

УДК: 378.147

*Мезенцева Д.А.,  
кандидат педагогических наук,  
ведущий инженер отдела образовательных технологий  
Университет ИТМО, г. Санкт-Петербург*

## **ОБЗОР СОВРЕМЕННЫХ ПОДХОДОВ К ПОДГОТОВКЕ СПЕЦИАЛИСТОВ В ОБЛАСТИ ПРОГРАММНОЙ ИНЖЕНЕРИИ**

***Аннотация:** в статье дан обзор актуальных практик в области подготовки программных инженеров в высшей школе. Рассмотрены шесть кейсов, описание которых включает характеристику образовательных проектов для обучения программных инженеров в зарубежных вузах, проблем обучения, на решение которых направлены данные образовательные проекты, а также тех современных подходов, методов и приемов, которые используются для обучения будущих IT-специалистов в рамках данных проектов.*

***Ключевые слова:** программная инженерия, профессиональные навыки, надпрофессиональные навыки, учебные подход, практикоориентированное обучение*

***Abstract:** the paper provides the overview of the modern teaching practices in the sphere of Software Engineering. The six cases presented in the article contain the description of educational projects for prospective software engineers from different foreign universities, problems that are tackled within the framework of these projects as well as actual teaching approaches, methods and techniques involved.*

***Keywords:** software engineering, hard skills, soft skills, teaching approaches, practice oriented education.*

Подготовка специалистов в области программной инженерии является непростой задачей в связи с постоянно возрастающей ролью программного обеспечения в самых разных сферах жизни общества и, как следствие, с интенсивным развитием этой отрасли, совершенствованием инструментальных средств разработки, увеличивающимся запросом на обучение большего количества высоко квалифицированных IT-специалистов.

В настоящий момент к знаниям будущих разработчиков предъявляются высокие требования. В соответствии с "Руководством к своду знаний по программной инженерии" (SWEBOOK, 2014) программный инженер должен обладать следующими знаниями, умениями и навыками: требования к программному обеспечению (ПО), проектирование, конструирование, тестирование и сопровождение ПО, управление конфигурацией, управление в программной инженерии, процессы программной инженерии, инструменты и методы программной инженерии, качество ПО [12]. Немаловажным для IT-специалиста является также изучение следующих дисциплин: математика, информатика, разработка компьютеров, общий менеджмент, управление проектами, управление качеством, системное проектирование [3, с.94].

Очевидно, что для обучения такому сложному комплексу знаний, умений и навыков в условиях бурно развивающейся IT-отрасли, преподавателям вузов, готовящим специалистов для данной индустрии, необходимо отслеживать новейшие тенденции в обучении по вышеперечисленным дисциплинам и использовать наиболее эффективные передовые учебные подходы и методы.

В статье дан обзор современных проблем и подходов в преподавании программной инженерии на основе шести кейсов из различных зарубежных вузов.

Проблемы, с которыми сталкиваются преподаватели зарубежных вузов, представленных в рассматриваемых кейсах, в основном схожи с проблемами

отечественных учебных заведений, готовящих IT-специалистов [2, с.292; 4, с.110; 8].

Эти проблемы можно объединить в две большие группы:

А. Проблемы, связанные с недостатком системности и сбалансированности обучения:

- общий низкий уровень знаний, умений и навыков выпускников;
- нарушен баланс в обучении профессиональным и надпрофессиональным навыкам;
- сложности в обучении студентов с разным уровнем подготовки;
- отсутствие системности в работе с наставниками, недостаточное количество консультаций и методических инструкций.

Б. Проблемы, связанные с недостаточной практикоориентированностью обучения:

- неспособность студентов применить полученные знания на практике;
- в условиях прохождения стажировок и практик студенты не могут познакомиться со всем многообразием производственных проблем;
- работа над примитивными проектами, которые неинтересны студентам и которые не могут быть использованы реальными заказчиками;
- фокус на одной какой-либо стадии разработки программного обеспечения;
- недостаток контакта с реальными заказчиками;
- недостаток активного участия в профессиональных семинарах, хакатонах, конкурсах и т.д.

Для решения данных проблем все чаще используются такие современные подходы и методы обучения, которые направлены на "активизацию мышления обучаемых", для которых характерны "высокая степень интерактивности, мотивации и эмоционального восприятия учебного процесса", которые позволяют "активизировать и развивать познавательную и творческую деятельность обучаемых; повышать результативность учебного

процесса; формировать и оценивать профессиональные компетенции, особенно в части организации и выполнения коллективной работы" [1, с.5]

В данной статье рассматриваются такие подходы, как 1) обучение в сотрудничестве, 2) ситуационное обучение, 3) смешанное обучение, 4) обучение через преодоление трудностей, 5) взаимное обучение, 6) когнитивное наставничество, 7) "спортивный" подход.

Ниже представлены шесть кейсов зарубежных университетов. Для каждого кейса описаны проблемы при подготовке программных инженеров, образовательные проекты, подходы, методы и соответствующие приемы, используемые для решения обозначенных проблем в рамках данных проектов.

*Таблица 1.*

### **Образовательный проект Университета Техаса**

<p><b>Университет Техаса, США</b> [9, с.52 ]</p> <p><b>Методы:</b></p> <ul style="list-style-type: none"><li>• Обучение в сотрудничестве</li><li>• Ситуационное обучение</li></ul>	<p><b>Проблемы обучения</b></p> <p>Проблемы командообразования: слабо развиты надпрофессиональные навыки, неравный вклад участников команды, разный уровень знаний обучающихся.</p>
<p style="text-align: center;"><b>Описание образовательного проекта в Университете Техаса</b></p> <p>Проект по программной инженерии с фокусом на развитии навыков командообразования рассчитан на 2 семестра.</p> <p><b>Цели проекта:</b></p> <p>1) сформировать студенческие команды, в которых работа построена на принципах обучения в сотрудничестве;</p>	

2) создать аутентичную среду, в которой у студентов будет возможность оценить свои знания на практике.

### **Результаты проекта:**

Семестр 1: техническое исследование, разработка прототипа, выполнение технического задания на разработку.

Семестр 2: разработка архитектуры, проектная документация, программный код, план тестирования.

### **Подходы**

**Обучение в сотрудничестве:** подход характеризуется взаимной ответственностью обучающихся за результаты работы, взаимопомощью, развитием надпрофессиональных навыков (командообразование, лидерство, менеджмент проектов, разрешение конфликтов и т.д.), рефлексией по поводу работы команды.

**Ситуационное обучение:** подход заключается в воспроизведении аутентичных ситуаций, обращении к лучшим практикам экспертов, использовании различных ролей в процессе обучения.

### **Методы и приемы обучения**

1. **Резюме:** формирование команд происходит на основе резюме, которые студенты создают для приема на выбранную позицию в проекте (например, ведущий разработчик, проектировщик системы, аналитик).

2. **Совещания:** преподаватель выдает студентам руководство для проведения совещаний и всю необходимую документацию; руководитель (из числа обучающихся) следит за фокусом дискуссии; все студенты попеременно выполняют роль секретаря, который ведет протоколы; студенты применяют фасилитационные техники: поясняющие вопросы, обращения к тем, кто не успел высказаться, исследование высказанных идей, подведение итогов и уточнение мнения всех участников.

3. **Ролевые дискуссии:** студенты не только занимают ту или иную "позицию" в проекте, но также выполняют роли, которые им предписываются в рамках проведения совещаний или выполнения групповых заданий (например, секретарь, стратег, критик, резюмирующий, проверяющий).

4. **Рефлексия:** участники команды в конце подводят итоги, называя три достижения команды и одну проблему.

5. **Оценивание:** оценивается как индивидуальный вклад каждого участника, так и работа всей команды, чтобы студенты понимали, что оценка их труда связана с взаимопомощью; индивидуальные и командные результаты работы команды фиксируются руководителем в журнале, по которому он регулярно отчитывается преподавателю.

6. **Опросы:** преподаватель получает обратную связь от участников и предоставляет результаты руководителям, чтобы они внесли предложения по улучшению работы своих команд.

*Таблица 2.*

**Образовательный проект Университета Карнеги-Меллона**

<p><b>Университет</b> <b>Карнеги-Меллона,</b> <b>США [11, с.21]</b></p> <p><b>Подход:</b></p> <ul style="list-style-type: none"> <li>• Смешанное обучение</li> </ul>	<p><b>Проблемы обучения</b></p> <p>Проблема создания фундаментального курса:</p> <ul style="list-style-type: none"> <li>• баланс профессиональных и надпрофессиональных навыков,</li> <li>• практикоориентированность,</li> <li>• обучение студентов с разным уровнем знаний.</li> </ul>
<p><b>Описание образовательного проекта</b> <b>в Университете Карнеги-Меллона</b></p>	

Курс в формате смешанного обучения по основам программной инженерии рассчитан на 14 недель, включает 11 модулей и командный проект продолжительностью 1 семестр.

Курс включает такие темы, как Командообразование и Технологии, Архитектура и Дизайн, Конструирование, Тестирование и Качество, Требования.

**Цели проекта:**

- 1) оптимальное использование времени обучения в аудитории;
- 2) более интенсивное взаимодействие между студентами;
- 3) баланс инженерных компетенций и компетенций менеджера.

**Результаты проекта:**

- 1) умение сравнивать различные подходы и методы для создания систем программного обеспечения (ПО);
- 2) умение применять и сочетать различные техники для разработки требований, проектирования, создания и тестирования ПО;
- 3) умение использовать современные инструменты и среды для разработки;
- 4) знание основных понятий и концепций в области моделирования;
- 5) умение планировать, оценивать, управлять проектами;
- 6) умение применять лучшие практики в области программной инженерии для разработки проектов, которые позволяют достичь сбалансированного результата с учетом требований заказчика и компромиссных инженерных решений.

**Подходы**

**Смешанное обучение:** образовательный подход, который совмещает очное и онлайн-обучение. Смешанное обучение предполагает элементы самостоятельного контроля обучающимся образовательного маршрута, времени, места и темпа обучения.

**"Перевернутый класс"**: это модель смешанного обучения, при котором студенты изучают теорию перед занятием при помощи материалов и видео-лекций, а на занятии применяют теоретические знания на практике в ходе упражнений под руководством преподавателя.

### **Методы и приемы обучения**

1. **Видео-лекции**: студенты просматривают видео-лекции по теме предстоящего занятия.

2. **Опросы**: преподаватели проводят в аудитории опросы по видео-лекциям для оценивания домашней работы, обсуждают ошибки и затруднения, приводят примеры правильных ответов.

3. **Мини-лекции**: мини-лекции в начале занятия предваряют работу в группах; в ходе таких лекций преподаватели дают дополнительные разъяснения; мини-лекции компенсируют недостаток взаимодействия с преподавателем при просмотре видео-лекций.

4. **Работа в группах**: студенты работают в группах над заданиями; преподаватели перемещаются от группы к группе, осуществляя руководство; в конце занятия студенты презентуют результаты и загружают их в систему в электронном виде; студенты сравнивают и обсуждают результаты своей работы, объясняют, почему их решения различаются.

#### **5. Задания на рефлексия:**

- ретроспективная дискуссия, во время которой обсуждаются следующие типичные вопросы: что было хорошо? где были проблемы? что можно исправить?
- мини-выступления членов команды – каждый участник отвечает на 3 вопроса: личный вклад на данный момент; планируемые действия; проблемы.

<p>6. <b>Tech Talks:</b> семинары по отдельным приложениям, языкам, концепциям, используемым для данного конкретного проекта.</p> <p>7. <b>Show-N-Tell:</b> презентации студентов, во время которых они рассказывают о тех инструментах или практиках, которыми им удалось овладеть; студенты получают дополнительные баллы за эти выступления.</p> <p>8. <b>Demo:</b> демонстрационные сессии, во время которых студенты демонстрируют свои разработки в рамках проекта.</p>
---

*Таблица 3.*

### Образовательный проект Университета Лафборо

<p><b>Университет</b> <b>Лафборо,</b> <b>Великобритания</b> <b>[7, с.209]</b></p> <p><b>Подход:</b></p> <ul style="list-style-type: none"> <li>• Обучение через преодоление трудностей</li> </ul>	<p><b>Проблемы обучения</b></p> <p>Недостаток опыта:</p> <p>за время обучения студенты могут выполнить ряд проектов, но не получить необходимый опыт, потому что в ходе обучения они не сталкиваются со всем многообразием проблем, с которыми обычно имеют дело профессионалы на рабочем месте.</p>
<p style="text-align: center;"><b>Описание образовательного проекта</b> <b>в Университете Лафборо</b></p> <p>Курс рассчитан на 8-10 недель обучения (120-160 студентов, разбитые на группы по 4-10 человек). Курс включал лекции, занятия для повторения материала и работу над проектом.</p>	

**Цели проекта:** научить студентов справляться с такими типичными проблемами на рабочем месте, как меняющиеся задачи, сложные клиенты, сжатые сроки.

**Результаты проекта:** организационные навыки, навыки проектирования, умение проанализировать требования и понять задачу, умение вести переговоры, умение находить компромиссы, умение планировать, адаптироваться, умение оценить качество.

### Подходы

**Обучение через преодоление трудностей:** подход базируется на предположении о том, что лучше всего усваивается то, что выучено на собственных ошибках в процессе преодоления трудностей. Основные положения подхода: студентам нужно не только разрешать ошибаться, но и создавать такие условия, в которых они будут ошибаться, преодолевая трудности, чтобы извлечь из этого максимальную пользу и быть готовыми к подобным ситуациям в дальнейшем.

### Методы и приемы обучения

**1. Смена команды:** студентов переводят из одной команды в другую несколько раз за время работы над проектом; это сигнал для обучающихся, что в их команде идет что-то не так; студенты учатся адаптироваться, четко выполнять свои функции и понимать как они дополняют друг друга.

**2. "Смена настроения":** заказчик постоянно меняет сроки и приоритеты; студенты учатся письменно фиксировать все переговоры, полагаться только на себя, проверять информацию и проявлять терпение.

**3. Смена методов работы:** менеджер может назначать или отменять совещания; новый менеджер может поменять правила; студенты учатся

планированию, умению адаптироваться к внешним неблагоприятным условиям, работе над качеством.

**4. Запрет сверхурочной работы:** рабочие часы ограничиваются, и сроки нельзя передвинуть; студенты учатся навыкам тайм-менеджмента.

**5. Заказчики с разными характерами:** создание ситуации, в которой заказчики могут проявлять себя по-разному; студенты учатся общаться дипломатично и находить баланс между необходимостью выполнить задачу и своими реальными возможностями.

**6. Обновление ПО:** студентам объявляют, что замена программного обеспечения не повлияет на их работу, но в результате возникают сложности; студенты учатся навыкам тайм-менеджмента, а также взвешиванию рисков, когда каждое новое приобретение может повлечь временные и денежные затраты.

*Таблица 4.*

### **Образовательный проект Университета Тасмании**

<p><b>Университет</b> <b>Тасмании, Австралии</b> <b>[6, с.41]</b></p> <p><b>Подход:</b></p> <ul style="list-style-type: none"><li>• Взаимное обучение</li></ul>	<p><b>Проблемы обучения</b></p> <p>Недостаток мотивации:</p> <p>У студентов низкая мотивация к тестированию ПО. Им интересно разработать продукт, но они недостаточно заботятся о его качестве.</p>
<p style="text-align: center;"><b>Описание образовательного проекта</b> <b>в Университете Тасмании</b></p> <p>Взаимное тестирование студентов в рамках курса по программной инженерии: студенты в командах из 4-5 человек разрабатывают ПО для</p>	

реальных заказчиков и перед сдачей проекта тестируют продукты в парах или командах.

Тестирование проводится в рамках 3-часовых семинаров, которые проходят несколько раз в течение семестра. Участие в тестировании влияет на финальную оценку. Студенты оценивают не только созданный продукт, но также исполнение своих ролей: разработчики оценивают тестировщиков, тестировщики оценивают представителя разработчиков как менеджера проекта.

**Цели проекта:** выяснить, какой формат тестирования мотивирует студентов, демонстрирует необходимость тестирования и обеспечивает более высокое качество разрабатываемого ПО.

**Результаты проекта:** формирование мотивации, повышение качества разрабатываемого ПО, развитие навыков тайм-менеджмента, развитие чувства общности и интереса к работе других, усиление взаимодействия и развитие сотрудничества, получение знаний не только от преподавателя, но и друг от друга.

### **Подходы**

**Взаимное обучение:** подход, базирующийся на идее массового сотрудничества, идеологии открытых образовательных ресурсов, в сочетании с сетевой организацией взаимодействия участников.

### **Методы и приемы обучения**

#### **Оценивание в парах:**

студентам предварительно объясняют, что поиск дефектов в коде одногруппника не только не повлечет снижения оценки, а наоборот улучшит качество разрабатываемого ПО и повысит оценку;

оценивание кода включает: поиск проблемы, предложение корректив, предложение более простого варианта кода, добавление иных комментариев.

**Оценивание в командах:**

- а) обзор и демонстрация ПО представителем разработчиков;
- б) тестирование и написание заключения разработчиками;
- в) использование различных ролей разработчиками: модератор, инспектор, секретарь, представитель команды разработчиков;
- г) ведение журнала, в котором отмечаются дефекты ПО;
- д) описание положительных сторон проекта;
- е) взаимное оценивание тестируемых и представителей команды разработчиков.

*Таблица 5.*

**Образовательный проект Университета Карнеги-Меллона**

<p><b>Университет Карнеги-Меллона, США [5, с.162]</b></p> <p><b>Подход:</b></p> <ul style="list-style-type: none"><li>• Когнитивное наставничество</li></ul>	<p><b>Проблемы обучения</b></p> <p>Проблема использования тех или иных методик обучения наставниками:</p> <p>студентам не нравится обучение, в ходе которого наставники помогают им выполнять задания исключительно при помощи наводящих вопросов; требуется более структурированное обучение.</p>
<p><b>Описание образовательного проекта в Университете Карнеги-Меллона</b></p>	

Использование приемов когнитивного наставничества для обучения магистров в области программной инженерии. Дисциплины: Анализ требований, Человеко-машинное взаимодействие, Менеджмент проектов.

### **Подходы**

**Когнитивное наставничество:** подход, основывается на идее о том, что эксперты в своей области часто игнорируют процессы внутреннего мышления, которые происходят при использовании того или иного комплексного навыка, поэтому преподаватель должен сделать видимыми процессы внутренних размышлений при выполнении конкретных процедур.

### **Методы и приемы обучения**

1. **Моделирование:** преподаватель демонстрирует выполнение задания обучающимся, проговаривает свои мысли и действия; в итоге, у обучающихся формируется модель поведения, которую они используют для выполнения задания, имитируя действия преподавателя.

2. **Консультирование:** преподаватель наблюдает за выполнением задания и дает обратную связь.

3. **Создание опор:** это поддержка, которую преподаватель оказывает студенту в той ли иной форме: образцы, шаблоны, руководства, дополнительные вопросы, рекомендации. По мере овладения навыком доля поддержки снижается.

4. **Артикуляция:** это методы, которые побуждают студентов описывать свои знания, логику, процесс решения задачи (серия уточняющих вопросов, "мысли вслух", критическое оценивание работы одногруппников).

5. **Рефлексия:** анализ и сравнение опыта "эксперта" и "новичка".

6. **Исследование:** преподаватель дает обучающимся пространство для самостоятельного поиска и обучает студентов стратегиям исследования.

## Образовательный проект Гавайского университета Маноа

<p><b>Гавайский университет Маноа, США</b> [10, с.41]</p> <p><b>Подход:</b></p> <ul style="list-style-type: none"> <li>• "Спортивный" подход</li> </ul>	<p><b>Проблемы обучения</b></p> <p>Проблема низкого уровня навыков программирования:</p> <ul style="list-style-type: none"> <li>• слабое владение языками, инструментами и технологиями;</li> <li>• неумение работать в сжатые сроки в ситуации стресса;</li> <li>• студенты не участвуют в профессиональных семинарах, хакатонах, конкурсах и т.д.;</li> <li>• обучающиеся оставляют профессию после выпуска.</li> </ul>
<p style="text-align: center;"><b>Описание образовательного проекта в Гавайском университет Маноа</b></p> <p>Курс по программной инженерии организован не по тематическому принципу, а на основе перечня навыков.</p> <p><b>Цели проекта:</b> научить студентов производить ПО высокого качества для реальных клиентов.</p>	
<p style="text-align: center;"><b>Подходы</b></p> <p><b>"Спортивный" подход:</b> подход, в соответствии с которым весь курс организуется на основе перечня навыков. Для тренировки каждого навыка студенты смотрят видео (5-20 мин), которые демонстрируют решение преподавателем какой-либо практической задачи. Преподаватель выполняет задание быстро, правильно, попутно объясняя выбор инструментов и</p>	

техники проектирования. Студенты повторяют "тренировку" до тех пор, пока не научатся делать это правильно и быстро.

### **Методы и приемы обучения**

**1. "Перевернутый класс":** студенты смотрят видео и знакомятся с материалами онлайн перед занятием.

**2. Тренировка:** студенты выполняют задания на скорость во время занятия индивидуально или в командах; задание в аудитории выполняется на ту же тему, что и домашнее задание с видео; если студент выполняет задание неправильно или не заканчивает вовремя, он не получает оценку; результаты выполнения заданий публикуются.

Очевидно, все перечисленные подходы, методы и приемы направлены как на решение проблем, связанных с недостаточной системностью и сбалансированностью обучения, так и проблем, связанных с недостаточной практикоориентированностью образовательных программ для будущих IT-специалистов. Перечисленные практики ориентированы как на развитие предметных навыков (например, проектирование, тестирование, сопровождение и управление конфигурацией ПО), так и надпредметных навыков (например, общий менеджмент, управлением проектами, управление качеством).

Использование таких приемов, как отбор в команду по резюме, имитация производственных совещаний, организация дискуссий по ролям (обучение в сотрудничестве, ситуационное обучение) помогает приблизить образовательный процесс к реальным рабочим условиям.

Применение смешанного обучения позволяет выделить больше времени на занятии для практических заданий. Например, можно чаще проводить семинары по отдельным приложениям, языкам и концепциям (tech talks), организовывать обмен лучшими практиками (show-n-tell) и демонстрации

студенческих разработок (demo), благодаря которым обучающиеся смогут познакомиться с большим количеством инструментов.

Обучение через трудности в представленном проекте в большей степени ориентировано на развитие надпрофессиональных навыков (управление временем, разрешение конфликтов, навыки командообразования и т.д.). Однако данный подход можно применять и для формирования профессиональных навыков обучающихся, для того чтобы познакомить их с наибольшим количеством сложных производственных ситуаций. Безусловно, все трудности не могут быть воссозданы в процессе выполнения проекта, но информирование обучающихся о тех проблемах и ошибках, с которыми им придется столкнуться в своей будущей трудовой деятельности представляется важным для подготовки компетентного специалиста в любой области.

Использование таких подходов, как взаимное обучение и когнитивное наставничество, стимулирует студентов к более глубокой проработке учебного материала и овладению профессиональными навыками на более высоком уровне в сотрудничестве как с наставниками, так и друг с другом.

Наконец, "спортивный" подход помогает обучающимся овладеть не только профессиональными навыками программирования, но также ориентирует их на работу в реальных производственных условиях, когда приходится работать в ситуациях стресса и нехватки времени.

Таким образом, в данной статье мы проанализировали ряд зарубежных практик, которые могут быть актуальны не только для обучения программных инженеров, но и для представителей других специальностей. Однако при подготовке будущих IT-специалистов применение практикоориентированных подходов и активных методов обучения имеет решающее значение в связи со стремительным развитием сферы информационных технологий и постоянно растущими требованиями и запросами работодателей. Несомненно, инновационные подходы и методы не могут полностью заменить

традиционное обучение, но, на наш взгляд, они могут дополнить его, сделав более сбалансированным, практикоориентированным и разнообразным.

### Список литературы:

1. Зарукина Е.В., Логинова Н.А., Новик М.М. Активные методы обучения: рекомендации по разработке и применению. СПб.: СПбГИЭУ, 2010. 59 с.
2. Казакова И.А. Программная инженерия: новые технологии подготовки специалистов в области разработки программного обеспечения / И.А. Казанцева // Университетское образование. Сборник статей XVI международной научно-методической конференции. Под редакцией В.И. Волчихина, Р.М. Печерской. – 2012. – С.93-94.
3. Личаргин Д.В., Кузнецов А.С., Царев Р.Ю. Активные методы обучения в рамках инициативы CDIO по направлению "Программная инженерия" / Д.В. Личаргин, А.С. Кузнецов, Р.Ю. Царев // Современные проблемы науки и образования. – 2014. – №3. С.292-297.
4. Ребус Н.А., Поколодина Е.В. О современных интерактивных методах подготовки ИТ-специалистов в контексте цифровой экономики / Н.А. Ребус, Е.В. Поколодина // Этап: экономический теория, анализ, практика. – 2020. – №3. С.110-123.
5. Bareiss R, Radley M. "Coaching via Cognitive Apprenticeship", in SIGCSE '10: Proceedings of the 41st ACM Technical Symposium on Computer Science Education, Milwaukee, USA, 2010.
6. Clark N. "Peer Testing In Software Engineering Projects", in Sixth Australasian Computing Education Conference (ACE 2004), Dunedin, New Zealand, 2004.
7. Dawson R. "Twenty Dirty Tricks to Train Software Engineers", in ICSE '00 Proceedings of the 22nd International Conference on Software Engineering, Limerick, Ireland, 2000.
8. Ellis H, Demurjian SA, Naveda JF (2009). Software engineering: Effective teaching and learning approaches and practices.

9. In: L. Yu (ed.), Developing communities of practice to prepare software engineers with effective team skills, Hershey PA: Engineering Science Reference pp. 52-70. Gates AQ, Villa EY, Salamah S (2014).
10. Johnson P. Design and evaluation of an "athletic" approach to software engineering education. 2019, J ACM Transactions on Computing Education 19(4):1-25.
11. Steinbach-Nordmann S. "Applying Blended Learning In An Industrial Context – An Experience Report", in Proceedings of the EC-TEL06 Workshops, Crete, Greece, 2006.
12. SWEBOK V3.0 (2014). Guide to software engineering body of knowledge.