

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ТЕСТИРОВЩИК**

**РЕФЕРАТ**

студента 1 курса 151 группы  
направления 09.03.04 — Программная инженерия  
факультета КНиИТ  
Иванова Александра Владимировича

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ .....  | 3  |
| 1 Кто такой тестировщик и чем он занимается? .....                  | 4  |
| 1.1 Определение профессии тестировщик.....                          | 4  |
| 1.2 В чём заключается работа тестировщика? .....                    | 4  |
| 2 Востребованность тестировщика .....                               | 5  |
| 3 Тестировщики .....  | 6  |
| 3.1 Уровни тестировщиков.....                                       | 6  |
| 3.2 Типы тестировщиков .....  | 6  |
| 4 Тестирование .....  | 9  |
| 4.1 Этапы работы тестировщика .....                                 | 9  |
| 4.2 Способы тестирования .....                                      | 9  |
| 4.2.1 Тестирование ПО и сервисов вручную .....                      | 9  |
| 4.2.2 Автоматическое тестирование .....                             | 9  |
| 4.2.3 Как автоматизировать тесты?.....                              | 9  |
| 4.3 Виды тестирования.....  | 10 |
| 4.4 Основные стратегии тестирования .....                           | 12 |
| 4.4.1 Стратегия белого ящика .....                                  | 12 |
| 4.4.2 Стратегия чёрного ящика .....                                 | 12 |
| 4.4.3 Стратегия серого ящика .....                                  | 12 |
| 5 Зарплата, обязанности и требования для работы тестировщиком ..... | 14 |
| 5.1 Зарплата .....  | 14 |
| 5.2 Обязанности .....   | 14 |
| 5.3 Требования .....  | 14 |
| 6 Как выучиться на тестировщика? .....                              | 15 |
| ЗАКЛЮЧЕНИЕ .....  | 16 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....                              | 17 |

## **ВВЕДЕНИЕ**

Позиция тестировщика считается одним из самых простых способов быстро войти в отрасль информационных технологий. Ходят слухи, что эту профессию легко освоить, работа у тестировщика несложная, да и платят специалисту по тестированию программного обеспечения почти как программисту. Давайте разберёмся в достоверности этой информации.

## **1 Кто такой тестировщик и чем он занимается?**

### **1.1 Определение профессии тестировщик**

Тестировщик — это специалист, который занимается проверкой работоспособности десктопных программ и мобильных приложений. В его задачи входит мониторинг ошибок в различных сценариях работы ПО, когда нужно убедиться, что программа отвечает на пользовательские действия так, как было изначально спланировано. А еще обеспечение информационной безопасности, проверка работоспособности в разных браузерах, операционных системах и устройствах.

### **1.2 В чём заключается работа тестировщика?**

Тестировщик анализирует, выполняет тестирование по сценариям и придумывает, где еще можно найти ошибки.

Если он находит такие ошибки (а находит обязательно — в этом его работа) он пишет об этом специальный отчет, по которому программисты устраняют ошибки [1].

## **2 Востребованность тестировщика**

Профессия тестировщика появилась не так давно: найти сейчас человека, допустим с 10-тилетним стажем работы, практически невозможно. Считается, что первыми тестерами были специалисты по качеству — сотрудники, которые проверяли программное обеспечение (ПО) на соответствие ГОСТам. Однако обязанности современного тестировщика не ограничиваются простой сверкой определенных показателей: тестирование сегодня — сложная, многоступенчатая работа, которая сопровождает процесс создания программного продукта с начала и до конца.

Что послужило толчком для появления новой профессии — тестировщик ПО? Все просто: постепенно в компаниях-разработчиках осознали, что ключевой составляющей успеха любого программного продукта является, в первую очередь, его качество. В результате, если еще несколько лет назад эта профессия считалась редкой и не очень востребованной, то сегодня спрос на тестировщиков на порядок превышает предложение (по некоторым оценкам, на одного соискателя приходится около 20 предложений от работодателей).

Тестировщик ПО может работать в IT-компаниях, разрабатывающих программное обеспечение и интернет-сайты, в составе независимых групп тестирования или в компаниях, занимающихся исключительно тестированием ПО: в любых организациях, внедряющих собственные системы автоматизации (например, многие банки имеют свои отделы разработки и тестирования ПО) [2].

## 3 Тестировщики

### 3.1 Уровни тестировщиков

1. **QA-analyst (аналитик)** — старт карьеры тестировщика. На этом этапе начинающий специалист владеет базовыми принципами, знаниями и навыками тестирования. В его обязанности входит относительно небольшой список задач: владение основным программным обеспечением, ручной поиск багов, выполнение тест-кейсов, ведение отчётности.
2. **QA-engineer (инженер)** — вторая ступень развития тестировщика. Помимо выполнения тест-кейсов, тестировщик игр или приложений такого уровня выполняет ряд других задач: проверка требования к продукту, оценка рисков, создание идеи по улучшению итогового качества цифрового продукта, разработка планов тестирования, анализ результатов тестирования.
3. **Автоматизатор** — тестировщик 3 уровня способен заменить сразу нескольких специалистов, что значительно ускоряет и упрощает работу компании, а также экономит ресурсы. Кроме глубоких знаний в области тестирования, автоматизатор уже владеет языками программирования: Python, Java, JavaScript, C, C++ и прочие.
4. **Тест разработчик (SDET)** — наиболее сложная и высокооплачиваемая профессия в сфере тестирования. Разработчик обязан владеть несколькими языками программирования, иметь большой опыт работы, полностью понимать нюансы тестируемой системы, корректировать действия других тестировщиков. В обязанности входят следующие задачи: установка, настройка и коррекция работы ПО, владение инструментами автоматизации тестирования, разработка тест-кейсов, разработка и внедрение вспомогательных средств тестирования, поддержка архитектуры автоматизации, конфигурация среды тестирования, менеджмент (управление персоналом).

### 3.2 Типы тестировщиков

- **Тестировщик мобильных приложений (мобильные тестировщики).** Мобильные тестировщики очень востребованы независимо от сферы. Это могут быть приложения для продаж товаров, услуг, рекламы, разработки ПО, развлекательные приложения и многое другое. Тестировщик при-

ложений выполняет тест-кейсы, отправляет отчеты аналитикам и разработчикам, чаще всего, выполняет несложные инструкции в альфа и бета версиях.

- **Тестировщик веб-приложений.** Веб-приложения — особый вид приложений, которые запускаются и управляются с помощью браузера. Продукт имеет визуально-интерфейсную часть (взаимодействует с пользователем) и клиентскую часть (сервер). Веб-приложения отличаются от веб-сайтов рядом параметров, их разработка относится к среднему сегменту по сложности. Тестировщик веб-приложений довольно востребованная профессия, особенно в крупных компаниях.
- **Тестировщик сайтов.** Сайт сегодня есть практически у любого продукта (компании), соответственно, тестировщик сайтов — как никогда актуальная специальность. От качества работы сайта во многом зависит удобство конечного пользователя, а значит и заинтересованность в регулярном посещении. Тестировщик сайтов с нуля может быстро внедриться в структуру компании, сначала выполнять более узконаправленные задачи, а затем масштабироваться. Тестировщик веб-сайтов может легко дорасти от регионального сайта до крупнейших корпораций (Google, Meta, Apple). Обучение на тестировщика сайтов в среднем занимает около 8–10 недель.
- **Тестировщик баз данных.** Тестировщик баз данных работает с огромным количеством информации. Он проводит структурные, функциональные и нефункциональные тесты, проверяет адекватность отображения информации, обновления записей в таблицах, свойства ACID-операций, целостность и сохранность данных, соответствие БД регламенту проекта. Базы данных являются важнейшей частью работы ПО, поэтому их безошибочная работоспособность в целом определяет цифровой продукт. Отметим, что тестировщик баз данных — это обычно специалист с математическим складом ума.
- **Тестировщик игр.** Тестировщик игр дома — «заветная мечта» многих молодых людей. Стоит понимать, тестировщик мобильных игр (мобильные тестировщики) или видеоигр для PC и консолей — это не круглосуточный хаотичный гейминг, а скрупулезная работа по выявлению багов (от очевидных до мельчайших). Тестировщик игр без опыта может начать свой путь с тестирования небольших игровых проектов для PC или мо-

бильных устройств (например, тестировщик android игр). В современной игровой индустрии наиболее популярными являются продукты с мультиплеером (онлайн-игры). В них происходит регулярное обновление/фиксы багов/добавление контента, все это требует серьезной командной работы IT-специалистов. Отдельно можно выделить и другого узкого специалиста — тестировщик игровых сценариев. Вывод, тестировщик игр — профессия относительно молодая, но очень перспективная.

- **Тестировщик интерфейсов.** Интерфейс — одна из главных составляющих цифрового продукта. Это именно то, на что смотрит пользователь: окна, меню, виджеты, вкладки, иконки и т.д. Тестировщик интерфейсов не занимается разработкой визуальных эффектов приложения или сайта, он проверяет работоспособность и оценивает адекватность визуализации продукта в той или иной ситуации (например, в браузере, на мобильном устройстве, планшете). Тестировщик интерфейсов также проверяет шрифты, цвета, формы, функциональное взаимодействие пользователя с интерфейсом.
- **Тестировщик программного обеспечения.** Тестировщик программного обеспечения — очень широкое понятие. Под ПО можно понимать практически любой цифровой продукт — сайт, приложение, игра и многое другое. Часто работодатели предлагают вакансию «тестировщик программного обеспечения» для поиска профессионалов с определенным опытом работы, владеющих сразу несколькими узкими специальностями [3].



## **4 Тестирование**

### **4.1 Этапы работы тестировщика**

1. Изучение документации по продукту: инструкции и рекомендации от разработчиков и продуктовых аналитиков.
2. Составление тест-кейсов для тестирования: какие функции нужно проверить и в какой последовательности, с учетом всех возможных сценариев поведения пользователя.
3. Тестирование.
4. Сбор и анализ полученных результатов: ошибки, сбои, некорректная работа, уязвимости.
5. Оформление полученных результатов в виде отчета с рекомендациями для разработчиков [4].

### **4.2 Способы тестирования**

#### **4.2.1 Тестирование ПО и сервисов вручную**

Тестирование ПО и сервисов вручную — когда специалист сам проходит все этапы работы с продуктом. Для этого он тестирует его в разных операционных системах и браузерах, а также на разных устройствах. Это самый дорогой и долгий способ.

#### **4.2.2 Автоматическое тестирование**

Автоматическое тестирование — с помощью автоматизированных инструментов. Подходит, когда есть готовый набор параметров для проверки, которые слишком долго перебирать вручную или же их применяют сразу для нескольких версий одного и того же продукта. Этот способ — более дешевый и быстрый, но все равно требует контроля со стороны тестировщика. Как правило, таким способом проверяют наиболее критичные функции — такие, как обработка платежей или защита персональных данных.

#### **4.2.3 Как автоматизировать тесты?**

Для автоматизации тестов прежде всего необходимо написать их программными средствами с использованием среды тестирования, которая подходит для вашего приложения. В качестве примера для PHP, Javascript и Ruby можно привести такие среды тестирования, как PHPUnit, Mocha, RSpec соответственно. Существует множество других вариантов для всех языков. Вы можете

самостоятельно поискать информацию и обратиться за помощью к сообществам разработчиков, чтобы выяснить, какая из сред тестирования оптимально подойдет в вашем случае.

Если тесты могут быть запущены как скрипт с вашего терминала, можно настроить их автоматический запуск сервером непрерывной интеграции, например Bamboo, или облачным сервисом, таким как Bitbucket Pipelines. Эти инструменты будут отслеживать состояние репозитория и запускать соответствующий комплект тестов каждый раз, когда в главном репозитории фиксируются изменения [5].

### 4.3 Виды тестирования

- **Модульные тесты** — тесты, работающие на очень низком уровне, близко к исходному коду приложения. Они заключаются в тестировании отдельных методов и функций классов, компонентов или модулей, используемых в ПО. Модульные тесты, как правило, не требуют больших расходов на автоматизацию и могут выполняться сервером непрерывной интеграции очень быстро.
- **Интеграционные тесты** — тесты, в ходе которых проверяется, хорошо ли работают вместе различные модули и сервисы, используемые приложением. Например, можно протестировать взаимодействие с базой данных или убедиться, что микросервисы работают вместе так, как задумано. Этот вид тестирования является более затратным, поскольку для проведения тестов требуется запуск различных компонентов приложения.
- **Функциональные тесты.** В функциональных тестах основное внимание уделяется бизнес-требованиям к приложению. Они проверяют только результат некоторого действия и не проверяют промежуточные состояния системы при выполнении этого действия. Иногда возникает путаница между понятиями интеграционных и функциональных тестов, так как и те и другие требуют взаимодействия нескольких компонентов друг с другом. Разница в том, что интеграционный тест нужен просто чтобы убедиться, что вы можете отправлять запросы к базе данных, тогда как функциональный тест будет ожидать получения из базы данных определенного значения в соответствии с требованиями продукта.
- **Сквозные тесты.** Сквозное тестирование копирует поведение пользователя при работе с ПО в контексте всего приложения. Оно обеспечива-

ет контроль того, что различные схемы действий пользователя работают должным образом. Сценарии могут быть как очень простыми (загрузка веб-страницы или вход в систему), так и гораздо более сложными (проверка почтовых уведомлений, онлайн-платежей и т.д.). Сквозные тесты очень полезны, но их выполнение обходится довольно дорого, к тому же, когда они автоматизированы, такие тесты тяжело обслуживать. Рекомендуется иметь в наличии несколько основных сквозных тестов и активнее полагаться на более низкие уровни тестирования (модульные и интеграционные тесты), чтобы получать возможность быстро выявлять критические изменения.

- **Приёмочное тестирование.** Приемочные тесты — это формальные тесты, которые проверяют, отвечает ли система требованиям бизнеса. При этом во время тестирования должно быть запущено само приложение, и основное внимание уделяется воспроизведению поведения пользователей. В ходе этого тестирования возможен даже замер производительности системы, и в случае несоответствия установленным требованиям внесенные изменения могут быть отклонены.
- **Тестирование производительности.** В тестах производительности оценивается работа системы при определенной рабочей нагрузке. С помощью таких тестов можно оценить надежность, скорость, масштабируемость и отзывчивость приложения. Например, это может быть наблюдение за временем отклика при выполнении большого количества запросов или определение поведения системы при работе со значительными объемами данных. Этот вид тестирования позволяет определить, соответствует ли приложение требованиям к производительности, найти узкие места, оценить стабильность при пиковом трафике и многое другое.
- **Smoke-тестирование.** Smoke-тесты — это базовые тесты, которые проверяют основные функциональные возможности приложения. Они должны выполняться быстро, поскольку цель таких тестов — убедиться, что основные возможности системы работают как запланировано. Smoke-тесты полезно запускать сразу после создания новой сборки (для определения, можно ли запускать более ресурсоемкие тесты) или сразу после развертывания (чтобы убедиться, что приложение работает правильно в новой, только что развернутой среде) [6].

## **4.4 Основные стратегии тестирования**

### **4.4.1 Стратегия белого ящика**

Тестирование по стратегии белого ящика — тестирование кода на предмет логики работы программы и корректности её работы с точки зрения компилятора того языка, на котором она писалась. Тестирование по стратегии белого ящика, также называемое техникой тестирования, управляемой логикой программы, позволяет проверить внутреннюю структуру программы. Исходя из этой стратегии, тестировщик получает тестовые данные путем анализа логики работы программы.

### **4.4.2 Стратегия чёрного ящика**

Тестирование чёрного ящика или поведенческое тестирование — стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций. При тестировании чёрного ящика, тестировщик имеет доступ к программе только через те же интерфейсы, что и заказчик или пользователь, либо через внешние интерфейсы, позволяющие другому компьютеру либо другому процессу подключиться к системе для тестирования. Например, тестирующий модуль может виртуально нажимать клавиши или кнопки мыши в тестируемой программе с помощью механизма взаимодействия процессов, с уверенностью в том, все ли идёт правильно, что эти события вызывают тот же отклик, что и реальные нажатия клавиш и кнопок мыши. Как правило, тестирование чёрного ящика ведётся с использованием спецификаций или иных документов, описывающих требования к системе. Обычно в данном виде тестирования критерий покрытия складывается из покрытия структуры входных данных, покрытия требований и покрытия модели (в тестировании на основе моделей) [7].

### **4.4.3 Стратегия серого ящика**

Проверка серого ящика — специальный метод тестирования программного обеспечения с неполным знанием его внутреннего устройства. Чтобы выполнить подобный вид тестов, не нужно иметь доступ к исходному коду ПО.

Все тесты создаются на базе простого знания алгоритмов, архитектуры и иных высокоуровневых характеристик поведения продукта.

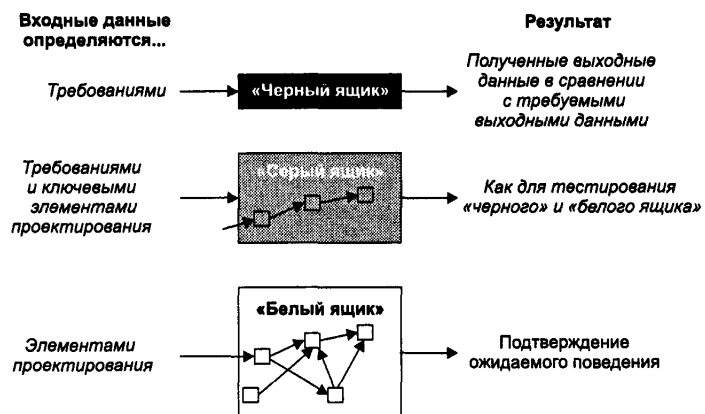


Рисунок 1 – Чёрный ящик, серый ящик, белый ящик.

## 5 Зарплата, обязанности и требования для работы тестировщиком

### 5.1 Зарплата

В июне 2022 года сервис SuperJob опубликовал исследование, посвящённое тому, сколько в среднем получают тестировщики и QA-инженеры в крупных городах России. Самая большая средняя зарплата — в Москве (150 000 рублей в месяц), а самая маленькая — в Волгограде (105 000 рублей в месяц).

| Средняя зарплата специалистов<br>по тестированию в городах России |           |                 |           |
|---|-----------|-----------------|-----------|
| Москва  | 150 000 ₽ | Нижний Новгород | 115 000 ₽ |
| Санкт-Петербург   | 125 000 ₽ | Новосибирск     | 115 000 ₽ |
| Волгоград   | 105 000 ₽ | Омск            | 110 000 ₽ |
| Воронеж   | 110 000 ₽ | Пермь           | 110 000 ₽ |
| Екатеринбург  | 120 000 ₽ | Ростов-на-Дону  | 115 000 ₽ |
| Казань  | 110 000 ₽ | Самара          | 110 000 ₽ |
| Краснодар   | 115 000 ₽ | Уфа             | 105 000 ₽ |
| Красноярск  | 110 000 ₽ | Челябинск       | 110 000 ₽ |

Источник данных: SuperJob

Рисунок 2 – Средние зарплаты в крупных городах России.

### 5.2 Обязанности

- Обнаружение, документирование и отслеживание дефектов.
- Взаимодействие с разработчиками, аналитиками.
- Работа с автосетами — запуск, разбор падений, написание.
- Участие в проработке задач [8].

### 5.3 Требования

- Знание методов и методик тестирования.
- Опыт тестирования веб-приложений.
- Опыт работы в командной строке linux-подобных операционных систем.
- Знание SQL.
- Опыт работы с системами контроля версий.
- Опыт работы с системами учёта багов [9].

## **6 Как выучиться на тестировщика?**

1. **Самостоятельное обучение.** Это полностью бесплатный путь, где сами себе избираете количество материалов и интенсивность познания профессии. Тем не менее, придется прочесть немало специализированной литературы, почитать ветки профильных форумов и посмотреть ролики на YouTube. Так или иначе, базу вы получите, но надо двигаться дальше, чтобы практиковаться и набираться реального опыта. Как минимум стажером на низкую зарплату в компанию могут взять.
2. **Обучение в высшем учебном заведении.** Стоит ли говорить, что это долгий путь? Тем не менее, это наиболее системное обучение, где помимо основного навыка вы получите кучу дополнительных и сможете вполне себе работать не только тестировщиком, но и программистом или QA-инженером.
3. **Прохождение онлайн-курсов.** Это наиболее быстрый способ получить навык в профессии, но курсы, однако, бывают откровенным надувательством и те грамоты и сертификаты, которые они выдают по окончании, для работодателя значат чуть больше, чем ничего [10].

## ЗАКЛЮЧЕНИЕ

Для того, чтобы стать тестировщиком нужно иметь очень много знаний. Важность тестировщика сложно переоценить, потому что хороший тестировщик должен не только найти ошибку, но и помочь разработчикам устранить её, а для этого нужно обладать soft-скиллами. Я бы точно не сказал, что тестирование — это лёгкий путь в сферу IT.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Сухорукова, Н. В. Тестирование: вчера, сегодня, завтра / Н. В. Сухорукова // *Вестник ТГПУ*. — 2011. — Vol. 2. <https://cyberleninka.ru/article/n/testirovanie-vchera-segodnya-zavtra/viewer>.
- 2 Jones, A. The future of software testing / A. Jones. — QASymphony, 2017. <https://angiejones.tech/the-future-of-software-testing/>.
- 3 Виды тестировщиков [Электронный ресурс]. — 2022. — URL: <https://testpro.io/ru/vidy-testirovshhikov/> (Дата обращения 06.05.2023). Загл. с экр. Яз. рус.
- 4 Блэк, Р. Ключевые процессы тестирования: планирование, подготовка, проведение, совершенствование / Р. Блэк. — Лори, 2006. — С. 544. ISBN5-85582-239-7.
- 5 Fewster, M. Software test automation: effective use of test execution tools / M. Fewster, D. Graham. — ACM Press/Addison-Wesley Publishing Co. 1515 Broadway, 17th Floor New York, NY United States, 1999. — P. 574. ISBN978-0-201-33140-0.
- 6 Различные виды тестирования ПО [Электронный ресурс]. — 2023. — URL: <https://www.atlassian.com/ru/continuous-delivery/software-testing/types-of-software-testing> (Дата обращения 01.05.2023). Загл. с экр. Яз. рус.
- 7 Тестирование белого и чёрного ящика [Электронный ресурс]. — 2023. — URL: <https://unetway.com/tutorial/testirovanie-belogo-asika-i-chernogo-asika> (Дата обращения 07.05.2023). Загл. с экр. Яз. рус.
- 8 Профессия тестировщик: какая зарплата у тестировщика и что нужно уметь? [Электронный ресурс]. — 2023. — URL: <https://myacademy.ru/baza-znaniy/stati/professiya-tester> (Дата обращения 06.05.2023). Загл. с экр. Яз. рус.
- 9 Савин, Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах / Р. Савин. — Издательство «Дело», 2007. — С. 312. ISBN978-5-7749-0460-0.

- 10 Кто такой тестировщик: чем он занимается, сколько зарабатывает и как им стать [Электронный ресурс]. — 2022. — URL: <https://kokos.com/blog/kto-takoj-testirovshchik/> (Дата обращения 06.05.2023). Загл. с экр. Яз. рус.