



Тестовая документация. ПО
для управления тестовой
документацией.

- Виды тестовой документации.
- Структуры тестовой документации.
- ПО для управления тестовой документацией.



- тест план;
- тестовая стратегия;
- чек-лист;
- тестовый сценарий (Test Case);
- ТЕСТ-СЬЮТ;
- пользовательская история (User Story);
- отчет о дефекте.



Тестовая документация – артефакты, которые тестировщик использует в своей работы

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 описывать следующее:

1. Что надо тестировать? Описание объекта тестирования: системы, приложения, оборудования.
2. Что будете тестировать? Список функций и описание тестируемой системы и её компонент в отдельности.
3. Как будете тестировать? Стратегия тестирования, а именно: виды тестирования и их применение по отношению к объекту тестирования.
4. Когда будете тестировать? Последовательность проведения работ: подготовка (Test Preparation), тестирование (Testing), анализ результатов (Test Result Analysis) 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).



Преимущества тест плана:

- Возможность приоритезации задач по тестированию.
- Построение стратегии тестирования, согласованной со всей командой.
- Возможность вести учет всех требуемых ресурсов (как технических, так и человеческих).
- Планирование использования ресурсов на тестирование.
- Просчет рисков, возможных при проведении тестирования.





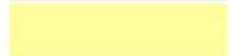
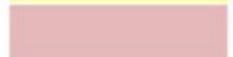
Тестовая стратегия определяет то, как мы тестируем продукт. Это набор мыслей и идей, которые направляют процесс тестирования.

В стратегии тестирования описывают:

- Тестовую среду.
- Анализ рисков проекта.
- Инструменты, которые будут использовать, чтобы провести автоматизированное тестирование и для других целей.
- План действий при непредвиденных обстоятельствах.

Как выглядит?

Стратегия может быть представлена как в виде традиционно расписанного документа, так и в более наглядном формате, например, используя таблицу:

Приоритет объектов системы и действий		Готовность тестов	
Высокий		Готово	
Средний		В процессе	
Низкий		Не готово	

Объект системы и действия	Стратегия	Функциональное тестирование	Нефункциональное тестирование				
			Юзабилити	Нагрузка	Скорость	Безопасность	Окружения
			Тесты	Тесты	Тесты	Тесты	Тесты
Объект 1	Т.к. это важный объект (один из ключевых объектов с высочайшим приоритетом), нам требуется детальное тестирование и подготовка тест-анализа на уровне S&T. После подготовки, эти тесты необходимо задокументировать, т.к. мы будем часто их проходить.	link	link	link		link	
Действие 1	Одно из ключевых действий. Требуется ТА по ДПЗ и согласование этого ТА со всей командой. Чаще всего здесь сразу вкратце выписывается, что именно планируем тестировать, какие параметры?	link			link	link	link
Действие 2	Одно из ключевых действий. Требуется ТА по ДПЗ и согласование этого ТА со всей командой.	link	link	link			
Действие 3	Важное действие, делаем ТА без согласования	link	link	link			
Действие 4	Сделаем полноценное ТА, если будет время, пока - исследовательское тестирование						
Объект 2	S&T						
Действие 1	Важное действие, делаем ТА без согласования	link	link	link	link		link
Действие 2	Сделаем полноценное ТА, если будет время, пока - исследовательское тестирование						

Тест-план:

- устанавливает цели процесса тестирования;
- определяет, что будет проверяться.

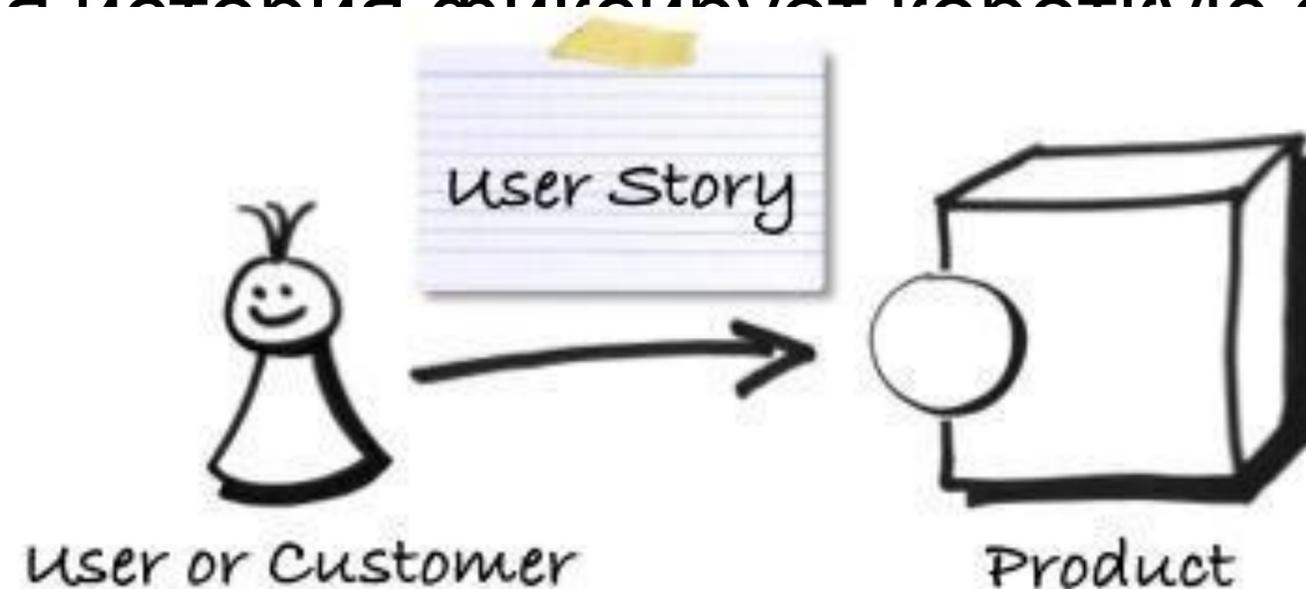
Стратегия тестирования:

- описывает, как достичь целей, поставленных в плане тестирования.

Пользовательские истории (User Story) — способ описания требований к разрабатываемой системе, сформулированных как одно или более предложений на повседневном или деловом языке пользователя. Пользовательские истории используются гибкими методологиями разработки программного обеспечения для спецификации требований.

Иными словами, **User Story** — это короткая формулировка намерения, описывающая что-то, что система должна делать для пользователя.

Пользовательская история формулирует короткую формулировку функции на картах онлайн-средства.



- Авторизоваться в моем портале мониторинга энергопотребления.
- Посмотреть ежедневный уровень потребления.
- Проверить мой текущий тариф.

Чек-лист - набор идей по тестированию, разработке, планированию и управлению. А также, это перечень формализованных тестовых случаев в удобном для проведения проверок виде. Тестовые случаи в чек-листе не должны быть зависимыми друг от друга.

Цель чек-листа – обеспечить стабильность покрытия требований проверками необходимыми и достаточными для заключения о соответствии им продукта. Особенностью является то, что чек-листы komponуются теми тестовыми случаями, которые показательны для определенного требования.



Чек лист обязательно должен содержать в себе следующую информацию:

- идея проверок;
- набор входных данных;
- ожидаемые результаты;
- булевая отметка о прохождении/непрохождении тестового случая;
- булевая отметка о совпадении/несовпадении фактического и ожидаемого результата по каждой проверке;
- может также содержать шаги для проведения проверки, данные об особенностях окружения и прочую информацию необходимую для проведения проверок.

Чек-лист должен обладать рядом важных свойств:

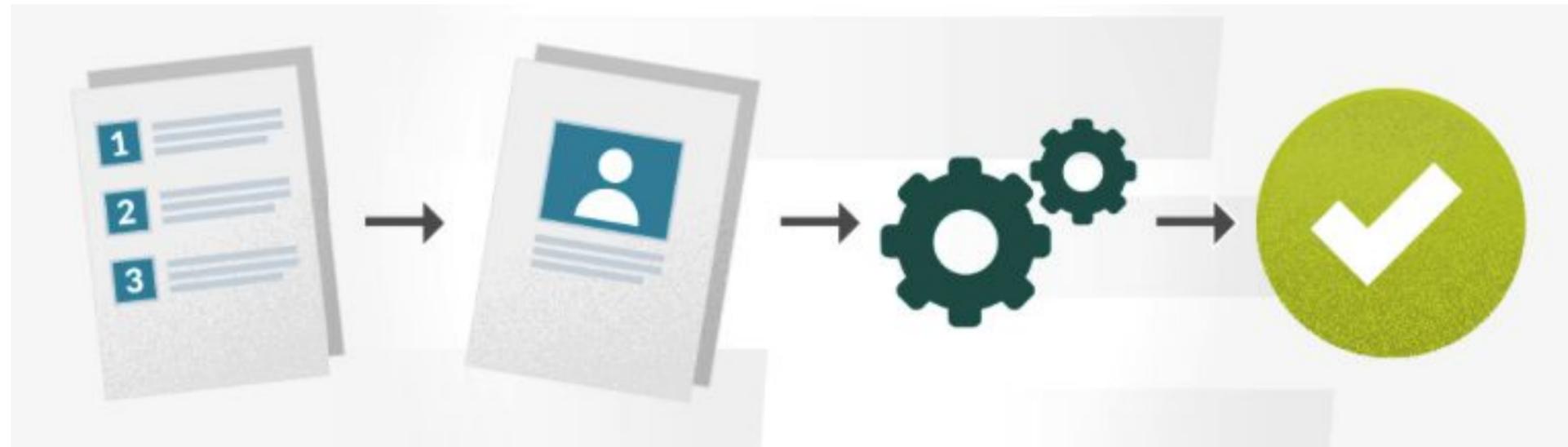
- Логичность.
- Последовательность и структурированность.
- Полнота и избыточность.

Правила составления чек-листов:

- Одна операция.
- Пункты чек-листа - это минимальные полные операции. Например, заказать изготовление визиток и доставить визитки в офис - это 2 разных операции. Поэтому в чек-листе они отображаются отдельными пунктами: визитки заказаны и визитки доставлены в офис.
- Пункты пишутся в утвердительной форме. Цель чек-листа – проверка готовности задачи, поэтому лучше составлять пункты в утвердительной форме - «заказаны, доставлены». Сравните формулировку: «заказать визитки» и «визитки заказаны».
- Оптимальное количество пунктов - до 20. Чек-листы не должны быть длинными. Если все же это требуется, то лучше разбить задачу на несколько этапов и составить к каждому этапу отдельный чек-лист.

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

Набор тест-кейсов (test case suite, test suite, test set) - совокупность тест-кейсов, выбранных с некоторой общей целью или по некоторому общему признаку.

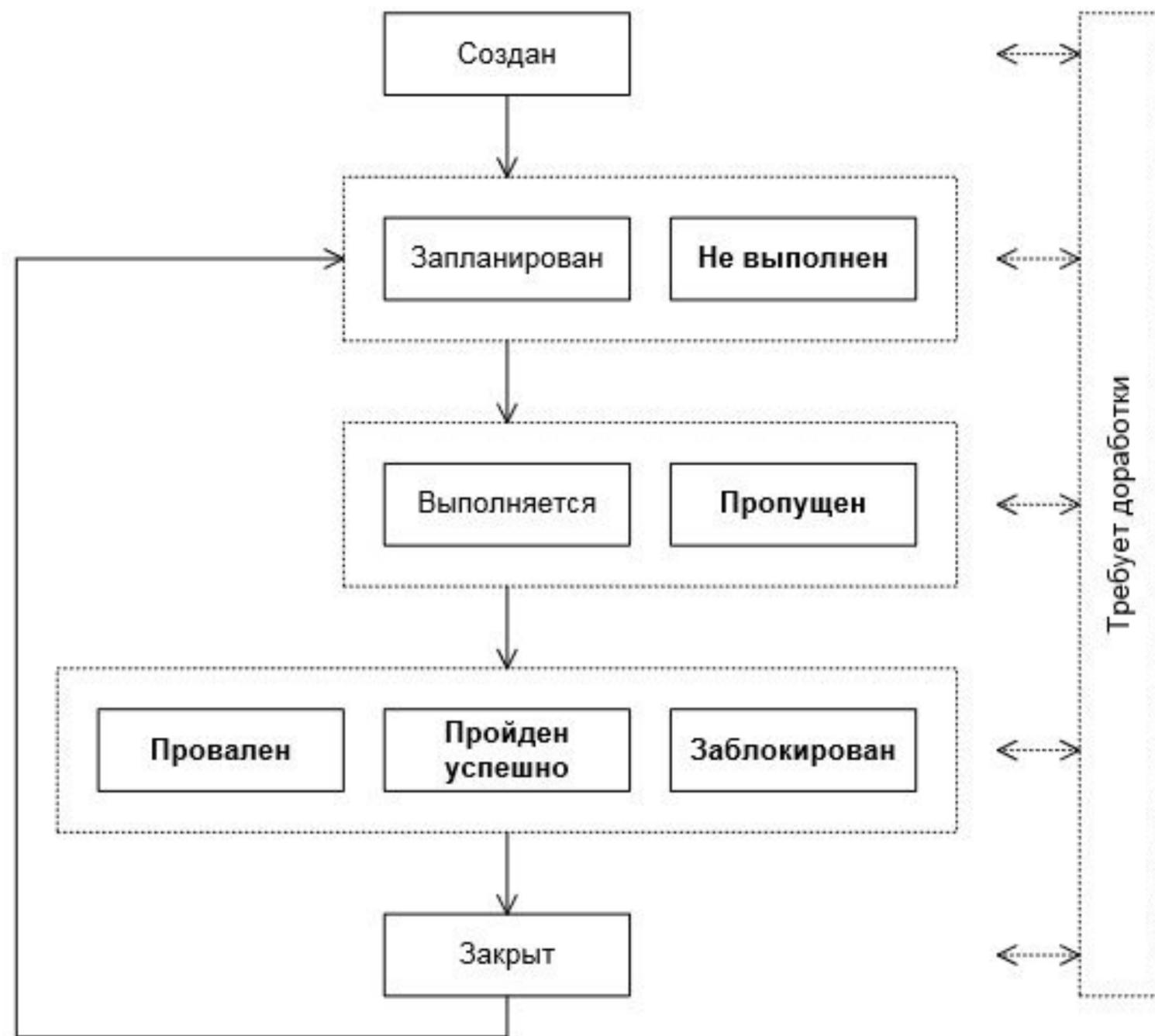


Зачем нам тест-кейсы?

Наличие тест-кейсов позволяет:

- Структурировать и систематизировать подход к тестированию (без чего крупный проект почти гарантированно обречён на провал).
- Вычислять метрики тестового покрытия (test coverage metrics) и принимать меры по его увеличению (тест-кейсы здесь являются главным источником информации, без которого существование подобных метрик теряет смысл).
- Отслеживать соответствие текущей ситуации плану (сколько примерно понадобится тест-кейсов, сколько уже есть, сколько выполнено из запланированного на данном этапе количества и т.д.).
- Уточнить взаимопонимание между заказчиком, разработчиками и тестировщиками (тест-кейсы зачастую намного более наглядно показывают поведение приложения, чем это отражено в требованиях)..
- Хранить информацию для длительного использования и обмена опытом между сотрудниками и командами (или, как минимум, не пытаться удержать в голове сотни страниц текста).
- Проводить регрессионное тестирование и повторное тестирование (которые без тест-кейсов было бы вообще невозможно выполнить).
- Повышать качество требований (написание чек-листов и тест-кейсов - хорошая техника тестирования требований).
- Быстро вводить в курс дела нового сотрудника, недавно подключившегося к проекту.

Жизненный цикл

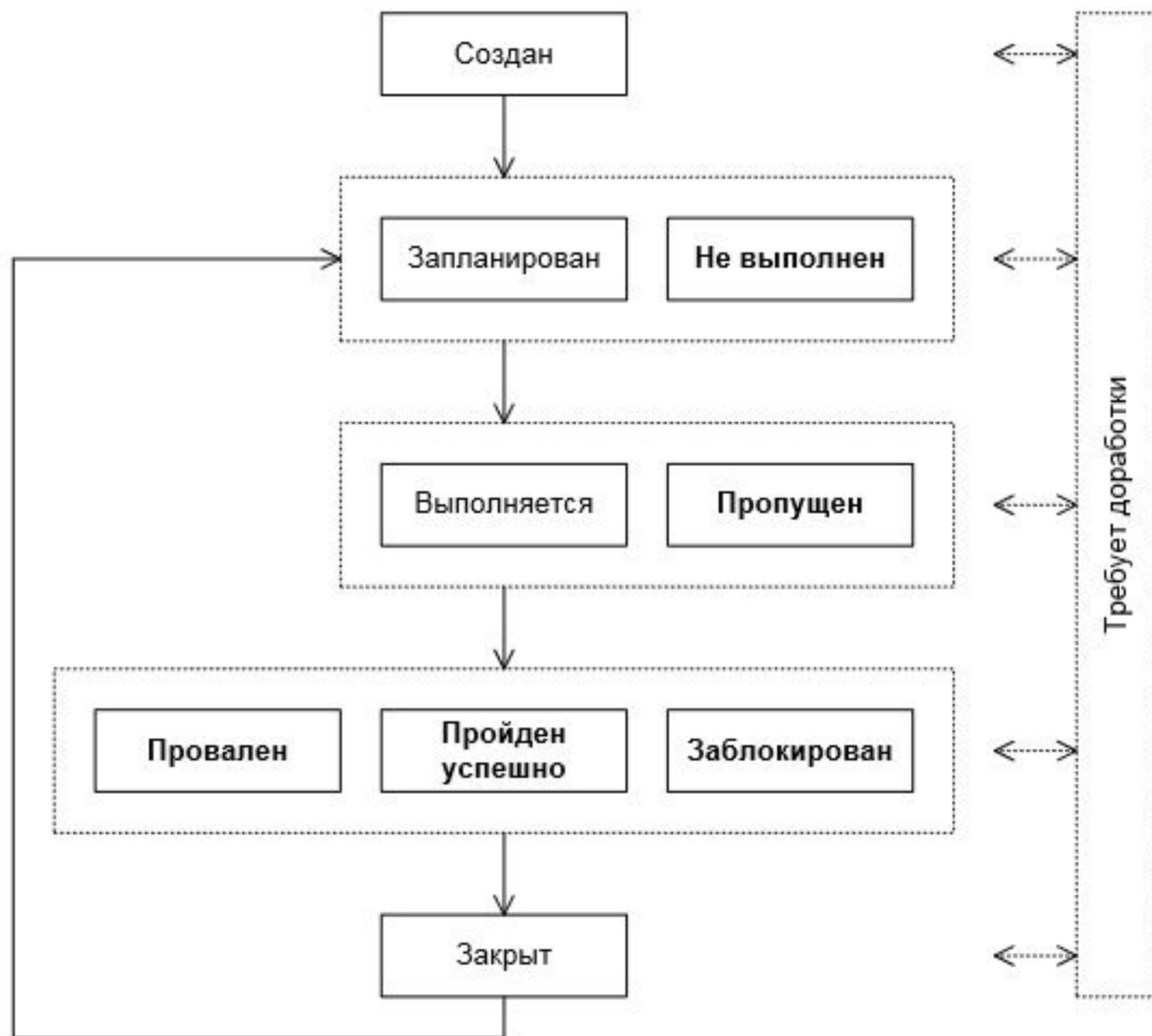


Создан (new) - типичное начальное состояние практически любого артефакта. Тест-кейс автоматически переходит в это состояние после создания.

Запланирован (planned, ready for testing) - в этом состоянии тест-кейс находится, когда он или явно включён в план ближайшей итерации тестирования, или, как минимум, готов для выполнения.

Не выполнен (not tested) - в некоторых системах управления тест-кейсами это состояние заменяет собой предыдущее («запланирован»). Нахождение тест-кейса в данном состоянии означает, что он готов к выполнению, но ещё не был выполнен.

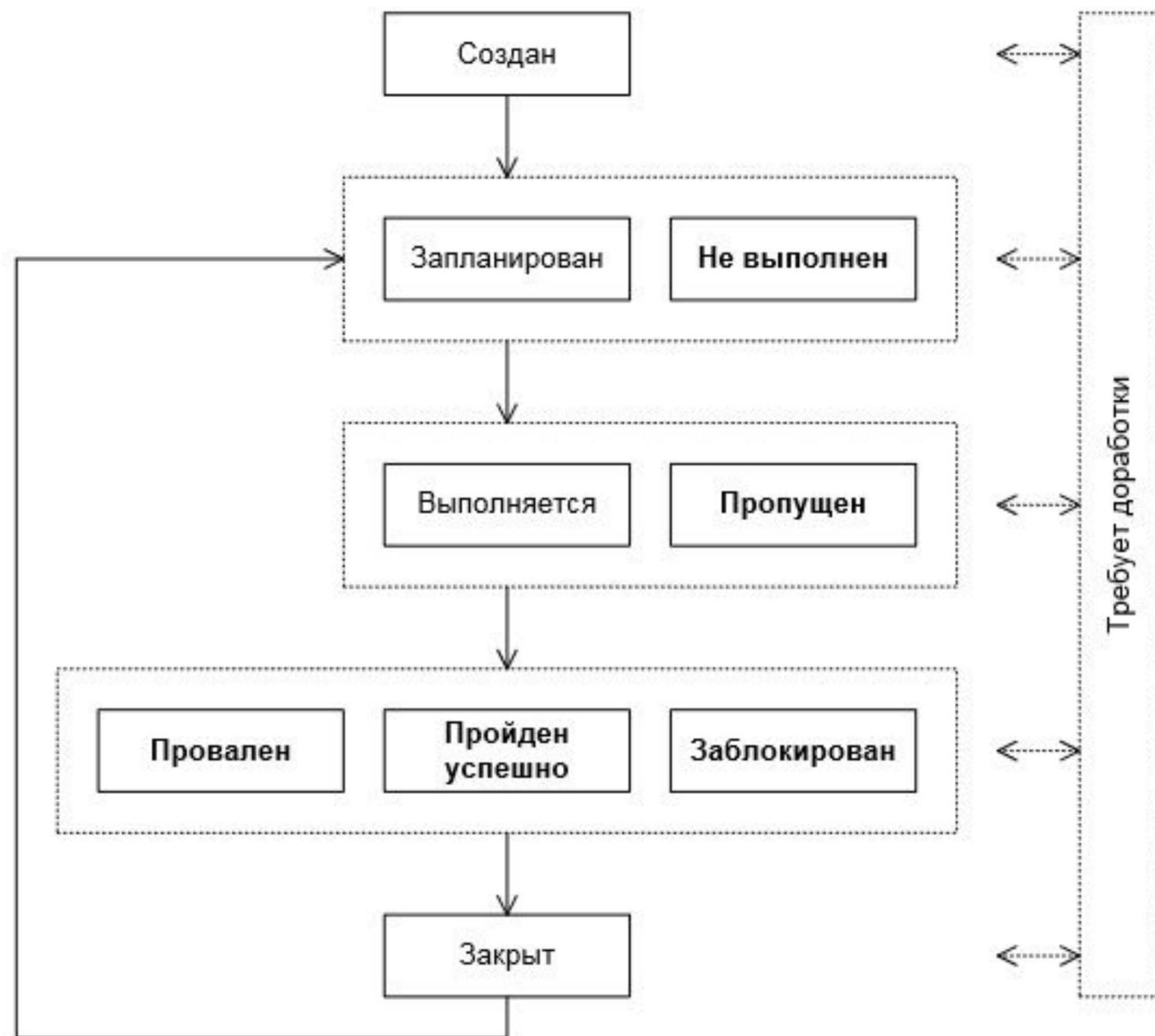
Жизненный цикл



Выполняется (work in progress) - если тест-кейс требует длительное время для выполнения, то он может быть переведён в это состояние для подчёркивания того факта, что работа идёт, и скоро можно ожидать её результатов. Если выполнение тест-кейса занимает мало времени, это состояние, как правило, пропускается, а тест-кейс сразу переводится в одно из трёх следующих состояний - «провален», «пройден успешно» или «заблокирован».

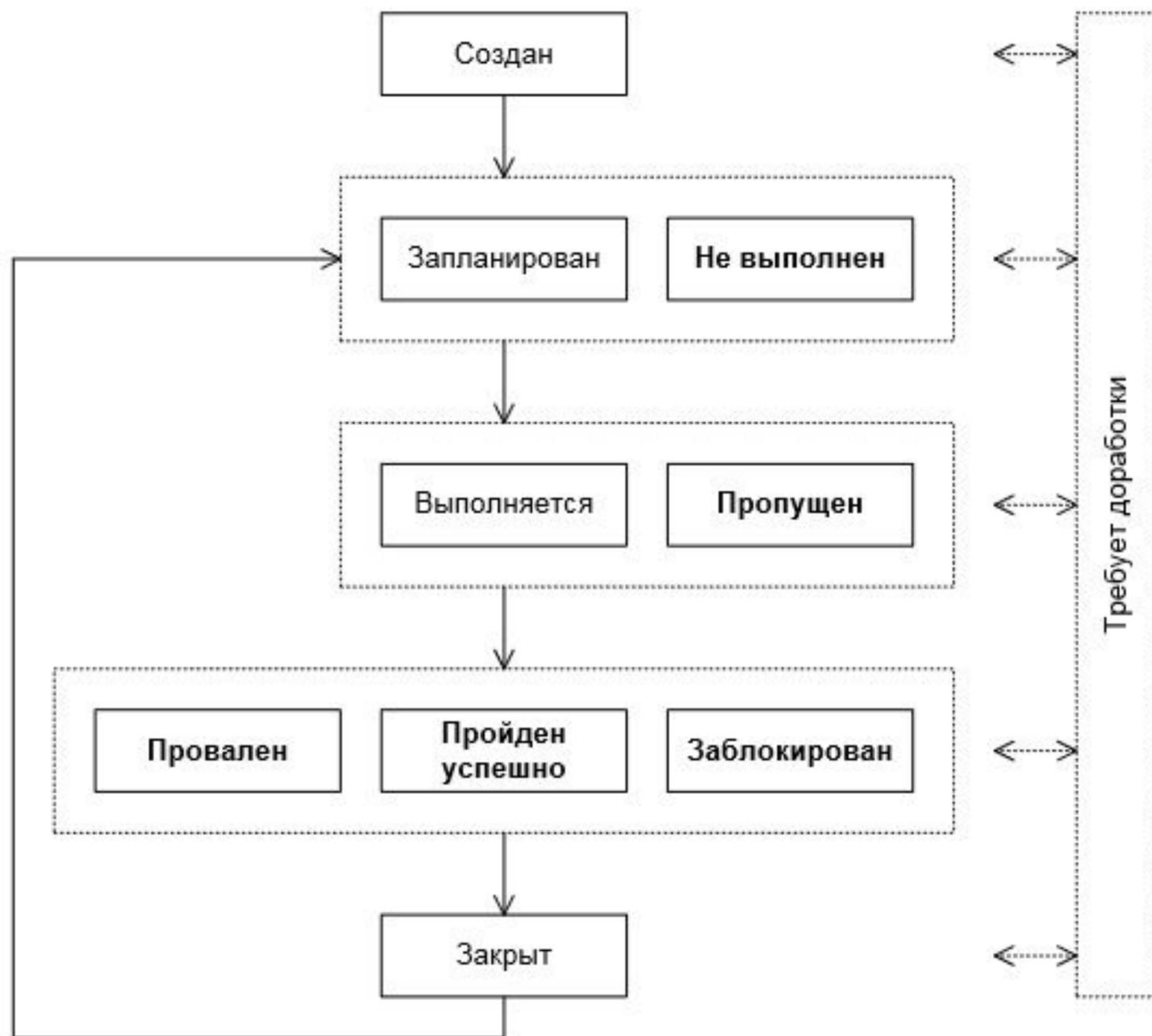
Пропущен (skipped) - бывают ситуации, когда выполнение тест-кейса отменяется по соображениям нехватки времени или изменения логики тестирования.

Жизненный цикл



Провален (failed) - данное состояние означает, что в процессе выполнения тест-кейса был обнаружен дефект, заключающийся в том, что ожидаемый результат по как минимум одному шагу тест-кейса не совпадает с фактическим результатом. Если в процессе выполнения тест-кейса был «случайно» обнаружен дефект, никак не связанный с шагами тест-кейса и их ожидаемыми результатами, тест-кейс считается пройденным успешно (при этом, естественно, по обнаруженному дефекту создаётся отчёт о дефекте).

Пройден успешно (passed) - данное состояние означает, что в процессе выполнения тест-кейса не было обнаружено дефектов, связанных с расхождением ожидаемых и фактических результатов его шагов.



Заблокирован (blocked) - данное состояние означает, что по какой-то причине выполнение тест-кейса невозможно.

Закрыт (closed) - очень редкий случай, т. к. тест-кейс, как правило, оставляют в состояниях «провален / пройден успешно / заблокирован / пропущен». В некоторых системах управления тест-кейс переводят в данное состояние, чтобы подчеркнуть тот факт, что на данной итерации тестирования все действия с ним завершены.

Требуется доработки (not ready) - как видно из схемы, в это состояние тест-кейс может быть переведён в любой момент времени, если в нём будет обнаружена ошибка, если изменятся требования, по которым он был написан, или наступит иная ситуация, не позволяющая считать тест-кейс пригодным для выполнения и перевода в иные состояния.

Идентификатор	Приоритет	Связанное с тест-кейсом требование	Заглавие (суть) тест-кейса	Ожидаемый результат по каждому шагу тест-кейса
UG_U1.12	A	R97	Галерея, загрузка файла	Галерея, загрузка файла, имя со спец-символами Приготовления: создать непустой файл с именем #\$\$%^&.jpg. 1. Нажать кнопку «Загрузить картинку». 2. Нажать кнопку «Выбрать». 3. Выбрать из списка подготовленный файл. 4. Нажать кнопку «ОК». 5. Нажать кнопку «Добавить в галерею».
Модуль и подмодуль приложения	Исходные данные, необходимые для выполнения тест-кейса	Шаги тест-кейса		1. Появляется окно загрузки картинки. 2. Появляется диалоговое окно браузера выбора файла для загрузки. 3. Имя выбранного файла появляется в поле «Файл». 4. Диалоговое окно файла закрывается, в поле «Файл» появляется полное имя файла. 5. Выбранный файл появляется в списке файлов галереи.

Идентификатор (identifier) - представляет собой уникальное значение, позволяющее однозначно отличить один тест-кейс от другого и используемое во всевозможных ссылках.

Приоритет (priority) - показывает важность тест-кейса. Он может быть выражен буквами (A, B, C, D, E), цифрами (1, 2, 3, 4, 5), словами («крайне высокий», «высокий», «средний», «низкий», «крайне низкий») или иным удобным способом. Количество градаций также не фиксировано, но, чаще всего, лежит в диапазоне от трёх до пяти.

Структура тест-кейса

Идентификатор	Приоритет	Связанное с тест-кейсом требование	Заглавие (суть) тест-кейса	Ожидаемый результат по каждому шагу тест-кейса
UG_U1.12	A	R97	Галерея, загрузка файла	Галерея, загрузка файла, имя со спец-символами Приготовления: создать непустой файл с именем #\$\$%^&.jpg. 1. Нажать кнопку «Загрузить картинку». 2. Нажать кнопку «Выбрать». 3. Выбрать из списка подготовленный файл. 4. Нажать кнопку «ОК». 5. Нажать кнопку «Добавить в галерею».
Модуль и подмодуль приложения			Исходные данные, необходимые для выполнения тест-кейса	Шаги тест-кейса

Связанное с тест-кейсом требование (requirement) показывает то основное требование, проверке выполнения которого посвящён тест-кейс (основное, поскольку один тест-кейс может затрагивать несколько требований).

Модуль и подмодуль приложения (module and submodule) указывают на части приложения, к которым относится тест-кейс, и позволяют лучше понять его цель.

Заглавие (суть) тест-кейса (title) призвано упростить и ускорить понимание основной идеи (цели) тест-кейса без обращения к его остальным атрибутам.

Структура тест-кейса

Идентификатор	Приоритет	Связанное с тест-кейсом требование	Заглавие (суть) тест-кейса	Ожидаемый результат по каждому шагу тест-кейса
UG_U1.12	A	R97	Галерея, загрузка файла	Галерея, загрузка файла, имя со спец-символами Приготовления: создать непустой файл с именем #\$\$%^&.jpg. 1. Нажать кнопку «Загрузить картинку». 2. Нажать кнопку «Выбрать». 3. Выбрать из списка подготовленный файл. 4. Нажать кнопку «ОК». 5. Нажать кнопку «Добавить в галерею».
Модуль и подмодуль приложения			Исходные данные, необходимые для выполнения тест-кейса	Шаги тест-кейса
				<ol style="list-style-type: none"> 1. Появляется окно загрузки картинки. 2. Появляется диалоговое окно браузера выбора файла для загрузки. 3. Имя выбранного файла появляется в поле «Файл». 4. Диалоговое окно файла закрывается, в поле «Файл» появляется полное имя файла. 5. Выбранный файл появляется в списке файлов галереи.

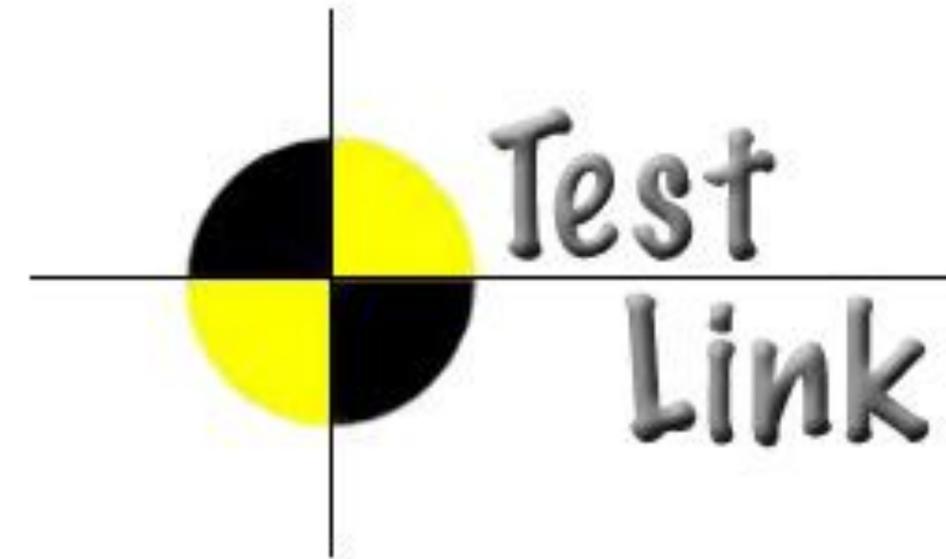
Исходные данные, необходимые для выполнения тест-кейса (precondition, preparation, initial data, setup), позволяют описать всё то, что должно быть подготовлено до начала выполнения тест-кейса.

Шаги тест-кейса (steps) описывают последовательность действий, которые необходимо реализовать в процессе выполнения тест-кейса.

Ожидаемые результаты (expected results) по каждому шагу тест-кейса описывают реакцию приложения на действия, описанные в поле «шаги тест-кейса». Номер шага соответствует номеру результата.

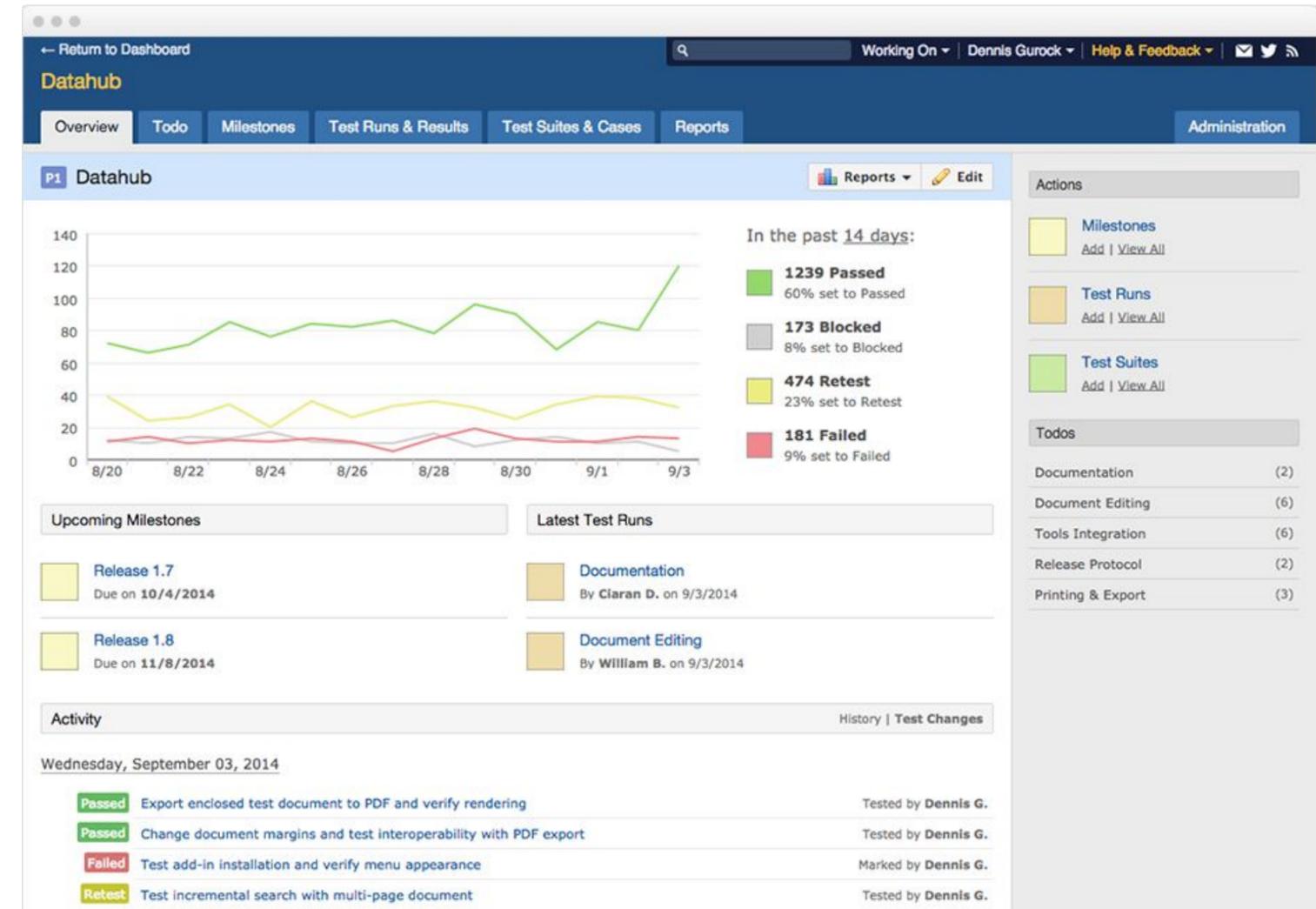
Инструменты для управления тест-кейсами:

- TestRail;
- TestLink;
- Jira + Zephyr.



TestRail является программным обеспечением для управления данными полученными в результате тестирования. Данный инструмент помогает отслеживать процессы, управлять программным обеспечением и организовывать команду.

С помощью TestRail можно создавать тест кейсы, управлять тестовыми наборами и координировать весь процесс тестирования программного обеспечения. TestRail предоставляет возможность повысить производительность и получить полный обзор хода процесса тестирования.



TestRail

1. Title (заглавие) здесь данное поле является обязательным для заполнения.
2. Section (секция) — очередная вариация на тему «Модуль» и «Подмодуль», позволяющая создавать иерархию секций, в которых можно размещать тест-кейсы.
3. Type (тип) здесь по умолчанию предлагает выбрать один из вариантов: automated (автоматизированный), functionality (проверка функциональности), performance (производительность), regression (регрессионный), usability (удобство использования), other (прочее).
4. Priority (приоритет) здесь представлен числами, по которым распределены следующие словесные описания: must test (обязательно выполнять), test if time (выполнять, если будет время), don't test (не выполнять).
5. Estimate (оценка) содержит оценку времени, которое необходимо затратить на выполнение тест-кейса.

The screenshot shows the 'Add Test Case' form in TestRail. It includes the following fields and sections:

- Title ***: A text input field (callout 1).
- Section ***: A dropdown menu (callout 2).
- Type ***: A dropdown menu (callout 3).
- Priority ***: A dropdown menu (callout 4).
- Estimate**: A text input field (callout 5).
- Milestone**: A dropdown menu (callout 6).
- References**: A text input field (callout 7).
- Preconditions**: A large text area (callout 8).
- Steps**: A list of steps, each with a 'Step Description' field (callout 9) and an 'Expected Result' field (callout 10).

At the bottom, there are buttons for 'Add Test Case' (with a green checkmark) and 'Cancel' (with a red X). An 'Add Step' button is also visible at the bottom right.

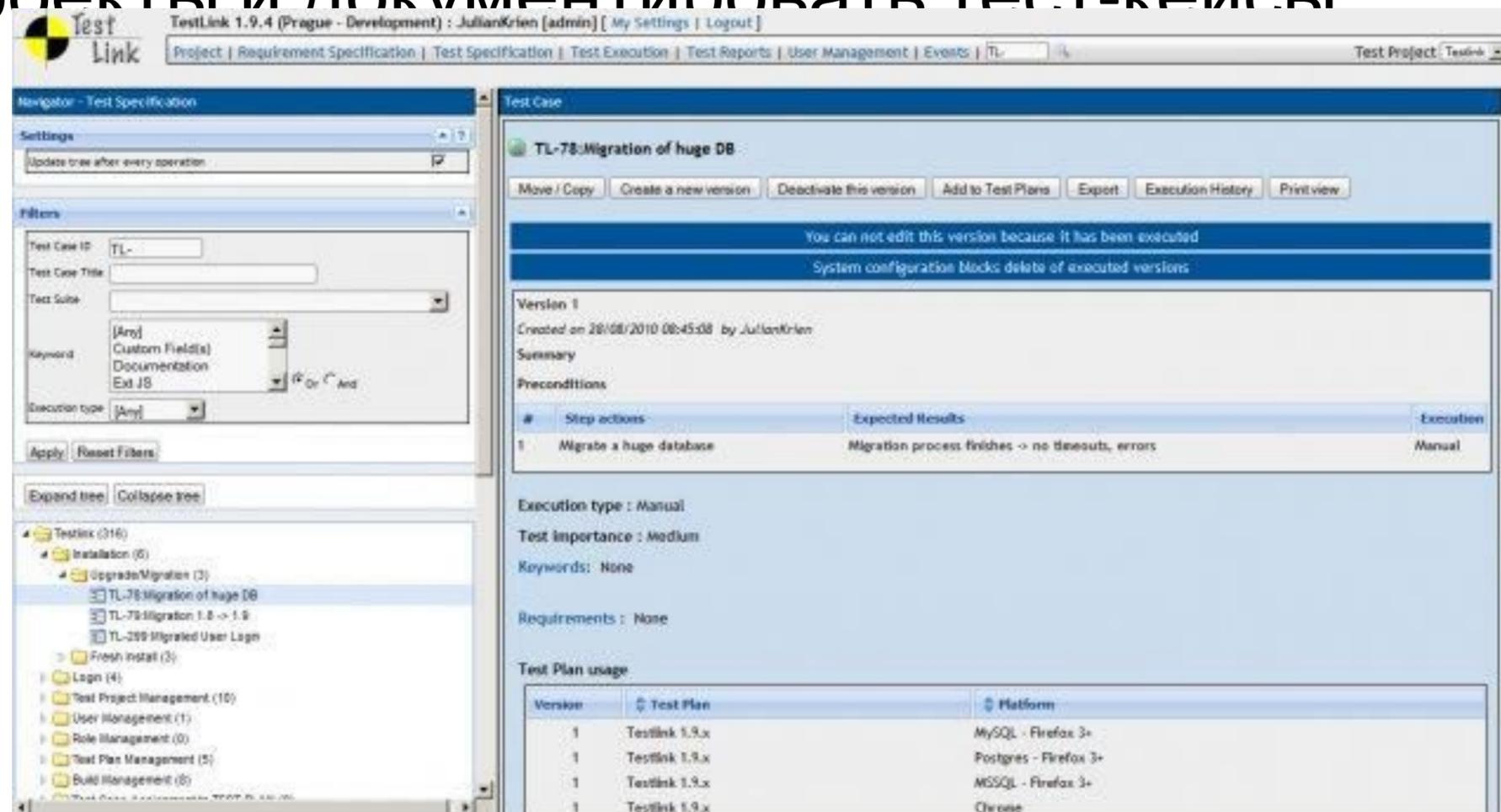
6. Milestone (ключевая точка) позволяет указать ключевую точку проекта, к которой данный тест-кейс должен устойчиво показывать положительный результат (выполняться успешно).
7. References (ссылки) позволяет хранить ссылки на такие артефакты, как требования, пользовательские истории, отчёты о дефектах и иные документы (требует дополнительной настройки).
8. Preconditions (приготовления) представляет собой классику описания предварительных условий и необходимых приготовлений к выполнению тест-кейса.
9. Step Description (описание шага) позволяет добавлять описание отдельного шага тест-кейса.
10. Expected Results (ожидаемые результаты) позволяет описать ожидаемый результат по каждому шагу.

The screenshot shows the 'Add Test Case' form in TestRail. The form is divided into several sections, each with a numbered callout:

- 1:** Title field.
- 2:** Section dropdown menu.
- 3:** Type dropdown menu.
- 4:** Priority dropdown menu.
- 5:** Estimate input field.
- 6:** Milestone dropdown menu.
- 7:** References input field.
- 8:** Preconditions text area.
- 9:** Step Description input field for the first step.
- 10:** Expected Result input field for the first step.

At the bottom of the form, there are two buttons: 'Add Test Case' (with a green checkmark) and 'Cancel' (with a red X). There is also a small 'Add Step' button at the bottom right.

TestLink – это веб-продукт с открытым кодом, который синхронизирует спецификацию требований и тестовую спецификацию. Данный инструмент тест-менеджмента был разработан компанией Teamtest. Языком создания TestLink является PHP. Благодаря TestLink, специалисты могут создавать тестовые проекты и документировать тест-кейсы



The screenshot displays the TestLink 1.9.4 (Prague - Development) web interface. The browser address bar shows the URL: `TestLink 1.9.4 (Prague - Development) : JulianKrien [admin] [My Settings | Logout]`. The page title is "Test Case". The main content area shows the configuration for test case "TL-78: Migration of huge DB".

Key elements visible in the interface include:

- Navigator - Test Specification:** A sidebar on the left with filters for Test Case ID, Title, Sub, Keyword, and Execution type. A tree view shows the project structure, including "Testlink (316)", "Installation (6)", "Upgrade/Migration (3)", and "Fresh install (3)".
- Test Case Details:** The main area shows the test case title "TL-78: Migration of huge DB", a message "You can not edit this version because it has been executed", and a table of execution steps.
- Execution Step Table:**

#	Step actions	Expected Results	Execution
1	Migrate a huge database	Migration process finishes -> no timeouts, errors	Manual

Additional details shown include "Execution type: Manual", "Test importance: Medium", "Keywords: None", and "Requirements: None". At the bottom, a "Test Plan usage" table is visible:

Version	Test Plan	Platform
1	Testlink 1.9.x	MySQL - Firefox 3+
1	Testlink 1.9.x	Postgres - Firefox 3+
1	Testlink 1.9.x	MSSQL - Firefox 3+
1	Testlink 1.9.x	Chrome

TestLink

1. Title (заглавие) здесь тоже является обязательным для заполнения.
2. Summary (описание) позволяет добавить любую полезную информацию о тест-кейсе (включая особенности выполнения, приготовления и т.д.).
3. Steps (шаги выполнения) позволяет описать шаги выполнения.
4. Expected Results (ожидаемые результаты) позволяет описать ожидаемые результаты, относящиеся к шагам выполнения.
5. Available Keywords (доступные ключевые слова) содержит список ключевых слов, которые можно проассоциировать с тест-кейсом для упрощения классификации и поиска тест-кейсов. Это ещё одна вариация идеи «Модулей» и «Подмодулей» (в некоторых системах реализованы оба механизма).
6. Assigned Keywords (назначенные ключевые слова) содержит список ключевых слов, проассоциированных с тест-кейсом.

The screenshot shows the 'Create Test Case' form in TestLink. It is divided into several sections:

- Test Case Title:** A text input field with a callout '1' pointing to it.
- Summary:** A rich text editor with a toolbar and a callout '2' pointing to the main text area.
- Steps:** A rich text editor with a toolbar and a callout '3' pointing to the main text area.
- Expected Results:** A rich text editor with a toolbar and a callout '4' pointing to the main text area.
- Keywords:** A section with two list boxes:
 - Available Keywords:** A list box with a callout '5' pointing to it.
 - Assigned Keywords:** A list box with a callout '6' pointing to it.

There are 'Create' buttons in the top right and bottom right corners of the form.

JIRA — это, главным образом, средство отслеживания ошибок, целью которого является контроль процесса разработки с задачами, ошибками и другими типами гибких карт.

Zephyr — один из многих плагинов JIRA, расширяющих возможности JIRA.

С помощью их комбинации Вы получите полный сервис, в соответствии с функциональностью инструментов управления тестированием:

- Создание тест-плана.
- Описание тестовых случаев.
- Выполнение тестирования.
- Создание отчетов.

Если тестовый продукт ведет себя неправильно, вы можете немедленно создать отчет о ошибке

Появились вопросы? Самое время задать их!





Спасибо за
внимание!