ПРОБЛЕМЫ ОЦЕНКИ НАДЕЖНОСТИ И КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ

Жарко Е.Ф., Полетыкин А.Г., Промысов В.Г.

Институт проблем управления им. В.А. Трапезникова РАН (ИПУ РАН), 117997, Россия, Москва, Профсоюзная ул., 65, e-mail: v1925@mail.ru

В статье рассматриваются аспекты оценки качества, надежности программного обеспечения в части теоретических основ, методов, основных тенденций и проблем в этой области.

Ключевые слова: надежность, обеспечение качества, программное обеспечение, АСУ ТП

THE PROBLEM OF RELIABILITY AND QUALITY ASSESSMENT OF SOFTWARE IN DIGITAL CONTROL SYSTEMS

Zharko E.F., Poletykin A.G., Promyslov V.G.

V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow, e-mail: v1925@mail.ru

The article describes aspects of software quality and reliability assessment. The aspects concern the theory, methods and general problems.

Key words: reliability, quality, software, I&C

1. Введение

Процесс развития автоматизации сложных технологических объектов, нарушения работоспособности которых приводят к большим экономическим, экологическим потерям, угрозам здоровью или жизни людей, характеризуется тенденцией разработки автоматизированных систем управления технологическими процессами (АСУ ТП), реализующих значительно более сложные алгоритмы управления и анализа данных с использованием сложных программно-технических комплексов [1]. Обеспечение надежности АСУ ТП на всех этапах ее жизненного цикла базируется на качественном и количественном анализе, который, согласно нормативной документации, также должен проводиться на всех этапах. Качественный и количественный анализ надежности должен учитывать две составляющие программно-технических комплексов (ПТК): аппаратную и программную. Однако количественный анализ надежности для программных компонентов комплексов (программ, программного обеспечения (ПО)), в отличие от аппаратной части, имеет трудности [2]. Эти трудности обусловлены отличиями в причинах возникновения отказов в аппаратной и программной составляющих ПТК. Трудность расчета надежности классической программы, функционирующей в рамках универсальной машины Тьюринга, состоит в том, что алгоритм ее функционирования не является случайным. Отказ программы появляется после наложения на детерминированную функцию, которая соответствует алгоритму программы, случайного процесса, описывающему входные данные для нее. Детерминированность алгоритма для классических программ приводит к тому, что в лучшем случае рассчитывается вероятность отказа для комплекса: входные данные + программа. Расчет вероятностных характеристик выходного процесса, даже в случае с известной функцией, может быть трудной задачей, а при наличии ошибок в ее реализации можно считать такую задачу не выполнимой. Проблема была осознана многими специалистами и, так как существует потребность в оценке надежности программ, применяемых в составе различных комплексов, появляются модели и методы, позволяющие оценить надежность программы [3].

Вместе с надежностью, иногда подменяя ее, используют термин «качество» программного обеспечения. Качество программного обеспечения можно определить как соответствие явно установленным функциональным и эксплуатационным требованиям, явно указанным стандартам разработки и неявным характеристикам. Качественные и количественные показатели качества программ, в отличие от вероятностных показателей надежности, можно эффективно использовать для анализа видов и последствий отказов, сравнительного анализа вариантов технических решений по обеспечению надежности, организации технического обслуживания. Качественные и количественные показатели качества программ имеют несомненную практическую ценность.

В статье анализируются наиболее часто встречающиеся методы расчета надежности программного обеспечения, показаны проблемы с применением данных методов. Приведен обзор основных подходов к оценке качества программного обеспечения.

2. Причины отказов аппаратуры и программного обеспечения

В ГОСТ 27.002-89 выделены следующие виды причин возникновения отказов аппаратуры:

1. несовершенство или нарушение установленных правил и (или) норм проектирования и конструирования (конструктивные ошибки);
2. несовершенство или нарушение установленного процесса изготовления или ремонта, выполняемого на ремонтном предприятии;
3. нарушение установленных правил и (или) условий эксплуатации;
4. естественные процессы старения, изнашивания, коррозии и усталости при соблюдении всех установленных правил и (или) норм проектирования, изготовления в эксплуатации.

Наибольший вес в потоке отказов аппаратуры имеют, как правило, отказы 2-го и 4-го вида.

Программа – это совокупность инструкций, выраженных на одном из языков, и записанная на материальном носителе долговременного или временного хранения. Отказ материального носителя программы это отказ аппаратуры, на которой выполняется программа или отказ носителя. Отказ программы проявляется как несоответствие значения на выходе программы заданному значению. Информационное содержание программы само по себе не меняется (не отказывает). Поэтому для отказов программ характерны причины видов 1, 2 и 3. Наибольшую долю среди всех отказов ПО занимают, как правило, отказы, вызванные первой причиной. Главная особенность отказов этого типа как для программ (иначе ошибки ПО), так и для аппаратуры, заключается в том, что вносятся ошибки в программу (аппаратуру) случайно, а проявляются – детерминировано при наступлении определенных событий. Для программ момент наступления события отказа определяется составом и значением набора входных данных, уровнем загрузки вычислительных ресурсов, информационным окружением программы на этапе ее выполнения и подобными факторами.

Накоплен большой опыт и теоретическая база по количественным методам анализа надежности аппаратных средств. Программа не может измениться во времени без изменения свойств материального носителя **сама по себе** и ее отказ есть проявление ошибок, содержавшихся в программе. Количественный анализ отказов программы имеет ряд проблем:

- сложность получения аналитического выражения для функции, описывающей работу программы;
- случайный процесс, связанный с входными данными, уровнем загрузки вычислительных ресурсов, информационным окружением может иметь сложное или не известное распределение;
- наличие ошибок в программе, нелинейно влияют на вид функции, описывающей работу программы, и вид этой функции не известен.

В работе [4] приводится классификация типов ошибок в программе по их происхождению:

1. Системные ошибки при постановке целей и задач создания программы;
2. Ошибки программирования в текстах программ и описаниях данных (синтаксические ошибки);
3. Алгоритмические ошибки разработки при непосредственном формулировании требований к функциям программы и алгоритмические ошибки реализации этих требований.

Первый тип ошибок не является специфическим для программного обеспечения и не является предметом рассмотрения. Подавляющее большинство ошибок второго типа исключается средствами автоматической проверки программ (компиляторами). С *алгоритмическими ошибками* дело обстоит иначе: убедиться перед эксплуатацией программы в том, что она работает правильно и в ней нет алгоритмических ошибок, можно только в процессе *тестирования* программы (тестирование позволяет выявлять все типы ошибок). Из-за большой области проверки, тестовое покрытие для любой реальной программы не является полным, т.е. всегда остается возможность того, что в программе существуют ошибки.

В классических языках программирования известно, что количество ошибок зависит от объема исходного кода программы, технологии программирования, квалификации персонала участвующего в разработке программы и средств, выделенных на тестирование [5]. Данные показатели могут считаться константами для замкнутых групп разработчиков с устоявшимися нормами разработки и тестирования.

Однако нелинейная связь между количеством ошибок в программе и вероятностью их проявления при использовании программы приводит к негативным результатам при попытке использовать оценку по количеству ошибок в программе для расчета вероятности ее отказа [4]. Несмотря на проблемы с обоснованием применимости вероятностных методов оценки надежности программного обеспечения, разработано и применяются большое количество методов количественной оценки надежности программ. Ниже приведены наиболее часто используемые методы и указаны проблемы с их применением.

3. Методы анализа надежности программ

Существует большое разнообразие областей применения моделей с точки зрения моделирования аппаратных и программных отказов, однако наибольшее внимание уделяется моделям оценки надежности программного обеспечения, способными быть интегрированными в существующую комплексную модель расчета надежности системы управления. В комплексной модели учитываются последствия видов отказов компонентов в цифровой системе в целом для объекта. Основные методы анализа надежности классифицируют в соответствии с их главной целью в соответствии с тем, как осуществляется анализ архитектуры программной системы:

1) Восходящий метод (главным образом направленный на исследования последствий единичных неисправностей):

a) анализ дерева событий (ETA) и модификации;

b) анализ видов и последствий отказов (FMEA) и модификации;

2) Нисходящие методы (исследующие последствия комбинаций неисправностей);

a) анализ дерева неисправностей (FTA);

b) Марковский анализ;

c) анализ сети Петри;

3) Исследование опасности и удобства использования (HAZOP);

4) Статистические методы оценки надежности.

Эти методы анализа применимы как для оценки характеристик качества, так и для оценок количественных характеристик при прогнозировании поведения системы в эксплуатации. Достоверность результата зависит от точности и правильности данных об основных событиях. На практике используют комбинации нисходящего и восходящего анализов, чтобы повысить полноту анализа.

В работе [4] выделены основные требования к моделям, используемым в методах расчета надежности:

1) Модель должна объяснять как уже произошедшие отказы, так и позволять прогнозировать отказы в будущем;
2) Модель должна основываться на существенных характеристиках моделируемой системы;
3) Модель должна основываться на понятных и достоверных предположениях;
4) Модель должна выражать в численной форме зависимости между отказами;
5) Модель должна основываться на простой и легко изучаемой концепции;
6) Исходные данные, требуемые для построения модели, должны приниматься достоверными значительной частью экспертного сообщества;
7) Модель должна различать одиночные и множественные отказы;
8) Модель должна различать отказы в выполнении функции и промежуточный отказ;
9) Модель должна позволять получить пользователю проверенные данные, включая вероятность отказа и оценку достоверности результата;
10) Модель должна позволять анализировать сценарии отказа цифровых компонентов во взаимодействии с нецифровыми компонентами;
11) Модель не должна использовать сиюминутную информацию о состоянии системы.

В таблице 1 приведено сравнение наиболее часто используемых в оценке показателей надежности методов с позиции их применения для программных компонентов. Данные для таблицы 1 в основном взяты из работы [4], в работе подчеркнут субъективный характер данных.

Можно видеть, что методы расчета надежности, приведенные в таблице 1, в целом имеют недостатки:

1. Неполнота компонентов и их отказов;
2. Отсутствие общепринятой философской основы программного моделирования интенсивности и вероятности отказов и методов для их количественной оценки;
3. Неубедительность оценки параметров отказа – интенсивности отказов, распределения режимов отказов и факторов отказа по общей причине (ООП).

Таблица 1. Сравнительная характеристика методов[1]

[1] В таблице приняты следующие обозначения: X – свойство покрыто, 0 – свойство не покрыто, ? – покрытие сомнительно.

Метод / Требование	1	2	3	4	5	6	7	8	9	10	11
Непрерывное дерево событий [6]	x	x	x	x	0	?	?	x	?	?	0
Динамическое дерево событий [7]	x	X	x	x	x	?	?	?	x	x	0
Марковские модели [2]	x	x	x	x	0	?	?	x	x	x	0
Сети Петри [8]	x	x	x	x	0	?	?	?	x	?	0
Методология динамических граф-потоков [9]	x	x	x	?	x	?	?	?	x	x	x
Динамическое дерево отказов [10]	x	?	?	?	x	?	x	?	x	?	x
Диаграмма последовательности событий [11]	x	x	x	x	0	?	?	?	x	x	0
Оценка по метрикам программного обеспечения [12]	x	?	0	0	?	?	x	x	0	0	x

Сомнительность применения методов расчета надежности для получения абсолютных значений показателей надежности не означает необходимости полного отказа от вероятностных методов их оценки. Методы могут быть использованы для анализа видов и последствий отказов отдельных компонентов АСУ ТП и для системы в целом, а также для анализа ее работоспособности. Марковские методы и сети Петри считаются [13] наиболее перспективными с точки зрения получения количественной оценки надежности программного обеспечения и учета взаимного влияния программных и аппаратных компонентов системы.

4. Оценки качества программного обеспечения: модели качества

Сложность процесса разработки и сопровождения ПО во многом обуславливается особыми требованиями, предъявляемыми к его качеству. Базовую модель качества можно определить как структурированный набор свойств, которые необходимы для удовлетворения определенных целей [14]. Преимущество применения базовой модели качества заключается в декомпозиции значимых для программного обеспечения объектов, таких, как процессы жизненного цикла, программный продукт, на ряд своих характеристик/подхарактеристик.

Пользователи ПО испытывают потребности в создании моделей качества ПО для оценки качества как качественно, так и количественно [15]. Модели качества, которые имеются в настоящее время, в большинстве случаев являются иерархическими моделями на основе критериев качества и связанных с ними показателей (метрик). Все модели качества могут быть разделены на три категории в соответствии с методами, на основе которых они были созданы. К первому виду можно отнести теоретические модели, основанные на гипотезе отношений между переменными качества. Ко второму виду относятся модели «управления данными», основанные на статистическом анализе. И, наконец, комбинированная модель, в которой интуиция исследователя используется для определения нужного вида модели, а анализ данных используется для определения констант модели качества. Но все эти модели связывают интересы пользователя, т.е. исходящие свойства системы, с внутренними свойствами, которые понятны разработчикам.

Качество ПО определяется в стандартах ISO/IEC 9126-1:2001 и ISO/IEC 25010:2011 как всякая совокупность его характеристик, относящихся к возможности удовлетворять высказанные или подразумеваемые потребности всех заинтересованных лиц.

Различаются понятия внутреннего качества, связанного с характеристиками ПО самого по себе, без учета его поведения; внешнего качества, характеризующего ПО с точки зрения его поведения; и качества ПО при использовании в различных контекстах – того качества, которое ощущается пользователями при конкретных сценариях работы ПО. Для всех этих аспектов качества введены метрики, позволяющие оценить их. Кроме того, для создания надежного ПО существенно качество технологических процессов его разработки. Взаимоотношения между этими аспектами качества по схеме, принятой в различных моделях качества, показано на рис. 1.

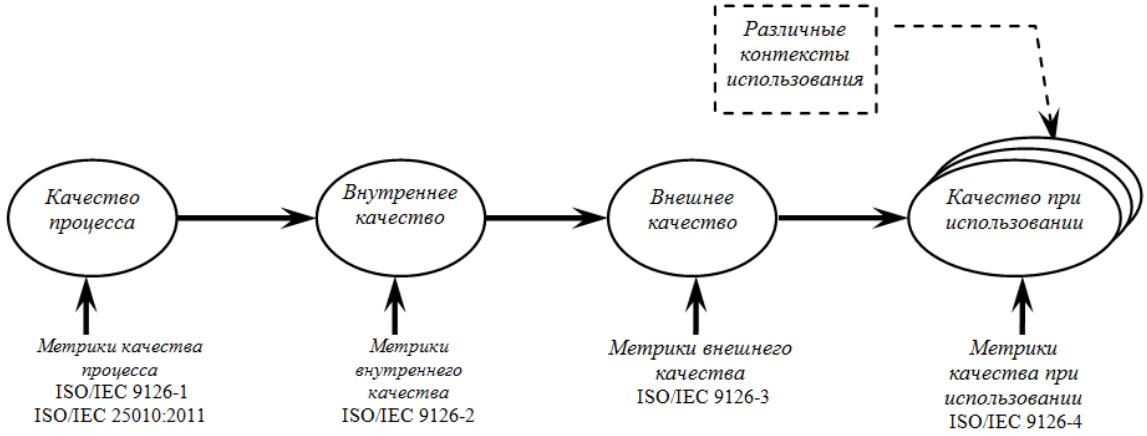

Рис. 1. Основные аспекты качества программного обеспечения по стандартам ISO/IEC 9126-1:2001 и ISO/IEC 25010:2011

На рис. 2 приведена модель оценивания качества ПО согласно ISO/IEC 9126.

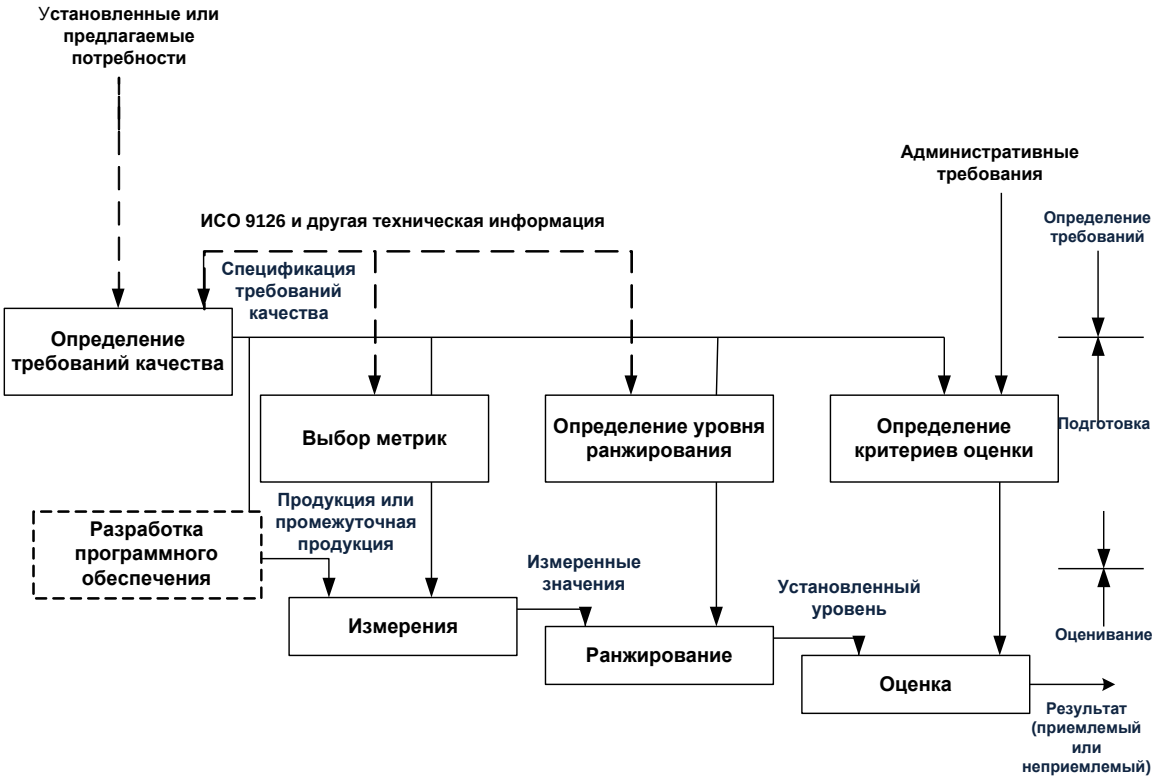

Рис. 2. Факторы и атрибуты внешнего и внутреннего качества программного обеспечения в соответствии с ISO/IEC 9126

5. Заключение

Проблема оценки надежности (безошибочности) программы, в отличие от проблемы создания качественной программы, видимо, не имеет решения в общем случае в рамках классической машины Тьюринга и существует некоторое количество фундаментальных проблем, связанных с детерминированным характером функционирования программы. Количественная оценка надежности систем, основанных на программных средствах, может быть получена только путем сочетания фактических данных из нескольких источников, однако и тогда будет существовать значительное недоверие к абсолютным цифрам для параметров надежности. В настоящее время не существует согласованных методов и данных об отказах для количественной оценки

надежности цифровых систем. Возможно, решение проблемы лежит в области перехода от классической универсальной машины Тьюринга к ее модификации в виде вероятностной машины Тьюринга или к функциональному программированию, которые свободны от вышеуказанных ограничений и допускают формальную верификацию программы. В первом случае вероятностная природа машины Тьюринга позволяет применять математический аппарат надежности, принятый в технике, для расчета надежности программного обеспечения. Во втором случае, возможность полной верификации кода позволяет решить проблему надежности признав программу абсолютно надежной в рамках заданного алгоритма. Данные подходы ждут своего обоснования.

Список литературы

1. Бывайков М.Е., Жарко Е.Ф., Менгазетдинов Н.Э., Полетыкин А.Г., Прангишвили И.В., Промыслов В.Г. Опыт проектирования и внедрения системы верхнего блочного уровня АСУ ТП АЭС // Автоматика и телемеханика. 2006. № 5. С. 65-79.
2. Smith D., DeLong T., Johnson B.W. A Safety Assessment Methodology for Complex Safety-Critical Hardware/Software Systems // International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies. Washington, DC, November, 2000.
3. Липаев В.В. Надежность программных средств. М.: СИНТЕГ, 1998.
4. Aldernir T., Miller D.W., Stovsky M.P., Kirschenbaurr J., Bucci P., Fentiman A.W., Mangan L.T. Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments (NUREG/CR-6901).
5. Halstead M.H. Elements of Software Science. New York: Elsevier, 1977.
6. Devooght J., Smidts C. Probabilistic Reactor Dynamics l: The theory of continuous event trees // Nuclear Science and Engineering. 1992. Vol. 111. No. 3. P. 229-240.
7. Acosta C., Siu N. Dynamic event trees in accident sequence analysis: Application to steam generator tube rupture // Reliab. Engng & System Safety. 1993. Vol. 41, No. 2. P. 135-154.
8. Goddard P.L. A Combined Analysis Approach to Assessing Requirements for Safety Critical Real-Time Control Systems // Reliability and Maintainability Symposium, 1996 Proceedings. International Symposium on Product Quality and Integrity., Annual. P. 110-115.
9. Stamataletos M. et.al. Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners, Version 1.1, August, 2002.
10. Andrews J.D., Dugan J.B. Dependency modeling using fault-tree analysis // Proceedings of the 17 International System Safety Conference, The System Safety Society, Unionville, Virginia, 1999. P. 67-76.
11. Matsuoka T., Kobayashi M. An analysis of a dynamic system by the GOFLOW methodology // Proc. ESREL'96/PSAM III, Crete, 1996. P. 1547-1552.
12. Smidts C., Li M. Validation of a Methodology for Assessing Software Quality. Report UMDRE 2002-07. February, 2002.
13. NEA/CSNI Recommendations on assessing digital system reliability in probabilistic risk assessment of nuclear power plants. 2009. 157 p.
14. Fitzpatrick R. Software Quality: Definitions and Strategic Issues. Staffordshire University, School of Computing Report. 1996. 35 p.
15. Жарко Е.Ф. Сравнение моделей качества программного обеспечения: аналитический подход // XII Всероссийское совещание по проблемам управления. ВСПУ-2014. Москва, 16-19 июня 2014 г.: Труды. М.: ИПУ РАН, 2014. С. 4585-4594.

Reference

1. Byvaikov ME, Zharko EF, Mengazetdinov NE, Poletkin AG, Prangishvili IV, Promyslov VG Experience in designing and implementing the system of the upper block level of the automated process control system of the nuclear power plant // Automation and telemechanics. 2006. № 5. P. 65-79.
2. Smith D., DeLong T., Johnson B.W. A Safety Assessment Methodology for Complex Safety-Critical Hardware / Software Systems // International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies. Washington, DC, November, 2000.
3. Lipaev V.V. Reliability of software. M .: SYNTHEG, 1998.

4. Aldernir T., Miller D.W., Stovsky M. P., Kirschenbaurr J., Bucci P., Fentiman A.W., Mangan L.T. Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments (NUREG / CR-6901).

5. Halstead M.H. Elements of Software Science. New York: Elsevier, 1977.

6. Devooght J., Smidts C. Probabilistic Reactor Dynamics l: The theory of continuous event trees // Nuclear Science and Engineering. 1992. Vol. 111. No. 3. P. 229-240.

7. Acosta C., Siu N. Dynamic event trees in accident sequence analysis: Application to steam generator tube rupture // Reliab. Engng & System Safety. 1993. Vol. 41, No. 2. P. 135-154.

8. Goddard P.L. A Combined Analysis Approach to Assessing Requirements for Safety Critical Real-Time Control Systems // Reliability and Maintainability Symposium, 1996 Proceedings. International Symposium on Product Quality and Integrity., Annual. P. 110-115.

9. Stamataletos M. et.al. Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners, Version 1.1, August, 2002.

10. Andrews J.D., Dugan J.B. Dependency modeling using fault-tree analysis // Proceedings of the 17 International System Safety Conference, The System Safety Society, Unionville, Virginia, 1999. P. 67-76.

11. Matsuoka T., Kobayashi M. An analysis of a dynamic system by the GOFLOW methodology // Proc. ESREL'96 / PSAM III, Crete, 1996. P. 1547-1552.

12. Smidts C., Li M. Validation of a Methodology for Assessing Software Quality. Report UMDRE 2002-07. February, 2002.

13. NEA / CSNI Recommendations on assessing the digital system reliability in a probabilistic risk assessment of nuclear power plants. 2009. 157 p.

14. Fitzpatrick R. Software Quality: Definitions and Strategic Issues. Staffordshire University, School of Computing Report. 1996. 35 p.

15. Zharko EF. Comparison of software quality models: an analytical approach // XII All-Russian meeting on governance issues. VSPU-2014. Moscow, June 16-19, 2014: Proceedings. Moscow: IPP RAS, 2014. S. 4585-4594.