Это список проверок который надо выполнить чтобы убедится что наша программа работает корректно. Здесь нет подробностей как именно проверять, какие данные вводить, какой результат мы ожидаем. Это просто список того что надо проверить. Начинать составлять чек лист стоит с изучения требований. Но если этого нет то необходимо понимание как должен работать продукт.

Explanation #\_2 тестировщику приходится работать с огромным количеством информации, выбирать из множества вариантов решения задач и изобретать новые. В процессе этой деятельности объективно невозможно удержать в голове все мысли, а потому продумывание и разработку тесткейсов рекомендуется выполнять с использованием «чек-листов» Чек-лист (checklist) — это просто набор идей: идей по тестированию, идей по разработке, идей по планированию и управлению — любых идей (тут наверно любых идей с точки зрения тестирования)

Важно понять, что нет и не может быть никаких запретов и ограничений при разработке чек-листов — главное, чтобы они помогали в работе. Иногда чек-листы могут даже выражаться графически. Для того чтобы чек-лист был действительно полезным инструментом, он должен обладать рядом важных свойств.

Логичность. Чек-лист пишется не «просто так», а на основе целей и для того, чтобы помочь в достижении этих целей. К сожалению, одной из самых частых и опасных ошибок при составлении чек-листа является превращение его в свалку мыслей, которые никак не связаны друг с другом. Последовательность и структурированность. Со структурированностью всё достаточно просто — она достигается за счёт оформления чек-листа в виде многоуровневого списка. Что до последовательности, то даже в том случае, когда пункты чек-листа не описывают цепочку действий, человеку всё равно удобнее воспринимать информацию в виде неких небольших групп идей, переход между которыми является понятным и очевидным (например, сначала можно прописать идеи простых позитивных тест-кейсов{82} , потом идеи простых негативных тест-кейсов, потом постепенно повышать сложность тест-кейсов, но не стоит писать эти идеи вперемешку).

Полнота и неизбыточность. Чек-лист должен представлять собой аккуратную «сухую выжимку» идей, в которых нет дублирования (часто появляется из-за разных формулировок одной и той же идеи), и в то же время ничто важное не упущено.

Чтобы проиллюстрировать принципы построения чек-листов, мы воспользуемся логикой разбиения функций приложения по степени их важности на три категории

Базовые функции, без которых существование приложения теряет смысл (т.е. самые важные — то, ради чего приложение вообще создавалось), или нарушение работы которых создаёт объективные серьёзные проблемы для среды исполнения. (См. «Дымовое тестирование»{79} ).

Функции, востребованные большинством пользователей в их повседневной работе. (См. «Тестирование критического пути» ).

Остальные функции (разнообразные «мелочи», проблемы с которыми не сильно повлияют на ценность приложения для конечного пользователя). (См. «Расширенное тестирование»{81} ).

Пример написания чек-листа для чайника (требований нет, ожидается что вы сами понимаете как он работает ):

1) Кипятит воду.

2) Вместительность 2 литра

3) Включение - выключение

4) Открытие-закрытие крышки

5) Отключается после закипания

и это вроде основные (важные) проверки, но можно накинуть ещо)))

6) Включить чайник без воды

7) Налить больше воды чем можно по инструкции

8) За сколько закипает вода в чайнике

9) Выключение электричества во время работы чайника

10) Открыть крышку во время работы чайника

11) Включить с открытой крышкой

12) Уровень шум

13) Внешний вид

14) Удобно ли наливать/выливать туда/оттуда воду

15) Ударопрочность

16) Включить в мороз/жару

Вот эти проверки это и есть чек лист того как мы будем проверять чайник.

Когда требований нет (или они не полные) то надо смотреть на предмет тестирования как пользователь и исходя из этого состовлять чек лист.

по автору № 2

чек лист - это высокоуровневый набор идей тестов.

Пример:

