

---

# СТРАТЕГИЯ ТЕСТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ ВУЗОМ НА ОСНОВЕ ДОКУМЕНТОВ С ТЕГОВОЙ РАЗМЕТКОЙ

Н.А. Сергеева

Научно-исследовательский вычислительный центр  
Московский государственный университет им. М.В. Ломоносова  
*Ленинские горы, Москва, Россия, 119991*

В статье рассматривается организация процесса автоматизированного тестирования информационных систем управления вузом. В основе технической части процесса лежит работа с документами, имеющими теговую разметку. Предложенная методика рассмотрена на примере тестирования автоматизированной информационной системы «Абитуриент» в Московском государственном университете им. М.В. Ломоносова.

**Ключевые слова:** тестирование программного обеспечения, информационные системы, теговая разметка, управление вузом.

В связи с высокой стоимостью информационных систем и важностью задач, возлагаемых на них в современном делопроизводстве, возрастают требования к качеству и надежности подобных программ. С целью выявления дефектов разрабатываемого программного продукта и определения степени удобства его использования проводится тестирование. Для информационных систем управления вузом тестирование приобретает особенное значение. Это связано с тем, что, преследуя цели усовершенствования учебного процесса и экономии денежных средств, к разработке и модификации такого рода программ часто привлекают недостаточно квалифицированных студентов. Таким образом, программное обеспечение, состоящее из модулей, написанных разными людьми с разным уровнем подготовки, требует особо тщательного тестирования.

Тестирование программного обеспечения (ПО) — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. Критерии подбора тестов, методы и инструменты их проведения, участники процесса тестирования и его временные рамки определяются при разработке стратегии тестирования.

Стратегия тестирования — это план проведения работ по тестированию системы или ее модуля, учитывающий специфику функциональности и зависимости с другими компонентами системы и платформы [2].

При выработке стратегии тестирования необходимо помнить об основных принципах, позволяющих проводить тестирование с наибольшей эффективностью:

— необходимой частью каждого теста должно являться описание ожидаемых результатов работы программы;

— необходимо подбирать тесты не только для предусмотренных входных данных, но и для непредусмотренных [3];

— при анализе результатов каждого теста необходимо проверять, не делает ли программа того, что она не должна делать [3];

— вероятность наличия необнаруженных ошибок в некоторой части программы прямо пропорциональна числу ошибок, уже обнаруженных в этой части (принцип скопления ошибок).

Разработка стратегии — первый и основополагающий этап тестирования, который определяет его дальнейший ход. Всего в процессе тестирования выделяют три этапа (рис. 1).



**Рис. 1.** Этапы процесса тестирования

Из рис. 1. видно, что на этапе проведения тестов выделяется два вида тестирования:

— функциональное тестирование — тестирование с целью проверки способности ПО выполнять функции, нужные пользователям;

— нефункциональное тестирование — проверка свойств системы, не связанных с заявленной функциональностью, но существенных при практической эксплуатации системы.

Для проведения функционального тестирования существует две методики: «черного» и «белого ящика»:

— тестирование «черного ящика» — стратегия тестирования, при которой программа рассматривается как объект, внутренняя структура которого неизвестна. Тестировщик вводит данные и анализирует результат, но, как именно работает программа, он не знает [3];

— тестирование «белого ящика» — стратегия тестирования, управляемого логикой программы, которая позволяет исследовать внутреннюю структуру про-

граммы. В этом случае тестирующий получает тестовые данные путем анализа логики программы [5].

Принципы «черного ящика» и «белого ящика» не являются альтернативными. Скорее, это дополняющие подходы, которые обнаруживают различные классы ошибок.

К выбору методов и средств тестирования различных классов программного обеспечения следует подходить дифференцировано, учитывая особенности не только класса в целом, но и конкретной программы и области ее применения. Для того чтобы определить стратегию тестирования информационных систем управления вузом, рассмотрим сначала, что отличает все информационные системы от других программ. Для выявления особенностей информационных систем выделим из всего многообразия программного обеспечения два больших класса: расчетные комплексы и информационные системы.

Под информационной системой (ИС) будем понимать программный комплекс, обеспечивающий возможность хранения, модификации и получения информации, представленной в удобном конечном пользователю виде. Расчетный комплекс — программный продукт, предназначенный для автоматизации расчетов на основе введенных данных и установленных пользователем настроек.

Сравнивая эти два класса, можно выделить следующие отличительные черты информационных систем:

— большое количество вводимой с клавиатуры текстовой информации в информационных системах против преобладающего использования мыши для выбора опций и настроек в расчетных программах, т.е. при работе с ИС значительная доля информации вводится с помощью клавиатуры, что влечет за собой необходимость акцентировать внимание тестировщика на контроле обработки вводимых с использованием клавиатуры данных;

— представление результатов работы информационной системы в виде документов в формате, удобном для дальнейшего ведения процесса, автоматизируемого с помощью разрабатываемого ПО. В расчетных программах основная функциональная нагрузка ложится на проведение расчетов, результаты которых совсем необязательно должны быть представлены в виде печатных документов, а чаще выводятся в виде сообщений на экране либо графической информации. ИС, в свою очередь, в качестве одной из своих основных функций имеют генерацию документов, т.е. для ИС генерируемая документация является необходимой, в то время как в случае расчетных программ она несет скорее сопровождающую функцию;

— использование СУБД для хранения данных в большинстве случаев. Здесь СУБД является большим дополнительным компонентом, взаимодействие которого с остальными компонентами системы требует дополнительной проверки. Особое внимание в ходе такого рода испытаний стоит уделить конфигурационному тестированию, так как ввиду большого количества пользователей, работающих с ИС, увеличивается количество возможных вариантов программно-аппаратных конфигураций, на которых будет функционировать система, что может потребовать изменения схемы работы с СУБД.

Внутри своего класса информационные системы, задействованные в различных сферах деятельности, будут иметь свои особенности, продиктованные специ-

фикой процессов, автоматизируемых с помощью отдельно взятого программного продукта.

Таким образом, информационные системы управления вузом будут иметь следующие особенности:

— в вузах как минимум раз в год происходит поступление большого количества информации о новых участниках процессов. Весь этот объем данных нужно своевременно и корректно занести в информационную систему, произвести его обработку и сгенерировать все необходимые документы;

— процессы, протекающие в вузах, тесно связаны с частой сменой статусов их участников (абитуриент — студент 1 курса — студент 2 курса — ... — выпускник — аспирант и пр.). К сожалению, в настоящее время невозможно полностью отказаться от использования физических копий документов при организации учебного процесса. Все это приводит к необходимости генерации большого количества исходящей документации;

— смена статусов действующих лиц системы происходит на основе занесения в систему дополнительной информации (например, оценок).

Принимая во внимание выявленные особенности как ИС в целом, так и ИС управления вузом в частности, можно сформулировать следующие рекомендации по их тестированию:

— следует с особой тщательностью проверять корректность занесения в базу данных вводимой текстовой информации, так как она является основой для генерации официальных документов. Здесь следует уделить внимание проверке соответствия того, что пользователь ввел в поле на форме, и что в конечном итоге оказалось в базе данных. Например, некоторые СУБД хранят числовые данные таким образом, что дробная часть при записи в базу может несколько отличаться от исходной. Аналогичные проблемы могут возникнуть с хранением текстовой информации, особенно нестандартных символов. Таким образом, в генерируемый документ может закрасться фатальная неточность;

— в соответствии с одним из принципов тестирования для каждого набора входных данных должен быть сгенерирован и проверен на правильность полный комплект исходящих документов. При выполнении тестирования вручную подобные проверки потребуют колоссальных затрат временных и материальных ресурсов. Но современные средства автоматизированного тестирования предоставляют ряд инструментов, позволяющих сравнивать полученные документы с образцом, не требуя участия человека. Генерация документации также должна происходить автоматически при выполнении программы под управлением тестировочного комплекса;

— набор данных для теста должен содержать документ с входящей информацией и комплект исходящих документов, соответствующих вводимым данным.

Практическая реализация тестирования информационных систем с учетом указанных рекомендаций ввиду больших объемов данных требует использования средств автоматизированного тестирования. Но при проведении тестирования с помощью специальных программных средств также стоит помнить о рациональном их использовании. Для тестирования информационных систем управления вузом целесообразно входящую информацию представлять в виде документов с теговой

разметкой. Создание таких документов значительно облегчит программную обработку входных данных, так как они имеют четкую структуру. С другой стороны, тестировщик будет иметь строго определенную схему создания файла, который гарантированно будет корректно обработан при выполнении тестирования.

Таким образом, для тестирования информационной системы управления вузом достаточно всю входящую информацию разбить на файлы двух типов. Первый тип файлов будет содержать информацию по участникам процессов (например, абитуриентам, студентам, аспирантам), а второй — информацию, необходимую для модификации данных, находящихся в системе (оценки).

Внутри каждого тега содержатся определенные данные об участнике процесса. Таким образом, можно определить, в каком месте документа находится информация, предназначенная для занесения в определенное поле на форме. Обработка таким образом структурированных данных может проводиться с помощью алгоритма, в основе которого лежит автомат.

Абстрактный автомат — математическая модель дискретного устройства, имеющего один вход, один выход и в каждый момент времени находящегося в одном состоянии из множества возможных. На вход этому устройству поступают символы одного алфавита, на выходе оно выдает символы другого алфавита. Такой алгоритм будет на основе тегов извлекать из документа всю необходимую информацию и заносить ее в соответствующие поля на форме.

После того как из файла будет получена вся информация, она должна быть передана тестируемой системе. Для занесения данных об абитуриенте в ИС «Абитуриент» используется форма, изображенная на рис. 2. Во время выполнения тест-кейса происходят действия, полностью повторяющие действия пользователя при заполнении формы: выделение нужной информации, распределение ее по полям формы и их заполнение. Автомат делает все это путем интерпретации размеченных файлов, которые являются исходными данными для теста.

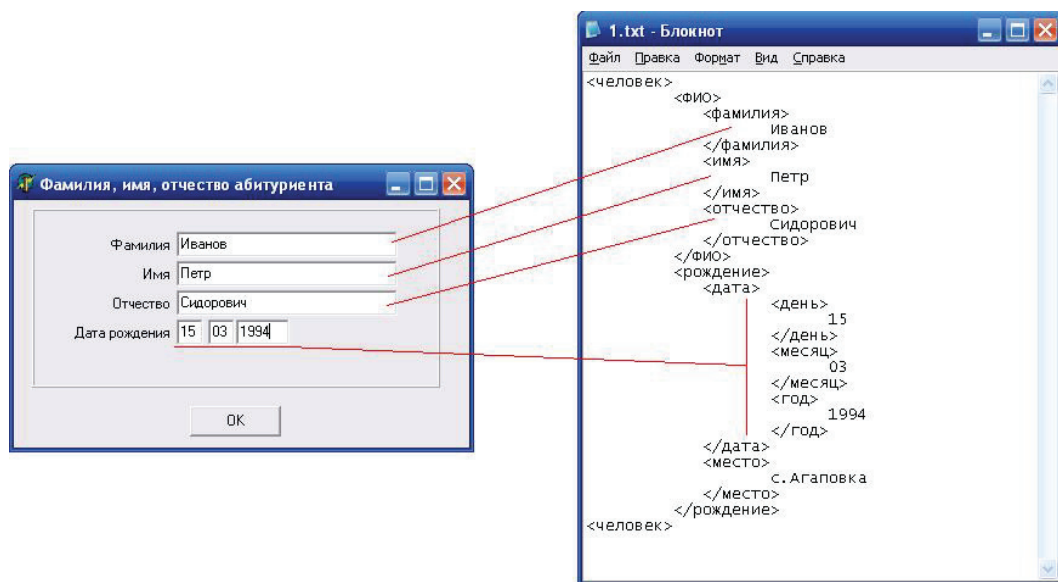


Рис. 2. Соответствие данных в документе полям формы

Дальнейший процесс тестирования заключается в проверке соответствия данным, оказавшимся в базе после сохранения внесенной в программу информации, посредством прямого обращения средства тестирования к таблицам базы данных и сравнения данных в них с данными во входящих документах. Здесь исходные данные опять извлекаются с помощью обработки размеченного тегами документа. Также необходимо сгенерировать в программе полный комплект исходящей документации и при помощи инструментов автоматизированного тестирования сравнить полученные файлы с заранее заготовленными образцами.

Применение автоматизированного тестирования имеет ряд преимуществ перед проведением тестов вручную. Во-первых, можно легко повторить весь процесс при модификации тестируемого программного обеспечения. При этом используются уже готовые входящие документы, файлы, являющиеся ожидаемыми результатами, а также не требуется заново переписывать коды сценариев, а лишь по необходимости вносить в них изменения. Во-вторых, сводится к нулю вероятность опечаток при вводе данных для тестирования. Кроме того, тестирующая система выполняет свою задачу беспристрастно и объективно. В-третьих, есть возможность ведения компактной и четкой отчетности по процессу и результатам тестирования: файлы с входными данными; тест-кейсы; результаты их работы, представленные в виде протокола. В-четвертых, сокращение временных затрат при проведении тестов очень быстро оправдывает время, потраченное на написание сценариев для среды тестирования.

Описанный подход к тестированию информационных систем управления вузом позволит не только учесть все их особенности, но и рационально использовать ресурсы для его проведения. Использование средств автоматизированного тестирования позволяет сделать процесс тестирования четко документированным, легко воспроизводимым, защищенным от случайных ошибок при занесении входящих данных, а также сократить время, затрачиваемое на обработку больших массивов данных, характерных для тестирования ИС управления вузом. Оформление входящих данных в виде документов с теговой разметкой решает вопрос представления их в распознаваемом программой виде, задавая тестировщику достаточно четкие правила. Обработка таких документов, организованная в виде автомата, интерпретирующего их, позволяет повторить все действия пользователя системы. Кроме того, такая организация позволяет повторно использовать те же документы для контроля правильности занесения данных в базу, что имеет большое значение в связи с использованием этих данных для генерации официальных документов. Проверка правильности сгенерированной документации проводится также в автоматическом режиме, что позволяет в краткие сроки установить строгое соответствие полученного файла образцу и получить документальное подтверждение этому в виде отчета среды тестирования.

#### ЛИТЕРАТУРА

- [1] *Аврамова О.Д., Владимиров А.В., Никитин В.В. и др.* Автоматизированная информационная система «Абитуриент-2009». — М.: Изд-во МГУ, 2009.
- [2] *Бейзер Б.* Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. — СПб.: Питер, 2004.

- [3] *Канер С., Фолк Д., Нгуен Е.К.* Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. — Киев: ДиаСофт, 2001.
- [4] *Колесник Н., Алтаев Г.* Учебник по SilkTest // URL: <http://www.silktutorial.ru>
- [5] *Майерс Г.* Искусство тестирования программ. — М.: Финансы и статистика, 1982.

**THE TESTING STRATEGY  
OF THE INFORMATION CONTROL SYSTEMS  
BY HIGH SCHOOL BASED ON THE DOCUMENTS  
WITH TAGGED TEXT MARKUP LANGUAGE**

**N.A. Sergeeva**

Computational Research Centre  
Moscow state university named after M.V. Lomonosov  
*Leninskiye Gory, Moscow, Russia, 119899*

This article discusses the possible organization of the process of automated testing of the information control systems by high school, based on the identified features. The basis of the technical part of the process is working with the documents that have tagged text markup. The proposed method is considered on the example of the automated information system «The Matriculate» at the Moscow State University named after M.V. Lomonosov.

**Key words:** software testing; information systems; markup documents; high school management.