


# Тестирование Основные определения



Мухортова Н.Н.



# Тестирование

---

Тестирование ПС - это процесс выполнения программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ.

Указанный набор данных называется тестовым или просто тестом.

# Тестирование

---

Тестирование программного обеспечения (Software Testing) - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. [IEEE Guide to Software Engineering Body of Knowledge, SWEBOOK, 2004]

# Тестирование

---

В более широком смысле, тестирование - это одна из техник контроля качества, включающая в себя активности по

1. планированию работ (Test Management),
2. проектированию тестов (Test Design),
3. выполнению тестирования (Test Execution)
4. и анализу полученных результатов (Test Analysis).

# Верификация

---

Верификация (Verification) - это процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа

# Валидация

---

Валидация (Validation) - это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе

Online validator for HTML, CSS

<https://validator.w3.org/>

# План Тестирования

---

План Тестирования (Test Plan) - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения

# Пример



## Test Plan Table Example

### CompuTech Website Test Plan for Firefox

HomePage

Test	User Requirement Addressed	Expected Result	Actual Result	Pass / Fail	Date	Corrective Action
Logo Displayed	The website uses the companies house style	CompuTech logo is displayed	As expected	Pass	01/12/09	None
Navigation Bar Appears	Website is easy to navigate	Navigation bar appears	As expected	Pass	01/12/09	None
Navigation Bar Drop Downs Work	Website is easy to navigate	When mouse goes over labels - menus drop down	Bar is static and drops do not work	Fail	01/12/09	Recode the JavaScript as it only works with IE
Navigation Bar Hyperlinks Work	Website is easy to navigate	When mouse goes over turns to red and can click on it to take you to correct page	As expected	Pass	01/12/09	Now I have fixed the previous error all hyperlinks work properly



# Тест дизайн

---

Тест дизайн (Test Design) - это этап процесса тестирования ПО, на котором проектируются и создаются тестовые случаи (тест кейсы), в соответствии с определёнными ранее критериями качества и целями тестирования

# Тестовый случай

---

Тестовый случай (Test Case) - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

## Пример Test Case.

ID	Номер требований	Исходные требования	Шаги	Ожидаемый результат			
No	Requirement	Pre-conditions	Test Case	Steps	Input Data	Expected Result	PASSED/FAILED/SKIPPED
Some Application							
1			Some Panel Name				
	1,2,3,4,5	Testing Some Panel features	1. Go to the Some Panel			1. There are several fields at the panel: Name, Surname and Age.	passed
			2. Enter special symbols at the field Name			2. The application must send an error "Wrong input for the field Name. Only chars are allowed"	passed

Заголовок

Не все Test Cases одинаковы, но примерный вид у всех

Пройден или нет

# Баг/Дефект Репорт

---

Баг/Дефект Репорт (Bug Report) - это документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

# Баг Репорт (Bug Report)

Шапка	
Короткое описание (Summary)	Короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации.
Проект (Project)	Название тестируемого проекта
Компонент приложения (Component)	Название части или функции тестируемого продукта
Номер версии (Version)	Версия на которой была найдена ошибка
Серьезность (Severity)	Наиболее распространена пятиуровневая система градации серьезности дефекта: <ul style="list-style-type: none"><li>• S1 Блокирующий (Blocker)</li><li>• S2 Критический (Critical)</li><li>• S3 Значительный (Major)</li><li>• S4 Незначительный (Minor)</li><li>• S5 Тривиальный (Trivial)</li></ul>
Приоритет (Priority)	Приоритет дефекта: <ul style="list-style-type: none"><li>• P1 Высокий (High)</li><li>• P2 Средний (Medium)</li><li>• P3 Низкий (Low)</li></ul>

# Тестовое Покрытие

---

Тестовое Покрытие (Test Coverage) - это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода

# Детализация и Время Прохождения

---

Детализация Тест Кейсов (Test Case Specification) - это уровень детализации описания тестовых шагов и требуемого результата, при котором обеспечивается разумное соотношение времени прохождения к тестовому покрытию

Время Прохождения Тест Кейса (Test Case Pass Time) - это время от начала прохождения шагов тест кейса до получения результата теста.

# Задачи тест дизайна

---

Первая задача: подготовить такой набор тестов и применить к ним ПС, чтобы обнаружить в нем по возможности большее число ошибок. Однако чем дольше продолжается процесс тестирования (и отладки в целом), тем большей становится стоимость ПС.

Вторая задача: определить момент окончания отладки ПС (или отдельной его компоненты). Признаком возможности окончания отладки является полнота охвата пропущенными через ПС тестами множества различных ситуаций, возникающих при выполнении программ ПС, и относительно редкое проявление ошибок в ПС на последнем отрезке процесса тестирования.



# Оптимизация

---

Для оптимизации набора тестов, т.е. для подготовки такого набора тестов, который позволял бы при заданном их числе (или при заданном интервале времени, отведенном на тестирование) обнаруживать большее число ошибок в ПС, необходимо:

- во-первых, заранее планировать этот набор и,
- во-вторых, использовать рациональную стратегию планирования тестов.

# Зачем нужна стратегия тестирования

---

- Для организации процесса в условиях ограниченных ресурсов. Поэтому для начала неплохо бы осознать, какими ресурсами мы располагаем.
- Для того, чтобы все участники проекта понимали роль тестирования, что оно может дать, какие профиты мы с этого получим. Чтобы у всех были равные ожидания и понимание, что вообще происходит в области контроля качества. А также для выявления возможных проблем, которые неизбежно станут очевидными в процессе обсуждения.

# Какие типы тестирования применять

---

Виды тестирования программного обеспечения

1. Функциональные
2. Нефункциональные
3. Связанные с изменениями

# Функциональные виды тестирования

---

1. Функциональное тестирование (Functional testing)
2. Тестирование безопасности (Security and Access Control Testing)
3. Тестирование взаимодействия (Interoperability Testing)

# Функциональное тестирование

---

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

Функциональные тесты основываются на функциях, выполняемых системой, и могут проводиться на всех уровнях тестирования (компонентном, интеграционном, системном, приемочном).

Как правило, эти функции описываются в требованиях, функциональных спецификациях или в виде случаев использования системы (use cases).

# Функциональное тестирование

---

Преимущества функционального тестирования:

- имитирует фактическое использование системы;

Недостатки функционального тестирования:

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

# Тестирование безопасности

---

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.

Необходимо проверить программное обеспечение на наличия известных видов уязвимостей

# Тестирование безопасности

---

Примеров уязвимостей и атак существует огромное количество. Даже проведя полный цикл тестирования безопасности, нельзя быть на 100% уверенным, что система по-настоящему обезопасена.

Процент несанкционированных проникновений, краж информации и потерь данных будет в разы меньше, чем у тех кто не проводил тестирования безопасности.



# Тестирование взаимодействия

---

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).

# Нефункциональные виды тестирования

---

1. Все виды тестирования производительности
2. Тестирование установки (Installation testing)
3. Тестирование удобства пользования (Usability Testing)
4. Тестирование на отказ и восстановление (Failover and Recovery Testing)
5. Конфигурационное тестирование (Configuration Testing)

# Виды тестирования производительности

---

1. нагрузочное тестирование (Performance and Load Testing)
2. стрессовое тестирование (Stress Testing)
3. тестирование стабильности или надежности (Stability / Reliability Testing)
4. объемное тестирование (Volume Testing)

# Нагрузочное тестирование

---

Нагрузочное тестирование или тестирование производительности - это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком-либо общем (разделяемом ими) ресурсе.

# Нагрузочное тестирование

---

1. Измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций
2. Определение количества пользователей, одновременно работающих с приложением
3. Определение границ приемлемой производительности при увеличении нагрузки (при увеличении интенсивности выполнения этих операций)
4. Исследование производительности на высоких, предельных, стрессовых нагрузках

# Стрессовое тестирование

---

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.

Также одной из задач при стрессовом тестировании может быть оценка деградации производительности, таким образом цели стрессового тестирования могут пересекаться с целями тестирования производительности.

# Тестирование стабильности

---

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.

# Объемное тестирование

---

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 Установки

---

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

В настоящий момент наиболее распространена установка ПО при помощи инсталляторов (специальных программ, которые сами по себе так же требуют надлежащего тестирования)

# Тестирование удобства пользования

---

Тестирование удобства пользования - это метод тестирования, направленный на

- установление степени удобства использования,
- обучаемости, понятности
- и привлекательности для пользователей разрабатываемого продукта

в контексте заданных условий

# Тестирование на отказ и восстановление

---

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 основную функционал систем), которые, в случае возникновения сбоев, обеспечат сохранность и целостность данных тестируемого продукта.

# Конфигурационное тестирование

---

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.)

# Связанные с изменениями виды тестирования

---

1. Дымовое тестирование (Smoke Testing)
2. Регрессионное тестирование (Regression Testing)
3. Тестирование сборки (Build Verification Test)
4. Санитарное тестирование или проверка согласованности/исправности (Sanity Testing)

# Дымовое тестирование

---

"При вводе в эксплуатацию нового оборудования ("железа") считалось, что тестирование прошло удачно, если из установки не пошел дым."

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.

# Дымовое тестирование

---

Вывод о работоспособности основных функций делается на основании результатов поверхностного тестирования наиболее важных модулей приложения на предмет возможности выполнения требуемых задач и наличия быстронаходимых критических и блокирующих дефектов.

В случае отсутствия таковых дефектов дымовое тестирование объявляется пройденным, и приложение передается для проведения полного цикла тестирования, в противном случае, дымовое тестирование объявляется проваленным, и приложение уходит на доработку.

# Регрессионное тестирование

---

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



# Тестирование сборки

---

Тестирование направленное на определение соответствия, выпущенной версии, критериям качества для начала тестирования.

По своим целям является аналогом Дымового Тестирования, направленного на приемку новой версии в дальнейшее тестирование или эксплуатацию.

Вглубь оно может проникать дальше, в зависимости от требований к качеству выпущенной версии.

# Санитарное тестирование

---

Санитарное тестирование - это узконаправленное тестирование достаточное для доказательства того, что конкретная функция работает согласно заявленным в спецификации требованиям.

Является подмножеством регрессионного тестирования.

Используется для определения работоспособности определенной части приложения после изменений произведенных в ней или окружающей среде. Обычно выполняется вручную.

# Источники

---

<http://www.protesting.ru/>

[https://habr.com/ru/company/true\\_engineering/blog/428053/](https://habr.com/ru/company/true_engineering/blog/428053/)