

Введение в тестирование
Урок 2

Создание чек-листов





Оглавление

На этом уроке	4
Что такое чек-лист?	4
Составление чек-листов	4
Последовательность действий	4
Правила составления	5
Оформление чек-листов	6
Варианты колонок в чек-листе	7
Простейший	7
Простейший с комментарием	7
С тестовыми данными	8
Проверки с детализацией по тестовым данным	8
Проверки с детализацией по версиям	8
Проверки с детализацией по окружениям	9
Виды чек-листов	9
Специальные и универсальные	9
Связанные и несвязанные	10
Правила выполнения чек-листа	12
Резолюция	13
Преимущества и недостатки чек-листов	13
Распространенные ошибки при составлении чек-листов	14
Программы для составления и хранения чек-листов	15
Google-таблицы	15
TMS-системы	16
RMS-системы	16
Демо	16

Урок 2. Создание чек-листов



Контрольные вопросы

20

Дополнительные материалы

20



На этом уроке

1. Узнаем, что такое чек-лист и по каким правилам его составлять и оформлять.
2. Познакомимся с видами чек-листов.
3. Ознакомимся с правилами выполнения чек-листа.
4. Перечислим преимущества и недостатки работы с чек-листами.
5. Обсудим распространенные ошибки при составлении чек-листов.
6. Перечислим программы, где можно составлять и хранить чек-листы.

Что такое чек-лист?

Чтобы не забыть купить продукты в магазине, мы составляем список покупок, а затем вычёркиваем или отмечаем галочкой то, что положили в корзину.

Такие списки сложно использовать людям, которые не знают о наших предпочтениях. Например, мы предпочитаем кокосовое молоко, но в списке записываем его кратко — «молоко». Если попросим кого-то другого зайти в магазин и купить продукты по списку, он может взять не кокосовое молоко, а козье. Номинально, он тоже купил молоко, но, не имея сведений о наших предпочтениях, не то, которое нужно.

На проекте всё аналогично. Список проверок (чек-лист) не детализирован, он только «напоминает», что нужно проверить. А как проверять вы решаете исходя из знания продукта и своего опыта.

Чек-лист (Check List) — это список, содержащий ряд проверок во время тестирования программного продукта.

Составление чек-листов

Чек-листы составляют на основе технического задания (ТЗ) — одного или нескольких документов, определяющих цель, структуру, свойства и методы проекта.

Например, при готовке пирога мы знакомимся с ТЗ — рецептом. Затем составляем чек-лист последовательности действий, чтобы не упустить важные моменты в готовке, идём по списку и в конце делаем вывод о готовности.



Также и тестировщик после знакомства с ТЗ составляет чек-лист, то есть список проверок программного обеспечения. Затем команда или один тестировщик проводят проверки и отмечают исполненные (пройденные) пункты чек-листа подобно тому, как мы вычеркиваем купленные по списку товары.

Последовательность действий

Сперва тестировщик знакомится с требованиями и только потом составляет по этим требованиям чек-лист.

Иногда студенты предполагают, что на момент составления чек-листа у тестировщика есть готовый продукт. Это не так — обычно тестировщики составляют тестовую документацию (то есть чек-листы) на основе технического задания.

Если нужно, требования делят на атомарные части, то есть декомпозируют, разбивают на небольшие составляющие. Это нужно, чтобы с ними было удобнее работать.

Последовательность действий при составлении чек-листа:

1. Определить все возможные проверки.
2. Сгруппировать проверки в разделы.
3. Определить тестовые данные, если они необходимы.
4. Подумать, какие колонки будут в чек-листе.

Затем составляется сам чек-лист.

Правила составления

1. Один пункт — одна операция.

Пункты чек-листа — это однозначные атомарные (выполняющее одно действие) и полные (логически завершённые) операции. Например, авторизация в профиль пользователя и редактирование данных пользователя — две разные задачи. В списке проверок их оформляют отдельными пунктами.

2. Пункты всегда начинаются с существительного или глагола неопределённой формы.

При составлении пунктов важно придерживаться унифицированных форм: начинать их с существительного («Проверка», «Добавление», «Отправка») или



неопределённой формы глагола («Проверить», «Добавить», «Отправить»). Так пункты будут понятными и однозначными.

3. Соблюдайте структуру.

Чтобы выполнять чек-лист было удобно, лучше составлять тесты так, чтобы они исходили из логики использования функционала. Например:

- в рамках раздела «Регистрация и Личный профиль» — регистрация на сайте, редактирование профиля;
- в рамках раздела «Форма обратной связи» — валидация полей, отправка письма, доставка письма.

4. Ничего не придумывайте.

Чек-лист важно составлять, опираясь на требования. Не придумывайте то, чего нет в требовании.

5. Общепринятые названия.

Хорошо, когда в команде все разговаривают на одном языке, используют общую терминологию.

Если один тестировщик в команде называет линию, разделяющую элементы на странице, дивайдером, а остальные — сепаратором, может возникнуть недопонимание.

Оформление чек-листов

Разберём пример чек-листа на проекте:



Профиль пользователя		
Карточка товара пользователя		
1	Изображение товара	passed
2	Каунтер позиции товара в слайдере	passed
3	Кнопка "Продать быстрее"	passed
4	Кнопка "Опубликовать в Одноклассниках"	failed
5	Кнопка "Снять товар с публикации"	failed
6	Отображение подключенного продвижения	passed
7	Доставка товаров	passed
8	Стоимость товара	passed
9	Название товара	passed
10	Описание товара	passed
...		
Настройки профиля пользователя		
1	...	
...		

Есть экран товара пользователя. Пользователь может перейти на карточку своего товара из профиля. На карточке есть слайдер (галерея с изображениями товара), кнопки продвижения (поднятия товара в выдаче), шеринга товара (возможность поделиться в соцсетях), а также цена, название и описание товара.

Логично определить, в каком модуле будет находиться карточка товара, затем определить название подмодуля — карточка товара. После идут сами проверки.

Важно! Напротив каждой из проверок есть статусы. Тестировщик ставит их только после выполнения каждого из пунктов чек-листа. Про них поговорим несколько позже.

Варианты колонок в чек-листе

В зависимости от того, что принято на проекте и как удобнее команде, есть несколько видов оформления чек-листов.

Простейший

В чек-листе две колонки: действие и результат — пройдено (passed) или не пройдено (failed). Исполнитель самостоятельно подбирает тестовые данные. Их необязательно фиксировать в чек-листе. Преимущества — быстрое составление.



Действие	Результат
Зарегистрироваться по номеру телефона	Ок
Зарегистрироваться по email	Ошибка
...	...

Простейший с комментарием

Похож на простейший, но с дополнительной колонкой для комментария. В ней указываются примечания, ссылки на созданные дефекты и прочая полезная информация.

Действие	Результат	Комментарий
Зарегистрироваться по номеру телефона	Ок	
Зарегистрироваться по email	Ошибка	Ошибка 500 Internal server error
...

С тестовыми данными

Кроме действия и результата в чек-листе указано, на каких тестовых данных будет проводиться тестирование.

Действие	Тестовые данные	Результат
Зарегистрироваться по номеру телефона	+79091234567	Ок
Зарегистрироваться по email	test@test.com	Ошибка
...		...

Проверки с детализацией по тестовым данным

Тестовые данные, на которых тестировщик выполняет проверку, могут указываться в столбце «Действие».

Действие	Результат	Комментарий
----------	-----------	-------------



Зарегистрироваться по номеру телефона		
• начинается +7, 10 цифр	Ок	Пользователь зарегистрирован
• начинается +7, 8 цифр	Ок	Ошибка «Некорректный номер»
...

Проверки с детализацией по версиям

В чек-листе хранится история прохождения для разных версий ПО. Это полезно для отслеживания: в одних местах чаще возникают ошибки, а в других всё относительно стабильно.

Действие	v. 1.0	v. 1.1	...
Зарегистрироваться по номеру телефона	Ок	Ок	...
Зарегистрироваться по email	Ок	Ошибка	...
...

Проверки с детализацией по окружениям

Задача тестирования — проверять работу ПО в разных условиях, то есть с разными окружениями. Веб-приложения должны работать без ошибок во всех браузерах, а мобильные — на всех устройствах и разных операционных системах.

Действие	Chrome	Firefox	...
Зарегистрироваться по номеру телефона	Ок	Ок	...
Зарегистрироваться по email	Ок	Ошибка	...
...

Виды чек-листов



Специальные и универсальные

Специальные чек-листы создают и используют для конкретных проектов. Их пункты соответствуют его специфике. По специальному чек-листу тестировщик проверяет возможность выполнить уникальное действие, предусмотренное требованиями.

Иными словами, проверить маркетплейсы Ozon, Wildberries, Яндекс.Маркет специальным чек-листом не получится из-за разного функционала.

Примеры пунктов специального чек-листа:

1. Профиль пользователя открывается в отдельной вкладке.
2. Название страницы «Профиль пользователя».
3. Отображается аватарка пользователя.
4. Присутствует обводка вокруг аватарки.

Универсальные чек-листы подходят для тестирования проектов одного типа.

Проверка по универсальному чек-листу не привязывается к графическим элементам или конкретной реализации. Проверяется сама возможность выполнить действие.

Для универсального чек-листа составляется абстрактный список проверок.

Например, универсальный чек-лист подходит для проверки общего функционала маркетплейсов Ozon, Wildberries, Яндекс.Маркет — «профиля» и «кнопки покупки товара»

Примеры пунктов универсального чек-листа:

1. Переход в профиль пользователя.
2. Отображается аватарка пользователя.
3. Отображается имя пользователя.

Также выделяют связанные, несвязанные чек-листы и подробный перечень.

Связанные и несвязанные

Пункты чек-листа могут быть в разной степени связаны друг с другом. В зависимости от этого выделяются три типа чек-листов:



1. Подробный перечень.
2. Связанный чек-лист.
3. Несвязанный чек-лист.

Подробный перечень (англ. laundry list) упрощает понимание и восприятие информации. Элементы списка группируются по категориям, порядок не влияет на точность данных.

По сути, главная ценность подробного списка — верно подобранные категории, по которым происходит группировка. Его можно сравнить с сортировкой белья разных цветов при стирке. Альтернативный перевод laundry list — список белья, отданный в стирку.

Пример подробного перечня для проверки сайта:

1. Вёрстка.
 - a. Вёрстка не «едет» во всех утверждённых в ТЗ браузерах.
 - b. Работают динамические элементы, нет явных ошибок отображения.
 - c. Нет горизонтального скролла.
 - d. Прочее.
2. Содержимое.
 - a. Нет орфографических и грамматических ошибок.
 - b. Удалены тексты заглушки.
 - c. На всех страницах есть контент.
 - d. Есть политика конфиденциальности и подтверждения согласия на обработку персональных данных от пользователей при сборе любых данных.
 - e. Прочее.
3. Функциональность.
 - a. Нет битых ссылок.
 - b. Все страницы сайта доступны из главной навигации.
 - c. Ссылка на открытую страницу подсвечивается в меню.
 - d. Прочее.
4. Безопасность и быстродействие.
 - a. Нет визуальных проблем с быстродействием.
 - b. Установлены последние версии используемых библиотек.



- c. Настроено резервное копирование.
- d. У посторонних пользователей нет доступа к закрытым файлам.
- e. Прочее.

Для проверки сайта не важно, будет ли сначала проверяться вёрстка, а потом содержимое, или наоборот. Важнее сгруппировать проверки таким образом, чтобы:

- одну группу проверок выполнял один человек;
- была возможность распараллелить проверки (взять разные проверки разными тестировщиками), чтобы повысить скорость работы.

В связанном чек-листе (англ. strongly sequential checklist) последовательность влияет на результат. Ключевой этап работы над созданием связанного чек-листа — корректное размещение информации.

Пример — предполётная контрольная таблица, которую использует персонал авиакомпаний. Её задача — определить, можно ли эксплуатировать конкретный самолёт. В ней важна строгая последовательность характеристик для адекватной оценки машины.

Пример связанного чек-листа для проверки поведения окна программы.

1. Новое окно открывается при двойном клике на ярлык.
2. Окно сворачивается в панель задач при нажатии кнопки «Свернуть».
3. Окно разворачивается из панели задач при нажатии на ярлык в панели задач.
4. Окно закрывается при нажатии кнопки «Заккрыть».

Пункты 2 и 3 нельзя поменять местами. Нельзя развернуть открытое окно, а на свёрнутом окне не будет кнопки «Заккрыть».

В несвязанном чек-листе (англ. weakly sequential checklist) последовательность размещения данных играет скорее психологическую роль, нежели логическую.

Несвязанный чек-лист — среднее между подробным перечнем и связанным чек-листом. Связь между элементами есть, но изменение порядка не повлияет на результат проверки.

Пример несвязанного чек-листа для тестирования социальной сети:

1. Пользователь может создать публикацию.
2. Пользователь видит публикации открытых аккаунтов, независимо от наличия подписки.



3. Пользователь видит публикации закрытых аккаунтов, на которые он подписан.
4. Пользователь не видит публикации закрытых аккаунтов, на которые он не подписан.
5. Пользователь может удалить публикацию.

Эти действия тестируются в любом порядке независимо друг от друга. Но психологически комфортнее идти именно в таком порядке.

Правила выполнения чек-листа

1. Не пропускать проверки чек-листа. Если какие-то пункты нельзя проверить из-за багов или недоступности среды, надо отметить их и вернуться к ним позже.
2. Если указываются тестовые данные, нужно использовать именно их.
3. Указывать результат и комментарий.

Резолюция

Когда исследуемый объект протестирован, нужно выставить кейсу резолюцию, то есть вердикт — прошёл он проверку или нет.

Рассмотрим распространённые статусы и разберёмся, какой из них когда используется:

- **passed** — успешно пройден
- **failed** — кейс не прошел проверку
- **skipped** — пропуск проверки
- **blocked** — проверка заблокирована
- **untested** — кейс еще не брали в работу

passed ставится, когда исследуемый объект проходит проверку в соответствии с ТЗ.

failed — исследуемый объект не соответствует ТЗ. Например, функционал сломан, не работает или его нет (разработчик потерял кнопку).

skipped — пропуск проверки кейса. Например, из-за нехватки времени или если в кейсе есть неточности: он устарел, из документации вы понимаете, что его уже изменили в проекте.



blocked — проверка заблокирована поломанным функционалом, от которого прямо или косвенно зависит тестируемый. Например, есть функционал «авторизация пользователя» и функционал «редактирование данных пользователя». Разработчик допустил ошибку в коде и функционал авторизации сломан — ему мы выносим резолюцию **failed**. Проверить редактирование данных пользователя не можем — выставляем статус **blocked**. В комментарии указываем, почему не можем протестировать этот функционал.

Преимущества и недостатки чек-листов

Преимущества:

1. Чек-листы затрагивают большее количество кейсов, так как при прохождении их можно по-разному выполнять.
2. Сокращают затраты на содержание и поддержку тестов.
3. Обеспечивают высокую скорость тестирования: не нужно отмечать результат каждого шага тестировщика, достаточно общего результата проверки.
4. Позволяют проходить и комбинировать тесты по-разному, в зависимости от предпочтений сотрудников.
5. Показывают статистику: кто, когда и что проходил — с детализацией по сборке продукта и окружению, на котором проводилось тестирование.
6. Улучшают представление о системе в целом, показывают статус её готовности.
7. Показывают объём проделанной и предстоящей работы по тестированию.

Недостатки:

1. Начинающие тестировщики не всегда эффективно проводят тесты без достаточно подробной документации.
2. Чек-листы невозможно использовать для обучения начинающих сотрудников, так как в них недостаточно подробных сведений.
3. Заказчику или руководству может быть недостаточно того уровня детализации, который предлагают чек-листы.
4. Неопределённость тестового набора: каждый тестировщик выполняет пункт чек-листа по-своему.



Распространённые ошибки при составлении чек-листов

✗ Не атомарные проверки

Неверно:

1. авторизоваться в профиле пользователя и изменить имя пользователя
2. редактирование номера телефона и почты

Верно:

1. авторизация по номеру телефона
2. переход в профиль пользователя
3. редактирование имени пользователя
4. редактирование номера телефона пользователя
5. редактирование почты пользователя

✗ Некорректная форма глагола

Неверно:

1. авторизовался в профиле пользователя
2. отредактировал номера телефона

Верно:

1. авторизация по номеру телефона
2. редактирование номера телефона

✗ Нет разделителя по модулям

Неверно:

1. авторизация по номер телефона
2. редактирование номера телефона
3. редактирование имени пользователя

Верно:

1. Авторизация
 - а. авторизация по номеру телефона
2. Профиль пользователя
 - а. редактирование номера телефона
 - б. редактирование имени пользователя



Программы для составления и хранения чек-листов

На уроке мы рассмотрели примеры чек-листов, составленных в Google-таблице. Но есть и другие инструменты.

Google-таблицы

Удобный инструмент: чек-листы наглядные, действия и результаты распределены по ячейкам. Всё просто и бесплатно. Но есть минусы:

- Google-таблицы могут быть запрещены политикой безопасности;
- Документы Excel неудобно модифицировать.

TMS-системы

Чек-листы можно составлять в таблицах Excel — на некоторых проектах так и делают. Но общепринято использовать специальные инструменты — Test Management System (TMS) или системы управления тестированием.

Популярные системы управления тестированием:

- Test Link
- Test IT
- Zephyr
- qTest
- PractiTest
- TestLodge
- TestRail
- Qase.io
- Tematoo
- Test Collab
- HP ALM
- Testuff
- XQual

Плюсы:

Минусы:



- удобное оформление тестовой документации
- структурированное отображение
- удобное выполнение кейсов
- наглядный отчёт после прохождения
- дополнительный инструмент в команде, который может требовать настройки
- большинство TMS — платные

RMS-системы

RMS (Requirements Management Systems) — это средства поддержки и автоматизации работы с требованиями на протяжении всего жизненного цикла разработки программного продукта. В них тоже можно составлять и хранить чек-листы, но это не очень удобно и затратно по времени.

Демо

Представим, что перед нами задача протестировать форму заявки на кредит по чек-листу. Разберёмся с последовательностью его выполнения.



ГАРАНТИРУЕМ БЕЗОПАСНОСТЬ ВАШИХ ДАННЫХ

Фамилия, имя и отчество

Укажите точно как в паспорте

Мобильный телефон
+7 (

Мы отправим решение по заявке на указанный номер

Электронная почта

На данный email будет направлена информация по заявке

☐ Я соглашаюсь с условиями и даю свое согласие на обработку и использование моих персональных данных, и разрешаю сделать запрос в бюро кредитных историй

ПОДАТЬ ЗАЯВКУ

Чек-лист:



		status	комментарий
Форма подачи заявки на кредит			
1	Отображение логотипа банка		
2	Наличие комментария про безопасность данных		
3	Поле ФИО		
4	Поле «Мобильный телефон»		
5	Поле «Электронная почта»		
6	Поле «Укажите желаемую сумму кредита»		
7	Комментарии под полями		
8	Чекбокс согласия с условиями и обработки данных		
9	Кнопка «Помощь»		
10	Кнопка «Подать заявку»		

Раздел, который нам предстоит протестировать — «Форма подачи заявки на кредит». В нём десять кейсов, которые нужно проверить.

Возьмём первую проверку — «Отображение логотипа банка». Нам нужно открыть форму подачи заявки на кредит, затем, на основе опыта и знаний о продукте, выполнить проверку.

Какие кейсы мы можем проверить, чтобы убедиться, что логотип отображается корректно? Давайте перечислим:

- дизайн логотипа (сверим с макетом);
- функционал (у логотипа есть гиперссылка, которая ведёт на главную страницу).

Убедившись, что мы всё проверили, можем поставить этому кейсу резолюцию (статус) **passed**. Поле «комментарий» не заполняем.

Теперь возьмём проверку №6: «Поле “Укажите желаемую сумму кредита”». Нам нужно вернуться на форму подачи заявки на кредит, затем, на основе опыта и знаний о продукте, выполнить проверку.



Чтобы убедиться, что поле «Укажите желаемую сумму кредита» отображается корректно, проверим кейсы:

- дизайн поля (сверим с макетом);
- позитивный кейс, что мы можем ввести необходимую нам сумму;
- проверим кейсы на границах (попытаемся взять кредит на сумму 0 и на сумму 10 000, если предположить, что она максимальная);
- проверим, что в поле есть плейсхолдер (подсказка) «Укажите сумму кредита»;
- проверим, что плейсхолдер скрывается, если поставить курсор в поле;
- проверим негативные кейсы: введём в поле отрицательную сумму, спецсимволы, буквы.

Убедившись, что мы всё проверили, данному кейсу могли бы поставить резолюцию (статус) **passed**. Но разработчик допустил ошибку — этого поля нет. Соответственно, мы не можем отметить, что кейс прошёл проверку и ставим статус **failed**. Поле «комментарий» заполняем. Добавляем в него:

- причину, по которой кейс не прошёл проверку (коротко);
- ссылку на заведённый баг-репорт (задокументированный найденный дефект, подробнее — в следующих лекциях).

Протестируем форму и заполним колонки «статус» и «комментарий».

		status	комментарий
Форма подачи заявки на кредит			
1	Отображение логотипа банка	passed	
2	Наличие комментария про безопасность данных	passed	
3	Поле ФИО	passed	
4	Поле «Мобильный телефон»	passed	
5	Поле «Электронная почта»	passed	
6	Поле «Укажите желаемую сумму кредита»	failed	данного поля нет
7	Комментарии под полями	passed	
8	Чекбокс согласия с условиями и обработки данных	passed	



9	Кнопка «Помощь»	failed	данной кнопки нет
10	Кнопка «Подать Заявку»	passed	

Контрольные вопросы

1. Что такое чек-лист и зачем он нужен?
2. Какой алгоритм составления чек-листов?
3. Каким правилам надо следовать, чтобы получился хороший чек-лист?
4. Какие есть виды чек-листов по последовательности проверок?
5. В каком случае проверке ставится статус passed? failed? skipped? blocked?
6. Какие плюсы у чек-листов?
7. Какие минусы у чек-листов?

Дополнительные материалы

1. [Структура, содержание и процесс написания проверок](#)
2. [Чек-лист тестирования WEB приложений](#)
3. [Чек-лист тестирования мобильных приложений](#)
4. [Чек-лист — как тестировать поиск](#)