

## Свойства качественного чек-листа

1. Чек-лист должен охватывать весь проверяемый элемент: его функционал, графику, текст, звуки и т.п.
2. Проверку элемента нужно логически разделить на отдельные, независимые друг от друга наборы тест-кейсов. Например:
  - Изменение параметров события с сервера.
    - i. Награда за квалификацию изменяется с сервера.
    - ii. Даты запуска события настраиваются с сервера.
  - Баланс наград.
    - i. Все виды наград начисляются игроку.
    - ii. Выдается 5 кэша за первый сундук в событии при победе.
3. Блоки и тест-кейсы в них необходимо расположить в порядке убывания их приоритета. Наиболее важным является проверка ключевой логики, так как ошибки в ней могут заблокировать или сломать весь элемент. В первую очередь выполняют те проверки, которые наиболее близки к действиям обычного игрока. Когда основная логика проверена, то приступают к проверке дополнительных возможностей элемента. После проверяют негативные сценарии.
4. Следует избегать избыточных, дублирующих друг друга тест-кейсов. Минимизировать их количество вам помогут техники тест-дизайна: классы эквивалентности, анализ граничных условий, матрица принятия решений. Информацию по техникам можно почитать у С. С. Куликов ["Тестирование программного обеспечения. Базовый курс"](#) или у Сэм Канер ["Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений"](#).
5. Тест-кейс должен не повторять требования, а проверять их.
6. Один пункт — одна проверка. Не нужно включать в пункт несколько разных проверок, иначе в ходе тестирования можно случайно пропустить одну из них.
7. Заголовки тест-кейсов нужно составлять в утвердительной форме.
  - Плохо: Приход нотификации.
  - Хорошо: Нотификация завершения приходит за 8 часов до окончания события.
8. В формулировке заголовка тест-кейса следует использовать принцип "Что? Где? Когда?".
  - Что происходит? — *"Событие запускается", "Игрок получает награду"*.
  - Где происходит? — *"В магазине декораций", "В матч3"*.
  - Когда происходит? — *"По нажатию кнопки", "После разворачивания"*.
9. Не стоит начинать проверки со слов "Проверить, что ...", "Убедиться, что ..." и их производных. Сюда же относится типичное слово "Попытаться" в негативных проверках. Чаще всего такие формулировки можно убрать без потери смысла. Например:
  - Плохо: Убедиться, что событие запускается после прохождения 19 уровня основного потока.

- Хорошо: Событие запускается после прохождения 19 уровня основного потока.
  - или*
  - Плохо: Попытаться ввести етоїі в поле имени игрока.
  - Хорошо: Ввести етоїі в поле имени игрока.
10. Не нужно использовать формулировки: штатно, корректно, некорректно, правильно, неправильно. Понятие “штатности” или “корректности” у каждого может ассоциироваться с разными вещами.
    - Плохо: Окно отображается корректно.
    - Хорошо: Окно не содержит графических артефактов.
  11. Не нужно использовать личную форму глагола в шагах для проверки в низкоуровневом тест-кейсе.
    - Плохо: Нажимаем кнопку.
    - Хорошо: Нажать кнопку.
  12. Нужно стараться избегать формулировок "Если ..., то ...", т.к. это вносит дополнительное условие, которое можно трактовать по-разному.
    - Плохо: Если игрок подключен к Facebook, то в окне отображается текст с призывом подключиться.
    - Хорошо: Отображается текст с призывом подключиться в окне при отключенном Facebook.
  13. Информация в тест-кейсе должна быть понятной и восприниматься однозначно. Если есть сомнения в том, что указанную информацию можно понять по-разному, то нужно подробно расписать её.
  14. В предусловии низкоуровневого тест-кейса указывается действия, которые необходимо совершить перед выполнением шагов кейса.
  15. Ожидаемый результат в низкоуровневом тест-кейсе указывается в настоящем времени. Также не нужно использовать слова “должен”, “должно быть” и их производных при описании результата.
  16. В заголовке и ожидаемом результате тест-кейса указывайте поведение системы так, чтобы исключить субъективное толкование.
  17. Если в тест-кейсе необходимо сделать запрос в канал на изменение параметров на сервере или сохранке, то следует написать пример такого запроса.
  18. Не должно быть зависимости тест-кейсов между собой. Тест-кейс всегда может быть удален и тогда связанный с ним кейс потеряет смысл.
  19. При написании тест-кейсов следует учитывать особенности мобильных платформ.
  20. В тексте не должно быть орфографических и пунктуационных ошибок.
  21. Ключевые слова, на которых необходимо заострить внимание, следует выделять **жирным**.
  22. Чек-лист должен выглядеть эстетично — тогда с ним будет приятно работать.
  23. Нужно стараться максимально перенести информацию из ТЗ и других необходимых документов в кейсы, чтобы не приходилось переходить по ссылкам в процессе проверки.
  1. В чек-лист обязательно включаем элементы, с которыми взаимодействует проверяемый объект. А именно:
    - a. tutorials;
    - b. сценки (достаточно 2-3 сценки);

- c. нотификации;
  - d. синхронизация между устройствами;
  - e. обновление;
  - f. изменение параметров с сервера;
  - g. нестабильный интернет (можно брать наработки из <https://app.asana.com/0/652997386710205/690760353609505>);
  - h. тип начисляемого кэша free/real при выдачи награды.
  - i. отображение и работа таймера
2. Большинство проверок не требует детализации и их стоит писать как высокоуровневые тест-кейсы. Проверки со сложным алгоритмом действий стоит оформить как низкоуровневый тест-кейс.
  3. В чек-лист включаем проверку событий **Swrve** только для нововведений.
  4. Для каждого пункта обязательно указываем: **Build Type** и **Platform**.
  5. При выставлении **Platform (iOS, Android, Amazon, UWP, Mac, Samsung, Huawei)** следует учитывать, что на **Android, Amazon, UWP, Mac, Samsung** и **Huawei** выполняется поверхностная проверка: проверяется работоспособность механики - достаточно пробежаться по всем её окнам, смотрится графика и функционал, который специфичен для данной платформы. Если кейс используется на нескольких платформах, то необходимо указать их все.
  6. Все кейсы по нотификациям для событий на **Android, Amazon, UWP, Mac, Samsung** и **Huawei** не смотрим, проверяем по паре нотификации на каждой указанной платформе.
  7. Поле **Приоритет** должно быть установлено в значение **Средний**.
  8. Поле **Статус** должно быть установлено значение **Готов**.