаться только на свой профессионализм, т.к. даже многократное прохождение по шагам воспроизведения в таком случае не гарантирует, что дефект был исправлен (возможно, через ещё 10–20 повторений он бы проявился).
- **Важность (severity)** показывает степень ущерба, который наносится проекту существованием дефекта. В общем случае выделяют следующие градации важности:
  - **критическая (critical)** — существование дефекта приводит к масштабным последствиям катастрофического характера: потеря данных, раскрытие конфиденциальной информации, нарушение ключевой функциональности приложения и т.д.
  - **высокая (major)** — существование дефекта приносит ощутимые неудобства многим пользователям в рамках их типичной деятельности: недоступность вставки из буфера

обмена, неработоспособность общепринятых клавиатурных комбинаций, необходимость перезапуска приложения при выполнении типичных сценариев работы.

- **средняя (medium)** — существование дефекта слабо влияет на типичные сценарии работы пользователей, и/или существует обходной путь достижения цели, например: диалоговое окно не закрывается автоматически после нажатия кнопок «ОК»/«Cancel», при распечатке нескольких документов подряд не сохраняется значение поля «Двусторонняя печать», перепутаны направления сортировок по некоему полю таблицы.
- **низкая (minor)** — существование дефекта редко обнаруживается незначительным процентом пользователей и почти не влияет на их работу, например: опечатка в глубоко вложенном пункте меню настроек, некое окно сразу при отображении расположено неудобно (нужно перетянуть его в удобное место), неточно отображается время до завершения операции копирования файлов.
- **Срочность (priority)** показывает, как быстро дефект должен быть устранён. В общем случае выделяют следующие градации срочности:
  - **наивысшая (ASAP, as soon as possible)** срочность указывает на необходимость устранить дефект настолько быстро, насколько это возможно. В зависимости от контекста «насколько быстро, насколько возможно» может варьироваться от «в ближайшем билде» до единиц минут.
  - **высокая (high)** срочность означает, что дефект следует исправить вне очереди, т.к. его существование или уже объективно мешает работе, или начнёт создавать такие помехи в самом ближайшем будущем.
  - **обычная (normal)** срочность означает, что дефект следует исправить в порядке общей очерёдности. Такое значение срочности получает большинство дефектов.
  - **низкая (low)** срочность означает, что, в обозримом будущем, исправление данного дефекта не окажет существенного влияния на повышение качества продукта.

- **Фактический результат (actual result)** - результат, полученный после прохождения шагов к воспроизведению.
- **Ожидаемый результат (expected result)** - описывает ожидаемое поведение ПО после прохождения шагов к воспроизведению.
- **Симптом (symptom)** — позволяет классифицировать дефекты по их типичному проявлению. Не существует никакого общепринятого списка симптомов. Более того, далеко не в каждом инструментальном средстве управления отчётами о дефектах есть такое поле, а там, где оно есть, его можно настроить. В качестве примера есть следующие значения симптомов дефекта: Косметический дефект (cosmetic flaw), Повреждение/потеря данных (data corruption/loss), Проблема в документации (documentation issue), Некорректная операция (incorrect operation), Проблема инсталляции (installation problem), Ошибка локализации (localization issue), Нереализованная функциональность (missing feature), Проблема масштабируемости (scalability), Низкая производительность (low performance), Крах системы (system crash), Неожидаемое поведение (unexpected behavior), Недружественное поведение (unfriendly behavior), Расхождение с требованиями (variance from specs), Предложение по улучшению (enhancement).
- **Комментарий (comments, additional info)** — может содержать любые полезные для понимания и исправления дефекта данные. Иными словами, сюда можно писать всё то, что нельзя писать в остальные поля.
- **Приложения (attachments)** — представляет собой не столько поле, сколько список прикрепленных к отчёту о дефекте приложений (копий экрана, вызывающих сбой файлов и т.д.)

### **3. ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ**

Не предусмотрены

#### **4. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ**

На основании проведенного вами тестирования с помощью чек-листа (ЛПР №1) оформите баг-репорт по выявленным вами дефектам.

Минимальное количество багов для которых необходимо составить баг-репорт - 5.

## **5. КОНТРОЛЬНЫЕ ВОПРОСЫ**

- Что такое дефект? Какие последствия присутствия дефекта могут быть?
- Чем баг отличается от “фичи”?
- Для чего необходимо на выявленные дефекты оформлять баг-репорты?

## **6. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА**

1. Курс лекций «Тестирование программного обеспечение»  
[Электронный ресурс] – Режим доступа:  
<https://sergeygavaga.gitbooks.io/kurs-lektsii-testirovanie-programnogo-obespecheni/content/lektsiya-2-ch4-vidi-i-napravleniya-testirovaniya.html>  
Дата обращения: 08.09.2021

**Требования к отчету**

Общие требования:

- 1) Шрифт – Times New Roman, 14 пт.
- 2) Интервалы: междустрочный – 1,5 строки, интервал до и после абзаца – 0 пт.
- 3) Отступ первой строки – 1,25
- 4) Рисунки и подписи к ним выравниваются по центру.

Требования к структуре отчета:

- 1) Титульный лист
- 2) Содержание
- 3) Основная часть:
  - 1.1. Отчет по решению заданий для выполнения
  - 1.2. Отчет по решению индивидуальных заданий
- 4) Выводы по работе
- 5) Ответы на контрольные вопросы



**ОГБПОУ «ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ»**

**Отчет по лабораторно-практической работе №3**

**учебной дисциплины**

***МДК 05.03 Тестирование ИС***

***Тема: «Анализ и обеспечение обработки исключительных ситуаций»***

**Выполнил:**

**студент \_\_\_\_\_ группы**

\_\_\_\_\_

**Проверил:**

**преподаватель, высшая к.к.**

**Грушевский Ю.В.**

**г. Томск – 2021 г**