



Интенсив QA

Вебинар №1:
Основы тестирования ПО



Преподаватель

Станислав Саргин

- 5 лет опыта в Тестировании ПО
- 3 последних года занимаю позицию QA Lead / QA Automation Engineer
- Ранее работал в Yandex
- Интересуюсь всеми аспектами тестирования и веб-разработки



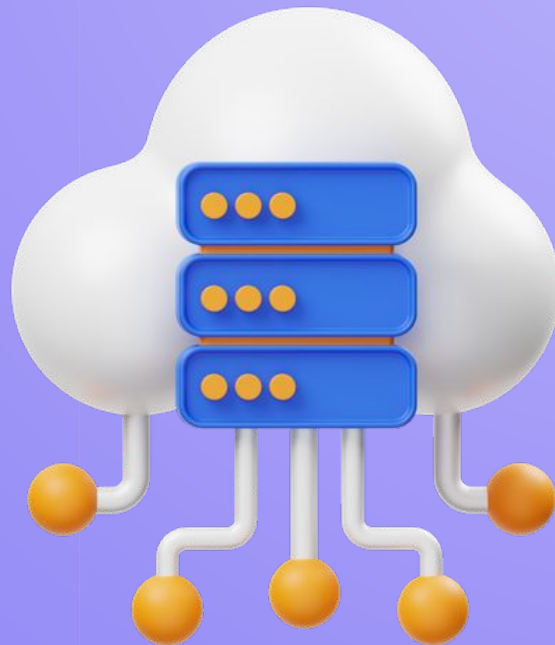
О компании

YLab Development

- Аутсорс-продакшн компания
- Реализация IT-проектов
- Автоматизация бизнес-процессов

Официальные сайты:

- <https://ylab.io>
- <https://university.ylab.io/>
- https://t.me/ylab_v_it
- <https://www.youtube.com/@ylabuniversity>



О курсе

Цель курса

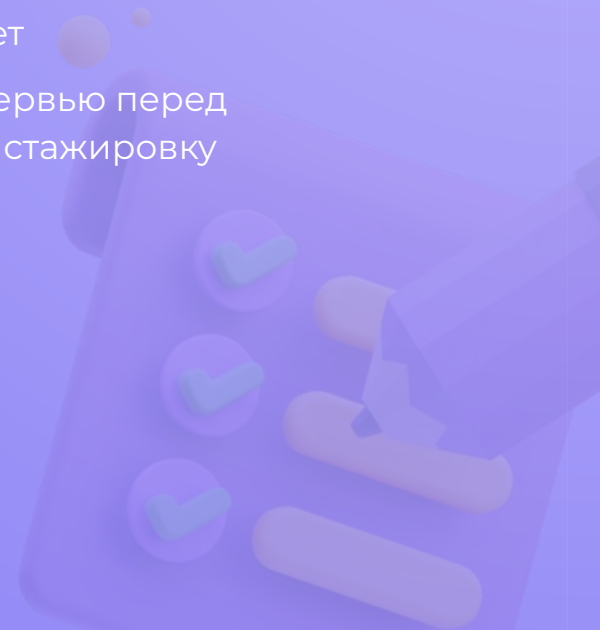
- Получение опыта тестирования ПО
- Формирование портфолио
- Знакомство с теорией тестирования и основными web-технологиями.

Объем

- Лекции – 3
- Домашних заданий – 3

Оценка

- Накопительная по итогу выполнения ДЗ
- Экзамена нет
- Личное интервью перед набором на стажировку



Требования к слушателям

Тестирование

- Знакомство с базовой теорией
Тестирования ПО

Web

- Общие знания о web-технологиях
- Понимание принципов клиент-серверного взаимодействия



Темы

- **Уровни тестирования ПО:** модульное, интеграционное, системное, приемочное
- **Виды тестирования:** функциональное, нефункциональное, регрессионное, дымовое, безопасности, производительности и др.
- **Типы тестирования:** ручное, автоматизированное, ручное с автоматизацией



Часть 1. Уровни тестирования

Модульное тестирование — тестирование отдельных компонентов программного обеспечения.

Является первым уровнем тестирования и проводится на уровне отдельных модулей приложения.

Оно позволяет выявить ошибки, связанные с неправильным функционированием отдельных модулей, а также упрощает отладку кода при их обнаружении.



Часть 1. Уровни тестирования



Интеграционное тестирование — тестирование, выполняемое для обнаружения дефектов в интерфейсах и во взаимодействии между интегрированными компонентами или системами.

Проводится после модульного тестирования и нацелено на проверку взаимодействия между различными модулями приложения.

Оно позволяет выявить ошибки, связанные с неправильным взаимодействием между модулями, а также убедиться в корректности работы приложения в целом.

Часть 1. Уровни тестирования

Системное тестирование — тестирование всей системы в целом, оно выполняется после интеграционного тестирования, чтобы проверить, работает ли вся система целиком должным образом.

В основном это тестирование типа «черный ящик», которое оценивает работу системы с точки зрения пользователя с помощью документа спецификации и оно не требует каких-либо внутренних знаний о системе, таких как дизайн или структура кода.



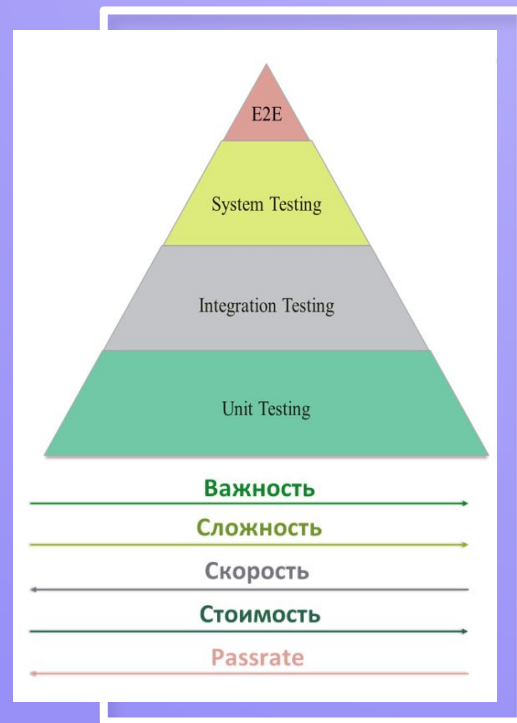
Часть 1. Уровни тестирования

Приемочное тестирование — формальное тестирование по отношению к потребностям, требованиям и бизнес процессам пользователя, проводимое с целью определения соответствия системы критериям приемки и дать возможность пользователям, заказчикам определить, принимать систему или нет.



Уровни тестирования:

- **Модульное (Unit testing)** - тестируется минимально-атомарный модуль программы, чаще всего это одна функция или метод;
- **Интеграционное (Integration testing)** - несколько модулей программы тестируются вместе;
- **Системное (System testing)** - вся программа тестируется полностью;
- **Приемочное (Acceptance testing)** - программа принимается заказчиком на соответствие заявленным требованиям либо тестировщики проходят end-to-end сценарии с точки зрения пользователя;



Часть 2. Виды тестирования



Функциональное тестирование — тестирование, основанное на анализе спецификации функциональности компонента или системы.

Функциональное тестирование выполняется чтобы убедиться, что каждая функция программного приложения ведет себя так, как указано в документе с требованиями.

Оно включает в себя тестирование основных функций, интерфейса, взаимодействия с другими системами, валидации входных данных и вывода результатов.

Часть 2. Виды тестирования

Для функционального тестирования принято использовать две техники:

- **Тестирование на основе требований:** содержит все функциональные спецификации, которые составляют основу для всех тестов, которые будут проводиться;
- **Тестирование на основе бизнес-сценариев:** содержит информацию о том, как система будет восприниматься с точки зрения бизнес-процесса;



Часть 2. Виды тестирования

Нефункциональное тестирование — проводится для проверки нефункциональных требований приложения, таких как производительность, безопасность, совместимость, надежность, удобство использования и т. д.

Оно проверяет, соответствует ли поведение системы требованиям по всем аспектам, не охваченные функциональным тестированием.

Когда ПО выходит на рынок и используется реальными конечными пользователями, у них есть шансы столкнуться с проблемами. Эти проблемы не связаны с функциональностью системы, но могут негативно повлиять на пользовательский опыт.



Часть 2. Виды тестирования

Регрессионное тестирование — тестирование уже протестированной программы, проводящееся после модификации для уверенности в том, что процесс модификации не внес или не активизировал ошибки в областях, не подвергавшихся изменению. Проводится после изменений в коде программного продукта или его окружении.

Это собирательное название для всех видов тестирования программного обеспечения связанных с изменениями, направленных на обнаружение ошибок в уже протестированных участках исходного кода, на проверку того, что новая функциональность не зааффектила старую.

Такие ошибки - когда после внесения изменений в программу перестаёт работать то, что должно было продолжать работать, - называют регрессионными ошибками.

Часть 2. Виды тестирования

Подтверждающее тестирование (Повторное тестирование)

— тестирование, при котором выполняются тестовые сценарии, которые были не пройдены при последнем запуске, с целью подтвердить успешность исправлений.

Данный тип тестирования, выполняется в новой сборке по проваленному на старой сборке тест-кейсу с тем же окружением и данными, для проверки того, что этот дефект теперь устранен.

Приоритет ре-теста выше регрессионных проверок, поэтому оно должно выполняться перед ними.



Часть 2. Виды тестирования

- **Дымовое тестирование** — этот вид тестирования направлен на проверку базовой функциональности продукта. Это позволяет выявить критические ошибки в начале процесса тестирования и сэкономить время и ресурсы на более детальном тестировании.
- **Тестирование безопасности** — этот вид тестирования направлен на проверку безопасности продукта и защищенности от внешних угроз.
- **Нагрузочное тестирование** — вид тестирования производительности, проводимый с целью оценить поведение компонента или системы под увеличивающейся нагрузкой для определения максимально допустимого уровня нагрузки для исследуемого компонента или системы.

Часть 2. Виды тестирования

- **Usability тестирование** — этот вид тестирования направлен на проверку удобства использования продукта для конечных пользователей. Это включает в себя проверку интерфейса, навигации и других аспектов, которые влияют на опыт пользователя.
- **Тестирование совместимости** — этот вид тестирования направлен на проверку совместимости продукта с различными операционными системами, браузерами, устройствами и другими программными и аппаратными средствами.
- **Тестирование на доступность** - этот вид тестирования направлен на проверку того, что продукт доступен для людей с ограниченными возможностями.



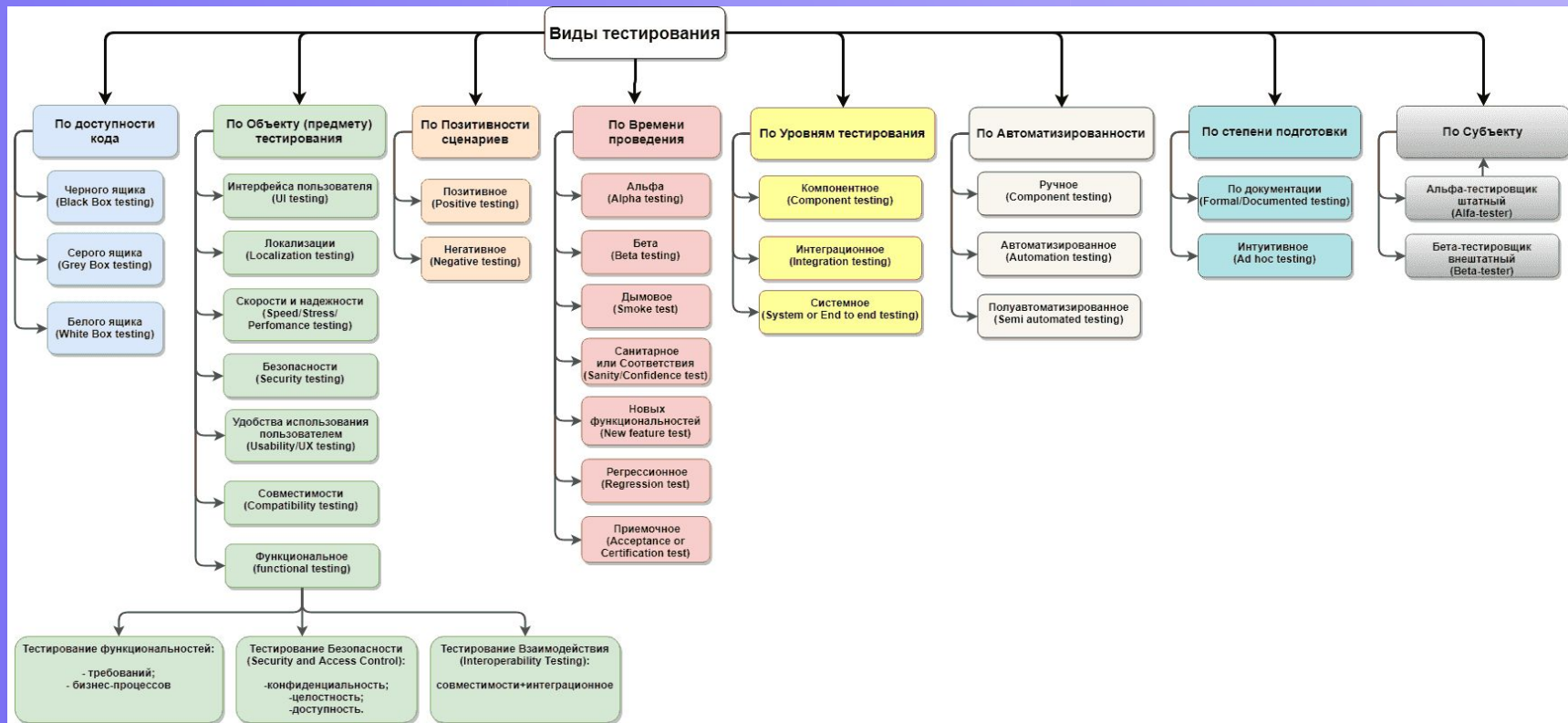
Часть 3. Типы тестирования

Ручное тестирование — это процесс проверки программного обеспечения на соответствие заданным требованиям и наличие ошибок путем ручного взаимодействия с приложением.

Ручное тестирование выполняется людьми, которые следят за поведением программного обеспечения и проверяют его функциональность, надежность, производительность, безопасность и другие характеристики.

Автоматизированное тестирование — это процесс проверки программного обеспечения на соответствие заданным требованиям и наличие ошибок с использованием специальных инструментов и скриптов.

Автоматическое тестирование выполняется без участия человека и позволяет ускорить процесс тестирования, улучшить его качество и снизить затраты на тестирование.





Спасибо за внимание!

Увидимся в следующих лекциях.

Наш университет: <https://university.ylab.io/>