



**Частное учреждение профессионального образования
«Высшая школа предпринимательства (колледж)»
(ЧУПО «ВШП»)**

Кафедра информационных технологий

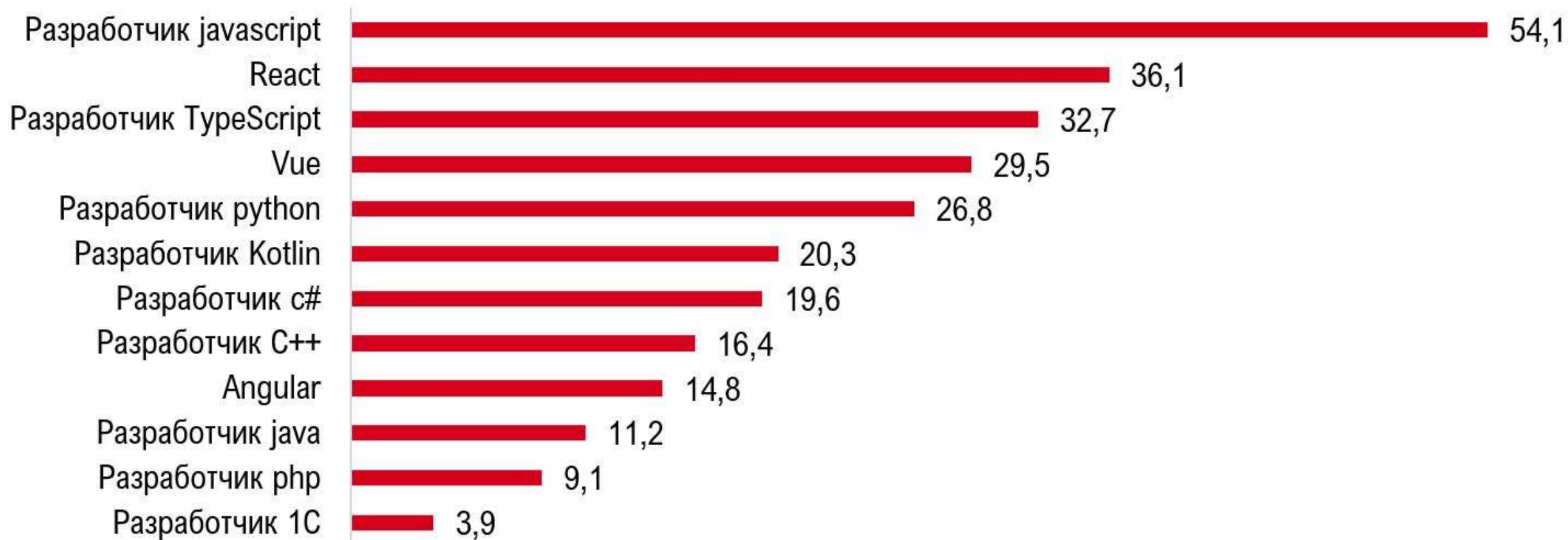
**Внедрение и поддержка
компьютерных систем**

Уровень предлагаемых зарплат в области разработки ПО Россия, 2023 год



Регионы	Без опыта	От 1 года до 3 лет	От 3 до 6 лет	Более 6 лет
Россия	50 000 ₹	71 500 ₹	93 580 ₹	100 500 ₹
Москва	67 360 ₹	81 590 ₹	99 700 ₹	121 680 ₹
Петербург	59 000 ₹	72 700 ₹	89 300 ₹	100 000 ₹
Северо-Западный ФО	54 860 ₹	71 000 ₹	87 500 ₹	99 000 ₹
Сибирский ФО	46 100 ₹	72 800 ₹	103 000 ₹	110 000 ₹
Центральный ФО	52 700 ₹	71 570 ₹	91 770 ₹	114 370 ₹
Уральский ФО	50 630 ₹	74 820 ₹	97 110 ₹	100 000 ₹
Южный ФО	46 950 ₹	60 250 ₹	75 650 ₹	80 000 ₹
Приволжский ФО	47 840 ₹	66 760 ₹	83 630 ₹	89 100 ₹
Северо-Кавказский ФО	47 550 ₹	50 050 ₹	69 590 ₹	85 000 ₹
Дальневосточный ФО	59 970 ₹	103 700 ₹	120 540 ₹	130 000 ₹

Уровень конкуренции (соотношение числа резюме и вакансий) по языкам программирования/стекам, Россия, 2023 год





**Что происходит,
когда мы открываем сайт
в браузере?**

A close-up shot from the movie Inception showing Leonardo DiCaprio and Matt Damon. DiCaprio is on the left, looking slightly to the right with a serious expression. Damon is on the right, seen in profile, looking towards DiCaprio. The background is blurred, showing what appears to be an office or meeting room setting.

МЫ ДОЛЖНЫ

ПОЙТИ ГЛУБЖЕ

Цель верификации и валидации в тестировании программного обеспечения

Верификация и валидация при тестировании жизненно важны для обеспечения того, чтобы разработчики использовали передовые методы создания программного обеспечения.

Цель состоит в том, чтобы избежать сбоев в работе приложений на критическом этапе и гарантировать, что они продолжают работать на благо пользователя.

Давайте сравним верификацию и валидацию и то, как они влияют на конечный продукт.

Верификация и валидация

Что такое верификация?

При верификации команда разработчиков изучает документы для создания программного обеспечения или приложения. Цель состоит в том, чтобы убедиться, что разработчик, которому поручен проект, соблюдает все изложенные требования. Логика кода должна соответствовать проектной документации независимо от языка программирования.

Что такое валидация?

Валидация включает в себя проверку приложения на разных этапах разработки, чтобы убедиться, что оно соответствует требованиям. Если в документе требуется веб-страница с функцией живого чата, то разработчик должен создать именно ее. Если что-то отсутствует или не соответствует запросу клиента, это следует выявить и исправить, чтобы получить ожидаемый продукт.

Верификация и валидация

Верификация всегда отвечает на вопрос: **"делаем ли мы продукт правильно?"**

Валидация всегда отвечает на вопрос: **"делаем ли мы правильный продукт?"**



Процесс верификации

Проверка требований: команда проекта подтверждает правильность всех пунктов в требованиях на основе данных, полученных от бизнеса. Заказчики должны иметь возможность вносить любые изменения или удалять ранее запрошенные. Команда должна получить окончательное одобрение от всех соответствующих заинтересованных сторон, чтобы убедиться, что конечный результат отвечает их потребностям.

Проверка дизайна: команда тестировщиков должна проанализировать дизайн программного обеспечения, включая макеты и прототипы. Охватывают ли логические модели баз данных все соответствующие поля для обработки данных, передаваемых туда и обратно между приложениями? Доволен ли заказчик удобством предлагаемого интерфейса? Прежде чем приступить к работе, необходимо прийти к единому мнению о достоверности проектной документации.

Проверка логики кода: команда проверяет код, чтобы убедиться в его полноте и согласованности. Они также должны проверить исходный код и связанные с ним артефакты, например дизайн базы данных, и убедиться, что они соответствуют требованиям.

Процесс верификации

Планирование: команда изучает взаимосвязи между различными требованиями, затем определяет все задачи, которые должны быть выполнены в ходе верификации. Назначаются ответственные за другие задачи, включая демонстрацию того, насколько хорошо функционируют различные компоненты, проверку обеспечения качества или выявление и составление отчетов о дефектах.

Выполнение: команда выполняет поставленные перед ними задачи и документирует результаты верификации.

Отчет: все результаты, полученные в ходе верификации, собираются в отчеты. Команда проверяет, функционирует ли приложение так, как задумано.

Процесс валидации

Проверка дизайна: На этом этапе команда описывает все бизнес-требования конечного пользователя. Затем составляется план валидации для каждого пункта, прежде чем подтвердить готовность программного обеспечения к разработке. После этого команда может получить одобрение от высшего руководства перед началом тестирования.

Проверка установки: Команда пытается установить приложение в соответствии с планом валидации. Цель состоит в том, чтобы убедиться, что процесс установки и все необходимое системное оборудование соответствуют требованиям проекта. Кроме того, тестировщики подтверждают, что тестовая среда функционирует аналогично производственной среде.

Процесс валидации

Проверка функциональности: Тестировщики подвергают продукт различным сценариям тестирования, чтобы убедиться, что он соответствует заданным пользовательским требованиям.

Цель - проверить все доступные функциональные возможности, чтобы определить, работает ли программное обеспечение в соответствии с требованиями заказчика.

Команда полагается на план валидации, чтобы убедиться, что тестовая среда подходит для реализации сценариев, описанных в требованиях. Таким образом, они могут гарантировать, что конечный результат соответствует ожидаемому и способен удовлетворить потребности клиента.

Процесс валидации

Проверка производительности: Тестирование производительности показывает, что программное обеспечение может функционировать так, как это необходимо бизнесу в реальных условиях. Клиенты могут сами проводить бета-тестирование, чтобы получить представление о продукте и понять, был ли он разработан в соответствии с их требованиями.

Взгляд со стороны помогает выявить ошибки и дефекты, которые команда разработчиков могла пропустить.

Проверка готовности продукта: После завершения всех других этапов процесса валидации продукт считается готовым к релизу. Это означает, что команда разработчиков может двигаться вперед, выпуская программное обеспечение в производственную среду.

Любые дефекты, обнаруженные после выпуска продукции, устраняются с помощью обновлений программного обеспечения.

Зачем проводится верификация и валидация?

Уменьшить количество дефектов в реальном продукте

Если новая часть программного обеспечения при выпуске не работает так, как предполагалось, качество продукта может пострадать. Заказчики и команда будут чувствовать себя более уверенно в том, что приложение будет работать без ошибок, если они тщательно его протестировали, используя тестовые сценарии, описанные в процессе верификации и валидации.

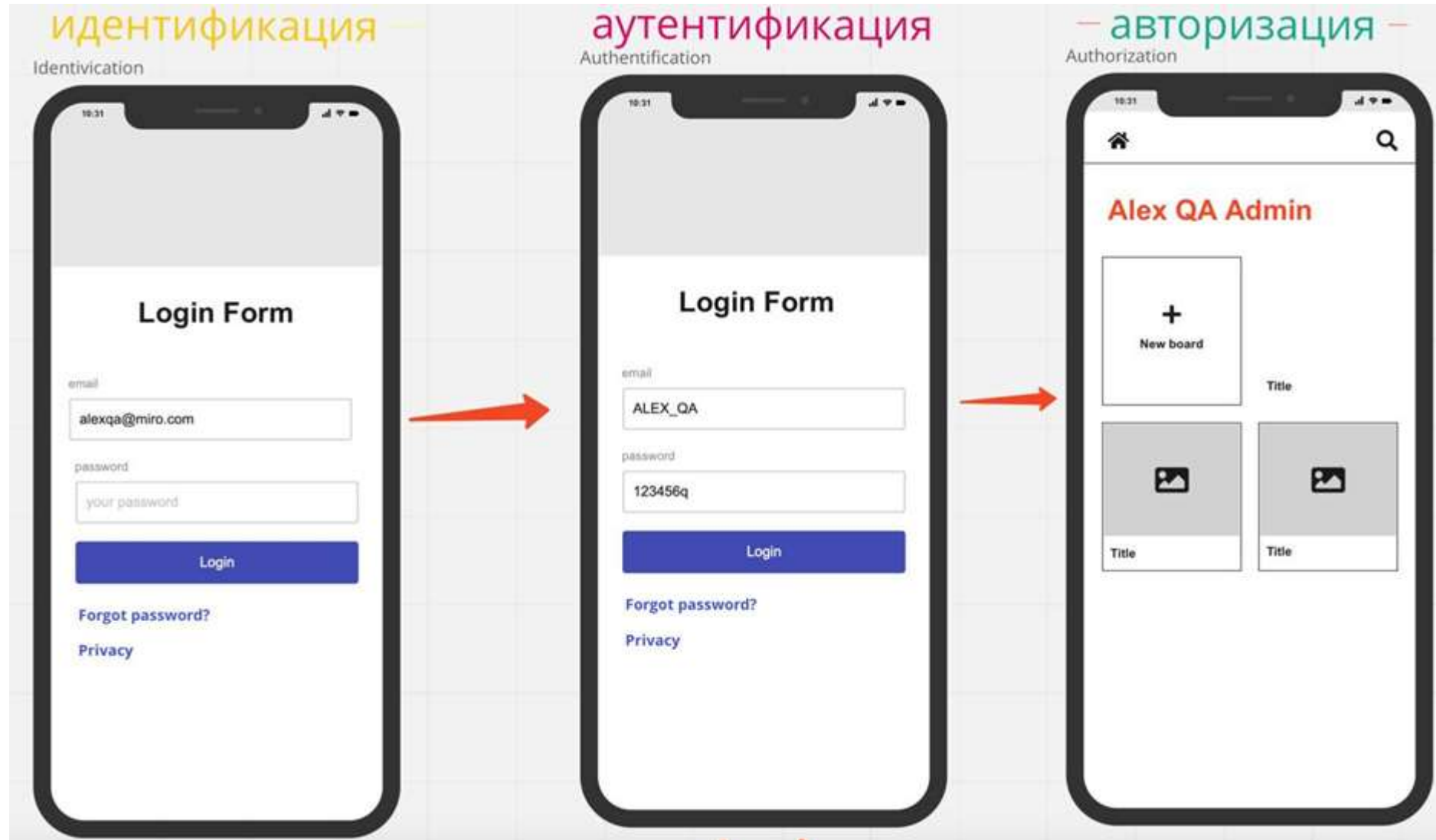
Убедиться, что дизайн соответствует техническим условиям

Последнее, чего хочет команда разработчиков — это вызвать недовольство клиента тем, что полученный продукт не соответствует его запросам. Регулярная проверка требований во время верификации и валидации помогает разработчикам не упустить критически важные функциональные и проектные требования, отмеченные в документации.

Идентификация, Аутентификация, Авторизация



Идентификация, Аутентификация, Авторизация. В чем же разница?



Идентификация

Цель **идентификации** – понять, кто “стучится в нашу систему”. Чаще всего для идентификации используются: емейл, имя пользователя, телефон...

Бывает возможность выбора. К примеру, залогиниться по имени пользователя или по номеру телефона. Важно то, что этот идентификатор будет являться уникальным значением. В Базе Данных это, вероятнее всего, будет столбец в таблице с уникальными данными (primary key). То есть в системе не может быть зарегистрировано два одинаковых идентификатора, два одинаковых номера телефона, два одинаковых имейла и так далее.

Например, при попытке зарегистрировать новый адрес электронной почты пользователь вводит свой идентификатор (логин), например: «Вася92», а система подсвечивает поле красным и сообщает, что такой пользователь уже зарегистрирован в системе, предлагая на выбор несколько других вариантов – именно это и будет пример идентификации.

Аутентификация

Процесс **аутентификации** - проверка пользователя на его подлинность.

Существует 3 фактора, которые задействуются в процессе аутентификации:

Знание

Например пароль, ПИН-код, секретное слово и т. д.

Владение

Второй фактор – это владение, является ли пользователь обладателем чего-то, к примеру – уникальных биометрических данных, присущих только ему.

Это очень хорошо распространено в телефонах, к примеру, когда девайс распознает владельца по отпечатку пальца.

Свойство

Пользователь имеет какой-то уникальный признак, и система его может аутентифицировать и пропустить дальше. К примеру: в случае использования мобильного банкинга или налогового приложения после его запуска система попросит у пользователя набор ключей. Пользователь

Авторизация

Авторизация это завершающий этап, когда:

- система проверила наш идентификатор;
- успешно прошел процесс аутентификации.

После чего следует наделение пользователя определенными правами. Возможно, система авторизовала нас как юзера с уже определенным набором возможностей и показывает нам информацию, который должен видеть обычный пользователь системы.

В данном случае мы зашли в систему как обычный пользователь, но ведь система могла авторизовать нас как администратора. Во втором случае система предоставила право, к примеру, на редактирование или удаление информации. С точки зрения тестирования у пользователя появляется просто огромное поле возможностей для проведения тестирования.

Это крайне важно с точки зрения безопасности – на сколько система правильно ведет себя на этапе идентификации, затем на этапе аутентификации и в итоге авторизации.

Тестирование совместимости

Тестирование на совместимость проверяет, работает ли программное обеспечение должным образом во всех браузерах, операционных системах, базах данных, устройствах, сетях и т. д.

Оно проверяет совместимость в нескольких сценариях и выявляет ошибки, которые могут сделать программное обеспечение непригодным для использования или некачественным.

Тестирование на совместимость можно отнести к нефункциональному тестированию программного обеспечения, поскольку оно рассматривает внешне видимые и идентифицируемые достоинства программы.

Что касается видов функционального тестирования, они оценивают качество программы изнутри, то есть как написан код, лучшие практики программирования, компиляции, безопасность и т. д.

Типы тестирования совместимости

Тестирование обратной совместимости

Как следует из названия, тестирование обратной совместимости направлено на обеспечение работы новых версий программного обеспечения со старым оборудованием/более старыми версиями программного обеспечения.

Тестирование обратной совместимости необходимо для обеспечения стабильной работы пользователей, использующих старые устройства, которые могут уступать в системных ресурсах.

Кроме того, если на этих устройствах используются какие-либо операционные системы, тестирование обратной совместимости гарантирует, что новая версия будет совместима с ними.

Тестирование прямой совместимости

Тестирование прямой совместимости направлено на установление совместимости программного обеспечения с существующими или будущими версиями устройства.

Пример тестирования совместимости

Представьте, что вам нужно запустить обновленную версию приложения VK для Android.

Приложение должно безупречно работать во всех браузерах, устройствах на базе Android, на настольных ПК, ноутбуках и т. д.

Тщательное тестирование, гарантирующее, что приложение работает на всех этих устройствах без сбоев и в соответствии с документацией, называется тестированием на совместимость.

Тестирование совместимости гарантирует, что приложение правильно загружается и корректно работает в разных браузерах и на всех устройствах на базе Android.

Различия в тестировании мобильных и веб-приложений

Веб-приложения – программные приложения, доступ к которым происходит через веб-браузер. В большинстве случаев для их хранения используются веб-сервера, а разработка происходит с использованием HTML, CSS, или JavaScript. Оптимизированные веб-приложения можно использовать на различных устройствах с активным подключением к Интернету, включая ПК, ноутбуки, мобильные телефоны и планшеты.

Мобильные приложения адаптированы и созданы специально для мобильных устройств. Наиболее распространены следующие типы мобильных приложений:

Мобильные веб-приложения. Это стандартные веб-приложения, адаптированные для работы на мобильных устройствах и доступные через мобильные браузеры. Они существенно отличаются от обычных мобильных приложений, хотя и могут их напоминать.

Нативные мобильные приложения, разработанные исключительно для определенных операционных систем. Эти мобильные приложения доступны для загрузки в соответствующих магазинах приложений (магазин Google Play, App Store, Microsoft Store и т. д.).

Различия в тестировании мобильных и веб-приложений

Самое важное различие при тестировании мобильных и веб-приложений заключается в том, что первое тестирует программные приложения для мобильных устройств, а второе тестирует веб-приложения на предмет функциональности, совместимости и удобства использования.

Мобильные приложения могут работать на более широком спектре устройств, такие как смартфоны, умные часы, планшеты, системы блокировки, фитнес-трекеры и планшеты. Это усложняет тестирование мобильных приложений по сравнению с веб-приложениями на мобильных устройствах.

Веб-приложения предназначены для стационарных ноутбуков и настольных компьютеров с классическими функциями wifi-маршрутизатора и курсора мыши, которые отсутствуют в случае мобильных приложений.

Привычки пользователей сильно изменились в последнее десятилетие. Если раньше устройства часто включались и выключались, а пользователи входили и выходили из онлайн приложений, то теперь они могут длительное время оставаться в приложениях на своих мобильных устройствах.

Различия в тестировании мобильных и веб-приложений

Мобильные приложения имеют более широкую пользовательскую базу, чем веб-приложения, поэтому тестирование обоих приложений выполняется по-разному с учетом таких факторов, как постоянство доступа к сети, управление уведомлениями, синхронизация приложений между устройствами и так далее.

Веб-приложения более ориентированы на бизнес, тогда как мобильные приложения ориентированы на клиента. В результате тестирование мобильных приложений фокусируется на взаимодействии с клиентами и его опыте работы с приложением.

Для облачной автоматизации тестирования мобильных и веб-приложений вам потребуется лаборатория мобильных устройств в облаке, а для классических веб-приложений — лаборатория портативных устройств в облаке.



Технические проблемы

Разработка веб-приложений и мобильных приложений отличается с точки зрения использования и работы с разными типами устройств. Вот несколько областей, по которым различают веб- и мобильное тестирование:

- Тестирование требований к оперативной памяти и размера
- Тестирование использование батареи устройства
- Тестирование подключения к Интернету
- Тестирование средств взаимодействия с пользователем
- Тестирование размера экрана
- Тестирование совместимости
- Типы приложений

BCE!