

Методологические Основы Разработки Образовательных Программ

Фирудин Агаев¹, Гюляра Мамедова², Лала Зейналова³, Рена Меликова⁴

^{1,2,3,4}AMEA, Институт Информационных Технологий, Баку, Азербайджан

¹depart10@iit.ab.az, ²gyula.ikt@gmail.com, ³lalamailala@bk.ru

Аннотация– В статье показано, что в связи с обновлением и изменением технологий аппаратного и программного обеспечения информационных технологий, необходимо менять подходы к подготовке кадров, разрабатывающих программные системы. В статье представлены требования международного стандарта SE (ACM)-2014 к образовательным программам программной инженерии для высших учебных заведений, приводятся примеры образовательных программ данного направления ведущих университетов США и Европы.

Ключевые слова– образовательная программа, программная инженерия, требования международных стандартов, стандарт SE (ACM)-2014, инструменты программной инженерии, методы программной инженерии, безопасность программного обеспечения

I. ВВЕДЕНИЕ

Одной из быстроразвивающихся областей информационных технологий является программная инженерия, в которой постоянно обновляются технологии и методологии программного и аппаратного обеспечения. В этой связи должны меняться и подходы к подготовке кадров для этой отрасли, образовательные программы должны учитывать динамику развития и постоянно обновляться. **Программная инженерия** — это область компьютерной науки и технологии, которая занимается построением программных систем, настолько больших и сложных, что для этого требуется участие слаженных команд разработчиков различных специальностей и квалификаций [1].

Массовое создание сложных программных средств промышленными методами и большими коллективами специалистов вызвало необходимость их четкой организации, планирования работ по требуемым ресурсам, этапам и срокам реализации. Программная инженерия — это интегрирование принципов математики, информатики и компьютерных наук с инженерными подходами.

Программное обеспечение играет центральную роль практически во всех аспектах повседневной жизни: коммуникации, производство, банковское дело и финансы, образование, транспорт, развлечения, медицина, сельское хозяйство и юриспруденция. Число, размер и области применения компьютерных программ значительно выросли, они помогают нам быть более эффективными и продуктивными, помогают нам более

эффективно решать проблемы и предоставляют нам более безопасные, более гибкие решения.

Ранние подходы к разработке программного обеспечения основывались на экспертной интерпретации опыта («структурированные» подходы), а также на использовании моделей, основанных на математическом формализме («формальные» подходы) [2]. В настоящее время знания в области программного обеспечения принимают множество форм: в виде сочетания моделей (формальных и неформальных), экспертного опыта и эмпирических оценок наилучшим образом подходящих для решения поставленных задач. Современные инженерные специалисты при создании и внедрении проектов эффективно работают как часть команды, при этом, программные инженеры могут выполнять несколько функций: исследование, разработка, проектирование, эксплуатация, тестирование, управление операциями продаж, консалтинг программных продуктов и т.д.

II. СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ УНИВЕРСИТЕТОВ США И ЕВРОПЫ

Образовательные программы университетов США и Европы по направлению «Программная инженерия» состоят из трех частей [3-10]:

- ✓ методы программной инженерии (*Software Engineering Methods*);
- ✓ инструментальные средства программной инженерии (*Software Engineering Tools*);
- ✓ безопасность программного обеспечения и систем (*Software and Systems Security*).

К методам программной инженерии относятся: математические методы (*Software Engineering Mathematics*), параллельные и распределенные системы (*Concurrency and Distributed Systems*), эффективное моделирование (*Performance Modelling*), гибкие методы (*Agile Methods - AGM*), управление рисками и качеством (*Managing Risk and Quality*), управление развитием программного обеспечения (*Software Development Management*), Разработка требований (*Requirements Engineering*), архитектура программного обеспечения (*Enterprise Architecture*), безопасность критических систем (*Safety Critical Systems*) и т.п.

К инструментальным средствам относятся: объектно-ориентированное программирование (*Object Oriented*

Programming); объектно-ориентированное проектирование (*Object Oriented Design*); шаблоны проектирования (*Design Patterns*); тестирование программного обеспечения (*Software Testing*); проектирование баз данных (*Database Design*); функциональное программирование (*Functional Programming*); параллельное программирование (*Concurrent Programming*); практика гибкой инженерии (*Agile Engineering Practices*); расширяемый язык разметки (*Extensible Markup Language*); сервис – ориентированная архитектура (*Service Oriented Architecture*); мобильные и сенсорные сети (*Mobile and Sensor Networks*).

К безопасности программного обеспечения и систем относят: принципы безопасности (*Security Principles*), безопасное и надежное программирование (*Secure and Robust Programming*); инфраструктура надежных вычислений (*Trusted Computing Infrastructure*); проектирование безопасности (*Design for Security*); анализ и управление рисками (*Risk Analysis and Management*); люди и безопасность (*People and Security*); безопасность сетей (*Network Security*); безопасность облаков (*Cloud Security*); компьютерная криминалистика (*Forensics*); безопасность данных и конфиденциальность (*Data Security and Privacy*); управление безопасностью и инцидентами (*Security and Incident Management*); безопасность мобильных систем (*Mobile Systems Security*); безопасность беспроводных сетей (*Security in Wireless Networks*).

Студентам этого направления необходимо дать представление о некоторых передовых инструментальных средствах, используемых в программной инженерии, чтобы они смогли проанализировать достоинства и ограничения инструментария и показать, какими должны быть лучшие решения. Они должны изучить несколько современных языков программирования.

Инженеры используют инструменты для систематического применения в проектировании процессов. Таким образом, выбор и использование соответствующих инструментов является ключевым аспектом проектирования.

Математический инструментарий (математический анализ, дискретная математика, математическая логика) необходим студенту для создания крупных программных проектов. Необходимо помнить, что студенты, научившиеся применять математическое мышление к разработке программного обеспечения, имеют огромные преимущества по сравнению с теми, кто подобными навыками не обладает.

Разработка программного обеспечения основывается на самых разнообразных дисциплинах и опирается на многие области информатики, на его теоретические и концептуальные основы, но она также требует от учащихся использования концепций из других областей, таких как математика, инженерия и управление проектами. Все студенты, изучающие программное обеспечение, должны научиться интегрировать теорию и практику, осознать важность абстракции и моделирования, оценить важность хорошего дизайна и

уметь приобретать специальные знания предметной области вне компьютерной дисциплины в целях поддержки разработки программного обеспечения для конкретных областей.

III. ТРЕБОВАНИЯ НОВОГО СТАНДАРТА SE (ACM) -2014

Выпускники, специализирующиеся по направлению «Программная инженерия», должны продемонстрировать следующее качество [11]:

1. Профессиональные знания. Показывать владение знаниями в области программного обеспечения, навыков и профессиональных стандартов, необходимых для начала практики в качестве программиста. Кроме того, выпускники должны научиться понимать и оценивать профессиональные проблемы и стандарты, связанные с этикой и профессиональным поведением, экономикой и социальными потребностями.

2. Технические знания. Должны демонстрировать понимание и применять соответствующие теории, модели и методы, которые служат основой для выявления проблем и анализа разработки программного обеспечения, внедрения, тестирования и документирования. Студенты должны уметь оценивать и анализировать процессы, которые они исследуют, а также решения, которые они принимают.

3. Работа в команде. При разработке программных продуктов они должны работать как индивидуально, так и в составе команды. При групповой работе студенты должны быть проинформированы о характере групп и групповой деятельности и возможной их роли в составе этих групп.

4. Понимание конечного пользователя. Специалисты этого направления должны демонстрировать понимание важности переговоров с заказчиками, предлагать эффективные решения и качественную продукцию.

5. Контекстные проектные решения. На протяжении всего обучения студенты должны быть ознакомлены с различными соответствующими подходами к инженерному проектированию и примерами их использования при разработке программного обеспечения для различных прикладных областей. Они должны быть способны понять сильные и слабые стороны доступных вариантов и последствия выбора конкретных подходов для конкретной ситуации. Предлагаемые ими проектные решения должны разрабатываться в контексте этических, социальных, правовых, безопасных и экономических проблем.

6. Компромиссные решения. Согласование противоречащих целей проекта, поиск приемлемых компромиссов в рамках ограничений стоимости, времени, знаний, существующих систем и организаций. Студенты должны заниматься упражнениями, которые подвергают их конфликтующим и изменяющимся требованиям. В таких случаях должен присутствовать сильный элемент реального мира, чтобы обеспечить реалистичность этого

опыта. Учебным подразделениям следует решать эти вопросы с целью обеспечения высококачественных функциональных и нефункциональных требований к программному проектированию

7. Постоянное профессиональное развитие.

Студенты должны изучать новые модели, методы и технологии по мере их возникновения и оценивать необходимость такого непрерывного профессионального развития. На протяжении всей программы обучения студентам следует искать новые знания и оценивать их полезность и актуальность. Продолжающаяся эволюция знаний, технологий, приложений, педагогики и практики в области программного обеспечения вместе с профессиональным характером разработки программного обеспечения требует постоянного пересмотра учебной программы и акцента на важности непрерывного обучения выпускников.

Требования к образовательным программам программной инженерии концентрируются на знаниях и педагогике, связанных с программным обеспечением инженерной учебной программы.

Для решения постоянно меняющегося характера разработки программного обеспечения образовательные учреждения должны принимать четкие стратегии реагирования на изменения. Это должно включать постоянный процесс обзора, который позволяет периодически обновлять отдельные компоненты учебной программы. Вузы, при условии ограниченности имеющихся у них ресурсов, должны признать важность того, чтобы оставаться в курсе достигнутых успехов как в области технологий, так и в педагогике. Кроме того, образование в области программного обеспечения должно быть нацелено на подготовку учащихся к обучению на протяжении всей жизни, что позволит им выйти за рамки сегодняшней технологии для решения задач будущего.

ВЫВОДЫ

В статье показано, что образовательная программа ВУЗа должна поддерживать идентификацию основных навыков и знаний, которыми должны обладать все выпускники программной инженерии, и основываться на

соответствующем определении знаний в области программного обеспечения. Описание этих знаний должно быть кратким и подходящим для учебного заведения, должно опираться на последние рекомендации SE (ACM) -2014 для высших учебных заведений и анализе образовательных программ ведущих университетов мира. В этом описании необходимо указать базовый набор обязательных тем для всех степеней инженерии программного обеспечения, ядро знаний должно быть широко признано сообществом образования в области программного обеспечения.

Необходимость в специалистах по программной инженерии диктуется потребностями информационно-коммуникационной отрасли: стремительным развитием ИТ-технопарков, рынка оффшорного и заказного программирования, информатизацией государственных структур, потребностями частного бизнеса. Индустрия программного обеспечения является развивающейся областью современной мировой экономики, а программное обеспечение — важнейшая составляющая информационных систем, в которых сконцентрирован интеллект инженеров-программистов.

ЛИТЕРАТУРА

- [1] В.В.Липаев Программная инженерия. Методологические основы: Учебник. - М.: Теис, 2010, 608 с.
- [2] R. J. Allen, D. Garlan, A formal basis for architectural connection, ACM Trans. on Software Engineering and Methodology, 6(3), 1997, 213–249.
- [3] <https://eecs.berkeley.edu/academics/courses>
- [4] <http://www.admissions.purdue.edu/majors/interest/engineering-technology/index.php>
- [5] <http://cs.stanford.edu/academics/courses>
- [6] http://catalog.caltech.edu/documents/79-catalog_16_17_part5.pdf
- [7] <https://moodle.polytechnique.fr/course/index.php?categoryid=187>
- [8] <https://www.ethz.ch/de/studium/anmeldung-bewerbung/master/programme.html>
- [9] <https://www.ox.ac.uk/sites/files/oxford/UGP%202018%20entry%20Courses.pdf>
- [10] <http://www.cl.cam.ac.uk/admissions/undergraduate/course/>
- [11] Software Engineering 2014, Curriculum Guidelines for Undergraduate, Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, 23 February 2015