Тестирование, оценка программного обеспечения

Я, **Борисик Марина Михайловна**
преподаватель БГУИР

Связаться со мной можно:
+375172938824
kafipie@tut.by

Что такое тестирование?

Тестирование программного обеспечения (Testing) – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.

Зачем нужно тестирование?

- **Общая концепция:**

  Конечной целью тестирования является предоставление пользователю качественного программного обеспечения .

- **Конкретизация (с практической точки зрения):**

  Обнаружить дефекты до того, как их найдут конечные пользователи, и предоставить эту информацию команде разработки.

- **+ проконтролировать исправление дефектов!**

Качество

Качество (Quality) – степень, с которой компонент, система или процесс соответствует зафиксированным требованиям и/или ожиданиям и нуждам пользователя или заказчика.

Любое несоответствие → **дефект!**

Дефект

Дефект (баг bug, ошибка) – ключевой термин тестирования, означающий отклонение фактического результата от ожидаемого.

Для обнаружения дефекта необходимо выполнить **три условия**:

1.Знать ***ожидаемый результат*.**

2.Знать **фактический результат.**

3.Зафиксировать **факт разницы между фактическим и ожидаемым результатом.**

Тестирование = поиск дефектов

Источником ожидаемого результата является **спецификация** – детальное описание того, как должно работать ПО. Однако сама спецификация может содержать дефекты: в этом случае следует уточнить требования!

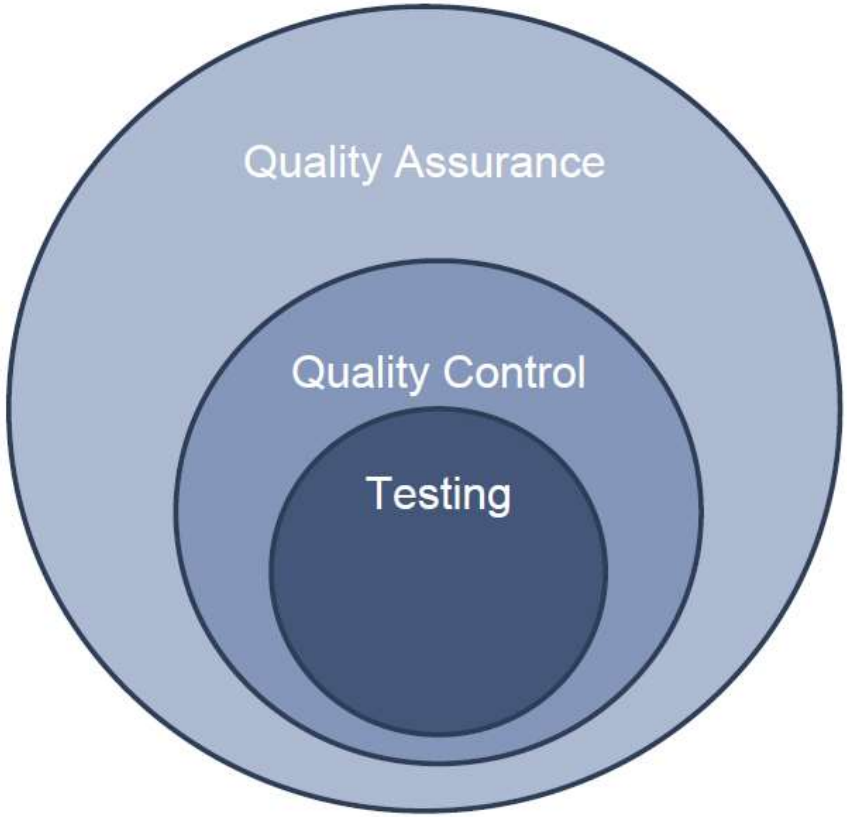
Источником фактического результата являются **тесты** – специальные, искусственно созданные ситуации, выбранные определенным образом.

Тогда тестирование — это проверка соответствия программного обеспечения спецификации (требованиям) на конечном наборе тестов.

Тестирование (Testing) – поиск и контроль устранения дефектов после разработки ПО.

Контроль качества (Quality Control) – оценка уровня качества программного обеспечения на различных этапах разработки и степени готовности ПО

Обеспечение качества (Quality Assurance) ПО – постановка и улучшение процессов разработки и тестирования ПО таким образом, чтобы дефекты не возникали. Выполняется на протяжении всего жизненного цикла ПО



Когда проводят тестирование?

Основные сложности тестирования

1.«Замыливание» взгляда от билда к билду.

2.Необходимость корректно и понятно донести до разработчика найденные дефекты и проконтролировать их исполнение.

3.Высокая ответственность (после тестировщика – только конечные пользователи).

4.Тестирование выполняется в конце проекта, поэтому как правило проходит в очень сжатые сроки в связи с необходимостью сдачи проекта в срок.

СПАСИБО!

Есть вопросы?

Тестирование как проект в проекте

Тестирование программного продукта как проект в проекте включает следующие стадии:

- **1. Планирование тестирования (QA-план).**
- **2.Разработка тестов.**
- **3.Исполнение тестирования (выполнение тестов).**

QA-план

- Планирование тестирования (Test Planning) – работа по составлению и поддержанию актуальности плана тестирования .

- План тестирования (Test Plan, QA Plan) – документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей, а также любые риски, требующие планирования на случай чрезвычайных обстоятельств.

- QA Менеджер составляет тест-план проекта и поддерживает его в актуальном состоянии.

QA-план

- 1.Цель
- 2.Области, подвергаемые тестированию
- 3.Области, не подвергаемые тестированию
- 4.Тестовая стратегия и подходы (*в т.ч. виды тестов и их композиция)*
- 5.Тестовое окружение
- 6.Критерии (*начала, приостановки, возобновления, завершения)*
- 7.Ресурсы (*программные, аппаратные, человеческие, временные, финансовые)*
- 8.Календарный план работ
- 9.Роли и ответственность
- 10.Оценка рисков
- 11.Документация (*шаблоны отчетов, периодичность, получатели)*
- 12.Метрики (*числовые характеристики показателей качества)*

Классификация видов тестирования (по покрытию):

- **Smoke Test:** поверхностное тестирование для определения пригодности сборки для дальнейшего тестирования;

уровень качества: Acceptable / Unacceptable.

- **Minimal Acceptance Test (MAT, Positive Test):** тестирование системы или ее части только на корректных данных/сценариях;

уровень качества: High / Medium / Low.

Классификация видов тестирования (по покрытию):

- **Acceptance Test (AT):** полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях;

уровень качества: High / Medium / Low

Классификация видов тестирования (по покрытию):

Тест на этом уровне покрывает все возможные сценарии тестирования:

проверку работоспособности модулей при вводе корректных значений;

проверку при вводе некорректных значений;

использование форматов данных отличных от тех, которые указаны в требованиях;

проверку исключительных ситуаций, сообщений об ошибках;

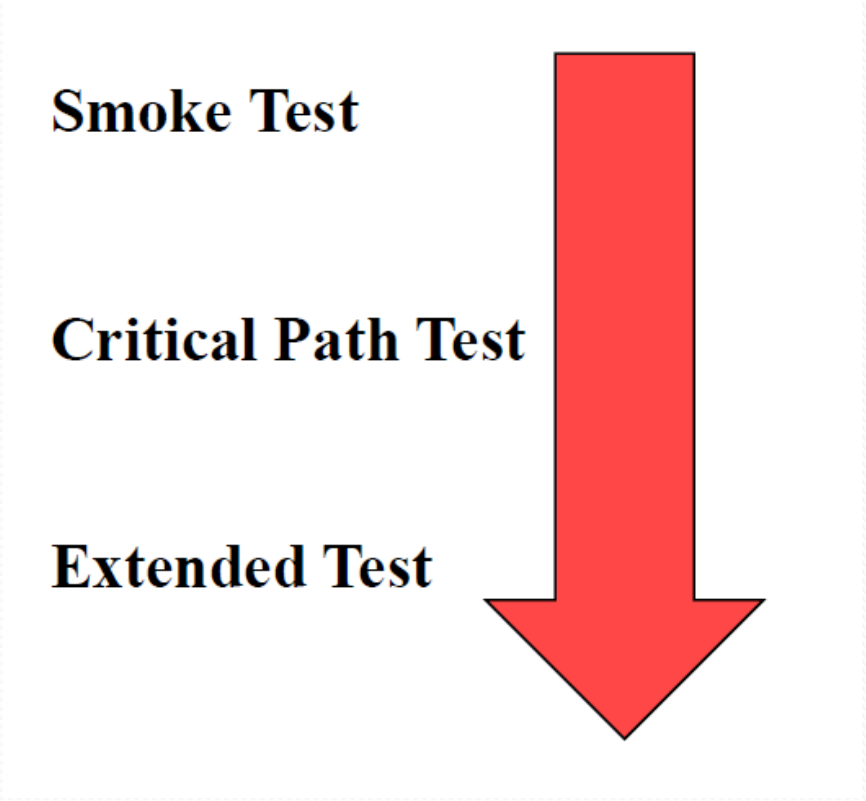
тестирование на различных комбинациях входных параметров; проверку всех классов эквивалентности;

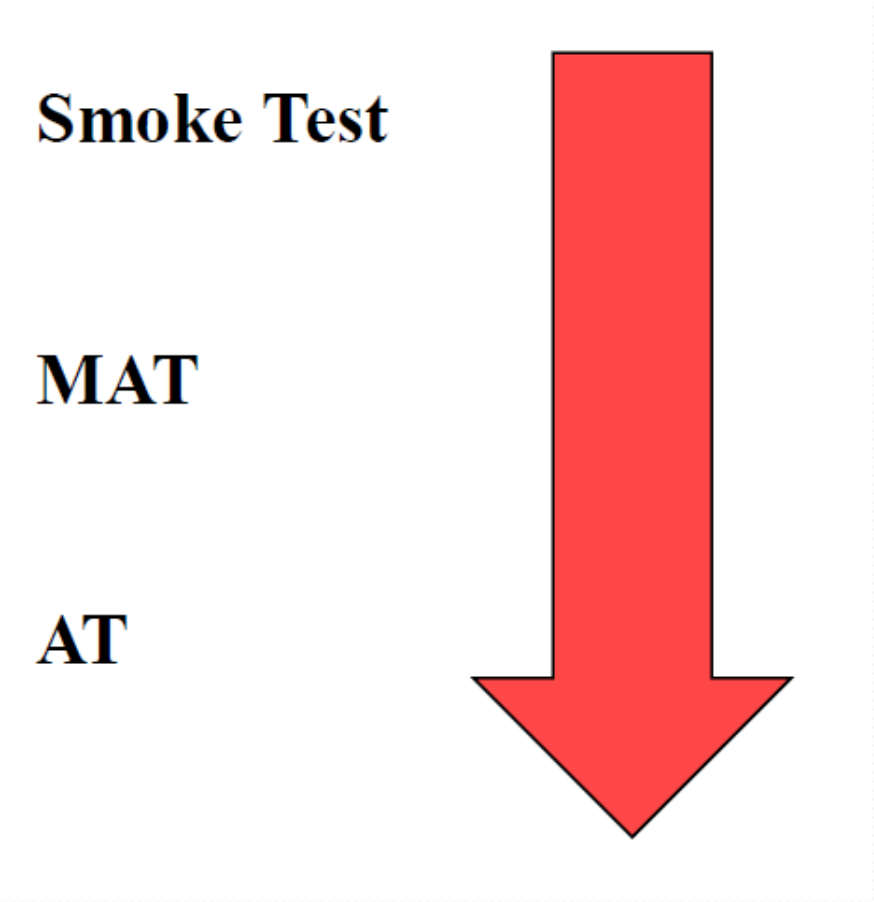
тестирование граничных значений интервалов;

сценарии не предусмотренные спецификацией

и т.д.

Классификация видов тестирования по покрытию (по глубине):





Базовая структура описания дефекта

- 1.Headline = Заголовок
- 2.Severity = Степень критичности (серьезности) дефекта
- 3.Description = Описание
- 4.Result = Фактический результат
- 5.Expected Result = Ожидаемое (корректное) поведение
- 6.Attachment = приложение (скриншоты)

H e a d l i n e –как написать?

Где? Что? Когда? (при каких условиях)

Пример:

Где?: Диалог "Преобразовать данные"

Что?: показывается сообщение об ошибке

Когда?: при нажатии кнопки "Преобразовать"

Характеристики headline:

- краткость (удобство чтения & отброшены все ненужные слова)
- информативность
- точная идентификация проблемы

+ особенности теста, если это может помочь разбору дефекта (версия операционной системы, браузеры)

Severity

Правила выставления критичности

Серьезность	Описание
Critical (критический)	Функциональная ошибка, которая блокирует работу части функционала или всего приложения; критичная с точки зрения реализации базовых задач пользователя или бизнеса заказчика.
Major (значительный)	Функциональная ошибка, которая нарушает нормальную работу приложения, но не блокирует работу части функционала в целом.
Average (средней значимости)	Не очень важная функциональная ошибка. Критичные дефекты GUI.
Minor (незначительный)	Редко встречающиеся незначительные функциональные дефекты. Дефекты GUI.
Enhancement (предложение по улучшению)	Функциональные предложения, советы по улучшению дизайна (оформления), навигации и др. Такой дефект является необязательным для исправления.

Critical (критический)

- Функции приложения недоступны или заблокированы.
- Основная задача приложения не может быть выполнена.
- Дефект существенно влияет на работу пользователя.
- Функция работает с серьезными нарушениями функциональных требований.
- Нарушение нормальной работы серьезно влияет на бизнес-цели заказчика, наносит урон его репутации, или нарушает положение о конфиденциальности информации.

Примеры:

- *Приложение невозможно установить.*
- *Приложение после обновления не работает.*
- *В результате возникновения дефекта произошла потеря данных.*
- *Механизм лицензирования продукта не работает.*
- *Все, что касается денег!!!*

Major (серьезный)

- Нормальная работа части приложения нарушена.
- Дефект вызвал серьезные функциональные или другие проблемы.
- Функция выполняется некорректно, и это влияет на работу пользователя.

Примеры:

- *Ввод определенных значений в поле блокирует функцию.*
- *Сохраняется только часть данных, и это влияет на работу других модулей.*
- *Система игнорирует некоторые данные, и это влияет на основные функции.*

Average (средний)

- Наличие дефекта не вызывает значительного ухудшения производительности, функционирования или удобства использования.
- Функция работает частично или не во всех случаях, но это серьезно не влияет на работу пользователя.
- Дефекты удобства использования, которые серьезно влияют на работу пользователя.

Примеры:

- *Проверка правильности ввода данных не проводится или выдает неверные результаты либо тип данных, вводимых в поле, не соответствует требованиям.*
- *«Уехал» текст за пределы окна.*
- *Неправильно написано название компании.*
- *Механизм навигации не работает должным образом.*
- *Количество полей на форме не соответствует требованиям.*

Minor (незначительный)

- Функция имеет несущественное значение или вообще не влияет на работу.
- Дефект GUI не влияет в значительной степени на точность или удобство выпускаемой версии.

Примеры:

- *Опечатки, нечеткие формулировки или ограниченная область видимости сообщений об ошибках.*
- *Названия полей, форм, таблиц и окон отличаются от указанных в сценарии.*
- *Длина поля не ограничена или не соответствует требованиям.*
- *Не выдаются подтверждения, сообщения об ошибках или информационные сообщения.*
- *Не все файлы или другие установленные проектом объекты удаляются после удаления приложения.*

Enhancement (рекомендации)

- Рекомендации по улучшению GUI.
- Предложения по незначительному улучшению функциональности.

Примеры:

- *Добавить кнопку «Наверх» на длинной форме.*

Description+Result

Стандартная структура:

1. Step 1:
2. Step 2:
3. …

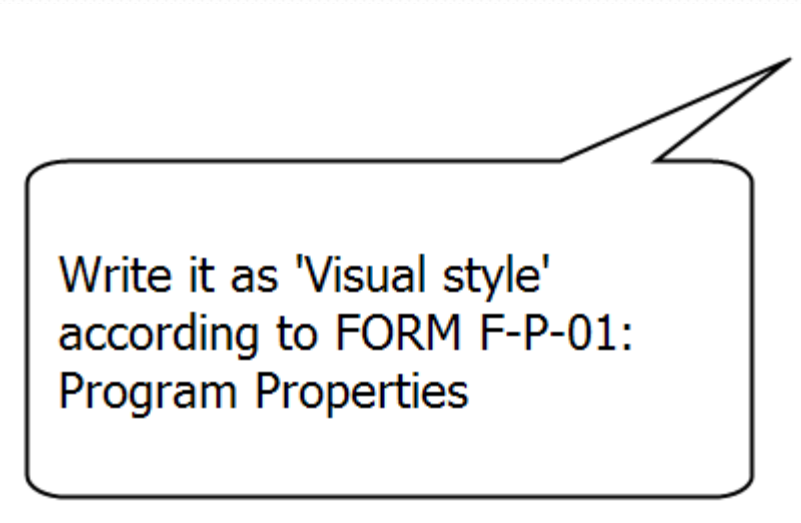
Result:

Используем аттачменты, в случаях где они могут ускорить понимание дефекта.

Обязательно для GUI дефектов (!)

ExpectedResult

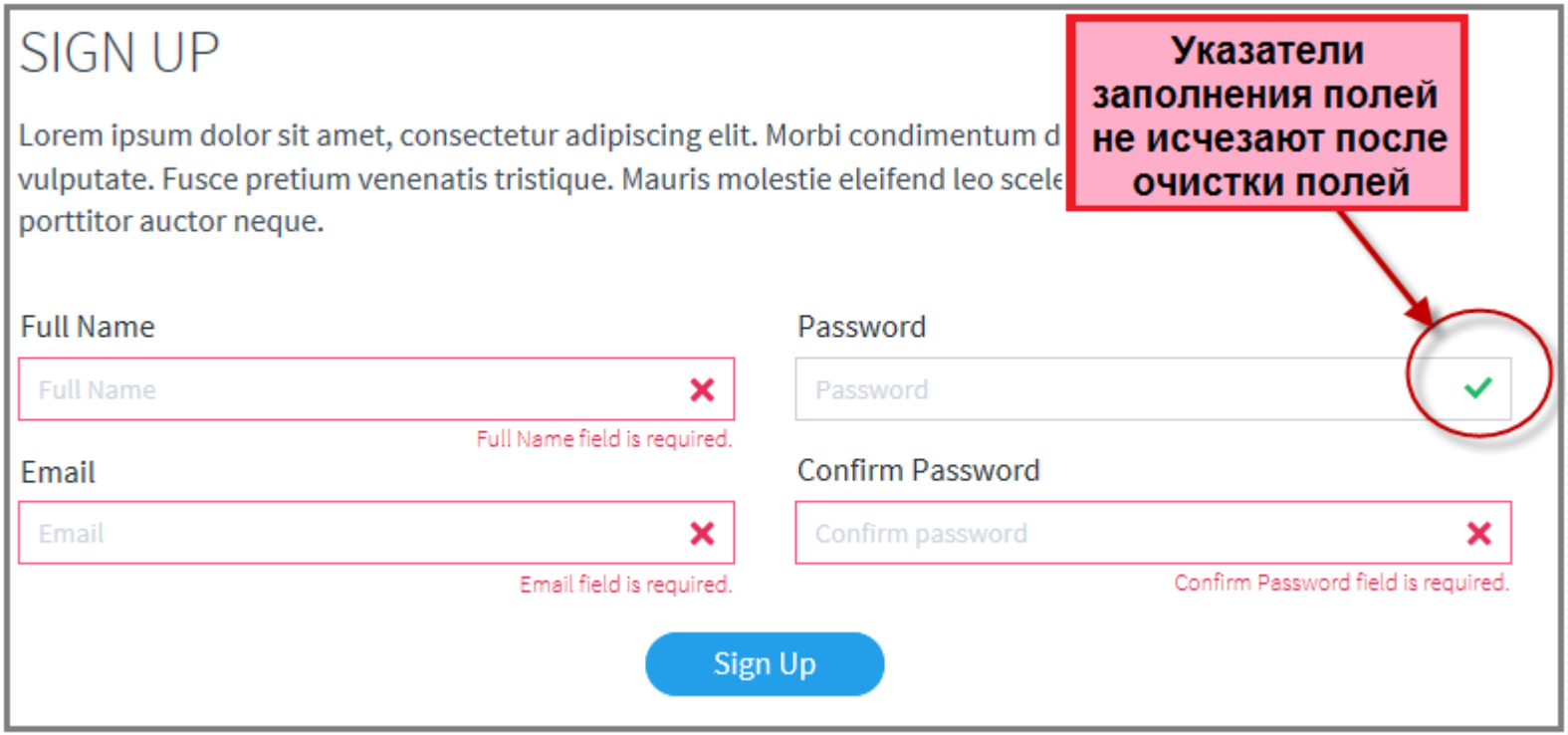
- ✓ Указание: что ожидается
- ✓ Аргументация



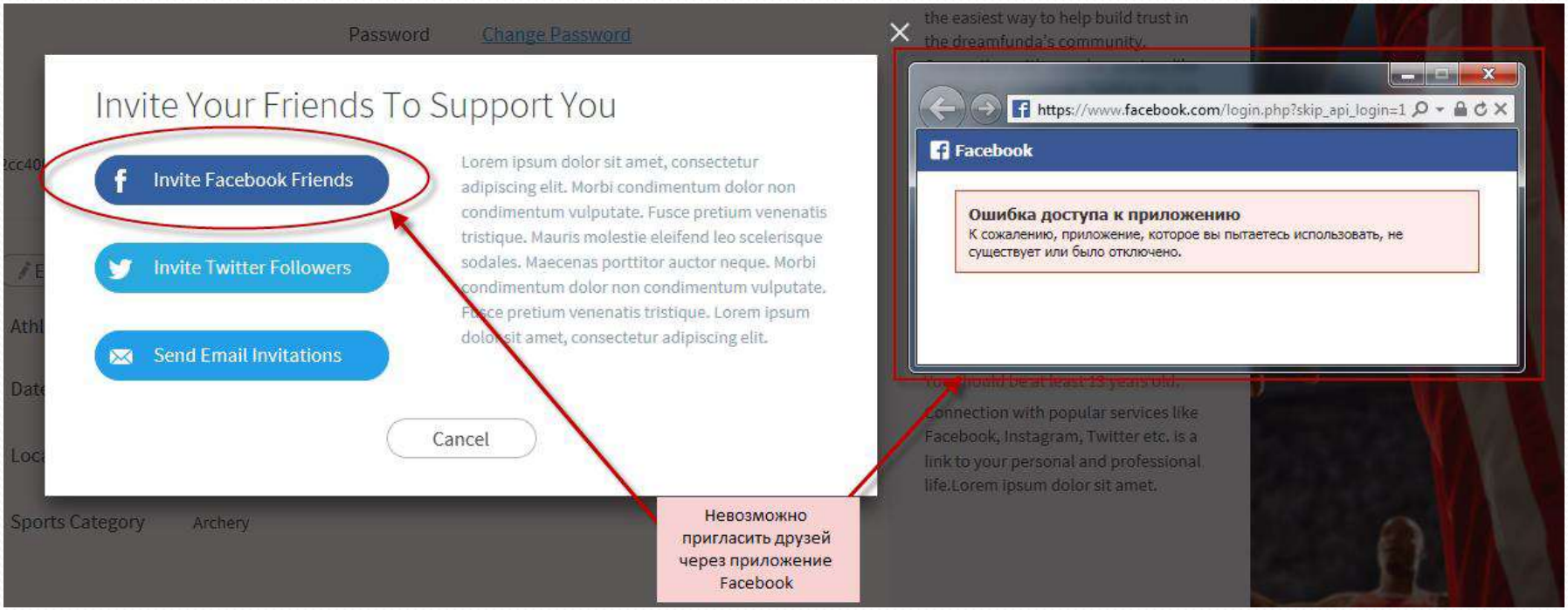
Attachment

- !!! Необходимо давать ссылку на соответствующее требование, к нарушению которого приводит фактический результат работы программного средства.
- Если это ускорит исправление дефекта – дополнительно еще и процитируйте это требование (например, *каким именно должно быть значение по умолчанию).*

Attachment


SIGN UP
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi condimentum d
vulputate. Fusce pretium venenatis tristique. Mauris molestie eleifend leo scele
porttitor auctor neque.
Указатели заполнения полей не исчезают после очистки полей
Full Name
Full Name
Full Name field is required.
Password
Password
Email
Email
Email field is required.
Confirm Password
Confirm password
Confirm Password field is required.
Sign Up

Attachment


Password
Change Password
Invite Your Friends To Support You
Invite Facebook Friends
Invite Twitter Followers
Send Email Invitations
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi condimentum dolor non condimentum vulputate. Fusce pretium venenatis tristique. Mauris molestie eleifend leo scelerisque sodales. Maecenas porttitor auctor neque. Morbi condimentum dolor non condimentum vulputate. Fusce pretium venenatis tristique. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Cancel
Sports Category
Archery
the easiest way to help build trust in the dreamfunda's community.
https://www.facebook.com/login.php?skip_api_login=1
Facebook
Ошибка доступа к приложению
К сожалению, приложение, которое вы пытаетесь использовать, не существует или было отключено.
Connection with popular services like Facebook, Instagram, Twitter etc. is a link to your personal and professional life.Lorem ipsum dolor sit amet.
Невозможно пригласить друзей через приложение Facebook

Рекомендации по хорошему описанию дефектов

- Чтобы внести дефект, его следует воспроизвести минимум два раза, и только после гарантированного повторения описать последовательность действий.
- Создавайте дефект и описывайте его сразу же, как только вы обнаружили ошибку. Откладывание «на потом» приводит к тому, что вы или вообще забудете об этом дефекте, или забудете о каких-то важных деталях.
- Описывайте дефект понятно. Используйте общеупотребимую лексику, точные названия программного средства.

Рекомендации по хорошему описанию дефектов

- Нельзя не описывать дефекты только потому, что их не получается воспроизвести. Факт невозможности выяснить причину дефекта в таком случае обязательно должен быть указан в описании дефекта.
- Дефект не должен содержать фразу: «Это не работает», дефект должен показать, что и при каких условиях не работает.
- Четко указывайте ожидаемый результат. При необходимости цитируйте спецификацию, а не просто ссылайтесь на нее.

Рекомендации по хорошему описанию дефектов

- Если существует какая-либо информация, которая поможет быстрее обнаружить или исправить дефект, - сообщите эту информацию.
- Описание дефекта – это технический документ, в котором нет места эмоциям.
- После описания дефекта еще раз перечитайте его. Убедитесь, что все необходимые поля заполнены, и все написано верно.

Рекомендации по хорошему описанию дефектов

Группируйте **однотипные дефекты:**

- GUI дефекты могут группироваться в один по признаку формы, на которой они находятся, т.е. если одна форма содержит несколько GUI дефектов, их можно объединить в один.
- Функциональные дефекты могут группироваться по модулям, страницам или полям, на которых они воспроизводятся (например, динамическое обновление не работает в модулях 1, 2 и 4 или отсутствует валидация на спецсимволы на всех полях страницы).

Рекомендации по хорошему описанию дефектов

Нельзя:

- Недопустимо объединять в один дефекты разного типа, например, функциональные и GUI.
- Группировка функциональных дефектов по признаку формы, на которой они найдены, не применяется.

Что дальше?

- Помимо собственно описания дефектов результаты тестирования вносят в рабочую тестовую документацию (Acceptance Sheet, Test Survey, Test Cases).
- Для этого напротив выполненной проверки указывают степень критичности обнаруженного дефекта, его номер и заголовок.
- Если по результатам конкретной проверки выявлено несколько дефектов, то перечисляют номера всех дефектов, а в качестве степени критичности и заголовка указывают наиболее серьезный дефект

Вспоминаем про Severity…

Severity (критичность, уровень серьезности/важности дефекта) – параметр оценки, который показывает, насколько дефект влияет на нормальную работу.

Выставляется тестировщиком на основании субъективной оценки и внутренних стандартов компании.

А еще есть Priority

Priority (приоритет, уровень срочности дефекта) – параметр оценки, который определяет очередность исправления дефектов разработчиком. Выставляется менеджером проекта в соответствии со срочностью исправления.

Priority - приоритетность дефекта для разработчика

Приоритет	Описание
Blocker	Дефект чрезвычайно важен для пользователей и должен быть исправлен сразу же, как только это возможно.
Critical	Дефект достаточно важен и нарушает нормальную работу приложения. Этот дефект должен быть исправлен после исправления дефектов с приоритетом “ Blocker”.
Major	Значение приоритета по умолчанию. Этот дефект должен быть исправлен после исправления дефектов с более высокими приоритетами.
Minor	Дефект не имеет большой значимости для пользователей. Его следует исправлять после исправления дефектов с более высокими приоритетами.
Trivial	Дефект имеет минимальную значимость. Его следует исправлять в самую последнюю очередь.

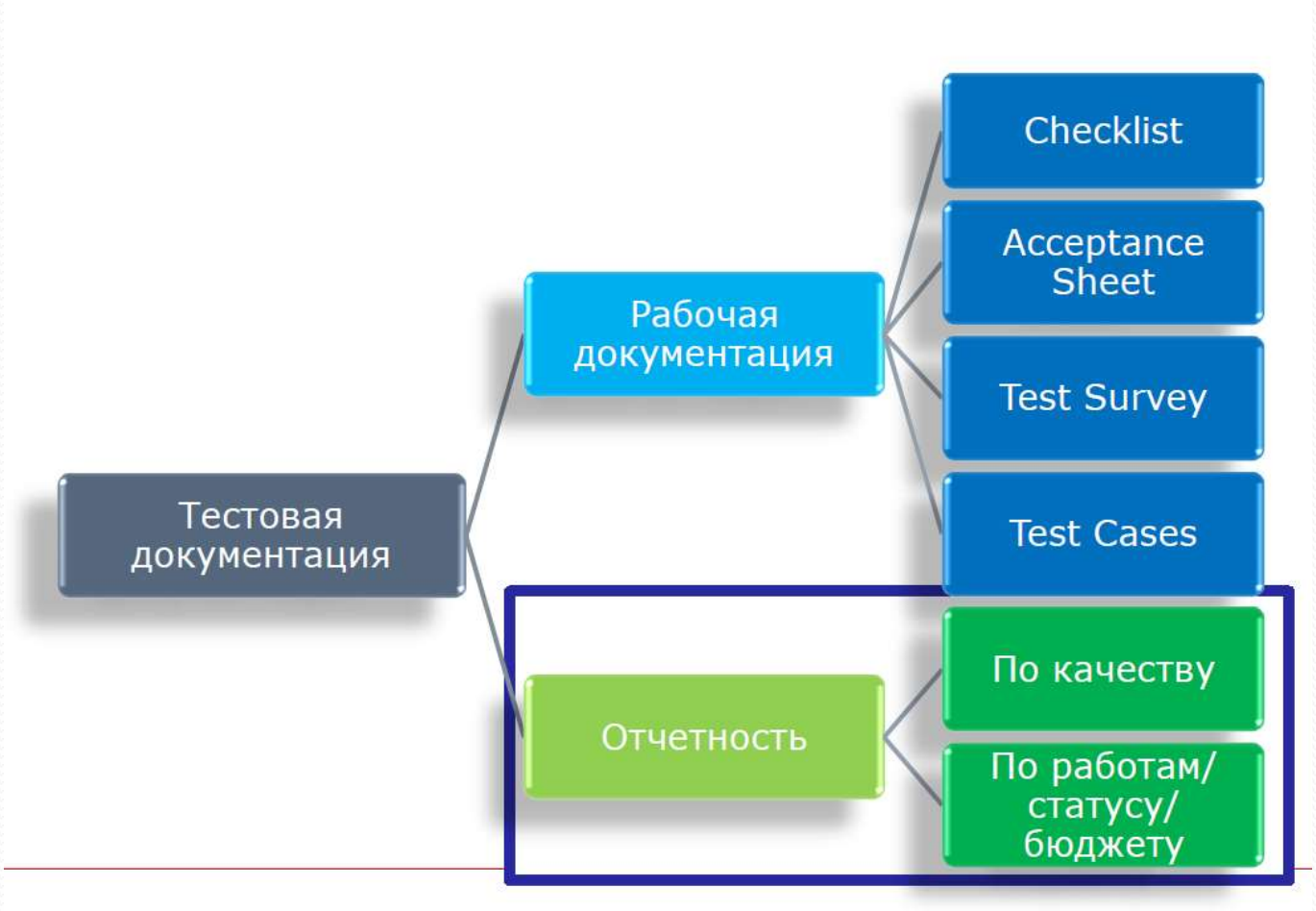
Priority - приоритетность дефекта для разработчика

Какая связь между Severity и Priority? НЕТ НИКАКОЙ СВЯЗИ!

Разработчик смотрит на Priority,

тестировщик – на Severity.

Виды тестовой документации



Итоговый отчет о результатах тестирования: что входит в отчет?

Общая информация: название проекта, номер сборки + *перечисление протестированных модулей (если тестировался не весь проект).*

- 1. Кто/когда тестировал.
- 2.Тестовое окружение.
- 3.Общая субъективная и/или объективная оценка качества приложения.
- 4.Обоснование выставленного качества: аргументы.
- 5.Детализированный анализ качества: графики.
- 6.Детализированный анализ качества по модулям.
- 7.ТОП-5 самых критичных дефектов.
- 8.Рекомендации.

GENERAL INFORMATION

Project name: …

Version: …

QA team

Company: ...

QA Manager: ...

QA Engineers: …

Test Date: …

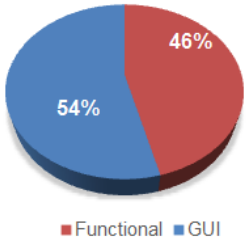
GENERAL QUALITY ANALYSIS

Текст обоснования выствленной оценки качества

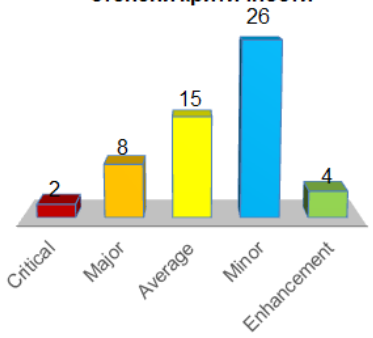
Smoke Test	Acceptable
Acceptance Test	Low
New defects	55
Test Documentation	Acceptance Sheet

DETAILED QUALITY ANALYSIS

Распределение дефектов по типу



Распределение дефектов по степени критичности



Top 5 most important issues

ID	Severity
IN-1	Critical
IN-19	Critical
IN-4	Major
IN-31	Major
IN-32	Major

Testing environment

URL: http://...

Application service: Apache 2.2.2.0

Browsers: Google Chrome

Recommendations

Ваши реккомендации

QUALITY BY TEST MODULES

Module	Quality	Comments
Главная страница	Medium	Аргументация выставленного качества для данного модуля
Каталог товаров	Low	Аргументация выставленного качества для данного модуля
Корзина	Low	Аргументация выставленного качества для данного модуля

Распределение дефектов по модулям

Ценные указания по составлению отчета

Аналитика

- Мало написать, что в приложении есть проблемы – нужно описать к чему они приводят, как влияют на работу пользователя.
- Если нужно показать, какие дефекты должны быть в первую очередь исправлены, то обязательно аргументировать почему должно быть именно так.
- И так в каждой части отчёта должен быть анализ, а не простое перечисление.

Ценные указания по составлению отчета

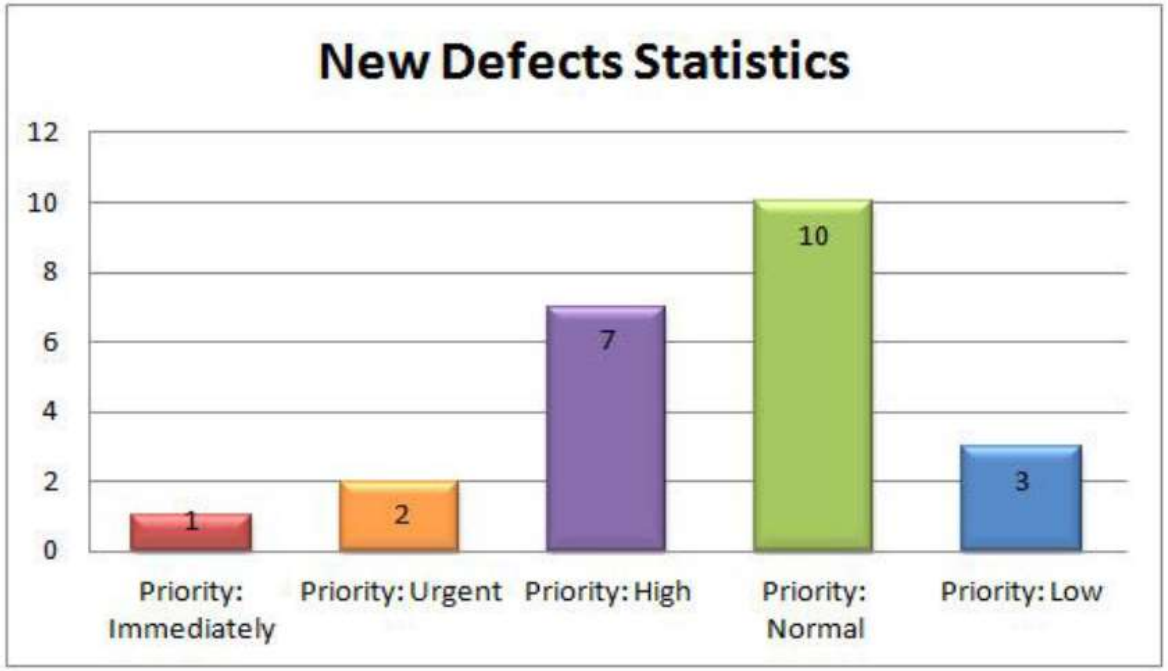
Представление информации

- Если приложение можно разделить на логические модули (блоки), то в отчёте должна присутствовать детализированная информация по качеству каждого, обычно в виде таблицы: название модуля-качество-комментарии.

Ценные указания по составлению отчета

Диаграммы и графики

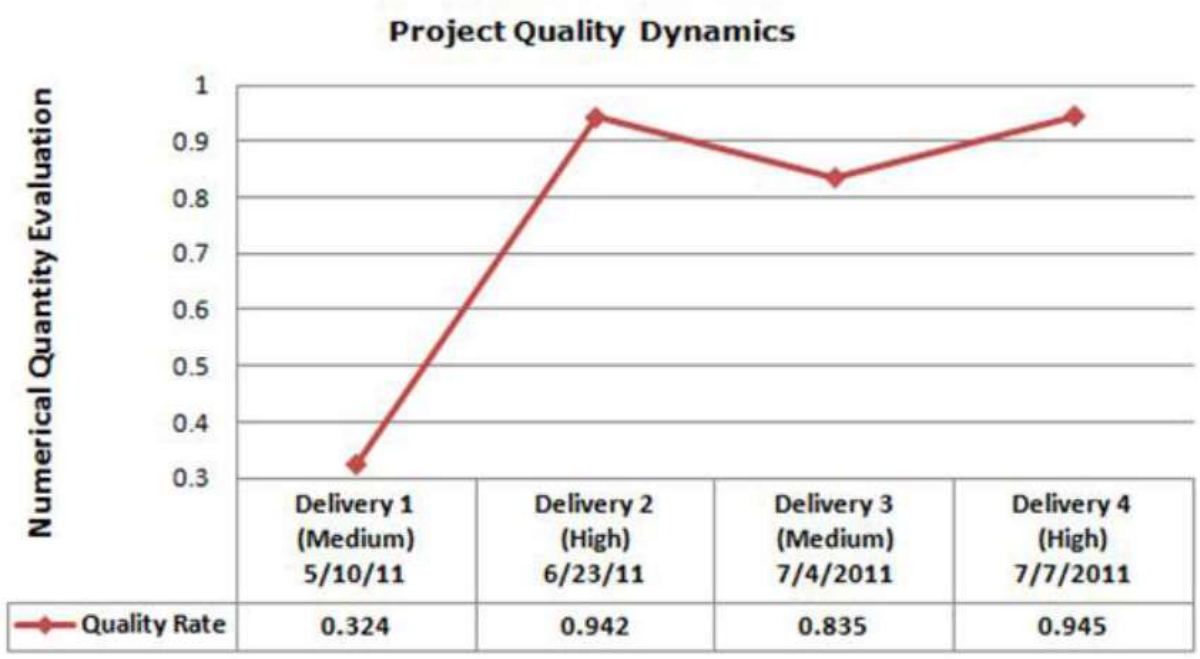
Столбчатые диаграммы лучше подойдут там, где важно количество дефектов:



Ценные указания по составлению отчета

Диаграммы и графики

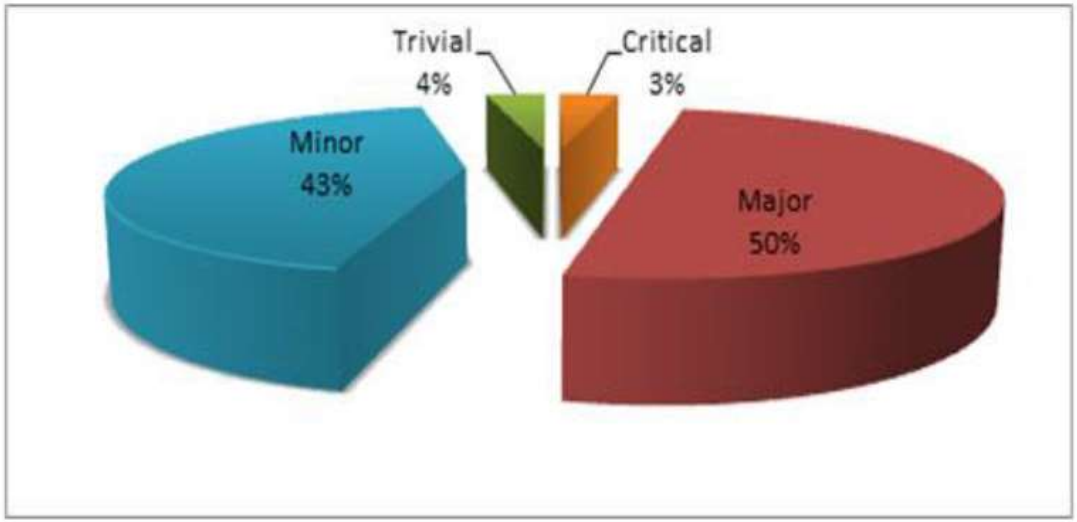
Показать динамику качества в финальном отчёте по всем сборкам лучше линейным графиком



Ценные указания по составлению отчета

Диаграммы и графики

Там, где нужно продемонстрировать процентное соотношение – лучше воспринимаются круговые диаграммы



Ценные указания по составлению отчета