кий політехнічний інститут». Збірник наукових праць. Тематичний випуск: Техніка і електрофізика високих напруг. – Харків: НТУ «ХПІ». – 2005. – № 49. – С. 71−84. **8.** Межгосударственный ГОСТ 1516.2-97. Электрооборудование и электроустановки переменного тока на напряжения 3 кВ и выше. Общие методы испытаний электрической прочности изоляции. – Минск: Изд-во стандартов, 1998. – 31 с. **9.** *Рудаков В.В., Бойко Н.И., Беспалов В.Д., Кравченко В.П. и др*. Высоковольтные импульсные конденсаторы разработки НИПКИ «Молния» НТУ «ХПИ» // Вісник Національного технічного університету «Харківський політехнічний інститут». Збірник наукових праць. Тематичний випуск: Електроенергетика і перетворююча техніка. – Харків: НТУ «ХПІ». – 2002. – № 7, т. 1. – С. 47-58. **10.** *Пекарь И.Р., Бочаров В.А., Штагер П.И. и др*. Многозазорные коммутаторы на 100 и 200 кВ (МЗК-100, МЗК-200) // Приборы и техника эксперимента. – 1984. – № 2. – С. 234. **11.** *Баранов М.И.* Сравнительный анализ работы двух схем построения генераторов высоковольтных поджигающих импульсов напряжения мощных электрофизических установок // Вісник Національного технічного університету «Харківський політехнічний інститут». Збірник наукових праць. Тематичний випуск: Техніка і електрофізика високих напруг. – Харків: НТУ «ХПІ». – 2006. – № 37. – С. 100-106.

Поступила в редколлегию 04.06.2007.

УДК 519.688

В.С.БРЕСЛАВЕЦ, канд.техн.наук; ***С.А.НИКИТИН*****;** НТУ «ХПИ»

АНАЛИЗ МЕТОДОВ ОБЕСПЕЧЕНИЯ НАДЕЖНОСТИ ФУНКЦИОНИРОВАНИЯ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ

У статті розглянуті основні методи ідентифікації збоїв програмного забезпечення, які можуть викликати системні ризики. Запропонована технологія аналізу системних ризиків, сформульовані основні вимоги для розробки програмного забезпечення, яке задовольняє критеріям максимального захисту від системних ризиків.

The basic methods for identification of software failures that can provoke system risks are discussed. A technique is proposed for analyses of system risks and a set of basic criteria is formulated to ensure system risk safe software development.

Введение. Существует несколько методов анализа рисков, связанных с функционированием программного обеспечения. Анализ рисков – это техника, предназначенная для обнаружения и обработки рисков, производимых системой с учетом ее оборудования, например неблагоприятные события, исходящие от других систем или интерфейсов системы, и затем изменяющая рекомендации по уменьшению опасности или преобразование ее риска к «приемлемому уровню». Традиционно такой анализ не включал в себя анализ программного обеспечения. Анализ рисков фокусируется на роли программного обеспечения относительно рисков вообще. Это пошаговая техника, которая может быть использована на любой стадии жизненного цикла.

Постановка проблемы. Однако, поскольку программное обеспечение, входящее в состав большинства систем, может послужить причиной физического повреждения, необходимость в идентификации сбоев программного обеспечения, которые могут вызвать системные риски (в дальнейшем будем называть их программными рисками) становится все более важным.

Анализ литературы. Разработано несколько технологий для выполнения анализа программных рисков [1]. Многие из них основываются на методах анализа безопасности аппаратного обеспечения [2]. Однако программное и аппаратное обеспечение во многом отличаются; ошибки аппаратного обеспечения чаще всего являются случайными, вызванными возрастом аппаратуры или некачественной сборкой [3-5], в то время как сбои программного обеспечения появляются в результате ошибок разработки и реализации [6-8]. Поэтому очень актуальным является развитие технологий для анализа программного обеспечения и его характеристик.

Целью статьи является анализ программных рисков, позволяющих сформировать рекомендации для уменьшения рисков или управления программными рисками и рисками, относящимися к интерфейсам между программным обеспечением и системой (включая аппаратное обеспечение и человеческий фактор). В его состав входят анализ требований, процесса разработки кода, интерфейса пользователя и внесение изменений.

Основная часть. Программные риски появляются в случае, когда программное обеспечение не разрабатывается корректно, обрабатывает некорректную информацию и при сбоях программного обеспечения при передаче информации. Технология анализа программных рисков является относительно молодой. Анализ программных рисков имеет прямое отношение к анализу системных рисков, поскольку зависит от его результатов. Анализ программных рисков должен:

- быть связанным каждым риском, определенным в результате анализа системных рисков
- удостовериться, что работа программного обеспечения не мешает выполнению целей и действий системы
- оценивать и давать рекомендации по нивелированию мешающего воздействия программного обеспечения на цели и работу системы.

Существует несколько технологий для выполнения анализа рисков. Некоторые из них используются для оценки аппаратного обеспечения (напр. анализ режима и результата сбоев и анализа дерева сбоев) и являются относительно новыми для области программного обеспечения. Каждая технология должна быть привязана к определенным приложениям с учетом их размера и стоимости. Полный анализ программных рисков предполагает при-

менение более одной технологии анализа программных рисков из-за достоинств и недостатков, присущих каждой технологии.

Анализ рисков – это итеративный четырехшаговый процесс (рис. 1), содержание каждого шага показано на рис. 2. Шаг 1 состоит из определения границ и назначения анализа рисков, планирования анализа рисков (задача 1) и определения анализируемой системы. Шаг 1 выполняется в самом начале проекта.

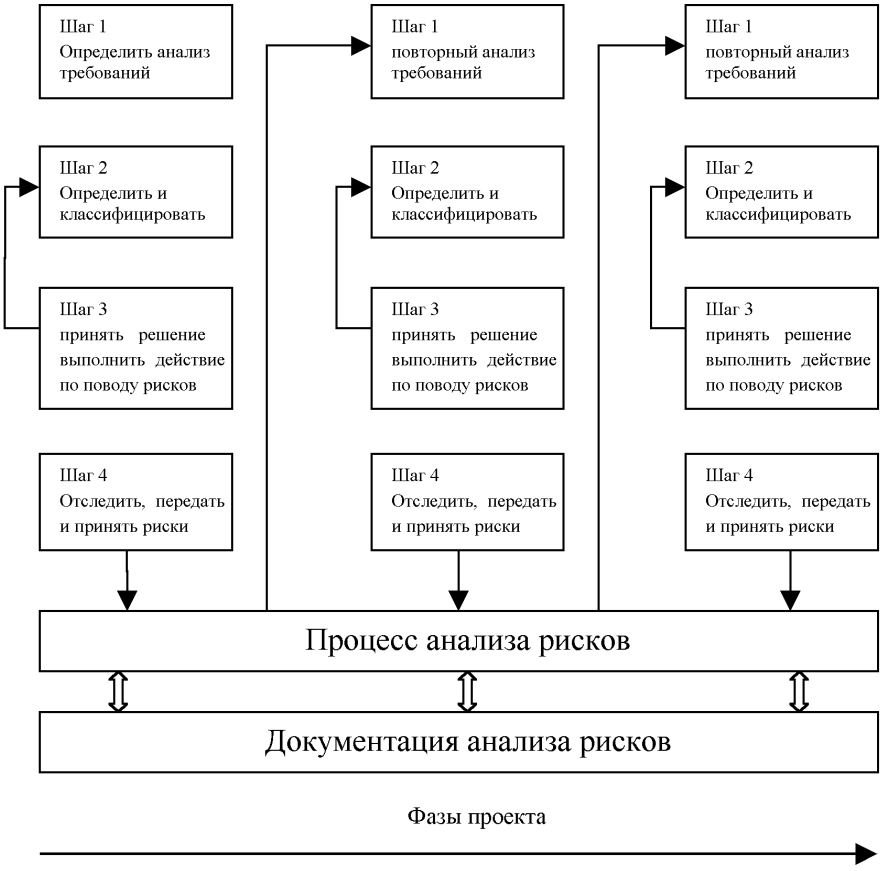

Рисунок 1 – Задачи анализа во время жизненного цикла проекта

На шаге 2 выполняется идентификация и классификация рисков относительно программного обеспечения (задания 3 и 4). Одним из примеров идентификации системных рисков, вызванных сбоями программного обеспечения, может служить представление SFMECA или SFTA, где режимы сбоев

программного обеспечения являются причинами системных сбоев. Например, в случае использования SFMECA для выполнения задачи 3 на потенциальные режимы сбоев во время функционирования могут относиться к:

– сервисному обслуживанию, являющемуся критичным по времени – основными режимами реакции на сбои являются отказ (сервис или функции не предоставляются) и передача полномочий (предоставление сервиса и функций в неположенное время или предоставление неправильно работающего сервиса или функций);
– значению сервиса – основными режимами сбоев являются неправильные значения, нулевые значения или отсутствие значений.

Другие шаги (3 и 4) анализа рисков относятся к уровню системного анализа. В зависимости от границ и назначения выполнение процессов анализа рисков состоит из ряда «циклов анализа рисков» во время выполнения всего проекта, и охватывает пересмотр анализа требований и шаги со 2 по 4, подразделяющиеся на 7 задач от 3 до 9.

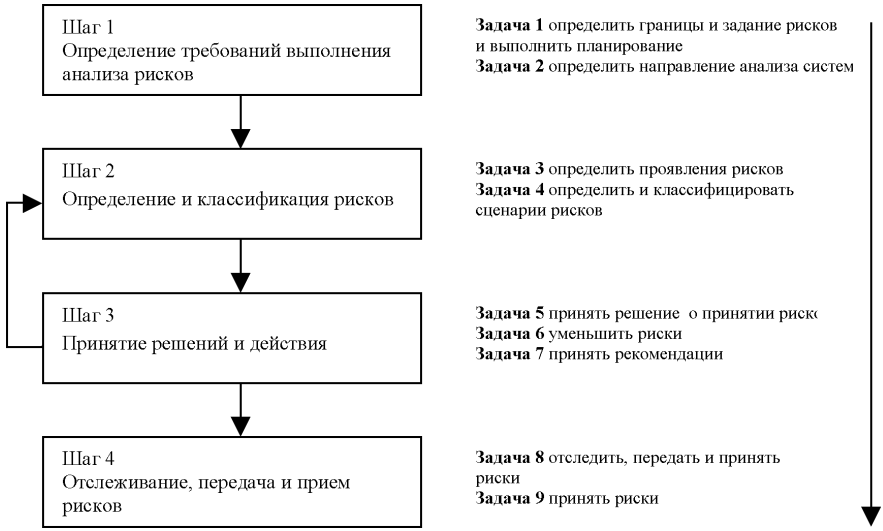

Рисунок 2 – Задачи шагов анализа

Критерии технологий анализа рисков можно сформулировать следующим образом:

– результаты применения технологии облегчат понимание путей увеличения рисков, их предотвращения или уменьшения;
– технология позволит моделировать и оценивать широкий диапазон режимов сбоев;
– технология может использоваться для аттестации персонала;

– технология может адаптироваться для систем заданной сложности в заданной области, содержащей заданные риски;
– технология может дать значимые результаты с использованием данных с количественными и качественными показателями;
– технология может адаптироваться к определенной фазе жизненного цикла, в которой она применяется;
– для поддержки технологии могут разрабатываться или поставляться коммерческие программные средства;
– есть возможность разработки полностью документированных примеров успешного применения или поставки письменных правил ее применения

Анализ риска, выполненный на стадии составления требований, позволяет убедиться в том, что определены требования безопасности системы и они могут быть прослежены от системных требований к требованиям ПО, разработки ПО, к руководству оператора, пользователя и по диагностике. Анализ системных рисков предоставляет входные данные для выполнения этого анализа. На этой стадии анализ рисков проверяет системные требования и требования к ПО. Рекомендации к разработке и требования по тестированию объединяются в системные спецификации, спецификацию требований к ПО, документацию по разработке ПО, план тестирования ПО, план управления конфигурацией и план управления проектом. Результаты выполнения технологии анализа на стадии выполнения требований представляются в виде обзора системных требований (предварительные результаты), обзора системы и предварительной разработки (окончательные результаты).

Анализ рисков на стадии разработки архитектуры определяет компоненты ПО, критичные по безопасности. Он начинается после обзора требований к ПО и должен быть завершен до начала составления программных кодов. Анализ рисков, выполненный на стадии составления требований, формирует входные данные для анализа на стадии разработки архитектуры. Анализ рисков на стадии разработки архитектуры определяет и анализирует компоненты программного обеспечения, критичные по безопасности, на этом же этапе разрабатывается план тестирования. Составляются рекомендации для написания программного кода. Результаты анализа рисков, выполненные на стадии разработки архитектуры, представляются в виде предварительного обзора разработки.

Выводы. В статье были рассмотрены различные риски при разработке программного обеспечения. Было выявлено, что анализ рисков, выполняемый на стадии детальной разработки и написания программных кодов, определяет пути для исключения рисков или снижения их вероятности. Анализ рисков, выполненный на стадии разработки архитектуры, является исходным для этого анализа. Результаты анализа рисков, выполненного на стадии де-

тальной разработки и составления программных кодов, представляется в виде критического обзора разработки. Они могут использоваться для функционального анализа, когда система еще не разделена на компоненты программного и аппаратного обеспечения.

Список литературы: 1. *Watson H.* and *McCabe Thomas J.* Structured testing: A Testing Methodology Using the Cyclomatic Complexity Metric // National Institute of Standards and Technology, NIST Special Publication 500-235, August 1996. **2.** ESA PSS-05-10, Guide to software verification and validation // ESA BSSC, Issue 1 Revision 1, March 1995. **3.** *De Vale John Peter* High Performance Robust Computer Systems // (Ph.D. Thesis, 2002), Pittsburgh, Pennsylvania October 2001 http://www.ece.cmu.edu/Ekoopman/thesis/devale.pdf. **4**. *Arlat Jean* et al Fault injection and dependability evaluation of fault-tolerant systems // IEEE Transactions on Computers. – Vol. 42. – № 8. – August 1993. – PP. 913-923. **5.** *Voas Jeffrey* and *McGraw Gary* Software Fault Injection: Inoculating Programs Against Errors // Ed. John Wiley & Sons, ISBN 0-471-18381-4. **6.** *J. C Laprie* Dependability: Basic Concepts and Terminology. Dependable Computing and Fault Tolerance // Vienna, Austria: Springer-Verlag, 1992. **7.** Standard Practice For System Safety – MIL-STD-882D US DoD 10 February 2000. **8.** Military Standard Software Development And Documentation. MIL-STD-498. U.S. DoD. December 1994.

Поступила в редколлегию 02.04.2007.

УДК 537.528:537.529

В.С.ГЛАДКОВ, канд.техн.наук; ***О.А.ГУЧЕНКО***; ***О.В.ШЕСТЕРІКОВ***; ***І.В.ЯКОВЕНКО***, докт.фіз.-мат.наук; НТУ «ХПІ»

ПРОПОНОВАНА ІНЖЕНЕРНА МЕТОДИКА РОЗРАХУНКУ ЕФЕКТИВНОСТІ РУЙНУВАННЯ БЕТОНУ В ЗАЛЕЖНОСТІ ВІД ПАРАМЕТРІВ ІМПУЛЬСУ НАПРУГИ ТА КЛАСУ БЕТОНІВ

Запропоновано інженерну методику розрахунку ефективності руйнування бетону при дії імпульсів напруги наносекундного діапазону в залежності від параметрів імпульсів та класу бетону, яка побудована на базі урахування електричного пробою повітряних пор у товщі бетонів.

Engineering procedure for calculation of efficiency of destruction of concrete under the action of voltage pulses of nanosecond range depending on pulse parameters has been proposed. The procedure is based on taking into account electrical breakdown of air pores in depth of concrete.

Метою цієї роботи є викладення розробленої інженерної методики розрахунку ефективності руйнування бетону в залежності від параметрів імпульсу напруги і класу бетонів

На цей час існує не дуже багато інженерних методик розрахунку ефективності електроімпульсного руйнування матеріалів, які за своїми властивостями подібні до бетону. У [1] розроблена інженерна методика розрахунку кі-