

# Тестирование Информационных систем

**Тестирование программного обеспечения (Software Testing)** — проверка соответствия реальных и ожидаемых результатов поведения программы, проводимая на конечном наборе тестов, выбранном определённым образом.

**Цель тестирования** — проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении, которые должны быть выявлены до того, как их обнаружат пользователи программы.

# Для чего проводится тестирование ПО?

- Для проверки соответствия требованиям.
- Для обнаружение проблем на более ранних этапах разработки и предотвращение повышения стоимости продукта.
- Обнаружение вариантов использования, которые не были предусмотрены при разработке. А также взгляд на продукт со стороны пользователя.
- Повышение лояльности к компании и продукту, т.к. любой обнаруженный дефект негативно влияет на доверие пользователей.

# Принципы тестирования

**Принцип 1** — Тестирование демонстрирует наличие дефектов (Testing shows presence of defects).

Тестирование только снижает вероятность наличия дефектов, которые находятся в программном обеспечении, но не гарантирует их отсутствия.

**Принцип 2** — Исчерпывающее тестирование невозможно (Exhaustive testing is impossible).

Полное тестирование с использованием всех входных комбинаций данных, результатов и предусловий физически невыполнимо (исключение — тривиальные случаи).

**Принцип 3** — Раннее тестирование (Early testing).

Следует начинать тестирование на ранних стадиях жизненного цикла разработки ПО, чтобы найти дефекты как можно раньше.

**Принцип 4** — Парадокс пестицида (Pesticide paradox).

Если повторять те же тестовые сценарии снова и снова, в какой-то момент этот набор тестов перестанет выявлять новые дефекты.

**Принцип 5** — Тестирование зависит от контекста (Testing is context depending). Тестирование проводится по-разному в зависимости от контекста. Например, программное обеспечение, в котором критически важна безопасность, тестируется иначе, чем новостной портал.

**Принцип 6** — Заблуждение об отсутствии ошибок (Absence-of-errors fallacy). Отсутствие найденных дефектов при тестировании не всегда означает готовность продукта к релизу. Система должна быть удобна пользователю в использовании и удовлетворять его ожиданиям и потребностям.

**Качество программного обеспечения (Software Quality)** - это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

**Обеспечение качества** (QA — Quality Assurance) и **контроль качества** (QC — Quality Control) — эти термины похожи на взаимозаменяемые, но разница между обеспечением качества и контролем качества все-таки есть, хоть на практике процессы и имеют некоторую схожесть.

**QC (Quality Control) — Контроль качества продукта** — анализ результатов тестирования и качества новых версий выпускаемого продукта.

К задачам контроля качества относятся:

- проверка готовности ПО к релизу;
- проверка соответствия требований и качества данного проекта.

**QA (Quality Assurance) — Обеспечение качества продукта —** изучение возможностей по изменению и улучшению процесса разработки, улучшению коммуникаций в команде, где тестирование является только одним из аспектов обеспечения качества.

К задачам обеспечения качества относятся:

- проверка технических характеристик и требований к ПО;
- оценка рисков;
- планирование задач для улучшения качества продукции;
- подготовка документации, тестового окружения и данных;
- тестирование;
- анализ результатов тестирования, а также составление отчетов и других документов.

# Характеристики качества ПО

**Функциональность (Functionality)** - определяется способностью ПО решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя, при заданных условиях использования ПО. Т.е. эта характеристика отвечает то, что ПО работает исправно и точно, функционально совместимо соответствует стандартам отрасли и защищено от несанкционированного доступа.

**Надежность (Reliability)** – способность ПО выполнять требуемые задачи в обозначенных условиях на протяжении заданного промежутка времени или указанное количество операций. Атрибуты данной характеристики – это завершенность и целостность всей системы, способность самостоятельно и корректно восстанавливаться после сбоев в работе, отказоустойчивость.

**Удобство использования (Usability)** – возможность легкого понимания, изучения, использования и привлекательности ПО для пользователя.

**Эффективность (Efficiency)** – способность ПО обеспечивать требуемый уровень производительности, в соответствии с выделенными ресурсами, временем и другими обозначенными условиями.

**Удобство сопровождения (Maintainability)** – легкость, с которой ПО может анализироваться, тестироваться, изменяться для исправления дефектов для реализации новых требований, для облегчения дальнейшего обслуживания и адаптирования к имеющемуся окружению.

**Портативность (Portability)** – характеризует ПО с точки зрения легкости его переноса из одного окружения (software/ hardware) в другое.

# Качество ПО

```
graph TD; A[Качество ПО] --- B[Функциональность]; A --- C[Надежность]; A --- D[Удобство использования]; A --- E[Эффективность]; A --- F[Удобство сопровождения]; A --- G[Портативность];
```

## Функциональность:

- функциональная исправность;
- соответствие стандартам;
- функциональная совместимость;
- безопасность;
- точность.

## Надежность:

- завершенность;
- восстанавливаемость;
- устойчивость к отказам.

## Удобство использования:

- удобство изучения;
- понятность;
- удобство и простота использования.

## Эффективность:

- эффективность по времени;
- эффективность использования ресурсов.

## Удобство сопровождения:

- стабильность;
- анализируемость;
- контролепригодность;
- изменяемость.

## Портативность:

- удобство установки;
- заменяемость;
- совместимость.

Документацию, которая используется на проектах по разработке ПО, можно условно разделить на две группы:

**Проектная документация** — включает в себя всё, что относится к проекту в целом.

**Продуктовая документация** — часть проектной документации, выделяемая отдельно, которая относится непосредственно к разрабатываемому приложению или системе.

## **Этапы тестирования:**

- Анализ продукта
- Работа с требованиями
- Разработка стратегии тестирования и планирование процедур контроля качества
- Создание тестовой документации
- Тестирование прототипа
- Основное тестирование
- Стабилизация
- Эксплуатация

**Стадии разработки ПО** — этапы, которые проходят команды разработчиков ПО, прежде чем программа станет доступной для широкого круга пользователей.

Программный продукт проходит следующие стадии:

- анализ требований к проекту;
- проектирование;
- реализация;
- тестирование продукта;
- внедрение и поддержка

**Аккаунт-менеджер** - менеджер по работе с клиентами, специалист, который работает с клиентами компании и обеспечивает их лояльность. Аккаунт-менеджер обеспечивает выполнение всех необходимых клиенту задач, находит к каждому заказчику индивидуальный подход, поддерживает с ним хорошие отношения (даже после того, как все заказы уже выполнены), предлагает ему новые услуги и продукты.

**Менеджер проекта** (Project manager, руководитель проекта, проект-менеджер; сокращенно - РМ, ПМ, РП) - лицо, ответственное за управление проектом. Менеджер проекта несет ответственность за достижение целей проекта в рамках бюджета, в срок и с заданным уровнем качества.

**Системный аналитик** (аналитик) является “мостиком” между заказчиком и членами команды. Переводит пожелания заказчика в формат точно описанных технических заданий.

**Системный архитектор** (архитектор) проектирует разрабатываемую систему на самом верхнем уровне и принимает ключевые решения по поводу технологий и методологий разработки. Активно занимается исследованиями и экспериментами, рисует многочисленные диаграммы и документирует архитектурные решения.

**Программист (разработчик)** пишет код на языках программирования, т.е. непосредственно кодирует логику работы программы. Также является ее первым пользователем и тестировщиком. Непосредственно отвечает за то, что программа работает и работает правильно (в соответствии с техническим заданием).

**Ведущий программист** (технический лидер, техлид) - программист, который с технической точки зрения принимает решения о формате реализации функционала и координирует работу команды разработчиков.

**QA-специалист** - специалист, который обеспечивает качество продукта (тестирует, контролирует и управляет качеством продукта).

**SDET-специалист** (контроль качества, автоматизация тестирования) - специалист, который проверяет и отвечает за качество продукта. Пишет код для автоматизации процесса тестирования на разных языках программирования. Помогает команде разработки с точки зрения технических вопросов, вопросов архитектуры и построения приложения

**QA lead** (ведущий специалист по управлению и контролю качества) - QA-специалист, который руководит командой тестирования.

**Тимлид** — лидер команды, обеспечивающий достижение проектных целей посредством организации работы команды, состоящей из сотрудников различных направлений компании, а также отвечающий за развитие участников команды, построение коммуникаций (как внутри, так и извне), дисциплину и управление составом команды.