
Планирование и организация процесса тестирования

Тест-план и его ключевые секции. Уровни и виды тестирования. Критерии качества. Типичные ошибки тестирования и универсальные рекомендации.

Содержание

1. Продукты, подвергаемые тестированию.
2. Планирование процесса тестирования, тест-план и его ключевые секции.
3. Функциональное и нефункциональное тестирование.
4. Классификация тестирования:
 - Направления и методы тестирования.
 - Уровни тестирования.
 - Виды тестирования.
5. Критерии качества.
6. Типичные ошибки тестирования и универсальные рекомендации.

Продукты, подвергаемые тестированию

Основные продукты, подверженные тестированию

Тестированию
подвергается

Програм-
ма



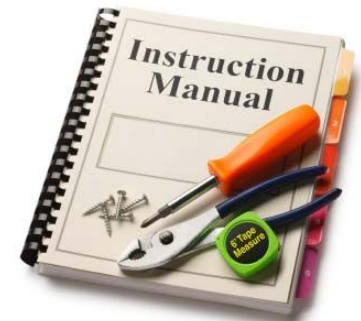
Код



Прототип



Докумен-
тация



Основные продукты, подвергаемые тестированию

Программа (software) – при непосредственном запуске и исполнении.

Код (code) программы – без запуска и исполнения или с исполнением отдельных участков.

Прототип программного продукта (product prototype) – как источник требований, способ получения обратной связи или способ оценки качества.

Документация (documentation) – проектная, пользовательская и любая иная.

**Планирование процесса
тестирования, тест-план и
его ключевые секции**

Планирование

Планирование (planning) – непрерывный процесс принятия управленческих решений и методической организации усилий по их реализации **с целью обеспечения качества** некоторого процесса на протяжении длительного периода времени.



Задачи планирования

Задачи планирования:

- оценка объёма и сложности работ;
- определение необходимых ресурсов и источников их получения;
- определение расписания, сроков и ключевых точек;
- оценка рисков и подготовка превентивных контрмер;
- распределение обязанностей и ответственности;
- согласование работ по тестированию с деятельностью участников проектной команды, занимающихся другими задачами.

Тест-план

Тест-план (test plan) – документ, описывающий и регламентирующий перечень работ по тестированию, а также соответствующие техники и подходы, стратегию, области ответственности, ресурсы, расписание и ключевые даты.



Ключевые секции тест-плана

Цель (purpose) – предельно краткое описание цели разработки приложения.

Частично эта секция напоминает бизнес-требования, но здесь информация подаётся в ещё более сжатом виде с акцентом на самых главных задачах.

Пример:

Корректное автоматическое преобразование содержимого документов к единой кодировке с производительностью, значительно превышающей производительность человека при выполнении аналогичной задачи.

Ключевые секции тест-плана

Области, подвергаемые тестированию (features to be tested) – перечень функций и/или нефункциональных особенностей приложения, которые будут подвергнуты тестированию.

Пример:

- *ПТ-1.*: дымовой тест.*
- *ПТ-2.*: дымовой тест, тест критического пути.*
- *ПТ-3.*: тест критического пути.*
- *БП-1.*: дымовой тест, тест критического пути.*
- *АК-1.*: дымовой тест, тест критического пути.*

Ключевые секции тест-плана

Области, не подвергаемые тестированию (features not to be tested) – перечень функций и/или нефункциональных особенностей приложения, которые не будут подвергнуты тестированию.

Пример:

- *СХ-1: приложение разрабатывается как консольное.*
- *О-3: не требует реализации.*
- *О-6: не требует реализации.*

Ключевые секции тест-плана

Тестовая стратегия (test strategy) и **подходы** (test approach) – описание процесса тестирования с точки зрения применяемых методов, подходов, видов тестирования, технологий, инструментальных средств и т.д.

Пример:

Уровни функционального тестирования:

- *Дымовой тест: автоматизированный с использованием командных файлов ОС Windows и Linux.*
- *Тест критического пути: выполняется вручную.*
- *Расширенный тест не производится, т.к. для данного приложения вероятность обнаружения дефектов на этом уровне пренебрежимо мала.*

В силу кроссфункциональности команды значительного вклада в повышение качества можно ожидать от аудита кода, совмещённого с ручным тестированием по методу белого ящика. Автоматизация тестирования кода не будет применяться в силу крайне ограниченного времени..

Ключевые секции тест-плана

Критерии:

- Приёмочные критерии (acceptance criteria).
- Критерии начала тестирования (entry criteria).
- Критерии приостановки тестирования (suspension criteria)
- Критерии возобновления тестирования (resumption criteria).
- Критерии завершения тестирования (exit criteria).

Пример:

Критерии возобновления тестирования: исправление более 50% обнаруженных на предыдущей итерации дефектов (см. метрику «Текущее устранение дефектов»).

Ключевые секции тест-плана

Ресурсы (resources):

- программные ресурсы (software)
- аппаратные ресурсы (hardware);
- человеческие ресурсы (staff);
- временные ресурсы (time);
- финансовые ресурсы (finance).

Пример:

- *Программные ресурсы: четыре виртуальных машины (две с ОС Windows 7 Ent x64, две с ОС Linux Ubuntu 14 LTS x64), две копии PHP Storm 8.*
- *Аппаратные ресурсы: две стандартных рабочих станции (8GB RAM, i7 3GHz).*

Ключевые секции тест-плана

Расписание (test schedule) – фактически, это календарь, в котором указано, что и к какому моменту должно быть сделано.

Пример:

- *25.05 – формирование требований.*
- *26.05 – разработка тест-кейсов и скриптов для автоматизированного тестирования.*
- *27.05-28.05 – основная фаза тестирования (выполнение тест-кейсов, написание отчётов о дефектах).*
- *29.05 – завершение тестирования и подведение итогов.*

Ключевые секции тест-плана

Роли и ответственность (roles and responsibility) – перечень необходимых ролей (например, «ведущий тестировщик», «эксперт по оптимизации производительности») и область ответственности специалистов, выполняющих эти роли.

Пример:

- *Старший разработчик: участие в формировании требований, участие в аудите кода.*
- *Тестировщик: формирование тестовой документации, реализация тестирования, участие в аудите кода.*

Ключевые секции тест-плана

Оценка рисков (risk evaluation). Перечень рисков, которые с высокой вероятностью могут возникнуть в процессе работы над проектом. По каждому риску даётся оценка представляемой им угрозы и приводятся варианты выхода из ситуации.

Пример:

- *Время (вероятность высокая): заказчиком обозначен крайний срок сдачи 01.06, потому время является критическим ресурсом. Рекомендуется приложить максимум усилий к тому, чтобы фактически завершить проект 28.05 с тем, чтобы один день (29.05) остался в запасе.*

Ключевые секции тест-плана

Документация (documentation). Перечень используемой тестовой документации с указанием, кто и когда должен её готовить и кому передавать.

Пример:

- *Требования. Ответственный – тестировщик, дата готовности 25.05.*
- *Тест-кейсы и отчёты о дефектах. Ответственный – тестировщик, период со-здания 26.05-28.05.*
- *Отчёт о результатах тестирования. Ответственный – тестировщик, дата готовности 29.05.*

Ключевые секции тест-плана

Метрики (metrics) – числовые характеристики показателей качества, способы их оценки, формулы и т.д.

Пример:

- *Покрытие требований тест-кейсами:*

$$R^C = \frac{R^{Covered}}{r^{Total}} \cdot 100\%$$

где

R^C – процентный показатель покрытия требования тест-кейсами,

$R^{Covered}$ – количество покрытых тест-кейсами требований,

R^{Total} – общее количество требований.

Минимальные границы значений:

- *Начальная фаза проекта: 40%.*
- *Основная фаза проекта: 60%.*
- *Финальная фаза проекта: 80% (рекомендуется 90% и более).*

Преимущества качественного тест-плана

- Хорошо распланированный, систематизированный подход позволяет достичь **лучших результатов**, а также позволяет обнаруживать большее количество ошибок, чем неорганизованная, плохо распланированная деятельность.
- Тест-план позволяет **управлять процессом** тестирования более эффективно.
- Тест-план позволяет **увидеть и понять минимальный уровень тестирования** и получить представление об уровне проводимого тестирования каждой области продукта.
- Тест-план позволяет **достичь соглашения между исполнителями, заказчиком и менеджером**, о том, каким образом и в какие сроки будет проводиться тестирование.

Функциональное и нефункциональное тестирование

Функциональное и нефункциональное тестирование

Функциональное тестирование (functional testing) – вид тестирования, направленный на проверку корректности работы функциональности приложения (корректность реализации функциональных требований).

Нефункциональное тестирование (non-functional testing) – вид тестирования, направленный на проверку нефункциональных особенностей приложения (корректность реализации нефункциональных требований), таких как удобство использования, совместимость, производительность, безопасность и т.д.

Цели

Выполнение функционального и нефункционального тестирования преследует такие цели как:

- Обнаружение дефектов в программном продукте (и создание условий для их устранения).
- Определение степени соответствия программного продукта требованиям и ожиданиям заказчика.
- Принятие решения о текущем состоянии качества продукта и возможности передачи продукта заказчику.

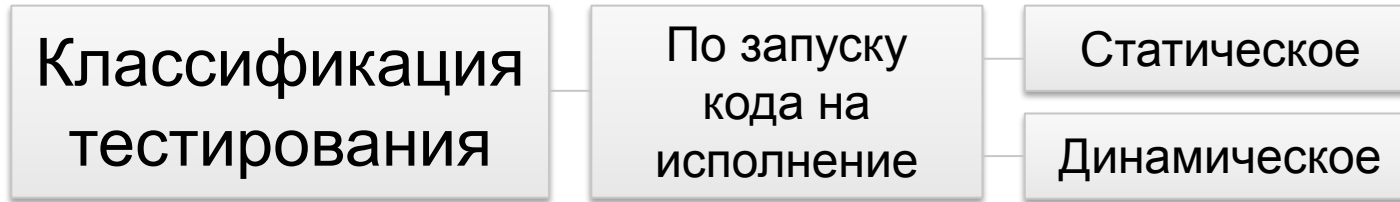
Классификация тестирования

Общая схема классификации тестирования



Направления и методы тестирования

Направления тестирования



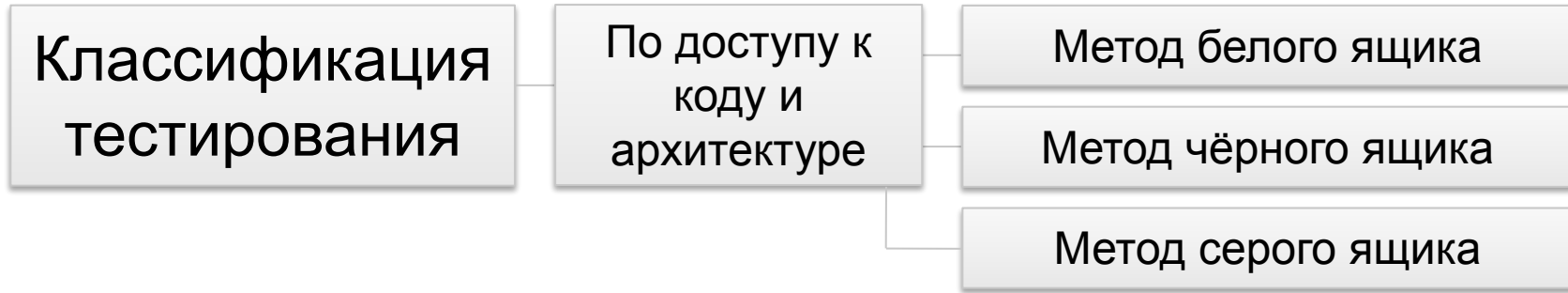
Статическое тестирование (static testing) – без запуска кода на исполнение.



Динамическое тестирование (dynamic testing) – с запуском кода на исполнение.



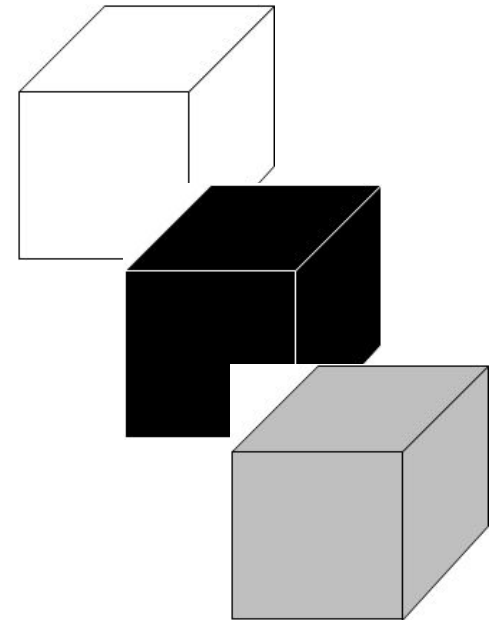
Методы тестирования



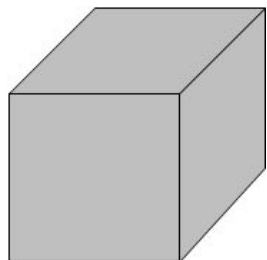
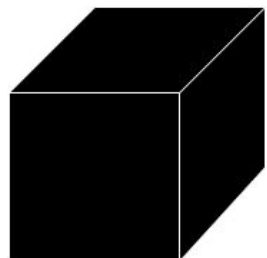
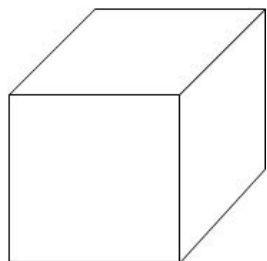
Метод белого ящика (white-box testing) – тестировщик имеет доступ к коду.

Метод чёрного ящика (black-box testing) – тестировщик не имеет доступа к коду.

Метод серого ящика (gray-box testing) – к части кода доступ есть, к части – нет.



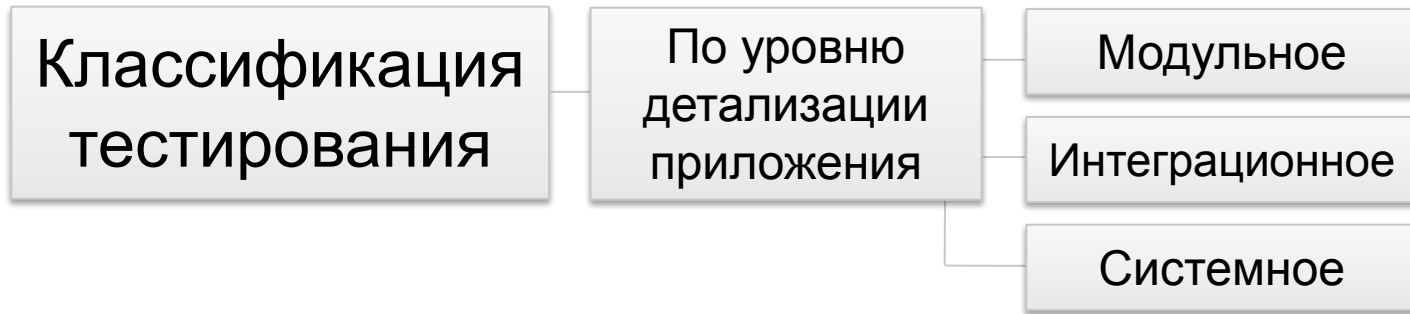
Методы тестирования



Преимущества	Недостатки
<ul style="list-style-type: none">• Показывает скрытые проблемы.• Упрощает диагностику проблемы.• Допускает тестирование на ранних стадиях.	<ul style="list-style-type: none">• Необходим квалифицированный персонал.• Сложно эмулировать работу пользователя.
<ul style="list-style-type: none">• Нет необходимости в знании программирования.• Тест-кейсы выполняются с точки зрения конечного пользователя.• Тест-кейсы можно разрабатывать, как только появились требования.	<ul style="list-style-type: none">• Возможна избыточность некоторых тест-кейсов.• Сложно разрабатывать тест-кейсы для незнакомого, «концептуального» приложения.• Сложнее прогнозировать объём работ.
<p>Объединяет в себе преимущества и недостатки методов белого и чёрного ящика.</p>	

Уровни тестирования

Уровни тестирования



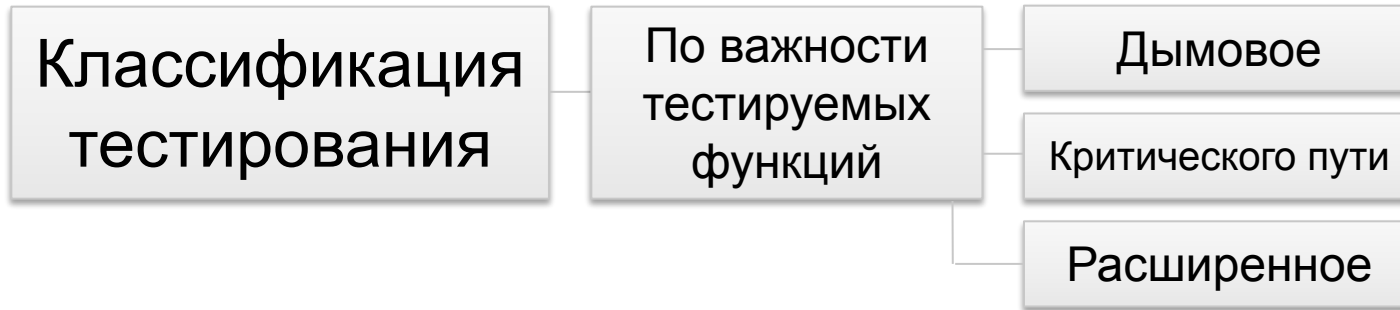
Компонентное тестирование (component testing, unit testing) – проверяются отдельные небольшие части приложения.

Интеграционное тестирование (integration testing) – проверяется взаимодействие между несколькими частями приложения.

Системное тестирование (system testing) – приложение проверяется как единое целое.



Уровни функционального тестирования



Дымовое тестирование (smoke test) – проверка самой важной функциональности, неработоспособность которой делает бессмысленной саму идею использования приложения.



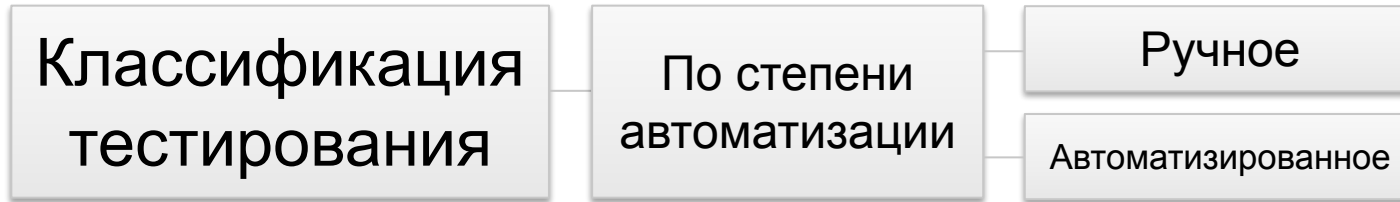
Тестирование критического пути (critical path test) – проверка функциональности, используемой типичными пользователями в типичной повседневной деятельности.



Расширенное тестирование (extended test) – проверка всей (остальной) функциональности, заявленной в требованиях.

Виды тестирования

Виды тестирования



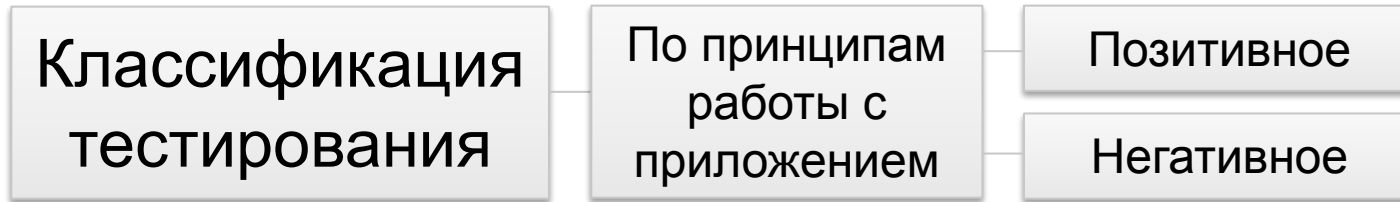
Ручное тестирование (manual testing) – тест-кейсы выполняет человек.



Автоматизированное тестирование (automated testing) – тест-кейсы частично или полностью выполняет специальное инструментальное средство.



Виды тестирования



Позитивное тестирование (positive testing) – работа с приложением выполняется по инструкции без никаких недопустимых действий, некорректных данных и т.д.



Негативное тестирование (negative testing) – в работе с приложением выполняются (некорректные) операции и используются данные, потенциально приводящие к ошибкам (классика жанра – деление на ноль).



ВНИМАНИЕ! ОЧЕНЬ ВАЖНО!

Негативное тестирование (negative testing) – в работе с приложением выполняются (некорректные) операции и используются данные, потенциально приводящие к ошибкам (классика жанра – деление на ноль).



Негативные тесты НЕ предполагают возникновения в приложении ошибки. Напротив – они предполагают, что верно работающее приложение даже в критической ситуации поведёт себя правильным образом (в примере с делением на ноль, допустим, отобразит сообщение «Делить на ноль запрещено»).

... и ещё несколько видов ...

Виды тестирования

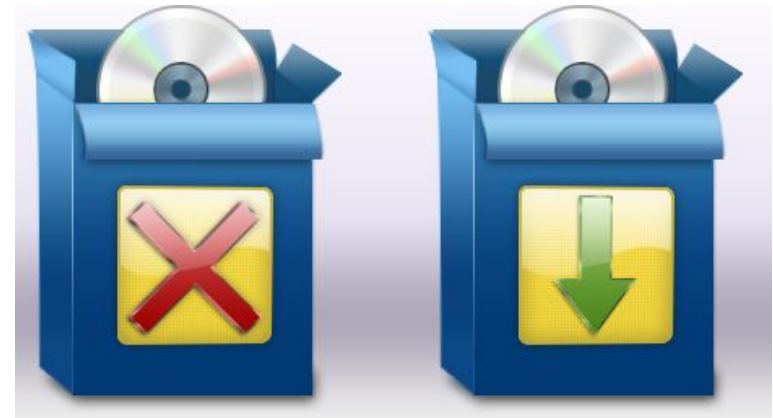
Классификация
тестирования

По целям и
задачам

Инсталляционное

Инсталляционное тестирование (installation testing) – тестирование, направленное на выявление дефектов, влияющих на протекание стадии инсталляции (установки) приложения.

uninstallation testing



Виды тестирования

Классификация
тестирования

По целям и
задачам

Регрессионное

Регрессионное тестирование (regression testing) – тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вызванные изменениями в приложении или среде его функционирования.



Виды тестирования

Классификация
тестирования

По целям и
задачам

Совместимости

Тестирование совместимости (compatibility testing) – тестирование, направленное на проверку способности приложения работать в указанном окружении. Проверяется:

- Совместимость с аппаратной платформой, операционной системой и сетевой инфраструктурой.
- Совместимость с браузерами и их версиями.
- Совместимость с мобильными устройствами.
- И так далее.



Виды тестирования

Классификация
тестирования

По целям и
задачам

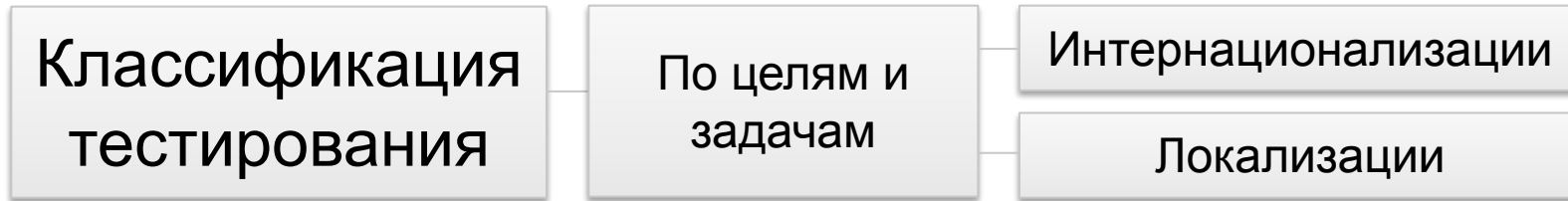
Удобства
использования

Тестирование удобства использования

(usability testing) – тестирование, направленное на исследование того, насколько конечному пользователю понятно, как работать с продуктом, а также на то, насколько ему нравится использовать продукт.



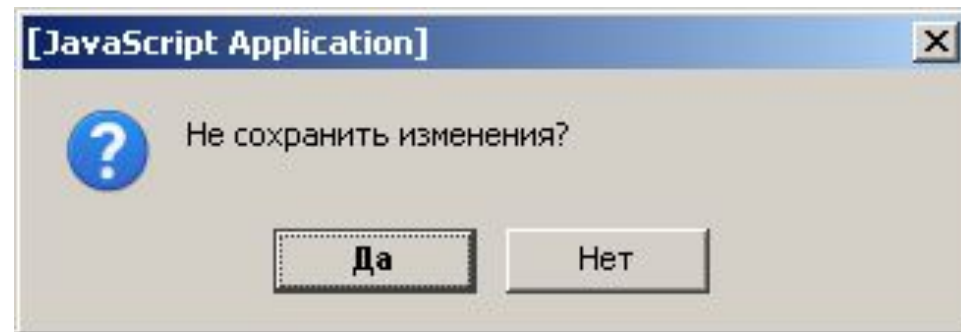
Виды тестирования



Тестирование интернационализации (internationalization testing, i18n testing) – проверка готовности продукта к работе с использованием различных языков и с учётом различных национальных и культурных особенностей.



Тестирование локализации (localization testing, l10n) – проверка корректности и качества адаптации продукта к использованию на том или ином языке с учётом национальных и культурных особенностей.

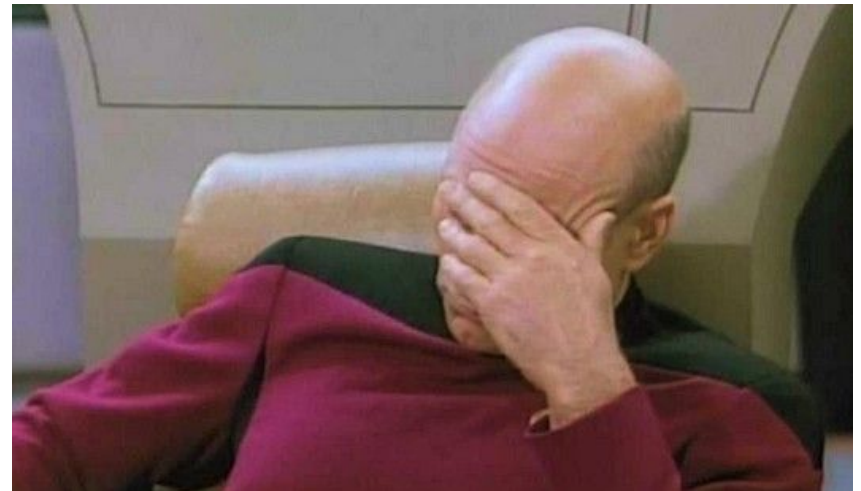


Виды тестирования

И ещё немного о тестировании локализации – суровая правда жизни с bash.im:

Попалась одна прога. Всё бы ничего, но русификация интерфейса не смогла оставить равнодушной. Итак – лучшее (орфография сохранена):

- *монитор силы (power monitor);*
- *пульт команды (command console);*
- *штепсельная вилка и игра (PnP);*
- *предметы Хелпера Браузера (???);*
- *смеситель (mixer);*
- *волна внутри (wave in);*
- *волна вне (wave out);*
- *сердечник ODBC (ODBC core);*
- *водители ODBC (ODBC drivers);*
- *резьбы (threads);*
- *побегите история (run history);*
- *освежите (refresh);*
- *о (about).*



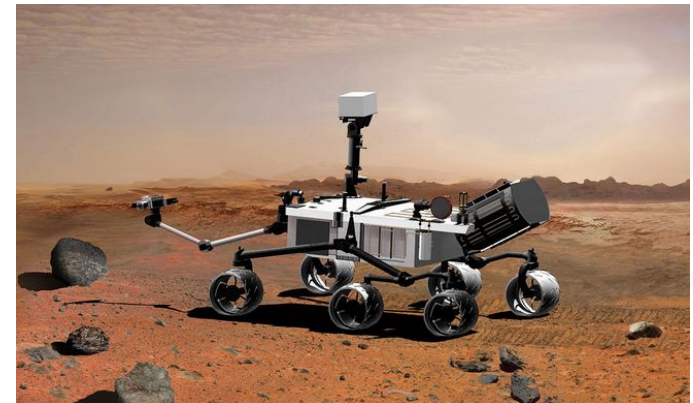
Виды тестирования

Классификация
тестирования

По целям и
задачам

Исследовательское

Исследовательское тестирование (exploratory testing) – частично формализованный подход, в рамках которого тестировщик выполняет работу с приложением по выбранному сценарию, который в свою очередь дорабатывается в процессе выполнения с целью более полного исследования приложения.



Критерии качества

Критерии качества

Критерии качества, приёмочные критерии (acceptance criteria) – любые объективные показатели качества, которым разрабатываемый продукт должен соответствовать с точки зрения заказчика или пользователя, чтобы считаться готовым к эксплуатации.

Пример:

Приёмочные критерии: успешное прохождение 100% тест-кейсов уровня дымового тестирования и 90% тест-кейсов уровня критического пути (см. метрику «Успешное прохождение тест-кейсов») при условии устранения 100% дефектов критической и высокой важности (см. метрику «Общее устранение дефектов»). Итоговое покрытие требований тест-кейсами (см. метрику «Покрытие требований тест-кейсами») должно составлять не менее 80%.

**Типичные ошибки
тестирования и универсальные
рекомендации**

Типичные ошибки тестирования

- Мысль о том, что вам удастся найти все дефекты.
- Мысль о том, что только команда тестировщиков в ответе за качество продукта.
- Мысль о том, что цель тестирования – это только поиск дефектов.
- Игнорирование сложных проблем, концентрация на мелочах.
- Игнорирование проблем удобства использования, локализации и т.д.
- Отсутствие должного внимания тестированию документации и инсталляционному тестированию.

Типичные ошибки тестирования

- Недостаток внимания проектированию тестов в угоду их выполнению.
- Написание некачественных отчётов о дефектах.
- Попытка автоматизировать всё подряд.
- Априорное принятие того факта, что программа работает корректно.
- Нежелание фиксировать некоторые ошибки (в т.ч. под влиянием разработчиков).
- Создание конфликтных ситуаций с разработчиками.

Универсальные рекомендации

- Тестирование должно начинаться с постановки целей.
- Хороший тест-кейс – это тест-кейс, который обнаруживает ошибку.
- Важно знать, когда прекращать тестирование.
- Неэффективно тестировать то, что вы создали сами.

- Концентрируйтесь на негативном тестировании, добивайтесь возникновения ошибки, выполняйте действия, которые нарушают требования.
- Будьте скептически настроены по отношению к любой функциональности.
- Привлекайте конечных пользователей.
- Особое внимание уделяйте функциям, которые будет использовать большинство пользователей.