

**Головкова Елена Александровна,**  
к.т.н., доцент, Ангарский государственный технический университет,  
e-mail: temnikova\_ea@bk.ru

**Прокопенко Ирина Борисовна,**  
обучающаяся группы ЭН-22-1, Ангарский государственный технический университет  
**Игамбердиев Сардорбек Сухробжонович,**  
Таджикистан

## **СПОСОБЫ УПРОЩЕНИЯ НАПИСАНИЯ ТЕСТ-КЕЙСОВ**

**Golovkova E.A., Prokopenko I.B., Igamberdiev S.S.**

### **WAYS TO SIMPLIFY WRITING TEST CASES**

**Аннотация.** Сформулирована основная цель тестирования программного обеспечения, описаны основные способы корректного составления тест-кейсов, приведен пример создания интеллектуальной карты в приложении XMind.

**Ключевые слова:** тестирование программного приложения, чек-лист, тест-кейс, интеллектуальная карта.

**Abstract.** The main purpose of software testing is formulated, the main ways of correctly compiling test cases are described, and an example of creating an intelligent map in the XMind application is given.

**Keywords:** software application testing, checklist, test case, smart card.

Основной целью тестирования программного обеспечения (ПО) является обнаружение ошибок в работе программы (функциональных и логических), тестирование безопасности и надёжности.

Существует большое количество различных видов тестирования, как ручного, так и автоматизированного.

Для корректного тестирования необходимо составить не только его план, подобрать методы и инструменты тестирования, но и написать тест-кейс, который представляет чёткое описание действий – алгоритм, который необходимо выполнить с целью проверки программы по заданным критериям. Тест-кейс описывает предусловия, шаги и ожидаемый результат.

Зачастую написание тест-кейса является нетривиальной задачей, поскольку тестировщику необходимо понимать всю логику работы ПО, помнить, что уже описано, а что – нет, поэтому существуют способы упрощения данного процесса.

Первый способ – сформировать чек-лист, после составления которого тестировщик приступает к написанию тест-кейсов, помечая в чек-листе то, что он уже описал. Однако, даже в чек-листах тестировщик может упустить этапы, которые должны быть учтены в тест-кейсе, из-за сложности и многофункциональности ПО.

Второй способ, на взгляд авторов, наиболее оптимальный – это построение интеллектуальной карты (mind map).

Mind map, описывая структуру программы (отдельного элемента или

модуля программы, функционала и т.д.), ветвится – представляет собой иерархическое дерево привычное и удобное для восприятия «пользователя». Поэтому, создавая интеллектуальную карту приложения, необходимо разбивать его на блоки, описывая их в виде ветвлений.

Рассмотрим простой пример, взяв за основу текстовый редактор «Блокнот», представленный на рисунке 1.

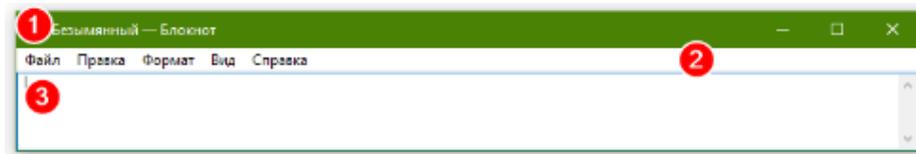


Рисунок 1 – Интерфейс тестового редактора «Блокнот»

На рисунке 2 представлен фрагмент интеллектуальной карты, построенный для элементов текстового редактора, отмеченных цифрами.



Рисунок 2 – Фрагмент mind map для тестового редактора «Блокнот»

Из рисунка 2 видно, как начинает ветвиться функционал приложения. У каждого ветвления есть метка: знак минуса – ветвление продолжается; зелёная галочка – конечная проверка. Благодаря этим меткам понятно все ли шаги выполнены или необходимо продолжить проверку (тестирование) программы.

После создания интеллектуальной карты можно приступить к составлению чек-листов, а потом тест-кейсов.

Благодаря интеллектуальным картам скорость создания чек-листов и тест-кейсов многократно увеличивается и упрощается. Популярным и удобным инструментом в этой области является русскоязычное приложение XMind.

## ЛИТЕРАТУРА

1. Темникова, Е. А. Процесс тестирования программного обеспечения в команде, работающей по SCRUM / Е. А. Темникова, К. И. Лопатина – Текст : непосредственный // Информационные технологии и проблемы математического моделирования сложных систем. – 2017. – С. 52-56.