



## ЭКОНОМИКА, УПРАВЛЕНИЕ, ПРАВО

УДК 004.4; 37.0

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ УЧЕТА МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

*И.В. Евдокимов, канд. техн. наук  
БрГУ, Братск*

*Созданы программное средство и базы данных для учета методического обеспечения учебного процесса. Раскрыта проблема недостаточной информированности обучающегося и работодателя о востребованности полученного образования. Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение №14.В37.21.2114.*

**Ключевые слова:** информационные технологии в образовании, база данных, программное обеспечение.

Братский государственный университет видит свою миссию в формировании единого научно-образовательного пространства северных территорий Восточной Сибири и зоны БАМ, способствующего решению задач их устойчивого социально-экономического развития. Стратегическую цель своего развития Братский государственный университет определил как постоянное улучшение качества подготовки, переподготовки и повышения квалификации инновационно ориентированных и востребованных специалистов, научно-педагогических кадров и развитие фундаментальных и прикладных исследований по важнейшим направлениям образования и науки для севера Иркутской области и зоны БАМ.

Принятая университетом политика развития состоит в планомерном достижении стратегической цели путем повышения эффективности учебного процесса, учебно-методической работы, научно-исследовательской деятельности и воспитательных мероприятий, которые базируются на финансово-экономической стабильности вуза, реализации комплексной программы управления качеством

его деятельности, развитию материально-технической базы, повышении квалификации и совершенствовании механизмов мотивации работников университета, создании безопасных и комфортных условий обучения и социального развития, содействию трудоустройству выпускников. Для ее реализации выделены основные направления развития:

- реализация комплексной программы управления качеством деятельности;
- повышение эффективности учебного процесса;
- повышение эффективности учебно-методической работы;
- развитие информационных ресурсов и др.

Реализация определенной таким образом стратегии развития позволит Братскому государственному университету укрепить свое ключевое место в активизации инновационной деятельности в северном регионе Восточной Сибири и зоны БАМ, развитию региональной инновационной инфраструктуры как одного из основных источников высокообразованных, инновационно ориентированных специалистов и перспективных научно-технических разработок.

Как показано в работе [1], недостаточное информирование обучающегося, а зачастую и работодателя, в современных экономических условиях является одной из важнейших проблем при формировании требований к качеству образования. На основании этого, и с учетом отмеченных выше основных направлений развития Братского государственного университета, актуальны разработка и дальнейшее совершенствование программных средств и баз данных [4 – 6] для учета методического обеспечения учебного процесса. Это объясняется тем, что, как показали предпроектные исследования, процесс обработки информации по компетенциям и компонентам учебно-методического комплекса (УМК) направлений подготовки бакалавров «вручную» очень трудоемкий. Создание и использование программного средства и баз данных учебно-методического комплекса направлений подготовки бакалавров позволяет сократить время на поиск информации об обеспеченности УМК и о компетенциях, на составление отчетов и др.

Процесс создания программного обеспечения начинается с выбора модели жизненного цикла. Спиральная модель представляет собой процесс разработки программного обеспечения, сочетающий в себе как проектирование, так и поэтапное прототипирование, что соответствует методу быстрой разработки программного обеспечения – RAD-технологии (Rapid Application Development).

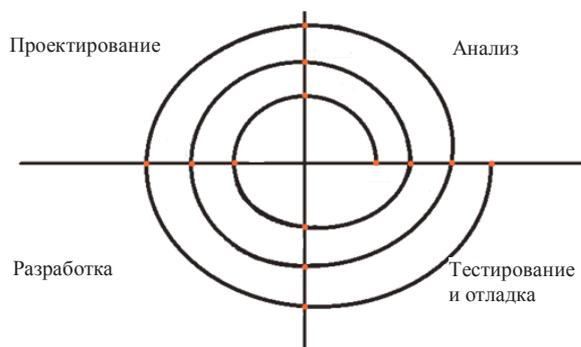


Рис. 1. Спиральная модель жизненного цикла программного средства.

RAD-технология обеспечивает создание на ранней стадии реализации действующей интерактивной модели программной системы так называемой системы-прототипа, позволяющей наглядно продемонстрировать пользователю будущую систему, уточнить его требования оперативно модифицировать интерфейсные элементы: формы ввода сообщений, меню, выходные документы, структуру диалога, состав реализуемых функций. Согласованная система-прототип служит спецификацией для дальнейшей разработки программного средства, что позволяет на ранних этапах проектирования выявить возможные ошибки проектирования и определить параметры будущей системы.

Главная задача – как можно быстрее показать пользователям программного обеспечения работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований.

Основная проблема процесса разработки по RAD-технологии заключается в определении момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена.

Для реализации технологии прототипного проектирования необходимо применять высокоуровневые инструментальные средства, которые позволяют быстро преобразовать прототип системы в функционирующую версию и внести в нее в дальнейшем необходимые изменения. Важнейшим этапом жизненного цикла является анализ требований заказчика, в ходе которого уточняются, формализуются и документируются требования к продукту. Анализ требований подразумевает их детализацию, гарантирующую, что требования понимают все заинтересованные лица, а также тщательное исследование требований на предмет ошибок, пробелов и других недостатков. Цель анализа – достаточно качественно и подробно [2] описать требо-

вания, позволяющие программисту начать проектирование, сборку и тестирование.

Зачастую отдельные требования стоит представить несколькими способами, например, в текстовой и графической форме. Это позволит выявить их особенности и проблемы, незаметные при представлении одним способом. Также это помогает всем заинтересованным лицам выработать согласованное представление об итогах разработки продукта. Графическая модель анализа отображает требования на высоком уровне абстракции. Модели позволяют выявить некорректные, несогласованные, отсутствующие и избыточные требования. Суть технического задания сводится к описанию задачи, которую требуется решить, на языке, понятном разработчику системы. Обычно разработка технического задания – это совместный труд экспертов в предметной области и специалистов, понимающих, как система может быть реализована. Анализ начинается, но не ограничивается, составлением технического задания и заключается в построении моделей предлагаемой задачи, которые приближают нас к ее решению.

Выполнение технического задания – это основной и достаточно действенный способ избежать на ранних этапах некоторых рисков, связанных с конечным результатом работы. Техническое задание разрабатывается на основе выдвинутых заказчиком требований. Являясь формализацией этих требований, техническое задание выступает гарантом того, что разработчик правильно понимает суть проблемы, которую предстоит автоматизировать. Чем больше деталей включает техническое задание, тем меньше вероятность того, что какие-то требования не будут реализованы. На основе данной документации легче проверить наличие всей необходимой функциональности готового продукта. В итоге, техническое задание является, во-первых, надежной формой фиксации соглашений между заказчиком и разработчиком программного средства и базы данных. Во-вторых,

оно позволяет снизить риск ошибок, возникающих вследствие того, что разработчик может что-то забыть реализовать или реализовать неточно в соответствии с устными договоренностями. И, наконец, в-третьих, техническое задание позволяет систематизировать проверку на соответствие разработанной базы данных ожиданиям заинтересованных лиц.

Итак, программное и информационное обеспечение (базы данных) разработаны, что подтверждено соответствующими свидетельствами о государственной регистрации в Федеральной службе по интеллектуальной собственности [4 – 6]. Краткие сведения о функциональных возможностях работы, например, с базой данных [6] приведены в сопутствующей документации. «База данных (БД) содержит сведения об учебно-методическом комплексе направления подготовки бакалавров «Управление персоналом» и предназначена для хранения, ввода, обработки и вывода информации. БД может применяться в учреждениях высшего профессионального образования сотрудниками, ответственными за методическую работу, заведующими кафедрами и т. д. БД может быть использована для формирования выборки компетенций бакалавра по запросу пользователя, что необходимо при работе с заказчиком образовательных услуг (работодатели, родители обучающихся, абитуриенты, аудиторы и др.). БД обеспечивает: сохранность данных; доступ к данным; упорядочение информации; добавление дополнительных сведений».

Интересы персонала, вовлеченного в процесс применения на практике разработанных информационных технологий учета методического обеспечения образовательного процесса, сосредоточены на проблеме управления качеством созданного программного обеспечения, что вызывает необходимость изучения исследуемой категории с позиций управления. Проведение такого анализа вызывает необходимость определения факторов, показателей качества программного обеспечения.

Качество программного обеспечения – обобщенная характеристика системы, выражающая степень его полезности пользователю. Для определения функциональной полноты, наличия технических возможностей к взаимодействию, совершенствованию и развитию необходимо использовать стандарты в области оценки показателей их качества.

Таблица 1

Характеристики качества информационной системы

Показатель	Субхарактеристики
1	2
Категорийно-описательные метрики	
Функциональные возможности	Функциональная пригодность Корректность (правильность) Способность к взаимодействию Защищенность Согласованность
Количественные метрики	
Надежность	Завершенность Устойчивость к дефектам Восстанавливаемость Доступность (готовность)
Эффективность	Временная эффективность Используемость ресурсов
Качественные метрики	
Практичность	Понятность Простота использования Изучаемость Привлекательность
Сопровождаемость	Анализируемость Изменяемость Стабильность Тестируемость
Мобильность	Адаптируемость Простота установки Сосуществование (соответствие) Замещаемость

В настоящее время зачастую выделяют шесть показателей качества программного обеспечения [3]. Исходя из принципиальных возможностей их измерения, все характеристики объединены в три группы (таблица 1):

1) категорийно-описательные, отражающие набор свойств и общие характеристики объекта (функции, категории ответственности, защищенности и важности), которые могут быть представлены номинальной шкалой категорий;

2) количественные, представляемые множеством упорядоченных, равноотстоящих точек, отражающих непрерывные закономерности, и описываемые интервальной или относительной шкалой. Эти показатели можно объективно измерить и численно сопоставить с требованиями;

3) качественные, содержащие несколько упорядоченных или отдельных значений (категорий), которые характеризуются порядковой или точечной шкалой набора категорий и устанавливаются в значительной степени субъективно и экспертно.

Для оценки надежности разработанного программного обеспечения можно использовать статическую модель надежности Миллса. Перед началом тестирования каждого модуля программы для ЭВМ [4] в исходный код искусственно вносится определенное количество ошибок лицом, не участвующим в процессе разработки и тестировании, но являющимся специалистом в области программирования. Ошибки вносятся случайным образом и фиксируются в протоколе искусственных ошибок. Затем производится процесс тестирования, и собирается статистика об ошибках, ведется их протоколирование в течение некоторого времени. В момент оценки надежности (по завершению всех тестов для функций, процедур и самого модуля в целом) по протоколу искусственных ошибок, сравниваемому с протоколом всех найденных ошибок, ошибки делились на собственные и искусственные. Модель Миллса построена,

исходя из предположения, что все ошибки (как естественные, так и искусственно внесенные) имеют равную вероятность быть найденными в процессе тестирования.

Отсюда следует соотношение, которое дает возможность определить первоначальное число ошибок  $N$ .

$$N = \frac{S \cdot n}{V},$$

где  $S$  – количество предварительно искусственно внесенных ошибок в программу разработчиком;  $n$  – число найденных собственных ошибок;  $V$  – число обнаруженных к моменту оценки искусственных ошибок.

Если первоначальное число ошибок  $N$  было больше числа найденных собственных ошибок  $n$ , то тесты для модуля перепроектируются или добавляются дополнительные тесты и производится повторное тестирование. Количество искусственно внесенных ошибок выбиралось, исходя из количества строк исходного модуля (чем больше модуль, тем больше количество  $S$ ).

Например, в модуль подсчета обеспеченности УМК при первоначальном тестировании было внесено 15 искусственных ошибок, на момент оценки найдено девять искусственных ошибок и две собственных:

$$N = \frac{15 \cdot 2}{9} = 3.$$

Тогда первоначальное число ошибок будет равно трем, т. е. одна ошибка была не найдена, поэтому необходимо произвести повторное тестирование с повторным внесением ранее найденных девяти искусственных ошибок. После тестирования были выявлены все 15 внесенных ошибок и три собственных. Следовательно, по модели можно было сделать предположение, что все ошибки в модуле найдены, и тестирование данного модуля следует прекращать. Так же, с помощью второй части модели Миллса, была рассчитана вероятность, с которой можно

высказать такое предположение, по следующей формуле:

$$C = \begin{cases} 1, & \text{если } n > K; \\ \frac{S}{S + K + 1}, & \text{если } n \leq K, \end{cases}$$

где  $K$  – предполагаемое количество ошибок;  $n$  – количество найденных ошибок при тестировании.

Если данная вероятность будет меньше 0,5, значит, следует увеличить количество искусственно вносимых ошибок и повторить процесс тестирования модуля.

Таким же способом была оценена надежность всей программы в целом, и вероятность того, что все ошибки найдены и устранены, составила 0,94, или 94 %. Что свидетельствует о высокой надежности разработанного программного изделия.

#### Литература

1. Патрусова А.М. Исследование качества образования как объекта управления // Проблемы социально-экономического развития Сибири. 2011. № 5. С. 36-43.
2. Евдокимов И.В., Вахрушева М.Ю. Показатели качества и надежности программного обеспечения // Труды Братского государственного университета. Сер. Проблемы управления социально-экономическим развитием регионов Сибири. 2012. Т.1. С. 154 – 157.
3. Евдокимов И.В., Вахрушева М.Ю. Разработка программного обеспечения на основе спиральной модели жизненного цикла // Труды Братского государственного университета: Сер. Проблемы управления социально-экономическим развитием регионов Сибири. 2011. Т. 2. С. 124 -126.
4. Прикладная информатика в экономике: учеб.-методический комплекс направления подготовки бакалавров /И.В. Евдокимов, Е.В. Новикова. Св. ГР № 2012618297 Рос. Федерация; зарег. в Реестре программ для ЭВМ 13.09.12.
5. Информационный менеджмент (bakalavr\_ИМ v.1.1): учеб.-методический комплекс направления подготовки бака-

лавров /Л.А. Грищенко, И. В. Евдокимов. Св. ГР № 2012620919 Рос. Федерация; зарег. в Реестре программ для ЭВМ 13.09.12.

6. Управление персоналом (bav-lavr\_U.P v.1.2): учеб.-методический ком-

плекс направления подготовки бакалавров / Н.Ю. Козина, И.В. Евдокимов. Св. ГР № 2012620919 Рос. Федерация; зарег. в Реестре программ для ЭВМ 13.09.12.

УДК 658.5.012.14; 681.5.015

## ПРОЦЕДУРА ИДЕНТИФИКАЦИИ КАК ЭТАП СОЗДАНИЯ СИСТЕМ УПРАВЛЕНИЯ И ПРИНЯТИЯ РЕШЕНИЙ

*И.В. Евдокимов, канд. техн. наук  
БрГУ, Братск*

*В работе предложен метод идентификации нелинейных систем в классе блочно-ориентированных моделей, основанный на методе составления и решения системы корреляционных уравнений. Создано программное обеспечение, реализующее разработанные вычислительные алгоритмы и позволяющее проводить структурную и параметрическую идентификацию нелинейных динамических объектов в классе моделей фильтра Заде с использованием входного сигнала в виде суммы гармонических компонент. Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение №14.В37.21.2114.*

**Ключевые слова:** нелинейные системы, идентификация объектов, нелинейные математические модели, программа для ЭВМ, блочно-ориентированные модели, фильтр Заде, модели Гаммерштейна, планирование эксперимента.

Теория идентификации в настоящее время является одним из наиболее интенсивно развивающихся разделов общей теории управления. Основное внимание при этом уделяется идентификации в классе нелинейных динамических моделей, которые наиболее адекватно отражают самые существенные и значимые характеристики большинства современных реальных производственно-технических, экономических и др. объектов. Действительно, современные системы управления разрабатываются, в основном, для производственных процессов, являющихся сложными многоэлементными системами. Если не вникать во внутреннюю структуру исследуемого объекта, а моделировать связь между его входным и выходным процессами, то это приводит к значительному упрощению модели. Задача идентификации объектов составляет

один из основных этапов создания систем управления и принятия решений.

В то же время, идентификация нелинейных динамических объектов (НДО) является принципиально более сложной и объемной задачей, чем линейная идентификация, представляющая частный случай нелинейной. Наиболее часто используемым на данный момент подходом к проблеме идентификации НДО является априорное задание некоторого (по возможности – достаточно представительного) класса допустимых моделей. В качестве такого класса в настоящем исследовании выбран так называемый фильтр Заде, позволяющий достаточно адекватно описывать широкий класс реальных объектов [1].

Такого типа модели относятся к классу блочно-ориентированных моделей. Они достаточно эффективны для описания сложных технологических процессов, ха-