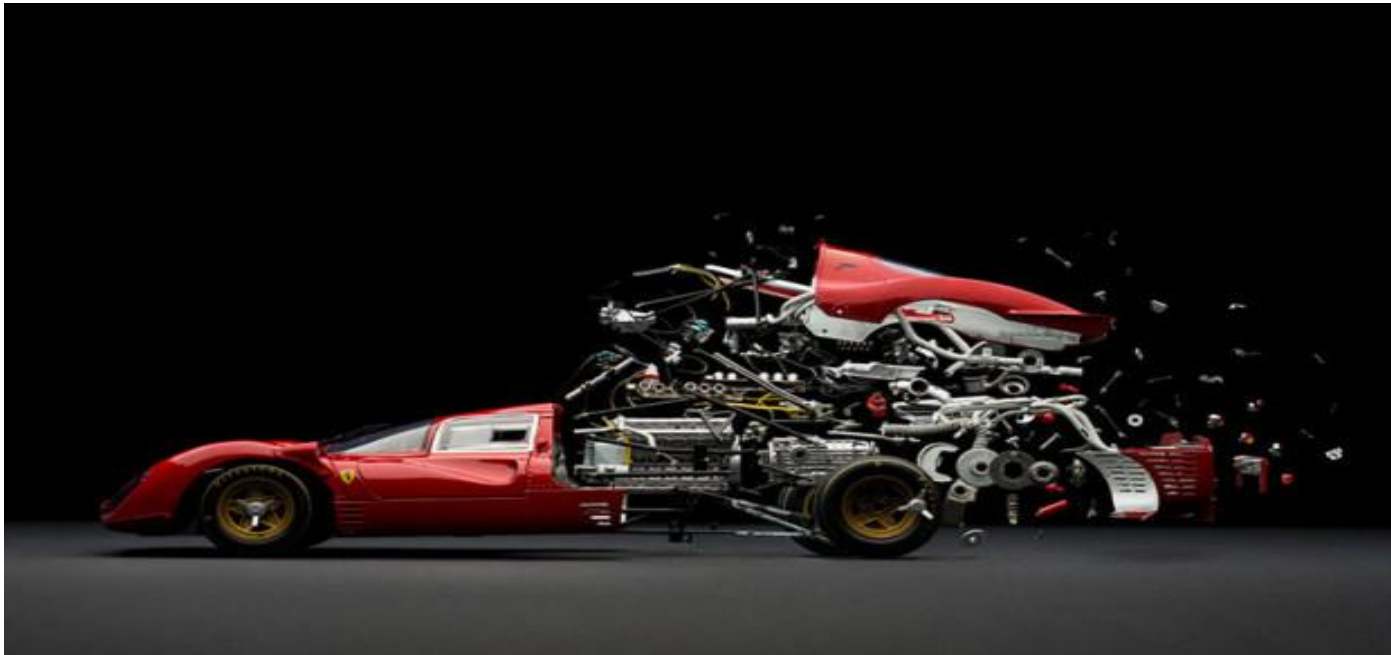


## ЛЕКЦИЯ 5

# Тестирование ПО: каким оно бывает?



# Какие типы тестов вы знаете?





Тестирование инсталляции и лицензирование  
кроссплатформенное  
Тестирование прототипа  
Валидация  
Тестирование требований  
Черный ящик  
Бета-тестирование  
Белый ящик  
Юзабилити  
По документации  
Доступности  
Интеграционное  
стрессовое  
Тестирование пользовательского интерфейса  
Автоматизированное  
Исследовательское  
Серый ящик  
Защищенность  
Альфа-тестирование  
кроссбраузерное  
Тестирование новых функций  
Локализация  
Приемочное тестирование  
объемно  
Системное  
Регрессионное  
АТ  
Модульное  
Smoke  
Функциональное  
Интуитивное  
Безопасность  
Smoke  
нагрузочное  
е

## Типы тестов:

По объекту

По покрытию

По знанию кода

По степени автоматизации

По подготовленности

По месту и времени проведения

По изолированности компонентов

## Типы тестов:

### По объекту:

- Функциональное;
- Тестирование графического интерфейса пользователя
- Юзабилити (тестирование удобства пользования;
- Интернационализации;
- Локализации;
- Совместимости (кроссбраузерное, кроссплатформенное);
- Производительности (нагрузочное, стрессовое);
- Безопасности;
- Тестирование требований;
- Тестирование прототипа;
- Тестирование инсталляции.

## Функциональное тестирование (Functional Testing)

*Тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает.*

### Тестирование графического интерфейса пользователя (GUI)

*предполагает проверку соответствия приложения требованиям к графическому интерфейсу, профессионально ли оно выглядит, выполнено ли оно в едином стиле.*

*Как правило, через пользовательский интерфейс и реализуется большая часть функциональности ПО.*

## Что нужно проверять?

- Горячие клавиши (hot keys)
- Клавиатурные сокращения (shortcuts)
- Порядок табуляции
- Окна сообщений
- Изменение размеров окон приложения
- Поддержка скроллинга

### Тестирование удобства пользования (Usability testing)

*Тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности программного продукта для пользователя при условии использования в заданных условиях эксплуатации (ISO 9241)*

### Тестирование доступности (Accessibility testing)

*Тестирование, которое определяет степень легкости, с которой пользователи с ограниченными возможностями могут использовать систему или ее компоненты.*

Практически все современные приложения, рассчитанные на широкую целевую аудиторию, должны быть доступны для использования людьми с ограниченными возможностями (зрительными, слуховыми, речевыми, когнитивными, неврологическими и др.), с учетом возраста, технических предпочтений и физических способностей.

## **Критерии тестирования доступности:**

- Все изображения на сайте должны иметь альтернативный текст
- Аудио содержимое должно иметь текстовый эквивалент
- Страница должна быть читабельна при увеличенном размере шрифтов
- Страница должна читаться на монохромном экране и др.

### Тестирование интернационализации (Internationalization testing)

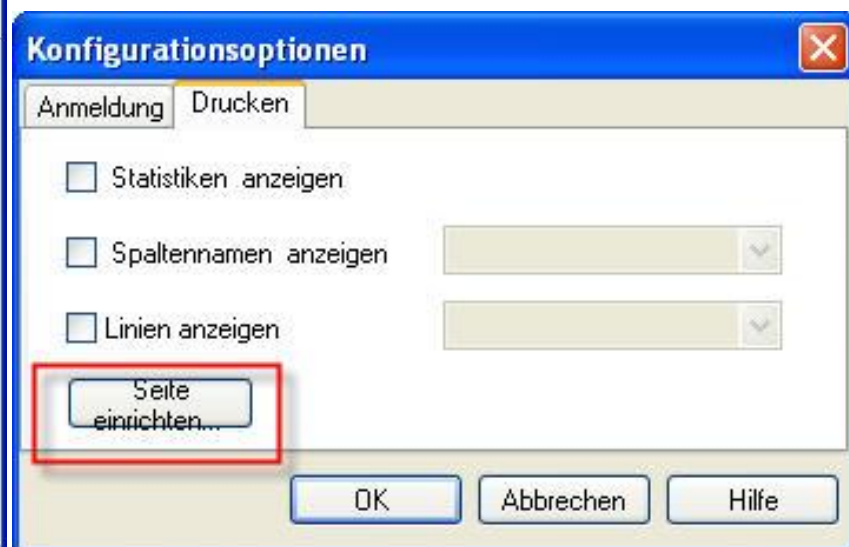
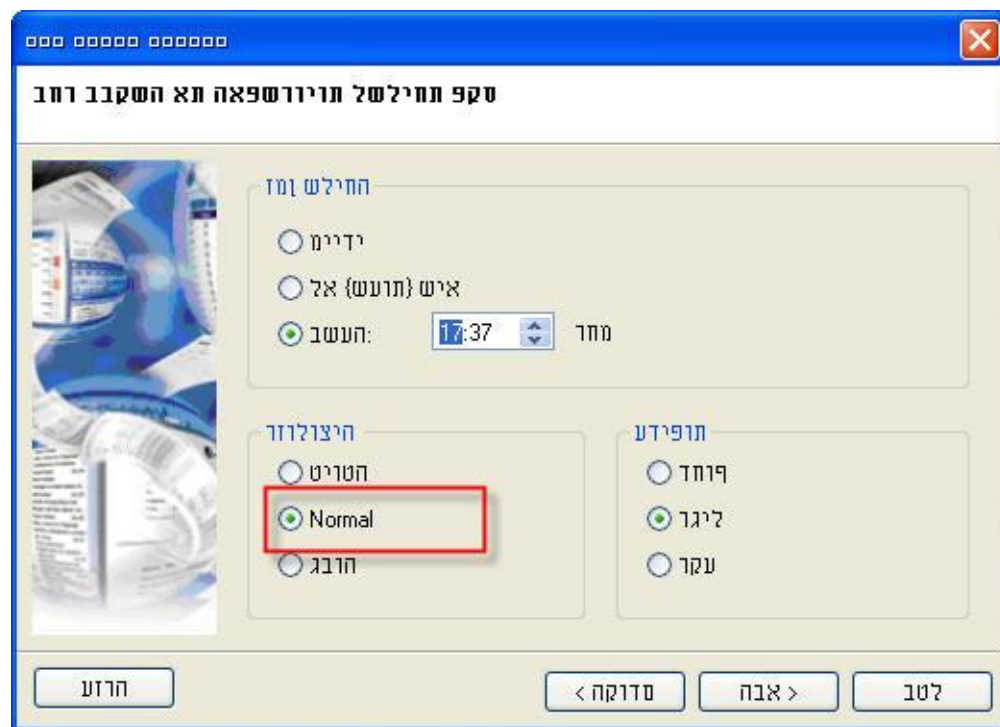
*Процесс, упрощающий дальнейшую адаптацию продукта к языковым и культурным особенностям региона, отличного от того, в котором разрабатывался продукт. Это адаптация продукта для потенциального использования практически в любом месте. Интернационализация производится на начальных этапах разработки, в то время как локализация — для каждого целевого языка.*

## Типы тестов по объекту:

### Тестирование локализации (Localization testing)

*Процесс адаптации программного обеспечения к языку какой-либо страны. Как частность - перевод пользовательского интерфейса, документации и сопутствующих файлов программы с одного языка на другой.*

## Примеры дефектов



### Кроссбраузерное тестирование (Crossbrowser testing)

*Тестирование программного продукта для определения его работоспособности на различных браузерах.*

**Цель** данного типа тестирования – удостовериться, что приложение будет эффективно работать не только с Internet Explorer, но также и с альтернативными браузерами (Safari, Opera, FireFox, Google Chrome и др.)

### Кроссплатформенное тестирование (Crossplatform testing)

*Тестирование программных продуктов на совместимость с различными операционными системами.*

**Цель** кроссплатформенного тестирования – удостовериться, что приложение работает одинаково корректно и эффективно на любых типах и версиях операционных систем (ОС - Windows, Mac OS и Linux/Unix).

### Нагрузочное тестирование (Load testing)

*Вид тестирования производительности, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке, например количестве параллельных пользователей и/или операций, а также определения какую нагрузку может выдержать компонент или система.*

### Стрессовое тестирование (Stress testing)

*Вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.*

### Тестирование безопасности (Security testing)

*представляет собой ряд действий, от разработки политики безопасности до тестирования на уровне приложения, операционной системы и сетевой безопасности.*

*В общем случае включает внедрение скриптов, операторов SQL, манипуляции с адресной строкой и исходным кодом программы*

### Тестирование требований (Requirements testing)

*Проверка требований на соответствие основным характеристикам качества и на отсутствие противоречий в самих требованиях.*

**Тестирование требований** позволяет выявить несоответствие в ожиданиях и интерпретациях разрабатываемого продукта на ранних этапах разработки и обеспечивает:

- *Снижение материальных затрат на разработку и тестирование продукта;*
- *Сокращение сроков сдачи готового продукта;*
- *Повышение качества продукта.*

## Критерии качества требований

- Корректность
- Недвусмысленность (однозначность, ясность)
- Полнота
- Непротиворечивость
- Упорядоченность по важности и стабильности
- Возможность проверки (тестируемость)
- Возможность модификации (изменяемость)
- Трассируемость (отслеживаемость)

## ***Цена ошибки***

«В 100 раз дороже исправить ошибки в требованиях, если на них указывает клиент, чем в процессе разработки этих требований»

«На исправление ошибки, выявленной на стадии работы над требованиями тратится в среднем 30 минут, тогда как на исправление ошибки, выявленной в ходе тестирования системы, необходимо от 5 до 17 часов»

## Типы тестов по объекту:

### Тестирование прототипа (Prototype testing)

*Метод выявления структурных, логических ошибок и ошибок проектирования на ранней стадии развития продукта до начала фактической разработки.*

## Основная цель тестирования

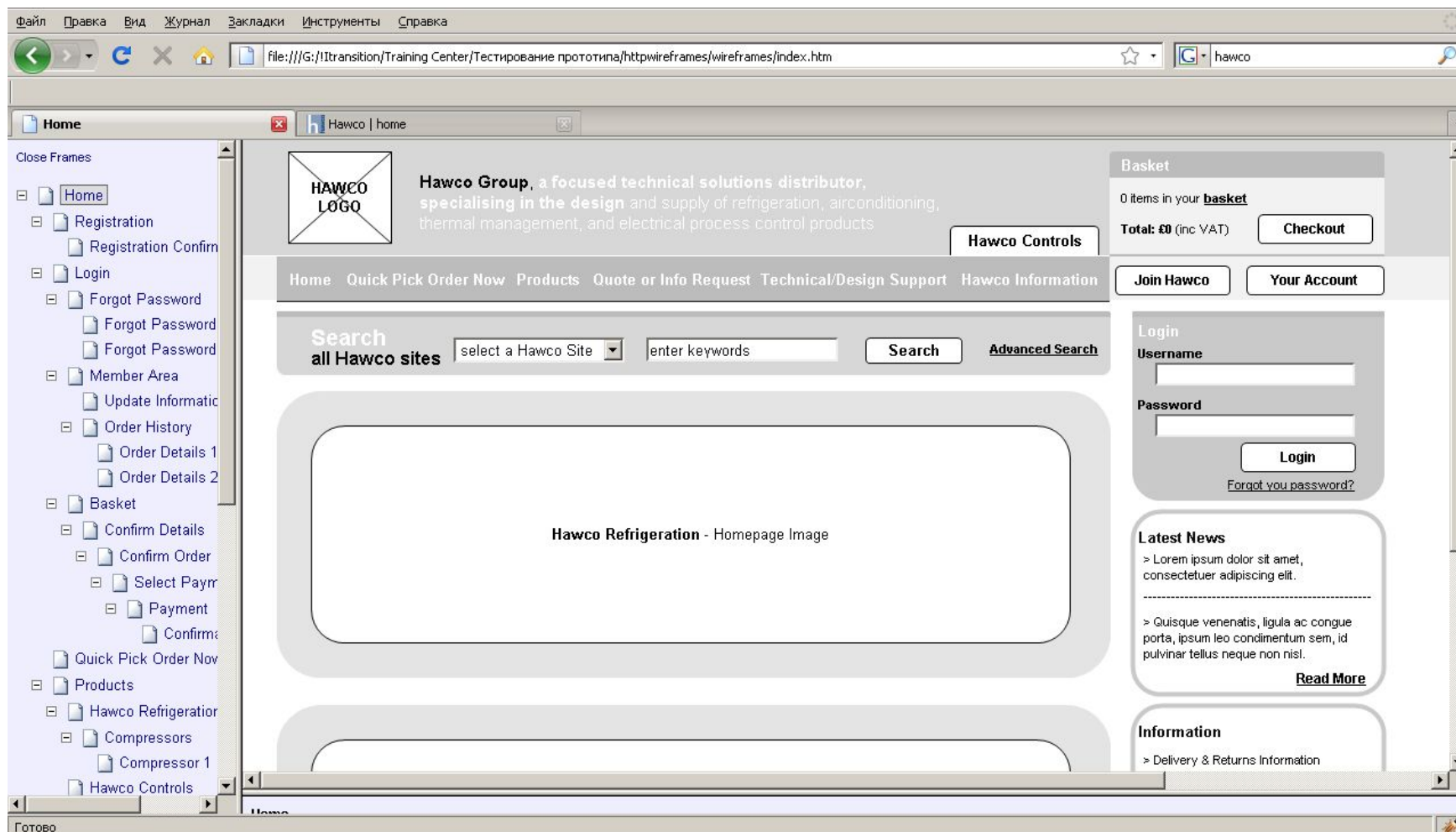
**прототипа:** выявить потенциальные проблемы в приложении, проверить, насколько приложение соответствует потребностям и ожиданиям пользователя, и обнаружить расхождения с требованиями к пользовательскому интерфейсу.

На данном этапе значительное внимание уделяется тестированию удобства пользования.

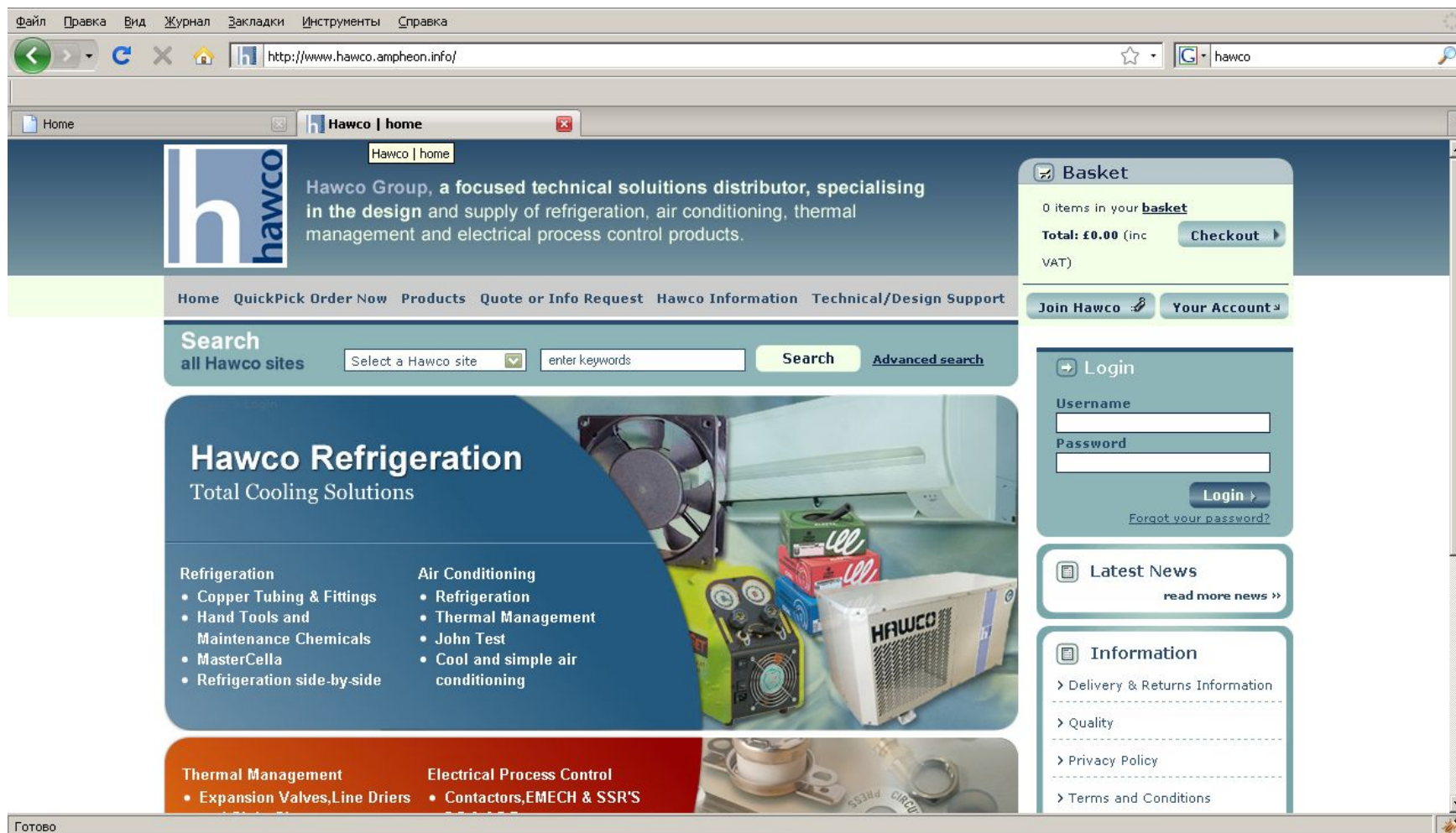
## Виды прототипов

- Интерактивный прототип
- Функциональный прототип
- Бумажный прототип
- Мокапы и нарезки

# Тестирование прототипа



# Тестирование прототипа



### Тестирование установки (Installation testing)

*Процесс тестирования устанавливаемости программного продукта.*

В ходе **тестирования установки** проверяется правильность процесса установки приложения и эффект, оказываемый установкой приложения на операционную систему конечного пользователя.

## Типы тестов:

### По покрытию:

- Smoke;
- MAT;
- AT.

## Типы тестов по покрытию:

### Смоук тест (тест на "дым", Входной тест, Тест работоспособности) (Smoke Test)

*Выборка из общего числа запланированных тестовых сценариев, покрывающая основную функциональность компонента или системы. Проводится с целью удостовериться, что базовые функции программы в целом работают корректно, без углубления в детали. Ежедневная сборка и тест "на дым" являются передовыми практическими методами.*

### Минимальное приемочное тестирование (Minimal Acceptance Test, MAT, Positive test)

*Вид теста, направленный на подтверждение того, что все функции приложения корректно работают с необходимыми данными.*

*Проверить работу приложения в определенных условиях. Для крупных и сложных приложений используется ограниченный набор сценариев и функций.*

### Приемочное тестирование (Acceptance Test, AT)

*Вид теста, направленный на подтверждение того, что приложение может использоваться по назначению при любых условиях.*

*Бизнес-логика + функциональность + проверка пользовательского интерфейса + тестирование удобства пользования + тестирование локализации (если применимо) — полная проверка интерфейса и функциональности в любых условиях. Эти тесты представляют полный набор услуг и могут проводиться в зависимости от целей конечных пользователей продукта.*

## Типы тестов:

### По знанию кода:

- Белый;
- Черный;
- Серый ящик.

## Типы тестов по знанию кода:

### Тестирование методом белого ящика (White-box testing, Clear-box testing, Code-based testing)

*Тестирование, основанное на анализе внутренней структуры компонента или системы.*

### Тестирование методом черного ящика (Black box testing)

*Тестирование, функциональное или нефункциональное, без знания внутренней структуры компонента или системы.*

## Типы тестов:

### По степени автоматизации:

- Автоматизированное;
- Ручное.

## Типы тестов по степени автоматизации:

### Автоматизация тестирования (Automatization testing)

*включает разработку скриптов, большей частью основанных на заранее определенных функциональных тестовых вариантах.*

## Типы тестов:

### По подготовленности:

- Интуитивное;
- Исследовательское;
- По документации.

## Типы тестов по подготовленности:

Свободное тестирование (Интуитивное тестирование, Спонтанное тестирование) (Ad hoc testing)

*Тестирование, выполняемое неформально; без формальной подготовки тестов, формальных методов проектирования тестов, определения ожидаемых результатов и руководства по выполнению тестирования.*

## Типы тестов по подготовленности:

### Исследовательское тестирование (Exploratory testing)

*Неформальный метод проектирования тестов, при котором тестировщик активно контролирует проектирование тестов в то время, как эти тесты выполняются, и использует полученную во время тестирования информацию для проектирования новых и улучшенных тестов*

## Типы тестов:

По месту и времени проведения:

- Альфа-, Бета- тестирование;
- Приемочное тестирование.

## Типы тестов по месту и времени проведения:



### Альфа-тестирование (Alpha testing)

*Моделируемое или действительное эксплуатационное тестирование потенциальными пользователями/заказчиками или независимой командой тестирования на стороне разработчиков, но вне разрабатывающей организации.*

## Типы тестов по месту и времени проведения:



### Бета-тестирование (Beta testing)

*Эксплуатационное тестирование потенциальными и/или существующими клиентами/заказчиками на внешней стороне никак не связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы.*

## Типы тестов:

По изолированности компонентов:

- Модульное;
- Интеграционное;
- Системное.

## Типы тестов по изолированности компонентов:



### Модульное тестирование (компонентное тестирование) (Module testing, Component testing, Unit testing)

*Тестирование отдельных компонент программного обеспечения.*

*Процесс, позволяющий проверить на корректность работы отдельные модули программы.*

*Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны.*

## Типы тестов по изолированности компонентов:

### Интеграционное тестирование (Integration testing)

*Тестирование, выполняемое для обнаружения дефектов в интерфейсах и во взаимодействии между интегрированными компонентами или системами.  
Или проверка того, как одно приложение взаимодействует с другим.*

## **Возможные проблемы:**

- дефекты самого продукта
- дефекты приложения, с которым происходит взаимодействие
- дефекты, возникающие в результате интеграции приложений

# Типы тестов по изолированности компонентов:



## Системное тестирование (System testing)

*Процесс тестирования системы в целом с целью проверки того, что она соответствует установленным требованиям.*

## Тестовые активности:

- Регрессионное тестирование;
- Валидация дефектов;
- Тестирование новых функций.

## Тестовые активности:



### Валидация дефектов (Проверка исправления дефектов) (Defect Validation, DV)

*Вид теста, направленный на проверку того, что все дефекты в данной версии продукта были исправлены.*

## Тестовые активности:

### Регрессионное тестирование (Regression testing)

*Тестирование ранее протестированной программы, проводимое 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 окружения.*

## Тестовые активности:

### Тестирование новых функций (New Features Test, NFT, CR-validation)

*Один из видов тестов, направленный на проверку полного и корректного поведения новых функций без негативных последствий для продукта.*

*Комплексное тестирование функциональности, на которую могло повлиять нововведение.*



**Спасибо, лекция  
завершена!**