их подготовки. Встает вопрос о роли высшего образования и самообучения в сфере информационных технологий и каким же всё-таки будет их оптимальное соотношение.

Встаёт вопрос о том, каким должен быть процесс обучения, чтобы студенты в учебных заведениях могли овладеть методологией усвоения новых знаний, которая поможет им заниматься самообразованием. Высшее образование дает широкие возможности для научной деятельности, без которой сложно представить какое-либо развитие. Но тут важно помнить, что инженер в сфере информационных технологий — это практик, и судят об эффективности его труда в основном по практическому результату. Любая научная теоретическая работа должна быть лишь базой для последующих инновационных разработок, имеющих практическое применение. Здесь и появляется вторая составляющая процесса подготовки специалиста в сфере информационных технологий – самообзразование.

Основой непрерывного самообразования является процесс самообучения, обеспечивающий студентам приобретение таких личностных качеств, знаний и умений, которые позволяют им адаптироваться в быстро меняющихся условиях профессиональной деятельности. Это предполагает овладение обучающимися способами самостоятельного приобретения знаний, формирование самостоятельности как профессионально значимого личностного качества будущего специалиста. В связи с этим одной из важнейших задач средней и высшей школ становится формирование готовности молодых людей к самообучению, что обеспечит их будущий личностный и профессиональный рост.

В большинстве случаев от программиста требуется поиск творческих решений. Обучить этому умению достаточно сложно и не всегда возможно [4]. Инициатором обучения должен быть сам человек, иначе никакое обучение не принесет желаемого результата. Высшие учебные заведения не могут предоставить будущим программистам весь необходимый багаж знаний, которого им хватит на всю жизнь, поэтому они должны научить студента рассуждать, решать нестандартные задачи, изучать большие объемы информации в сжатые сроки, участвовать в различных учебных проектах.

Следует выделить факторы, которые усложняют процессы образования и самообразования в сфере информационных технологий [2]: стремительное развитие информационных технологий; высокие требования к технической и математической подготовке обучаемого; дефицит высококвалифицированных преподавательских кадров; необходимость знания английского языка; высокие требования к самоорганизованности обучаемого и др.

Это лишь одни из немногих факторов, которые необходимо учитывать при разработке программы обучения специалиста в сфере информационных технологий.

В рамках рассматриваемой темы, были выдвинуты некоторые гипотезы:

Г1 – Технологии, которые преподаватели изучали обучаясь в высшем учебном заведение, сейчас не востребованы и сильно устарели.

Г2 – Технологии устаревают за 3-5 лет.

Г3 – Преподаватели изучают новые технологии самостоятельно, не посещая специальные курсы и тренинги.

В результате проведенных исследований [3] выяснилось, что все три гипотезы верны. Это позволило вывести те требования к процессу обучения специалистов сферы информационных технологий, которые повысят его эффективность. Преподаватели в сфере информационных технологий должны находиться в курсе последних тенденций в развитии той или иной отрасли. Каждые 3 года необходимо анализировать и обновлять образовательную программу, иначе студенты будут изучать устаревшие технологии, которые им не пригодятся в будущем. Приоритет самообучения при подготовке специалистов сферы информационных технологий. Основная задача средней и высшей школы – формирование навыков и умений, дающих возможность грамотно и эффективно заниматься самообразованием.

Список использованных источников:

1. Ruzic-Dimitrijevic L. Challenges IT Instructors Face in the Self-Education Process, 2014 – 35-48 с.
2. Cohen E. Challenges of information technology in the 21st century, 2002 – 12 с.
3. Cox M..The changing nature of researching IT in education, 2007 – 22-40 с.
4. McGill T. Current issues in IT education, 2003 – 70 с.

РЕВЕРС-ИНЖИНИРИНГ ИСХОДНОГО КОДА ПРИ ПОМОЩИ ЯЗЫКА ШАБЛОНОВ

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

Быков А.А.

Пилецкий И.И. – к.ф.-м.н., преподаватель, БГУИР

Обратная разработка как процесс исследования устройства или программы с целью понять её устройство известен давно. Широко известна методологии SADT как методология моделирования сложных систем [1]. С появлением этой методологии связано появление идей реинжиниринга бизнес процессов. Однако внедрение моделирования в процесс промышленной разработки затрудняется сложностью инструментов реверс-инжиниринга и, как следствие, слабой связанностью процесса моделирования и

процесса разработки. Появление инструментов автоматической генерации кода на основе шаблонов в IDE в последние годы позволило упростить написание программистами часто используемых конструкций. Хотя средства моделирования в основном используются для высокоуровневого описания структуры и функций проекта, а язык шаблонов для интерактивного быстрого написания непосредственно кода проекта, эти два инструмента могут дополнить работу друг друга. С этой целью был разработан сервер генерации и реверс-инженеринга исходного кода при помощи упрощенного языка шаблонов FMPP.

Цель системы – упростить разработку и анализ исходного кода программ, используя шаблоны как отдельных языковых конструкций, так и, группы нескольких файлов. Система активно использует язык шаблонов FMPP для описания библиотеки шаблонов. Описание проектируемой программы в системе состоит из 3-х элементов: исходного кода проектируемой программы, шаблонов кода и модели проектируемой программы. Технически решение представляет собой сервер, позволяющий выполнять преобразования между кодом и моделью программы в обоих направлениях, а также обновлять шаблоны на основе исходного кода. Сервер отслеживает изменение каждого из 3-х элементов и выполняет одно из трех преобразований. Основными операциями являются преобразование между кодом и моделью. Обновление шаблонов выполняется только в том случае, если изменившийся исходный код не может быть разложен полностью при помощи библиотеки шаблонов.

Модель системы описывается согласно методологии SADT и хранится в виде связанных XML файлов, описывая структуру и характеристики проекта [2]. Каждый файл модели связан с шаблоном исходного кода. Система использует 2 вида файлов модели – общие и индивидуальные. Общие для нескольких шаблонов данные хранятся в файле model.xml. Может использоваться несколько таких файлов по одному во вложенной папке. Такие файлы представляют собой дерево, наследуя данные, определенные в коревых файлах. Индивидуальные данные сохраняются в файлах <имя файла исходного кода>.xml. Файлы описывают данные, используемые только в одном файле исходного кода, и являются конечными узлами дерева данных.

Шаблоны связывают блоки модели проекта и их реализацию в проекте при помощи языка FMPP [3]. Используемые шаблоны кода можно разделить на библиотеку общих шаблонов и шаблоны исходного кода для данного проекта. Библиотека шаблонов позволяет переносить наработанные в предыдущих проектах проектные решения, а шаблоны индивидуальные для проекта – конечные реализации этих решений для данного проекта. Шаблоны библиотеки хранятся в директории tmpl и могут генерировать сразу несколько файлов исходного кода. Индивидуальные шаблоны могут генерировать код только для одного файла исходного кода. Чаще всего эти шаблоны генерируются сервером автоматически и являются отражением изменений кода по сравнению с шаблоном из библиотеки. То, что один шаблон библиотеки может генерировать несколько исходных файлов, разрывает связь между шаблоном и файлом исходного кода. Для описания связи файлов исходного кода с шаблонами библиотеки используются ссылки в начале каждого файла. Эти ссылки оформляются как комментарии для используемых языков программирования. Символы комментирования ссылок определяются в шаблонах.

Базовой конструкцией шаблона является переменные, определенные в файлах данных. Они определяются при помощи имени внутри фигурных скобок, начинающихся со знака “$”. Также переменные используются в управляющих конструкциях, а также как параметры для макросов [3]. Для генерации кода согласно некоторого условия используется конструкция “if”. Логическим условием для конструкции является переменная. Для генерации повторяющихся конструкций используется конструкция “for”. Разработанная система позволяет гибко разрабатывать приложения на основе шаблонов, анализировать исходный код проектов.

Список использованных источников:

1. Brackett, J., and C. McGowan: "Applying SADT to Large System Problems", SofTech Technical Paper TP059,January 1977.
2. Ross, D.: "Structured Analysis (SA): A Language for Communicating Ideas", IEEE Transactions on Software Engineering, vol. 3, no. 1, January 1977
3. http://fmpp.sourceforge.net/index.html.

МОНИТОРИНГ И УПРАВЛЕНИЕ ФУНКЦИОНАЛЬНЫМ СОСТОЯНИЕМ МАШИНИСТА ЭЛЕКТРОПОЕЗДА

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

Гедранович Ю. А.

Яшин К. Д. - к.техн.н., доцент

Описана методика проведения эксперимента по оценке возможности управления функциональным состоянием машиниста на основе биологической обратной связи по параметрам тремора и кожно-гальванической реакции.

Предсменный и внутрисменный контроль функционального состояния машиниста электропоезда является одним из важнейших мероприятий по обеспечению безопасности перевозок железнодорожным транспортом. Только за 2009 год экспертными комиссиями признаны непригодными к работе 139 человек (0,5 % от общей численности работников) [1]. Неудовлетворительные результаты медицинских осмотров демонстрируют необходимость более тщательного подхода к усовершенствованию навыков саморегуляции функционального состояния машинистами для преодоления эмоциональных стрессов и состояния тревоги.