

Формирование профессиональных компетенций современного разработчика ПО

В. В. Кулямин, В. А. Омельченко, О. Л. Петренко
{kuliamin, vitaliy, olga}@ispras.ru

Аннотация. Разработка программного обеспечения (ПО) состоит не только из инженерно-технических видов деятельности. Она включает также когнитивные и социальные аспекты, адекватное внимание к которым не менее важно для успеха, а в сложных проектах играет решающую роль. Важность этих аспектов также повышается в тех проектах, где используются новые подходы к разработке ПО, включая формальные методы.

Поэтому для обеспечения успешного применения подходов такого рода, участники таких проектов должны обучаться не только используемым в них техникам, но и навыкам социального характера, позволяющим прилагать их к практическим задачам в рамках определенного контекста. Серьезную помощь в этом могут оказать курсы и тренинги на основе *активных методов обучения*, поскольку они гармонично сочетают в себе обучение обоим видам необходимых умений — содержательно такой курс учит формальным методам, а форма процесса обучения способствует развитию необходимых когнитивных и социальных навыков.

1. Введение

Развитие современной науки и техники связано с использованием информационных технологий в различных областях. Эффективность информационных технологий и качество их результатов во многом определяются характеристиками используемого программного обеспечения (ПО). В настоящее время большинство требований к этим характеристикам формируются не самими программистами, а другими группами пользователей информационных технологий, которые чаще всего не знакомы с особенностями и спецификой работы программистов. Поэтому требования к ПО разнообразны и часто противоречивы. Они описывают такие его характеристики, как надежность, понятность интерфейса, удобство обучения работе с ПО, его пригодность для использования в заданной предметной области, производительность, набор платформ, на которых оно должно работать, и т.д.

Для удовлетворения всех этих требований необходим адекватный анализ потребностей пользователей и ограничений предметной области в рамках каждого конкретного проекта по разработке ПО, и, конечно же, нужны

специалисты, умеющие проводить такой анализ. Чтобы обеспечить стабильное развитие отрасли производства ПО на дальнюю перспективу и соответствие ее возможностей нуждам современного общества необходимо готовить специалистов, владеющих необходимыми знаниями и навыками для решения задач такого рода.

Обычно разработка программного обеспечения описывается как деятельность, связанная с решением технических задач и увязкой этих решений с необходимыми экономическими показателями производимых продуктов и услуг. Практика показывает, что есть неучитываемые таким подходом, но играющие ключевую роль в успешности проектов виды человеческой деятельности, включенные в разработку ПО. Решения, принимаемые всеми разработчиками на различных этапах, и способность членов проектной команды эффективно обмениваться информацией как внутри команды, так и с другими заинтересованными лицами, играют в успешности проекта *большую* роль, чем используемые в нем техники разработки. Навыки принятия решений в условиях нехватки информации и умение эффективно общаться важны не только для людей, играющих ключевые роли в проекте, таких как его руководитель, бизнес-аналитик или архитектор, но и для обычных разработчиков. Много раз в ходе проекта им приходится принимать мини-решения, о том, реализовать ли некую функциональность в виде одного компонента или двух, использовать ли тот или иной алгоритм, и, конечно же, им очень часто приходится общаться между собой, с архитектором, тестировщиками и руководителем проекта, а иногда и с пользователями.

Сферы деятельности, в которых должен разбираться разработчик ПО для создания программ высокого качества, представлены на Рис. 1. Навыки, необходимые любому разработчику, могут быть (не слишком строго) разделены на три вида.

- *Технические навыки*, которые чаще всего рассматриваются как единственные необходимые. Они характеризуют знание разработчиком основных методов и техник разработки программ и умение использовать их на практике в контексте решаемых на практике задач.
- *Когнитивные навыки*, связанные с пониманием требований и ограничений в рамках новой предметной области, умением отделять существенное от несущественного для данной задачи, упорядочиванием задач по их приоритетности, построением решений на основе неполной информации о требованиях и выбором из них наиболее подходящего.
- *Социальные навыки*, заключающиеся в умении поддерживать эффективное общение с другими людьми, принимать цели проекта, умении работать в команде и координировать свою работу с нуждами ее членов. Социальные навыки также определяют стиль общения человека — то, как он/она слушает других людей, реагирует на услышанное, участвует в обсуждениях и выдвигает аргументы в пользу собственной точки зрения.

Внимание к социальным навыкам было, наверное, впервые привлечено осознанием в 50-е-60-е годы в США того факта, что большинство людей живут и работают в группах, но чаще всего не отдают себе отчета в том, как они в них участвуют, какими их видят другие люди, какие реакции вызывает у окружающих их поведение. Одной из первых работ, в которых эта идея развивалась в приложении к разработке ПО, является книга Ф. Брукса «Мифический человек-месяц» [1].

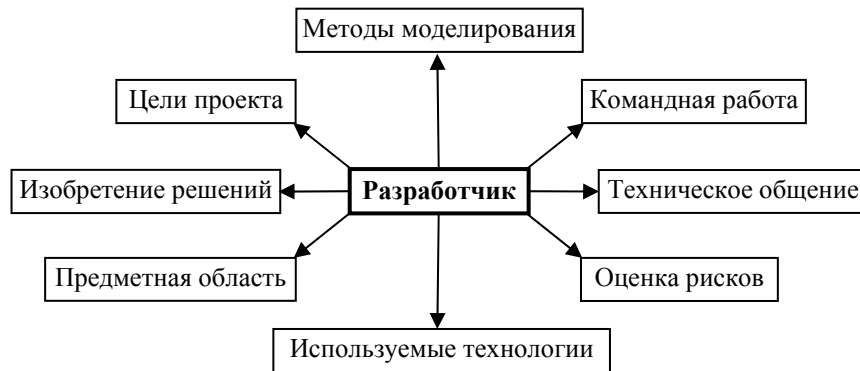


Рис. 1. Области, в которых должен разбираться разработчик современного ПО.

Проблемы овладения такими навыками достаточно актуальны в более широком контексте. В частности, меморандум Европейской Комиссии о постоянном обучении [2] подчеркивает важность для эффективного обучения таких социальных умений, как умение действовать уверенно, умение направлять собственные действия на достижение нужных результатов и умение принимать рискованные решения, а также таких когнитивных навыков, как способность учиться, приспосабливаться к изменяющемуся окружению, быстро воспринимать необходимые новые навыки и находить нужную информацию в широком информационном потоке, обрушивающемся на каждого включенного в общественную жизнь человека в наше время.

Обычно эти навыки считают необходимыми только для тех студентов, чья будущая профессиональная деятельность связана с общением с людьми, т.е. для обучающихся на менеджеров, учителей, социальных работников и пр. Но в тех случаях, когда в проекте используются новые подходы к разработке ПО, например, формальные методы, такие навыки становятся критическими как для успеха проекта самого по себе, так и для успешного приложения новых техник.

Это вызвано значительно более сложными познавательными проблемами в рамках таких проектов — нужно не только вникнуть в предметную область и понять задачи, которые предстоит решить, но и изложить полученные знания в

новой, необычной форме, необходимой для эффективного использования формальных подходов. Кроме того, само понимание задач должно быть гораздо более глубоким и детальным, иначе формальные методы становятся гораздо тяжелее традиционных в использовании. А число микро-решений, которые необходимо принять, становится значительно больше.

Серьезные социальные навыки необходимы, чтобы сделать эту работу понятной и полезной для различных заинтересованных лиц и других разработчиков, не использующих формальные методы напрямую. Применяя новые техники, нужно помнить, что люди обычно не любят иметь дело с новыми вещами, что нужно уметь разговаривать с живыми людьми, а лишь средство давления, используемое против них, — а персональные привязанности, эмоции, амбиции и привычки имеют большое значение. Нужно уметь работать в разных организациях, учитывая доминирующий в них вид организационной культуры при получении необходимой информации и представлении результатов проекта.

Таким образом, успешность создания ПО, связана не только с техническими знаниями участниками проектной команды, но и с их умением адаптировать свои личные качества и навыки к контексту проекта и сочетать их с интересами коллектива. Иначе «невежество, амбициозность, эгоизм, леность, слабость и разного рода страсти приводят к тому, что общие интересы ослабевают, уступая место интересам личным, и это обстоятельство порождает вечную борьбу» [3], а не успех.

Итак, для обеспечения успеха при применении формальных методов на практике, нужно обучать персонал не только их методологическим основам, но и прививать ему необходимые когнитивные и социальные навыки. Эти навыки должны стать для такого инженера или студента не менее привычными, чем умение доказывать теоремы.

Мы утверждаем, что активные методы обучения [4,5] являются гораздо более эффективным инструментом обучения передовым методам разработки ПО, и формальным методам в частности, чем традиционные подходы к обучению. Активные методы эффективно вырабатывают у студентов необходимые социальные навыки за счет акцента на автономную работу, самостоятельный выбор направления действий, самостоятельное принятие решений, необходимость общения с другими учащимися, необходимость подстраиваться под изменения в рамках образовательного процесса. В то же время классические подходы к обучению подталкивают обучаемых к пассивному восприятию информации и превращают их в хорошо информированных людей, неспособных без внешнего руководства использовать свои огромные знания на практике в постоянно меняющемся контексте.

Активное обучение построено на том, что преподаватель более не является просто источником информации, а вовлекает студентов в процесс обучения, координирует их самостоятельную деятельность, вынуждает их

самостоятельно размышлять над материалом курса и применять изучаемые методы на практике. Классический подход к обучению, основанный на предоставлении информации, напротив, нацеливает учащихся на запоминание и воспроизведение в дальнейшем некоторого объема знаний. Такой подход делает акцент на внимании и запоминании данных, в то время, как для эффективного овладения социальными и когнитивными навыками должны быть задействованы все аспекты восприятия и эмоциональной сферы обучаемого.

Активные методы обучения — это методы, не направленные на изложение готовых знаний и запоминание их студентами, а побуждающие учащихся к активной мыслительной и практической деятельности в процессе овладения учебным материалом. Активная деятельность в обоих этих направлениях должна стать основой самостоятельного овладения ими знаниями и умениями.

Многие авторы ([6]) выделяют следующие признаки активных методов обучения:

- принудительная активизация мышления, вынуждающая обучаемого быть активным независимо от его желания;
- достаточно длительное время вовлечения обучаемых в учебный процесс, поскольку их активность должна быть не кратковременной и эпизодической, а в значительной степени устойчивой и длительной;
- самостоятельная творческая выработка решений и повышенная степень мотивации.

В рамках одной статьи невозможно описать все техники, которые могут помочь при развитии необходимых разработчикам ПО когнитивных и социальных навыков. Далее мы попытались предоставить несколько примеров таких техник, которые подталкивают студентов к активному мышлению и использованию получаемых при обучении знаний в практической деятельности.

2. Активное обучение формальным методам

Для каждого вида профессиональной деятельности важны те или иные технические, когнитивные и социальные навыки. Для области формальных методов мы выделили следующие группы социальных и когнитивных навыков, имеющих большое значение:

- умение выслушивать оппонентов;
- умение эффективно общаться с другими людьми в рамках профессиональной деятельности, где под эффективностью понимается способность получения необходимой информации в кратчайшие сроки и с наименьшими затратами со стороны коллег;
- умение адаптировать свои знания к изменяющимся реальным условиям.

Далее рассматриваются несколько приемов, которые позволяют развить у студентов наряду с техническими навыками перечисленные выше когнитивные и социальные. Эти техники были использованы на практике и продемонстрировали хорошие результаты. Это ролевая игра, кластер, мозговой штурм и дебаты.

2.1. Ролевая игра

Ролевая игра является одной из техник, нацеленных на выработку навыков самостоятельного решения проблем, хотя в обучении техническим и инженерным предметам он используется не очень часто [7,8]. *Ролевая игра* — это искусственно сконструированная последовательность ситуаций, имитирующая те или иные важные стороны реальной деятельности. Во время игры ее участники играют определенные роли и ведут себя, исходя из соответствующих точек зрения на определенную ситуацию. Цель такой игры заключается в том, чтобы найти возможные решения проблем, которые возникают в рассматриваемых ситуациях.

Например, следующая игра может быть организована для студентов старших курсов с целью оценки ценности предложенной научной идеи.

Участники разыгрывают различные роли, чтобы получить оценку идеи с различных точек зрения.

- ‘Пропагандист’ объясняет идею, ее перспективы и возможные положительные последствия ее разработки и применения;
- ‘Теоретик’ рассматривает связи идеи с основами предметной области и другими теоретическими вопросами;
- ‘Технический эксперт’ оценивает полезность и важность идеи с точки зрения эффективного применения ее на практике, а также удобство ее использования;
- ‘Менеджер’ выясняет влияние и последствия использования идеи на практике на промышленные процессы, распределение работ, командную работу;
- ‘Инвестор’ оценивает потенциальные выгоды от возможных применений идеи, возможность и экономическую оправданность разработки коммерческих продуктов на ее основе, или интеграции практических применений идеи в различные существующие решения.

В Таб. 2 представлен пример результата ролевой игры — оценки идеи с различных точек зрения.

Ролевые игры позволяют повысить интерес к изучаемому материалу, выявить пробелы в знаниях и развить социальные навыки. Однако, ролевая игра ведется по жестким правилам и не может отразить многообразие ситуаций и динамизм, присущие реальной жизни. Очень трудно составить набор ролевых игр, которые позволят отработать большинство важных реальных ситуаций в данной области.

Идея: <i>Генерация тестовой последовательности в результате обхода графовой модели тестируемой системы как неизвестной местности</i>				
Пропагандист	Теоретик	Технический эксперт	Менеджер	Инвестор
Графовые модели сложных систем могут быть представлены очень компактно.	Явные графовые модели дают возможность лучше контролировать ход тестирования.	Идея актуальна как достаточно надежный метод контроля качества.	Идея обещает некоторое повышение качества ПО.	Идея может быть реализована в виде продукта, только если продукт будет нацелен на использование при разработке ПО с повышенными требованиями к надежности.
Достаточно сложные ситуации могут быть протестированы автоматически.	Может использоваться как способ построения адаптивных тестовых последовательностей.	Неявные графовые модели тяжелее описать, особенно тем, кто прежде с ними не сталкивался. Этот подход нужно сделать более удобным для использования.	Нет ясной связи между тестированием и требованиями — зачем нам тестировать все эти ситуации? Для того, чтобы научиться использовать этот метод, тестирующим необходим специальный тренинг.	Нужно предложить менее трудоемкий способ описания графов.
Качества такого тестирования очень высокое.		Необходим инструмент с графическим пользовательским интерфейсом, который будет помогать строить модели такого рода.	Идея может использоваться для создания модулей тестирования, подключаемых к средам разработки.	

Таб. 2. Пример результатов ролевой игры.

Эффективность ролевых игр можно повысить за счет использования метода "развивающего дискомфорта". Действенность этого метода основана на психологической перестройке личности под воздействием стихийно возникающих или специально организованных эмоционально негативных ситуаций. В результате такой перестройки меняется характер проявляемой

активности, и человек приобретает умение, с одной стороны, успешно выходить из критической ситуации, а с другой, позитивно отражать эти ситуации в своем социальном опыте. Иначе говоря, в соответствии с этим методом надо не только не бояться сложных, негативных ситуаций, но использовать их для поступательного развития у студентов навыков принятия решений и уверенности в собственных силах.

2.2. Кластер

Методы активного обучения предполагают не только создание новой методики обучения и изменение критериев оценки подготовки студентов, но, прежде всего, осознание преподавателем своей новой роли, в которой главной является помощь обучаемым в самостоятельном овладении предметом, а не его преподавание. При активном обучении приоритетными являются не усвоение и воспроизведение готовых знаний, а самостоятельное приобретение и особенно применение полученных знаний. Активные методы обучения вынуждают студентов вести активную работу со знанием, задумываясь над ним, анализируя его и сравнивая с уже имеющимися у них знаниями. При активном обучении не должен поддерживаться подход к обучению некоторых студентов, когда они говорят: "А вы скажите как нужно, мы так и сделаем".

Некоторые способы изменения отношения к учебному материалу основаны на личностном принятии знаний, в результате которого полученное знание становится не пересказом какого-либо учебника, а собственным знанием студента. Отмечено, что студенты, обучающиеся по такой системе, хорошо объясняют полученные знания, так как они ими приняты. Для привязки знаний к личному опыту можно использовать *кластер* [9]. Кластер создает условия для построения ассоциативных связей между новым материалом и имеющимся опытом, причем эти связи основаны не только на подобию, но и на эмоциональном восприятии, интуиции, а также на основе предположений, возможно ошибочных.

Перед началом лекции по новой теме каждый студент рисует кластер на эту тему. Последовательность действий при рисовании кластера следующая:

1. Посередине чистого листа написать ключевое слово или фразу, ключевую для данной темы.
2. Вокруг него без какого-либо плана или упорядоченности записываются слова или фразы, выражающие идеи, факты, образы, подходящие для данной темы или кажущиеся как-то связанными с ней.
3. По мере записи, появившиеся слова соединяются связями с ключевым понятием. Такое соединение может быть иерархическим. При этом должны соблюдаться следующие правила:
 - не нужно бояться записывать все, что приходит на ум. Дать волю воображению и интуиции;

- продолжать работу, пока не кончится время или не иссякнут идеи;
- стараться построить как можно больше связей и не следовать по заранее определенному плану.



Рис. 3. Пример кластера понятия 'Формальные спецификации'.

На Рис. 3. представлен пример кластера понятия 'Формальные спецификации'. Перед началом рисования кластера преподаватель должен четко сказать, сколько времени отводится на эту работу. Когда кластер у каждого нарисован, нужно попросить одного студента (только по желанию!) нарисовать свой кластер на доске. Если желающих рисовать нет, то преподаватель рисует свой собственный кластер. Обсуждать ничего не нужно. Каждый самостоятельно сравнивает свой кластер с тем, который нарисован на доске и при желании может внести изменения в свой кластер. После окончания лекции каждый студент может дорисовать кластер: добавить новое знание, или что-нибудь убрать.

Очень полезно при возможности организовать обсуждение кластера в малых группах или в парах. Результатом такого обсуждения должны стать невыясненные вопросы, ответы на которые должны быть получены на последующих занятиях. Кроме того, такое обсуждение позволяет повысить коммуникабельность студентов.

2.3. Мозговой штурм

Для студентов старших курсов, начинающих исследовательскую деятельность, очень важно иметь некоторый набор идей в качестве отправной точки исследований. Для получения стартового набора идей может использоваться *мозговой штурм* [10]. Мозговой штурм организуется в несколько этапов.

1. Участники штурма должны быть ознакомлены с темой исследований студента.
Эта часть может быть организована следующим образом.
 - (a) Студент и его руководитель подготавливают конспект по теме, проблемам исследуемой области и используемым в ней методам.
 - (b) Каждый участник штурма читает подготовленный конспект.
 - (c) Участники формулируют вопросы, касающиеся проблем рассматриваемой области. Эти вопросы записываются на доске.
 - (d) Студент делает доклад на заданную тему, в котором должны быть сформулированы его интересы в исследуемой области и даны ответы на вопросы участников штурма.
 - (e) После доклада вопросы, на которые были получены ответы (то, что ответ на вопрос получен, должен подтвердить участник, задавший этот вопрос), отмечаются знаком "+". Никаких новых вопросов задавать нельзя.
2. Участники предлагают идеи возможных направлений дальнейшего исследования. При этом должны соблюдаться следующие правила.
 - (a) Каждый участник имеет право на выступление в течение 30 секунд. Во время такого выступления должно быть сформулировано не больше одной идеи.
 - (b) Количество выступлений одного участника неограниченно.
 - (c) Выступление нельзя прерывать.
 - (d) Каждый участник имеет право ничего не говорить.
 - (e) Идея выступления должна быть сформулирована по возможности четко, желательно в одном предложении.
 - (f) Во время выступления нельзя обсуждать, критиковать и оценивать предыдущие выступления.
 - (g) Запрещено использование в выступлениях выражений "очевидно, что...", "вы не понимаете...", "не была дана научная концепция...", "я хочу уточнить...", "я хочу объяснить..." и т.п.
 - (h) Порядок и корректность выступлений обеспечивает ведущий штурма.
 - (i) Все предложенные идеи записываются ведущим.
3. Участники оценивают все предложенные идеи на субъективной основе, но в то же время некоторым систематическим образом, например, на основе таблицы ПМИ (Плюс-Минус-Интересно). Этот процесс может быть организован следующим образом.

- (а) Для каждой предложенной идеи и каждого участника подготавливаются листы бумаги, содержащие название идеи и три колонки для записей. В первую колонку вписываются аргументы в поддержку идеи, во вторую — против нее, в третьей колонке записываются интересные аспекты, касающиеся данной идеи. В другой схеме наряду с плюсами и минусами формулируются возможные последствия применения идеи на практике.
- (б) Каждый участник получает листы со всеми идеями и заполняет их. Необязательно оценивать все идеи. Пример заполненного листа показан в Таблице 4.
- (с) Все листы собираются для составления интегральных оценок.
4. Студент и его научный руководитель подводят итоги рассмотрения всех идей и определяют наиболее перспективные идеи.

Идея: <i>Использование методов морфологического анализа для нахождения дефектов в ПО</i>		
Плюсы	Минусы	Интересные моменты
Совсем новая	Необходим исходный код ПО	Может быть использована в комбинации с обычными методами статического анализа
Может быть использована для нахождения очень тонких дефектов	Нужно прояснить понятие дефекта Если в качестве дефекта будет рассматриваться корректный код, нужно предпринимать сложные действия по исправлению такой ситуации	

Таб. 4. Пример субъективной оценки идеи, полученной в результате мозгового штурма.

Организация мозгового штурма должна помогать участникам генерировать и высказывать интересные идеи, какими бы фантастическими или безумными они не казались на первый взгляд. Также участники должны быть освобождены от давления авторитетных мнений, они не должны быть вынуждены защищать свои идеи — любые идеи должны приниматься к рассмотрению.

<i>Стандарт POSIX предоставляет строгое описание функциональности операционной системы</i>	
За	Против
<p>Стандарт по определению является строгим описанием. Он создается на основе формальных правил, регламентирующих описания отдельных функций.</p> <p><i>Поддержка:</i> Если почитать POSIX, легко заметить, что его требования оформлены в соответствии с определенным набором правил. POSIX представляет выработанный большой группой экспертов консенсус по поводу функциональности операционных систем. Он существует более 20 лет, за это время все существенные дефекты были найдены и устранены.</p> <p>Множество операционных систем успешно используют POSIX как рекомендации по предоставляемым функциям.</p>	<p>Формальные правила оформления документа не гарантируют строгости представленного в нем содержательного текста.</p> <p>POSIX является компромиссом между многими крупными поставщиками операционных систем (ОС), взгляды которых на их разработку могут значительно отличаться.</p> <p><i>Поддержка:</i> Двусмысленности неизбежны в стандарте, объединяющем несколько разных подходов к проектированию ОС.</p> <p>Возьмем, например, функцию создания потока (thread). POSIX говорит, что эта функция <i>может</i> взаимодействовать с подсистемой управления динамической памятью, но ничего не говорит о том, <i>в чем именно может заключаться</i> это взаимодействие.</p>

Таб. 5. Пример результата дебатов.

2.4. Дебаты

Одним из важных навыков для разработки ПО является умение выслушать мнение, отличное от собственного, понять, на чем оно основывается, и принять это во внимание в дальнейшей работе. *Дебаты* [11] позволяют отработать данный навык.

В начале на доске записывается спорное утверждение, которое будет обсуждаться. Каждый участник по отдельности или каждая группа участников выдвигает аргументы, поддерживающие или опровергающие рассматриваемое утверждение. Каждый аргумент может быть в свою очередь поддержан другими утверждениями. Все аргументы и поддерживающие их утверждения записываются на доске. Целью дебатов является предоставление каждому участнику материала для формирования собственной точки зрения на обсуждаемую тему. Другой целью является тренировка навыков в отстаивании своей точки зрения и понимании других, отличных от нее. Студенты знают, что им могут возразить, они психологически готовы к этому, они тренируются концентрироваться на существенных деталях, анализировать и оценивать различные точки зрения на предмет обсуждения и относиться с уважением к противоположным и, может быть, непопулярным точкам зрения.

Примеры записей на доске после дискуссии по поводу некоторого утверждения представлены в Таблице 5.

3. Заключение

В статье предложена идея интегрированного обучения техническим, когнитивным и социальным навыкам, необходимым при использовании передовых методов разработки ПО, в частности формальных методов, на практике. Такое интегрированное обучение может быть организовано на основе методов активного обучения. В статье представлены несколько таких методов, помогающих вырабатывать необходимые навыки социального плана — умение слушать и принимать во внимание противоположные точки зрения, умение взаимодействовать эффективно при обсуждении профессиональных проблем, умение применять знания в реальных нестандартных ситуациях.

В статье описаны четыре техники обучения — ролевая игра, кластер, мозговой штурм и дебаты. Авторы успешно применяли эти техники в обучении современным методам тестирования ПО, основанного на моделях [12,13], и уверены, что они могут использоваться для обучения любым формальным методам и особенно полезны при обучении их использованию на практике.

Было проведено несколько тренингов с использованием описанных техник и некоторых других. Тренинги продемонстрировали, что при таком обучении студенты лучше усваивают знания и становятся более активным и самостоятельными в практической работе.

Литература

- [1]. Ф. Брукс. Мифический человек-месяц или как создаются программные системы. Спб., Символ-Плюс, 2001.
- [2]. A Memorandum on Lifelong Learning: Commission Staff Working Paper. European Commission. Directorate General for Education and Culture. European Commission, 2000. SEC (2000) 1832.

- [3]. А. Файоль, Г. Эмерсон, Ф. Тейлор, Г. Форд. Управление — это наука и искусство. Республика, М. 1992.
- [4]. S. C. Brock. Practitioners' views on teaching the large introductory college course. 12 pp. Kansas State University Center for Faculty Evaluation and Development. ERIC Document Reproduction Service No. ED 171 208, 1976.
- [5]. Labinowicz, Ed. The Piaget Primer: Thinking, Learning, Teaching. Menlo Park, CA: Addison-Wesley Publishing Co., 1980.
- [6]. Педагогические технологии. Под ред. В. С. Кукушкина. М., 2004.
- [7]. Dennis M. Adams. Simulation Games: An Approach to Learning. Worthington, Ohio, Charles A. Jones Publishing Company, 1973.
- [8]. John P. Hertel, and Barbara J. Mills. Using Simulations to Promote Learning in Higher Education: An Introduction. Stylus Publishing, Herndon, VA, 2002.
- [9]. Tony Buzan. Use Both Sides of Your Brain. New York: Dutton, 1974.
- [10]. James L. Marra. Advertising Creativity: Techniques For Generating Ideas. Englewood Cliffs, NJ: Prentice Hall, 1990.
- [11]. Ronald T. Hyman. Improving Discussion Leadership. New York: Columbia Univ., Teachers College Press, 1980.
- [12]. O. L. Petrenko, V. A. Omelchenko. Rapid Trainings on Specification Based Testing Tools. In Proc. of 1-st South-East European Workshop on Formal Methods, Thessaloniki, Greece, 2003.
- [13]. В. В. Кулямин, В. А. Омельченко, О. Л. Петренко. Обучение передовым технологиям разработки ПО: проблемы и методы их решения. Труды ИСП РАН, т. 5, 2004, стр. 101–120.