

# ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

УДК 159.93

## ТЕСТИРУЮЩАЯ СИСТЕМА С ИСПОЛЬЗОВАНИЕМ КОНСТРУКТОРА ТЕСТОВ

**С.Б. Тарасов, Ю.С. Павлюткин**

В статье рассматриваются вопросы разработки тестирующей системы с использованием конструктора тестов. Определены основные требования для разрабатываемой системы. Выработаны предложения по ее реализации. Даны рекомендации по использованию конструктора тестов.

---

The issues of testing system development with test designer are under consideration. The basic system requirements, stages of development process and recommendation of test designer practical application are the results of this research.

---

### **КЛЮЧЕВЫЕ СЛОВА**

Тестирование, конструктор, анализ, оценка, тип, информационные технологии, web сервер, интернет.

### **1. ВВЕДЕНИЕ**

В новых вузовских программах увеличивается число часов выделяемых на самостоятельную работу студентов. Поэтому контроль качества усвоения учебного материала становится весьма актуальной проблемой.

Традиционные методы опроса во время проведения аудиторных занятий не позволяют полностью решить указанную проблему. Одним из эффективных средств ее решения является активное использование тестирующих систем. Хотя тестирование появилось давно, его бурное развитие приходится на последние десятилетия, что связано с развитием новых информационных технологий. Большинство современных разработок поддерживают режим онлайн тестирования, который является одним из эффективных и простых методов проверки контроля знаний.

Многие системы тестирования имеют конструкторы тестов, которые позволяют достаточно просто включать новые тесты, сводя весь процесс создания теста к его описанию. После ввода описания, тестирующая система автоматически настраивается на описание без необходимости внесения изменений в программу.

Конструктор тестов, как правило, предоставляет широкие возможности. В частности, в нём предусматривается создание закрытых (с выбором одного или нескольких вариантов ответа) и открытых (со свободным вводом ответа) типов тестирования, задание количество вопросов и ответов, задание времени тестирования, возможность выбора одного или нескольких ответов.

Однако, как правило, такие системы являются платными, не предоставляют открытый код и включают избыточный набор функций. В этом случае более целесообразным представляется разработка новой системы, которая была бы лишена этих недостатков.

На основании анализа существующих разработок были определены наиболее важные требования к конструктору тестов:

- конструктор тестов должен представлять собой Интернет-ресурс. Сама тестирующая программа и результаты тестирования хранятся на сервере;
- конструктор тестов должен иметь простой и удобный интерфейс доступный широкому кругу пользователей;
- конструктор тестов допускает использование открытых и закрытых тестов;
- конструктор позволяет создавать как психологические, так и общеобразовательные тесты для проведения интернет-тестирования в учебном заведении или для индивидуальной самостоятельной подготовки;
- в конструкторе тестов должна быть возможность установки количества вопросов в тесте.
- в конструкторе тестов должна быть возможность задания количества ответов на вопрос - одиночный или множественный выбор;
- при формировании теста можно задавать время, отводимое на тестирование;
- по результатам тестирования можно выводить распределение итоговых баллов и сохранять результаты тестирования в базе данных.

Для удобства эксплуатации системы, контроля выполнения тестирования были включены следующие категории пользователей:

**Администратор** - поддерживает функционирование системы, выявляет и исправляет ошибки. В функции администратора входит добавление прав доступа пользователям, добавление информации об обновлениях системы, работа с базой данных.

**Эксперт** - имеет возможность создавать, изменять, удалять тесты. Эксперту доступна функция получения результатов по каждому отдельно взятому вопросу из созданного им теста. Выбрав конкретного пользователя, эксперт может получить данные по весовым баллам и по времени, затраченному на ответ. Эти данные могут помочь эксперту оценить ответы тестируемого, качество созданного им теста и в случае необходимости внести корректировки в тест.

**Тестируемый** - эта категория пользователей, которая выполняет уже созданные и доступные для прохождения тесты. По завершению тестирования, в личном кабинете тестируемого появляются результаты прохождения.

Исходя из требований к каждой группе пользователей, в системе должны быть реализованы следующие возможности:

- управление системой администратором;
- выполнение регистрации и авторизации пользователей;
- разделение прав доступа между группами пользователей;
- реализация обмена сообщениями, для получения обратной связи между пользователем, экспертом и администратором

С помощью диаграммы языка UML приведено представление целевых групп и их роли в функционировании системы тестирования (рис.1).

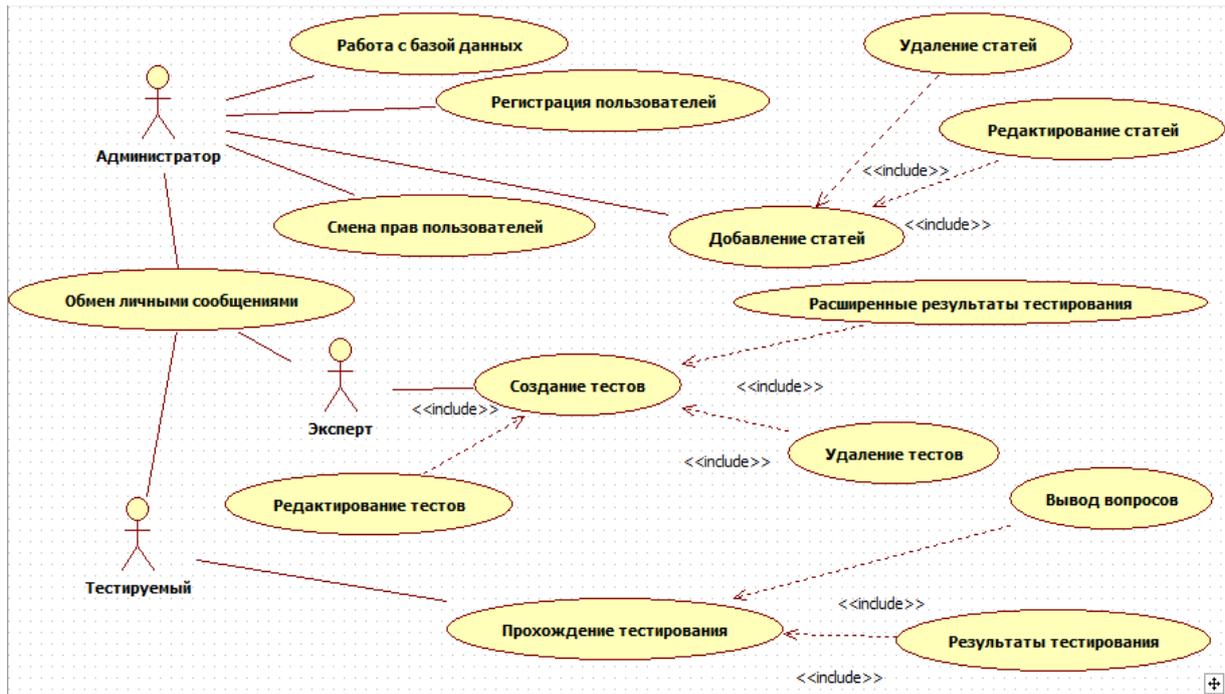
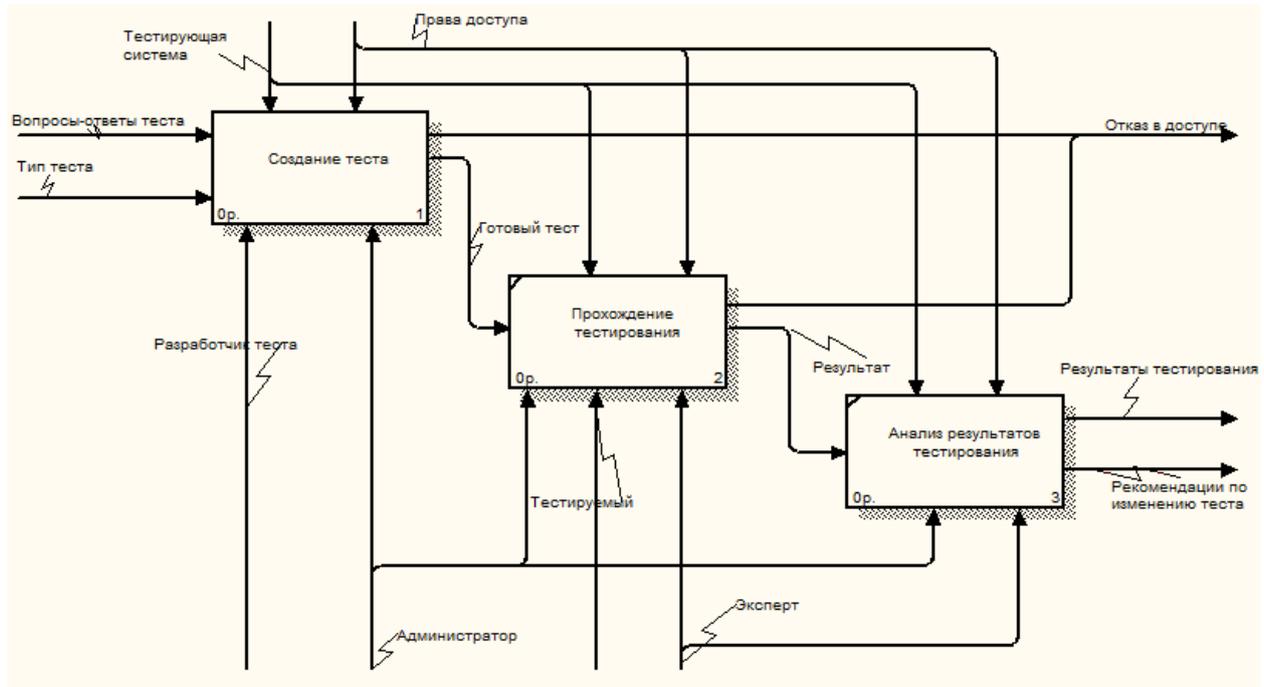


Рис.1. Целевые группы.

## 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

На этапе проектирования были уточнены характеристики системы, режимы ее функционирования. Была разработана детализированная функциональная модель с использованием методологий IDEF0 и DFD.

Разработка такой смешанной модели позволила использовать преимущества каждой методологии: с помощью методологии IDEF0 создается графическая модель всех взаимодействующих в системе процессов, с помощью методологии DFD создается модель с описанием внешних источников данных и доступа к ним по отношению к системе в целом. В качестве инструментального средства создания моделей использовался пакет BPWIN. На рис.2 представлен подуровень IDEF0 диаграммы.



**Рис.2. Подуровень IDEF0 диаграммы.**

На диаграмме видно, какие ресурсы необходимы для функционирования тестирующей системы. На вход поступают вопросы и ответы теста, тип тестирования. Тип тестирования задается на этапе создания теста, также как и количество вопросов и ответов теста. Управляющими данными будут являться тестирующая система и права доступа. Механизмы поступают снизу. Ими являются эксперт, администратор и тестируемый. У каждого свои права доступа. Так как права пользователей разделены, то не каждому доступен тот или иной этап. Только администратор в праве полностью контролировать работоспособность ресурса. Для эксперта доступны функции создания тестирования и получения результатов тестирования, а обычному пользователю предоставляется возможность проходить доступные задания и получать свои результаты. На выходе получаем отказ в доступе, если не хватает прав или произошла ошибка. Если ошибок не возникло, получаем результаты тестирования и видоизменяем тест, если это необходимо.

Дальнейшая декомпозиция позволила более полно представить функционирование системы, определить внешние источники данных. На рис. 3 представлена DFD диаграмма описывающая этап создания теста.

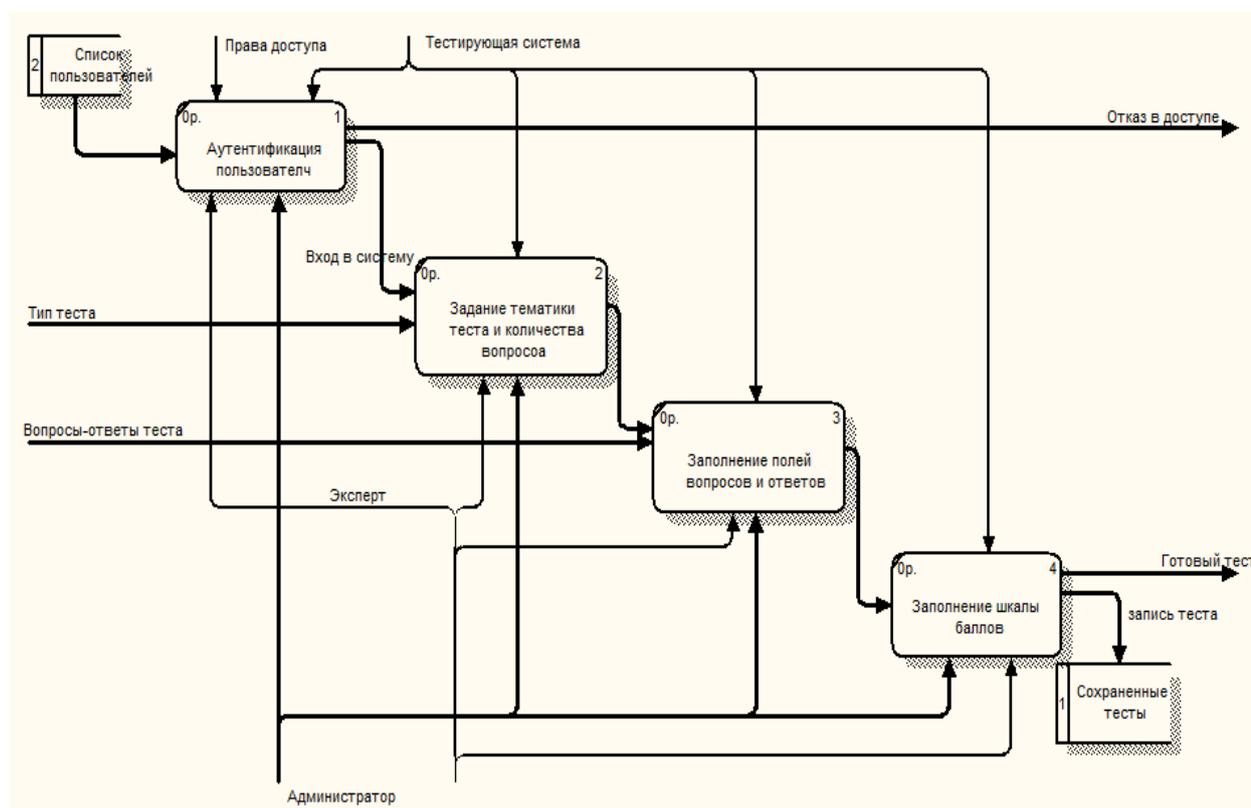


Рис.3. Создание теста.

На этапе создания теста участие принимают лишь администратор и эксперт. Из списка пользователей выбирается зарегистрированный и авторизованный пользователь, который имеет статус, указанный на схеме механизмов. Если не происходит отказа в доступе, то осуществляется переход на следующий шаг. Это задание тематики теста и его параметров. Сюда входит определение количества вопросов, ответов, типа тестирования. Далее предлагается заполнить поля с вопросами, ответами и весовыми баллами, которые в последствие будут выводиться для прохождения.

На заключительном этапе предлагается распределить весовые баллы и описать результат тестирования. Тест сохраняется в базе данных и становится доступным для прохождения. Администратор обеспечивает работоспособность ресурса и удаляет нежелательные элементы из базы данных.

Дальнейшая декомпозиция не проводилась, так как разработанная функциональная модель позволила полно описать все процессы, происходящие в системе, описать внешние источники данных.

После создания функциональной модели было выполнено проектирование базы данных средствами пакета ERWin. При разработке информационной модели базы данных (рис. 4) учитывались следующие требования:

- в базе данных должна храниться вся необходимая информации;
- сокращение избыточности и дублирования данных;
- обеспечение целостности базы данных.

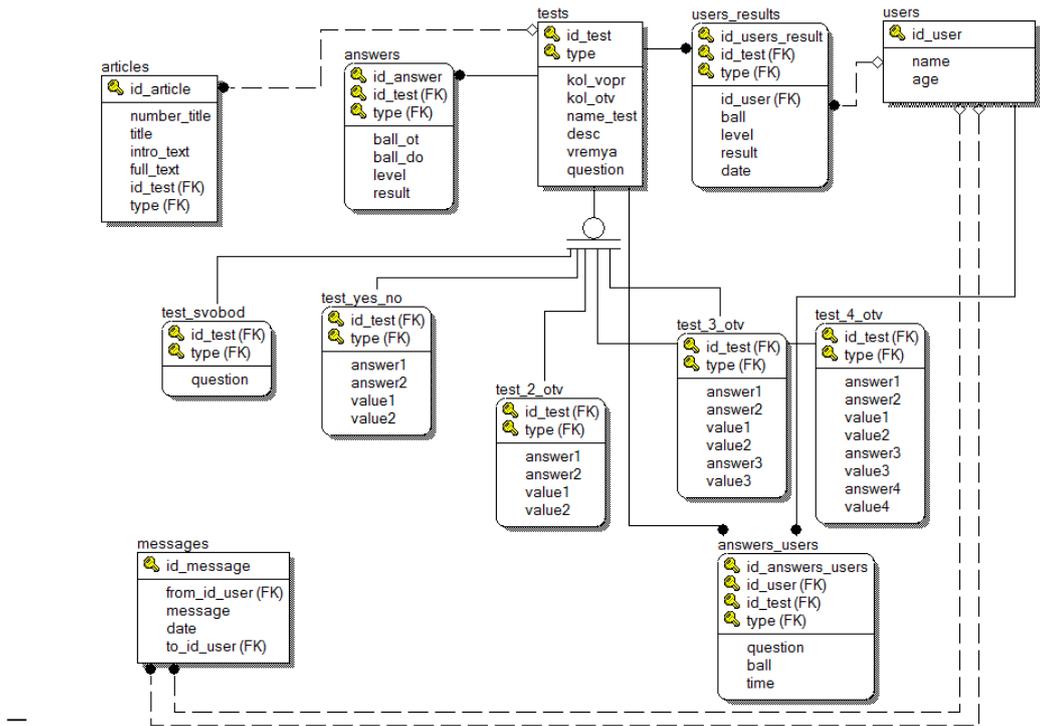


Рис. 4. Логическая модель базы данных.

На схеме представлены таблицы, содержащие информацию по пользователям, тестам и результатам тестирований:

### 3. ВЫБОР ПРОГРАММНЫХ СРЕДСТВ И РЕАЛИЗАЦИЯ СИСТЕМЫ ТЕСТИРОВАНИЯ

При разработке тестирующей системы использовались широко распространенные программные средства: веб-сервер Denver, СУБД MySQL, скриптовый язык программирования PHP, PHPMyAdmin – веб-приложение с открытым кодом, которое предназначено для контроля и администрирования СУБД MySQL, Datatables – плагин библиотеки jQuery, который позволяет оперировать таблицами, вносить в них изменения в режиме онлайн.

В результате разработки была создана программа конструирования тестов и проведения тестирования, которая представляет комплекс серверного и прикладного программного обеспечения, включающий web-сервер, СУБД, интерфейсы администратора, эксперта и тестируемого. Установка и настройка системы может быть выполнена на сервер или домашний компьютер путем копирования файлов и папок на локальный сервер или сервер хоста.

### 4. ОСНОВНЫЕ РЕЖИМЫ РАБОТЫ СИСТЕМЫ

Система рассчитана на работу трех групп пользователей: администраторы, эксперты и тестируемые.

Администратору (по умолчанию он один) предоставляется возможность контролировать пользователей, которые зарегистрированы в системе. Это необходимо как для оценки общего количества пользователей, так и для назначения им прав доступа. Администратор также имеет возможность просматривать все созданные тесты и при необходимости их удалять.

Эксперт назначается администратором, который передает ему соответствующие права на создание теста в конструкторе. На рис. 5 показана первая страница конструктора.

Введите название теста:

 ?

Введите описание теста:

 ?

Выберите из списка тип теста:

Одиночный выбор ?

Количество вопросов:  ?

Количество ответов:  ?

Время на тестирование:  ?

**Рис. 5. Начало диалога конструктора тестов.**

Конструктор допускает задание одного из четырех типов тестирования:

- одиночный выбор варианта ответа;
- множественный выбор варианта ответа;
- тест со свободным вводом ответа;
- тест с вариантами ответа «да», «нет».

Задав первый тип тестирования, конечному пользователю будет доступен выбор лишь одного варианта ответа. При выборе второго типа тестирования, пользователю станет доступен выбор нескольких вариантов ответа. Для оценки сложности ответа в системе предусмотрено использование весовых баллов: за ответ на простой вопрос пользователю дается меньше баллов, за ответ на сложный вопрос - больше.

При задании третьего типа тестирования вариантов ответа в тесте предложено не будет. Эксперт заполнит лишь вопросы для теста, а ответы будет давать пользователь в свободной форме в поле для ввода. Выбрав четвертый тип теста, испытуемому будут выводиться вопросы лишь с двумя вариантами ответа, которые носят названия «да» и «нет».

Задание поля *Количество ответов* зависит от типа тестирования. При выборе свободного ввода ответа данное поле блокируется. При выборе типа теста «да»/ «нет», поле автоматически подставляет значение «2».

Поле *Время на тестирование* является не обязательным к заполнению. Если есть необходимость ограничить прохождение заданий во времени, то в данное поле вводится количество секунд, которое отводится на тест.

После заполнения первой страницы и перехода на вторую страницу конструктора заполняются поля с вопросами, ответами и весовыми значениями теста (рис. 6).

Изменить Поиск:

id	Вопрос	Первый ответ	Второй ответ	Балл для первого ответа	Балл для второго ответа
12				0	0
12				0	0
12				0	0

id    question    answer1    answer2    value1    value2

Записи с 1 до 3 из 3 записей Предыдущая **1** Следующая

[Далее](#)

Рис. 6 Формирование теста.

Тестируемый – пользователь, у которого нет доступа к панелям администратора и эксперта. Для него доступны функции прохождения уже имеющихся тестовых заданий (рис.7).

Выход из трудных жизненных ситуаций

Вопрос номер 1

**Рассказываете ли вы другим людям о своих проблемах и неприятностях:**

- нет, так как считаю, что это не поможет;
- да, если для этого есть подходящий собеседник;
- не всегда, так как иной раз самому тяжело думать о них, не то, что рассказывать другим.

[Далее »](#)

Рис.7. Прохождение тестирования.

По окончании тестирования, система оповестит испытуемого о том, что тест пройден и предложит посмотреть результаты.

Обмен личными сообщениями.

В системе реализована функция обмена личными сообщениями между всеми категориями пользователей. Сделана она для того, чтобы получать комментарии по пройденным тестам, оповещать администратора о возможных неполадках, делиться результатами тестирования с любым зарегистрированным в системе пользователем.

## 5. ЗАКЛЮЧЕНИЕ

Разработанная система предоставляет широкие возможности по созданию различных общеобразовательных и психологических тестов. Использование конструктора тестов существенно упрощает и ускоряет процесс создания тестов, сводя его к вводу новых описаний. Наличие простого и удобного интерфейса позволяет привлекать к разработке тестов пользователей обладающих лишь начальными знаниями в области интернет-технологий.

## ЛИТЕРАТУРА

1. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 3-е изд. — СПб.: Питер, 2015. — 688 с.: ил. — (Серия «Бестселлеры O'Reilly»).
2. Зандстра, Мэтт. PHP: объекты, шаблоны и методики программирования, 3-е изд. : Пер. с англ. – М. : ООО И.Д. Вильямс, 2011. – 560 с.
3. Шмуллер, Джозеф. Освой самостоятельно UML за 24 часа, 3-е издание. : Пер. с англ. – М. : Издательский дом Вильямс, 2005. – 416с.
4. DataTables Editor Manual: [Электронный ресурс]. URL: <http://www.editor.datatables.net/manual/index>. (Дата обращения 20.02.2015).
5. jQuery Accordion: [Электронный ресурс]. URL: <http://www.jqueryui.com/accordion/>. (Дата обращения 15.04.2015).
6. А.Карелин. Большая энциклопедия психологических тестов. Издательство: Эксмо, 2007. – 416с.
7. Описания тестов: [Электронный ресурс]. URL: <http://www.azps.ru/tests/>. (Дата обращения 01.04.2015).

*Работа поступила 18.11.2015*