УДК 004.942: 052:056

https://orcid.org/0000-0002-8698-0972
https://orcid.org/0000-0002-7860-9663
https://orcid.org/0000-0002-3014-5521

В.Г. ІВАНОВ, В.О. ЛИФАР, О.К. ЛИФАР

ТЕОРЕТИКО-МЕТОДИЧНІ АСПЕКТИ КОНЦЕПЦІЇ ЗАБЕЗПЕЧЕННЯ НЕОБХІДНОГО РІВНЯ ПОВНОТИ БЕЗПЕКИ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ ОБ'ЄКТАМИ ПІДВИЩЕНОЇ НЕБЕЗПЕКИ

Анотація. *Представлені аспекти сучасних підходів до вирішення науково-технічної проблеми щодо забезпечення необхідного рівня повноти безпеки технічних засобів АСУ ТП об'єктами підвищеної небезпеки. Сформульовано завдання досліджень і теоретико-методична концепція визначення показників надійності і безпеки апаратних і програмних засобів АСУТП. Розглянуто існуючі та запропоновано оригінальні методи визначення нормуючих показників надійності при проведенні SIL-аналізу. Розглянуто проблеми підготовки фахівців до забезпечення необхідного рівня SIL при розробці АСУТП.*

Ключові слова: *рівень повноти безпеки, функціональна безпека, електричні/електронні/програмовані електронні пристрої, надійність, безпека, інженерія програмного забезпечення, комп'ютерна інженерія, комп'ютерні науки, моделі процесорів, інформаційні технології.*

DOI: 10.35350/2409-8876-2019-16-3-36-48

Вступ

Оцінка і профілактика техногенних ризиків, попередження великих аварій при експлуатації промислових об'єктів підвищеної небезпеки вимагають зваженого підходу, нормуються національним і міжнародним законодавством і є актуальною науково-технічною проблемою. Рішення такої проблеми у вигляді теоретично обґрунтованих методів і моделей оцінки рівня надійності та класифікації небезпеки наслідків відмов для автоматизованих систем управління технологічними процесами (АСУТП), а також реалізації комплексної інформаційної технології підтримки прийняття рішень по забезпеченню інтегрального рівня безпеки АСУТП становить значний інтерес для розробників і користувачів таких систем для великих промислових підприємств. До них відносяться: хімічні і нафтохімічні заводи, засоби транспортування небезпечних вантажів і речовин.

На жаль, цій проблемі приділяється мало уваги в науково-технічній сфері, так як аналіз великих промислових аварій і катастроф найчастіше завершується висновком причини технологічних відмов, зовнішніх впливів або людських факторів. Іноді причинами таких небезпечних аварій є не діагностована відмова АСУТП. Однак, це слабо доказові припущення, особливо в умовах зростаючої складності і інтеграції технологічних процесів.

© В.Г. Іванов, В.О. Лифар, О.К. Лифар, 2019

Дослідники і вчені, які займалися питаннями SIL: Michael A. Mitchell, Кулямін В.В., Glisente Landrini, Ковальов І.В., Буй Д.Б., Скобелєв В.Г., Гайдамакін Н.А., Wong Y.K., Eriksson J., Grindal M. та інші.

Незважаючи на безліч стандартних методів і підходів щодо забезпечення необхідного рівня повноти безпеки (РПБ) для АСУТП, існує велика кількість протиріч, неточностей, проблем формалізації і невирішених питань оцінки ризику відмов елементів управління технологічними процесами, що представляє значні складності при прийнятті рішень і сертифікації управляючих комплексів на відповідність інтегральному рівню безпеки (Safety integrity level – SIL) для систем з використанням електричних, електронних, програмованих електронних пристроїв (Е/Е/ПЕ – Е/Е/РЕ). Особливі труднощі відчувають розробники базових електронних, електричних, електронних програмованих пристроїв і програмного забезпечення АСУТП. Так як для центральної електронної частини управління спочатку невідомі функціональні призначення вхідних сигналів, їх інформаційна значимість і відповідність певним видам негативних наслідків у разі їх спотворення, істотно ускладнюється інтерпретація небезпеки таких відмов. Це саме можна сказати і до обробки інформації процесорами і вироблення керуючих сигналів. Завдання спрощується в разі, коли АСУТП представлена повним замкнутим контуром від датчиків і вимірювальних пристроїв, структурою прийому, обробки сигналів і видачі керуючих сигналів аж до виконавчих пристроїв і механізмів. В цьому випадку можлива більш-менш певна інтерпретація і аналіз небезпеки наслідків відмов системи або спотворення інформації в ній.

Проте, проблема проведення SIL аналізу і оцінка рівня повноти безпеки для центральних частин розроблюваних АСУТП (див. на рис. 1) є актуальною для встановлення верхньої межі РПБ, а також прийняття рішень щодо технологій розробки таких систем.

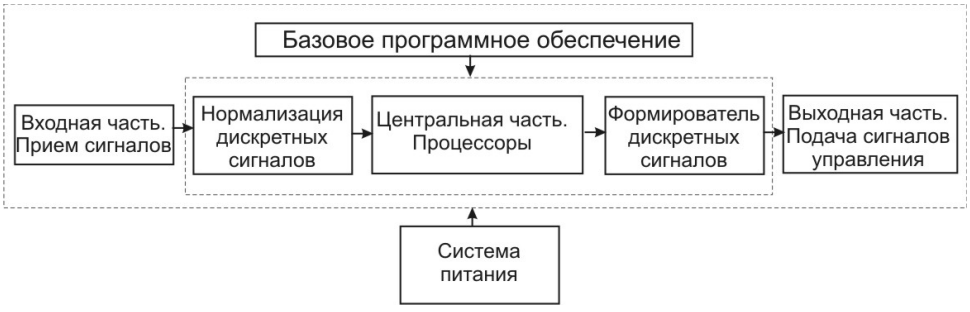

Рисунок 1 – Типова структура базової частини АСУТП

Оцінка рівня повноти безпеки для апаратних частин АСУТП регламентована в повному обсязі [1-4] і передбачає проведення аналізу причин і наслідків відмов (Failure modes and effects analysis – FMEA), а також їх критичності. За результатами аналізу визначаються види наслідків відмов елементів розглянутих блоків АСУТП або її центральної частини, і далі, з використанням методів оцінки ризику, обчислюються кількісні показники ймовірностей відмов.

Найбільш складною частиною вирішення проблеми оцінки РПБ для розроблюваних комплексів є визначення надійності і безпеки програмного забезпечення. Відомі і регламентовані методи [1-3] засновані в основному на рангових оцінках, що у великій мірі нівелює вірогідність визначення рівня надійності програмного забезпечення і представляє велику трудність для розробників базового програмного забезпечення (ПО) центральних частин комплексів АСУТП.

Необхідність розробки методів оцінки кількісних показників надійності програмних засобів, що забезпечують функціональну, експлуатаційну та технічну безпеку роботи АСУТП, обумовлена практичною відсутністю таких методів і проблемою узгодження якісних і кількісних критеріїв, що характеризують рівень повноти безпеки розроблюваних комплексів.

1. Аналіз даних і постановка задачі досліджень

Рівень повноти безпеки відображає ступінь ризику експлуатації об'єктів критичної області експлуатації. У цьому сенсі під «ризиком» мається на увазі настання певних наслідків з певною ймовірністю (або частотою для заданого періоду експлуатації). Проблема профілактики техногенного ризику об'єктів підвищеної небезпеки, обумовленого відмовами АСУТП, може бути розв'язана в результаті послідовного вирішення декількох задач:

1) аналіз виникнення і розвитку процесів відмов елементів АСУТП і оцінка ймовірності таких подій;

2) аналіз наслідків розглянутих відмов і віднесення їх до певної категорії (небезпечні; безпечні; діагностуються; що не діагностуються; критичні; що не критичні; які не впливають на безпеку) на підставі оцінки масштабів таких наслідків;

3) оцінка надійності програмного забезпечення АСУТП (показників ймовірності відмов: середня ймовірність відмови на запит виконання функції безпеки за час T1 (Probability of Failure on Demand) – PFDavg (T1); середня частота небезпечних відмов у годину (Hazardous Failure Probability) – PFH;

4) розробка вимог до діагностики і методів верифікації ПЗ для всіх стадій життєвого циклу;

5) аналіз отриманих показників надійності програмної і апаратної частини АСУТП і вироблення рішень (рекомендацій) за технологією розробки АСУТП на основі порівняльного аналізу нормативних і поточних показників надійності.

Розглядається дві ситуації, для яких може проводитися аналіз і визначення РПБ з метою сертифікації системи управління:

1. При розробці базового комплексу АСУТП без конкретної прив'язки до об'єкта управління. При цьому необхідно визначити нижню межу РПБ, яка забезпечує інтегральний рівень безпеки не гірше заявленого рівня.

2. При створенні АСУТП з повною прив'язкою до об'єкта управління і оцінкою показників ризику, обумовленого експлуатаційною безпекою АСУТП. При цьому функціональна і технічна безпека програмного забезпечення відноситься до внутрішньої складової експлуатаційної безпеки ПЗ.

Вхідними даними для визначення показників надійності апаратної частини АСУТП є показники надійності (напрацювання на відмову, паспортні дані про ймовірність відмов на запит і на період експлуатації та ін.) окремих елементів Е/Е/ПЕ пристроїв. У першому випадку до таких елементів відносяться тільки фізичні елементи базового комплекту АСУТП (без вимірювальних і виконавчих пристроїв). У другому випадку аналізуються всі Е/Е/ПЕ елементи, включаючи датчики, що передають, і виконавчі пристрої. Завдання оцінки SIL для цих двох ситуацій також відрізняються тим, що в першому випадку апріорі встановлюється поняття «безпечного стану» як відмова системи або припинення її працездатності за умови повного і однозначного діагностування такого стану і нормально безаварійного відключення АСУТП. При цьому такі відмови або зупинки вважаються безпечними. Всі відмови елементів АСУТП, що призводять до спотворення або припинення виконання закладених функцій системи управління, вважаються апріорі небезпечними. У другому випадку рівень небезпеки відмови елемента системи управління встановлюється на підставі аналізу наслідків такої відмови для функціонування технологічних елементів об'єкта управління.

Теорія надійності програмного забезпечення отримала розвиток одночасно з розповсюдженням програмного забезпечення в керуючих комплексах [9, 11]. Для отримання кількісних показників надійності програмного забезпечення необхідно отримати велику кількість статистичного матеріалу, що містить інформацію про динаміку виявлення помилок в закінчених програмних утвореннях. Особливі труднощі представляє взаємодія цих утворень, так як область визначення повного функціоналу програмного комплексу навіть середньої складності найчастіше не може бути визначена повною мірою.

2. Об'єкт, мета та завдання розробок

Об'єкт дослідження – інформаційна підтримка процесів прийняття рішень при створенні керуючих комплексів АСУТП, спрямованих на досягнення необхідного рівня повноти безпеки.

Дана розробка проводиться з метою створення методів, моделей (їх композицій) і програмних засобів інформаційних технологій, які могли б забезпечити процес підтримки рішень при розробці автоматизованих систем управління об'єктами підвищеної небезпеки. Вивчення і розв'язання проблеми забезпечення необхідного рівня повноти безпеки завдань можливо на основі застосування методів оцінки надійності та ймовірності відмов окремих апаратних і програмних складових АСУТП з урахуванням їх взаємного впливу в інтегральному ризику.

Предметом дослідження є моделі, методи, інформаційна технологія оцінки і порівняльного аналізу рівня повноти безпеки на всіх стадіях життєвого циклу створення, експлуатації та ліквідації керуючих комплексів.

3. Існуючі методи і підходи до оцінки рівня повноти безпеки

Оцінка рівня повноти безпеки для апаратних частин АСУТП досить повно регламентована міжнародними стандартами [1-6] і рядом інших керівних документів, в тому числі створених провідними компаніями в області розробки керуючих комплексів. Наприклад, Глізенте Ландрін [12] представляє ряд статей, в яких детально і дохідливо пояснює підходи і методи визначення кількісних показників, якісних характеристик і критерії вибору компонент для застосування в розподілених системах управління і спеціальних системах забезпечення безпеки з різними рівнями SIL, які рекомендовані в стандартах МЕК 61508 та 61511. Розглядаються також практичні приклади використання таких критеріїв. У статтях М. А. Мітчела [13] зроблено спробу роз'яснити узагальнені підходи до визначення SIL і застосування методів, описаних в стандартах до конкретних систем безпеки. В основі таких підходів лежать методи диференційованого аналізу причин і наслідків відмов FMEA (Failure modes and effects analysis) або з урахуванням їх критичності (FMECA). При цьому враховується принцип ALARP (As Low as Reasonable Practible - низький, наскільки це можливо) для зниження ризику реалізації небезпек, що викликаються відмовами до прийнятної величини.

Найбільшу трудність і невизначеність при такому підході викликають: визначення та формалізація функцій безпеки і встановлення однозначних зв'язків між значущими видами відмов елементів керуючої системи і впливом таких відмов на масштаби небезпечних наслідків.

На рис. 2 представлені відношення різних етапів створення системи функціональної безпеки заданого рівня з дотриманням стандартів IEC.

Рисунок 2 – Області застосування стандартів IEC
для оцінки надійності АСУТП

Найбільш поширеними методами є якісні і напівякісні методи ранжирування ризику при оцінці поточного і необхідного рівня SIL. Однак уявна простота застосування таких методів значно нівелюється рівнем їх недостовірності.

Первинна проблема виникає при застосуванні HAZOP з подальшим поділом функцій безпеки. Аналіз небезпеки і працездатності систем безпеки

проводиться методами експертних оцінок і не дозволяє в повній мірі виділити функціонал безпеки з функцій засобів подвійного призначення або навіть засобів захисту. Формалізація причинно-наслідкових зв'язків відмов систем управління і наслідків таких відмов без кількісних показників надійності і ризику є в значній мірі профанацією. У зв'язку з цим актуальною є розробка методів і моделей, що комбінують HAZOP і FMEA з можливістю формалізації причинно-наслідкових зв'язків відмов і подій, ними викликаних, до рівня графів або дерев відмов і дерев подій. При цьому важливо вийти на кількісні показники надійності і небезпеки, а не тільки на рангові оцінки.

Безпеку програмних засобів необхідно оцінювати на всіх стадіях життєвого циклу: при системному аналізі проекту, проектуванні, розробці, тестуванні, верифікації та валідації, тестових випробуваннях, експлуатації та супроводі, модифікації і створенні нових версій, виведенні з експлуатації. На всіх стадіях будь-які впливи можуть мати наслідки для безпеки і змінюють показники надійності.

Ця обставина сприяла появі деяких технологій [14-16]:

1. Microsoft Solutions Framework (MSF). Методики призначені для створення проектів та прийняття рішень на принципах адаптованої моделі колективної розробки засобами Microsoft Visual Studio. Розробка ПО реалізується поетапно з використанням розподілених контрольних точок («водоспад»), а етапи розробки можуть повторюватися («спіраль»).

2. Rational Unified Process (RUP). Проект оформляється у вигляді розподіленої Web бази знань з використанням засобів пошуку та виділення подій. Методи забезпечують розподіл ролей та обов'язків у команді програмістів і реалізуються засобами автоматизації окремих етапів створення.

3. EXtreme Programming (XP). Методи орієнтовані на підвищення ефективності взаємодії як команди програмістів, так і постановників і замовників за рахунок циклів погоджень і перевірок чергових частин вимог замовника.

Основні проблеми, які виникають при використанні зазначених технологій і можуть призводити до відмов функціонування ПО, є помилки програмування і алгоритмізації, що може бути усунуто в достатній мірі методами комплексного тестування, перевірки та затвердження при розробці і супроводі ПЗ.

При цьому використовуються наступні види тестування:

– модульне тестування – для груп незалежних модулів із замкнутою повнотою функціонування;

– інтеграційне тестування – враховує функціональні зв'язки між групами модулів;

– системне тестування – перевірка коректності всього пакету ПО, відповідності продуктивності, критичним навантаженням, помилок користувача, стійкості до програмних і апаратних збоїв.

Верифікація та валідація ПО передбачена стандартами [17-19].

Етапи розробки систем захисту ПО передбачають [20-23]:

– пошук і виділення функцій безпеки ПЗ;
– визначення принципів безпеки функціонування ПЗ;
– види і критерії відмов ПЗ;

– рівні безпеки функціонування ПЗ;
– перелік зовнішніх і внутрішніх впливів, які становлять загрозу безпеці;
– ресурси, необхідні для забезпечення РПБ;
– формування і реалізація систем захисту ПЗ.

Категоріювання видів відмов і їх виявлення є трудомісткою функцією і вимагає високої кваліфікації і глибокого аналізу функціональних зв'язків усередині системи безпеки.

Виділення ресурсів необхідно виконувати з дотриманням принципів надмірності як ресурсів пам'яті, так і часу виконання елементів робочого циклу. При цьому важливо забезпечити:

– контроль зовнішніх даних на відповідність області визначення і застосування ПО;
– кошти on-line котроля правильності виконання програм і трансляції даних;
– засоби реагування на загрози національній безпеці (пастки);
– оперативні процедури відображення виявлення дефектів (визначуваних) і відновлення обчислень після збоїв.

При цьому більш дієвими є системи безпеки, інтегровані у вихідний код до компіляції. Однак такий підхід суттєво ускладнює код і процедури верифікації.

Засоби забезпечення безпеки повинні протистояти зовнішнім і внутрішнім загрозам з заданим рівнем надійності, більш ефективним, ніж це передбачено заявленим РПБ. При цьому необхідно враховувати, що повне усунення будь-яких проявів таких загроз нездійсненно.

Для реалізації систем захисту зазвичай необхідно формувати команду таких фахівців:

– менеджер безпеки проекту (лідер), який зобов'язаний забезпечити вимоги замовника з безпеки засобів АСУТП;
– архітектори систем захисту і розробки базової специфікації функціоналу ПС при критичних рішеннях;
– фахівці, які розробляють весь функціонал компонентів захисту і зв'язок деталей функціоналу (алгоритмізація) для коректного створення вихідного коду і його верифікації;
– програмісти, рівень яких відповідає вибраній специфікації коду;
– фахівці, які здійснювали фонову перевірку і тестування коду;
– фахівці, здатні розробити підсумкові документи з експлуатації систем безпеки відповідно до вимог стандартів.

Верифікація ПЗ проводиться різними методами, які повинні бути обрані на початковій стадії розробки.

Одним з найбільш поширених і недорогих методів є експертні оцінки. Наприклад, оцінка по Фагану (Fagan software inspection) [24] заснована на використанні наскрізного технічного контролю (brainstorming). Додатково можуть використовуватися методи інспекції інтерфейсу користувачів [25] і експертизи якості архітектури і захисту ПО [26-27].

Застосування статичного аналізу вихідного коду і його архітектури. Однак цей метод пов'язаний зі значними труднощами в застосуванні керуючих систем критичного значення в зв'язку з неможливістю прямого транслювання коду

таких систем в загальноприйняті мови високого рівня, що обмежує можливості автоматизації перевірки компонент функціонального ПЗ.

Формальні і напівформальні методи верифікації ПЗ засновані на розробці вимог до логіко-алгебраїчних моделей і абстрактних моделей. Такі моделі в деяких випадках можуть бути формалізовані до логічного рівня і забезпечити розробку інструментальних засобів автоматизованого процесу дозволу ряду завдань верифікації ПЗ. Приклад побудови предикат в деяких випадках логічних обчислень з отриманням кількісних показників ймовірності відмови наводиться в цій статті.

4. Пропозиції та методи комплексної оцінки ризику і РПБ АСУТП

В результаті проведеного аналізу та вивчення нормативної бази можна запропонувати окремі методи визначення кількісних показників РПБ на основі стохастичних показників надійності дискретних елементів керуючої системи і якісних показників програмного забезпечення АСУТП.

Визначення РПБ досліджуваної апаратної частини складової АСУТП пропонується здійснювати гібридними методами експертного аналізу, що поєднує стандартний підхід до визначення області небезпеки наслідків відмов окремих елементів апаратного забезпечення на базі HAZOP аналізу і автоматизованих методів оцінки ймовірності таких відмов. Аналіз небезпеки і працездатності необхідно проводити з використанням спеціальних протоколів, в яких відображаються причинно-наслідкові зв'язки між можливими причинами відмов вихідних елементів, їх впливом на працездатність системи управління і наслідками втрати функцій системи в результаті відмов. Використання структурованих записів таких причинно-наслідкових зв'язків, оформлених, наприклад, засобами мови структурованої розмітки (xml), дозволяє автоматизувати процес створення узагальненої математичної моделі оцінки SIL для заявленої системи управління. Така модель представляється кортежем (або графом) рівня надійності та безпеки і може бути формалізована до стану згортки / розгортки дерев відмов (FTA) і дерев подій (ETA). Причому в якості вихідних (ініціюючих) відмов або подій можуть розглядатися і елементи програмного забезпечення, що застосовується в АСУТП.

Авторами статті були виконані дослідження при постановці завдань, розробці алгоритмів, верифікації та впровадженні програмних засобів підтримки прийняття рішень при оцінці ризику великих промислових підприємств [28-29]. Перевірені можливості вищеописаного протоколу і достовірність результатів обчислень автоматизованих побудов FTA і ETA на базі логічних відносин причинно-наслідкових зв'язків аналізованих елементів АСУТП. Використання логічних операцій І (Заборона), АБО (див. Таб. 1) [30] для низхідного методу розгортки дерев відмов і бінарного розгалуження подій при впливах засобів захисту, представлених в деревах подій, дозволяє здійснити кількісні оцінки ймовірності виникнення негативних наслідків відмов елементів АСУ ТП.

Таблиця 1 – Відповідність формул визначення ймовірності логічних операцій

І (∧)	АБО (∨)	виключає АБО (⊕)
$P_e = \prod_{i=1}^{n} P_i$	$P_e = 1 - \prod_{i=1}^{n} (1 - P_i)$	$P_e = \sum_{i=1}^{n} P_i$

Програмні засоби (приклад наведено на рис. 3.) підтримки автоматизованого процесу формування FTA, ETA на основі використання протоколу аналізу HAZOP дозволяють в повній мірі здійснити проект кількісної оцінки SIL і виділити і сортувати поєднання відмов, які впливають на критичність наслідків у міру їх значущості, що дає можливість оптимізувати прийняті рішення.

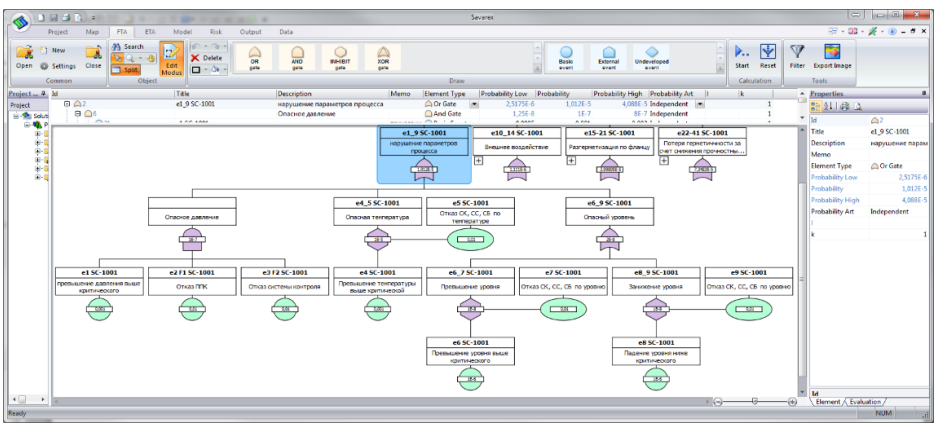

Рисунок 3 – Приклад автоматичної розгортки дерева відмов з графа протоколу HAZOP

Автономне тестування модулів програмного забезпечення базових частин АСУТП об'єктів підвищеної небезпеки може бути виконано на основі абстрактних синтаксичних побудов деревовидної форми [31-33].

Висновки

Існуючі моделі і методи, що дозволяють встановити рівень повноти безпеки систем управління об'єктами підвищеної небезпеки, не в повній мірі відповідають сучасним вимогам проведення сертифікаційних процедур. Науково-технічна проблема комплексної системної підтримки прийняття рішень в області створення систем безпеки АСУТП є актуальною.

Зниження суб'єктивної складової оцінки ризику і рівня РПБ є важливим завданням при сертифікації засобів АСУТП і може досягатися методами кількісної оцінки ймовірності відмов апаратної і програмної складових комплексів управління об'єктами критичної значущості.

Раціональними для оцінки ймовірності відмов апаратної частини є методи дерев відмов (FTA) для ініціюючих небезпечних подій і метод дерев подій (ETA) для відмов систем захисту і визначення сценаріїв наслідків таких відмов.

Розробка програмних комплексів, за допомогою яких може бути реалізована інформаційна технологія підтримки прийняття рішень при забезпеченні необхідного рівня SIL для керуючих комплексів об'єктів підвищеної небезпеки, актуальна і здійсненна при використанні пропонованих в статті методів і моделей.

СПИСОК ЛІТЕРАТУРИ

1. Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью = Ч. 1. Общие требования: национальный стандарт Российской Федерации ГОСТ Р МЭК 61508-1-2007 / Федеральное агентство по техническому регулированию и метрологии. – М.: Стандартинформ, 2008. – V, 44 с.
2. Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью = Ч. 2. Требования к системам: национальный стандарт Российской Федерации ГОСТ Р МЭК 61508-2-2007 / Федеральное агентство по техническому регулированию и метрологии. – М.: Стандартинформ, 2008. – V, 58 с.
3. Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью = Ч. 3. Требования к программному обеспечению: национальный стандарт Российской Федерации ГОСТ Р МЭК 61508-3-2012 / Федеральное агентство по техническому регулированию и метрологии. – М.: Стандартинформ, 2014. – V, 97 с.
4. Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью = Ч. 6. Руководство по применению ГОСТ Р МЭК 61508-2-2007 и ГОСТ Р МЭК 61508-3-2007 : национальный стандарт Российской Федерации ГОСТ Р МЭК 61508-6-2007 / Федеральное агентство по техническому регулированию и метрологии. - Москва : Стандартинформ, 2008. – V, 62 с.
5. Функциональная безопасность в непрерывных производствах. Руководство по безопасности процессов / национальный стандарт Российской Федерации ГОСТ Р МЭК 61511-1-2011 / Федеральное агентство по техническому регулированию и метрологии. – М.: Стандартинформ, 2013. – V, 66 с.
6. Руководство по функциональной безопасности для систем, связанных с безопасностью, и других применений с уровнем SIL2, SIL3 в соответствии со стандартами МЭК 61508 и МЭК 61511 / GM International Technology for safety / Via San Fiorano 70, 20058 Villasanta (MI) Italy, 2013. – D100, 77 p.
7. Dr. David J. Smith. Reliability, Maintainability and Risk. Practical methods for engineers. Butterworth-Heinemann. The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK. 225 Wyman Street, Waltham, MA 02451, USA. Eighth edition 2011.
8. Безопасность программных средств: модели и методы (обзор) / Д. Б. Буй, В. Г. Скобелев // Радіоелектронні і комп'ютерні системи. - 2014. – № 1.
9. Липаев В. В. Обеспечение качества программных средств. Методы и стандарты. М.: СИНТЕГ, 2001. – 380 с.
10. Lyu M. R. Software Fault Tolerance. Published by John Wiley & Sons Ltd, 1996.

11. Ковалев И. В., Золотарёв В. В., Жуков В. Г., Жукова М. Н. Методика построения модели безопасности автоматизированных систем // Программные продукты и системы. 2012. № 2. С. 16.
12. Ландрини, Г. Критерии выбора компонентов с уровнем SIL 3 для РСУ и систем ПАЗ в соответствии со стандартами МЭК / Глизенте Ландрини // Современные технологии автоматизации. - 2009. – N 3. – С. 110-114.
13. Michael A. Mitchell. SIL – it is not difficult. Valve World Conference 2010. «Промышленная безопасность». – 2011. – № 5 (74).
14. Microsoft solutions framework. – URL: http://www.microsoft.com/Rus/Msdn/msf/Default.mspx.
15. Rational Unified Process. Методология и технология. Материалы компании Interface Ltd– URL: http://www.interface.ru/home.asp?artId=779.
16. Бек К. Экстремальное программирование / К.Бек. – СП-б: Питер, 2002. – 224 с.
17. IEEE 610.12-1990 Standard glossary of soft-ware engineering terminology, corrected edition [Текст]. – IEEE, 1991.
18. IEEE 1012-2004 Standard for verification and validation [Текст]. – IEEE, 2005.
19. ISO/IEC 12207 Systems and software engi-neering – software life cycle processes [Текст]. – ISO, 2008.
20. Галатенко В.А. Основы информационной безопасности [Текст] / В.А. Галатенко. – М.: ИНТУИТ, 2003. – 208 с.
21. Липаев В.В. Технологические процессы и стандарты обеспечения функциональной безопасности в жизненном цикле программных средств [Текст] / В.В. Липаев // Jet Info. – 2004. – № 3. – С. 2-19.
22. Липаев В.В. Функциональная безопасность программных средств [Текст] / В.В. Липаев // Jet Info. – 2004. – № 8. – С. 3-28.
23. Волобуев С.В. Философия безопасности социотехнических систем: информационные аспекты [Текст] / С.В. Волобуев. – М.: Вузовская книга, 2004. – 360 с.
24. Fagan M.E. Design and code inspections to reduce errors in program development [Текст] / M.E. Fagan // IBM Systems Journal. – 1976. – N 3. – P. 182-211.
25. Константайн Л. Разработка программного обеспечения [Текст] / Л. Константайн, Л. Ло-квуд. – СПб: Питер, 2004. – 592 с.
26. Anderson R. Security engineering: a guide to building dependable distributed systems [Текст] / R. Anderson. – NY: John Wiley & Sons, 2001. – 1040 p.
27. Dobrica L. A survey on software architecture analysis methods [Текст] / L. Dobrica, E. Niemela // IEEE Transactions on software engineering. – 2002. – № 7. – P. 638-653.
28. Лифарь В. А. Разработка метода оптимизации проведения ремонтно-восстановительных работ с учетом показателей риска / В. А. Лыфарь, С. А. Сафонова, В. Г. Иванов // Технологический аудит и резервы производства. – 2015. – № 2/2(22) – С. 11-17.
29. Лифар В.О. Моделі, методи та інформаційні технології оцінки техногенного ризику об'єктів підвищеної небезпеки: дис. ... д-ра техн. наук : 05.13.06 / Лифар В. О.; [Місце захисту: Чорноморський національний університет імені Петра Могили]. – Миколаїв, 2017. – 309 с.
30. Хенли Э. Дж., Кумамото Х. Надежность технических систем и оценка риска: Пер. с англ. В. С. Сыромятникова – М.: Машиностроение, 1984. – 528 с.
31. S. Nair, R. Jetley, A. Nair, "A Static Code Analysis Tool for Control System Software", SANER 2015, Montréal, Canada, pp. 459-463.
32. F. Narisco, A.-R. Bolivar, F. Hidrobo, O. Gonzalez, "A Syntactic Specification for the Programming Languages of the IEC 61131-3 Standard", Advances in Computational Intelligence, Man-Machine Systems and Cybernetics, pp. 171-176, 2010.

33. Müller and M. I. Schwartzbach, “Static Program Analysis”, Department of Computer Science Aarhus University, Denmark, 113 p., 2018. http://users-cs.au.dk/amoeller/spa/spa.pdf.

REFERENCES

1. (2008) Functional safety of electrical, electronic, programmable electronic safety-related systems. Part 1. General requirements: national standard of the Russian Federation GOST R IEC 61508-1-2007 / Federal Agency for Technical Regulation and Metrology. – Moskva.: Standartinform. – V, 44 s.
2. (2008) Functional safety of electrical, electronic, programmable electronic safety-related systems. Part 2. Requirements for systems: national standard of the Russian Federation GOST R IEC 61508-2-2007 / Federal Agency for Technical Regulation and Metrology. – Moskva.: Standartinform. – V, 58 s.
3. (2014) Functional safety of electrical, electronic, programmable electronic safety-related systems. Part 3. Software requirement: national standard of the Russian Federation GOST R IEC 61508-3-2012 / Federal Agency for Technical Regulation and Metrology. – Moskva: Standartinform. – V, 97 s.
4. (2008) Functional safety of electrical, electronic, programmable electronic safety-related systems. Part 6. Guidelines on the application of GOST R IEC 61508-2-2007 and GOST R IEC 61508-3-2007 : national standard of the Russian Federation / Federal Agency for Technical Regulation and Metrology. - Moscow: Standartinform. – V, 62 s.
5. (2013) IEC 61511:2004 Functional Safety – Safety Instrumented Systems for the Process Industry Sector/ national standard of the Russian Federation / Federal Agency for Technical Regulation and Metrology. – M .: Standartinform. – V, 66 s.
6. (2013) Functional safety guidelines for safety related systems and other applications with SIL2, SIL3 level in accordance with IEC 61508 and IEC 61511 / GM International Technology for safety / Via San Fiorano 70, 20058 Villasanta (MI) Italy – D100, 77 p.
7. Dr. David J. Smith (2011). Reliability, Maintainability and Risk. Practical methods for engi-neers. Butterworth-Heinemann. The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK. 225 Wyman Street, Waltham, MA 02451, USA. Eighth edition.
8. Software Security: Models and Methods (review) (2014). / D. B. Bui, V. G. Skobelev // Radio Electronics and Computer Systems. No. 1. (In Ukrainian).
9. Lipaev V.V. (2001). Software quality assurance. Methods and standards. Moskva: SINTEG, 380 s. (In Russian).
10. Lyu M. R. (1996). Software Fault Tolerance. Published by John Wiley & Sons Ltd.
11. Kovalev I.V., Zolotarev V.V., Zhukov V.G., Zhukova M.N. (2012). Methodology for constructing a security model for automated systems // Software Products and Systems. No. 2. P. 16. (In Russian).
12. Landrini, G. (2009). Criteria for choosing components with a SIL 3 level for DCS and PAZ systems in accordance with IEC / Glisente Landrini standards // Modern Automation Technologies. N 3. – s. 110-114
13. Michael A. Mitchell. (2011). SIL - it is not difficult. Valve World Conference 2010. "Industrial Safety". № 5 (74)
14. Microsoft solutions framework. URL: http://www.microsoft.com/Rus/Msdn/ msf/Default.mspx.
15. Rational Unified Process. Methodology and technology. Company Materials Interface Ltd URL: http://www.interface.ru/home.asp?artId=779.
16. Beck K. Extreme Programming. (2002) - SP-b: Peter, s. 224. (In Russian).
17. IEEE 610.12-1990 Standard glossary of soft-ware engineering terminology, corrected edition – IEEE, 1991.
18. IEEE 1012-2004 Standard for verification and validation IEEE, 2005.

19. ISO/IEC 12207 Systems and software engi-neering – software life cycle processes – ISO, 2008.
20. Galatenko V.A. (2003). The basics of information security. - Moskva.: INTUIT, s. 208. (In Russian).
21. Lipaev V.V. (2004). Technological processes and standards for ensuring functional safety in the software life cycle // Jet Info. No. 3. s. 2-19. (In Russian).
22. Lipaev V.V. (2004). Functional Security Software // Jet Info. No. 8. s. 3-28. (In Russian).
23. Volobuev S.V. (2004). The safety philosophy of socio-technical systems: informational aspects. Moskva: University Book, s. 360. (In Russian).
24. Fagan M.E. (1976). Design and code inspections to reduce errors in program development IBM Systems Journal. N 3. – P. 182-211.
25. Konstantin L. (2004). Software Development - St. Petersburg: Peter, s. 592. (In Russian).
26. Anderson R. (2001). Security engineering: a guide to building dependable distributed systems. NY: John Wiley & Sons, s. 1040.
27. Dobrica L. (2002). A survey on software architecture analysis methods IEEE Transactions on software engineering. № 7. s. 638-653.
28. Lifar V. A. Safonova S. A., V. G. Ivanov (2015). Development of a method for optimizing repair and restoration work taking into account indicators Technological audit and production reserves. – No. 2/2 (22) – s. 11-17. (In Ukrainian).
29. Lifar V.O. (2017). Models, Methods and Information Technologies for Evaluating Technogenic Risics of Public Prospects: Dis. ... Dr. tech. Sciences: 05.13.06; [I will clean it up: The Chornomorsk National University of the Name of Peter Mogili]. – Mikolaev, s. 309.
30. Henley E.J., Kumamoto H. (1984). Reliability of technical systems and risk assessment:, s. 528.
31. S. Nair, R. Jetley, A. Nair, (2015). “A Static Code Analysis Tool for Control System Software”, SANER, Montréal, Canada, s. 459-463.
32. F. Narisco, A.-R. Bolivar, F. Hidrobo, O. Gonzalez (2010). “A Syntactic Specification for the Programming Languages of the IEC 61131-3 Standard”, Advances in Computational Intelligence, Man-Machine Systems and Cybernetics. s. 171-176.
33. Müller and M. I. Schwartzbach. (2018). “Static Program Analysis” Department of Computer Science Aarhus University, Denmark, s. 113. http://users-cs.au.dk/amoeller/spa/spa.pdf.

Стаття надійшла до редакції 11.07.2019.