

информации.

Исследование поддержано проектом CERES. Centers of Excellence for young REsearchers (Reg.no. 544137-TEMPUS-1-2013-SK-JPHES),



Список использованных источников:

1. Документация по языку Python. [Электронный ресурс] – Режим доступа: <https://docs.python.org/3/> . – Дата доступа: 03.04.2017.
2. Документация по средствам Qt. [Электронный ресурс] – Режим доступа: <http://doc.qt.io/qt-5/json.html>. – Дата доступа: 03.04.2017.

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ С ПРИМЕНЕНИЕМ BDD ПОДХОДА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Жданко Т.В.

Матвейчук Н. М. – канд. физ.-мат. наук, доцент

Тестирование является составляющей частью разработки программного обеспечения. Возрастающее количество исходного кода, а также возрастающее число разработчиков, вызывает все большее количество ошибок в разрабатываемом приложении. Одной из основных функций тестирования является обнаружение этих ошибок как можно раньше. Скорость, с которой можно идентифицировать эти ошибки, полностью зависит от подхода тестирования, количества тестировщиков, их опыта, а также инструментов, используемых в процессе тестирования. Эффективная автоматизация тестирования является одной из наиболее важных составляющих быстрой и непрерывной поставки высококачественного программного обеспечения.

Существуют различные подходы к автоматизации тестирования программного обеспечения. Мы часто слышим такие аббревиатуры, как TDD и BDD. Подход разработки через тестирование (TDD, англ. TestDrivenDevelopment) заключается в написании модульных, интеграционных и функциональных тестов для разрабатываемого приложения, перед написанием самого кода приложения. Т.е. сперва разрабатываются тесты, проверяющие корректность работы еще не реализованного программного кода. Разумеется, этот тест не проходит. Затем разрабатывается код, который выполняет действия, требуемые для успешного прохождения теста.

BDD(англ. BehaviourDrivenDevelopment) подход заключается в описании поведения системы с точки зрения эксперта предметной области, а не программиста. Здесь рассматривается не результат выполнения какого-либо модуля, а та работа, которую он выполняет. Этот принцип рассматривают как продолжение TDD.

Одним из инструментов программной поддержки BDD является Cucumber. Тесты пишутся с помощью сценариев, которые понятны любому пользователю. В сценариях используются следующие элементы BDD синтаксиса:

- Given - условие;
- When - событие;
- Then - результат.

Cucumber интерпретирует тесты на используемом языке программирования и использует Selenium WebDriver для управления тестами в браузере.

На рисунке 1 представлен простой сценарий авторизации пользователя.

```
@run
Feature: Test Login

Scenario Outline: Login Success and Failure
  Given I navigate to the mock application
  When I try to login with '<type>' credentials
  Then I should see that I logged in '<status>'

Examples:
| type      | status      |
| valid     | successfully |
| invalid   | unsuccessfully |
```

Рисунок 1 – Листинг тестового сценария

Как мы видим, Cucumber может использовать один и тот же тест многократно, используя различные входные данные, которые представлены в таблице. В файлах со сценариями используются теги (@run), с

помощью которых, Cucumber читает соответствующие файлы и запускает тесты.

Для реализации тестовой логики созданных сценариев создается класс, где каждый метод имеет аннотацию @Given, @When или @Then, которая содержит регулярное выражение, соответствующее строке (шагу) созданного тестового сценария [2].

Основные преимущества Cucumber:

- 1) Новые тесты разрабатываются с высокой скоростью. Используя реализованные шаги уже созданных тестов, даже начинающие тестировщики могут быстро и качественно создать множество подобных сценариев, при этом не написав нового кода.
- 2) Готовые тестовые сценарии легко читабельны и понятны всем – от инженера по тестированию до заказчиков.
- 3) Ненадобность логирования при написании тестов – каждый шаг (действие пользователя) по сути своей является логированием.
- 4) Необходимые шаги для описания дефектов берутся из отчета [3].

Список использованных источников:

1. Habrahabr [Электронный ресурс]. – Электронные данныею –Режим доступа: <https://habrahabr.ru/post/160257/>.
2. Quizful [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.quizful.net/test/test-driven-development>.
3. Quora [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.quora.com/What-are-the-disadvantages-of-BDD>.

РЕШЕНИЕ ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ ФРЕДГОЛЬМА С ПОМОЩЬЮ MAPLE

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Козлова А.А.

Калугина М.А.– канд. физ.-мат. наук, доцент

Теория интегральных уравнений занимает важное место в современной математике и ее приложениях. В частности, практическое значение уравнений, которые были исследованы еще Фредгольмом и дали рождение целому разделу функционального анализа, и сейчас трудно переоценить. Аналитическое решение даже частных случаев этих уравнений является сложной математической проблемой. Были разработаны алгоритмы и их Maple-версии для получения аналитического и/или приближенного решений большого класса уравнений Фредгольма I и II рода. В докладе приведена сравнительная характеристика работы трех из этих алгоритмов относительно точного решения на примере одного уравнения.

“Интегральными уравнениями называются уравнения, содержащие неизвестную функцию под знаком интеграла” [1]. Если в уравнении использован интеграл Фредгольма [2]

$$\int_b^a K(x, t)\varphi(t)dt,$$

то различают два основных типа этих уравнений: уравнения I рода

$$\int_a^b K(x, t)\varphi(t)dt = f(x)$$

и уравнения II рода

$$\varphi(x) - \lambda \int_a^b K(x, t)\varphi(t)dt = f(x).$$

Интегральные уравнения встречаются в различных областях науки и их приложениях [2]. Уравнения Фредгольма первого рода являются типичными при математической обработке экспериментальных данных, восстановлении размытого изображения и т.д. К однородным интегральным уравнениям Фредгольма второго рода приводят задачи о собственных колебаниях систем.

Аналитическое решение даже частных случаев этих уравнений является сложной математической проблемой, а встроенные функции математических систем часто не могут осилить не берущиеся аналитически интегралы. Поэтому на практике используют приближенные и численные методы решения. Это методы последовательных приближений, определителей Фредгольма, решения уравнений с вырожденным ядром, квадратур, резольвент и т.д.[2, 4]. Таким образом, вычислительные подходы к решению интегральных уравнений представляют собой целое направление научного исследования.