



Компания хороших идей

**Bellintegrator**

# Технологический процесс тестирования. Тестовые артефакты

Докладчик: Герасимова Елена

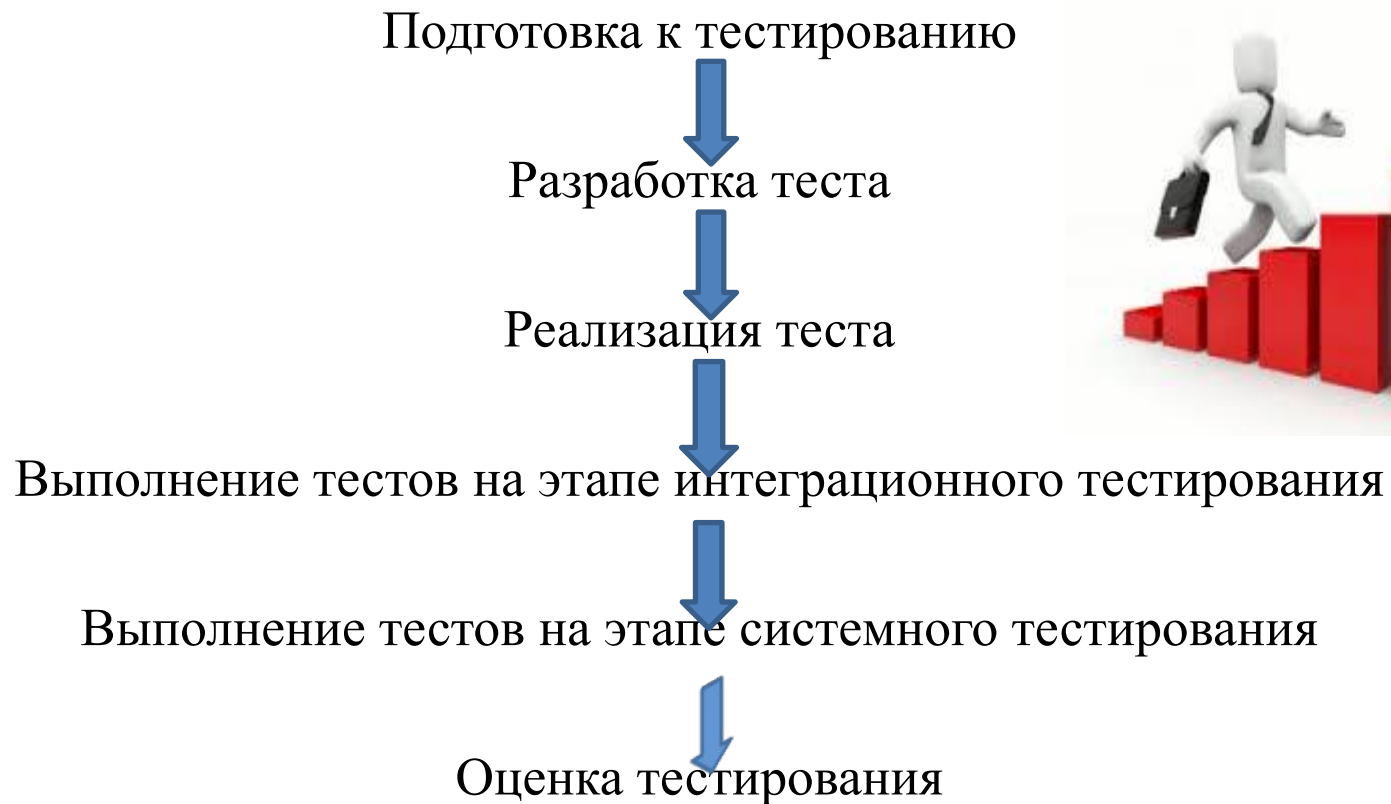
# План занятия

- Условия для проведения тестирования
- Тестовые артефакты
- Тест дизайн
- Дефекты
- Отчет о тестировании

**Технологический процесс** — это упорядоченная последовательность взаимосвязанных действий, выполняющихся с момента возникновения исходных данных до получения требуемого результата.



# Этапы технологического процесса тестирования



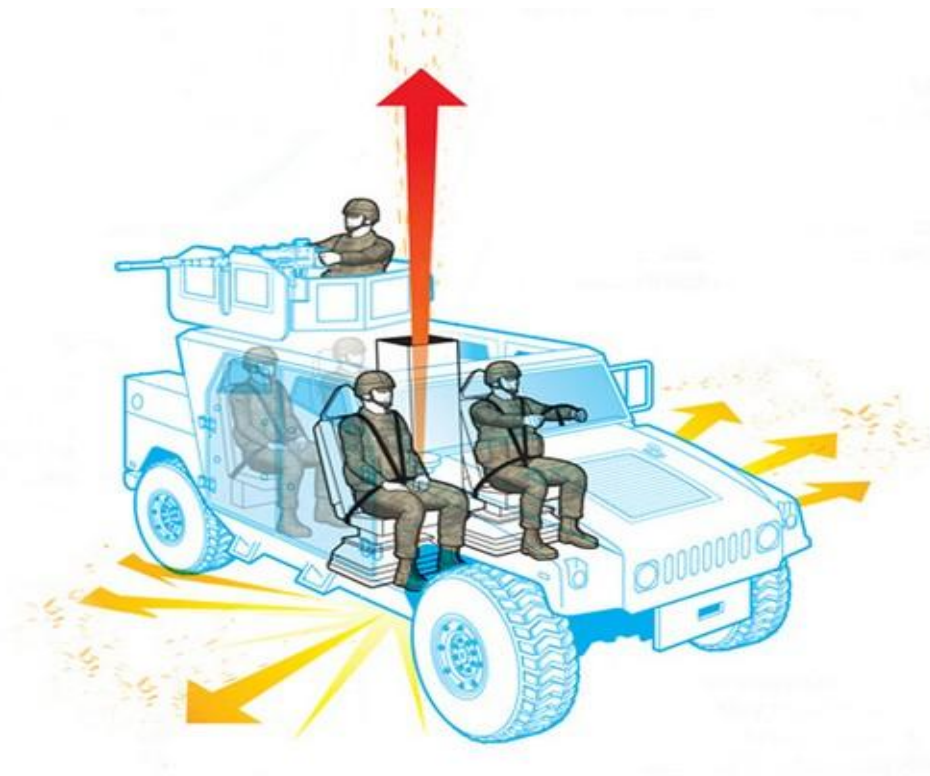
# Этап 1

## Подготовка к тестированию



# Этап 2

## Разработка теста



# Этап 3

## Реализация теста



# Этап 4

## Выполнение тестов на этапе интеграционного тестирования





# Этап 5

## Выполнение тестов на этапе системного тестирования



# Этап 6

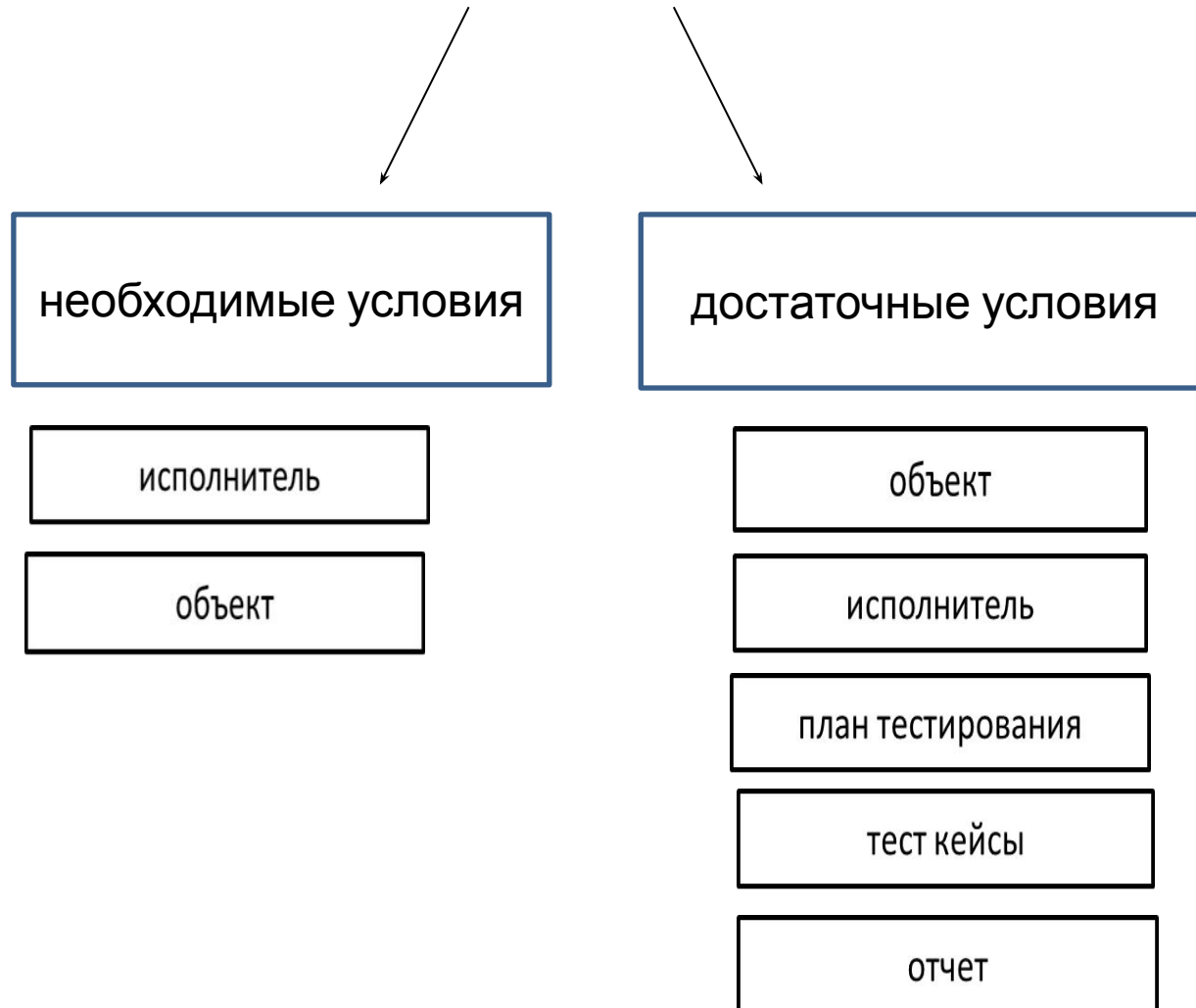
## Оценка тестирования



# Условия для проведения тестирования



# Условия для проведения тестирования



# Тестовые артефакты

- План тестирования
- Стратегия тестирования
- Набор тест кейсов и тестов
- Матрица соответствия требований
- Список проверки (Чек-лист)
- Дефекты / Баг Репорты
- Отчет о тестировании



# Test case

Тест кейсы разделяются по ожидаемому результату на :

- Позитивный тест кейс
- Негативный тест кейс



# Стадии работы по написанию тест кейса

- Сбор документации
- Проектирование тестового сценария
- Сбор доп. документации
- Практическое применение
- Корректировка кейса и рекомендации к нему

# Стандартные атрибуты тест-кейса

- Номер
- Название
- Предварительные шаги
- Шаги
- Ожидаемый результат
- Фактический результат





# Тест дизайн



# Тест дизайн

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



# Тестовое покрытие

Это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.



# Существующие подходы к оценке и измерению тестового покрытия

- Покрытие требований
- Покрытие кода



# Техники тест-дизайна

Эквивалентное  
Разделение

Анализ граничных  
Значений

Предугадывание  
ошибки

Причина /  
Следствие



## Пример шаблона тест-кейса

Действие	Ожидаемый результат
1. Открываем форму отправки сообщения	<ul style="list-style-type: none"><li>•Форма открыта</li><li>•Все поля по умолчанию пусты</li><li>•Обязательные поля помечены - *</li><li>•Кнопка "Отправить" не активна</li></ul>
2. Заполняем поля формы: <ul style="list-style-type: none"><li>•Тип обращения</li><li>•Контактное лицо</li><li>•Контактный телефон</li><li>•Сообщение</li></ul>	<ul style="list-style-type: none"><li>•Поля заполнены</li><li>•Кнопка "Отправить" - активна (Enabled)</li></ul>
3. Нажимаем кнопку "Отправить"	<ul style="list-style-type: none"><li>•Если введенные данные корректны -<ul style="list-style-type: none"><li>◦Сообщение "Заявка отправлена" выведено на экран.</li><li>◦Новая заявка появилась в списке на странице "Заявки".</li></ul></li><li>•Если введенные данные НЕ корректны -;<ul style="list-style-type: none"><li>◦Валидационное сообщение со всеми ошибками выведено на экран.</li><li>◦Заявка НЕ появилась в списке на странице "Заявки".</li></ul></li></ul>

## Пример позитивного тест-кейса (все поля ОК)

Действие	Ожидаемый результат
1. Открываем форму отправки сообщения	<ul style="list-style-type: none"><li>•Форма открыта</li><li>•Все поля по умолчанию пусты</li><li>•Обязательные поля помечены - *</li><li>•Кнопка "Отправить" не активна</li></ul>
2. Заполняем поля формы: <ul style="list-style-type: none"><li>•Тип обращения = Консультация</li><li>•Контактное лицо = йцукенгшщзйцуkenгшщзйцуке</li><li>•Контактный телефон = +7-916-111-11-11</li><li>•Сообщение</li></ul>	<ul style="list-style-type: none"><li>•Поля заполнены</li><li>•Кнопка "Отправить" - активна (Enabled)</li></ul>
3. Нажимаем кнопку "Отправить"	<ul style="list-style-type: none"><li>•Сообщение "Заявка отправлена" выведено на экран.</li><li>•Новая заявка появилась в списке на странице "Заявки".</li></ul>

## Пример негативного тест кейса (поле Контактное лицо - NOK):

Действие	Ожидаемый результат
1. Открываем форму отправки сообщения	<ul style="list-style-type: none"> <li>• Форма открыта</li> <li>• Все поля по умолчанию пусты</li> <li>• Обязательные поля помечены - *</li> <li>• Кнопка "Отправить" не активна</li> </ul>
2. Заполняем поля формы: <ul style="list-style-type: none"> <li>• Тип обращения = Консультация</li> <li>• Контактное лицо = @#\$%^&amp;;.?,&gt; V№"!()_{}[&lt;~</li> <li>• Контактный телефон = (916)333-33-33</li> <li>• Сообщение = йццуйцуйц(...)йцу - 1024 символа</li> </ul>	<ul style="list-style-type: none"> <li>• Поля заполнены</li> <li>• Кнопка "Отправить" - активна (Enabled)</li> </ul>
3. Нажимаем кнопку "Отправить"	<ul style="list-style-type: none"> <li>• Валидационное сообщение со всеми ошибками выведено на экран: "В поле "Контактное лицо" запрещено использование цифр и спец. символов."</li> <li>• Заявка НЕ появилась в списке на странице "Заявки".</li> </ul>



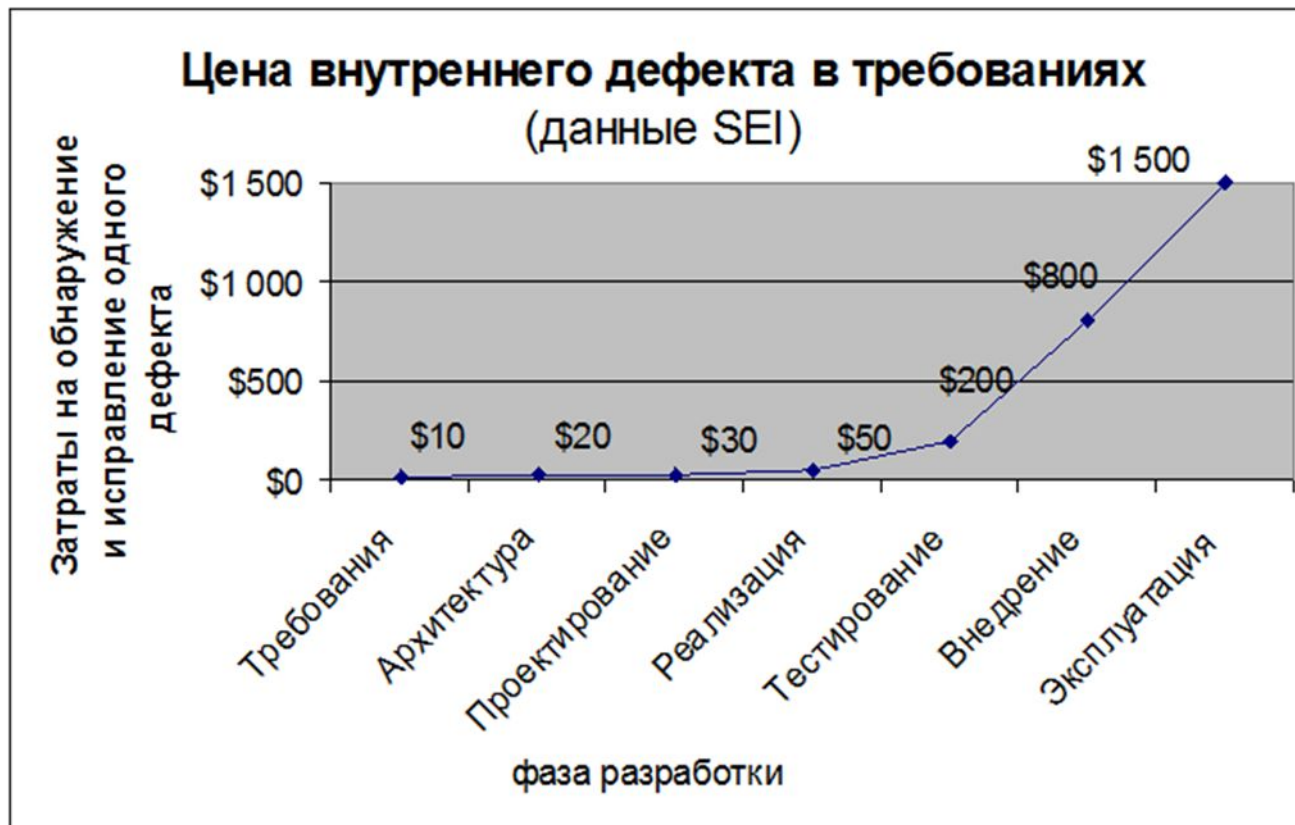
# Дефекты



Дефект — это несоответствие требованиям или функциональным спецификациям



# Цена дефекта



# База дефектов

База дефектов - это база данных, в которую заносятся обнаруженные в приложении дефекты



# Недостаточно просто фиксировать дефекты — их надо классифицировать

- Кричитность (severity)
- Приоритет (priority)



# По степени критичности

- Блокирующие (Blocker)
- Важные (Major)
- Обычные (Normal)
- Малозначимые (Minor)



# По приоритету

- Высокий
- Средний
- Низкий



# Системы отслеживания ошибок

- BUGS
- Bugzilla
- JIRA
- Trac
- Track Studio
- HP Quality Center
- Team Foundation Server





# Важные составляющие описания дефекта

- Краткое описание
- Подробное описание
- Воспроизводимость и шаги для воспроизведения дефекта
- Своевременность и актуальность регистрации дефекта
- Серьезность дефекта



# Распространенные ошибки при описании

- Сокращение инструкции по воспроизведению ошибки
- Отсутствие описания ошибочного поведения
- Отсутствие описания ожидаемого поведения
- Отсутствие обоснования ожидаемого поведения
- Повторное открытие старых дефектов
- Тестирование устаревшей версии программы
- Завышение приоритета дефекта



# Хороший отчет о дефекте

- Простота
- Полнота
- Объективность
- Нейтральность.

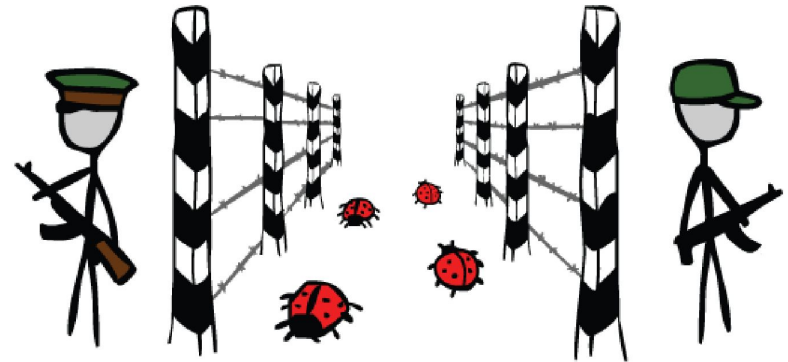


# Жизненный цикл дефекта



# Защита дефекта

- Экспертиза силами коллег
- Отстаивание собственной позиции
  - Уверенность в правильности дефекта
  - Настойчивость в доказательстве права дефекта на существование
  - Умение признать собственную ошибку



# Валидация дефектов

- Проверка исправления ошибки
- Изменение статуса проверенной ошибки
- Проверка функциональности, связанной с ошибкой
- Просмотр дубликатов просматриваемого дефекта



# Требования к протоколу тестирования

- **Список сценариев тестирования в соответствии с заданием на тестирование**
- **Отметки о результатах выполнения каждого шага сценария (Passed, Failed, N/A)**
- **Указания на дефекты, обнаруженные в рамках шага или рядом с ним (ID, важность, new or known)**
- **Информация о тестировании вне плана (при его отсутствии или отклонении в сторону)**
  - Чек-лист по тестируемым областям или функциональным элементам
  - Список дефектов для валидации и результаты их валидации (Closed or Opened)
  - Список обнаруженных дефектов с указанием ID & важности.
- **Информация о конфигурации клиентской машины и сервера**
- **Информация о дате тестирования, версии приложения, версии плана тестирования**
- **Информация о суммарном времени, затраченном на тестирование**

# Отчет о результатах тестирования





# Структура отчета о результатах тестирования

- Команда тестировщиков
- Описание процесса тестирования
- Краткое описание
- Расписание
- Рекомендации
- Статистика по ошибкам
- Список новых ошибок
- Статистика по всем ошибкам





До тестирования



После тестирования

Не бывает совершенных программ.  
Бывают неотестированные.

Доклад окончен.  
Спасибо за внимание!