

# Виды тестирования

Дисциплина «Тестирование ИС»



Все виды тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

- Функциональные;
- Нефункциональные;
- Связанные с изменениями.

Функциональные тесты базируются на **функциях и особенностях, а также на взаимодействии с другими системами**, и могут быть представлены на всех уровнях тестирования:

- компонентном;
- модульном;
- интеграционном;
- системном;
- приемочном.

Функциональные виды тестирования рассматривают внешнее поведение системы.

Виды функциональных тестов:

- ★ Функциональное тестирование (Functional testing);
  - Тестирование безопасности (Security and Access Control Testing);
  - Тестирование взаимодействия (Interoperability Testing).

## **Функциональное тестирование или Functional Testing.**

Функциональное тестирование рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом.

Тестирование функциональности может проводиться в двух аспектах:

- Требования;
- Бизнес-процессы.

## **Преимущества функционального тестирования:**

- имитирует фактическое использование системы;

## **Недостатки функционального тестирования:**

- возможность упущения логических ошибок в программном обеспечении;
- вероятность избыточного тестирования.

## Виды функциональных тестов:

- Функциональное тестирование (Functional testing);
- ★ Тестирование безопасности (Security and Access Control Testing);
- Тестирование взаимодействия (Interoperability Testing).

## **Тестирование безопасности или Security and Access Control Testing.**

**Тестирование безопасности** - это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.



Общая стратегия безопасности основывается на трех основных принципах:

- Конфиденциальность;
- Целостность;
- Доступность.

## Виды функциональных тестов:

- Функциональное тестирование (Functional testing);
- Тестирование безопасности (Security and Access Control Testing);
- ★ Тестирование взаимодействия (Interoperability Testing).

## **Тестирование взаимодействия или Interoperability Testing.**

**Тестирование взаимодействия (Interoperability Testing)** – это функциональное тестирование, проверяющее способность приложения взаимодействовать с одним и более компонентами или системами и включающее в себя тестирование совместимости (compatibility testing) и интеграционное тестирование (integration testing).

Нефункциональное тестирование описывает тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. **В целом, это тестирование того, "Как" система работает.**

Виды нефункциональных тестов:

- ★ Все виды тестирования производительности:
  - нагрузочное тестирование (Performance and Load Testing);
  - стрессовое тестирование (Stress Testing);
  - тестирование стабильности или надежности (Stability / Reliability Testing);
  - объемное тестирование (Volume Testing).
- Тестирование установки (Installation testing);
- Тестирование удобства пользования (Usability Testing);
- Тестирование на отказ и восстановление (Failover and Recovery Testing);
- Конфигурационное тестирование (Configuration Testing).

## Тестирование производительности (Performance testing)

Задачей тестирования производительности является определение масштабируемости приложения под нагрузкой, при этом происходит:

- измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций;
- определение количества пользователей, одновременно работающих с приложением;
- определение границ приемлемой производительности при увеличении нагрузки (при увеличении интенсивности выполнения этих операций);
- исследование производительности на высоких, предельных, стрессовых нагрузках.

## Стрессовое тестирование (Stress Testing)

Стрессовое тестирование позволяет проверить насколько приложение и система в целом работоспособны в условиях стресса и также оценить способность системы к регенерации, т.е. к возвращению к нормальному состоянию после прекращения воздействия стресса. Стрессом в данном контексте может быть повышение интенсивности выполнения операций до очень высоких значений или аварийное изменение конфигурации сервера.

## Объемное тестирование (Volume Testing)

Задачей объемного тестирования является получение оценки производительности при увеличении объемов данных в базе данных приложения, при этом происходит:

- измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций;
- может производиться определение количества пользователей, одновременно работающих с приложением.



## Тестирование стабильности или надежности (Stability / Reliability Testing)

Задачей тестирования стабильности (надежности) является проверка работоспособности приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки. Время выполнения операций может играть в данном виде тестирования второстепенную роль. При этом на первое место выходит отсутствие утечек памяти, перезапусков серверов под нагрузкой и другие аспекты влияющие именно на стабильность работы.

## Виды нефункциональных тестов:

- Все виды тестирования производительности:
  - нагрузочное тестирование (Performance and Load Testing);
  - стрессовое тестирование (Stress Testing);
  - тестирование стабильности или надежности (Stability / Reliability Testing);
  - объемное тестирование (Volume Testing).
- ★ Тестирование установки (Installation testing);
- Тестирование удобства пользования (Usability Testing);
- Тестирование на отказ и восстановление (Failover and Recovery Testing);
- Конфигурационное тестирование (Configuration Testing).

## **Тестирование Установки или Installation Testing**

Тестирование установки направленно на проверку успешной инсталляции и настройки, а также обновления или удаления программного обеспечения.

## Виды нефункциональных тестов:

- Все виды тестирования производительности:
  - нагрузочное тестирование (Performance and Load Testing);
  - стрессовое тестирование (Stress Testing);
  - тестирование стабильности или надежности (Stability / Reliability Testing);
  - объемное тестирование (Volume Testing).
- Тестирование установки (Installation testing);
- ★ Тестирование удобства пользования (Usability Testing);
- Тестирование на отказ и восстановление (Failover and Recovery Testing);
- Конфигурационное тестирование (Configuration Testing).

## **Тестирование удобства пользования или Usability Testing**

**Тестирование удобства пользования** - это метод тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.

# Нефункциональные виды тестирования

Тестирование удобства пользования дает оценку уровня удобства использования приложения по следующим пунктам:

- производительность, эффективность (efficiency) - сколько времени и шагов понадобится пользователю для завершения основных задач приложения, например, размещение новости, регистрации, покупка и т.д.?
- правильность (accuracy) - сколько ошибок сделал пользователь во время работы с приложением?
- активизация в памяти (recall) – как много пользователь помнит о работе приложения после приостановки работы с ним на длительный период времени?
- эмоциональная реакция (emotional response) – как пользователь себя чувствует после завершения задачи - растерян, испытал стресс? Порекомендует ли пользователь систему своим друзьям?

## Виды нефункциональных тестов:

- Все виды тестирования производительности:
  - нагрузочное тестирование (Performance and Load Testing);
  - стрессовое тестирование (Stress Testing);
  - тестирование стабильности или надежности (Stability / Reliability Testing);
  - объемное тестирование (Volume Testing).
- Тестирование установки (Installation testing);
- Тестирование удобства пользования (Usability Testing);
- ★ Тестирование на отказ и восстановление (Failover and Recovery Testing);
- Конфигурационное тестирование (Configuration Testing).

**Тестирование на отказ и восстановление (Failover and Recovery Testing)** проверяет тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться после возможных сбоев, возникших в связи с ошибками программного обеспечения, отказами оборудования или проблемами связи (например, отказ сети).

**Целью** данного вида тестирования является проверка систем восстановления (или дублирующих основной функционал систем), которые, в случае возникновения сбоев, обеспечат сохранность и целостность данных тестируемого продукта.



## Виды нефункциональных тестов:

- Все виды тестирования производительности:
  - нагрузочное тестирование (Performance and Load Testing);
  - стрессовое тестирование (Stress Testing);
  - тестирование стабильности или надежности (Stability / Reliability Testing);
  - объемное тестирование (Volume Testing).
- Тестирование установки (Installation testing);
- Тестирование удобства пользования (Usability Testing);
- Тестирование на отказ и восстановление (Failover and Recovery Testing);
- ★ Конфигурационное тестирование (Configuration Testing).

**Конфигурационное тестирование (Configuration Testing)** — специальный вид тестирования, направленный на проверку работы программного обеспечения при различных конфигурациях системы (заявленных платформах, поддерживаемых драйверах, при различных конфигурациях компьютеров и т.д.)

После проведения необходимых изменений, таких как исправление бага/дефекта, программное обеспечение должно быть **пере тестировано для подтверждения того факта, что проблема была действительно решена.**

На следующем слайде перечислены виды тестирования, которые необходимо проводить после установки программного обеспечения, для подтверждения работоспособности приложения или правильности осуществленного исправления дефекта.

Виды тестов связанных с изменениями:

- ★ Дымовое тестирование (Smoke Testing);
  - Регрессионное тестирование (Regression Testing).

## Дымовое тестирование или Smoke Testing

Понятие дымовое тестирование пошло из инженерной среды:

*«При вводе в эксплуатацию нового оборудования считалось, что тестирование прошло удачно, если из установки не пошел дым.»*

В области же программного обеспечения, дымовое тестирование рассматривается как короткий цикл тестов, выполняемый для подтверждения того, что после сборки кода (нового или исправленного) устанавливаемое приложение, стартует и выполняет основные функции.

Виды тестов связанных с изменениями:

- Дымовое тестирование (Smoke Testing);
- ★ Регрессионное тестирование (Regression Testing).

## Регрессионное тестирование или Regression Testing

Регрессионное тестирование - это вид тестирования направленный на проверку изменений, сделанных в приложении или окружающей среде (починка дефекта, слияние кода, миграция на другую операционную систему, базу данных, веб сервер или сервер приложения), для подтверждения того факта, что существующая ранее функциональность работает как и прежде. Регрессионными могут быть как функциональные, так и нефункциональные тесты.

Сам по себе термин "Регрессионное тестирование", в зависимости от контекста использования может иметь разный смысл. **Сэм Канер**, к примеру, описал 3 основных типа регрессионного тестирования:

- **Регрессия багов (Bug regression)** - попытка доказать, что исправленная ошибка на самом деле не исправлена
- **Регрессия старых багов (Old bugs regression)** - попытка доказать, что недавнее изменение кода или данных сломало исправление старых ошибок, т.е. старые баги стали снова воспроизводиться.
- **Регрессия побочного эффекта (Side effect regression)** - попытка доказать, что недавнее изменение кода или данных сломало другие части разрабатываемого приложения



Спасибо за внимание!