

### Лекция 3. Факторы, влияющие на эффективность и качество тестирования. Функциональные обязанности специалиста по тестированию ПП

#### 3.1. Факторы, влияющие на эффективность и качество тестирования

Понятие **эффективность** (лат. исполнение, действие) имеет два смысла: 1) это продуктивность использования ресурсов в достижении какой-либо цели, например, убить муху из пушки (действительно с точки зрения результата, но не эффективно с точки зрения использования ресурсов для достижения намеченной цели); 2) это способность выполнять работу и достигать необходимого или желаемого результата с наименьшей затратой времени и усилий например, убить муху с помощью мухобойки (эффективно с обеих позиций).

Термин эффективность может применяться в отношении производства продукта (экономическая эффективность) и в отношении работы организации.

**Экономическая эффективность** – это соотношение полезного результата и затрат факторов производственного процесса. Для измерения эффективности производства используются показатели производительности труда, фондоотдачи, рентабельности, прибыльности, окупаемости и др. С их помощью сопоставляются различные варианты развития производства, решения его структурных проблем.

**X-эффективность** в экономике – концепция эффективности функционирования фирмы, введена Х.Лейбенштайном. Присуща фирме, производящей максимально возможный объём продукции при имеющемся наборе ресурсов и наилучшей из доступных технологий.

Термин «**эффективность тестирования**» может относиться как к ПП, так и к процессу обнаружения ошибок. В обоих случаях под эффективностью **понимают** выявление при минимальных затратах времени и средств наибольшего количества ошибок в ПП.

Разработаны и применяются **метрики**, позволяющие оценить эффективность и качество работы тестировщиков, определить уровень зрелости команды обеспечения качества – QA(Quality Assurance)-команды. К ним относятся:

1. *Эффективность тестов и тестовых наборов* = Кол-во обнаруженных ошибок / Кол-во кейсов в тестовом наборе.

Метрика отражает не только эффективность тест-кейсов, но и качество тест дизайна, его изменения.

2. *Коэффициент ошибок, пропущенных на продуктив* = Кол-во ошибок обнаруженных после выпуска релиза / Общее кол-во ошибок обнаруженных в процессе тестирования и после выпуска релиза.

Значение коэффициента не должно превышать 0,1. Но и эта цифра говорит о том, что 10% дефектов не выявлено во время тестирования.

3. *Реальное время работы QA-команды* = Отношение времени потраченного командой непосредственно на QA активности (анализ, дизайн, оценки, тестирование, рабочие встречи и др.) к общему времени работы.

Если этот коэффициент не ниже 0,5-0,6, то работу команды считают эффективной.

4. *Доля неподтвержденных (отклоненных) дефектов* = Число дефектов не принятых к исправлению/ Общее кол-во зарегистрированных дефектов. Если эта доля превышает 20%, то в команде возможны расхождения в понимании термина «дефект».

*А также:* плотность дефектов, приходящаяся на отдельный модуль в течение итерации или релиза, коэффициент повторно открытых дефектов, средняя стоимость исправления дефекта, среднее время жизни дефекта и другие.

Важными показателями **качества тестирования** являются: *оценка покрытия* программы (тест-стратегии, тест-плана) избыточными тестами, число ошибок в программе (*процент обнаруженных ошибок*). Существуют методики оценки числа первоначальных ошибок в программном продукте (методики Миллса, Холстеда и др.). Считается, что после выбора языка программирования потенциальное количество ошибок можно оценить даже до написания программы.

**Факторы**, влияющие на эффективность и качество тестирования можно разбить на четыре группы:

1. Факторы, связанные с личными качествами **специалиста по тестированию** (команды тестировщиков), начальным уровнем их знаний, стажем работы, опытом посещения курсов повышения квалификации, здоровьем и т.п.

2. Факторы, связанные с **объектом тестирования** (это программы, модули, интерфейс, сборка, документация на ПП, требования к ПП и др.), его размером, сложностью и т.п.

3. Факторы, связанные с **технологией (процессами) тестирования**. Компонента «технология» имеет отношение к применяемым методикам тестирования, контроля и корректирования процесса тестирования, которые могут повысить эффективность тестов.

4. Факторы, связанные с **оборудованием**, применяемым при тестировании. Компонента «оборудование» связана с используемыми техническими и программными средствами, а также наличием необходимых пособий, дополнительного оснащения.

Рассмотрим их подробнее. Начнём с обсуждения требований, предъявляемых к специалисту по тестированию.

**Тестировщик** (тестер, специалист по тестированию) – специалист, принимающий участие в тестировании программного обеспечения с целью выявления ошибок в его работе и осуществляющий контроль качества продукта после его разработки.

### 3.2. Функциональные обязанности тестировщика ПО

Функции тестировщиков могут сильно различаться в зависимости от многих факторов. Тем не менее можно составить список работ, которые обычно возлагаются на специалистов по тестированию. В их числе:

#### 1. Анализ предметной области.

Тестировщик должен изучить документацию на предназначенный для тестирования программный продукт, сам ПП, область его применения (какие задачи, для каких бизнес-процессов, на каком оборудовании, какова квалификация пользователей). Это позволит ему смоделировать ситуации, которые могут возникнуть в условиях эксплуатации программного обеспечения.

Моделируются обычно три линии поведения пользователя (табл. 3.1): ленивый, добросовестный и злонамеренный пользователь. В случае разграничения прав доступа количество возможных моделей поведения существенно увеличивается. Например, ленивый, добросовестный и злонамеренный гость, ленивый, добросовестный и злонамеренный сотрудник, ленивый, добросовестный и злонамеренный начальник.

Таблица 3.1 – Модели поведения пользователей

<b>Пользователь</b>	<b>Линия поведения пользователя при работе с ПП</b>	<b>Задачи тестировщика</b>
Ленивый	Инструкций не читает, следует интуиции и стереотипам	Найти несоответствия интерфейса программы существующим стереотипам.
Добросовестный	Действует в строгом соответствии с инструкциями.	Найти ошибки в логике работы программы и в документации на программу.
Злонамеренный	Использует программу непредусмотренным способом	Проверить способность ПП противостоять деструктивным действиям пользователя

**2. Разработка и документирование планов тестирования.** **Тест-план** – это документ, описывающий и регламентирующий перечень необходимых работ по тестированию, а также соответствующие техники и подходы, стратегию, области ответственности, критерии, ресурсы, возможные риски, расписание и ключевые даты.

**3. Определение требований к тестам, разработка тестовых процедур, проектирование и документирование тестов.** Тестировщик создаёт и использует разнообразные наборы входных данных, предусмотренные и не предусмотренные авторами программы. Эта информация фиксируется в *тест-кейсах*, *чек-листах* др. документах

**Чек-лист** – документ (контрольный список, набор идей), содержащий перечень того, что нужно проверить. Список может быть многоуровневым. Анализируя продукт, тестировщик выписывает его основные особенности (необычные возможности ПП, особые функции, то, что привлекает внимание), действия, их параметры. При необходимости он делает пометки «вникнуть глубже».

Чек-лист позволяет не забывать о важных тестах, фиксировать результаты своей работы и отслеживать статистику о статусе программного продукта, его готовность к выпуску. Каждый тестировщик по-своему проходит чек-лист, расширяя тестовый набор своей экспертизой. Чек-листы можно и НУЖНО согласовывать с разработчиками, аналитиками. Когда вся команда включается в процесс тестирования, результативность тестирования повышается.

Пример чек-листа для тестирования игрового приложения (позитивные тесты) представлен на рисунке 3.1.

**Преимущества** чек-листов по сравнению с тест-кейсами:

- сокращение затрат на содержание и поддержку тестов: их не надо писать;
- возможность проходить и комбинировать тесты по-разному, в зависимости от предпочтений сотрудников;
- расширение тестового покрытия за счёт отличий при прохождении;
- нивелирование эффекта пестицида (тесты на основе детальных тест-кейсов перестают находить баги, зато шаг влево, шаг вправо - ошибки);
- отсутствие рутины.

**Недостатки** чек-листов: отсутствие достаточно подробной документации по тестированию ПП не позволяет а) эффективно проводить тесты начинающим

тестировщикам, б) использовать чек-листы для обучения начинающих сотрудников, в) достичь взаимопонимания с заказчиком и руководством.

$f_x$			
	A	B	
1		Группы	Кейсы
2	№1	Установка приложения	1 установка из оф.источников Google Play/App Store
3			2 установка из APK файла
4			2.1 установка в память телефона
5			2.2 установка в флеш накопитель телефона
6	№2	Интернет соединение для игры	1 запуск с выключенным интернет соединением
7			2 запуск с EDGE соединением
8			3 запуск с 3g соединением
9			4 запуск с 4g/LTE соединением
10			5 запуск с WiFi соединением
11			6 игра при смене соединения с WiFi на 3g/4g/edge
12			7 игра при резком отключении интернета

Рисунок 3.1 – Фрагмент чек-листа

**Тест-кейс** (тестовый случай) – это набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства. Под тест-кейсом также может пониматься соответствующий документ, представляющий формальную запись тест-кейса. [4] Для проверки одного требования может понадобиться несколько тест-кейсов. Для реализации полного тестового покрытия требований к приложению, должен быть создан хотя бы один тест-кейс для каждого требования. Некоторые методологии (например, RUP) рекомендуют создавать для каждого требования минимум, два тест-кейса: 1) для позитивного и 2) негативного тестирования. Пример тест-кейса приведён на рисунке 3.2. [4].

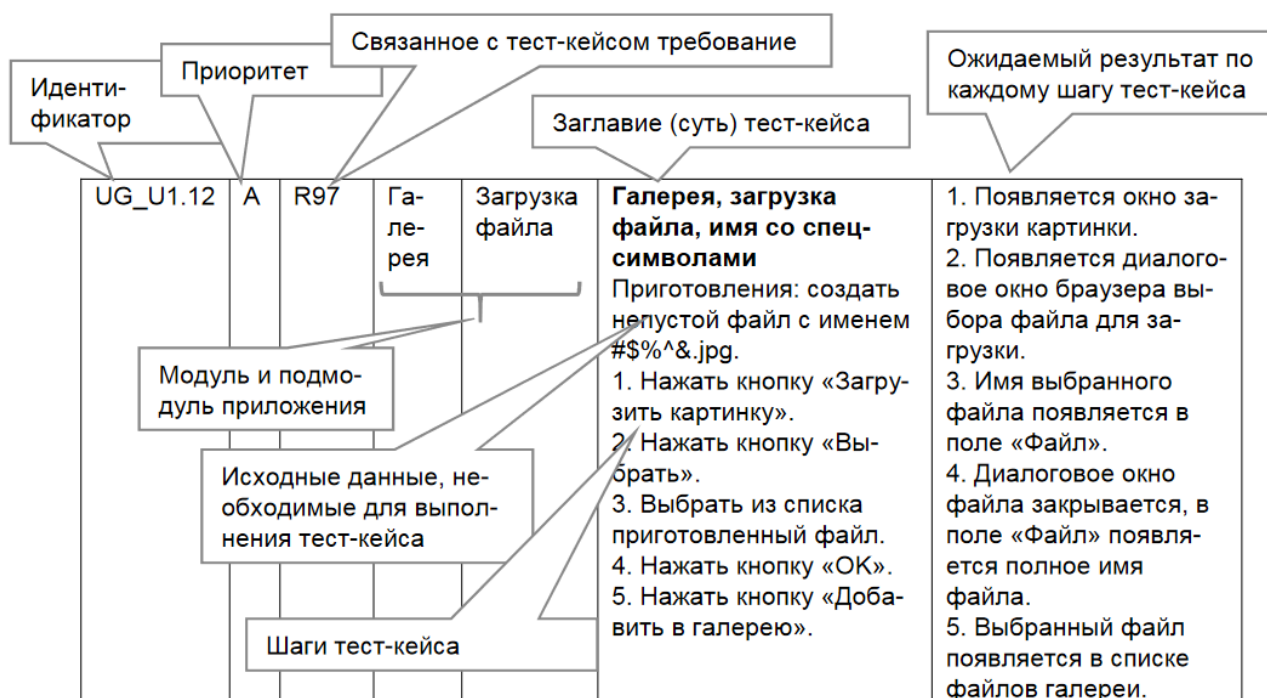


Рисунок 3.2 – Пример тест-кейса

В интеграционном и системном тестировании часто встречаются высокоуровневые тест-кейсы, в которых достаточно подробно (в чек-листе только идея и кратко) описаны общие идеи и операции теста (без конкретных входных данных и ожидаемых результатов).

Наличие тест-кейсов позволяет:

- структурировать и систематизировать подход к тестированию;
- контролировать тестовое покрытие, принимать меры по его увеличению;
- контролировать выполнение планов;
- улучшить взаимопонимание между заказчиком, разработчиками и тестировщиками (тест-кейсы зачастую нагляднее показывают поведение приложения, чем это отражено в требованиях);
- проводить регрессионное и повторное тестирование;
- повышать качество требований (создание чек-листов и тест-кейсов – один из способов тестирования требований);
- быстро ввести в курс дела подключившегося к проекту нового сотрудника;
- сохранить тестовые данные для последующего использования, обучения тестировщиков и обмена опытом между сотрудниками и командами.

Таким образом, если у вас низкоквалифицированные сотрудники (или этого требует руководство) необходимо создавать и поддерживать подробные тест-кейсы, а если в команде квалифицированные сотрудники, то можно ограничиться чек-листами.

#### **4. Проведение тестирования и составление отчетов о ходе тестирования.**

Тестирование может проводиться ручным и машинным способом. Существует достаточно большое число видов и методов тестирования. В отчетах фиксируются выполненные тесты и полученные результаты. Выявленные ошибки подлежат классификации и занесению их в соответствующую базу данных. Необходимо оценивать эффективность тестирования и покрытие функционала и кода тестами. Опрос разработчиков позволит определить качество выставления ошибок. Об эффективности и качестве тестирования говорит также уровень удовлетворённости пользователей. Его можно выяснить посредством анкет и обратной связи. Пропущенные ошибки и причины их пропуска надо анализировать.

**5. Анализ и обсуждение** способов и результатов тестирования, обсуждение дефектов

**Обеспечение качества** ПП и, в частности, выявление ошибок – **задача общая** для всей команды. В таблице 3.2 приведены результаты тестирования реального ПП одной из вологодских компаний. Количество выявленных в течение некоторого периода времени ошибок разбито по критичности и по отделам компании, которыми они были обнаружены.

Таблица 3.2 – Результаты тестирования в разрезе отделов

Критичность ошибки	Количество ошибок, выявленных отделом			Итого
	Отдел тестирования	Отдел проектирования	Отдел внедрения	
Hight	34	13	35	82
Medium	11	16	7	34
Low	2	4	0	6
Итого	47	33	42	122

Обеспечение качества выпускаемых программных продуктов путем проведения различных видов тестирования и фиксирования информации об

ошибках с целью последующего их исправления в данной компании возложено на отдел *тестирования*. Отдел *проектирования* занимается разработкой новых и поддержкой созданных ПП, в рамках заявленной функциональности, а также развитием программных продуктов и своевременным внесением в них изменений. Задача отдела *внедрения* — обеспечение правильной работы разработанных в компании ПП в рамках заявленной функциональности в разнообразных условиях эксплуатации. Из таблицы видно, что отделом тестирования выявлено только около 40% ошибок и не малый вклад внесли остальные отделы.

Цели, методы, результаты тестирования должны обсуждаться с командой. В каждом проекте, компании, команде цели свои собственные. Мнение руководящих менеджеров, разработчиков и тестировщиков может различаться. Тестировщику надо подстраиваться под текущие проектные цели.

6. Участие в проведении **опытных эксплуатаций** ПП.

7. Составление **отчетов** об имеющихся проблемах.

8. **Контроль** процесса **ликвидации** выявленных ошибок разработчиком;

9. Общение с разработчиками, **консультирование** клиентов.

10 Соблюдение **стандартов** тестирования. Примеры стандартов:

- ISO / IEC 25000: 2014 Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Руководство (взамен ISO-9126 и ISO-14598);
- ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов;
- ГОСТ Р ИСО/МЭК 12119-2000 Информационная технология (ИТ). Пакеты программ. Требования к качеству и тестирование;
- ANSI/IEEE 829–1983. Стандарт на документацию тестирования компьютерных программ;
- ANSI/IEEE 1008–1986. Тестирование программных модулей и компонент ПС.

Также должность тестировщика может предполагать составление технической документации на русском и иностранном языках и некоторые другие работы

### 3.3. Виды тестировщиков

В зависимости **от участия**, которое специалист принимает в процессе тестирования, различают следующие роли:

- **тест-аналитик** выполняет анализ продукта, разбивает его на составные части, расставляет приоритеты тестирования и составляет логическую карту приложения;
- **тест-дизайнер** на основании информации, полученной от аналитика, разрабатывает стратегии тестирования и тесты;
- **Тестировщик-исполнитель** проводит тестирование по уже готовым тест-кейсам, составляет описание выявленных ошибок.

В некоторых организациях в обязанности тестировщика входит все вышеперечисленное.

По **административной иерархии** (квалификации) выделяют должности: **младший** тестировщик, **тестировщик**, **старший** тестировщик, **ведущий** тестировщик, **руководитель** группы тестирования. Должностные обязанности специалистов по тестированию различных квалификационных уровней приведены

в таблице 3.3. Этот список может меняться от компании к компании, но не очень значительно.

Таблица 3.3 – Должностные обязанности тестировщиков

Квалификационный уровень. <b>Должность.</b> Направление деятельности работников
<p><b>1. Тестировщик, младший тестировщик.</b> Проводит различные виды тестирования; тестирует ПП на разных аппаратных платформах; проводит тестирование различных материалов, передаваемых пользователю вместе с ПП; вносит предложения по совершенствованию тестируемого ПП; оформляет отчет по результатам тестирования; консультирует пользователей и сотрудников других отделов.</p>
<p><b>2. Старший тестировщик.</b> Участвует в разработке плана работ и отвечает за соблюдение сроков; классифицирует ошибку, указывает возможную причину возникновения; консультирует пользователей и сотрудников других отделов; принимает меры по устранению возможных рисков; несет ответственность за качество работы; сообщает о возникающих проблемах, которые могут привести к срыву сроков или ухудшению качества работ, а так же их досрочном выполнении; оформляет отчет по результатам тестирования; повышает свою квалификацию; принимает участие в обучении стажеров; оценивает уровень знаний и навыков сотрудников отдела в рамках своих компетенций.</p>
<p><b>3. Ведущий тестировщик.</b> Проводит различные виды тестирования; тестирует на разных аппаратных платформах и различном программном окружении; проводит тестирование различных материалов, передаваемых пользователю вместе с программным продуктом; принимает участие в исследованиях по выработке рекомендаций для пользователей по особенностям настройки различного программного обеспечения и конфигурации оборудования; вносит предложений по совершенствованию тестируемого программного продукта; оформляет отчет по результатам тестирования; консультирует пользователей и сотрудников других отделов; разрабатывает новые и поддерживает в актуальном состоянии существующие методики и контрольные примеры тестирования; предпринимает меры по устранению возможных рисков.</p>
<p><b>4. Руководитель группы тестирования, отдела тестирования.</b> Разрабатывает план работы группы; отвечает за качество и сроки работ сотрудников своей группы; выполняет различные виды тестирования программных продуктов и находит ошибки, выявление которых требует подготовительной работы, основанной на знаниях программного продукта, типовых действиях пользователя; выполняет экспертную оценку критичности обнаруженных ошибок для передачи конечному пользователю; подготавливает методическое обеспечения процесса тестирования; актуализирует описание методики тестирования в случае тестирования измененной функциональности; участвует в разработке новых технологий тестирования; управляет персоналом, принимает решения и несет ответственность за распределение ресурсов и за работу персонала подразделения; несет ответственность за результат; осуществляет стратегическое управление направлением; консультирует пользователей и сотрудников других отделов; привлекается как эксперт или исполнитель для наиболее сложных работ; сообщает о возникающих проблемах, которые могут привести к срыву сроков или ухудшению качества работ, а так же их досрочном выполнении; подготавливает отчет и отзыв о работе сотрудников группы в установленный срок; повышает свою квалификацию, а так же способствует повышению квалификации своих подчиненных; принимает участие в обучении стажеров; оценивает уровень знаний и навыков сотрудников группы в рамках своих компетенций.</p>

Тестировщиков можно классифицировать на а) тест-инженеров и б) devops-инженеров, обслуживающих инфраструктуру.

**Тест-инженер** – разработчик тестов, акцентирующий силы и знания на методологии тестирования, способный использовать различные инструменты для решения поставленной задачи. **DevOps-инженер** занимается внедрением методологии DevOps (Development Operations). Культура, методика DevOps возникла в 2008 году и была призвана стать связующим звеном между командами разработки и эксплуатации, которые сваливали друг на друга вину за проявляющиеся в ПО проблемы. DevOps-инженер совмещает профессии админа, разработчика, тестировщика, менеджера и синхронизирует все этапы создания программного продукта (от написания кода до тестирования и выпуска приложения), автоматизирует выполнение их задач, внедряя различные программные инструменты.

Специалистов **по пользовательскому тестированию** разделяют на альфа- и бета-тестеров:

- **альфа-тестер** – пользователь (тестировщик, программист, бухгалтер и т.п.), взявший на себя в какой-либо форме обязательства по полному или частичному тестированию программы, находящейся на стадии разработки («Альфа-версии», как правило, не полнофункциональной);
- **бета-тестер** – подвергает пользовательскому тестированию «Бета-версии» программы - так называемых «релиз-кандидатов» готовых к выпуску или уже опубликованных официально версий.

В целом можно выделить еще несколько ролей, связанных с тестированием. Например, **Security-инженер** отвечает за прогон security-тестов (тестирование безопасности) и изучение уязвимостей в используемых компонентах, **инженер-инструменталист** (tool-engineer) снабжает процесс необходимыми средствами (среда разработки и тестирования, средства нагрузочного тестирования, система отслеживания ошибок и др.), **инженер по автоматизации** в числе прочего автоматизирует прогон тестов. Обеспечение качества (Quality Assurance) ПО на всех этапах его разработки, выпуска и эксплуатации – задача **QA-тестера** (менеджера, инженера). В идеале ответственность за качество ПП на плечи тестировщиков возлагаться не должна: управление качеством (QA) и тестирование две непересекающиеся сущности (рис.3.3)

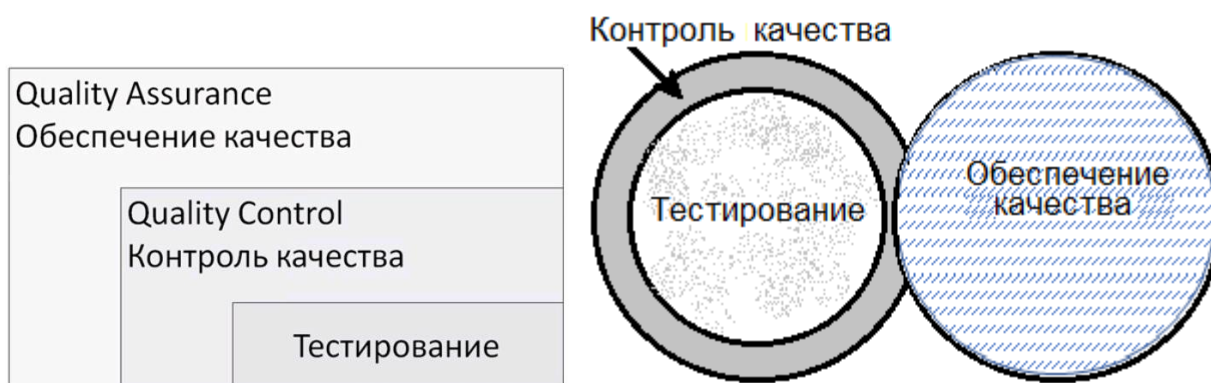


Рисунок 3.3 – Тестирование и обеспечение качества

### 3.4. Требования к тестировщикам

Требования к тестировщикам могут отличаться от компании к компании и зависят от функциональных обязанностей и от видов тестирования, которые необходимо выполнять (ручное функциональное, автоматизированное,



тестирование безопасности, тестирование удобства использования и т.д.), а также от отраслевой специализации.

В большинстве компаний к соискателю на должность тестировщика предъявляют требование наличия **высшего образования** (предпочтительно технического). Однако на низшие должности тестировщиков-исполнителей («обезьян») могут принимать людей, которые в лучшем случае, обладают навыками пользователя компьютером и каким-то продуктом. При этом требования к уровню необходимых навыков и специализации варьируются в зависимости от тестируемого программного обеспечения.

Тестировщик – это **инженерная** профессия, а значит, он должен разбираться в инженерных процессах разработки, тестирования, схемотехники и т.д. Функции инженерной деятельности меняются в зависимости от сферы приложения инженерного труда (наука, проектно-технические разработки, производство), от специализации, места в должностной структуре, этапа проводимой разработки, должности. Успешность в профессии в значительной мере определяется уровнем подготовки (технической квалификации).

**Уровень подготовки** обычно показывают перечислением знаний, умений и навыков, которыми обладает претендент. Основными требованиями к соискателю, как правило, являются:

**а) знание** и понимание

- жизненного цикла разработки программного продукта и места тестирования в данном процессе;
- основных стандартов, методологий и технологий построения программных продуктов, технологий построения баз данных;
- операционных платформ (в т.ч. мобильных, например iOS, Android);
- (на базовом уровне) языков программирования, языка запросов SQL, скриптовых языков и основ языка разработки текущих проектов;
- типов создаваемых приложений (Веб-приложения, банковское ПО и т.п.) методологий и технологий в них применяемых;
- типов, технологий и методик тестирования (CMM, RUP, MSF);
- стандартов в области тестирования;
- автоматизированных средств разработки и тестирования;
- английского языка на уровне, достаточном для чтения и написания технических текстов.

**б) умение**

- работать с кодом и не бояться заглянуть в «чёрный ящик», чтобы помочь разобраться с проблемой разработчику;
- с одной стороны, играть роль пользователя ПП (воспроизводить его поведение), а с другой – роль инженера (анализировать, поведение системы, входящие параметры и полученные результаты);
- взаимодействовать с клиентами: трансформировать просьбы и пожелания внешних клиентов в технические требования;
- взаимодействовать с членами команды. Это способствует обмену информацией по проекту и объективному видению ситуации;
- ясно и доступно излагать свои мысли, создавать системную и тестовую документацию;
- быстро переключаться между задачами.

**в) навыки**

- опытного пользователя ПК,
- программирования на базовом уровне, работы с базами данных, администрирования прикладных программ и популярных операционных систем;

- организации и проведения различных видов тестирования программного обеспечения.
- проектирования комплектов теста
- использования инструмента автоматизированного тестирования и написания автотестов (Silk Test, Rational Robot, Rational ClearQuest, TrackStudio и т.д.) .

Помимо технических знаний, умений и навыков для успешного решения задач тестирования и обеспечения качества ПП тестировщик должен обладать рядом **индивидуальных характеристик**. Необходимыми личностными качествами тестировщика являются:

- *логическое мышление* – способность выделять все факторы, влияющие на состояние исследуемого объекта; устанавливать причинно-следственные связи; выделять ключевые показатели, определяющие состояние проблемы или влияющие на результат;
- *аналитические способности* – способность анализировать информацию, поведение системы, полученные результаты;
- *внимательность* – умение замечать неточности и противоречия;
- *обучаемость* – наличие внутренней мотивации на приобретение новых знаний, навыков, опыта; освоение и применение в своей деятельности новой информации, технологий, моделей поведения;
- *творческий подход* – умение создавать необычные и оригинальные идеи, обобщения, теории.
- *ответственность* – выполнение принятых на себя обязательств;
- *коммуникабельность* – умение устанавливать контакт, слушать и слышать, задавать вопросы;
- *доброжелательность* – проявление вежливости и уважения, учет мнения других, достижение консенсуса в любой ситуации конфликта;
- *лидерские качества* – умение создавать и поддерживать комфортную атмосферу в коллективе, дипломатические навыки в общении с сотрудниками, способность фокусироваться на всех системных процессах и расставлять приоритеты в плане работ и т.п.
- а также терпение, усидчивость, педантичность (предельная точность и аккуратность в каких-либо действиях человека; чрезмерная склонность к соблюдению формальных требований), хорошая память, умение адаптироваться к существующим задачам, умение принимать решения в критических ситуациях, умение отстаивать свою точку зрения на любом уровне и некоторые другие.

### 3.5. Мотивация к профессии тестировщика

Как и у любой другой профессии у профессии тестировщик есть свои **минусы**:

- Монотонность и однообразие трудового процесса (но есть мнения, что однообразие и монотонность – видимость);
- Наличие риска остаться узкопрофильным специалистом;
- Негативное отношение со стороны разработчиков: «Тестировщики – кликеры и багозаводильцы».

Однако, **плюсов** больше. В их числе:

1. **Востребованность** на рынке специалистов. Это перспективная и молодая профессия. Квалифицированных тестировщиков по-прежнему не хватает
2. В соответствии с V-моделью **тестировщик – центр** процесса разработки ПО. Участвуя во всех этапах проекта, тестировщики получают хорошее представление о процессах и целях продукта.

3. Получение **знаний** и практических **навыков в области разработки ПО** (следствие 2), возможность расширять свой кругозор. При наличии хорошей квалификации тестировщика границы между ним и разработчиками размыты.

4. Возможность влиять на качество продукта и получить **доступ к управлению продуктом**: в формировании хорошего продукта должны участвовать прямо или косвенно все заинтересованные стороны, но тестировщики единственные из инженеров имеют право вето на выпуск версии продукта

5. **Исследовательский характер работы**, сочетающий **творчество** (искать баги, ломать то, что выглядит рабочим, находить ошибки там, где другой даже и не подозревал) и технологии («настоящий тестировщик — это летчик-испытатель»). Инженер-тестировщик – это (в идеале) человек, который разрешает технические проблемы, связанные с разработкой ПО, это исследователь из мира разработки ПО (символ инженера-тестировщика – лупа, с её помощью можно выявлять дефекты, «прожигать дырочки», добывать огонь, наблюдать за звёздами). В рамках простоя есть возможность заняться автоматизацией своей деятельности (например, процесса подготовки документации) или улучшением продуктов, им используемых.

6. Любовь и **уважение**: хороших тестировщиков любят и ценят все.

7. Возможность **удаленной работы** (страна и расстояние часто не имеют значения) и получение опыта и знаний в удалённой разработке ПО (с целью дальнейшей работы программистом)

8. Отличные возможности для **фриланса** – работа с заказчиками из разных регионов и стран, в IT-компаниях, в составе независимых групп.

Также у тестировщиков хорошие **перспективы роста и дальнейшего развития**. Профессиональное и карьерное развитие специалиста в области тестирования может выглядеть следующим образом:

— **Профессиональный путь развития**: младший тестировщик → старший тестировщик → ведущий тестировщик → эксперт.

— **Административный путь развития**: ведущий тестировщик → руководитель.

Возможна **гибридная** модель развития: профессиональный рост, плавно перетекающий в административный.

Однако профессиональное совершенствование специалиста по тестированию ПП может осуществляться и в **смежных областях**. Так как тестирование находится на стыке различных прикладных ИТ-дисциплин, то с минимальными трудозатратами высококлассный тестировщик сможет попробовать свои силы в таких видах деятельности (это далеко не полный перечень направлений его развития) как:

- консалтинг в сфере тестирования (постановка/аудит процессов тестирования, обучение специалистов вопросам, связанным с тестированием, и т.п.);
- управление качеством (внедрение/аудит систем менеджмента качества);
- бизнес-анализ (выявление, описание существующих бизнес-процессов, проектирование новых, разработка требований);
- ИТ-консалтинг (постановка и автоматизация бизнес-процессов);
- разработка ПО (программирование). На современном этапе развития технологий приветствуется кроссфункциональность команды, когда тестировщик и программист во многом взаимозаменяемы;
- управление проектами по созданию или внедрению информационных систем.

Таким образом, профессия тестировщика является интересной и перспективной.