

17. Мальцев И.М., Михайлов К.А., Михайлова Н.А. Практические аспекты построения учебных планов на базе ФГОС // Интеграл. – 2011. – № 4. – С. 113.
18. Развернутое руководство по использованию программного комплекса «Планы» / Сайт лаборатории математического моделирования и информационных систем (ММиИС). URL: <http://www.mmis.ru/Portals/0/Planu.pdf> (дата обращения: 20.03.2015).
19. Автоматизированная система «Нагрузка ВУЗа». Сайт лаборатории математического моделирования и информационных систем (ММиИС). URL: <http://www.mmis.ru/Default.aspx?tabid=170> (дата обращения: 20.03.2015).
20. Гаврилец Е.З., Медведева О.А. Автоматизированная система формирования учебных планов и распределения учебной нагрузки преподавателей кафедры вуза // Современные наукоемкие технологии. – 2007. – № 2. – С. 40-41.
21. Аврамова О.Д., Зуева С.Е., Наумова Т.А. и др. Автоматизированная информационная система «Учебный план». – М.: Изд-во МГУ, 2006. – 83 с.
22. Аврамова О.Д., Болотова И.Н., Владимиров А.В. и др. Автоматизированная информационная система «Педагогическая нагрузка». – М.: Изд-во Моск. ун-та, 2011. – 46 с.
23. Мунтян Е.Р., Поленов М.Ю., Костюк А.И. Автоматизированная программная среда поддержки управленческих решений // Известия ЮФУ. Технические науки. – 2014. – № 6 (155). – С. 101-108.
24. Язык XML – Описание технологии. URL: <http://www.codenet.ru/webmast/xml/part2.php> (дата обращения: 20.03.2015).
25. Старых В.А., Дунаев С.Б., Коровкин С.Д. Спецификация и форматы обмена данными в разнородных информационных системах на базе XML-технологий. URL: <http://www.citforum.ru/internet/xml/xmltech/> (дата обращения: 20.03.2015).

Статью рекомендовал к опубликованию д.т.н., профессор А.М. Белевцев.

**Мунтян Евгения Ростиславна** – Южный федеральный университет; e-mail: [ermuntyan@sfedu.ru](mailto:ermuntyan@sfedu.ru); 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371550; кафедра вычислительной техники; старший преподаватель.

**Поленов Максим Юрьевич** – e-mail: [mypolenov@sfedu.ru](mailto:mypolenov@sfedu.ru); кафедра вычислительной техники; к.т.н.; доцент.

**Костюк Андрей Иванович** – e-mail: [aikostyuk@sfedu.ru](mailto:aikostyuk@sfedu.ru); тел.: 88634371608; кафедра вычислительной техники; к.т.н.; доцент.

**Muntyan Evgenia Rostislavna** – Southern Federal University; e-mail: [ermuntyan@sfedu.ru](mailto:ermuntyan@sfedu.ru); 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371550; the department of computer engineering; senior lecturer.

**Polenov Maxim Yuryevich** – e-mail: [mypolenov@sfedu.ru](mailto:mypolenov@sfedu.ru); the department of computer engineering; cand. of eng. sc.; associate professor.

**Kostyuk Andrey Ivanovich** – e-mail: [aikostyuk@sfedu.ru](mailto:aikostyuk@sfedu.ru); phone: +78634371608; the department of computer engineering; cand. of eng. sc.; associate professor.

УДК 681.142

**В.А. Балыбердин, А.М. Белевцев, О.А. Степанов**

## **АНАЛИЗ ОСНОВНЫХ ПРОЦЕССОВ ОБЕСПЕЧЕНИЯ НАДЁЖНОСТИ ПРОГРАММНЫХ СРЕДСТВ АСУ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ**

*Анализируются базовые процессы и процедуры обеспечения надёжности программных средств (ПС), являющиеся предметом обсуждения при подготовке и формировании современных стандартов в области качества ПС. Целью анализа является акцентирование усилий разработчиков на основных процессах обеспечения надёжности ПС с учетом их вклада в повышение общей надёжности создаваемых ПС как важного компонента специа-*

лизированных АСУ. Отмечается, что особенно большие затраты и сложности устранения связаны с ошибками, обнаруживаемыми в процессе эксплуатации ПС. Отсюда вытекает важная задача проверки корректности ПС непосредственно в процессе его создания. Такая проверка должна проводиться на всех этапах создания ПС. Приводится и обсуждается общая схема и содержательная структура рассматриваемых процессов оценки и обеспечения качества (надёжности) ПС. Значительное внимание уделяется вопросам организации и проведения динамического тестирования ПС, включая тестирование на базе спецификаций (стратегия «чёрного ящика») и тестирование на базе структуры ПС (стратегия «белого ящика»). Кратко анализируются различные способы организации тестирования для каждой из рассматриваемых стратегий. Отмечаются их основные достоинства и недостатки. Приводятся рекомендации по их применению в интересах повышения надёжности ПС АСУ специального назначения. При рассмотрении процессов демонстрации ПС выделяется ряд подпроцессов, заключающихся в исследовании прототипа ПС, версии ПС для пользователя и пробной версии ПС. Приводится вербальная характеристика каждого из этих подпроцессов. Различия между этими подпроцессами заключаются в степени реализации задаваемых требований к ПС. Приводятся рекомендации по использованию рассмотренных процедур для повышения надёжности ПС. В выводах работы отмечается, что при практической организации работ по созданию высоконадёжных ПС особое внимание следует обращать на внедрение методов и способов обеспечения надёжности ПС на ранних этапах создания ПС, поскольку стоимость последующего исправления допущенных на этих этапах ошибок в ПС существенно возрастает.

*Программные средства; надёжность; тестирование; верификация; валидация.*

**V.A. Baliberdin, A.M. Belevtsev, O.A. Stepanov**

#### **ANALYSIS OF BASIC SOFTWARE RELIABILITY CONTROL PROCESSES FOR SPECIAL PURPOSE COMPUTER SYSTEMS**

*Basic processes and procedures to ensure software reliability being the subject matter of software quality are analysed. The basic processes to ensure software reliability are treated taking into account their contribution into total reliability of software being produced. It is pointed out that the most expensive software defects are those found in the course of software exploitation. Now an imported problem of software correctness control in the course of its development follows. Such actions may be initiated during all the stages of software development. A basic diagram and the major structure of the processes involved are discussed. Much consideration is received to software dynamic testing including "black box" and "white box" testing. A variety of techniques for testing organizations for each the strategies covered are analyzed in short. The main strengths and weaknesses are noted out. Some recommendations for these methods applications are given. In software demonstration processes some subprocesses are stood out such as software prototype, user version and mock-up. Verbal characteristics of each the processes are given. The distinctions between the processes lay with the level of basic requirements for software. Some recommendations for using these procedures to increase the software reliability are given. It is stressed in draw that during the reliable software development processing much considerations should be paid to promoting the techniques for each reliable software development stage. In view of the fact that the cost of removing these defects later is high.*

*Software; reliability; software testing; verification; validation.*

Среди характеристик качества программных средств (ПС) АСУ специального назначения (СН) надёжность ПС занимает особое место [1, 9–13]. Известно, что ПС АСУ, являясь продуктом интеллектуального труда, неизбежно содержит некоторое число дефектов – ошибок, допущенных человеком-разработчиком ПС. Известны примеры серьезных ошибок в ПС, приведших к потере человеческих жизней, космических аппаратов или к масштабным нарушениям работы инфраструктурных сетей [1, 2, 23, 24]. По мере развития АСУ СН возрастает сложность ПС АСУ, что приводит к увеличению количества ошибок в нем и, соответственно, – росту ущерба от этих ошибок [9–11]. Поэтому исследование проблемы обеспечения надёжности ПС АСУ СН приобретает важное значение.

В работе авторов [1] рассмотрен ряд общих вопросов, связанных с оценкой и обеспечением надежности ПС АСУ СН и определены некоторые первоочередные мероприятия, направленные на решение рассматриваемой проблемы. Целью настоящей работы является акцентирование усилий разработчиков на основных процессах обеспечения надёжности ПС с учетом их вклада в повышение общей надёжности создаваемых ПС как важного компонента специализированных АСУ

Опытным путем установлено, что чем позже обнаружена ошибка в ПС, тем выше затраты на её устранение. Особенно большие затраты и сложности устранения связаны с ошибками, обнаруживаемыми в процессе эксплуатации ПС [5, 6, 19–21].

На рис. 1 представлена диаграмма, характеризующая относительные затраты на исправление допущенных ошибок в ПС на различных этапах тестирования и при эксплуатации ПС. Диаграмма составлена по материалам [5, 6].

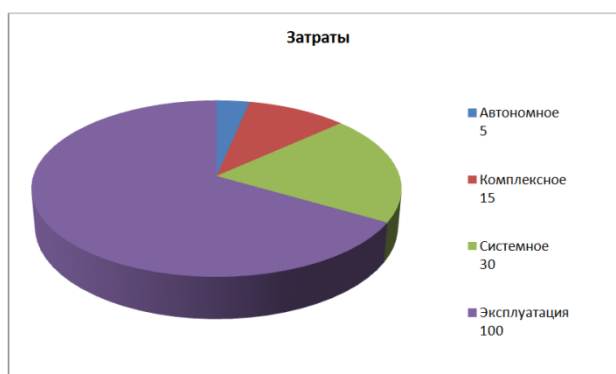


Рис. 1. Относительная величина затрат на устранение ошибки в ПС на различных этапах тестирования ПС и при их эксплуатации

Соответствующие процедуры осуществления такой проверки получили наименование тестирования, верификации и валидации ПС. В известной литературе существуют некоторые расхождения в трактовке указанных терминов. Поэтому целесообразно внести определенные уточнения, опираясь на имеющиеся исследования, проведенные в последнее время [1–4].

*Тестирование* ПС – это процесс исполнения ПС, направленный на обнаружение дефектов (ошибок) в ПС. Здесь важно подчеркнуть два момента – это целевая установка процесса (поиск и обнаружение ошибок) и исполнение ПС (тестирование будем относить лишь к исполняемым элементам ПС). Заданная целевая установка побуждает именно к обнаружению и исправлению допущенных ошибок в ПС, а не к демонстрации работоспособности ПС, что имеет большой практический смысл. И второй момент – понятие «тестирование» следует относить лишь к исполняемым элементам ПС, хотя имеется и более широкая трактовка этого понятия [7]. Однако в отечественной литературе утвердилась приведенная выше трактовка, которой и будем далее придерживаться [2–4, 14, 20–22].

Более общим, чем тестирование, является понятие *верификации* ПС.

Целью процесса верификации ПС является подтверждение того, что верифицируемый объект (описание, проект, программный код и т.д.) соответствует предъявленным ПС требованиям. Процесс верификации включает в себя различного рода проверки (инспекции), тестирование кода, анализ промежуточных документов и т.д.

*Валидация* ПС – это процесс, целью которого является подтверждение того, что созданные ПС удовлетворяют ожиданиям заказчика (пользователя) [8].

Иными словами, если в процессе верификации ПС проверяют их соответствие заданным требованиям, то при валидации – идут далее, т.е. проверяют соответствие ПС реальным потребностям их применения.

В качестве иллюстрации на рис. 2 приведено схематическое представление взаимосвязи рассмотренных понятий.

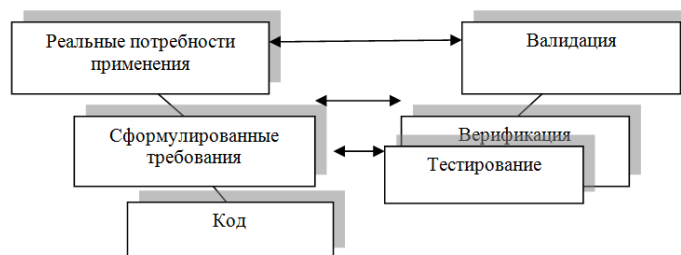


Рис. 2. Взаимосвязь тестирования, верификации и валидации

Вопросы проведения валидации ПС часто связываются с заключительными этапами создания ПС, когда в процессе демонстрации ПС Заказчик начинает осознавать, что заданные им требования к ПС не вполне адекватно отражают его реальные потребности в ПС. Компромисс обычно достигается за счёт корректировки требований и соответствующей доработки ПС. Такой процесс достаточно трудно формализуем, и требует рассмотрения с учетом полной конкретики решаемых задач.

Достаточно полное представление о рассматриваемых процессах оценки и обеспечения качества (надёжности) ПС, по-видимому, содержится в стандарте [7]. Общая схема процессов, затрагиваемых в этом стандарте, включает четыре базовых процесса:

- ◆ статические проверки;
- ◆ динамическое тестирование;
- ◆ демонстрация возможностей;
- ◆ анализ в рамках верификации и валидации ПС.

Каждый из этих процессов, в свою очередь включает ряд подпроцессов. Так процесс статических проверок включает подпроцессы инспекций, обзоров, модельной верификации и статического анализа документов, связанных с разработкой ПС. Каждый из этих подпроцессов имеет свои особенности, вытекающие из практики создания больших систем ПС.

Общим является то, что все действия по статической проверке ПС осуществляются без исполнения программного кода и основываются на рассмотрении и анализе сформулированных требований (спецификации требований, техническое задание), описаний проектных решений (эскизный и технический проекты), исходных кодов (тексты программ), руководств и инструкций, планов проверок и испытаний ПС и т.д.

Опыт разработки больших систем ПС показывает, что такие проверки позволяют выявить существенные, глубинные просчёты при создании ПС и, в конечном счёте, выражаются в экономии материальных затрат и времени на создание ПС.

Динамическое тестирование ПС обычно подразделяют на две большие группы:

- 1) тестирование на базе спецификаций (стратегия «чёрного ящика»);
- 2) тестирование на базе структуры ПС (стратегия «белого ящика»).

Рассмотрим эти подходы подробнее.

Стратегию «чёрного ящика» называют также тестированием с управлением по данным, или тестированием с управлением по входу-выходу. При этом тестируемое ПС рассматривается как некий «чёрный ящик», внутренняя структура которого не анализируется. Задача тестирования заключается в том, чтобы проверить работоспособность ПС на совокупности сочетаний возможных исходных данных.

Ясно, что такая задача может быть решена строго (с уверенностью 100 %), если в качестве тестовых последовательностей будут использованы все возможные наборы входных данных. Однако достичь этого при тестировании реальных ПС, как правило, не удаётся ввиду чрезвычайно большого количества таких наборов данных. Поэтому на практике обычно применяют компромиссный подход, основанный на анализе совокупности входных данных тестируемого ПС и построении множества таких их наборов, которое бы обеспечивало наиболее вероятное «покрытие входа» решаемой задачи.

Стратегию «белого ящика» называют также тестированием, управляемым логикой программы. Существо его сводится к тому, чтобы на базе исследования внутренней логики ПС построить тестовые последовательности, обеспечивающие прохождение различных разветвлений информационно-вычислительного процесса.

На первый взгляд представляется, что достаточно построить такой набор тестовых последовательностей подобного рода, в котором бы каждый оператор выполнялся хотя бы один раз, и задача тестирования будет решена. На самом деле проверка ПС связана не только с выходом на каждый оператор, но и с исходными данными, подаваемыми на каждый оператор. Анализ показывает, что в итоге опять приходим к «проклятию размерности», т.е. к необходимости построения чрезвычайно большого набора входных данных для исчерпывающего тестирования маршрутов в ПС.

В свою очередь в каждой из этих групп используется некоторое число специфических методов тестирования. Представленный, например, в стандарте [7] набор существующих методов тестирования ПС включает около двадцати наименований, что говорит о важности и сложности решения рассматриваемой проблемы.

Количество методов в каждой из групп (1) и (2) – более десятка. Выбор среди этих методов представляет далеко не простую задачу для исследователя. В целом решение такой задачи должно основываться на практическом опыте тестирования ПС. В качестве основных в плане оценки надёжности ПС стандарт [7] рекомендует использовать следующие четыре метода из группы методов на базе спецификации ПС:

- ◆ случайное тестирование;
- ◆ тестирование на базе переходов между состояниями;
- ◆ тестирование возможных ошибок;
- ◆ тестирование на базе анализа граничных значений.

Охарактеризуем кратко эти методы.

Модель случайного тестирования основывается на определении набора возможных значений входных данных объекта тестирования. Функция распределения входных данных при этом может быть выбрана на основе ожидаемого распределения реального входного потока. При отсутствии информации о распределении входного потока обычно принимают равномерное распределение.

Модель тестирования на базе переходов между состояниями основана на определении совокупности состояний, которые может занимать объект тестирования, а также переходов между этими состояниями и результатов таких переходов. Состояния модели должны быть различимы, определены, а их число – конечно.

Тестирование возможных ошибок основывается на модели выявления возможных ошибок, набор которых определяется на основе имеющихся знаний и опыта работы с подобными объектами тестирования.

Модель тестирования на базе анализа граничных значений связывается с разделением входных и выходных данных объекта тестирования на некоторое число наборов с определяемыми границами. Значения на границах, внутри и вне их определяются в зависимости от требуемого уровня покрытия объекта тестирования.

Тестирование в рамках стратегии белого ящика выражается в достижении рациональной степени покрытия тестами логики программы. При этом различают следующие основные методы [5, 7]:

- ◆ покрытие операторов;
- ◆ покрытие решений;
- ◆ покрытие условий;
- ◆ покрытие решений-условий;
- ◆ комбинаторное покрытие условий.

Рассмотрим кратко указанные подходы. Метод *покрытия операторов* предполагает, что набор тестов должен быть построен таким образом, чтобы обеспечить использование каждого оператора по крайней мере один раз. К сожалению, такой критерий покрытия является достаточно слабым, поскольку не позволяет обнаруживать целый ряд ошибок, обусловленных, например, неверной записью переходов в программе.

Более сильным является метод *покрытия решений*, согласно которому формируется набор тестов, обеспечивающих принятие на каждом решении значений «истина» и «ложь», по крайней мере, один раз. Иначе говоря, каждое направление перехода в программе должно быть реализовано хотя бы один раз. Покрытие решений обычно удовлетворяет условию покрытия операторов. Поскольку каждый оператор лежит на некотором пути, то при выполнении каждого перехода каждый оператор должен быть выполнен. Вместе с тем возможны и специальные варианты, не укладывающиеся в эту схему. Например, в случае наличия нескольких входов в программу.

Более сильным, по сравнению с предыдущими, является метод *покрытия условий*. Этот метод требует такого набора тестов, который был бы достаточен для того, чтобы все возможные результаты каждого условия в решении выполнялись, по крайней мере, один раз. Однако поскольку покрытие условий также не всегда приводит к выполнению каждого оператора, то этот метод требует дополнительных усилий по обработке, например, ситуаций с несколькими входами.

Метод *покрытия решений/условий* требует такого набора тестов, при котором бы все возможные результаты каждого из условий в решении выполнялись, по крайней мере, один раз, все результаты каждого решения выполнялись бы, по крайней мере, один раз и каждой точке входа передавалось управление не менее, чем один раз. Метод является достаточно мощным, однако он недостаточно чувствителен к ошибкам в логических выражениях.

Метод *комбинаторного покрытия условий* лишен указанных недостатков. Этот метод требует создания такого числа тестов, при котором все возможные комбинации результатов условия в каждом решении и все точки входа выполнялись бы, по крайней мере, один раз.

Как видно из проведенного анализа, методы тестирования, применяемые в рамках стратегии «белого ящика» достаточно трудоёмки и требуют детального ознакомления со структурой и особенностями конкретной реализации исследуемых ПС. Обеспечить выполнение таких требований на практике не всегда удаётся в достаточной мере. Поэтому прикладное использование методов тестирования в рамках стратегии «белого ящика», по-видимому, следует рекомендовать при проведении тестирования ПС на этапах его создания, когда тестирование выполняется силами Разработчика ПС, которому хорошо известны все особенности программной реализации ПС. Как отмечается в [5], тестирование отдельных модулей ПС может быть ориентировано на использование стратегии «белого ящика». При переходе к комплексному тестированию ПС уже более рациональным является использование стратегии «чёрного ящика».

На тех этапах, на которых ответственность за принятие решений возложена на Заказчика ПС, также целесообразно использовать методы стратегии «чёрного ящика».

Заметим, что в дополнение к рассмотренным группам стратегий тестирования ПС иногда добавляют третью – тестирование на базе имеющегося опыта. Однако этот подход не является систематизированным и не получил большого распространения.

Обратимся теперь к процессам демонстрации ПС. Здесь принято различать следующие подпроцессы, заключающиеся в исследовании:

- ◆ прототипа ПС;
- ◆ версии ПС для пользователя (заказчика);
- ◆ пробной версии ПС (mock-up).

Различие между этими подпроцессами заключается в степени реализации задаваемых требований к ПС.

В прототипе ПС обычно реализованы все основные требования и задача заключается в демонстрации пользователю возможностей ПС. Версия ПС для пользователя это – рабочий вариант ПС, который передается Заказчику для использования в работе с тем, чтобы в последующем учесть и реализовать возможные замечания. Пробная версия ПС (mock-up) имеет основной целью отработку человеко-машинных интерфейсов до наиболее приемлемого Заказчиком варианта.

Далее следуют процессы анализа всего накопленного, в результате реализации предыдущих процессов, материала в интересах верификации и валидации ПС. Для этих целей используется широкий ассортимент методов и подходов, включая моделирование и расчёты с помощью формальных математических методов, доказательства на базе алгебры логики и т.д. [7, 15–18]

В заключение проведенного анализа можно сделать следующие основные выводы.

1. Вопросы обеспечения надёжности ПС приобретают всё более важное значение по мере развития возможностей систем и комплексов управления специального назначения.
2. На международном уровне рекомендована для практического использования определённая последовательность процедур поддержки процессов разработки ПС, ориентированная на создание высоконадёжных ПС. Соответствующие процедуры определяются на уровне международных стандартов, работа над которыми еще полностью не завершена.
3. При практической организации работ по созданию высоконадёжных ПС особое внимание следует обращать на внедрение методов и способов обеспечения надёжности ПС на ранних этапах создания ПС, поскольку стоимость последующего исправления допущенных на этих этапах ошибок в ПС существенно возрастает.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Балыбердин В.А., Белевцев А.М., Степанов О.А.* Вопросы оценки и обеспечения надёжности программных средств АСУ специального назначения // Известия ЮФУ. Технические науки. – 2014. – № 5 (154). – С. 115-120.
2. *Кулямин В.В.* Методы верификации программного обеспечения. – М.: ИСП РАН, 2008. – 117 с.
3. *Синицын С.В., Налютин Н.Ю.* Верификация программного обеспечения. – М.: МИФИ (ГУ), 2006. – 157 с.
4. *Степанченко И.В.* Методы тестирования программного обеспечения. – Волгоград: РПК «Политехник», 2006. – 77 с.

5. *Майерс Г.* Искусство тестирования программ: Пер. с англ. / Под ред. Б.А. Позина. – М.: Финансы и статистика, 1982. – 176 с.
6. *Майерс Г.* Надёжность программного обеспечения: Пер. с англ. / Под ред. В.Ш. Кауфмана. – М.: Мир, 1980. – 360 с.
7. ISO/IEC 29119-1. Systems and Software Engineering. – Part 1: Concepts and Definitions. – 2012.
8. ISO/IEC 25010. Systems and Software Engineering. – Systems and Software Quality Requirements and Evaluation. – 2010.
9. *Липаев В.В.* Надёжность программных средств. – М.: Синтег, 1998. – 222 с.
10. *Полонников Р.И., Никандров А.В.* Методы оценки показателей надёжности программного обеспечения. – СПб.: Политехника, 1992. – 78 с.
11. *Штрик А.А., Осовецкий Л.В., Мессих И.Г.* Структурное проектирование надёжных программ встроенных ЭВМ. – Л.: Машиностроение, 1989. – 296 с.
12. *Холстед М.Х.* Начала науки о программах: Пер. с англ. В.М. Юфы. – М.: Финансы и статистика, 1981. – 128 с.
13. *Липаев В.В.* Процессы и стандарты жизненного цикла сложных программных средств. Справочник. – М.: Синтег, 2006. – 260 с.
14. *Василенко Н.В., Макаров В.А.* Модели оценки надёжности программного обеспечения // Вестник Новгородского государственного университета. – 2004. – № 28. – С. 126-132.
15. *Степанов О.А., Шумило Д.А.* Метод оценки программных средств АСУВ по результатам тестирования // Материалы всероссийской конференции «Современные тенденции развития теории и практики управления в системах специального назначения». – М.: Системпром, 2014. – С. 11-14.
16. *Балыбердин В.А., Белевцев А.А., Степанов О.А.* Об оценке компонентов информационного и лингвистического обеспечения АСУ // Известия ЮФУ, Технические науки. – 2013. – № 5 (142). – С. 34-40.
17. *Балыбердин В.А., Степанов О.А., Шумило Д.А.* К оценке надёжности программных средств АСУВ // Актуальные проблемы защиты и безопасности // Труды 14-й Всероссийской научно-практической конференции. Вооружение и военная техника. Т. 1. – СПб., 2011. – С. 603-608.
18. *Балыбердин В.А., Пенкин О.М., Полунин А.И.* Проблемные вопросы создания и внедрения новых информационных технологий в автоматизированных системах военного назначения. – М.: СВТП РИА, 2001. – 144 с.
19. *Балыбердин В.А., Зубарев И.В., Панов В.В., Степанов О.А.* Прикладные аспекты автоматизации управления войсками и оружием в современных условиях. – М.: 3 ЦНИИ МО РФ, 2013. – 240 с.
20. *Липаев В.В.* Надёжность и функциональная безопасность комплексов программ реального времени. – М.: ИСП РАН, 2013. – 176 с.
21. *Липаев В.В.* Функциональная безопасность программных средств. – М.: СИНТЕГ, 2004. – 348 с.
22. *Жоголев Е.А.* Технология программирования. – М.: Научный мир, 2004. – 216 с.
23. *Kauffels F.-J.* Practical LANs analised. – N.Y.-Toronto: Ellis Horwood Limited, 1989. – 334 p.
24. *Gormer birgit.* Novell Netware. – Munchen: Markt&Technic AG, 1990. – 260 p.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Тютиков.

**Белевцев Андрей Михайлович** – МАТИ – Российский государственный технологический университет имени К.Э. Циолковского; e-mail: ambelevtsev@yandex.ru; г. Москва, Оршанская-3; тел.: +79218550885; д.т.н.; профессор.

**Балыбердин Валерий Алексеевич** – 3 Центральный НИИ Министерства обороны РФ; e-mail: baliberdinv@yandex.ru; г. Москва, Погонный, 10; д.т.н.; профессор; ведущий научный сотрудник.

**Степанов Олег Алексеевич** – e-mail: stepoleg@post.ru; к.т.н.; доцент.

**Belevtsev Andrey Michailovitch** – МАТИ-university; e-mail: ambelevtsev@yandex.ru; 3, Orshanskaiyi, Moscow, Russia; phone: +79218550885; dr. of eng. sc.; professor.



**Baliberdin Valery Alekseevitch** – 3 Central Defence Institute; e-mail: baliberdinv@yandex.ru; 10, Poronny, Moscow, Russia; dr. of eng. sc.; professor; science worker.

**Stepanov Oleg Alekseevich** – e-mail: stepoleg@post.ru; cand. of eng. sc.; associate professor.

УДК 004.891

**Е.В. Корохова, И.С. Шабаршина, А.В. Петракова, А. С. Санина**

### **МОДЕЛИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ ПРОЦЕДУРЫ ВЫБОРА АВТОНОМНОЙ ТЕПЛОГЕНЕРИРУЮЩЕЙ СИСТЕМЫ**

*В условиях многообразия видов теплогенерирующего оборудования, представленного на рынке, неопределенности или нечеткости требований заказчика задача выбора эффективной системы отопления здания рассматривается как многокритериальная и слабоструктурированная, для решения которой должны использоваться современные интеллектуальные методы и системы. Основные критерии, влияющие на выбор отопительной системы, отражают следующие требования потребителей: качество получаемого тепла; затраты на создание и эксплуатацию системы; назначение и характер использования помещения и др. Возрастает востребованность экологически чистых технологий и эргономичных конструктивных решений, повышается интерес к практическому использованию альтернативных источников энергии. Строительные компании предлагают клиенту различные альтернативы реализации теплогенерирующей системы, однако выбор конкретного варианта зачастую производится лишь на основании личного опыта сотрудников данной компании. Поэтому задача создания экспертной системы для поддержки принятия решений по выбору эффективной теплогенерирующей системы является актуальной. Рассмотрены основные этапы разработки экспертной системы на основе производственной модели: на основе проведенного анализа существующих теплогенерирующих систем, используемых видов топлива, требований заказчика и экспертных мнений определено множество возможных вариантов систем; формализованы критерии оценки альтернатив; разработан алгоритм выбора наилучшей альтернативы, построено дерево решений, на основе которого реализован прототип экспертной системы. Научная ценность работы заключается в разработке алгоритма многокритериального выбора автономной отопительной системы на основе построенного дерева решений. Практическую ценность исследования составляют: результаты анализа отопительных систем, позволивших сформировать множество критериев, однозначно определяющих эффективный вариант автономной отопительной системы; демонстрационный прототип экспертной системы, реализующий предложенный алгоритм; дерево решений для выбора эффективной автономной отопительной системы. Предлагаемая экспертная система может быть использована проектными, строительными организациями и частными лицами для принятия решений на основе многокритериального выбора эффективной автономной теплогенерирующей системы.*

*Теплогенерирующая система; экспертная система; дерево решений; моделирование; принятие решений.*

**E.V. Korohova, I.S. Shabarshina, A.V. Petrakova, A.S. Sanina**

### **MODELING AND CONSTRUCTION OF THE ALGORITHM TO SELECT AUTONOMOUS HEAT PRODUCING SYSTEM**

*Regards to the variety of heat producing equipment at the market, uncertainties of the customer's requirements, the problem of selecting the effective heat producing system is analysed as a multiobjective or ill-defined problem. In this case, Modern Smart Methods or Systems should be used to solve the problem. The main criteria, which have an impact on the selection of the heating system, include the following customer's requirements: quality of the heat received, production and implementation costs, purposes and the nature of the used building etc. There appear a demand of using environmentally friendly technologies and ergonomic structural solutions, and also, the more focus is done to the implementation of alternative energy sources. The greater number of construction companies offer*