



PerformanceLab

Обеспечение производительности ИТ

Функциональное тестирование

www.performance-lab.ru

v6
26.11.2011

Темы:

1. Введение (основные понятия)
2. Тестирование в жизненном цикле ПО
3. Виды и типы тестирования
4. Тестирование условий
5. Тестирование документации
6. Функциональное тестирование
7. Артефакты тестирования
8. Что такое тестовые требования
9. Что такое тестовые сценарии
10. Что такое тестовые наборы
11. Дефект: виды дефектов, описание дефекта, жизненный цикл дефекта
12. Баг трекинговые системы
13. Автоматизированное тестирование

Тестирование – Процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов.

Тестирование - процесс *выполнения* программы с целью обнаружения ошибок

Качество ПО – степень соответствия присущих характеристик требованиям

Качество ПО - способность программы делать то, что ждет от нее пользователь

Надежность - Способность программного продукта функционировать при заданных условиях на протяжении определенного периода времени, или для определенного количества операций

Тестируемость - Способность программного продукта предоставлять возможность для тестирования внесенных изменений.

Среда тестирования - инфраструктура для подготовки проведения тестирования и интерпретации результатов. Как правило, это копия продуктива.

Тестовое покрытие - часть системы, тестируемая данным набором проверок.

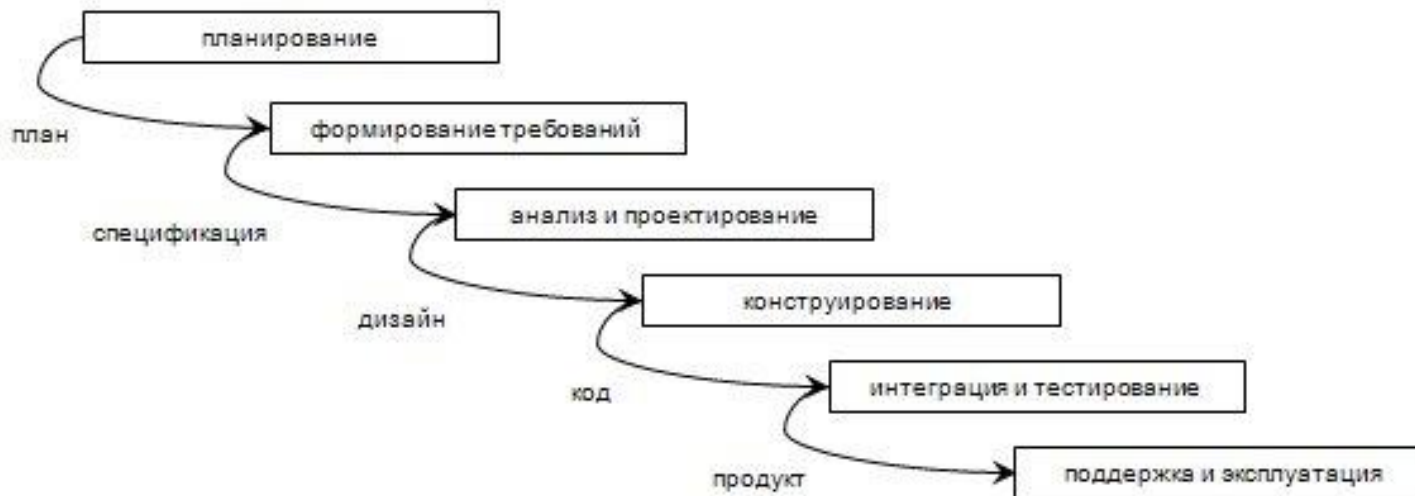
Тесткейс - Набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнение определенного пути программы или же для проверки соответствия определенному требованию

Позитивный тест – проверка на правильных данных, реакция на корректные действия.

Негативный тест – проверка на неправильных данных, реакция на некорректные действия

Ошибка - различие между полученным и описанным или ожидаемым результатом (поведением) системы

Предложена в 1970 г. Уинстоном Ройсом.



Преимущества:

Полная и согласованная документация на каждом этапе;
Легко определить сроки и затраты на проект.

Недостатки:

В водопадной модели переход от одной фазы проекта к другой предполагает полную корректность результата (выхода) предыдущей

Спиральная модель

Разработана в середине 1980-х годов **Барри Боэмом**

Преимущества:

Детальное управление рисками

Недостатки:

определение момента перехода на следующий этап

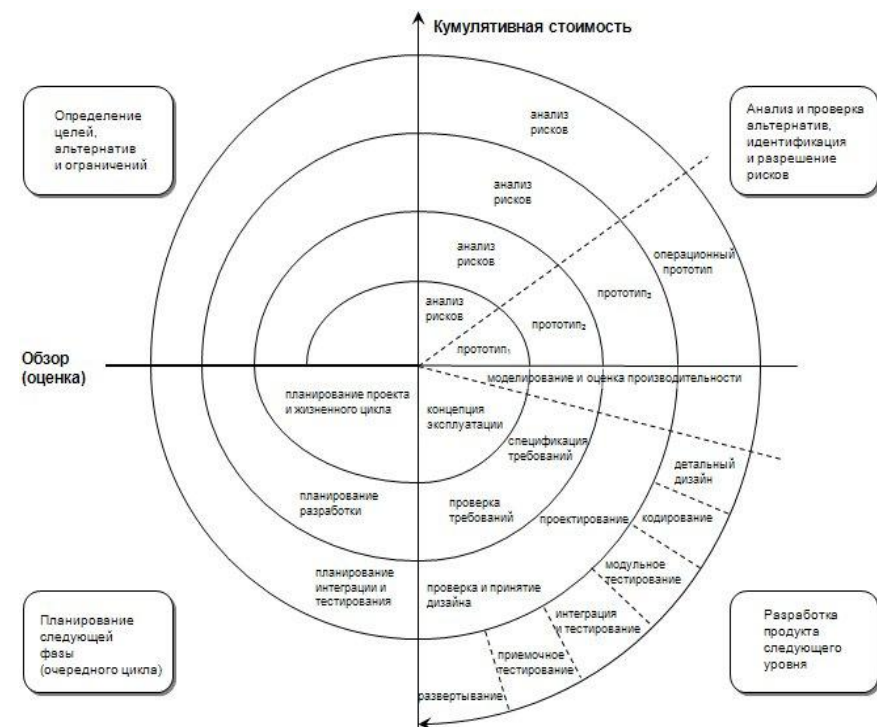
На каждой итерации оцениваются:

риск превышения сроков и стоимости проекта;

необходимость выполнения ещё одной итерации;

степень полноты и точности понимания требований к системе;

целесообразность прекращения проекта.



Последовательность итераций, каждая из которых напоминает «мини-проект»



Преимущества: 1-я итерация

2-я итерация

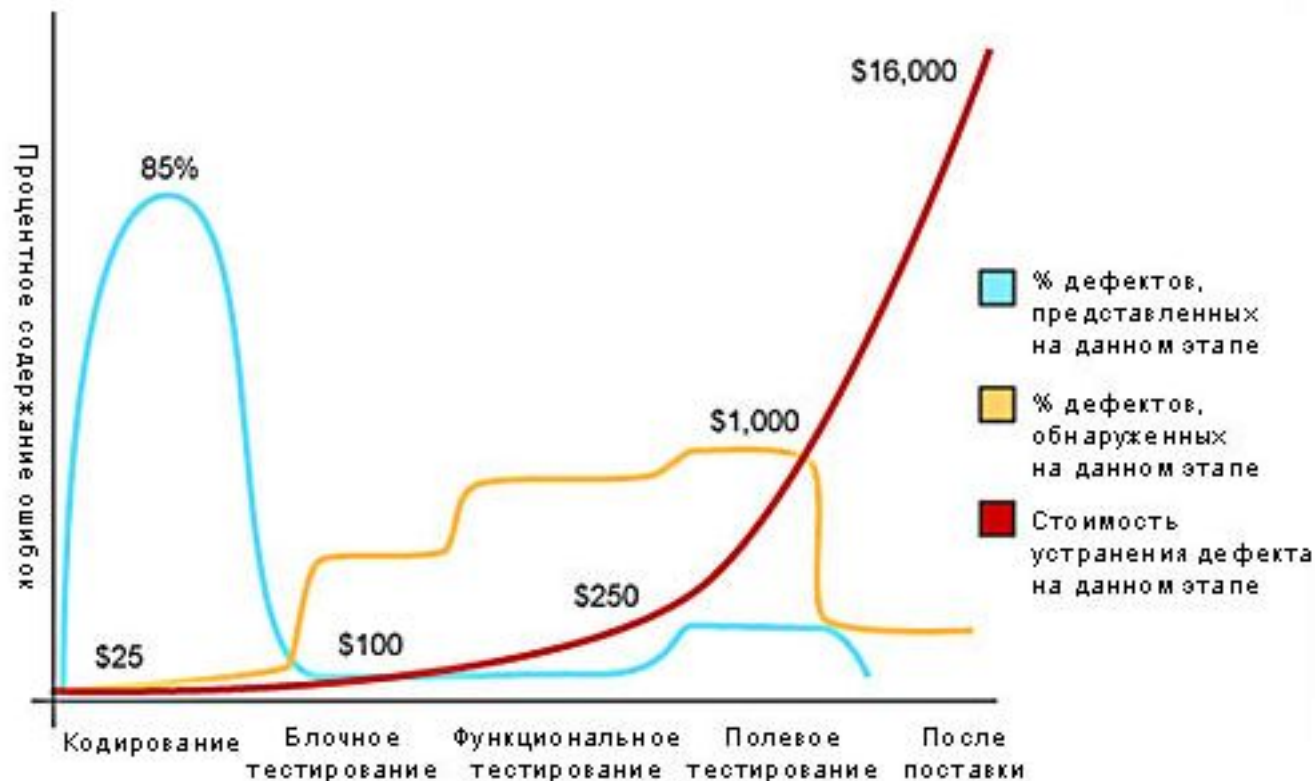
3-я итерация

- Проще контролировать отдельные шаги
- Легко определить сроки и затраты на отдельный шаг

Недостатки:

- Нет целостного понимания готовности проекта
- Трудно определить затраты на весь проект
- Увеличение объёма тестирования
- Проектирование под “черновик”

Стоимость ошибок при разработке ПО



Чем позже найдена ошибка, тем дороже стоит ее исправление.

Статическое тестирование - Тестирование осуществляется путем анализа артефактов (документации, кода).

Анализ может производиться как вручную, так и с помощью специальных инструментальных средств. Целью анализа является раннее выявление ошибок и потенциальных проблем в продукте

Динамическое тестирование - Тестирование осуществляется путем запуска программного кода, компоненты или системы в целом.

По степени автоматизированности:

- Ручное тестирование (manual testing)
- Автоматизированное тестирование (automated testing)
- Полуавтоматизированное тестирование (semiautomated testing)

По степени изолированности компонентов:

- Компонентное (модульное) тестирование (component/unit testing)
- Интеграционное тестирование (integration testing)
- Системное тестирование (system/end-to-end testing)

По времени проведения тестирования:

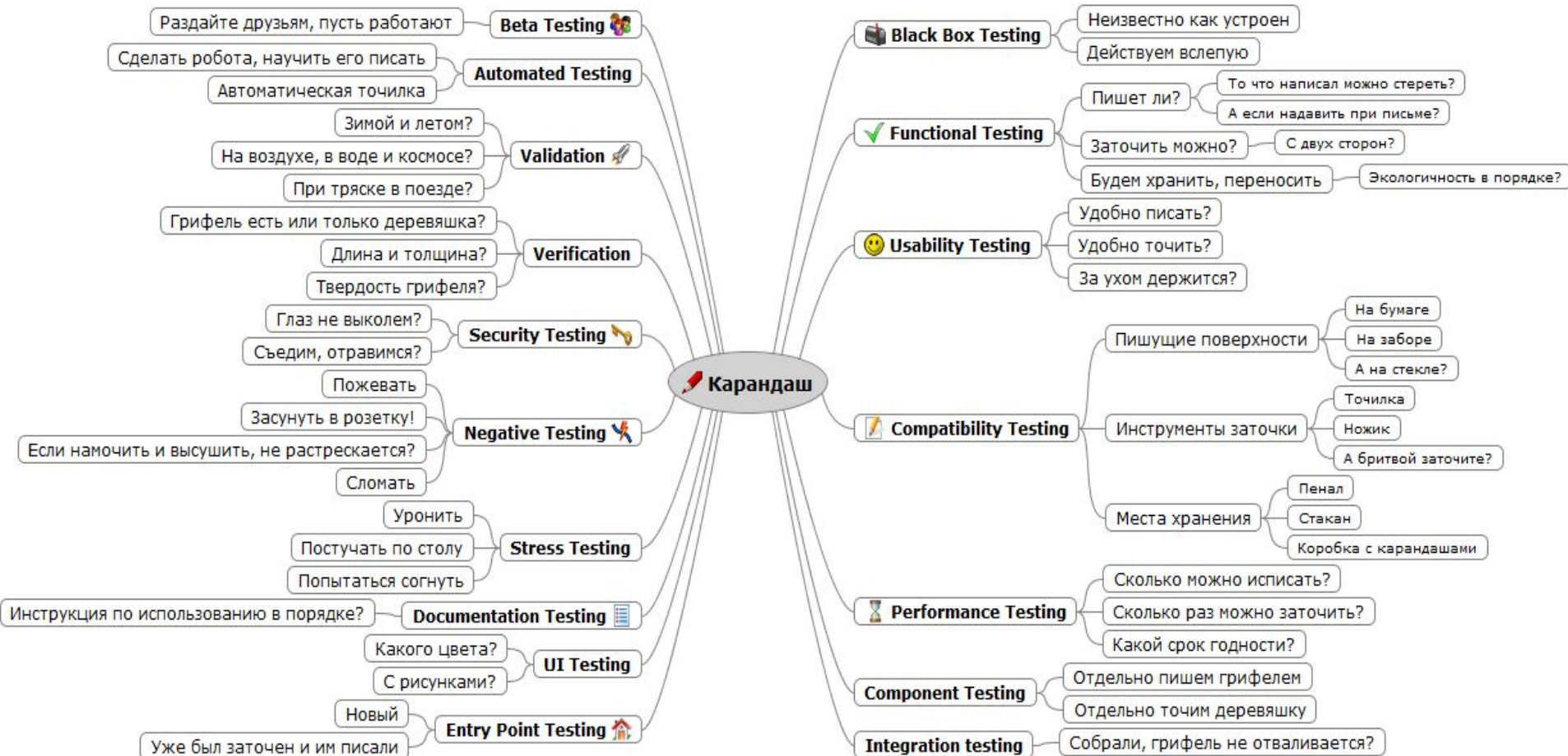
- Альфа тестирование (alpha testing)
- Тестирование при приёме (smoke testing)
- Тестирование новых функциональностей (new feature testing)
- Регрессионное тестирование (regression testing)
- Тестирование при сдаче (acceptance testing)
- Бета тестирование (beta testing)

По признаку позитивности сценариев:

- Позитивное тестирование (positive testing)
- Негативное тестирование (negative testing)

По степени подготовленности к тестированию:

- Тестирование по документации (formal testing)
- Эд Хок (интуитивное) тестирование (ad hoc testing)



Один из способов проверки устойчивости системы на значениях, близких к предельным, - создавать для каждого входа как минимум три тестовых примера:

- Значение внутри диапазона
- Минимальное значение
- Максимальное значение

Для большей уверенности в работоспособности системы используют пять (семь) тестовых примеров:

- Значение внутри диапазона
- Минимальное значение
- Минимальное значение -1
- Максимальное значение
- Максимальное значение $+1$

Такой способ проверки называется проверкой на граничных значениях. Он позволяет выявлять проблемы, связанные с выходом за границы диапазона.

Целью тестирования документации является

- Выявление доступной документации описывающей систему
- Получение выявленной документации
- Выяснение степени актуальности полученной документации
- Анализ документации позволяет выявить возможные ошибки еще на стадии проектирования системы
- Использование документации для поиска примеров тестовых сценариев



Функциональное тестирование – тестирование, основанное на анализе спецификации функциональности компонента или системы.

Методы функционального тестирования:

- **Белый ящик** - тестирование на основе анализа структуры кода;
- **Черный ящик** - поведенческое тестирование, глобальное представление, основанное на спецификации;
- **Серый ящик** - промежуточный уровень, на котором определен интерфейс между компонентам



Тестирование методом «Белого ящика»

Тестирование методом "белого ящика" предполагает обработку системы как "прозрачного объекта" и позволяет заглянуть внутрь, фокусируя внимание на использовании знаний о конкретном программном обеспечении для правильного подбора тестовых данных.

Цель этого вида тестирования в том, чтобы проверить каждую ветвь кода, каждый путь, каждый оператор, **проверить сам код.**



Тестирование методом «Черного ящика»

Тестирование методом "Черного ящика" предполагает обработку системы как "непрозрачного объекта", таким образом знание внутренней структуры в явном виде не используется. Тестирование этим методом обычно подразумевает проверку функциональных возможностей. При тестировании программного обеспечения методом "Черного ящика" тестировщик знает только набор вводимых параметров и ожидаемые на выходе результаты, каким образом программа достигает этих результатов ему неизвестно.



Преимущество - не требует знания языков программирования.

Цель - проверить расхождение поведения программы с документацией.

BR, BRD, CR, ЗНИ – требования к доработке системы

Realize Notes – полный перечень доработок и исправленных ошибок в системе – Очень важный документ

T3 – описание реализации дорабатываемого функционала

Руководство пользователя – описывает как с этим работать

Руководство администратора – описывает как это установить



Методика тестирования – описывает виды и стратегию тестирования, ограничения предъявляемые к тестированию конфигурацию тестовых сред порядок проверки, критерии начала и окончания тестирования и др.

План тестирования – используется как дополнение к методике делая акцент на тестируемом функционале.

Тестовые наборы – наборы тестовых сценариев по которым осуществляется тестирование, являются приложениями к методике/плану тестирования

Описание тестового стенда – внутренний артефакт группы. Описывает пароли логины доступы ссылки к системам. Должен быть всегда актуален.

Регламенты – описание процедур установленных внутри группы или компании

Список может изменяться и зависит от договоренностей с заказчиком.

Отчет о тестировании – содержит ответы на вопросы заданные в методике тестирования. А также главный вопрос готово ли ПО к эксплуатации

Артефакты описанные ниже являются также приложением к отчету.

- Список тестовых сценариев с указанием их статуса – пройден/ не пройден и причины.
- Список открытых на момент завершения тестирования дефектов
- Список всех дефектов найденных в процессе тестирования



011
012
013
014

План тестирования

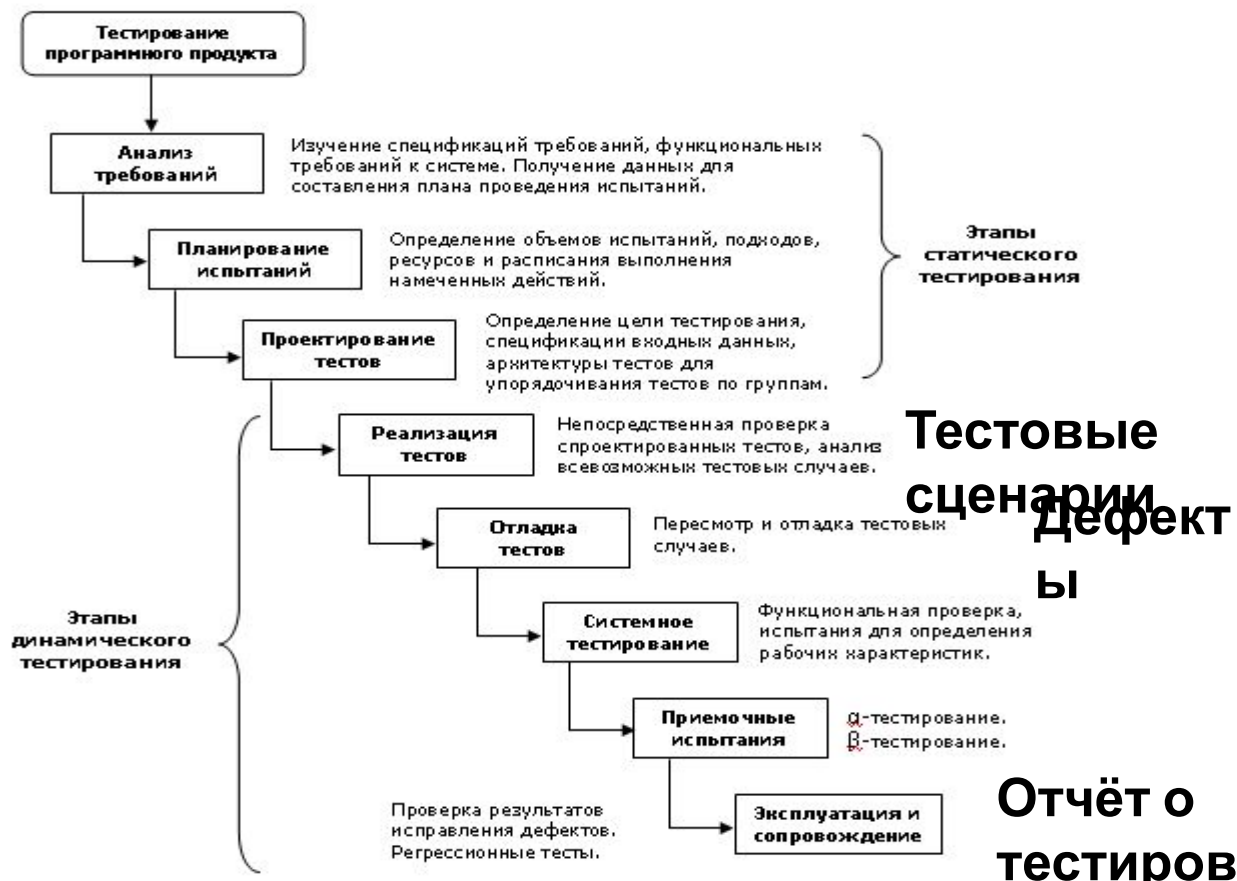


Схема 1. Каскадный процесс тестирования программных продуктов.



Тестовые требования

Требование – это формализованное описание свойств системы.

Виды требований:

- Бизнес-требования
- Функциональные требования
- Нефункциональные требования
- Тестовые требования

Тестовое требование – это формализованное описание свойств системы, которые необходимо протестировать.

Критерии к тестовым требованиям:

- корректность;
- недвусмысленность;
- полнота;
- непротиворечивость;
- упорядоченность по важности и стабильности;
- проверяемость (верифицируемость или тестопригодность);
- модифицируемость;
- трассируемость;
- понятность.



Тестовые сценарии

Тестовый сценарий – это последовательность **ДЕЙСТВИЙ** для достижения фактического результата.

Состав тестового сценария

- Название
- Предварительные условия
- Необходимые действия
- Ожидаемый результат
- Полученные результаты
- Отметка о прохождении

Сценарий должен быть прост и понятен всем: неопытному тестировщику, программисту, бизнес-пользователю.

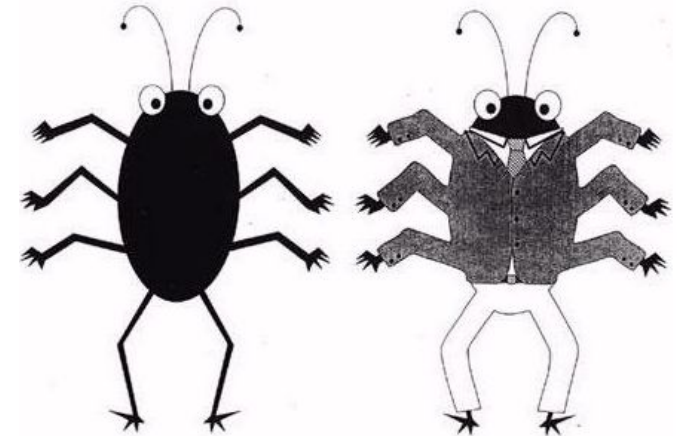
Тестовый набор – набор тестов, которые принадлежат к одной функциональности.

№1	Генеральный договор (входящее). ЖД	MANUAL	✓ Passed	07.06.2007	11:43:21
№1	Персональный договор. ЖД	MANUAL	✓ Passed	07.06.2007	11:43:24
№1	Пролонгированный персональный договор. ЖД	MANUAL	✓ Passed	07.06.2007	11:43:32
№1	Дополнительный к персональному. ЖД	MANUAL	✓ Passed	07.06.2007	11:43:36
№1	Персональный в рамках генерального. Журнал договоров.	MANUAL	✓ Passed	07.06.2007	11:43:38
№1	Пролонгированный в рамках генерального. ЖД	MANUAL	✓ Passed	07.06.2007	11:43:51
№1	Дополнительный в рамках генерального. ЖД	MANUAL	✓ Passed	07.06.2007	11:43:55
№1	Дополнительный (нулевой) к персональному. ЖД	MANUAL	✓ Passed	07.06.2007	11:43:58
№1	Валютный персональный без курса. ЖД	MANUAL	N/A	07.06.2007	11:43:17
№1	Расторжение персонального. ЖД	MANUAL	✓ Passed	07.06.2007	11:44:05
№1	Расторжение (частичное + стороно) персонального. ЖД	MANUAL	✓ Passed	07.06.2007	11:44:13
№1	Расторжение в 2 этапа персонального договора. ЖД	MANUAL	✓ Passed	07.06.2007	11:44:09
№1	Попадение КВ в журнал договоров.	MANUAL	N/A	07.06.2007	11:43:01
№1	Удаление строки журнала договоров в открытом периоде. ЖД	MANUAL	✓ Passed	07.06.2007	11:44:15
№1	Удаление строки журнала договоров в закрытом периоде. ЖД	MANUAL	✓ Passed	07.06.2007	11:44:19



Минимальные данные о дефекте:

- краткое описание
- дата
- автор (обнаруживший дефект)
- ссылка на систему и ее версию (build)
- приоритет (Priority)
- серьезность проблемы (Severity)
- описание (предусловия, шаги для воспроизведения, ожидаемый результат, действительный результат)
- состояние, статус



Классификация по типу:

- ошибки в функциональности
- ошибки эргономики модуля или бизнес-процесса
- ошибки документирования
- ошибки производительности
- ошибки локализации
- ошибки совместимости
- ошибки безопасности

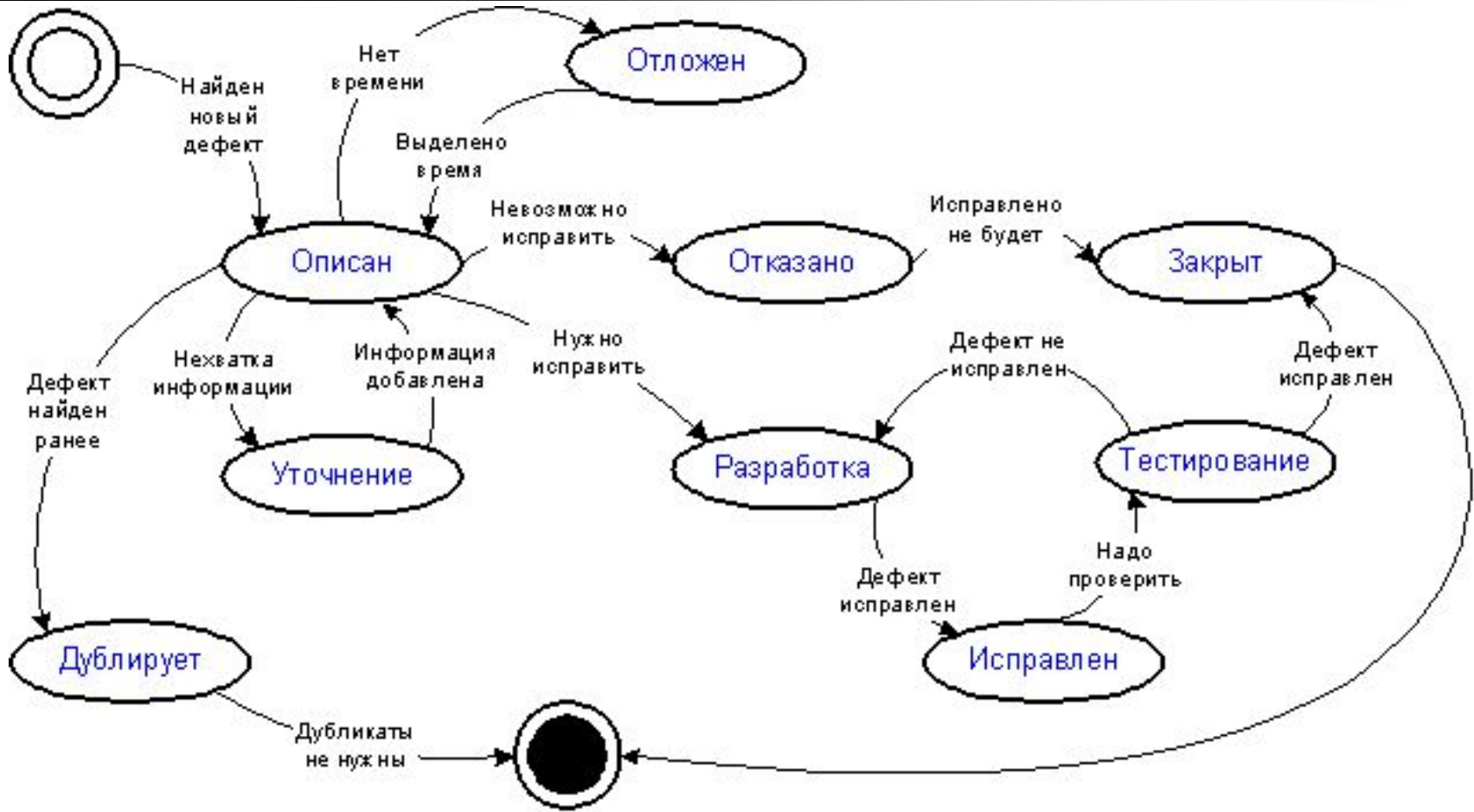
Степени критичности:

- Максимальная критичность
- Высокая критичность
- Нормальная критичность
- Низкая критичность



011
012
013
014

Жизненный цикл дефекта



Система трекинга багов — это инфраструктура, позволяющая:

- создавать,
- хранить,
- просматривать и
- модифицировать информацию

Каждый баг, занесенный в СТБ, представляет собой виртуальную учетную карточку



Книги рекомендуемые к прочтению:

1. **Андреас Голзер, Алаин Захм** "Оптимизация качества для достижения высоких бизнес-результатов"
2. **Бейзер Б.** "Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем"
3. **Элфрид Дастин, Джефф Рэшка, Джон Пол** "Автоматизированное тестирование программного обеспечения"

Сайты рекомендуемые к изучению

<http://software-testing.ru/>

