ТЕСТОВЫЕ АРТЕФАКТЫ

ТЕСТОВЫЕ АРТЕФАКТЫ

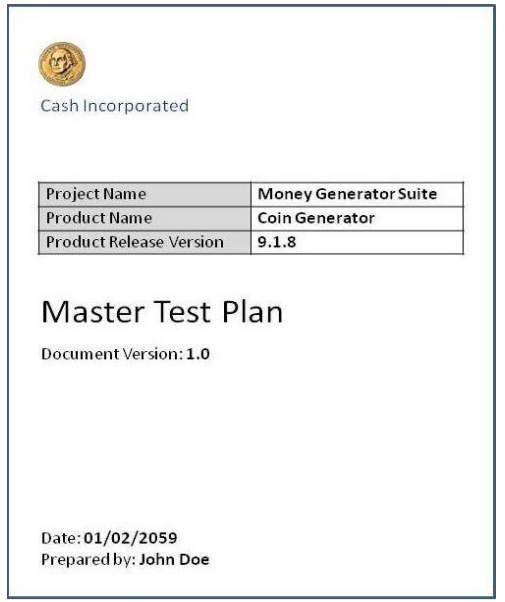
В соответствие с процессами или методологиями разработки ПО, во время проведения тестирования создается и используется определенное количество **тестовых артефактов** (документы, модели и т.д.).

Наиболее распространенными тестовыми артефактами являются:

- *План тестирования (Test Plan)*
- *Набор тест кейсов и тестов (Test Case & Test suite)*
- *Дефекты / Баг Репорты (Bug Reports / Defects)*

ТЕСТ ПЛАН

Test Plan (План тестирования) - 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.


Cash Incorporated

Project Name	Money Generator Suite
Product Name	Coin Generator
Product Release Version	9.1.8

Master Test Plan

Document Version: 1.0

Date: 01/02/2059
Prepared by: John Doe

Шаблоны тест планов:

- Test Plan Template RUP **(Rational Unified Process)**
- Test Plan Template IEEE 829

План проведения нагрузочного тестирования

СОДЕРЖАНИЕ ТЕСТ ПЛАНА

- **Что надо тестировать?**
  - описание объекта тестирования: системы, приложения, оборудования
- **Что будете тестировать?**
  - список функций и описание тестируемой системы и её компонент в отдельности
- **Как будете тестировать?**
  - стратегия тестирования, а именно: **виды тестирования** и их применение по отношению к объекту тестирования
- **Когда будете тестировать?**
  - последовательность проведения работ: подготовка (Test Preparation), тестирование (Testing), анализ результатов (Test Result Analisys) в разрезе запланированных фаз разработки
- **Критерии начала тестирования**:
  - готовность тестовой платформы (тестового стенда)
  - законченность разработки требуемого функционала
  - наличие всей необходимой документации

Критерии окончания тестирования:

- результаты тестирования удовлетворяют критериям качества продукта:
  - **требования к количеству открытых багов** выполнены
  - выдержка определенного периода без изменения исходного кода приложения **Code Freeze** (CF)
  - выдержка определенного периода без открытия новых багов **Zero Bug Bounce (ZBB)**

СОДЕРЖАНИЕ ТЕСТ ПЛАНА

1) Introduction (Введение)
2) Test Items (Объекты тестирования)
3) Features To Be Tested (Функциональности для тестирования)
4) Features Not To Be Tested (Функциональности которые не будут тестироватся)
5) Approach (Стратегия тестирования (виды, подходы, методы))
6) Item Pass/Fail Criteria (Критерии успешности тестирования)
7) Suspension Criteria and Resumption Requirements (Критерии остановки и возобновления тестирования)
8) Test Deliverables (Тестовые результаты)
9) Environmental Needs (Тестовое окружение)
10) Responsibilities (Ответсвенность)

ТЕСТОВЫЙ СЛУЧАЙ

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

Под тест кейсом понимается структура вида:
Action > Expected Result > Test Result

Действие	Ожидаемый результат	Результат теста

Пример:

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

Пример оформления тест кейса

АТРИБУТЫ ТЕСТ-КЕЙСА

1) Номер (ID)
2) Название (Summary/Name)
   Предусловие (PreConditions)
3) Шаги тест кейса и описание (Steps and Descriptions)
4) Ожидаемый результат (Expected result)
5) Пост-условие (PostConditions)
6) Автор (Designer)
7) Статус (Status)
8) Дата создания (Created)

Test Name:			
Status:			
Created Date:			
Designer:			
Pre Conditions:			
Steps	Description	Expected Result	Test Result (passed/failed/blocked)
Step 1			
Step 2			
Step 3			
Step 4			
Post Conditions:			

АТРИБУТЫ ТЕСТ-КЕЙСА

- Номер — уникальный идентификатор тест-кейса. Его удобно использовать для одинакового понимания, о какой проверке идет речь (например, дать ссылку в баге).
- Название — краткое описание **сути** проверки. Должно быть понятным! Кратко, но емко.
- Предварительные шаги — описание действий, которые необходимо выполнить, но прямого отношения к проверке они не имеют (например, зарегистрироваться в системе для проверки создания элемента).
- Шаги — описание действий, необходимых для проверки (например, создание элемента).
- Ожидаемый результат (ОР) — сама проверка: что мы ожидаем получить после выполнения шагов ("Элемент создан").

Название:		
Функция:		
Действие	Ожидаемый результат	Результат теста: • пройден • провален • заблокирован
Предусловие:		
Шаги теста:		
1	1	
2	2	
3	3	
Постусловие:		

СВОЙСТВА ТЕСТ-КЕЙСОВ

Тест-кейсы могут быть:

- Специфичными или общими.
- Простыми или сложными.
- Независимыми или связанными друг с другом.
- Позитивными или негативными.

СПЕЦИФИЧНОСТЬ ИЛИ ОБЩНОСТЬ

- Когда все детали прописаны до мелочей, при повторных выполнениях теста всегда будут выполняться строго одни и те же действия, что снижает вероятность обнаружить ошибку.
- Слишком общий тест-кейс сложно выполнять по многим объективным и субъективным причинам, а потому он вполне может остаться невыполненным.
- Однако интеграционные тесты, как правило, бывают более общими, чем иные. Это связано со спецификой интеграционного тестирования.
- Если в тесте прописано много мелких деталей, возрастает время его создания и поддержки.
- Однако недостаток деталей может усложнить работу новичка.

ПРОСТОТА ИЛИ СЛОЖНОСТЬ

Простые тесты оперируют за раз одним объектом.

- Каковы *преимущества* простых тест-кейсов?
  - Их легко выполнять.
  - Они понятны новичкам.
  - Они упрощают диагностику ошибки.
  - Они делают наличие ошибки очевидным.
- Каковы *преимущества* сложных тест-кейсов?
  - Больше шансов что-то сломать.
  - Пользователи, как правило, используют сложные сценарии.
  - Программисты сами редко проверяют такие варианты.

Следует постепенно повышать сложность тестов

ПРОСТОТА ИЛИ СЛОЖНОСТЬ

Где в ниже перечисленном простые тест-кейсы, а где – сложные?

Набор 1:

- 1. Откройте файл «1.txt». Файл открыт.
  2. Введите слово «Дом». Появляется слово «Дом.
  3. Сохраните файл. Кнопка «Сохранить» становится неактивной.

Набор 2:

- 1. В документе размером более 100 Мб создайте таблицу 100×100, в ячейку 50×50 вставьте картинку размером 30 Мб, применив к ней функцию «Авторасположение». Проверьте результат.

НЕЗАВИСИМОСТЬ ИЛИ СВЯЗАННОСТЬ

- Каковы *преимущества* *независимого самостоятельного тест-кейса*?
  - Его легко и просто выполнить.
  - Такие тесты могут работать даже после краха приложения на других тестах.
  - Такие тесты можно группировать любым образом и выполнять в любом порядке.
- Каковы *преимущества* наборов *тесно связанных тестов*?
  - Они имитируют работу реальных пользователей.
  - Они удобны для интеграционного тестирования.
  - Они удобны для разбиения на части тестов с большим количеством шагов.
  - Следующий в наборе тест использует данные и состояние приложения, подготовленные предыдущим.

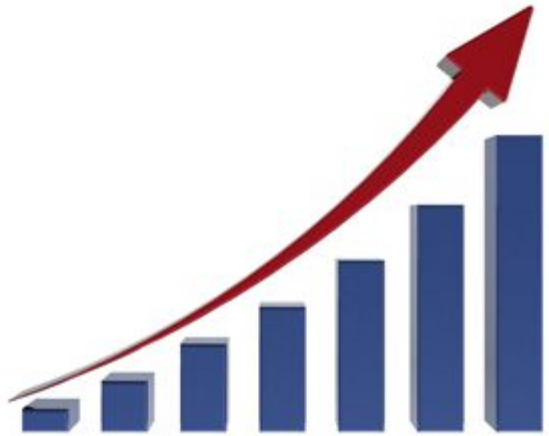
Промышленным стандартом являются независимые тесты.

ПОЗИТИВНОСТЬ ИЛИ НЕГАТИВНОСТЬ

- **Позитивные тесты** проверяют, что приложение делает то, на что оно рассчитано (т.е. такие тесты используют корректные данные и условия выполнения).
  Негативные тесты проверяют работу приложения в нестандартных условиях (при получении некорректных данных или команд или при работе в некорректных условиях).

Обе разновидности тестов важны и нужны, однако следует помнить последовательность их разработки и выполнения:

1. Простые позитивные.
2. Простые негативные.
3. Сложные позитивные.
4. Сложные негативные.



ПРИМЕРЫ (ОДИН ОЖИДАЕМЫЙ РЕЗУЛЬТАТ)

Есть внутренний сайт компании, которая проводит интернет www.test.ru. На сайте можно заводить карточки обслуживаемых зданий и карточки их жильцов. Карточки создает администратор. Тестовый стенд, на котором проверяются доработки перед выкладкой в PROD (он же *production*, окружение для пользователей) находится по другому адресу — www.dev_test.ru.

Тест-кейс № 1. Создание жильца без ФИО.

Шаги

- Зайти на сайт www.dev_test.ru (логин - test, пароль - test).
- Войти под учеткой администратора (логин - admin, пароль - 1)
- Перейти на вкладку "Жильцы".
- Нажать на кнопку "Создать карточку жильца".
- Нажать на кнопку "Сохранить", не заполняя никакие данные.

Ожидаемый результат

Появляется сообщение об ошибке:

"Заполните обязательные поля, отмеченные *", карточка не сохраняется.

ПРИМЕРЫ (НЕСКОЛЬКО ОЖИДАЕМЫХ РЕЗУЛЬТАТОВ)

Когда говорят о нескольких ожидаемы

- **Даны несколько вариантов вводимых результат.**
- **Несколько шагов, а не только последн**

Тест-кейс № 2. Создание жильца, проверка п

Шаги:

- Зайти на сайт www.dev_test.ru (логин - test,
- Войти под учеткой администратора (логин
- Перейти на вкладку "Жильцы"
- Нажать на кнопку "Создать карточку жильц
- Заполнить поле ФИО (см "Ожидаемый резу
- Нажать на кнопку "Сохранить".

Ожидаемый результат

Вводимое значение	Ожидаемый результат
Киселева Ольга Евгеньевна	Ок, карточка сохраняется
<Оставить поле пустым>	Ошибка – «Заполните обязательные поля, отмеченные *», карточка не сохраняется
2*4*6*8*11*14*17*20*23*26*29*32*35*38*41*	Ошибка – «Максимальная длина поля – 40 символов, введено - 41», карточка не сохраняется. (Такую строчку легко сформировать с помощью инструмента perlclip)
&*%#(^$@*&;	Ошибка – «Поле ФИО может содержать только буквы русского алфавита» (*см. статью про идиотов и ограничения*), карточка не сохраняется
Kiseleva Olga Evgenievna	Ошибка – «Поле ФИО может содержать только буквы русского алфавита» (*см. статью про идиотов и ограничения*), карточка не сохраняется
...	...

ПРИМЕР (ОШИБКИ)

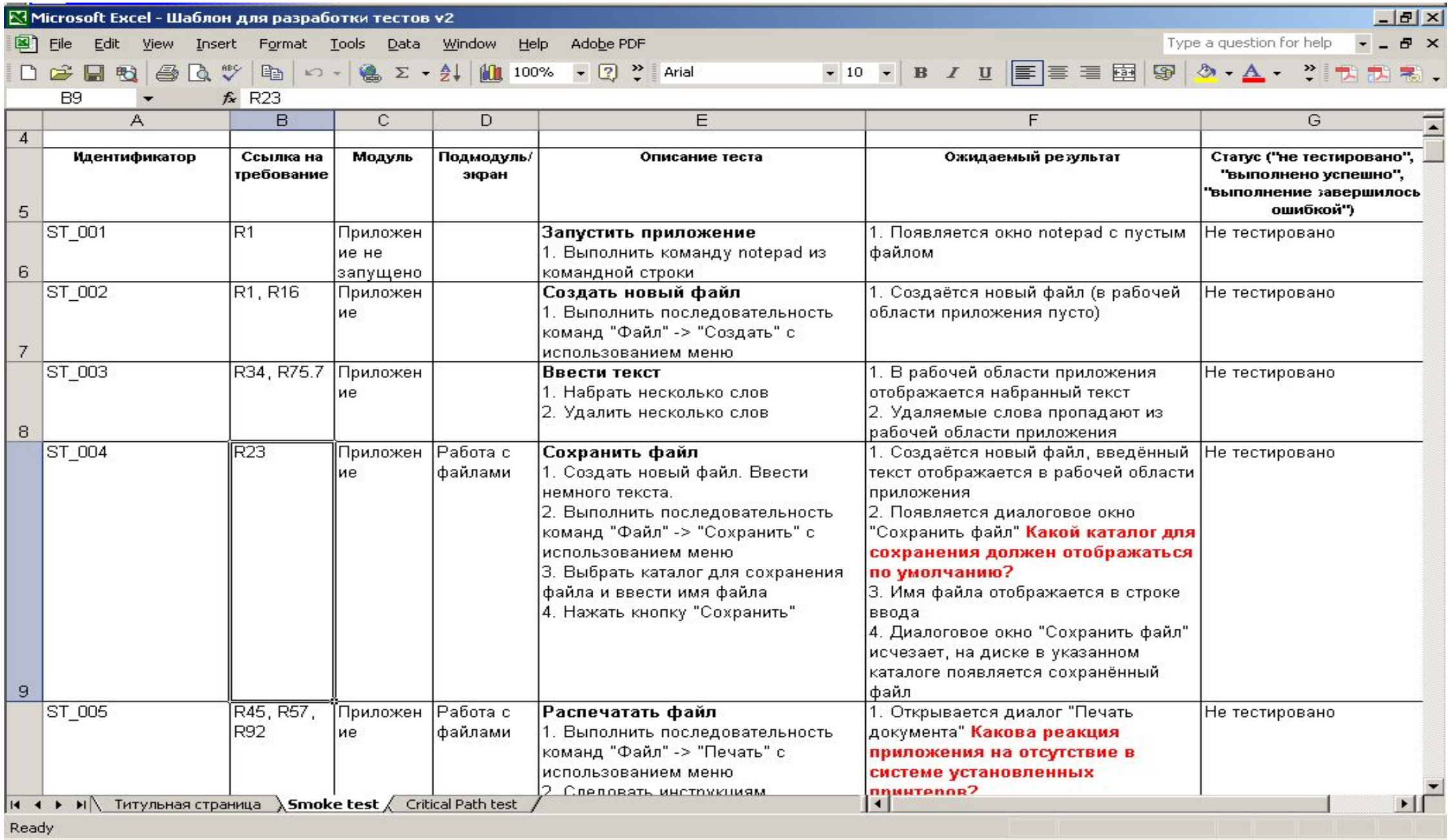
1. **Абстрактное название**
2. **Повелительное наклонение**
3. **Нет ссылки на сайт**
4. **Идет ссылка на PROD (окружение для пользователей)**
5. **Слишком детализировано**
6. **Нет нужной информации - непонятно, как авторизоваться**
7. **Нет описания проверки**

Тест-кейс № 01. Создание жильца.

Шаги:

- Зайди на сайт www.test.ru.
- Нажми на кнопку "Войти" в правом верхнем углу экрана.
- Залогинься с правами администратора.
- Перейди на вкладку "Жильцы".
- Нажми на кнопку "Создать карточку жильца".
- Введи корректные ФИО, например, "Иванов Иван Иванович" и сохрани карточку.

Ожидаемый результат — карточка создана.



	A	B	C	D	E	F	G
4							
5	**Идентификатор**	**Ссылка на требование**	**Модуль**	**Подмодуль/ экран**	**Описание теста**	**Ожидаемый результат**	**Статус ("не тестировано", "выполнено успешно", "выполнение завершилось ошибкой")**
6	ST_001	R1	Приложение не запущено		**Запустить приложение** 1. Выполнить команду notepad из командной строки	1. Появляется окно notepad с пустым файлом	Не тестировано
7	ST_002	R1, R16	Приложение		**Создать новый файл** 1. Выполнить последовательность команд "Файл" -> "Создать" с использованием меню	1. Создаётся новый файл (в рабочей области приложения пусто)	Не тестировано
8	ST_003	R34, R75.7	Приложение		**Ввести текст** 1. Набрать несколько слов 2. Удалить несколько слов	1. В рабочей области приложения отображается набранный текст 2. Удаляемые слова пропадают из рабочей области приложения	Не тестировано
9	ST_004	R23	Приложение	Работа с файлами	**Сохранить файл** 1. Создать новый файл. Ввести немного текста. 2. Выполнить последовательность команд "Файл" -> "Сохранить" с использованием меню 3. Выбрать каталог для сохранения файла и ввести имя файла 4. Нажать кнопку "Сохранить"	1. Создаётся новый файл, введённый текст отображается в рабочей области приложения 2. Появляется диалоговое окно "Сохранить файл" **Какой каталог для сохранения должен отображаться по умолчанию?** 3. Имя файла отображается в строке ввода 4. Диалоговое окно "Сохранить файл" исчезает, на диске в указанном каталоге появляется сохранённый файл	Не тестировано
	ST_005	R45, R57, R92	Приложение	Работа с файлами	**Распечатать файл** 1. Выполнить последовательность команд "Файл" -> "Печать" с использованием меню 2. Следовать инструкциям	1. Открывается диалог "Печать документа" **Какова реакция приложения на отсутствие в системе установленных принтеров?**	Не тестировано

ТЕСТОВЫЙ НАБОР (TEST SUITE)

- Test Suite - тестовый набор или тестовый комплект это набор тест кейсов, которые объединены тем что относятся к одному тестируемому модулю, функциональности, приоритету или одному типу тестирования.
- Каждый тест сьют состоит из более чем одного тест кейса и зачастую выполняется всей «пачкой» в процессе тестирования.

ЧЕК ЛИСТ

- **Чек лист** (Check list — контрольный список) — список, содержащий ряд необходимых проверок для какой-либо работы. Отмечая пункты списка, сотрудник может узнать о состоянии/корректности выполнения этой работы.

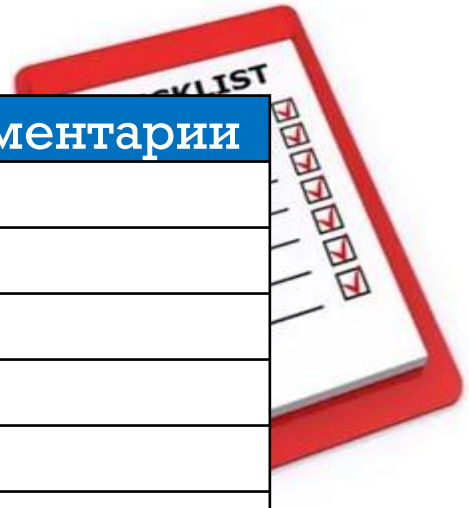
ЧЕК ЛИСТ

Зачем нужен чек-лист

- Не забыть что-то проте
- Помогает осуществлят

Что должно быть в чек

- перечень для проверки
- характеристики прило детализации.

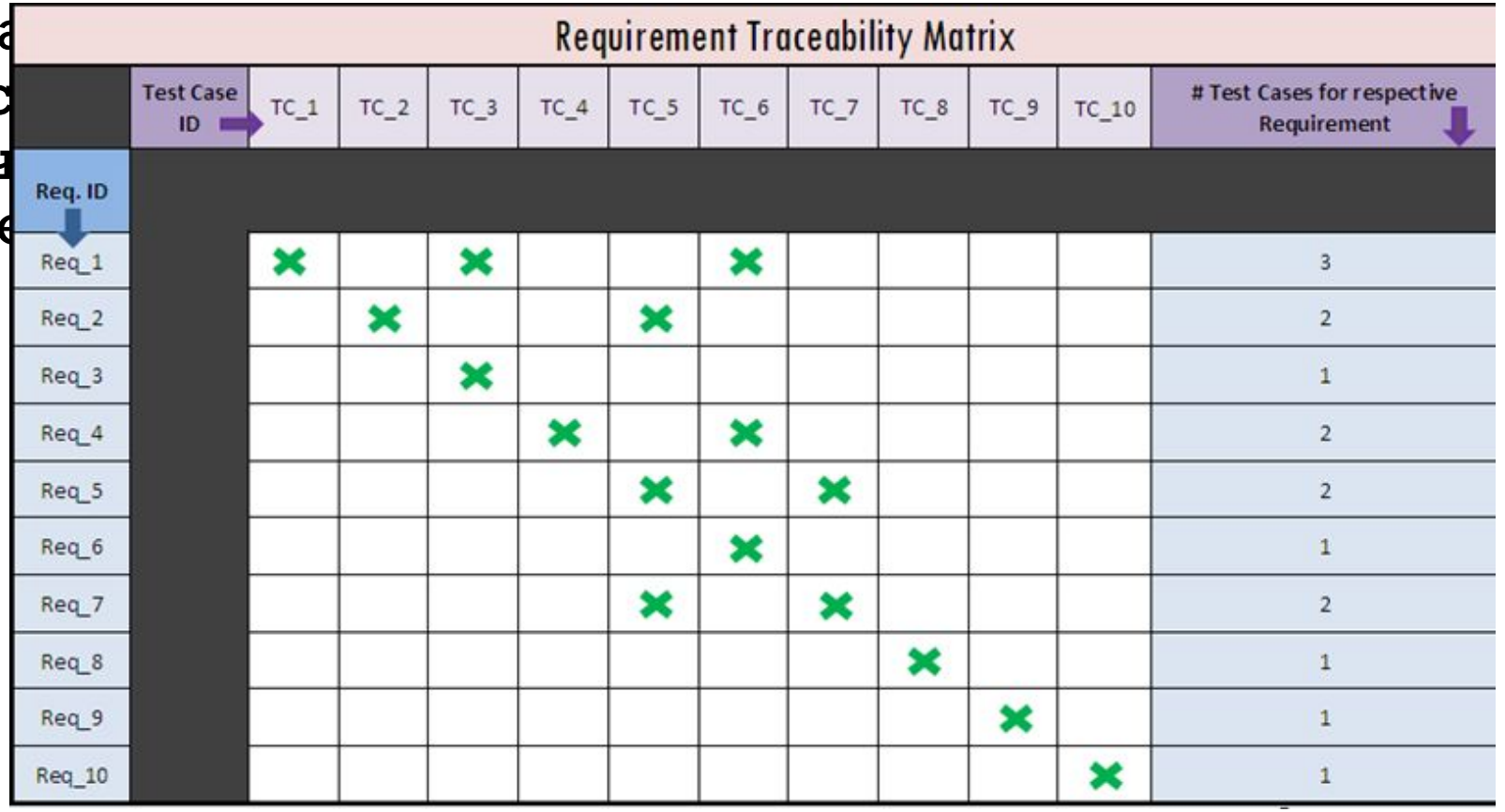


Проверка	Результат	Комментарии
Операции с файлами	ok	
Создание файла	ok	
Открытие файла	ok	
Сохранение документа	ok	
Печать	ok	
Редактирование файлов	bugs	
Отмена	ok	
Копирование	ok	
Вырезание	ok	
Вставка	ok	
Удаление	ok	
Поиск	fail	bug #123
Поиск с заменой	fail	bug #126
Вставка даты	ok	
Форматирование	ok	
Перенос строки	ok	
Изменение шрифта	ok	
Справка	ok	

МАТРИЦА СООТВЕТСТВИЯ ТРЕБОВАНИЙ

Traceability matrix — это двумерная таблица, содержащая соответсвие функциональных требований (functional requirements) продукта и подготовленных тестовых сценариев (test cases).

В заголовках колонок та
заголовках строк — тес
отметка, означающая, ч
тестовым сценарием те



Requirement Traceability Matrix

Req. ID	Test Case ID →	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8	TC_9	TC_10	# Test Cases for respective Requirement ↓
Req_1		✖		✖			✖					3
Req_2			✖			✖						2
Req_3				✖								1
Req_4					✖		✖					2
Req_5						✖		✖				2
Req_6							✖					1
Req_7						✖		✖				2
Req_8									✖			1
Req_9										✖		1
Req_10											✖	1

БАГ ИЛИ ДЕФЕКТ РЕПОРТ

Bug Reports / Defects - это документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Пример оформления баг репорта

ОСНОВНЫЕ ПОЛЯ БАГ / ДЕФЕКТ РЕПОРТА

Шапка	
Короткое описание (Summary)	Короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации.
Проект (Project)	Название тестируемого проекта
Компонент приложения (Component)	Название части или функции тестируемого продукта
Номер версии (Version)	Версия на которой была найдена ошибка
Серьезность (Severity)	Наиболее распространена пятиуровневая система градации серьезности дефекта: •S1 Блокирующий (Blocker) •S2 Критический (Critical) •S3 Значительный (Major) •S4 Незначительный (Minor) •S5 Тривиальный (Trivial)
Приоритет (Priority)	Приоритет дефекта: •P1 Высокий (High) •P2 Средний (Medium) •P3 Низкий (Low)
Статус (Status)	Статус бага. Зависит от используемой процедуры жизненного цикла

ОСНОВНЫЕ ПОЛЯ БАГ / ДЕФЕКТ РЕПОРТА (ПРОДОЛЖЕНИЕ)

Автор (Author)	Создатель баг репорта
Назначен на (Assigned To)	Имя сотрудника, назначенного на решение проблемы
Окружение	
ОС / Сервис Пак и т.д. / Браузера + версия / …	Информация об окружении, на котором был найден баг: операционная система, сервис пак, для WEB тестирования - имя и версия браузера и т.д.
…	
Описание	
Шаги воспроизведения (Steps to Reproduce)	Шаги, по которым можно легко воспроизвести ситуацию, приведшую к ошибке.
Фактический Результат (Result)	Результат, полученный после прохождения шагов к воспроизведению
Ожидаемый результат (Expected Result)	Ожидаемый правильный результат
Дополнения	
Прикрепленный файл (Attachment)	Файл с логами, скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на способ решения проблемы

ГРАДАЦИЯ СЕРЬЕЗНОСТИ ДЕФЕКТА

S1 Блокирующая (Blocker)
Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

S2 Критическая (Critical)
Критическая ошибка, неправильно работающая ключевая бизнес логика, проблема, приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

S3 Значительная (Major)
Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

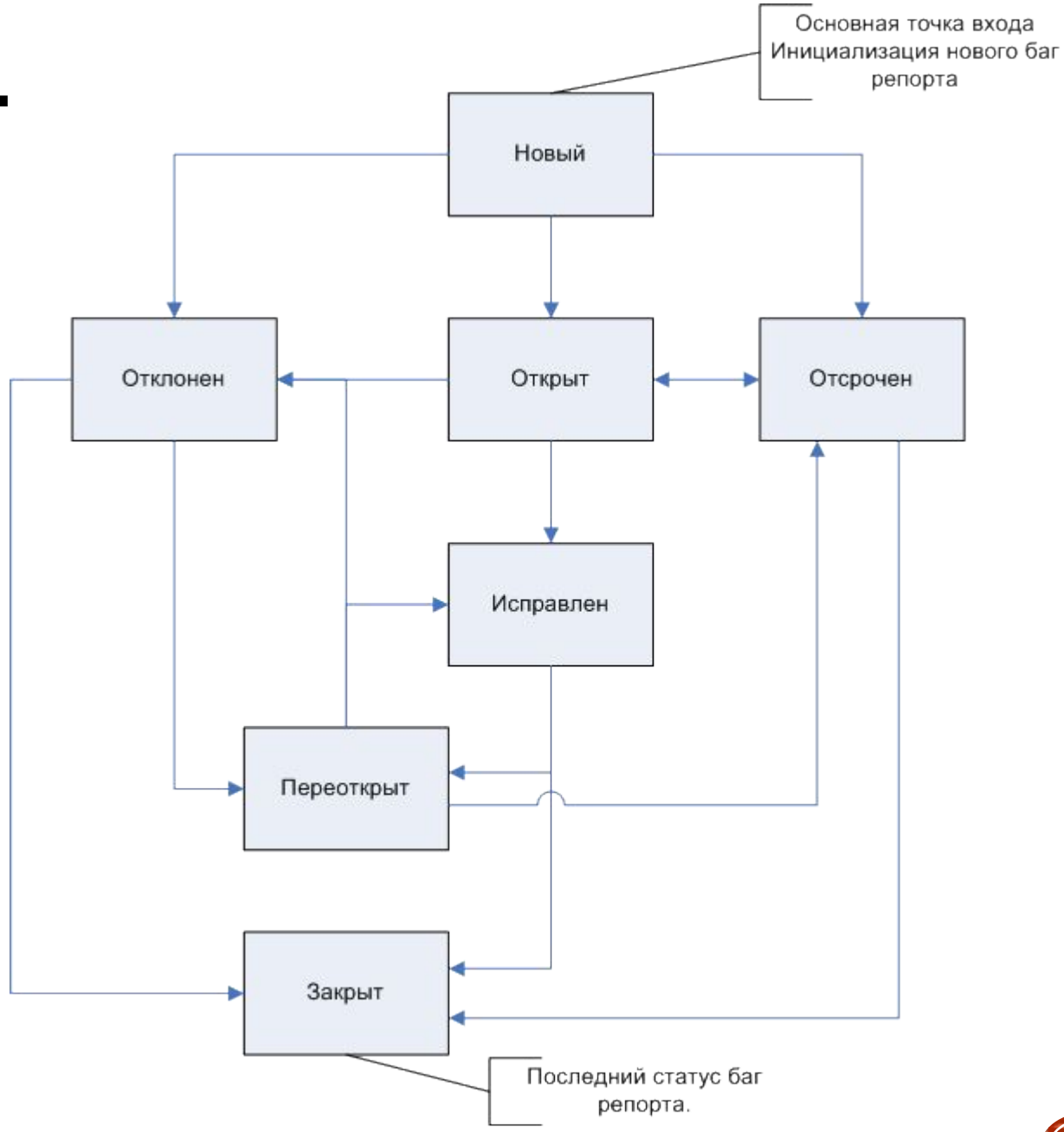
S4 Незначительная (Minor)
Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 Тривиальная (Trivial)
Тривиальная ошибка, не касающаяся бизнес логики приложения, не оказывающая

ГРАДАЦИЯ ПРИОРИТЕТА ДЕФЕКТА

- **P1 Высокий (High)**
  Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.
- **P2 Средний (Medium)**
  Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.
- **P3 Низкий (Low)**
  Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

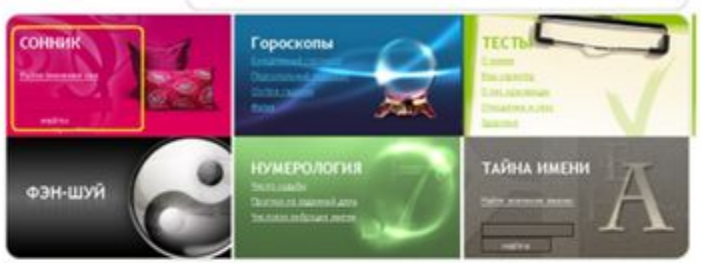
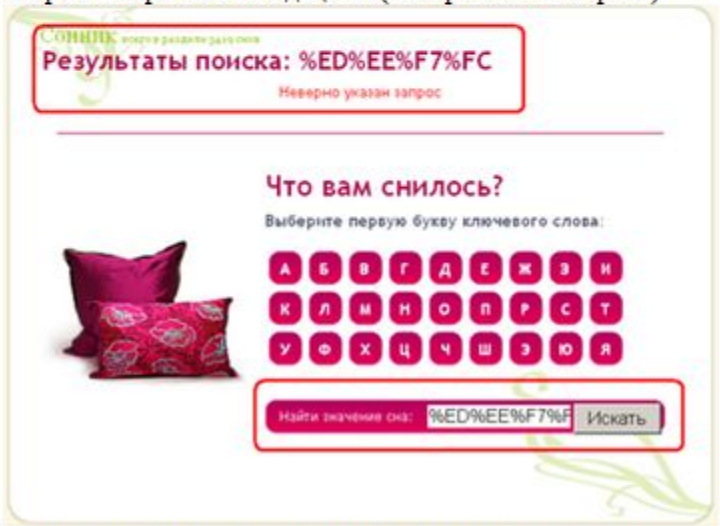
ЖИЗНЕННЫЙ ЦИКЛ БАГА



ПРИМЕР ОФОРМЛЕНИЯ БАГ РЕПОРТА

Баг Репорт

Короткое описание	Поиск в соннике на главной странице, с использованием русских слов, работает не правильно.
Проект	http://www.ameno.ru/
Компонент приложения	Поиск в соннике
Номер версии	0.001
Важность: • S1 Блокирующая (Blocker) • S2 Критическая (Critical) • S3 Значительная (Major) • S4 Незначительная (Minor) • S5 Тривиальная (Trivial)	**S3 Значительная (Major)**
Приоритет: • P1 Высокий (High) • P2 Средний (Medium) • P3 Низкий (Low)	заполняется менеджером
Статус	Новая
Автор	Алексей Булат
Назначен на	имя разработчика

Шаги воспроизведения	1. Открываем главную страницу сайта: http://www.ameno.ru/ --> Внизу страницы находим раздел: СОННИК (см. копию экрана - выделено желтой рамкой)  3. Введите поисковое слово, например "ночь" 4. Нажмите кнопку "Найти"
Фактический Результат	Запрос не прошел валидацию. (смотри копию экрана) 
Ожидаемый результат	Поиск прошел удачно, описание требуемого сна показано верно.