

# Занятие №4

## Артефакты тестирования

### Жизненный цикл тестирования



# План на сегодня

- Вспоминаем материал прошлой лекции
- ЖЦ тестирования ПО
- Артефакты тестирования
- Работа с багтрекинговой системой



# Вспоминаем

- Что такое проект?
- Участники проекта?
- Цикл разработки ПО?
- Модели разработки ПО?
- Методологии разработки ПО?



# Жизненный цикл тестирования



## Software Testing Life cycle



# Артефакты тестирования

- План тестирования (Test Plan)\*
- Варианты использования (Use Cases)\*
- Список проверки (Checklist)\*
- Тестовые сценарии (Test Cases)\*
- Матрица соответствий (Traceability Matrix)
- Отчет об ошибке (Bug Report)\*
- Отчет о тестировании (Test Result Report)
- ...

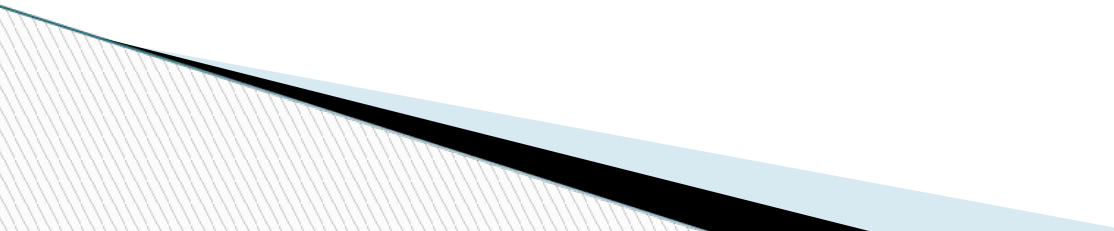


# План тестирования (Test Plan)

- ▣ **Тест план (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.

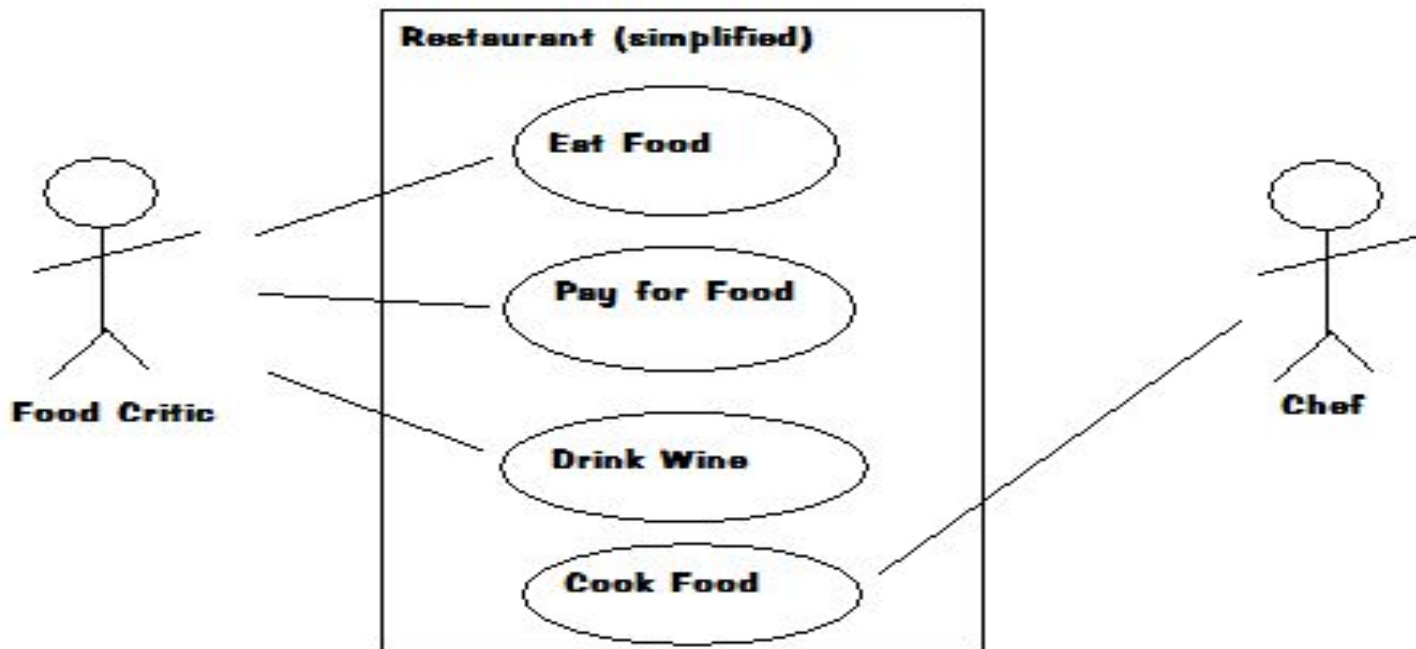


# План тестирования (Test Plan)

- Что надо тестировать?
  - Что будем тестировать?
  - Когда будем тестировать?
  - Критерии начала тестирования
  - Критерии окончания тестирования
  - Окружение тестируемой системы (описание программно-аппаратных средств)
  - Необходимое для тестирования оборудование и программные средства
- 

# Варианты использования (Use Cases)

- Описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды. Система может отвечать на внешние запросы Актора (англ. Actor), может сама выступать инициатором взаимодействия.





# Список проверки (Checklist)

- Чек-лист - это документ, описывающий что должно быть протестировано. При этом чек-лист может быть абсолютно разного уровня детализации.
- Составляющие:
  - Функционал для тестирования
  - Ожидаемый результат (+\ -)
  - Статус



# Тестовые сценарии (Test Cases)

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

<b>Action</b>	<b>Expected Result</b>	<b>Test Result (passed/failed/blocked)</b>
Open page "login"	Login page is opened	Passed

# Виды Тестовых Случаев

- ▣ **Позитивный тест кейс** использует только корректные данные и проверяет, что приложение правильно выполнило вызываемую функцию.
- ▣ **Негативный тест кейс** оперирует как корректными так и некорректными данными (минимум 1 некорректный параметр) и ставит целью проверку исключительных ситуаций (срабатывание валидаторов), а также проверяет, что вызываемая приложением функция не выполняется при срабатывании валидатора.



# Тестовые сценарии (Test Cases)



**ЭТОТ НЕЛОВКИЙ МОМЕНТ  
КОГДА НАПИСАЛ ТЕСТ КЕЙСЫ**



**А ПОТОМ САМ НЕ МОЖЕШЬ В  
НИХ РАЗОБРАТЬСЯ...**

# Typical written Test Case attributes

- Test Case ID
- Test Case Name/Summary
- Test Steps
- Expected Result
- Status (Pass/Fail/Not Tested)
- Author

- Case Priority
- Comments
- Update History
- Related Requirement (s)
- Is Automated



# Отчет об ошибке (Bug Report)

- ❑ Баг или дефект репорт - это документ, описывающий ситуацию или последовательность действий, которые привели к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.



# У БАГОВ ТОЖЕ ЕСТЬ ЧУВСТВА

НАЙДЯ БАГ:  
СООБЩИТЕ О НЕМ

БАГИ НЕ ЛЮБЯТ  
БЫТЬ ЗАБЫТЫМИ



НАЙДЯ БАГ:  
ИЗУЧИТЕ ЕГО

БАГИ ЛЮБЯТ  
БЫТЬ ПОНЯТЫМИ



НАЙДЯ БАГ:  
СДЕЛАЙТЕ СНИМОК

БАГИ ЛЮБЯТ ХРАНИТЬ  
ПАМЯТЬ О ВСТРЕЧЕ



НАЙДЯ БАГ:  
УЗНАЙТЕ ЕГО ДРУЗЕЙ

БАГИ ОБЩИТЕЛЬНЫ



НАЙДЯ БАГ:  
СООБЩИТЕ О НЕМ БЫСТРО

ИНАЧЕ БАГИ ОБЖИВУТСЯ И  
ПОСТРОЯТ СЕБЕ ДОМА



НАЙДЯ БАГ:  
БУДЬТЕ ЧЕСТНЫ

БАГИ НЕ ЛЮБЯТ  
СПЛЕТЕН



НАЙДЯ БАГ:  
ЗАПОМНИТЕ КАК ВЫ  
ВСТРЕТИЛИ ЕГО

БАГИ РОМАНТИЧНЫ



НАЙДЯ БАГ:  
НЕ ИГНОРИРУЙТЕ ЕГО

БАГИ МОГУТ УКУСИТЬ,  
ЕСЛИ ИХ НЕ ЦЕНИТЬ





# Структура баг репорта

- Короткое описание (Summary)
- Детальное описание (Description)
- Проект (Project)
- Компонент приложения (Component)
- Номер версии (Version)
- Серьезность (Severity)
- Приоритет (Priority)
- Статус (Status)
- Автор (Author)
- Назначен на (Assigned To)
- ОС / Сервис Пак / Версия приложения...
- Описание (Steps+Results)
- Прикрепленный файл (Attachment)





# Серьезность и Приоритет дефекта

- ▣ **Серьезность (Severity)** - это атрибут, характеризующий влияние дефекта на работоспособность приложения.
- ▣ **Приоритет (Priority)** - это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Можно сказать, что это инструмент менеджера по планированию работ. Чем выше приоритет, тем быстрее нужно исправить дефект.

Severity  
Vs  
Priority

# Градация Серьезности (Severity) и Приоритета (Priority)

## ▣ Severity:

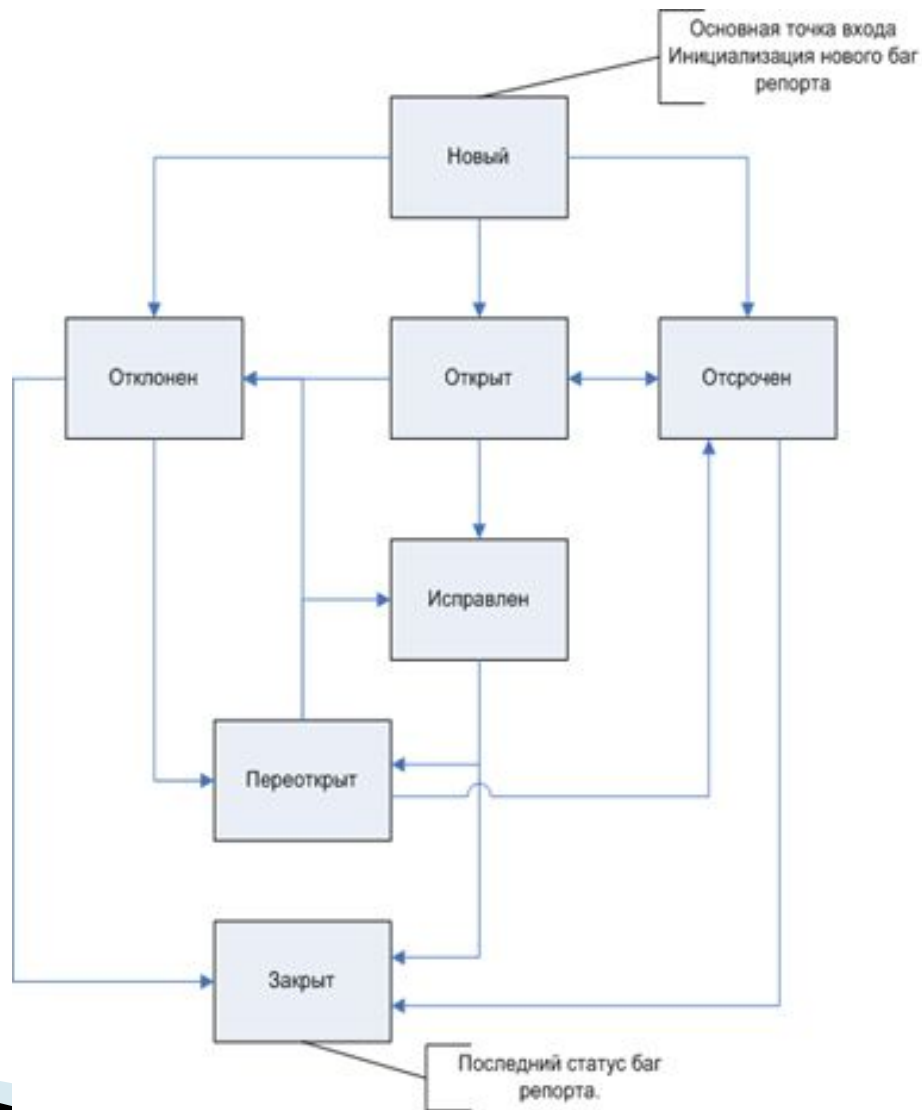
- S1 Блокирующая (Blocker)
- S2 Критическая (Critical)
- S3 Значительная (Major)
- S4 Незначительная (Minor)
- S5 Тривиальная (Trivial)

## ▣ Priority:

- P1 Высокий (High)
- P2 Средний (Medium)
- P3 Низкий (Low)



# Жизненный цикл бага



# Написание баг репортов

- ▣ «Прочитав короткое описание бага (Bug Summary), я должен понять в чем состоит проблема, прочитав детальное описание бага (Bug Description) я должен знать строку кода, которую править.» ©
- ▣ Принцип «Что? Где? Когда?»



# Обязательные поля баг репорта

- Короткое описание (Bug Summary)
- Детальное описание (Description)
- Серьезность (Severity)
- Шаги к воспроизведению (Steps to reproduce)
- Результат (Actual Result)
- Ожидаемый результат (Expected Result)
- Версия приложения (Build found)





**KEEP  
CALM  
IT'S  
FINALLY  
OVER**

# Работа с Mantis Bug Tracker

- <http://qa07.besaba.com/> - адрес трекера
- Логин приходил на почту
- Пароль устанавливался при регистрации
- `magazqa.besaba.com` - имя проекта
- Assign to - Vladislav.Zotke
- **View Status - PRIVATE!!!**
  
- <http://magazqa.besaba.com/> - Тестовый сайт

