Условия формирования компетенций по специальности "Программная инженерия"

Богдан Денисович Терехин
студент
Дальневосточный Федеральный Университет
Владивосток, Россия
terehin@dvfu.ru
0000-0000-0000-0000

Денис Алексеевич Старцев
студент
Дальневосточный Федеральный Университет
Владивосток, Россия
starcev@dvfu.ru
0000-0000-0000-0000

Данила Андреевич Раздобаров
студент
Дальневосточный Федеральный Университет
Владивосток, Россия
razdovarov@dvfu.ru
0000-0000-0000-0000

Поступила в редакцию 25.10.2022
Принята 18.11.2022
Опубликована 01.12.2022

10.25726/q1125-2979-0698-u

Аннотация

Начиная с 2012 года, среди приоритетных направлений образования и науки по обучению студентов и аспирантов, стажировки научных и научно-педагогических работников в ведущих высших учебных заведениях и научных учреждениях за рубежом, которые относятся к информатике и вычислительной технике, три – программная инженерия, программное обеспечение систем и инженерия программного обеспечения – относятся к одной специальности: 121 – Инженерия программного обеспечения. Кроме того, значительная часть других приоритетных направлений (математическое и компьютерное моделирование; информационно-коммуникационные технологии; системы искусственного интеллекта; системное программирование и др.) являются касательными к ней. Технологии и средства разработки программных продуктов и систем определены как одно из приоритетных направлений научных исследований и научно-технических разработок в России на период до 2020 года. Эти и ряд других законодательных инициатив нашего государства является свидетельством неотложной общественной потребности в компетентных специалистах по инженерии программного обеспечения, подготовленных на основе лучших мировых стандартов и передового зарубежного опыта и способных к проектированию, апробации, внедрения и коммерциализации инновационных технологий ИПО. Анализ мирового опыта по подготовке специалистов по ИПЗ уместно начать с ретроспективного обзора эволюции самого понятия "Инженерия программного обеспечения" и основных этапов развития этой отрасли. Целью статьи является анализ основных этапов развития ИПЗ как области знаний, выделение фундаментальных составляющих подготовки будущих инженеров программистов и определение тенденций развития этой отрасли на ближайшее десятилетие.

Ключевые слова
программная инженерия, компетенция, специальность, формирование.

Введение

Первое употребление термина «программная инженерия» (software engineering) датируется августом 1966 года (Aleem, Capretz, Ahmed, 2016), но годом появления инженерии программного обеспечения (ИПО) как отрасли считается 1968 год, в котором в Германии состоялась первая конференция, которая называлась «Software engineering». Главной тематикой конференции было проектирование, производство и обслуживание программного обеспечения. Один из главных участников конференции П. Наур (Peter Naur) отмечал, что работа проектировщиков программного обеспечения похожа на работу архитекторов и инженеров-строителей, особенно тех, кто занимается проектированием крупных гетерогенных конструкций, таких как города и промышленные предприятия (Elsakova R., Leskina J.,). Термин «программная инженерия» (software engineering) был умышленно выбран как провокационный: «это понятие означало, что производство программного обеспечения должно базироваться на том же типе теоретических основ и практических применений, что и в традиционных отраслях инженерии» (Murphy-Hill, Zimmermann, Nagappan, 2014).

Происхождение программной инженерии участники конференции связывали с кризисом программного обеспечения – термином, предложенным председателем программного комитета Ф. Л. Бауэром (Friedrich Ludwig «Fritz Bauer»). По его мнению, кризис заключался в невозможности применения «"кустарных» (полуинтуитивных) методов разработки для производства крупных масштабируемых программных систем: "существующее программное обеспечение разрабатывается любителями (независимо от того, где – в университетах, компаниях или на производстве) с помощью мастерства одиночек (в университетах) или большого количества работников («миллион обезьян») на производстве, является ненадежной и требует постоянного «технического обслуживания» (причем слово «обслуживание» неправильно используется для обозначения сбоев и отказов, которые ожидаются от производителя с самого начала), есть неопрятным, непрозрачным и неусовершенствуемом (или по крайней мере слишком дорогим, чтобы это сделать). И наконец, существующее программное обеспечение поступает слишком поздно и стоит дороже, чем ожидалось, и не оправдывает надежд, которые на него возлагались» (Murphy-Hill, Zimmermann, Nagappan, 2014). По результатам работы конференции в 1971 году Ф. Л. Бауэр дал полушутливое определение ИПЗ как части информатики, слишком тяжелая для информатиков, и более серьезное – как создание и использование рациональных принципов инженерии для получения экономичного, надежного и эффективно работающего на реальных компьютерах программного обеспечения (Murphy-Hill, Zimmermann, Nagappan, 2014).

Материалы и методы исследования

По мнению таких участников конференции, А. Дж. Перлис (Alan Jay Perlis) и Ф. Л. Бауэр, для проектирования программного обеспечения математическая подготовка не требуется, но ее наличие придает программному проекту элегантности, ведь программные системы являются математическими за природой и должны быть построены по уровням и модулями, что образуют математическую структуру (Oliveira, Tereso, Machado, 2014).

Основными критериями проектирования были выбраны общие критерии (общие для разных систем), пользовательские требования, надежность и логическая полнота. Среди технологий проектирования обсуждались последовательность шагов проектирования, структура программного проекта, обеспечение обратной связи через мониторинг и моделирование, применение высокоуровневых языков программирования и тому подобное.

Относительно профессиональной подготовки специалистов по ИПЗ А. Дж. Перлис и Е. Е. Девид-младший (Edward Emil David Jr.) поставили ряд проблемных вопросов (Абрамова, Войнова, 2019):

1. Можно ли работать инженером-программистом без формального образования по соответствующей специальности?
2. Совпадает ИПЗ с компьютерными науками?

3. Как лучше подготовить специалиста с ИПЗ: по бакалаврской программе в университете, на курсах повышения квалификации или по двухгодичной программе подготовки после стандартного школьного образования?

4. Будут специалисты, подготовленные по этим программам, рост и перспективы в нашем обществе?

5. Будут ли они достаточно полезными для фирмы, правительства или университета, и является ли их значение таким, что они могут реализовать свои таланты в других видах деятельности, или они навсегда обречены оставаться программистами?

6. Какие программы подготовки необходимых специалистов с ИПЗ независимо от уровня образования?

7. Или ИПЗ действительно отличается от того, что мы сейчас называем системной инженерией?

8. Что общего имеют ИПЗ и компьютерная инженерия с традиционным инженерным образованием в Соединенных Штатах Америки или Западной Европы?

Д. Т. Росс (Douglas Taylor Ross), отвечая на поставленные вопросы, подчеркивал на необходимости отдельного формального образования с ИПЗ на уровне бакалавра (Алексашина, Горбунова, 2017). Другие участники дискуссии поднимали вопросы практической подготовки как специалистов, так и преподавателей на соответствующей специальности.

Конференция 1969 года (Бутенко, Семенова, 2019), состоявшаяся в Италии, была посвящена технологиям ИПЗ. В отличие от предыдущей, профессиональная подготовка специалистов по ИПЗ стала предметом обсуждения на отдельной секции конференции.

А. Дж. Перлис, продолжая начатую на предыдущей конференции дискуссию, акцентировал внимание на трех вопросах:

1. Существует ли реальное различие между ИПЗ и компьютерными науками?
2. Если она существует, то нужно изучение ИПЗ как отдельной дисциплины?
3. По какой форме должны излагаться университетские курсы по ИПО?

Результаты и обсуждение

Обсуждение этих и других вопросов, включая то, достаточно ли устоявшимися являются компьютерные науки, чтобы их можно было обучать студентов, и имеют ли они определенные явные базовые принципы.

На первый вопрос А. Дж. Перлис отвечал утвердительно: «Я думаю, что все те из нас, кто работает в университетах, хорошо понимают сущность Phd. D. программы по компьютерным наукам: здоровая доза логики, теории автоматов и вычислений, чуть меньшая – численного анализа, один-два курса по углубленному программированию того или иного вида, немного искусственного интеллекта, и толику еще чего-то» (Котлярова, Сериков, 2018). Ф. Л. Бауэр отметил, что в Германии рабочая группа, которая разрабатывает программы подготовки, назвала соответствующий предмет информатикой (“Informatik”): «Мы ожидаем, что наши студенты сами сделают выбор, будут ли они специалистами по компьютерным наукам или по инженерии программного обеспечения» (Котлярова, Сериков, 2018). По мнению А. Дж. Перлиса, для ИПЗ фундаментальными являются курсы исследование операций и управление проектами – столь же фундаментальными, как и курс теории автоматов, и более фундаментальными, чем любой математический курс (Котлярова, Сериков, 2018).

Определенным продолжением этой дискуссии является другая публикация А. Дж. Перлиса, в которой он подчеркивает общественную значимость профессии инженера по программному обеспечению и насущную необходимость разработки соответствующих образовательных программ их подготовки. Специализацией таких инженеров является программное обеспечение-а именно его проектирование, производство и обслуживание. К заданным на обсуждаемых конференциях проблемным вопросам А. Дж. Перлис добавляет еще несколько:

1. Если подготовка специалистов по ИПЗ будет происходить в университете, то на какой кафедре или факультете?

2. Программа подготовки должна быть отдельной, или она может быть вариативной частью другой программы?

3. Будут ли по такой программе специалисты подготовлены для решения системных проблем, которые возникнут в будущем?

4. Почему мы говорим об инженерии, а не о науке?

Автор предлагает начинать с магистерской программы, далее распространяя ее на бакалавриат и докторат. "Цель состоит в том, чтобы сосредоточиться на известных инструментах и их эффективном использовании, а не на периодах интенсивных инноваций и открытий. ... На мой взгляд, профессиональная подготовка по инженерии программного обеспечения является сплавом математики, теории управления, компьютерных наук и практического опыта, полученного при работе с актуальными программными системами и связанными проблемами» (Магомедов, 2021).

Конференции по ИПЗ, проведенные под эгидой научного комитета, в целом определили сферу ИПЗ и пути подготовки соответствующих специалистов в учебных заведениях:

1. Методы и средства ИПЗ применяются к большим сложным программным системам, которые не могут быть созданы одним лицом или небольшим коллективом разработчиков.

2. Несмотря на свое название, ИПЗ должна включать вопросы взаимодействия программной и аппаратной составляющих компьютерной системы.

3. Целью ИПЗ является разработка программных систем с заранее определенными уровнями качества, надежности и эффективности в условиях ограничения ресурсов (временных, человеческих, материальных, программно-аппаратных и тому подобное). В связи с этим, в отличие от компьютерных наук (информатики), вопросы исследования операций и управления проектами для ИПЗ обеспечения являются фундаментальными.

4. Подготовка специалистов по ИПЗ в вузах является целесообразной на уровне бакалавриата. В процессе подготовки целесообразно сочетать теоретическую и практическую подготовку (на производстве или с применением заимствованных из производства методов разработки программного обеспечения).

5. Существенным для подготовки специалистов по ИПО является изучение артефактов, создаваемых в процессе программной инженерии: документации, программного кода, меморандумов, групповых обсуждений и др.

6. В обучении ИПЗ («технологии программирования») с самого начала остро встала проблема быстрого устаревания технологического содержания обучения, решение которой заключается в его фундаментализации через выделение базовых основ отрасли.

Последнее вызвало наибольшую дискуссию, по результатам которой было определено, что овладение основами компьютерных наук («информатики» по Ф. Л. Бауэру и «математической инженерии» по Е. В. Дейкстрею) является фундаментом профессиональной подготовки по ИПО. Тенденции развития профессиональной подготовки специалистов по инженерии программного обеспечения

В 2018 году в статье, посвященной 50-летию подготовки специалистов по ИПЗ ученые представили современную трактовку ИПЗ как «составляющей компьютерных наук, которая создает практичные, экономически выгодные решения для вычислительных задач и задач обработки информации, преимущественно путем применения научных знаний и разработки программных систем, что служат человечеству» (Нечаева, Зимина, 2016). По мнению авторов, современная ИПЗ базируется на трех группах ключевых принципов:

1. Основные концепции компьютерных наук, связаны со структурами данных, алгоритмами, языками программирования и их семантикой, анализом, вычислительностью, моделями вычислений т. д:

– абстракция предоставляет возможность контролировать сложность;

– структурирование задач часто делает их более доступными; существует ряд общих структур;

– символические репрезентации необходимы и достаточны для решения проблем, связанных с информацией;

– точные модели поддерживают анализ и прогнозирование — общие структуры задач приводят к каноническим решениям.

2. Основы инженерии, связанные с архитектурой, процессами инженерии, компромиссами и расходами, стандартизацией, качеством и гарантиями и другие составляющие, обеспечивающие подход к проектированию и решению проблем, который учитывает прагматические аспекты:

– источником инженерного качества являются инженерные решения;

– качество программного продукта зависит от верности инженера инженерному артефакту;

– инженерия требует согласования противоречивых ограничений;

– инженерные навыки улучшаются вследствие тщательной системной рефлексии опыта.

3. Социально-экономические основы, которые включают процесс создания и эволюции артефактов, а также вопросы, связанные с политикой, рынками, удобством использования и социально-экономическими воздействиями; это обеспечивает основу для формирования инженерных артефактов, которые будут соответствовать их назначению:

– ограничения на затраты и время значимы, а не просто возможны;

– технология улучшается экспоненциально, в отличие от человеческих способностей;

– успешная разработка программного обеспечения зависит от командной работы творческих людей;

– цели бизнеса и политики так же накладывают ограничения на проектирование и разработку программного обеспечения, как и технические соображения;

– функциональность программного обеспечения глубоко встроена в институциональные, социальные и организационные механизмы.

Исследователи, опираясь на практический опыт, отмечают, что ряд ошибок во время проектирования и реализации программного обеспечения допускается специалистами через слишком узкую трактовку исторически сложившихся подходов к ИПЗ. Таких «узких подходов исследователи выделяют три: «машиностроительный», «административный», «инструментальный».

«Машиностроительный» подход заключается в слишком буквальной аналогии с понятием технологии машиностроительного производства, что предполагает жестко регламентированная последовательность технологических операций, которые должны гарантировать получение продукции с заданным качеством в минимальной зависимости от индивидуальных (например, творческих) возможностей отдельных работников.

"Административный" подход акцентирует на важности различных организационно-управленческих мероприятий по внедрению тех или иных стандартов, регламентов работ, типовых или базовых программ и тому подобное.

"Инструментальный" подход является наиболее распространенным заблуждением, по мнению исследователей. Его суть заключается в том, что поиск решения проблем повышения производительности и качества результатов программирования сводится к поиску или разработке тех или иных инструментальных средств от личных до систем «сквозной» автоматизации работ и тому подобное.

Статья Б. Боэма, опубликованная более 40 лет назад, содержит все основные составляющие современной ИПЗ – несмотря на устарелость примеров, выделены ним научные принципы доказали свою жизнеспособность. В 1987 году, анализируя исторические аспекты ИПЗ (Пашаева, Кочеткова, 2018), Б. Боэм выделяет три ранние работы оказали значительное влияние на становление и развитие ИПЗ.

Первая из них обобщает опыт проектирования автоматизированной системы управления авиацией и противовоздушной обороной SAGE (Semi-Automatic Ground Environment) (Oliveira, Tereso, Machado, 2014) – наиболее амбициозного проекта информационной системы противовоздушной обороны США и Канады 1950-х гг., который объединил ведущих радарных инженеров, инженеров связи, компьютерных инженеров и инженеров по программному обеспечению. В рамках проекта SAGE была

разработана Lincoln Labs Utility System на помощь тысячам программистов, участвовавших в разработке программного обеспечения SAGE. Она включала в себя ассемблер, библиотеку и систему управления сборками, ряд полезных утилит, а также средства тестирования и отладки. Система SAGE успешно удовлетворяла техническим спецификациям примерно через один год. Ведущий разработчик SAGE Г. Д. Бенингтон (Herbert D. Benington) указывал, что ему было легко выделить тот фактор, что привел к такому успеху: «мы все были инженерами и были обучены организовать наши усилия на инженерных принципах». Он обобщил опыт разработки SAGE в описании процесса проектирования большой программной системы, состоящей из 9 фаз:

1. операционный план (operational plan) определяет широкие требования к проектированию всей системы управления, состоящий из машины, оператора и программной системы. Этот план должен быть подготовлен совместно с инженерами компьютерных систем и конечными пользователями системы;
2. эксплуатационные спецификации (machine specifications, operational specifications), которые точно определяют «передаточную функцию» системы управления. В этом представлении компьютер, его терминальное оборудование и системная программа рассматриваются как «черный ящик»;
3. программные спецификации (program specifications) описывают реализацию «черного ящика» системной программой. Эти спецификации организуют программу в подпрограммы компоненты и таблицы, указывают основные каналы межпрограммного взаимодействия, а также совместное использование машинного времени и данных каждой подпрограммы;
4. после завершения операционных и программных спецификаций, готовятся детальные спецификации кодирования (coding specification), которые определяют «передаточную функцию» каждого компонента подпрограммы в терминах обработка глобальных и локальных данных;
5. каждый компонент программируется (coding) с помощью спецификаций кодирования. В идеале этот этап должен быть простым механическим переводом; на самом деле, Программирование раскрывает несоответствие спецификаций кодирования (а иногда и операционных спецификаций);
6. после программирования каждая подпрограмма отдельно тестируется на соответствие параметров (parameter testing). На этом этапе тестирование выполняется в среде, которая имитирует соответствующие части программной системы. Каждый тест, выполненный на этой фазе, документируется в наборе спецификаций теста, который детализирует используемую среду и полученные результаты;
7. после завершения тестирования параметров подпрограмм, постепенно компонуется и проверяется вся система (assembly testing), используя сначала модельного, а затем реальные данные;
8. после завершения сборки программы, она проверяется в его операционной среде в стрессовом режиме (shakedown);
9. программа готова к работе и оценке (evaluation).

Модель Г. Д. Бенингтона явно не предполагает возвратов к предыдущим фазам (хотя в статье и упоминается о необходимости пересмотра даже первой фазы по результатам реализации спецификаций кодирования во время программирования), поэтому в дальнейшем подобные модели назвали «водопадными».

Резкое уменьшение стоимости машинного времени связано прежде всего со сменой аппаратной составляющей (уменьшение размера и энергопотребления транзисторов, увеличению времени их непрерывной работы, объединение в интегральные схемы т. д), привело к появлению подхода «сначала закодируй, а потом уже проверяй и исправляй», который и привел к обсуждаемой выше кризиса программного обеспечения.

Так, уже в процессе реализации SAGE программная составляющая системы стала более значимой, чем аппаратная: Б. Боэм указывает, что «даже в SAGE все более доминирующими становились адресованы психологам вопросы человеко-машинного взаимодействия, чем вопросы, адресованные радарным инженерам... Быстрое расширение спроса на программное обеспечение превысило предложение инженеров и математиков. Программа SAGE начала нанимать и обучать

разработке программного обеспечения специалистов по гуманитарным, социальным наукам, иностранным языкам и искусству. Для таких неинженерных специалистов ... и был гораздо более комфортный подход "кодируй и исправляй". Они часто были очень креативными, но их исправление часто привело к тяжелому в обслуживании спагетти-кода. ... Существенной в этом отношении была «хакерская культура» ..., а частыми ролевыми моделями были «программисты-ковбои». (Абрамова, Войнова, 2019).

Ройс У.У. (Winston Walker Royce) по классификации Б. Боэма относится к специалистам-эмпирикам – так, он не предлагает никаких научных принципов ИПЗ, а описывает исключительно собственный опыт разработки систем высокой сложности – программных систем для планирования, управления и послеполетного анализа миссий космических аппаратов.

Для уменьшения рисков, связанных с применением итеративной схемы проектирования, У. У. Ройс предлагает внести в нее пять дополнений:

1. этап предварительного проектирования программы между этапом определения программных требований и этапом анализа. Для реализации этого этапа У. У. Ройс предлагает:

a. начать процесс проектирования с проектировщиками программы, а не аналитиками или программистами;

b. разработать, определить и распределить режимы обработки данных даже за риска ошибок: способы обработки, функции, структуру базы данных, распределять время выполнения, интерфейсы и режимы работы с операционной системой, описать обработку ввода и вывода и определить предварительные операционные процедуры;

c. написать понятный, информативный и актуальный обзорный документ, из которого каждый участник проекта сможет получить элементарное понимание проектируемой системы;

2. обеспечение актуальности и полноты документации - по Ройсу У.У., как можно больше: "первым правилом управления разработкой программного обеспечения является безжалостное выполнение требований к документации. ... первый шаг заключается в изучении состояния документации. Если с документацию серьезная проблема, моя первая рекомендация проста: заменить менеджмент проекта; остановить все действия, не связанные с документацией; привести документацию к соответствующим стандартам. Управление программным проектом просто невозможно без очень высокой степени документирования» (Пашаева, Кочеткова, 2018);

3. «двойное» проектирование – «после документации второй важнейший критерий успеха заключается в новизне программного продукта. Если программное обеспечение разрабатывается впервые, его окончательная версия, которая поставляется клиенту, должна быть второй версии. ... Заметим, что это просто процесс, выполненный в миниатюре, масштабе, относительно небольшом по отношению к общим усилий. С помощью моделирования (руководитель проекта) может, по крайней мере, выполнить экспериментальную проверку некоторых ключевых гипотез и (слишком оптимистичных человеческих ожиданий)» (Пашаева, Кочеткова, 2018);

4. планирование, управление и мониторинг тестирования программного обеспечения. Как замечает Ройс У.У., более раннее обращение к процедурам, связанным с тестированием, необходимо потому, что этап тестирования является одним из последних этапов проектирования, а потому может стать "узким местом" проекта, преодоление проблем на котором потребует дополнительных ресурсов;

5. привлечение заказчика – «в связи с тем, проектирование программного обеспечения имеет широкие возможности для интерпретации даже после предварительных согласований, важно привлекать заказчика на формальной основе» (Котлярова, Сериков, 2018).

Заключение

Программная инженерия представляет собой современное образовательное направление для подготовки программистов и ИТ-профессионалов мирового уровня, продвинутых в области computer science и software engineering для разработки, тестирования и эксплуатации программного обеспечения и программных систем в целях реализации цифровой экономики России.

Список литературы

1. Абрамова Е.А., Войнова М.Е. CRM-система как фактор успешной реализации бизнес-процессов в современной компании // Проблемы экономики, финансов и управления производством. 2019. № 44. С. 42-46.
2. Алексашина Е.С., Горбунова О.Н. Профессиональные стандарты в образовании с учетом потребности цифровой экономики России // Социально-экономические явления и процессы. 2017. Т. 12, № 5. С. 204-209.
3. Бутенко Ю.И., Семенова Е.Л. Влияние лингвистических особенностей текстов стандартов на информационный поиск // Филологические науки. Научные доклады Высшей школы. 2019. № 4. С. 29-35.
4. Котлярова И.О., Сериков Г.Н. Партнерство субъектов образовательного процесса в непрерывной научно-исследовательской деятельности аспирантов // Вестник ЮУрГУ. Серия «Образование. Педагогические науки». 2018. Т. 10, № 2. С. 6-16.
5. Магомедов Р.М. Анализ возможностей использования платформы Saleforce CRM на российском рынке // Самоуправление. 2021. № 1 (123). С. 312-314.
6. Нечаева М.А., Зимина Л.В. Особенности информационных систем управления взаимодействием с клиентами // Экономическая среда. 2016. № 1 (15). С. 59-64.
7. Пашаева С.С., Кочеткова Н.В. Сравнительный анализ отечественных и зарубежных программ, используемых в управлении // Современные научные исследования в сфере экономики: сб. результатов науч. исследований. Киров, 2018. С. 821-824.
8. Aleem S., Capretz L., Ahmed F. Game development software engineering process life cycle: A systematic review. J. of Software Engineering Research and Development, 2016, vol. 4, art. 6. DOI: 10.1186/s40411-016-0032-7.
9. Elsakova R., Leskina J. The methods of mastering future managers' professional culture // ICERI2017 Proceedings. 2017. P. 1994-1999.
10. Murphy-Hill E., Zimmermann T., Nagappan N. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? Proc. ICSE, 2014, pp. 1-11. DOI: 10.1145/ 2568225.2568226.
11. Oliveira J., Tereso A., Machado R. An application to select collaborative project management software tools. In: New Perspectives in Information Systems and Technologies, vol. 1. AISC, 2014, vol. 275, pp. 467-476. DOI: 10.1007/978-3-319-05951-8_44.

Conditions for the formation of competence in the specialty "software engineering"

Bogdan D. Terekhin
student
Far Eastern Federal University
Vladivostok, Russia
terehin@dvfu.ru
0000-0000-0000-0000

Denis A. Startsev
student
Far Eastern Federal University
Vladivostok, Russia
starcev@dvfu.ru
0000-0000-0000-0000

Danila A. Razdobarov
student
Far Eastern Federal University
Vladivostok, Russia
razdovarov@dvfu.ru
0000-0000-0000-0000

Received 25.10.2022
Accepted 18.11.2022
Published 01.12.2022

10.25726/q1125-2979-0698-u

Annotation

Since 2012, among the priority areas of education and science for teaching students and graduate students, internships of scientific and scientific and pedagogical workers in leading higher educational institutions and scientific institutions abroad, which relate to computer science and computer technology, three are software engineering, software systems and software engineering - belong to the same specialty: 121 - Software engineering. In addition, a significant part of other priority areas (mathematical and computer modeling; information and communication technologies; artificial intelligence systems; system programming, etc.) are tangent to it. Technologies and tools for developing software products and systems are identified as one of the priority areas for scientific research and scientific and technical development in Russia for the period up to 2020. These and a number of other legislative initiatives of our state are evidence of an urgent public need for competent software engineering specialists trained on the basis of the best world standards and advanced foreign experience and capable of designing, testing, implementing and commercializing innovative IPO technologies. It would be appropriate to start the analysis of the world experience in the training of IPT specialists with a retrospective review of the evolution of the very concept of "Software Engineering" and the main stages in the development of this industry. The purpose of the article is to analyze the main stages in the development of the IPP as a field of knowledge, highlight the fundamental components of the training of future software engineers and determine the trends in the development of this industry for the next decade.

Keywords

software engineering, competence, specialty, formation.

Reference

1. Abramova E.A., Vojnova M.E. CRM-sistema kak faktor uspeshnoj realizacii biznes-processov v sovremennoj kompanii // Problemy ekonomiki, finansov i upravleniya proizvodstvom. 2019. № 44. S. 42-46.
2. Aleksashina E.S., Gorbunova O.N. Professional'nye standarty v obrazovanii s uchetom potrebnosti cifrovoj ekonomiki Rossii // Social'no-ekonomicheskie yavleniya i processy. 2017. T. 12, № 5. S. 204-209.
3. Butenko YU.I., Semenova E.L. Vliyanie lingvisticheskih osobennostej tekstov standartov na informacionnyj poisk // Filologicheskie nauki. Nauchnye doklady Vysshej shkoly. 2019. № 4. S. 29-35.
4. Kotlyarova I.O., Serikov G.N. Partnerstvo sub"ektov obrazovatel'nogo processa v nepreryvnoj nauchno-issledovatel'skoj deyatel'nosti aspirantov // Vestnik YUUrGU. Seriya «Obrazovanie. Pedagogicheskie nauki». 2018. T. 10, № 2. S. 6-16.
5. Magomedov R.M. Analiz vozmozhnostej ispol'zovaniya platformy Saleforce CRM na rossijskom rynke // Samoupravlenie. 2021. № 1 (123). S. 312-314.
6. Nechaeva M.A., Zimina L.V. Osobennosti informacionnyh sistem upravleniya vzaimodejstviem s klientami // Ekonomicheskaya sreda. 2016. № 1 (15). S. 59-64.

7. Pashaeva S.S., Kochetkova N.V. Sravnitel'nyj analiz otechestvennyh i zarubezhnyh programm, ispol'zuemyh v upravlenii // Sovremennye nauchnye issledovaniya v sfere ekonomiki: sb. rezul'tatov nauch. issledovanij. Kirov, 2018. S. 821-824.

8. Aleem S., Capretz L., Ahmed F. Game development software engineering process life cycle: A systematic review. J. of Software Engineering Research and Development, 2016, vol. 4, art. 6. DOI: 10.1186/s40411-016-0032-7.

9. Elsakova R., Leskina J. The methods of mastering future managers' professional culture // ICERI2017 Proceedings. 2017. P. 1994-1999.

10. Murphy-Hill E., Zimmermann T., Nagappan N. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? Proc. ICSE, 2014, pp. 1-11. DOI: 10.1145/ 2568225.2568226.

11. Oliveira J., Tereso A., Machado R. An application to select collaborative project management software tools. In: New Perspectives in Information Systems and Technologies, vol. 1. AISC, 2014, vol. 275, pp. 467-476. DOI: 10.1007/978-3-319-05951-8_44.