

# Выбор системы автоматизированного тестирования решений задач по программированию

А.В. Якушин, И.Ю. Гладких

**Аннотация**— Настоящая статья посвящена анализу систем автоматизированного тестирования (САТ) решений задач по программированию, определению их характеристик и рассмотрению различных представителей этих систем. Анализ был произведен путем изучения документаций САТ, их экспериментальному использованию и сравнению результатов их работы. В тексте статьи предложены различные свойства САТ и описаны наиболее функциональные и распространённые системы, что должно помочь при выборе и дальнейшем использовании САТ.

**Ключевые слова**— системы автоматизированного тестирования, электронное обучение, обучение программированию, проверка задач по программированию.

## I. ВВЕДЕНИЕ

Для повышения эффективности в своей деятельности человечество все чаще использует автоматизированные системы, берущие на себя часть труда человека. Педагогика не стала исключением из этого правила. Информационные технологии позволяют уменьшить объем работы преподавателей по проверки различных тестовых материалов [1]. Представителями таких технологий являются системы автоматизированного тестирования решений задач по программированию (САТ решений задач по программированию или в дальнейшем просто САТ).

САТ представляет собой программный комплекс, позволяющий проверять программную реализацию некоторого решения задачи по программированию путем тестирования этой реализации на множестве специальных тестов и сопоставляет результаты тестов с эталонными результатами [2], кроме того, отслеживает процесс работы приложений и контролируется потребление ресурсов системы.

На данный момент существует несколько САТ, и хотя все они призваны выполнять схожие функции, обладают

существенными различиями как в реализации программного комплекса и дополнительных функциях, так и используемых методах тестирования и подготовки к тестированию. Определим ключевые качества САТ, по которым необходимо оценивать эффективность применения того или иного САТ к использованию.

Критерии оценки САТ [2-6]:

- Клиентское приложение
- Серверное приложение
- Документация
- Интеграция
- Контроль над тестируемыми приложениями
- Надежность
- Распределение вычислений
- Масштабируемость вычислений
- Перечень языков программирования
- Объем заготовок
- Отчетность

## II. КРИТЕРИИ ОЦЕНКИ САТ

### A. Клиент-серверная архитектура САТ

САТ представляет из себя программный комплекс [2]. При этом в большинстве случаев он ориентирован на ситуацию, когда необходимо производить сбор данных – программные реализации решений задач – с различных узлов сети, компьютеров авторов программ, в то же время результаты тестирования всех работ должны быть доступны преподавателю. Таким образом, САТ должна придерживаться клиент-серверной архитектуры приложений [3]. Очевидно, на оценку эффективности и удобства САТ влияют характеристики клиентского, серверного приложений комплекса, а также канала связи между ними [3].

### B. Клиентское приложение

Клиентское приложение может накладывать часть ограничений на клиентов САТ [3]. Возможно несколько случаев качества клиентского приложения.

В первом случае клиентское приложение отсутствует как таковое, таким образом, вся деятельность пользователей должна происходить на одном и том же компьютере. В случае же существования клиентского программного обеспечения (ПО) на оценку САТ влияют ограничения, накладываемые на это ПО, такие как критерии к аппаратной или программной частям системы или квалификацию пользователя.

Статья получена 12 декабря 2015.

Статья подготовлена в рамках выполнения государственных работ в сфере научной деятельности по базовой части государственного задания № 2014/389, НИР №2193.А.В. Якушин ФГБОУ ВПО «Тулский государственный педагогический университет им.Л.Н.Толстого» (yakushin@tspu.tula.ru).

И.Ю. Гладких ФГБОУ ВПО «Тулский государственный педагогический университет им.Л.Н.Толстого» (gladkihiu@gmail.com).

### *C. Серверное приложение*

Аналогично клиентскому ПО серверное ПО накладывает собственные ограничения на программную и аппаратную составляющую сервера САТ [3], отдельные требования выставляют требования к квалификации администратора сервера. Не исключен случай отсутствия отдельного серверного ПО, тогда существуют две различных категории. В первой категории находится программный комплекс, в котором серверное ПО не существует как отдельный пакет, но определенные настройки в программе САТ выделяют сервер от клиентов. Во второй категории, все узлы САТ равноправны, в таком случае вся отчетность как по проведенным тестам, так и по работе системы доступна всем клиентам в равной степени.

### *D. Документация*

Как и любое ПО на САТ накладываются определенные требования к документации проекта [4] как для конечных пользователей системы, так и для администраторов, обеспечивающих работоспособность системы. Однако объем документации и освещенные аспекты функционирования САТ могут существенно различаться. Документация может ограничиваться лишь техническими требованиями ПО, так и описанием использования системы. Уровень детальности и охвата документации влияют на конечную оценку САТ [4]. Так недостаточность документации может привести к непредвиденным ошибкам в функционировании САТ или затруднить изучение системы.

### *E. Интеграция*

Важный аспект оценки САТ – это ситуация, в которой используется САТ. Это может быть проведение учебных занятий, соревнований и конкурсов по программированию или использование в качестве контрольно-измерительных материалов [5]. В зависимости от этого назначения САТ может быть представлена как отдельный электронный ресурс, специализированная служба, интегрированная в какую-либо оболочку системы электронного обучения или как стандартная функция этой системы. Уровень интеграции влияет на конечную оценку САТ: как много действий необходимо пользователям для использования САТ, какова нагрузка на аппаратную и программную составляющую информационных технологий, задействованных при использовании САТ.

### *F. Контроль над тестируемыми приложениями*

По своей работе САТ необходимо запускать приложения функции, которых в общем случае неизвестны, как не определена и угроза этих приложений системе, в которой производится тестирование. В связи с чем, важной функцией САТ выступает функция контроля за тестируемыми приложениями. САТ должна контролировать потребляемые тестируемыми приложениями ресурсы системы [6]. Кроме того, необходим контроль за потенциально опасными и вредоносными действиями приложений, кроме обеспечения безопасности системы

от вредоносных тестируемых приложений САТ может производить некоторый анализ программного кода или тестируемых решений [6]. Таким образом, контроль и анализ тестируемого ПО влияет на конечную оценку САТ.

### *G. Надежность*

Опасность может нести не только тестируемые программы, но и сам программный комплекс САТ. В первую очередь критической характеристикой выступают технологии, задействованные в функционировании САТ [4]. Кроме того, важны влияния САТ на функционирования операционной системы и аппаратной части во время активного тестирования программ и в режиме ожидания [6]. Свое влияние оказывает и сетевая архитектура приложения, канал передачи данных и безопасность доступа к САТ [3, 6]. Таким образом, надежность САТ влияет на конечную оценку системы.

### *H. Распределение вычислений*

Тестирование программных реализаций и подготовка к их тестированию (компиляция программного кода, подготовка множества тестов и т.д.) может быть распределено между клиентской и серверной частью САТ разными способами [3].

Компиляция программного кода может происходить как на клиентской машине как элемент подготовки к тестированию, так и сервере как часть проверки программного решения. Само тестирование также может происходить как на сервере, так и на клиентских машинах, что зависит от функциональных возможностей конкретной САТ и поставленных требованиях к производительности и безопасности тестирования. Кроме того, существует возможность использования нескольких серверных элементов САТ на различных машинах, таким образом дистанцируя функцию управления системой от непосредственного тестирования приложений. Некоторые САТ поддерживают возможность использования отдельных публичных или корпоративных серверов для обеспечения тестирования на них, в таком случае дистанцированное тестирование выступает как отдельный сервис. Возможность распределения вычислений САТ влияет на ее конечную оценку.

### *I. Масштабируемость вычислений*

Важным аспектом САТ является ее производительность и максимальный объем используемых для вычислений ресурсов [5]. В архитектуре САП может быть заложено ограничение на характеристики данных для тестируемых приложений, или объем доступных тестов приложений за единицу времени. Эти ограничения могут быть вызваны некоторым аспектом монетизации САТ, так некоторые разработчики накладывают ограничения на объемы используемых в тестировании данных, размеры приложений и количество проводимых тестов с целью стимуляции пользователей САТ к покупке других версий программного продукта или расширения этих

пределов

#### *J. Перечень языков программирования*

Многие САТ в качестве программного решения для тестирования используют программный код, который будет скомпилирован уже внутри системы, дабы избежать ошибок, связанных с различием в архитектуре среды тестирования и среды компилирования. Таким образом, одной из характеристик САП становится перечень языков программирования, которые поддерживаются системой. При этом, так как языки программирования развиваются, появляются новые версии и языки, важен не только объем этого множества, но и возможность его дополнения [1, 2]. Поддержка языков программирования может совершенствоваться двумя путями: обновлениями от разработчиков САТ и ручной настройкой администраторов. В зависимости от объема и актуальности перечня поддерживаемых языков программирования, возможности модификации этой опции администратором САТ, а также качеством и своевременностью обновлений может измениться конечная оценка САТ.

#### *K. Объем заготовок*

В зависимости от поставленной задачи САТ, важна функция повторного использования тестов, переноса их между различными версиями САТ [5], а также объем операций необходимых для использования САТ с момента установки. Разработчики САТ предоставляют различный объем и качество заранее подготовленных материалов для использования в САТ. Эти материалы могут представлять собой заранее подготовленные задания, конкурсы или же шаблоны задач. Объем и качество подобных заготовок влияет на удобство как в начале использования, так и при постоянном использовании САТ. Что в свою очередь влияет на определение оценки САТ

#### *L. Отчетность*

При выборе САТ необходимо учитывать, как именно необходимо контролировать деятельность пользователей САТ [3-5]. На что влияет цель использования САТ. Мониторинг тестирования может демонстрировать результаты лишь после полного проведения тестирования ПО или же демонстрировать процесс тестирования, отображая информацию по отдельным тестам и самому процессу работы ПО и САТ. В то же время такой мониторинг требует дополнительных ресурсов, а контроль лишь результатов гораздо экономичнее. Объем информации по результатам тестирования также может различаться, в зависимости от САТ кроме результатов самого тестирования ПО может быть предоставлена информация по процессу тестирования, сравнение с другими решениями или же данные связанные с автором ПО. Различаются и формы подачи информации отчетности, САТ может обладать функцией настройки выводимой информации, формы вывода и способа передачи пользователю. Отчетность и возможность ее

настройки влияет на конечную оценку САТ

### III. САТ. РАЗЛИЧНЫЕ РЕШЕНИЯ

Описанные выше свойства в полной мере применимы к различным САТ, как к свободно распространяемым, так и к проприетарным вариантам. На данный момент наиболее распространены следующие системы автоматизированных тестирования и проверки решений задач по программированию

- PC2
- DOMjudge
- Ideone.com и Onlinejudge
- Timus Online Judge
- Ejudge
- Contester

### IV. ОПИСАНИЕ РАЗЛИЧНЫХ САТ

#### *A. PC<sup>2</sup>*

PC<sup>2</sup>(Programming Contest Control) – САТ разработанный Университетом штата Калифорния, Сакраменто [7]. Программный комплекс включает клиентское и серверное приложения. Обе части написаны с использованием технологии Java, чем достигнута кроссплатформенность САТ, но существенная уязвимость самой системы и дополнительные требования к предустановленному ПО. Архитектура САТ предполагает использование локального корпоративного сервера, внешний общедоступный сервер для нескольких организаций не предусмотрен. Документация проекта покрывает весь доступный функционал системы (английский язык). Для использования САТ предназначены две различные оболочки: графическая и консоль управления. Не предусмотрено мастера установки и настройки системы, все необходимые манипуляции с конфигурацией необходимо выполнить через соответствующий интерфейс или непосредственно в файлах конфигурации. PC<sup>2</sup> поддерживает различные компиляторы и интерпретаторы. Расширение перечня компиляторов происходит вручную администратором или после установки новой версии системы. Перечень поддерживаемых интерпретаторов администратором не управляется, в него входят PERL, PHP, Ruby, Python, и shell.

#### *B. DOMjudge*

DOMjudge – САТ Исследовательской Ассоциации A-Eskwadraat Утрехтского университета, Нидерланды [8]. Архитектура САТ предусматривает серверное ПО и клиентский доступ через web-интерфейс. DOMjudge позволяет разделять процесс обработки данных между несколькими серверами, в том числе возможно перенаправление вычислений с локального сервера на внешний (вне локальной сети). САТ накладывает ограничения на серверные операционные системы: предполагается использование только на UNIX-системах, рекомендуется Debian. Установка и настройка системы происходит через специальные файлы

сценариев, возможна конфигурация системы путем ручного редактирования файлов. Развитая система документации и поддержки проекта, включает в себя различные заготовки и шаблоны задач, наборов задач и соревнований. Относительно небольшой перечень поддерживаемых языков программирования (C, C++, Java (GNU), Java (Oracle), Haskell и Pascal) может быть расширен администратором путем ручного редактирования файлов конфигурации или с помощью скриптового файла мастера.

### C. Ideone.com и Onlinejudge

Ideone.com и Onlinejudge – пара систем взаимно дополняющих друг и друга, и функционирующих в симбиозе, потому представлены как одна САТ [9-11]. Ideone.com – ресурс, предоставляющий возможность синтаксической проверки программных кодов и запуска программ на их основе в виртуальной среде на общедоступном сервере [9]. Onlinejudge – это плагин для системы электронного обучения moodle [10, 11], дополняющий стандартный набор доступных видов заданий заданиями по программированию с ответом в виде программного кода; этот плагин позволяет проверять и тестировать программный код как локально на сервере, так и с помощью Ideone.com. В подобной связке интерфейс системы осуществляется через стандартный web-интерфейс moodle.

Onlinejudge при использовании локального сервера обладаем полным контролем над приложениями в среде ОС Linux и лишь частичным в Windows-платформах. Установка и настройка плагина полностью соответствуют стандартным установке и настройке плагинов в системе moodle. Локальная версия Onlinejudge поддерживает небольшое количество языков программирования, увеличение перечня языков программирования происходит ручным редактированием файлов конфигурации. Ideone.com поддерживает более 60 языков программирования, в том числе интерпретаторов, перечень постоянно расширяется администрацией проекта. Документация проектов охватывает необходимый минимум по установке и использованию систем.

### D. Timus Online Judge

Timus Online Judge – разработка Уральского федерального университета [12]. В свободном доступе представляет собой web-ресурс - каталог задач по программированию и их решений на различных языках программирования; по утверждению самих разработчиков - «крупнейший в России архив задач по программированию с автоматической проверяющей системой»

### E. Ejudge

Ejudge – САТ отечественной разработки [13]. Архитектура САТ предусматривает серверное ПО и заготовку интерфейса клиентского доступа через web-интерфейс. Клиентский интерфейс не представлен отдельным полноценным приложением, ресурсом или сервисом. Для обеспечения доступа к системе

пользователям администратору необходимо самостоятельно разработать web-ресурс в соответствии со стандартами Ejudge. Работы возможно только с использованием локального сервера вычислений с операционной системой на основе Linux или Windows (рекомендуется Linux-платформы). Проект обладает объемной документацией, в том числе на русском языке. САТ поддерживает самые распространенные интерпретаторы и компиляторы, расширение их перечня происходит путем редактирования файлов конфигурации.

### F. Contester

Contester – САТ отечественной разработки [14]. Архитектура САТ предусматривает серверное ПО и клиентский доступ через web-интерфейс. Contester предназначен для использования на локальном сервере организации, ОС Linux или Windows – версии для различных типов платформ обладают одинаковым функционалом. Установка САТ происходит с помощью мастера установки. Настройка системы возможна как через web-интерфейс, так и путем редактирования файлов конфигурации. Contester обладает широким перечнем различных заготовок задач и турниров. Документация проекта включает минимальное описание функций и возможностей системы необходимое для использования Contester. Перечень доступных для проверки языков программирования определяется системой автоматически, путем сканирования системы сервера, или редактируется администратором с помощью файлов конфигурации.

## V. ЗАКЛЮЧЕНИЕ

В зависимости от поставленных целей и возможностей пользователей (технические ресурсы, квалификация пользователей, материальное обеспечение процесса тестирования) определяется выбор той или иной САТ [5]. При выборе САТ необходимо четко определить приоритеты применения САТ и выделить критические характеристики для системы, по которым и будет проведен выбор. В тоже время возможно использование нескольких разных САТ для различных целей, однако при использовании различных САТ одной организацией или даже одними и теми же пользователями потребует от пользователей дополнительных усилий в параллельном освоении различных интерфейсов и функционалов, что может существенно сказаться на качестве работы с САТ.

## БИБЛИОГРАФИЯ

- [1] Mikroyannidis A., Domingue J. Interactive learning resources and linked data for online scientific experimentation. Proceedings of the 22nd international conference on World Wide Web companion, WWW '13 Companion, Republic and Canton of Geneva, Switzerland, 2013, pp. 431–434, available at: [dl.acm.org/citation.cfm?id=2487788.2487959](http://dl.acm.org/citation.cfm?id=2487788.2487959).
- [2] Nunome A., Hirata H., Fukuzawa M., Shibayama K. Development of an e-learning back-end system for code assessment in elementary programming practice. Proceedings of the 38th annual ACM SIGUCCS fall conference, SIGUCCS '10, New York, NY, USA, 2010, pp. 181–186, available at: [doi.acm.org/10.1145/1878335.1878381](http://doi.acm.org/10.1145/1878335.1878381).

- [3] Уолтон, Ш. Создание сетевых приложений в среде Linux. Руководство разработчика. [Текст] / Ш. Уолтон. - М.: Вильямс, 2001. - 464 с.
- [4] Лисков, Б. Использование абстракций и спецификаций при разработке программ [Текст] / Б. Лисков, Дж. Гатэг. - М: Мир, 1989. - 424 с.
- [5] Корнеев, Г.А. Автоматизированная система тестирования программ. [Текст] / Г.А. Корнеев // Материалы VIII международной конференции "Современные технологии обучения <<СТО-2002>>". 24 апреля 2002 г. - СПб.: СПбГЭТУ, 2002. - Том 2- С.327-329.
- [6] Майерс, Г. Искусство тестирования программ [Текст] / Г. Майерс под ред. Б.А. Позина. - М.: Финансы и статистика, 1982. - 272 с.
- [7] CSUS Programming Contest Control (PC^2) [Электронный ресурс]. - Режим доступа: <http://pc2.ecs.csus.edu> - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [8] DOMjudge - Programming Contest Jury System [Электронный ресурс]. - Режим доступа: <https://www.domjudge.org> - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [9] Ideone.com - Online Compiler and IDE \_\_ C\_C++, Java, PHP, Python, Perl and 40+ other compilers and interpreters [Электронный ресурс]. - Режим доступа: <http://ideone.com/faq> - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [10] GitHub - hit-moodle\_onlinejudge\_ Online judge plugin for Moodle 1.9.x [Электронный ресурс]. - Режим доступа: <https://github.com/hit-moodle/onlinejudge> - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [11] GitHub - hit-moodle\_moodle-local\_onlinejudge\_ Online Judge plugin for Moodle 2.x [Электронный ресурс]. - Режим доступа: [https://github.com/hit-moodle/moodle-local\\_onlinejudge](https://github.com/hit-moodle/moodle-local_onlinejudge) - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [12] Timus Online Judge [Электронный ресурс]. - Режим доступа: <http://acm.timus.ru> - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [13] ejudge home page \_ ejudge contest management system [Электронный ресурс]. - Режим доступа: <https://ejudge.ru> - Заглавие с экрана. - (Дата обращения 04.12.2015)
- [14] Система Contester [Электронный ресурс]. - Режим доступа: <http://www.contester.ru> - Заглавие с экрана. - (Дата обращения 04.12.2015)

# Selection of Automated testing solutions programming tasks

Aleksey Yakushin, Ilya Gladkikh

***Abstract***— This article is devoted to the analysis of automated testing (SAT) solutions to problems in programming, defining their characteristics and consideration of various representatives of these systems. The analysis was made by studying the documentation SAT, their experimental use, and compared the results of their work. The text of the article proposed various properties of SAT and describes the most common functions and systems, which should help in the selection and the further use of SAT.

***Keywords***— system of automated testing, e-learning, education of programming, verification tasks on programming.