

Разработка и применение автоматизированной обучающей системы для совершенствования процесса профессиональной подготовки специалистов по тестированию программного обеспечения

И.С. Полевицков, М.В. Калин

Пермский национальный исследовательский политехнический университет

Аннотация: Рассмотрена проблема повышения эффективности подготовки специалистов по тестированию программного обеспечения (ПО) на основе применения автоматизированной обучающей системы (АОС) для управления процессом формирования профессиональных компетенций. Для уменьшения трудоемкости и повышения эффективности процесса тестирования стоит задача обучить начинающего специалиста современным техникам тестирования, в том числе с учетом опыта реальных программных проектов. Применение средств автоматизации может интенсифицировать и повысить качество выполнения задачи обучения. В работе предлагаются модели и алгоритмы для управления формированием профессиональных знаний и навыков специалистов по тестированию с использованием современных математических методов. Работа выполнена при поддержке стипендии Президента РФ молодым ученым и аспирантам (№ стипендии СП-100.2018.5), назначенной Советом по грантам Президента Российской Федерации.

Ключевые слова: автоматизированные обучающие системы, тестирование программного обеспечения, подготовка кадров, профессиональные знания и навыки.

1 Введение. Актуальность исследования

При создании программного обеспечения (ПО) требуется правильная организация всех этапов процесса разработки, основанная на применении технологических принципов, и, в частности, грамотная организация этапа тестирования как неотъемлемого, наиболее устоявшегося средства современной системы обеспечения качества программного продукта [1-3].

Для уменьшения трудоемкости и повышения эффективности процесса тестирования используются разные подходы. С одной стороны, разрабатываются и применяются различные средства автоматизации тестирования [4]. С другой стороны, стоит задача обучить начинающего специалиста современным техникам тестирования, в том числе с учетом опыта реальных программных проектов. Применение средств автоматизации может повысить качество выполнения задачи обучения [5-8].

Обучение методам тестирования ПО является неотъемлемой составляющей подготовки бакалавров по направлению 09.03.04 «Программная инженерия» на кафедре ИТАС ПНИПУ. Однако, практическая составляющая процесса обучения в настоящее время обладает существенными недостатками как для студентов, так и для преподавателя.

В первую очередь, отсутствуют средства автоматического контроля и самоконтроля выполнения практических заданий для отработки навыков тестирования. Ручная проверка таких заданий преподавателем является весьма однообразной и утомительной работой, отнимающей много времени.

Также отсутствует достаточное количество примеров, показывающих в наглядном виде различные возможные ситуации и частные случаи в процессе тестирования. Особенно этот недостаток действителен для студентов заочного отделения, у которых крайне мало часов аудиторных занятий с преподавателем. Преподавателю требуется большое количество времени, чтобы подготовить много таких примеров и объяснить студентам.

Существующие средства автоматизации профессионального обучения не позволяют в полной мере преодолеть указанные выше недостатки.

Таким образом, разработка обучающей системы для автоматизации контроля выполнения практических заданий при обучении методам тестирования ПО на основе научно-обоснованных моделей и методов является востребованной задачей.

2 Функциональные возможности автоматизированной обучающей системы для подготовки специалистов по тестированию ПО

Функциональные требования к разрабатываемой автоматизированной обучающей системе (АОС) для подготовки специалистов по тестированию ПО описаны в виде диаграммы вариантов использования UML [1] (рис. 1).

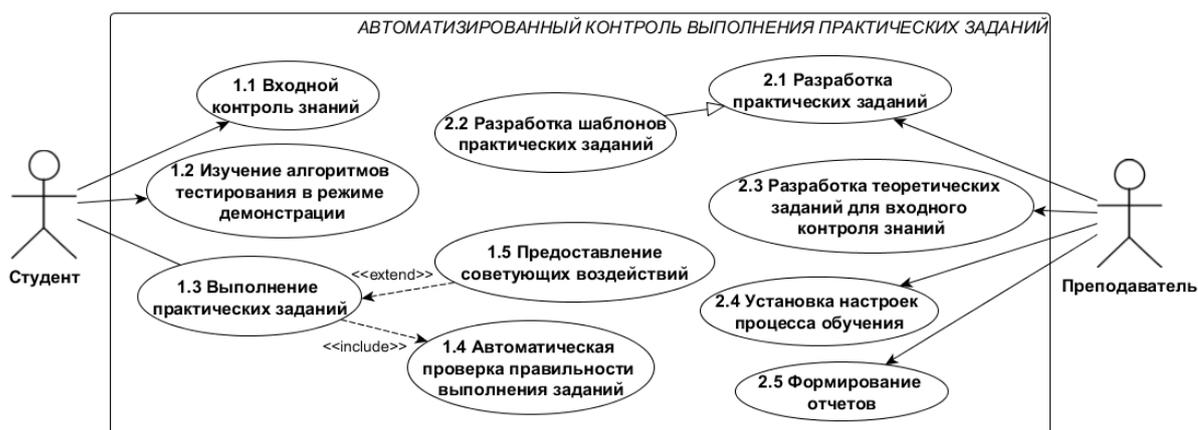


Рис. 1. – Функциональные требования к АОС

Разработан прототип данной системы. АОС обеспечивает изучение студентами алгоритмов тестирования в режиме демонстрации. В наглядной форме показываются конкретные пошаговые примеры решения задач. Студент может корректировать различные параметры в задачах, чтобы всесторонне рассмотреть частные случаи тестирования.

Система позволяет автоматически (в соответствии с настройками преподавателя) генерировать и оценивать практические задания для обучения различным техникам тестирования ПО и для проверки сформированных навыков. Перед выполнением практических заданий предусмотрен входной контроль знаний (как для оценки знаний теории по тестированию ПО, так и по смежным дисциплинам, например, дискретной математике и математической логике).

При генерации заданий будут использованы как различные алгоритмы, лежащие в основе известных методов тестирования [1, 9] (например, тестирование базового пути; тестирование, основанное на разбиении по эквивалентности и анализе граничных значений; тестирование на основе диаграмм причинно-следственных связей), так и данные о тестировании реальных проектов, хранящиеся в базе данных автоматизированной системы.

3 Алгоритм выполнения практических заданий с применением АОС

Любое практическое задание в АОС можно разделить на последовательность частей, определяемых изучаемым методом тестирования ПО. Например, для тестирования базового пути [1] такими частями могут являться: построение потокового графа на основе текста программы; вычисление цикломатической сложности; построение множества путей и т.д.

Алгоритм выполнения студентом практического задания в АОС можно обобщенно представить последовательностью шагов:

1) Предоставление студенту задания, сформированного на основе начальных настроек преподавателя.

2) Формирование студентом решения i -й части задания на основе предоставляемых системой инструментов.

3) Оценка системой выполнения i -й части задания и получение коэффициента $K_i \in [0;1]$, определяющего степень правильности выполнения (вычисляется по некоторому алгоритму, определяемому спецификой данной части задания). Предоставление советующих воздействий (подсказок) обучаемому, если это было предусмотрено преподавателем. Возможен возврат к шагу № 2 при низком значении K_i либо других условиях, заданных преподавателем. Здесь происходит увеличение i на 1.

4) Если все части задания выполнены, то переход к шагу 5. Иначе переход к шагу 2.

5) Получение комплексной оценки, определяющей степень правильности выполнения всего задания: $K = \sum_{i=1}^n w_i K_i$, где w_i – «вес» (сложность) выполнения i -й части задания (установлен преподавателем в настройках); n – число частей задания; K_i – оценка за i -е задание.

6) Далее, используя шкалу (установленную ранее преподавателем), обучаемому выставляется оценка по 5-балльной системе за выполнение

задания, например: $0,9 \leq K \leq 1$ – отлично; $0,8 \leq K < 0,9$ – хорошо; $0,7 \leq K < 0,8$ – удовл.; $K < 0,7$ – неуд.

7) На экране отображается оценка выполнения задания, а также, если это было предусмотрено преподавателем, предоставляются советующие воздействия обучаемому (различные рекомендации, подсказки, разбор допущенных ошибок и т.п.).

4 Алгоритм сравнения потоковых графов при оценке заданий по теме «Тестирование базового пути»

Рассмотрим на примере алгоритм сравнения потоковых графов при выполнении студентом задания по теме «Тестирование базового пути» (рассмотрена часть задания, в которой на основе пронумерованного преподавателем текста программы требуется построить потоковый граф):

1) Студенту предоставляется пронумерованный текст программного модуля. Пример подобного модуля на языке Java представлен на рис. 2:

```
int w=0; // 1
for(int i=5 /*1*/; i<10 /*2*/; i++ /*4*/)
    w+=5; //3
System.out.println("w="+w); //5
```

Рис. 2. – Пронумерованный текст программного кода

2) Студент строит потоковый граф. Пример построенного студентом графа представлен на рис. 3.

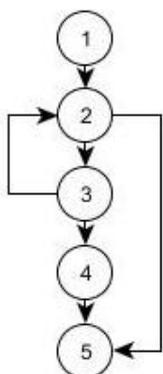


Рис. 3. – Граф, построенный студентом

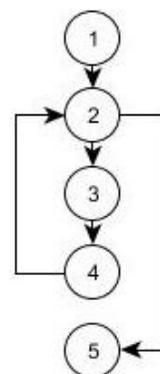


Рис. 4. – «Эталонный» граф

3) Системой автоматически сравнивается множество $M_{\text{дуг}}^{\text{обуч.}}$ дуг графа, построенного обучаемым, с множеством $M_{\text{дуг}}^{\text{этал.}}$ дуг «эталонного» графа (рис. 4), который требовалось построить. Результат сравнения можно представить в виде $K_i = f(M_{\text{дуг}}^{\text{обуч.}}, M_{\text{дуг}}^{\text{этал.}})$, где $K_i \in [0;1]$ – оценка правильности построенного графа. Чем ближе значение K_i к 1, тем больше совпадают множества, и, наоборот, чем ближе значение K_i к 0, тем больше множества различаются.

Если студент построил граф, который имеет несколько начальных или несколько конечных вершин, то $K_i = 0$, так как потоковый граф явно построен неверно (должна быть одна начальная и конечная вершина).

Значение K_i можно представить, как степень сходства [5] между ответом обучаемого (множество $M_{\text{дуг}}^{\text{обуч.}}$) и эталонным ответом (множество $M_{\text{дуг}}^{\text{этал.}}$), рассчитывающуюся по формуле:

$$K_i = \frac{Q_a}{L + Q_b},$$

где $L = |M_{\text{дуг}}^{\text{этал.}}|$ – мощность множества $M_{\text{дуг}}^{\text{этал.}}$,

$Q_a = |M_{\text{дуг}}^{\text{обуч.}} \cap M_{\text{дуг}}^{\text{этал.}}|$ – число совпадающих элементов во множествах $M_{\text{дуг}}^{\text{обуч.}}$ и $M_{\text{дуг}}^{\text{этал.}}$,

$Q_b = |M_{\text{дуг}}^{\text{обуч.}}| - Q_a$ – число элементов во множестве $M_{\text{дуг}}^{\text{обуч.}}$, которых нет в $M_{\text{дуг}}^{\text{этал.}}$.

Произведем расчет для рассматриваемого примера:

$$M_{\text{дуг}}^{\text{этал.}} = \{(1, 2) (2, 3) (2, 5) (3, 4) (4, 2)\}$$

$$M_{\text{дуг}}^{\text{обуч.}} = \{(0, 1) (1, 2) (1, 4) (2, 1) (2, 3) (3, 4)\}$$

$$K_i = \frac{Q_a}{L + Q_b} = \frac{4}{5 + (6 - 4)} = \frac{4}{7} \approx 0,57$$

4) Если было предусмотрено в настройках, отображаются советующие воздействия обучаемому (например, отображается «эталонный граф», перечисляются или подсвечиваются недостающие или лишние дуги графа).

5 Заключение

Благодаря рассмотренным выше преимуществам, разрабатываемая АОС позволяет осуществлять формирование и контроль навыков при изучении наиболее известных методов тестирования программного обеспечения (ПО).

АОС может применяться для профессиональной подготовки специалистов по тестированию ПО: при обучении студентов вузов и учреждений среднего профессионального образования, при внутрифирменном обучении сотрудников предприятий.

АОС может быть интегрирована в систему дистанционного обучения или в другую систему, используемую для автоматизации образовательного процесса, а также в систему автоматизированного управления процессом тестирования ПО на предприятиях [10] для повышения квалификации сотрудников. Более того, алгоритмы, используемые для генерации заданий, могут быть использованы и для генерации тест-кейсов с целью формирования рекомендаций специалисту по тестированию ПО.

Литература

1. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. Стандарт третьего поколения. СПб.: Питер, 2012. 608 с.

2. Бурякова Н.А., Чернов А.В. Классификация частично формализованных и формальных моделей и методов верификации программного обеспечения // Инженерный вестник Дона. 2010. №4. URL: ivdon.ru/ru/magazine/archive/n4y2010/259.

3. Яловой И.О. Анализ требований и управление изменениями программных проектов // Инженерный вестник Дона. 2008. №4. URL: ivdon.ru/ru/magazine/archive/n4y2008/102.

4. Куликов С.С. Тестирование программного обеспечения. Базовый курс.

Минск: Четыре четверти, 2017. 312 с.

5. Карпова И.П. Исследование и разработка подсистемы контроля знаний в распределенных автоматизированных обучающих системах: дис. ... канд. техн. наук: 05.13.13. М., 2002. 200 с.

6. Fayzrakhmanov R.A., Polevshchikov I.S., Khabibulin A.F., Shklier F.I., Fayzrakhmanov R.R. ANYCRANE: Towards a better Port Crane Simulator for Training Operators // Proc. of the 15th International Industrial Simulation Conference (ISC'2017), Warsaw (Poland), 31 May – 1 June 2017. pp. 85-87.

7. Fayzrakhmanov R.A., Polevshchikov I.S. Increased of Efficiency in the Automated Training of Fuelling Machine Operators Using Iterative Simulation Learning // World Applied Sciences Journal. 2013. Vol. 22. № SPL.ISSUE2. pp. 70-75. URL: [idosi.org/wasj/wasj22\(tt\)13/12.pdf](http://idosi.org/wasj/wasj22(tt)13/12.pdf).

8. Зорин Ю.А., Посов И.А. Инструментальные системы построения и получения многовариантных тестовых заданий // Компьютерные инструменты в образовании. 2014. №1. С. 14-25.

9. Калинин М.В., Полевщиков И.С. Автоматизация процесса тестирования программного обеспечения на основе построения диаграмм причин-следствий // Инновационные технологии: теория, инструменты, практика. 2017. Т. 1. С. 45-51.

10. Полевщиков И.С. Автоматизированная система управления процессом тестирования программного обеспечения в ходе промышленной разработки // Решение. 2018. Т. 1. С. 191-193.

References

1. Orlov S.A., Tsil'ker B.Ya. Tekhnologii razrabotki programmno obespecheniya: Uchebnik dlya vuzov. 4-e izd. Standart tret'ego pokoleniya [Software development technologies: A textbook for universities. 4th ed. The standard of the third generation]. SPb.: Piter, 2012. 608 p.



2. Buryakova N.A., Chernov A.V. Inzhenernyj vestnik Dona (Rus), 2010, №4. URL: ivdon.ru/ru/magazine/archive/n4y2010/259.
3. Yalovoy I.O. Inzhenernyj vestnik Dona (Rus), 2008, №4. URL: ivdon.ru/ru/magazine/archive/n4y2008/102.
4. Kulikov S.C. Testirovanie programmnoho obespecheniya. Bazovyy kurs [Software Testing. Basic course]. Minsk: Chetyre chetverti, 2017. 312 p.
5. Karpova I.P. Issledovanie i razrabotka podsistemy kontrolya znaniy v raspredelennykh avtomatizirovannykh obuchayushchikh sistemakh [Research and development of the knowledge control subsystem in distributed automated learning systems]: dis. ... kand. tekhn. nauk: 05.13.13. M., 2002. 200 p.
6. Fayzrakhmanov R.A., Polevshchikov I.S., Khabibulin A.F., Shklyayev F.I., Fayzrakhmanov R.R. ANYCRANE: Towards a better Port Crane Simulator for Training Operators. Proc. of the 15th International Industrial Simulation Conference (ISC'2017), Warsaw (Poland), 31 May – 1 June 2017. pp. 85-87.
7. Fayzrakhmanov R.A., Polevshchikov I.S. World Applied Sciences Journal. 2013. Vol. 22. № SPL.ISSUE2. pp. 70-75. URL: [idosi.org/wasj/wasj22 \(tt\) 13/12.pdf](http://idosi.org/wasj/wasj22(tt)13/12.pdf).
8. Zorin Yu.A., Posov I.A. Komp'yuternye instrumenty v obrazovanii. 2014. №1. pp. 14-25.
9. Kalin M.V., Polevshchikov I.S. Innovatsionnye tekhnologii: teoriya, instrumenty, praktika. 2017. Vol. 1. pp. 45-51.
10. Polevshchikov I.S. Reshenie. 2018. Vol. 1. pp. 191-193.