DOI: https://doi.org/10.32836/2521-6643-2020.2-60.5
УДК 004.054:052.42

О. А. Руденко, кандидат технічних наук, доцент кафедри комп'ютерних та інформаційних технологій і систем Національного університету «Полтавська політехніка імені Юрія Кондратюка»
А. В. Головко, інженер-програміст Науково-виробничого підприємства «Радікс», студент Національного університету «Полтавська політехніка імені Юрія Кондратюка»
Ю. Л. Поночовний, кандидат технічних наук, старший науковий співробітник, доцент кафедри інформаційних систем та технологій Полтавської державної аграрної академії

УДОСКОНАЛЕННЯ СТЕНДУ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ФУНКЦІЙ FPGA КОМПОНЕНТІВ АПАРАТНО-ПРОГРАМНИХ КОМПЛЕКСІВ ПЛАТФОРМИ RADICS

У статті показана роль стадії тестування і відлагодження у життєвому циклі програмних продуктів при оцінці їх надійності. Виділені етапи тестування апаратно-програмних комплексів. Проаналізовані дослідження, спрямовані на оптимізацію стадії тестування і відлагодження програмних продуктів. Розглянуто характеристики факторів, врахування яких при тестуванні підвищує точність оцінки надійності програмних продуктів. Визначені особливості моделювання, спрямовані на створення засобів тестування програмних продуктів. Виконано опис архітектури платформи RadICS. Продемонстровано приклад розробки алгоритму для перевірки якості та надійності функцій блоків бібліотеки AFBL. Проаналізовано конфігурацію випробувального стенду з ручним тестуванням та запропоновано його модифікацію на основі обладнання та програмного забезпечення компанії National Instruments.

Ключові слова: *якість та надійність програмного забезпечення, валідація, Application Logic, автоматизація процесу тестування, NI Test Stand, життєвий цикл програмних продуктів, FPGA, Radiy Platform Configuration Tool, AFBL.*

В статье показана роль стадии тестирования и отладки в жизненном цикле программных продуктов при оценке их надежности. Выделены этапы тестирования аппаратно-программных комплексов. Проанализированы исследования, направленные на оптимизацию стадии тестирования и отладки программных продуктов. Рассмотрены характеристики факторов, учет которых при тестировании повышает точность оценки надежности программных продуктов. Определены особенности моделирования, направленные на создание средств тестирования программных продуктов. Выполнено описание архитектуры платформы RadICS. Продемонстрирован пример разработки алгоритма для проверки качества и надежности функций блоков библиотеки AFBL. Проанализированы конфигурации испытательного стенда с ручным тестированием и предложена его модификация на основе оборудования и программного обеспечения компании National Instruments.

Ключевые слова: *качество и надежность программного обеспечения, валидация, Application Logic, автоматизация процесса тестирования, NI Test Stand, жизненный цикл программных продуктов, FPGA, Radiy Platform Configuration Tool, AFBL.*

The article shows the role of the testing and debugging stage in the life cycle of software products in assessing their reliability. Stages of testing of hardware and software complexes are allocated. The researches directed on optimization of a stage of testing and debugging of software products are analyzed; in which the characteristics of the method of accelerated testing of software reliability for graphical interfaces are considered and generalized; the division into stages of testing with comparison of results with model indicators is offered; the influence of test information on the assessment of software reliability is considered and the function of test coverage in the form of the generalized logistic function is offered. The characteristics of the factors, the consideration of which during testing increases the accuracy of assessing the reliability of software products, are considered. Features of modeling aimed at creating tools for testing software products are identified. Describes the architecture of the RadICS platform, which consists of a single instrument chassis containing a logic module, as well as up to fourteen other input/ output and fiber-optic modules. Application Logic for the RadICS platform are developed using a specialized Integrated Development Environment - Radiy Platform Configuration Tool. In this Integrated Development Environment, each Application Logic Block is represented graphically using visualization tools, which simplifies the management of functions (logical, mathematical, etc.). The Application Logic Block set in the Radiy Plat-

form Configuration Tool creates a function block library (AFBL) and consists of more than one hundred block elements. An example of algorithm development for checking the quality and reliability of AFBL library block functions is demonstrated. The configuration of the test bench with manual testing is analyzed and its modification on the basis of the equipment and software of the National Instruments company is offered.

Keywords: *software quality and reliability, validation, Application Logic, test process automation, NI Test Stand, software lifecycle, FPGA, Radiy Platform Configuration Tool, AFBL.*

Вступ. Бурхливий розвиток індустрії автоматизованих систем потребує нових підходів до створення технологій та вдосконалення наявних. При цьому зростає роль програмного забезпечення та вимоги до його надійності. Надійність є ключовою складовою якості, що регламентується рядом нормативних документів та міжнародних стандартів [1, 2].

На всіх етапах життєвого циклу програмних продуктів проводяться процедури, спрямовані на забезпечення їх надійності, що потребує матеріальних і часових затрат.

Ключовим етапом життєвого циклу програмних продуктів, в процесі забезпечення надійності, є стадія тестування і відлагодження. Стадія тестування має, в загальному випадку, свій «життєвий цикл», що містить ряд ітерацій:

– загальне планування і аналіз вимог;
– уточнення критеріїв приймання;
– уточнення стратегії тестування;
– розроблення тест-кейсів;
– виконання тест-кейсів;
– фіксація виявлених дефектів;
– аналіз результатів тестування;
– складання звітності.

Вимоги до проведення тестування та відлагодження програмних продуктів визначаються міжнародними стандартами [3-6], методи і принципи тестування описані в ряді навчальних посібників [7, 8].

Питання оптимізації стадії тестування і відлагодження програмних продуктів знайшли відображення у роботах [9-12], зокрема, в [9] розглянуті та узагальнені характеристики методу прискореного тестування надійності

програмного забезпечення для графічних інтерфейсів; в [10] запропоновано розбиття на етапи тестування з порівнянням результатів з модельними показниками; в [11] розглядається вплив тестової інформації на оцінку надійності програмного забезпечення; в [12] запропонована функція тестового покриття у вигляді узагальненої логістичної функції, що описує зміну тестового покриття під час тестової процедури.

Питання урахування різних факторів, необхідних при проведенні тестування розглядаються у роботах [13-18]. В [13] запропоновано включення S-подібних функцій тестування в модель надійності програмного забезпечення, що враховує недосконале відлагодження. В [14] розглянуте присвоєння імовірнісних пріоритетів механізму тестування, що здійснюється за допомогою робочого профілю програмного забезпечення. Це дослідження спрямоване на створення тестових кейсів та тестових наборів з точки зору ймовірнісної надійності, використовуючи запропоновану структуру, засновану на робочому профілі програмного забезпечення та профілі тестування. В [15-18] розглянуті різні підходи до оцінки надійності програмних засобів, що враховують фактор вторинних дефектів.

У дослідженнях [7, 9] показано, що скорочення тривалості етапів тестування можливе або шляхом створення спеціальних умов випробування, або через автоматизацію та прискорення виконання окремих фаз тестування, які, як правило, виконуються у ручному режимі.

Мета. Метою публікації є вдосконалення стенду тестування функцій FPGA компонентів апаратно-програмних комплексів платформи RadICS для підвищення ступеня автоматизації процесів тестування та скорочення їх тривалості.

1. Аналіз апаратних засобів платформи RadICS. Платформа RadICS являє собою модульну платформу і містить стандартизовані модулі, такі як логічний модуль, цифрові та аналогові модулі вводу/виводу (I/O), базуються на використанні мікросхем програмованої логіки (FPGA). Обладнання платформи RadICS встановлено більш ніж в 70 основних системах безпеки та управління на діючих атомних електростанціях (АЕС) різних країн світу.

Архітектура платформи RadICS складається з єдиного приладового шасі, що містить логічний модуль (LM), а також до 14 інших модулів вводу-виводу та оптоволоконного зв’язку. Ця одноканальна конфігурація є базовою конфігурацією інформаційно-керуючих систем (ІКС). Для систем ава-

рійного захисту АЕС така конфігурація має сертифікат рівня безпеки SIL 3 відповідно до вимог IEC 61508. Система на базі платформи RadICS може бути спроектована і виготовлена і в інших конфігураціях (наприклад, для забезпечення вимог класу 1E для систем аварійного захисту АЕС США та Канади). При цьому не тільки підвищується рівень вимог до неї, але і розширюється набір функцій, які платформа повинна забезпечити. Але після розробки нових або удосконалення старих функціональних компонент, необхідно знову виконувати їх тестування, тобто потрібен час та знання.

Раніше тестування займало близько трьох тижнів включаючи створення алгоритмів, налаштування системи, перевірки та документування звіту.

Враховуючи, що етап тестування функцій платформи RadICS був відпрацьований при сертифікації вимог IEC 61508, доцільно використати його напрацювання та розробки для подальшого вдосконалення тестового стенда і скорочення часу тестування без втрат якісних характеристик (повноти тестового покриття, тощо).

2. Аналіз програмних засобів та тестового стенда, що використовувалися на етапі інтеграційного тестування. Оскільки ІКС на базі платформи RadICS функціонує в автономному режимі, то інтеграційне тестування системної логіки додатків (Application Logic), виконується логічним модулем (LM) цієї платформи у процесі діагностування працездатності компонент ІКС. Додатки Application Logic для платформи RadICS розроблені з використанням спеціалізованого інтегрованого середовища розробки (Integrated Development Environment, IDE) – Radiy Platform Configuration Tool (RPCT). У даному IDE за допомогою засобів візуалізації кожний блок-додаток (Application Logic Block, ALB) представлений у графічному вигляді, що спрощує керування функціями (логічними, математичними, тощо). Кожен «графічний» AFB у середовищі RPCT пов’язаний з апаратно реалізованим у FPGA компонентом AFB в логічному модулі платформи RadICS.

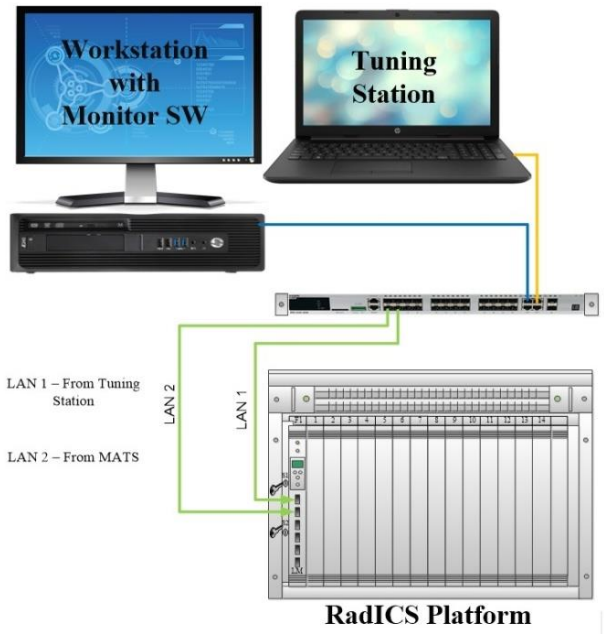

Рис. 1. Початкова конфігурація випробувального стенда для ручного тестування блоків AFBL

Множина AFB у середовищі RPCT формує бібліотеку функціональних блоків (AFBL) та складається з більш ніж 100 елементів-блоків. Ці функціональні блоки також можуть мати велику кількість параметрів вводу/виводу або навіть різні конфігурації.

Для проведення процедур інтеграційного тестування у ручному режимі використовувався спеціальний тестовий випробувальний стенд (Test Bad, TB), зображений на рис.1, що складався з:

– станції налаштувань (стаціонарний комп'ютер чи ноутбук з клієнтом налаштувань для генерації вхідних даних в AFB);

– робочої станції (робоча станція з програмою Monitor для збору вихідних даних);

– комутатора з портами для підключення оптоволоконних ліній;

– тестованої системи SUT (System Under Test) платформи RadICS (шасі платформи RadICS з LM).

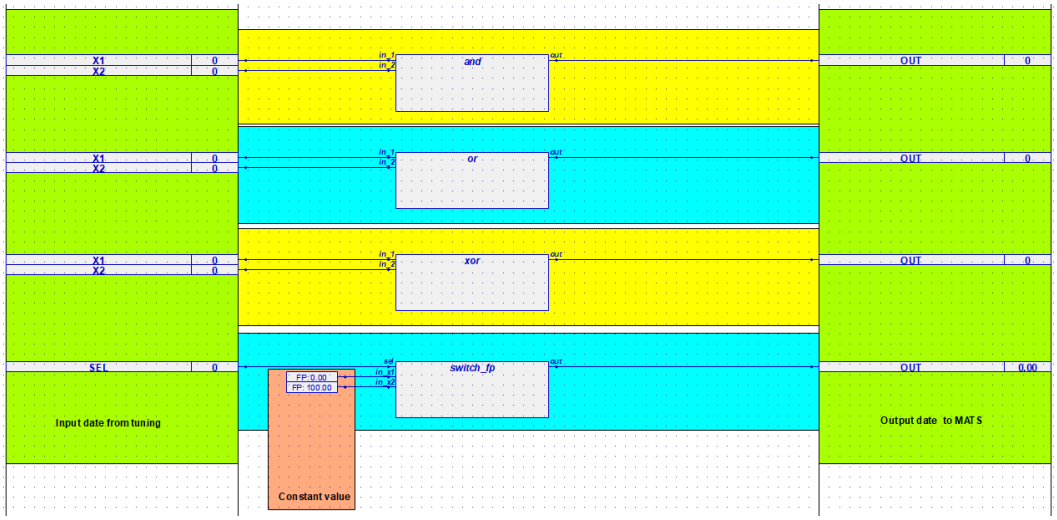

Рис. 2. Тестовий приклад для перевірки конфігурації блоків (AND, OR, XOR, SWITCH_FP) з бібліотеки AFBL

Для забезпечення необхідного тестового покриття під час інтеграційного тестування за допомогою RPCT для окремої конфігурації, яка може включати один, або декілька AFB розробляється тестовий приклад (рис. 2), який вручну прошивається до модуля LM через оптичний порт LAN1. Результати тестування через другий оптичний порт LAN2 модуля LM зчитуються робочою станцією з ПЗ Monitor.

Таким чином, інтеграційне тестування всієї бібліотеки AFBL виконувалась вручну двома інженерами-випробувачами (одна людина для тестування, а інша для документування результатів тестування) і займало два-три тижні.

3. Конфігурація модифікованого тестового випробувального стенда для автоматизованого інтеграційного тестування. Оскільки у багатьох проектах RadICS також використовуються апаратні стенди та програмні засоби компанії National Instruments (NI), і це обладнання показало високу надійність і точність вимірювань, було прийнято рішення використати його для автоматизованого випробувального стенда. Для процедур тестування у National Instruments розроблені програмні пакети LabVIEW та Test Stand. ПЗ Test Stand є спеціалізованим саме для автоматизації тестових процедур. Але його використання вимагає модифікації випробувального стенда, як показано на рис. 3.

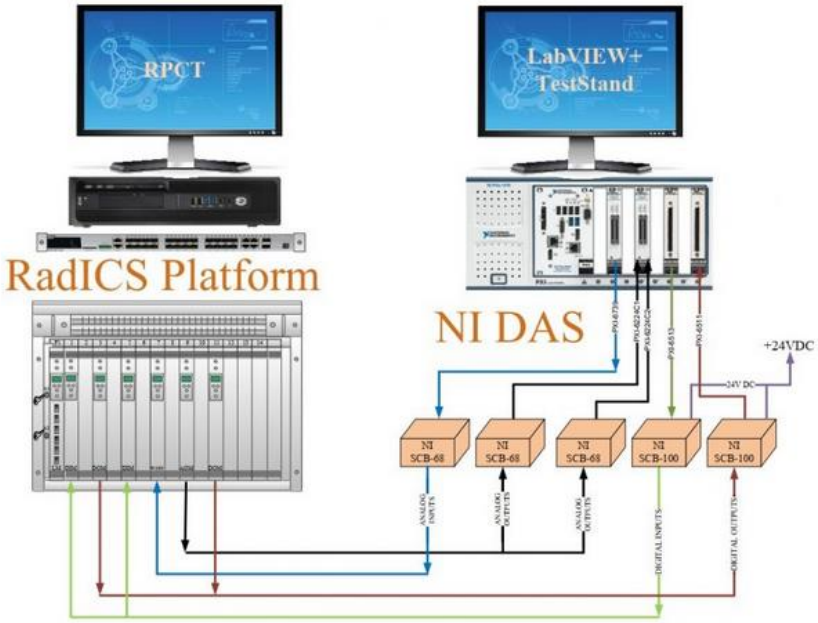

Рис. 3. Модифікований випробувальний стенд для автоматизованого тестування блоків AFBL

До конфігурації модифікованого тестового стенда входить SUT (платформа RadICS з комутатором оптоволоконних ліній та комп'ютер з IDE RPCT) і ТВ (представлений системою збору даних NI Data Analysis System, DAS). Використання DAS дозволяє зчитувати реальні сигнали з виходів модулів вводу/виводу платформи RadICS і цим самим відмовитися від моніторингу віртуальних сигналів через ПЗ Monitor у ручному режимі. При цьому кількість модулів у шасі обладнання NI і RadICS невелика (тільки по одному модулю кожного типу). Така мінімізація досягнута шляхом використання незалежних конфігурацій програмного забезпечення для кожного тестового прикладу. Це означає, що LM може зберігати у своїй пам'яті до 12 конфігурацій шасі з різним набором апаратного забезпечення. З іншого боку, ПЗ Test Stand дозволяє розробнику тестових прикладів форматувати HW-модулі та канали по різному перед кожним тестом.

Використання ПЗ NI Test Stand дозволило розробнику створювати незалежні тестові приклади для кожного блоку з бібліотеки AFBL. Очікуваний результат і вхідні дані для кожного тесту беруться з зовнішнього файлу да-

них CSV і можуть бути змінені інженером-тестувальником, якому не потрібно мати практичну підготовку для роботи у складному середовищі LabVIEW (рис.4). Виконання тестового прикладу повністю автоматичне і послідовне.

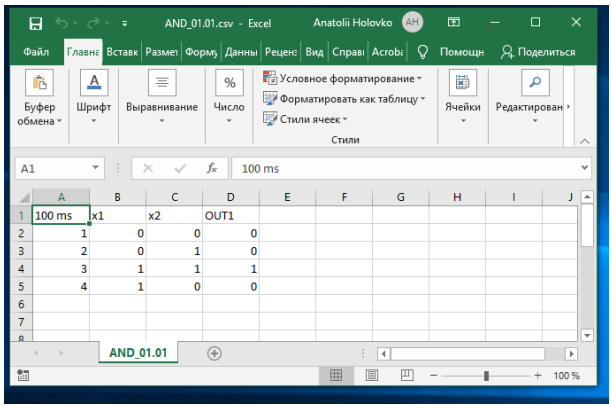

Рис. 4. CSV файл даних

Крім того, ПЗ TestStand автоматично складає зведений звіт з результатами всіх тестів. Звіт може бути представлений в різних форматах даних, таких як HTML, XML.

Як показали результати інтеграційного тестування платформи RadICS, автоматизований процес за допомогою NI Test Stand відповідає всім вимогам нормативної та керівної документації.

Висновки та перспективи подальших досліджень у даному напрямі. У роботі запропоновано модифікацію тестового випробувального стенда для автоматизованого тестування блоків AFBL платформи RadICS шляхом використання обладнання DAS та програмного забезпечення Test Stand компанії National Instruments.

Автоматизація тестування дозволила отримати наступні переваги (у порівнянні з попередньою версією тестового стенда з ручним режимом):

– істотно скорочено час тестування (загальний час тестування менш як 10 хвилин у порівнянні з 2-3 тижнями);

– вхідні і вихідні сигнали пов’язані з фізичними каналами модулів вводу/виводу, а не їх віртуальним поданням у логічному модулі LM;

– зменшено негативний вплив фактору людських помилок на резуль-

тати проведення тестів;

– зменшено вимоги до кваліфікації інженера-тестувальника, зокрема до практичних навичок роботи у середовищі LabVIEW.

Напрямками подальших досліджень є:

– розширення множини тестових прикладів шляхом їх накопичення при тестуванні різних конфігурацій платформи RadICS;

– накопичення та обробка результатів тестування для визначення адекватних моделей оцінювання надійності та функційної безпечності ПЗ зокрема та ІКС на базі платформи RadICS в цілому.

Список використаних джерел:

1. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. URL: https://www.iso.org/standard/35733.html

2. IEC 60050-192:2015. International Electrotechnical Vocabulary (IEV) - Part 192: Dependability. URL: https://webstore.iec.ch/publication/21886

3. IEEE 829-2008 - IEEE Standard for Software and System Test Documentation. URL: https://standards.ieee.org/standard/829-2008.html

4. IEEE Standard for System and Software Verification and Validation, in IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004) , vol., no., pp.1-223, 25 May 2012, doi: 10.1109/IEEESTD.2012.6204026.

5. IEEE 1016-2009 - IEEE Standard for Information Technology--Systems Design--Software Design Descriptions. URL: https://standards.ieee.org/standard/1016-2009.html

6. IEC 61508-3:2010. Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 3: Software requirements. URL: https://webstore.iec.ch/publication/5517

7. Благодатских В.А. Волнин В.А., Поскакалов К.Ф. Стандартизация разработки программных средств: Учеб. пособие. Под ред. О.С. Разумова. М.: Финансы и статистика, 2005. 288 с.

8. Аврутов В.В., Бурау Н.И. Надежность и диагностика приборов и систем: Учебное пособие. К.: НТУУ «КПИ», 2014. 156 с.

9. Liu, Q., Wu, Y., Lu , M. (2015). Study of GUI oriented software reliability accelerated testing method. "Proceedings of the First International Conference on Reliability Systems Engineering (ICRSE)", p.p. 1-6, DOI:

10.1109/ICRSE.2015.7366479.

10. Honda, K., Washizaki, H., Fukazawa, Y., Munakata, K., Morita, S., Uehara, T., Yamamoto, R. (2015). Detection of unexpected situations by applying software reliability growth models to test phases. "Proceedings of the 2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)". № 18, p.p. 2-5, DOI: 10.1109/ISSREW.2015.7392024.

11. Okamura, H., Takekoshi, Y., Dohi, T. (2015). Fine-Grained Software Reliability Estimation Using Software Testing Inputs. "Proceedings of the 2015 IEEE International Conference on Software Quality Reliability and Security (QRS)". № 20, p.p. 85-92, DOI: 10.1109/QRS.2015.22.

12. Zhou, B., Lei, H., Guo, W. (2015). Software reliability modeling with the generalized logistic test coverage function. "Proceedings of the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)". № 23, p.p. 106-109, DOI: 10.1109/ICSESS.2015.7339015.

13. Li, Q., Li, H., Minyan, L. (2015). Incorporating S-shaped testing-effort functions into NHPP software reliability model with imperfect debugging. "Journal of Systems Engineering and Electronics". Vol. 26, № 1, p.p. 190-207, DOI: 10.1109/JSEE.2015.00024.

14. Ali-Shahid, M.M., Sulaiman, S. (2015). Improving reliability using software operational profile and testing profile. "Proceedings of the 2015 International Conference on Computer Communications, and Control Technology (I4CT)". № 18, p.p. 384-388, DOI: 10.1109/I4CT.2015.7219603.

15. Маевский Д. А., Жеков О.П. Использование теории временных рядов для выделения вторичных ошибок на этапе тестирования программного обеспечения. Радіоелектронні і комп'ютерні системи. 2011. № 2 (16). С. 82-85.

16. Одарущенко О.Н., Руденко А.А., Харченко В.С. Метод оценивания надежности программных средств с учетом вторичных дефектов. Радіоелектронні і комп'ютерні системи. 2012. № 7 (59). С. 294-300.

17. Rudenko O., Odarushchenko E., Rudenko Z., Rudenko M., "The Secondary Software Faults Number Evaluation Based on Correction of the Experimental Data Exponential Line Approximation", Conference Proceedings of 2018 IEEE 9thInternational Conference on Dependable Systems, Services and Technologies DESSERT'2018, Kyiv, 2018, pp. 401-405.

18. Руденко О. А. Оцінювання кількості вторинних дефектів програм-

них засобів шляхом комплексування модифікованих моделей росту надійності Джелінські-Моранди і Шика-Волвертона / О. А. Руденко, О. М. Одарущенко, З. М. Руденко, О. Б. Одарущенко // Системи управління, навігації та зв'язку. 2020. Вип. 1. С. 97-100.

References:

1. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. URL: https://www.iso.org/standard/35733.html

2. IEC 60050-192:2015. International Electrotechnical Vocabulary (IEV) - Part 192: Dependability. URL: https://webstore.iec.ch/publication/21886

3. IEEE 829-2008 - IEEE Standard for Software and System Test Documentation. URL: https://standards.ieee.org/standard/829-2008.html

4. IEEE Standard for System and Software Verification and Validation, in IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004) , vol., no., pp.1-223, 25 May 2012, doi: 10.1109/IEEESTD.2012.6204026.

5. IEEE 1016-2009 - IEEE Standard for Information Technology--Systems Design--Software Design Descriptions. URL: https://standards.ieee.org/standard/1016-2009.html

6. IEC 61508-3:2010. Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 3: Software requirements. URL: https://webstore.iec.ch/publication/5517

7. Blagodatskikh V.A., Volnin V.A., Poskakalov K.F. Standardization of software development. Ed. O.S. Razumova. Moscow: Finance and Statistics, 2005. 288 p.

8. Avrutov V.V., Burau N.I. Reliability and diagnostics of devices and systems. K .: NTUU "KPI", 2014. 156 p.

9. Liu, Q., Wu, Y., Lu , M. (2015). Study of GUI oriented software reliability accelerated testing method. "Proceedings of the First International Conference on Reliability Systems Engineering (ICRSE)", p.p. 1-6, DOI: 10.1109/ICRSE.2015.7366479.

10. Honda, K., Washizaki, H., Fukazawa, Y., Munakata, K., Morita, S., Uehara, T., Yamamoto, R. (2015). Detection of unexpected situations by applying software reliability growth models to test phases. "Proceedings of the 2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)". № 18, p.p. 2-5, DOI: 10.1109/ISSREW.2015.7392024.

11. Okamura, H., Takekoshi, Y., Dohi, T. (2015). Fine-Grained Software Reliability Estimation Using Software Testing Inputs. "Proceedings of the 2015

IEEE International Conference on Software Quality Reliability and Security (QRS)". № 20, p.p. 85-92, DOI: 10.1109/QRS.2015.22.

12. Zhou, B., Lei, H., Guo, W. (2015). Software reliability modeling with the generalized logistic test coverage function. "Proceedings of the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)". № 23, p.p. 106-109, DOI: 10.1109/ICSESS.2015.7339015.

13. Li, Q., Li, H., Minyan, L. (2015). Incorporating S-shaped testing-effort functions into NHPP software reliability model with imperfect debugging. "Journal of Systems Engineering and Electronics". Vol. 26, № 1, p.p. 190-207, DOI: 10.1109/JSEE.2015.00024.

14. Ali-Shahid, M.M., Sulaiman, S. (2015). Improving reliability using software operational profile and testing profile. "Proceedings of the 2015 International Conference on Computer Communications, and Control Technology (I4CT)". № 18, p.p. 384-388, DOI: 10.1109/I4CT.2015.7219603.

15. Maevsky D.A., Zhekov O.P. Using time series theory to isolate secondary errors at the software testing stage. Radioelectronic and computer systems. 2011. No. 2 (16). P. 82-85.

16. Odarushchenko O.N., Rudenko A.A., Kharchenko V.S. A method for evaluating the reliability of software tools taking into account secondary defects. Radioelectronic and computer systems. 2012. No. 7 (59). P. 294-300.

17. Rudenko O., Odarushchenko E., Rudenko Z., Rudenko M., “The Secondary Software Faults Number Evaluation Based on Correction of the Experimental Data Exponential Line Approximation“, Conference Proceedings of 2018 IEEE 9thInternational Conference on Dependable Systems, Services and Technologies DESSERT’2018, Kyiv, 2018, pp. 401-405.

18. Rudenko O.A., Odarushchenko O.M., Rudenko Z.M., Odarushchenko O.B. Estimation of the number of secondary defects of software by complexing modified models of growth of reliability of Dzhelinski-Moranda and Shika-Wolverton. Control, navigation and communication systems. 2020. vol. 1. P. 97-100.