



Тестовые артефакты, документация

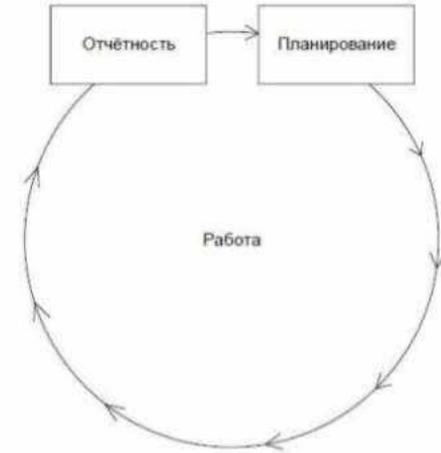
Risk-Based Testing



Жизненный цикл тестирования

Жизненный цикл тестирования заключается в следующих этапах:

1. Планирование и анализ требований
2. Уточнение критериев приемки
3. Уточнение стратегии тестирования
4. Разработка тестовой документации
5. Выполнение тест-кейсов и Фиксация найденных дефектов
6. Анализ результатов тестирования и Отчетность



Планирование (Planning) — непрерывный процесс принятия управленческих решений и методической организации усилий по их реализации с целью обеспечения качества некоторого процесса на протяжении длительного периода времени.

Отчетность (reporting) - это сбор и распространение информации о результатах работы (включая текущий статус, оценку прогресса и прогноз развития ситуации).



Планирование

Планирование какого-либо процесса определяется как деятельность, связанная с оптимальным распределением ресурсов, постановкой задач для успешного проведения такого процесса.

В тестировании планирование состоит из:

- Создания тест-плана
- Продумывания стратегии тестирования
- Оценки трудозатрат
- Прогнозирования сроков и составления графика проведения тестирования
- Деятельности по оценке рисков
- Определения используемых инструментов

Зачастую результатом планирования является созданный отдельный документ – План тестирования (тест-план). Однако его отсутствие не означает, что этап планирования был упущен вовсе.



Отчетность

Отчёты могут делиться на два вида относительно времени:

(Недельный, дневной, месячный)/ промежуточный.

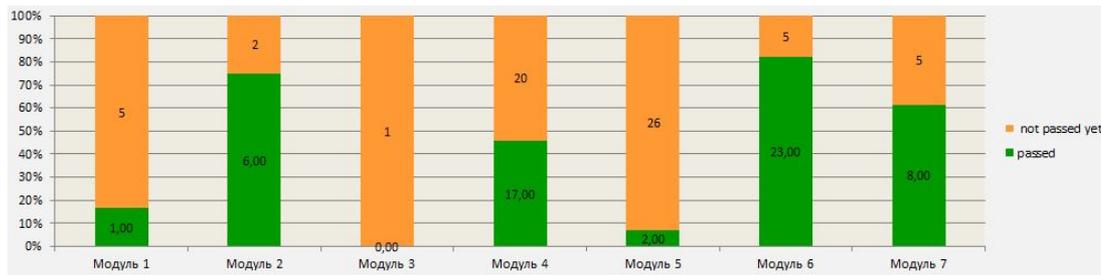
В нем обязательно должны содержаться две главных метрики:

- Оценка степени готовности продукта.
- Оценка проведенных работ по тестированию за время между отчетностями (прогресс).

Конечный /финальный.

В финальном отчете важно показать общий взгляд на проделанную работу (в контексте установленных метрик) и эволюцию продукта

Модуль	121	Прохождение	
	Общее количество ТК по модулю	passed	not passed yet
Модуль 1	6	1	5
Модуль 2	8	6	2
Модуль 3	1	0	1
Модуль 4	37	17	20
Модуль 5	28	2	26
Модуль 6	28	23	5
Модуль 7	13	8	5





Документация (Тестовые артефакты)

Тестовая документация - документ, облегчающий жизнь проектной команде.

Нужна она для:

1. Обеспечения стабильности покрытия требований проверками;
2. Сэкономить время на этапах тестирования, сводя их к проведению проверок, анализу и передачи результатов;
3. Снизить входной уровень квалификации тестировщика для проведения проверок



Тестовые артефакты



- баг - репорт
- замечание
- запрос на изменение
- отчет о тестировании (тест - репорт)

- тест- план
- тестовый комплект
- чек - лист
- тест - кейс



Bug Report

Дефект (он же баг) – это несоответствие фактического результата выполнения программы ожидаемому результату. Дефекты обнаруживаются на этапе тестирования программного обеспечения (ПО), когда тестировщик проводит сравнение полученных результатов работы программы (компонента или дизайна) с ожидаемым результатом, описанным в спецификации требований.

Итак, как только мы обнаруживаем баг, нам необходимо его задокументировать для продолжения жизненного цикла дефекта. Документ, который описывает баг, называется – баг репорт.

Баг репорт (bugreport) – это технический документ, который содержит в себе полное описание бага, включающее информацию, как о самом баге (короткое описание, серьезность, приоритет и т.д.), так и о условиях возникновения данного бага. Баг репорт должен содержать правильную, единую терминологию, описывающую элементы пользовательского интерфейса и события данных элементов, приводящих к возникновению бага.



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: <ul style="list-style-type: none">• S1 Блокирующий (Blocker)• S2 Критический (Critical)• S3 Значительный (Major)• S4 Незначительный (Minor)• S5 Тривиальный (Trivial) (подробнее смотрите ниже в разделе Серьезность дефекта)
Приоритет (Priority)	Приоритет дефекта: <ul style="list-style-type: none">• P1 Высокий (High)• P2 Средний (Medium)• P3 Низкий (Low) (подробнее смотрите ниже в разделе Приоритет дефекта)
Статус (Status)	Статус бага. Зависит от используемой процедуры и жизненного цикла бага (bug workflow and life cycle)
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



Test-report

Тест-репорт (Test report) — отчет о выполнении тест-кейсов, в нем обычно отмечается общая статистика, количество выполненных тест-кейсов и количество найденных ошибок.

Данный артефакт имеет огромную важность для принятия решения о продолжении тестирования или изменения его стратегии. Test Summary Report чаще всего пишется после регрессии продукта или в конце спринта.

Для чего:

- Дает информацию о качестве продукта “на данный момент” как самим тестировщикам, так и заказчику.
- Наглядно демонстрирует проблемы продукта. Эта статистика поможет принять решение о том, чему уделить больше внимания в следующем спринте.
- Показывает какие процессы на проекте следует улучшить, чтобы повысить качество продукта.



Test-report

Test Runs

Test series

New test run

NUMBER OF TESTS

7 in latest run



PEAK MEMORY

56.04 MB in latest run



Test runs

Date	Version	Duration	Status	Results	Devices
Aug 1, 2018, 8:43:31 AM	1.0 (1)	17 mins	⚡ FAILED	2/7 tests failed	3
Jul 27, 2018, 7:29:07 AM	1.0 (1)	6 mins	⚡ FAILED	2/7 tests failed	1
Jul 27, 2018, 7:00:28 AM	1.0 (1)	4 mins	⚡ FAILED	2/6 tests failed	1
Jun 28, 2018, 1:35:29 PM	1.0 (1)	3 mins	✅ PASSED	5 tests passed	1



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.

Хороший тест план должен как минимум описывать следующее:

- **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, тестирование, анализ результатов 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))



Test-plan

Наименование проекта		Introduction:	Веб-решение для подготовки и отправки отчетов в налоговые органы		
Дата завершения первого этапа	01.05.2018	Objective:	предоставление своевременной актуальной обратной связи по проекту всем заинтересованным лицам в рамках критериев качества: функциональность, надежность, практичность, эффективность, сопровождаемость, переносимость		
Дата сдачи проекта	01.09.2018	Principles:	Раннее тестирование; оптимальное кол-во документации; предотвращение багов; разумная автоматизация с участием всей команды; распространение идей качества на всю команду		
In Scope		Entry Conditions (for test execution):	People		
1	Функциональное тестирование	1 Реализован MVP для микросервисов	1	Менеджер проекта	
2	Автоматизация функционального тестирования	2 Есть точка входа (API или GUI) для приложения и редактора	2	Скрам-мастер	
3	Тестирование безопасности	3 Создана библиотека фронтových компонентов	3	Бизнес-аналитик	
4	Нагрузочное тестирование	4 Нет известных блокирующих багов	4	Команда разработки	
5	Тестирование юзабилити		5	Тестировщики	
6	Конфигурационное тестирование	Exit Conditions (for test execution):	Test Environment		
7	Тестирование на отказ и восстановление	1 Все включенное в план тестирование завершено	1	Микросервисы располагаются в докер-контейнерах	
8	Тестирование взаимодействия микросервисов на уровне глубокого бэкенда	2 Нет багов с высоким приоритетом критичности	2	Предполагается, что большая часть пользователей будет работать в Windows-системах и браузерах IE, Google Chrome и Mozilla Firefox	
9	Тестирование установки (плагинов)	3 Низкоприоритетные баги не влияют на основной сценарий работы и их список согласован с разработчиками и менеджерами	3	Все тестирование, кроме нагрузки и безопасности, базируется в TFS + MTM + VS	
Out of Scope		Risks	Timescales (each iteration)		
1	Не будет регрессионного тестирования каждого из видов отчетов по одному и тому же документообороту	1 Неправильно спроектированная архитектура	1	Планирование	2 дня
2	Автоматизация всех кейсов для всех видов тестирования	2 Неправильные/изменяющиеся приоритеты	2	Тест-анализ и Тест-дизайн	2 дня
		3 Недоступность/отсутствие тестовых сред	3	Разработка тестовых скриптов	3 дня
		4 Некорректная оценка трудозатрат	4	Выполнение тестов	2 дня
			5	Подведение итогов	1 день



Чек лист (Check List)

Это список проверок для тестирования продукта.

Чек-листы устроены предельно просто. Любой из них содержит перечень блоков, секций, страниц, других элементов, которые следует протестировать

Сайт "checklist.example"
Регистрация
Email
Соцсети
Валидация полей
Авторизация
Пользователь активирован
Пользователь не активирован <input type="checkbox"/>
Восстановление пароля
Валидация полей
Профиль
Изменение имени
Изменение email
Изменение пароля
Валидация полей
Подписка
Удаление аккаунта



Чек лист (Check List)

Сайт "checklist.example"	Тестировщик 1	Тестировщик 2	Тестировщик 3
	Firefox Browser 70.0.1	Google Chrome 78.0.3904.108	Opera 65.0.3467.42
Регистрация			
Email	Passed	Passed	No run
Соцсети	Failed	Passed	Failed
Валидация полей	Passed	Passed	Passed
Авторизация			
Пользователь активирован	Passed	Passed	Failed
Пользователь не активирован	Passed	Skipped	Passed
Восстановление пароля	Passed	No run	Passed
Валидация полей	Passed	Failed	Skipped
Профиль			
Изменение имени	Passed	Passed	Failed
Изменение email	Failed	Skipped	Passed
Изменение пароля	Failed	Passed	Skipped
Валидация полей	Passed	Skipped	
Подписка	Failed	Passed	Passed
Удаление аккаунта	No run	No run	No run



Test case

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

Под тест кейсом понимается структура вида:

- Выполняемое действие (Action)
- Ожидаемый результат (Expected result)
- Фактический результат (Test result)

Action	Expected Result	Test Result (passed/failed/blocked)
Открыть страницу входа	Страница входа открылась	passed



Test case

Непосредственно сам тестовый случай состоит из 3 частей:

- **PreConditions**(Предусловия) – либо список шагов, которые приводят проверяемую систему в состояние, пригодное для тестирования, либо список проверок условий того, что система уже находится в необходимом состоянии.
- **Test Case Description**(Описание тестового случая) – список действий, с помощью которых осуществляется основная проверка функционала (после которой и сверяется фактический результат с ожидаемым).
- **PostConditions**(Постусловия) – список действий, которые возвращают систему в исходное состояние.

ТЕСТ-КЕЙС №1. Постройка казармы (FlatArt Team)

Предусловия: Создать или проверить наличие построенной казармы. Количество ресурсов НЕ позволяет построить казарму.

Шаги:

- 1) Зайти в игру
- 2) Нажать кнопку "ИГРАТЬ" и начать игровую сессию
- 3) Нажать кнопку "ЗДАНИЯ" и выбрать здание "Казарма"
- 4) Выбрать свободную ячейку на игровом поле
- 5) Нажать на кнопку "ПОСТРОИТЬ"

Ожидаемый результат	Фактический результат
Постройка не осуществлена. Вывод сообщения о нехватке ресурсов.	

Постусловия: если казарма уже построена, удалить ее



Test case

Тест - кейсы проверяют не более одной идеи. При этом два или более ожидаемых результата легитимны.

К плохому стилю относится:

- зависимость тест - кейсов друг от друга;
- нечеткая формулировка шагов;
- нечеткая формулировка идеи тест - кейса / ожидаемого результата.

Хороший стиль - объединять тест - кейсы в тест - комплекты. (как правило один тест - комплект - один файл. Как правило, один тест - комплект включает в себя тест - кейсы родственные друг другу, что они проверяют определенный участок интернет- проекта, описанные в определенном спеке

Test case



Datahub

Overview **Todo** Milestones Test Runs & Results **Test Cases** Reports

C1 Edit Test Case

Title *
Verify CSV import with attached test data file

Section * Tables & Images **Template *** Test Case (Steps) **Type *** Functional **Priority *** Medium

Estimate
5m

Exploratory Session
Test Case (Steps)
Test Case (Text)

Preconditions

Verifies the CSV import functionality for the template module. No document has been opened and all user settings have been reset to default values before executing this test case.

The preconditions of this test case. Reference other test cases with [C#] (e.g. [C17]).

Steps

- 1 Open the New Document wizard by selecting File | New.
* The dialog is displayed
* All default templates have been loaded
* The Empty template has been pre-selected
- 2 Select the Corporate template and confirm it by clicking OK.
* A new document is opened, using the Corporate template
* The page counter in the statusbar displays `1/1`
* The document area is focused
- 3 Load the attached CSV file into the new document to import it
* The CSV imports without errors
* All 8 columns are displayed
* All 214 records have been imported

Save Test Case Cancel Add Step



Test case

Test Project

Overview | Todo | Milestones | Test Runs & Results | **Test Cases** | Reports | Administration

Test Cases

Successfully updated the test case.

Sort: Section | Filter: None

Sprint-142-2016.07.29 (0)

ID	Title
<input type="checkbox"/>	

Add Case | Add Subsection

ISVRNSOA-103: Uygunluk - Pozisyona göre puanlama (3)

ID	Title
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	C63 Test case-1 uygunluk kontrolu-1
<input type="checkbox"/>	C64 Test case-2 uygunluk kontrolu-2
<input type="checkbox"/>	C65 Test case-3 uygunluk kontrolu-3

Add Case | Add Subsection

C63 Test case-1 uygunluk kontrolu-1

Type	Priority	Estimate	References
Other	Medium	None	ISVRNSOA-103

Preconditions

önkoşullar

Steps

kariyer.net açılır.
login sayfasına gidilir.
login olunur.
...
....

Expected Result

Adayın uygun olması beklenir :)

Contains 2 sections and 3 cases. [Edit description](#)

Subgroups | Add Section

Sprint-142-2016.07.29

ISVRNSOA-103: Uygunluk - Pozisyon:



Test Case vs Check List

Отличие между ними в том, что **чек-листы** показывают направление тестирования, а **тест-кейсы** подробно описывают как тестировать

преимущества чек-листов по сравнению с тест-кейсами:

- нивелирование эффекта пестицида в регрессионном тестировании (эффект, при котором при регулярном прогоне тестовых сценариев ошибки перестают находиться.)
- расширение тестового покрытия за счёт отличий при прохождении
- сокращение затрат на содержание и поддержку тестов: не надо писать много букв!
- отсутствие рутины, которую так не любят квалифицированные тестировщики
- возможность проходить и комбинировать тесты по-разному, в зависимости от предпочтений сотрудников



Матрица требований

Матрица соответствия **требований** - это двумерная таблица, содержащая соответствие функциональных **требований** (functional requirements) продукта и подготовленных тестовых сценариев (test cases).

Матрица соответствия **требований** используется QA-инженерами для валидации покрытия продукта тестами.

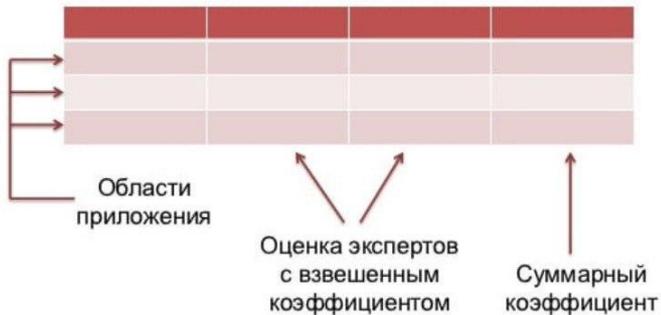
	Тест-кейс 1	Тест-кейс 2	Тест-кейс 3	Тест-кейс 4	Тест-кейс 5
Требование 1.	+				
Требование 2.		+			
Требование 3.		+	+		
Требование 4.					+
Требование 5. ...				+	



Risk-based testing

Тестирование, основанное на рисках - это тип тестирования, который функционирует как **организационный принцип**, используемый **для определения приоритетов тестирования**, основываясь на риске сбоев, важности функций и вероятности возникновения сбоя

Для проверки выбирают самые рискованные области создаваемого программного обеспечения. Это позволяет предусмотреть негативные сценарии и успешно реализовать бизнес-цели заказчика.



Область	Заказчик x2	Тестер x1	Итог
Область 1	2	1	5
Область 2	1	2	4
Область 3	3	3	9
Область 4	4	1	9