

АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ПРИКЛАДНЫХ РЕШЕНИЙ НА ПЛАТФОРМЕ 1С:ПРЕДПРИЯТИЕ

Макареня С.Н.

Международный институт дистанционного образования Белорусского национального технического университета, Минск, Республика Беларусь, makar_sn@mail.ru

В проектах по разработке программного обеспечения (ПО) помимо основной задачи по реализации заявленной функциональности существует не менее важная задача по обеспечению качества ПО. Качество ПО (Software quality) – это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности. Одним из устоявшихся способов контроля качества является тестирование. Тестирование ПО (Software testing) — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. [1]

В данной статье рассмотрим один из подходов к автоматизированному тестированию прикладных решений на платформе 1С:Предприятие.

Конфигурация «Система проектирования прикладных решений» (СППР) предназначена для автоматизации проектирования конфигураций на платформе 1С:Предприятие версии 8.3 и ведения проектной документации. Система проектирования прикладных решений предоставляет механизм автоматизированного тестирования при разработке прикладных конфигураций 1С:Предприятие с использованием сценариев работы пользователя и соответствующих тестов. [2]

СППР поддерживает хранение тестовых сценариев двух видов:

- простые сценарии представляют собой простую последовательность линейных шагов, где под шагом понимается любая операция, которая эмулирует действия пользователя. Это может быть нажатие на кнопку, заполнение поля, формирование отчета и другие действия;
- процессы – это более сложный сценарий в виде последовательности простых тестов.

Для написания сценариев используется модификация языка Gherkin, которая разработана специально для использования с платформой 1С:Предприятие. Это декларативный, легко воспринимаемый человеком язык, который имеет определенный синтаксис.

Обобщенная модель жизненного цикла тестового сценария представлена следующими этапами [3]:

- создание теста – текст сценария формируется вручную, или производится запись действий пользователя в тестируемой конфигурации;
- проверка и отладка теста – проверяется правильность выполнения сценария теста и его корректировка в случае обнаружения некорректного исполнения;
- встраивание теста в прогон автоматических тестов на регулярной основе – добавление корректного теста в набор тестов для проведения регрессионного теста и проведение тестирования. Регрессионное тестирование представляет собой набор всех видов тестирования программного обеспечения, направленных на обнаружение ошибок в уже протестированных участках исходного кода;
- рефакторинг теста – выполняются анализ и изменение сценария теста в связи с изменением функциональности тестируемого приложения;
- удаление теста из инфраструктуры – выполняется в том случае, когда тест теряет свою актуальность и не может быть далее использован для проведения регрессионного тестирования.

Для написания автоматических тестов в СППР используется фреймворк Vanessa-Automation, который представляет собой инструмент, направленный на уменьшение затрат на поддержку и повышение надежности использования тестов. Этот инструмент позволяет создавать и запускать тестовые сценарии.

Тестовые сценарии пишутся в соответствии с подходом к разработке и тестированию, при котором сначала создаются тесты, которым должен удовлетворять код, и только потом выполняется реализация кода. Это так называемая разработка через тестирование [4].

Для написания сценариев тестирования используется язык Turbo Gherkin, который является расширенной версией языка Gherkin (расширен синтаксис языка). Именно эта версия языка Gherkin используется для работы с фреймворком Vanessa-Automation (для упрощения в дальнейшем будем говорить о языке Gherkin).

Этот язык позволяет решить сразу две задачи:

- автоматизировать процесс тестирования;
- составлять проектную документацию.

Использование языка Gherkin удобно по ряду причин:

- сценарии, написанные на этом языке, определяют поведение системы;
- сценарии написаны в простой форме (ориентированы на человека) и могут быть понятны всем участникам проекта;
- файлы, содержащие в себе сценарии, одновременно являются и исполняемыми автоматическими тестами;
- тестовая документация и программный код автоматических тестов хранятся в одном проекте и неотделимы друг от друга;
- наличие словаря доступных шагов позволяет расширять возможности тестировщиков по написанию различных вариантов сценариев, не обращаясь к программному коду.

Текст сценария состоит из шагов. Любой шаг начинается с ключевого слова. Список ключевых слов: «И», «Когда», «Тогда», «Затем», «Дано», «Допустим», «Если», «Пусть», «То», «К тому же», «Также», «Но», «А». Модификация языка Gherkin поддерживает условия и циклы.

Порядок выполнения действий по формированию тестового сценария с записью действий пользователя выглядит в общем случае следующим образом:

1. Выполнить в СППР на форме сценария команду «Выполнить - Запуск фреймворка тестирования» с данным сценарием. При этом откроется фреймворк Vanessa-Automation.
2. Открыть закладку «Test Clients», выбрать тестируемую базу и подключиться к ней.
3. Перейти на закладку «Работа с UI» (Работа с пользовательским интерфейсом) и начать запись действий пользователя в выбранном клиенте тестирования по команде «Начать запись». Если тестовый клиент был закрыт, то он откроется. Если тестовый клиент был открыт, то произойдет повторное подключение к нему.
4. На тестовом клиенте выполнить необходимые пользовательские действия.
5. По команде «Завершить запись» действий пользователя на закладке «Работа с UI» завершить тест в тестовом клиенте. Все действия пользователя записываются в формате XML (закладка Исходный XML). При этом на закладке Сценарий поведения появится текст сценария тестирования на языке Gherkin.
6. Скопировать текст полученного сценария и вставить на закладке «Текст сценария» формы созданного сценария работы пользователей (рисунок 1.).
7. Заполнить в сценарии группу, описывающую содержание выполняемых действий (после *), и отформатировать текст сценария по команде «Отформатировать».
8. Записать сценарий по команде «Записать».

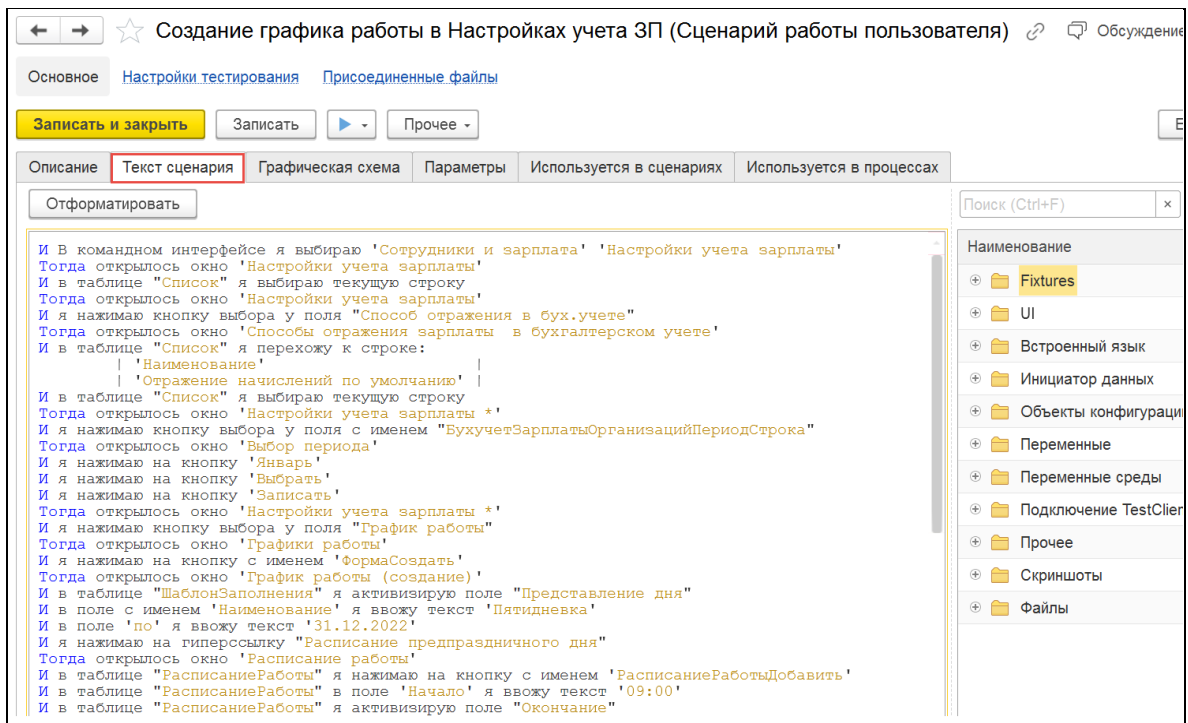


Рисунок 1 – Сценарий работы пользователя на языке Gherkin.

При этом запись в оперативную память завершается, тестовый клиент получает из памяти сформированный массив информации и формирует сценарий в формате XML. XML-файл передается в обработку Vanessa-Automation и преобразуется в сценарий на языке Gherkin на закладке «Работа с UI» в окне тестов.

Далее текст теста переносится через буфер обмена операционной системы в информационную базу СППР, где выполняется связывание с тестом метаданных, ошибок, ответственных, функциональных моделей и другой информации технического характера.

В завершение необходимо выполнить команду «Подготовить сценарий к выполнению» на закладке «Работа с UI».

Перед использованием обработки Vanessa-Automation в конфигураторе базы СППР необходимо настроить параметры базы (Сервис – Параметры): на закладке «Запуск 1С:Предприятие-Дополнительно» в группе Автоматическое тестирование перевести переключатель в состояние Запускать как менеджер тестирования.

Другой вариант настройки – в списке информационных баз для выбранной базы добавить ключ запуска /TESTMANAGER.

Также необходимо подготовить эталонные базы. Справочник Эталонные базы для тестирования (раздел Тестирование) используется для хранения параметров подключения к эталонным базам тестирования.

Любой сценарий можно запустить вручную в справочнике Сценарии работы пользователей по команде «Запуск тестов» (в группе «Выполнить» на форме списка или элемента справочника).

Запуск сценария также может выполняться CI-сервером, который будет запускать сценарий в установленное время (например, ночью). CI (Continuous Integration – «непрерывная интеграция») – это практика разработки программного обеспечения, которая заключается в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем.

Для управления версиями кода конфигурации используется распределенная система управления GIT. Функции CI-сервера и GIT-сервера выполняет система управления хранилищем (репозиторием) кода GIT Lab.

Чтобы выполнить запуск сценария CI-сервером, необходимо выбранный сценарий выгрузить (сериализовать) в текстовый формат в формате yaml (команда «Выгрузить сценарии» на форме «Настройки тестирования сценариев»).

Выполняется загрузка выгруженного сценария в каталог на диске для использования в GIT Lab. Далее выгруженный сценарий используется CI-сервером для запуска сборок и тестирования конфигурации.

Если при выполнении теста конфигурации был получен негативный результат, то CI-сервер результаты теста вместе со скриншотами, текстами ошибки и другой технической информацией (артефактами) анализирует на предмет наличия дублей, выполняет их свертку и данные по ошибкам передает в СППР, которая автоматически регистрирует ошибку.

При автоматическом выполнении тестирования в случае негативного завершения теста информация об ошибке записывается в справочник Ошибки (раздел Главное) с помощью регламентного задания «Загрузить ошибки из файлов».

В этом случае на форме ошибки поле «Метод выявления» имеет значение «Автотесты».

Для минимизации времени анализа инцидентов СППР предоставляет пользователям комплект файлов с технической информацией, которые могут помочь быстро разобраться с причиной ошибки и устранить ее. Доступ к этой информации выполняется по гиперссылке «Присоединенные файлы» на форме справочника Ошибки.

К ошибке прикладываются:

- база, на которой запускался сценарий (dt-файл);
- файлы, содержащие системную информацию работы сервера или компьютера, в которые заносятся определенные действия;
- скриншот ошибки – позволяет наглядно увидеть проявление ошибки;
- дополнительные файлы.

Пользователь имеет возможность выполнить переход на форме ошибки к строке сценария, которая привела к ошибочной ситуации.

Результатом исполнения сценариев в Vanessa-ADD являются:

1. Отчетность в различных форматах: junit, Cucumber, Allure.
2. Скриншоты, либо каждого окна, либо скриншот снятый в момент возникновения ошибки. Скриншоты могут являться частью отчетности и выполнении сценариев.
3. Автоматическое формирование «черновики» пользовательских инструкций в форматах HTML, Markdown.

Наиболее информативной и полезной для целей тестирования является отчетность в формате Allure.

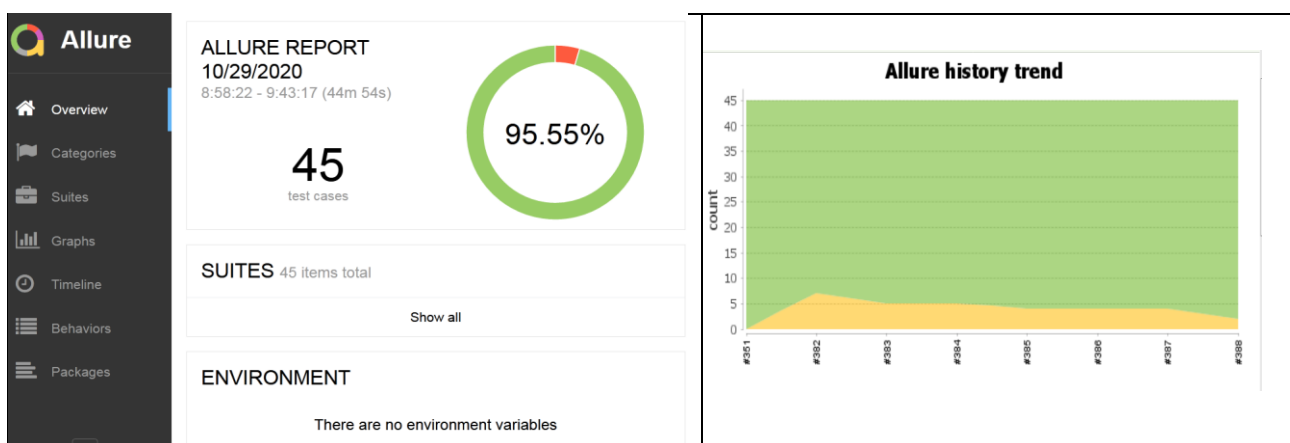


Рисунок 2 – Отчет о результатах автоматического тестирования в формате Allure.

Тестирование – это обязательный этап при модификации ПО. Игнорирование данного этапа увеличивает расходы при исправлении ошибок после доработки и ввода в эксплуатацию протестированных или плохо протестированных изменений ПО. Платформа

1С:Предприятие предусматривает механизмы автоматизированного тестирования и при верном подборе специалистов по тестированию качество полученного результата повышается и максимально приближается к требованиям заказчика.

Список литературы:

1. Кулаков К.А., Димитров В. М. Основы тестирования программного обеспечения. Учебное электронное пособие для обучающихся Института математики и информационных технологий. / Петрозаводск: ПетрГУ, 2018.
2. 1С:Предприятие 8. Конфигурация «Система проектирования прикладных решений» Редакция 2.0. Руководство пользователя. /М: Фирма «1С», 2019.
3. Шаев В.В. Тестирование компонентов и комплексов программ. Учебник. /– М.: СИНТЕГ, 2010.
4. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. / М. ЛОРИ. 2003.