

УДК 004.4

**А.А. Темичев, Р.А. Файзрахманов**Пермский национальный исследовательский политехнический университет,  
Пермь, Россия**АНАЛИТИЧЕСКИЙ ОБЗОР СРЕДСТВ АВТОМАТИЗАЦИИ  
ТЕСТИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ПРИМЕНИТЕЛЬНО  
К СИСТЕМАМ МОНИТОРИНГА**

Рассматриваются основные современные программные средства автоматизации тестирования производительности распределенных приложений, такие как IBM Rational Performance Tester, HP Load Runner, Apache JMeter, SilkPerformer, WAPT, с точки зрения возможности их применения к оценке производительности программного обеспечения (ПО) распределенных систем мониторинга (СМ). Представлена краткая характеристика и приведены основные особенности каждого программного средства автоматизации тестирования. Представлено описание типовой системы мониторинга, приведены особенности тестирования производительности их ПО, которым должны удовлетворять средства оценки производительности. Рассмотрены основные методы тестирования производительности ПО систем мониторинга. Выполнен аналитический обзор программных средств автоматизации тестирования с учетом требований СМ. Приведены результаты сравнительного анализа, показаны недостатки этих средств при использовании с СМ, а также даны возможные пути их устранения.

**Ключевые слова:** аналитический обзор, средства тестирования производительности, нагрузочное тестирование, система мониторинга.

**A.A. Temichev, R.A. Fayzrakhmanov**

Perm National Research Polytechnic University, Perm, Russian Federation

**ANALYTICAL REVIEW OF AUTOMATION PERFORMANCE  
TESTING TOOLS APPLIED TO MONITORING SYSTEMS**

The article deals the basic modern software tools for automate performance testing of distributed applications, such as IBM Rational Performance Tester, HP Load Runner, Apache JMeter, SilkPerformer, WAPT, in terms of their applicability to performance evaluation of software of distributed monitoring systems (MS). Brief characteristics and the main features of each software test automation tools are considered. A short description of a typical monitoring system is presented. The features of performance testing of MS that must be satisfied by performance tools are showed. The basic methods of testing performance of monitoring systems are discussed. The article includes analytical review of software test automation tools with the requirements of monitoring systems. Results of the comparative analysis are provided. In conclusion, disadvantages and limitations of these tools using monitoring system are indicated and the possible ways of their elimination are given.

**Keywords:** analytical review, performance testing tools, load testing, monitoring system.

**Введение.** В связи с постоянным ростом темпов автоматизации различных сфер техногенной деятельности возрастает необходимость в контроле над состоянием различных объектов удаленно, без непосредственного наблюдения. В связи с этим используются специальные автоматизированные системы, получившие название системы удаленного мониторинга или удаленной телеметрии (рис. 1).



Рис. 1. Типовая схема системы мониторинга

Подобные системы позволяют удаленно получать информацию о параметрах мониторируемого объекта из любой точки земного шара, посредством передачи данных через Интернет. Характер мониторируемых объектов является различным – это могут быть и станции метеонаблюдений, и сложные объекты промышленных производств, и объекты энергетической сферы [1], искажение информации о состоянии которых может представлять угрозу жизни и здоровью человека.

Таким образом, вполне логично, что к функционированию таких систем предъявляются достаточно жесткие требования, которым они должны удовлетворять. К основным требованиям можно отнести:

1. **Производительность** – означает способность СМ длительно работать с большими объемами входящих данных с удаленных объектов и сохранять при этом полную работоспособность, избегая таких явлений как порча и потеря данных вследствие неспособности их обработать в установленное время, кратковременное зависание, большое время отклика на действия пользователя и другие подобные проблемы.

2. Корректная обработка дынных – означает способность системы корректно обрабатывать входную информацию, например, данные, не содержащие ошибок, и данные, содержащие различные нежелательные отклонения, такие как сбитые временные метки, неполные сообщения, искаженные значения измеряемых параметров. Система должна автоматически обрабатывать такие данные и, если процент ошибочных данных может повлиять на правильную работу, оповещать об этом оператора.

3. Отказоустойчивость – означает способность системы сохранять работоспособность при различных внештатных ситуациях.

4. Безопасность – означает способность максимально минимизировать вероятность подмены данных, информационного саботажа, что может исказить представление о реальном состоянии удаленных объектов с целью различных действий злоумышленного характера.

И если четвертое требование не полностью относится непосредственно к системам мониторинга, так как безопасность в данном случае существенно зависит от программной платформы, на которой базируется СМ, то первые три являются основополагающими для любой СМ. Соответствие этим требованиям является ключевой задачей при построении СМ.

**Особенности тестирования СМ.** С целью обеспечения соответствия вышеуказанным требованиям СМ подвергаются тщательному тестированию [2, 3]. Следует отметить, что системы мониторинга имеют особенности, которые необходимо учитывать при их тестировании.

Одной из особенностей являются поддерживаемые протоколы передачи данных, так как с развитием Ethernet системы мониторинга уже не являются локальными системами, а могут иметь широкое географическое распределение, исходя из этого современные системы мониторинга создаются с учетом поддержки технологий промышленного Ethernet и соответствующих протоколов. В соответствии с моделью OSI на прикладном уровне такими протоколами являются: HTTP/HTTPS, TCP, Modbus TCP, PROFIBUS, FTP, SNMP др.

Исходя из вышеизложенного, средства оценки производительности должны поддерживать хотя бы часть из этих протоколов, чтобы их можно было использовать для оценки производительности систем мониторинга. Еще одной особенностью систем мониторинга является устройство их системы получения данных с удаленных устройств. Она может быть устроена несколькими способами, основными из которых являются:

1. Способ работы по запросу, где система мониторинга, работая как клиент, опрашивает удалённые устройства с определенным интервалом времени, тем самым запрашивая у них необходимые данные.

2. Способ работы по уведомлению, когда заранее настроенные удаленные устройства отправляют данные системе мониторинга в определенное время, а та, в свою очередь, посылает подтверждение и производит их получение и обработку.

Таким образом, средства тестирования должны поддерживать, по крайней мере, один из этих способов.

К последним особенностям, которым должно удовлетворять средство тестирования производительности, следует отнести способность имитировать множество потоков данных с удаленных устройств и возможность формирования пакетов данных различного содержания.

Исходя из архитектуры СМ, наиболее подходящими методами, с учетом упомянутых особенностей, являются [4, 5, 6] методы тестирования производительности, включающие в себя:

- 1) нагрузочное тестирование;
- 2) тестирование стабильности;
- 3) стресс-тестирование;
- 4) конфигурационное тестирование.

Рассмотрим каждый вид тестирования в отдельности:

Тестирование производительности – в инженерии программного обеспечения – это тестирование, которое проводится с целью определения, как быстро работает вычислительная система или её часть под определённой нагрузкой. Также оно может служить для проверки и подтверждения других атрибутов качества системы, таких как масштабируемость, надёжность и потребление ресурсов. Тестирование производительности принято подразделять на следующие направления:

1. Нагрузочное тестирование – определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству). Нагрузочное тестирование обычно проводится для того, чтобы оценить поведение приложения под заданной ожидаемой нагрузкой. Этой нагрузкой может быть, например, ожидаемое количество одновременно работающих пользователей приложения, а в случае СМ – количество подключенных к системе ПЛК, совершающих заданное число транзакций за интервал времени.

2. Тестирование стабильности – проводится, чтобы убедиться в том, что приложение выдерживает ожидаемую нагрузку в течение длительного времени. При проведении этого вида тестирования осуществляется наблюдение за потреблением приложением памяти для выявления потенциальных утечек. Кроме того, такое тестирование выявляет деградацию производительности, выражающуюся в снижении скорости обработки информации и/или увеличении времени ответа приложения после продолжительной работы по сравнению с началом теста.

3. Стресс-тестирование – выполняется с целью оценки надёжности и устойчивости системы в условиях превышения пределов нормального функционирования. Этот тип тестирования проводится для определения надёжности системы во время экстремальных или диспропорциональных нагрузок и отвечает на вопросы о достаточной производительности системы в случае, если текущая нагрузка сильно превысит ожидаемый максимум. Обычно стресс-тестирование лучше обнаруживает устойчивость, доступность и обработку исключений системой под большой нагрузкой, чем то, что считается корректным поведением в нормальных условиях.

4. Конфигурационное тестирование – выполняется для тестирования производительности не системы с точки зрения подаваемой нагрузки, а с точки зрения эффекта влияния на производительность изменений в конфигурации. Примером такого тестирования могут быть эксперименты с различными методами балансировки нагрузки. Конфигурационное тестирование также может быть совмещено с нагрузочным, стресс-тестированием или тестированием стабильности.

**Средства оценки производительности.** К наиболее известным средствам, способным выполнять вышеприведенные виды тестирования, относят:

- 1) IBM Rational Performance Tester;
- 2) HP Load Runner;
- 3) Apache JMeter;
- 4) SilkPerformer;
- 5) WAPT.

Рассмотрим каждое из них подробнее: IBM Rational Performance Tester (RPT) [7] – инструмент тестирования производительности, разрабатываемый подразделением IBM Rational Software, используется для измерения производительности и масштабируемости веб- и основанных на клиент-серверной архитектуре приложений. Выполнение такого рода

тестов позволяет определить узкие места во взаимодействии компонентов приложения и изыскать способы их возможного устранения.

К основным возможностям и особенностям RPT относятся:

1) поддержка широкого спектра приложений и протоколов, таких как HTTP/ HTTPS, SAP, Siebel, SIP, TCP Socket, Citrix, Windows Sockets и др.;

2) возможность быстрого создания тестов без написания кода и без требования навыков программирования;

3) наличие настраиваемого графического интерфейса пользователя;

4) наличие полнофункционального редактора тестов с древовидной структурой, обеспечивающего как общее высокоуровневое, так и подробное представление тестов;

5) автоматизация изменения данных тестирования, а также возможность вставлять пользовательский код на языке программирования Java для гибкой настройки теста;

6) автоматизация идентификации динамических характеристик сервера и управление ими;

7) гибкое моделирование различной пользовательской нагрузки;

8) возможность тестирования в средах ОС Windows, Linux и средах, построенных на основе технологий мэйнфреймов;

9) формирование отчетов в режиме реального времени, что позволяет немедленно выявлять проблемы производительности и воспроизводить представление веб-страниц в формате HTML в ходе тестирования;

10) сбор и интеграция данных о серверных ресурсах с данными о производительности приложений, получаемыми в режиме реального времени;

11) поддержка операционных систем семейства Windows и Linux благодаря лежащей в основе платформе Eclipse, написанной на языке программирования высокого уровня Java;

12) низкие требования к объему памяти и вычислительной мощности процессора, что обеспечивает проведение масштабного тестирования в многопользовательской среде при ограниченных аппаратных ресурсах;

13) наличие Performance Testing SDK (инструментарий разработчика ПО), позволяющее разработчику программного обеспечения использовать функции RPT.

HP LoadRunner [8] – это программный комплекс для автоматизированного нагрузочного тестирования, разрабатываемый компанией HP. На сегодняшний день HP LoadRunner занимает 77 % на рынке автоматизированного нагрузочного тестирования. HP LoadRunner позволяет эмулировать одновременное обращение сотен и тысяч пользователей, чтобы изучить поведение приложения при реальной нагрузке. При этом собирается информация о ключевых компонентах инфраструктуры (веб-серверы, серверы баз данных и т.д.). Результат может быть детально проанализирован, чтобы понять причины конкретного поведения системы. Программа имеет соответствующие наборы инструментов для проведения тестирования. Также в состав HP LoadRunner входит набор инструментов для работы по различным протоколам с приложением (удаленно, через прокси-сервер и т.п.).

К основным возможностям и особенностям HP LoadRunner относятся:

1) поддержка множества приложений и протоколов, таких как: HTTP/ HTTPS /HTML, TrueClient, SAP, LDAP, Flex, .NET, GUI(UFT), Citrix, RDP, ODBC,COM/DCOM, IMAP,MAPI, POP3, SMTP, FTP, Windows Sockets, CORBA, RMI, JMS, Jacada, Siebel и др.;

2) возможность интеграции в интегрированные среды разработки Microsoft Visual Studio и Eclipse;

3) поддержка операционных систем семейства Windows и Linux;

4) наличие гибкого графического интерфейса пользователя;

5) поддержка мониторинга широкого спектра параметров производительности;

6) наличие богатых средств визуализации, построения отчетов и анализа данных, собранных в ходе тестирования производительности;

7) наличие большого набора инструментов написания тестовых сценариев (скриптов), включая возможность записи действий пользователя при составлении сценариев тестирования с возможностью последующего редактирования записанного сценария;

8) возможность детального анализа и контроля процесса выполнения сценария с возможностью просмотра всех промежуточных характеристик в режиме реального времени;

9) поддержка локализации на русском языке;

10) наличие возможности интеграции с облаком;

11) управления процессом тестирования через мобильное приложение для операционной системы Google Android.

Apache JMeter [9] – инструмент для проведения тестирования производительности, являющийся свободным программным обеспечением с открытыми исходным кодом, написанным на языке программирования высокого уровня Java и разрабатываемым Apache Software Foundation. Изначально Apache JMeter разрабатывался как средство тестирования производительности веб-приложений, впоследствии список поддержки различных технологий расширился. Он может использоваться для моделирования большой нагрузки на сервере, группе серверов, сети, чтобы протестировать их максимальную нагрузочную способность или разложить общую производительность под различными типами загрузки. Apache JMeter способен выполнять тестирование производительности на динамичных и статичных ресурсах.

К основным возможностям и особенностям Apache JMeter относятся:

1) поддержка широкого перечня приложений/протоколов, таких как: HTTP/HTTPS, SOAP, FTP, JDBC, LDAP, JMS, SMTP, POP3, IMAP, MongoDB, TCP и др.;

2) поддержка собственных команд и сценариев тестирования

3) поддержка мультиплатформенности благодаря 100%-ной разработке на языке программирования высокого уровня Java;

4) поддержка анализа загрузки сети;

5) полная поддержка многопоточности, что позволяет выполнять параллельную обработку запросов, функций и других возможных операций;

6) наличие ясного и понятного графического интерфейса пользователя, что позволяет быстро создавать и корректировать выполняемые тесты, а также устранять различные неисправности;

7) поддержка кэширования и оффлайновое исследование/воспроизведение результатов тестирования;

8) заменяемые генераторы нагрузки, что дает неограниченные возможности тестирования;

9) поддержка параллельного сбора статистической информации одновременно несколькими модулями анализа;

10) поддержка различных дополнений для визуализации и анализа данных, что позволяет гибко настроить систему «под себя»;

11) возможность изменения теста прямо в процессе работы, а также возможность гибкого манипулирования данными;

12) поддержка большого спектра языков скриптования сценария тестового испытания (BSF- и JSR223 – совместимые языки).

SilkPerformer [10] – инструмент для автоматизированного нагрузочного тестирования различного уровня сложности. Инструмент создан компанией Borland, которая в настоящее время приобретена британской компанией Micro Focus. SilkPerformer является мощным и в то же время простым в использовании инструментом нагрузки и стресс-тестирования корпоративного класса. Визуальный сценарий и возможность тестирования нескольких прикладных сред с тысячами виртуальных пользователей позволяют тщательно проверить корпоративные приложения на надежность, производительность и масштабируемость, прежде чем они будут развернуты, независимо от их размера и сложности. Мощный анализ первопричин и инструменты управления отчетностью помогают изолировать проблемы и быстро принимать решения.

К основным возможностям и особенностям SilkPerformer относятся:

1) поддержка множества приложений/протоколов, таких как: HTTP/ HTTPS, TTP(S)/HTML, SNMP, SOAP, FTP, LDAP, MAPI, IMAP, DCOM, SMTP, POP3, SSL, .NET, Citrix , TCP/IP, UDP, Jacada и др.;

2) создание тестов и прогонов приложения с помощью интуитивно понятного интерфейса SilkPerformer или с использованием интегрированной среды Eclipse;

3) возможность различной визуализации данных, построения отчетов и анализа результатов тестирования;

4) тестирование в широком диапазоне корпоративных сред с помощью гибких, совместно используемых, мультипротокольных типов виртуальных пользователей;

5) анализ нагрузочных тестирований в реальном времени во избежание появления недействительных результатов тестирования, требующих продолжительных повторных прогонов прецедентов тестирования;

6) наличие возможности работы в распределенной конфигурации, с целью создания большей нагрузки;

7) легко настраиваемые тесты со случайными пользовательскими данными – без необходимости написания хотя бы одной строки кода;

8) наличие возможности гибкой настройки тестового сценария с использованием встроенного языка программирования BDL (Benchmark Description Language).

WAPT [11] является инструментом для нагрузочного и стрессового тестирования веб-сайтов и любых приложений, подобных по строению и архитектуре. Разрабатывается в новосибирской компании SoftLogica LLC. Продукт создает нагрузку на тестируемый сайт путем эмуляции типичной активности сотен или даже тысяч пользователей, работающих с сайтом одновременно. В процессе теста сайт ведет себя таким же образом, как и при реальной нагрузке того же уровня, выдавая те же параметры производительности.

К основным возможностям и особенностям WAPT относятся:

1) поддержка широкого перечня приложений/протоколов, таких как: HTTP/HTTPS, SOAP, JSON, GWT, .NET, Silverlight, SNMP, WNI и др.;

2) поддержка локализации на русский язык;

3) поддержка мониторинга параметров производительности сервера и баз данных;

4) наличие графического интерфейса пользователя;

5) возможность работы в распределенной конфигурации;

6) наличие средств построения отчетов и богатые возможности визуализации и анализа параметров производительности;

7) возможность просмотра характеристик производительности в режиме реального времени;

8) наличие специального мастера создания тестового сценария, что позволяет упростить период адаптации пользователя;

9) поддержка записи действий пользователя при составлении сценариев тестирования с последующим редактированием, а также возможность написания скриптов тестового сценария на языке программирования JavaScript.

**Критерии оценки.** С целью проведения сравнительного анализа были выделены две группы критериев. В общем виде выделенные критерии можно подразделить:

1) на общие – первичные критерии выбора, не затрагивающие функциональности;

2) специальные – напрямую связанные с функциональными возможностями.

К общим критериям оценки можно отнести:

- 1) уровень автоматизации – объем ручных операций оператора пользователя;
- 2) доступность (стоимость);
- 3) поддерживаемые операционные системы;
- 4) системные требования;

К специальным же критериям, предъявляемым к средствам тестирования системами мониторинга, исходя из их архитектуры и принципов работы, можно отнести следующие:

- 1) мониторинг показателей производительности;
- 2) ручное конфигурирование содержимого ответа в СМ;
- 3) ручную правку сценария тестирования;
- 4) поддержку необходимых протоколов передачи данных;
- 5) журналирование всех событий;
- 6) возможность независимого ответа на запросы СМ с разных портов, эмулируя множество устройств;
- 7) гибкое управление генерируемой нагрузкой;
- 8) возможность динамического изменения содержимого ответа при повторных запросах от СМ для имитации данных различного содержания: без ошибок, с различными типами ошибок, имитация неполного сообщения.

**Сравнительный анализ.** С целью оценки средств автоматизации тестирования производительности используется следующая целевая функция:

$$G_j = \left( \sum_{i=1}^K g_{ij} \right) \rightarrow \max_j, \text{ при } j = \overline{1-N}, \quad (1)$$

где  $j$  – оцениваемое средство тестирования производительности,  $i$  – индекс критерия оценки,  $K$  – количество критериев для оценки,  $N$  – количество оцениваемых средств тестирования производительности,  $g_{ij}$  – оценка полезности  $j$ -го средства относительно  $i$ -го критерия,  $G_j$  – общая эффективность  $j$ -го средства с критериями,  $j = \overline{1-K}$ .

Для определения весовых коэффициентов примем 5-балльную систему оценки применимости средства тестирования производительности относительно указанного критерия, где 0 – неприемлемо, 1 – плохо, 2 – удовлетворительно, 3 – хорошо, 4 – очень хорошо, 5 – превосходно. Коэффициенты выставляются на основе экспертных оценок (таблица).

Оценки по каждому критерию

| Критерий                                       | IBM Rational Performance Tester | HP Load Runner | SilkPerformer | Apache JMeter | WAPT |
|--|---------------------------------|----------------|---------------|---------------|------|
|  | $g_{ij}$                        |                |               |               |      |
| 1. Поддержка операционных систем               | 5                               | 5              | 2             | 5             | 2    |
| 2. Уровень автоматизации                       | 5                               | 4              | 5             | 3             | 4    |
| 3. Доступность (стоимость)                     | 1                               | 1              | 2             | 5             | 1    |
| 4. Системные требования                        | 4                               | 4              | 5             | 4             | 5    |
| 5. Поддерживаемые протоколы передачи данных    | 5                               | 5              | 5             | 5             | 4    |
| 6. Гибкое управление нагрузкой                 | 5                               | 4              | 4             | 5             | 3    |
| 7. Журналирование всех событий                 | 5                               | 5              | 5             | 4             | 5    |
| 8. Динамическое изменение содержимого ответа   | 2                               | 2              | 1             | 1             | 1    |
| 9. Ручная правка сценария тестирования         | 5                               | 5              | 4             | 5             | 4    |
| 10. Ручное конфигурирование содержимого ответа | 4                               | 4              | 4             | 4             | 2    |
| 11. Возможности мониторинга нагрузки           | 5                               | 5              | 5             | 4             | 5    |
| 12. Средства анализа результатов               | 5                               | 5              | 5             | 3             | 5    |
| $G_j$  | 51                              | 49             | 47            | 48            | 41   |

Для упрощения интерпретации результатов сравнительного анализа на основе расчетов, сделанных в таблице, по формуле (1) построим гистограмму (рис. 2).

Согласно результатам проведенного сравнительного анализа следует, что наиболее эффективными средствами автоматизации процесса нагрузочного тестирования применительно к использованию в системах мониторинга являются продукты компаний IBM и HP. Они макси-

мально из всех представленных средств поддерживают все необходимые требования со стороны систем мониторинга, обладают очень хорошими средствами визуализации и анализа данных тестирования, а также, что важно, возможностью ручной правки тестовых сценариев и скриптов. Общим недостатком этих средств, так же, как и у SilkPerformer и WAPT, является достаточно высокая стоимость, особенно в случае IBM и HP. В противовес этому средство Apache JMeter является полностью бесплатным аналогом, с открытым исходным кодом, что в перспективе позволяет адаптировать его к применению совместно с любой системой мониторинга путем внесения изменений в исходные коды. Данного преимущества лишены все остальные представленные в обзоре средства, так как, являясь закрытыми продуктами, не позволяют изменять и подстраивать себя под систему, не предусмотренную разработчиком.

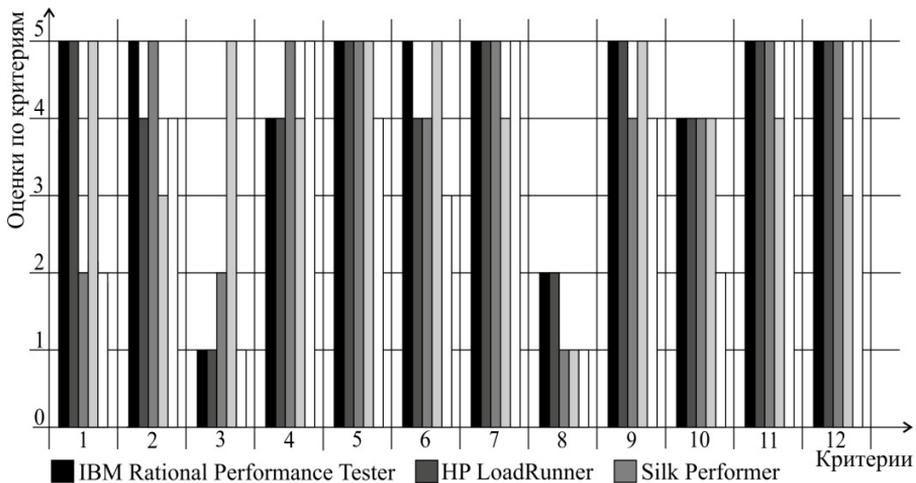


Рис. 2. Диаграмма результатов оценки

Еще одним недостатком или ограничением, идущим от изначального предназначения представленных в обзоре средств, а именно работы с веб-приложениями с точки зрения применимости к оценке производительности систем мониторинга, является в большинстве случаев невозможность работы как сервера, а не как клиента, что делает трудновыполнимой задачу тестирования систем, работающих по схеме, когда удаленные устройства системы мониторинга выступают в роли серверов, а не клиентов по отношению к серверу системы мониторинга, отдавая данные по запросу от СМ.

Говоря об общем применении средств автоматизации нагрузочного тестирования, следует отметить, что они не способны обеспечить гарантированный поиск и устранение проблемных мест, так как они могут только найти только те узкие места, на поиск которых они запрограммированы. Если потенциальное узкое место в системе будет существовать в ее непроверенной области, то средство автоматизации не сможет найти проблему, и система может быть реально применена с наличием таких проблемных мест, которые в будущем могут дать нежелательный эффект, что приведет к их затратному решению уже в процессе эксплуатации.

**Заключение.** Таким образом, дальнейшей перспективой развития средств автоматизации тестирования видится внедрение в них интеллектуальных компонентов [12, 13], способных принимать решения в процессе выполнения тестирования не по жестко заданным алгоритмам и сценариям, а с учетом изменяющихся факторов и информации, получаемой с различных средств мониторинга, производительности, и ошибок, а также на основе предыдущего опыта.

### **Библиографический список**

1. Кычкин А.В. Долгосрочный энергомониторинг на базе программной платформы OpenJEVis // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2014. – № 1(9). – С. 5–15.
2. Темичев А.А., Кычкин А.В. Программный симулятор ПЛК VIDA350 системы энергоменеджмента Му-JEVis // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2011. – № 5. – С. 210–220.
3. Файзрахманов Р.А., Темичев А.А. Подход к тестированию программного обеспечения распределенных систем мониторинга на этапе эксплуатации // Инновационные технологии: теория, инструменты, практика (Innotech 2013): материалы V Междунар. интернет-конф. молод. ученых, аспирантов, студентов; Пермь, 1–30 ноября 2013 г. – Пермь, 2014. – С. 264–268.

4. Richard B. ISTQB 100 Success Secrets: ISTQB Foundation Certification Software Testing. – The ISTQB Certified Software Tester 100 Most Asked Questions. – Emereo Publishing, 2008. – 170 p.

5. Crispin L., Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams. – Addison Wesley Professional, 2009. – 533 p.

6. Glenford J. Myers, Sandler C., Badgett T. The Art of Software Testing. – 3rd ed. – Wiley, 2011. – 240 p.

7. Справочные материалы по Rational Performance Tester [Электронный ресурс] // Сайт компании IBM. – URL: <http://www.ibm.com/developerworks/downloads/r/rpt/> (дата обращения: 05.11.2014).

8. Справочные материалы по программе LoadRunner [Электронный ресурс] // Сайт компании HP. – URL: <http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/> (дата обращения: 05.11.2014).

9. Справочные материалы по программе JMeter [Электронный ресурс] // Сайт Apache Software Foundation. – URL: <https://jmeter.apache.org/index.html> (дата обращения: 05.11.2014).

10. Справочные материалы по программе SilkPerformer [Электронный ресурс] // Сайт Micro Focus. – URL: <http://www.borland.com/Products/Software-Testing/Performance-Testing/Silk-Performer/Features> (дата обращения: 05.11.2014).

11. Справочные материалы по программе WAPT [Электронный ресурс] // Сайт SoftLogica Inc. – URL: <http://www.loadtestingtool.com> (дата обращения: 05.11.2014).

12. Файзрахманов Р.А., Долгова Е.В., Рахманов А.А. Задача адаптивного управления насосным оборудованием на основе интеллектуальных технологий // Электротехника. – 2013. – № 11. – С. 17–19.

13. Петроченков А.Б., Кычкин А.В., Темичев А.А. Интегрированная диагностическая поддержка распределённой системы мониторинга энергоданных // Автоматизация и современные технологии. – 2014. – № 9. – С. 10–16.

## References

1. Kychkin A.V. Dolgosrochnyi energomonitoring na baze programmnoi platformy OpenJEVis [Long-term energy monitoring based on software platform OpenJEVis]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2014, no. 1(9), pp. 5-15.

2. Temichev A.A., Kychkin A.V. Programmnyi simulator PLK VIDA350 sistemy energomenedzhmenta My-JEVis [Software PLC simulator VIDA350 of energy management system My-JEVis]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2011, no. 5, pp. 210-220.

3. Faizrahmanov R.A., Temichev A.A. Podkhod k testirovaniuu programmogo obespecheniia raspredelennykh sistem monitoringa na etape ekspluatatsii [The approach to software testing of distributed monitoring systems during their operation]. *Materialy V Mezhdunarodnoi internet-konferentsii molodykh uchenykh, aspirantov, studentov "Innovatsionnye tekhnologii: teoriia, instrumenty, praktika"*. Perm', 2014, pp. 264-268.

4. Richard B. ISTQB 100 Success Secrets: ISTQB Foundation Certification Software Testing. The ISTQB Certified Software Tester 100 Most Asked Questions. *Emereo Publishing*, 2008. – 170 p.

5. Crispin L., Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams. *Addison Wesley Professional*, 2009. 533 p.

6. Glenford J. Myers, Sandler C., Badgett T. The Art of Software Testing. *Wiley*, 2011. 240 p.

7. Spravochnye materialy po Rational Performance Tester [Reference materials for Rational Performance Tester]. *Sait kompanii IBM*, available at: <http://www.ibm.com/developerworks/downloads/r/rpt/> (accessed 05 November 2014).

8. Spravochnye material po programme LoadRunner [LoadRunner program reference materials]. *Sait kompanii HP*, available at: <http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/> (accessed 05 November 2014).

9. Spravochnye materialy po programme JMeter [JMeter program reference materials]. *Sait Apache Software Foundation*, available at: <https://jmeter.apache.org/index.html> (accessed 05 November 2014).

10. Spravochnye materialy po programme SilkPerformer [SilkPerformer program reference materials]. *Sait Micro Focus*, available at: <http://www.borland.com/Products/Software-Testing/Performance-Testing/Silk-Performer/Features> (accessed 05 November 2014).

11. Spravochnye materialy po programme WAPT [WAPT program reference materials]. *Sait SoftLogica Inc*, available at: <http://www.loadtestingtool.com> (accessed 05 November 2014).

12. Faizrakhmanov R.A., Dolgova E.V., Rakhmanov A.A. Zadacha adaptivnogo upravleniia nasosnym oborudovaniem na osnove intellektual'nykh tekhnologii [The issue of the pumping equipment adaptive control on the basis of intellectual technologies]. *Elektrotehnika*, 2013, no. 11, pp. 17-19.

13. Petrochenkov A.B., Kychkin A.V., Temichev A.A. Integrirovannaia diagnosticheskaia podderzhka raspredelennoi sistemy monitoringa energodannykh [Integrated diagnostic support of distributed data monitoring system]. *Avtomatizatsiia i sovremennye tekhnologii*, 2014, no. 9, pp. 10-16.

### **Сведения об авторах**

**Темичев Андрей Александрович** (Россия, Пермь) – аспирант кафедры информационных технологий и автоматизированных систем Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: schwarzeschwert@yandex.ru).

**Файзрахманов Рустам Абубакирович** (Россия, Пермь) – доктор экономических наук, профессор, заведующий кафедрой информационных технологий и автоматизированных систем Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: itas@pstu.ru).

### **About the authors**

**Temichev Andrey Aleksandrovich** (Perm, Russian Federation) is a postgraduate student of the Department of Information Technologies and Computer-Aided Systems, Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., schwarzeschwert@yandex.ru).

**Fayzrakhmanov Rustam Abubakirovich** (Perm, Russian Federation) is a Doctor of Economic Sc., Professor, the head at the Department of Information Technologies and Computer-Aided Systems, Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: itas@pstu.ru).

Получено 06.07.2015