УДК 681.3.06

В.В. СКЛЯР, В.С. ХАРЧЕНКО

РЕАЛИЗАЦИЯ ПРОЦЕССА ВЕРИФИКАЦИИ ДЛЯ РАЗРАБОТКИ НАДЕЖНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Abstract: *Requirements of standards to verification of software are generalised, that permits to structure this process and to develop a method of integrated verification which is directed to assurance of software reliability. The method of integrated verification is universal and can be used for any type of software.*
Key words: *software verification, standards, life cycle.*

Анотація: *Узагальнені вимоги стандартів до процесу верифікації програмного забезпечення (ПЗ), що дозволило структурувати цей процес та розробити метод комплексної верифікації, спрямований на забезпечення надійності ПЗ. Метод комплексної верифікації є універсальним і може бути застосований для ПЗ любого типу.*
Ключові слова: *верифікація програмного забезпечення, стандарти, життєвий цикл.*

Аннотация: *Обобщены требования стандартов к процессу верификации программного обеспечения (ПО), что позволило структурировать данный процесс и разработать метод комплексной верификации, направленный на обеспечение надежности ПО. Метод комплексной верификации является универсальным и может быть применен для ПО любого типа.*
Ключевые слова: *верификации программного обеспечения, стандарты, жизненный цикл.*

1. Введение

В результате развития информационных технологий возросла роль программного обеспечения (ПО) в выполнении требований к надежности компьютерных систем, в том числе, и компьютерных систем технических комплексов критического использования (ТККИ) [1]. Отказы компьютерных систем управления ТККИ составляют около 20% от всех отказов оборудования [2, 3]. Среди отказов компьютерных систем доля отказов ПО по различным оценкам для обслуживаемых систем составляет около 30% [1], для необслуживаемых систем – около 90% [3].

Основным методом повышения надежности ПО является верификация. Под верификацией программного обеспечения подразумевается процесс, направленный на подтверждение соответствия ПО заданным требованиям путем различного рода проверок и обеспечения объективных доказательств [4–6]. Следует подчеркнуть, что процесс верификации должен носить комплексный характер, охватывающий все этапы разработки ПО (анализ требований, проектирование, кодирование, интеграцию), а также изменения, вносимые при сопровождении ПО. Необходимость верификации ПО для ТККИ установлена в стандартах по программной инженерии, а также в различных отраслевых стандартах [7–9].

Общая структура жизненного цикла (ЖЦ) ПО описана в стандарте ДСТУ 3918-1999 (ИСО/МЭК 12207:1995) «Информационные технологии – Процессы жизненного цикла программного обеспечения». Серия стандартов Международной электротехнической комиссии (МЭК) 61508 описывает требования к функциональной безопасности электрических, электронных и программируемых электронных систем, важных для безопасности. Третья часть МЭК 61508 определяет требования к ПО таких систем, включая требования к процессу верификации. Требования к процессу верификации ПО изложены также в стандарте IEEE 1012-1998 «Верификация и валидация программного обеспечения», разработанном Институтом инженеров по электротехнике и электронике, США (Institute of Electrical and Electronics Engineers – IEEE).

Кроме того, в различных критических отраслях разработаны стандарты, определяющие требования к верификации ПО, выполняющего функции, важные для безопасности ТККИ:

– RTCA DO-178B (1992) «Рассмотрение программного обеспечения при сертификации бортовых систем и оборудования» – стандарт Радиотехнической комиссии по аэронавтике (Radio Technical Commission for Aeronautics), содержащий требования к верификации ПО для авиационной техники;

– ECSS-E-10-02A (1998) «Разработка космических систем – Верификация» – стандарт Европейской кооперации по космической стандартизации (European Cooperation for Space Standardization), содержащий требования к верификации ПО для ракетно-космической техники;

– МАГАТЭ NS-G-1.1 (2000) «Программное обеспечение для компьютерных систем, важных для безопасности атомных электростанций. Руководство по безопасности» – стандарт Международного агентства по атомной энергии (International Atomic Energy Agency), содержащий требования к верификации ПО для информационных и управляющих систем АЭС;

– MISRA-C (2004) «Руководство по применению языка С в критических системах» – стандарт Исследовательской ассоциации по вопросам ПО для автомобильной промышленности (Motor Industry Software Research Association), содержащий требования к верификации ПО для автомобильной техники.

Однако, несмотря на обилие стандартов и наличие публикаций по вопросам верификации ПО, в настоящее время не разработана единая концепция, которая позволила бы обобщить требования к процессу верификации ПО и при необходимости произвести адаптацию процесса верификации для ПО различных программируемых компонент, разработанного на различных языках программирования.

Цель статьи – разработка метода комплексной верификации ПО, адаптируемого для ПО различных типов.

2. Разработка структурной схемы процесса верификации ПО ИУС ТККИ, базирующейся на анализе требований стандартов

Анализ стандартов, содержащих требования к верификации ПО ИУС ТККИ, позволил выделить следующие группы требований:

– требования к полноте верификации, определяющие этапы верификации, выполняемые задачи по обеспечению полноты верификации требований, и методики, применяемые для выполнения задач; на объем верификации оказывает влияние апробированность верифицируемого продукта, поскольку при повторном использовании ПО объем выполняемых действий может быть обосновано уменьшен; кроме того, объем выполняемых действий и применяемых методик зависит от степени влияния ИУС на безопасность ТККИ (категория безопасности ИУС), а также от типа программируемого компонента (целевого вычислителя) и используемого языка программирования (например, для верификации ПО на языке программирования Assembler и для ПО, разработанного с использованием проблемно-ориентированных языков, могут применяться различные методы верификации);

– требования к независимости участников верификации от разработчиков ПО; степень независимости лиц, проводящих верификацию, от лиц, выполнявших разработку ПО, определяется категорией безопасности ИУС;

– требования к устранению недостатков по результатам верификации; типы выявляемых недостатков и способы их устранения в общем случае определяются методиками верификации;

– требования к документированию процесса верификации.

Кроме того, стандарты по безопасности требуют, чтобы в процессе лицензирования ИУС ТККИ была проведена экспертная оценка и/или сертификация процесса верификации ПО. Такая оценка проводится на основании разработанных в процессе верификации документов и заключается в проверке соответствия проведенной верификации всем рассмотренным выше требованиям.

Обобщенная структура процесса верификации ПО представлена на рис. 1.

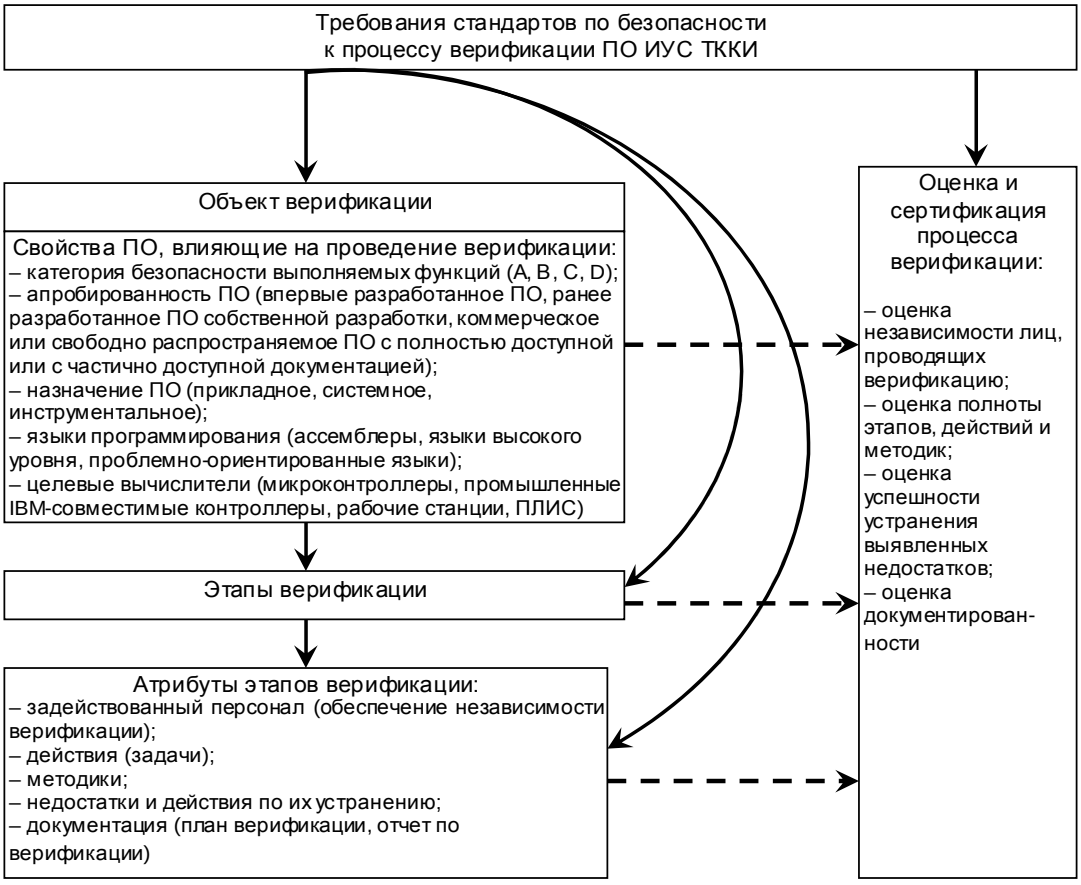

Рис. 1. Обобщенная структура процесса верификации ПО

Основой для проведения верификации ПО являются стандарты, содержащие требования по безопасности к ИУС ТККИ [7–9]. Требования к этапам и атрибутам этапов верификации определяются, исходя из свойств верифицируемого объекта. Кроме того, согласно требованиям стандартов по безопасности, должна быть проведена независимая оценка процесса верификации ПО органом государственного регулирования. Данный факт на рис. 1 отображен в виде пунктирной линии.

3. Разработка метода комплексной верификации ПО ИУС ТККИ

Проведенный анализ структуры процесса верификации позволил перейти к непосредственной разработке метода комплексной верификации ПО ИУС ТККИ, который включает следующую

последовательность действий:

1. Анализ объекта верификации

1.1. Определение категории безопасности функций, выполняемых ПО.

1.2. Определение характеристик ПО, влияющих на объем верификации (согласно рис. 1, к таким характеристикам относятся апробированность, назначение, язык программирования, целевой вычислитель).

2. Планирование верификации

2.1. Определение степени независимости лиц, проводящих верификацию.

2.2. Для каждого из этапов верификации: определение перечня выполняемых задач и применяемых для этого методик.

2.3. Определение мероприятий по устранению недостатков, выявленных в процессе верификации.

2.4. Определение мероприятий по документированию процесса верификации.

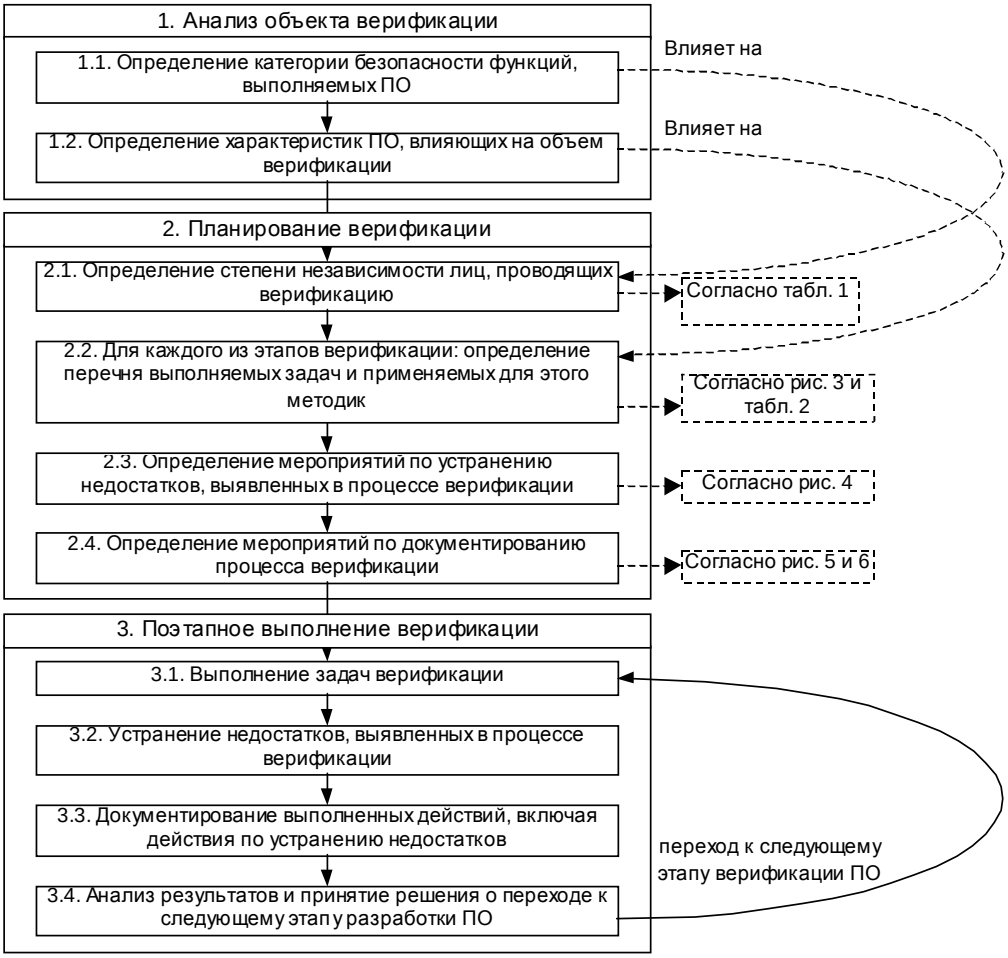

Рис. 2. Структура метода комплексной верификации ПО ИУС ТККИ

3. Выполнение верификации (указанные ниже действия выполняются для каждого из этапов верификации).

3.1. Выполнение задач верификации.

3.2. Устранение недостатков, выявленных в процессе верификации.

3.3. Документирование выполненных действий, включая действия по устранению недостатков.

3.4. Анализ результатов и принятие решения о переходе к следующему этапу разработки ПО.

Структура метода комплексной верификации ПО ИУС ТККИ приведена на рис. 2. Кроме приведенной выше последовательности действий, на рис. 2 указаны логические связи между свойствами верифицируемого ПО и составляющими процесса верификации. Выполнение действий по верификации ПО детализировано в последующих разделах статьи (ссылки даны на рис. 2). Выполнение действий верификации должно строго соответствовать плану, всякое отклонение от плана должно быть обосновано и документировано.

3.1. Определение степени независимости лиц, проводящих верификацию

Перед проведением верификации должна быть определена степень независимости лиц, проводящих верификацию, от лиц, выполнявших разработку ПО. В табл. 1 приведены требования к независимости верификации для разных уровней безопасности компьютерных систем согласно стандарту МЭК 61508-1:1998 «Функциональная безопасность электрических, электронных и программируемых систем, важных для безопасности – Часть 1: Общие требования». Согласно табл. 1, должна быть определена степень независимости специалистов, выполняющих верификацию ПО ИУС ТККИ.

Таблица 1. Соответствие между категорией безопасности, уровнем интегрированности системы и требованиями к независимости верификации

Категория безопасности ТККИ	Последствия аварии ТККИ	Уровень интегрированности системы	Требуемая интенсивность отказов	Требования к независимости верификации
D	Незначительный ущерб без риска для людей	1	$10^{-6} \div 10^{-5}$ 1/час	Специалисты, не участвовавшие в разработке
C	Серьезная угроза здоровью и жизни нескольких человек	2	$10^{-7} \div 10^{-6}$ 1/час	Подразделение, независимое от разработчиков
B	Гибель нескольких человек	3	$10^{-8} \div 10^{-7}$ 1/час	Организация, независимая от организации-разработчика, или подразделение, независимое от разработчиков (при наличии в организации специализированного подразделения или филиала)
A	Гибель большого количества людей	4	$10^{-9} \div 10^{-8}$ 1/час	Организация, независимая от организации-разработчика

3.2. Этапы верификации ПО

Структура жизненного цикла ПО представлена на рис. 3. Согласно требованиям к процессам жизненного цикла ПО, процесс должен завершаться верификацией соответствующего продукта. Процесс верификации ПО включает следующие этапы [6, 9]:

– верификация требований к программному обеспечению;

– верификация проекта программного обеспечения;

– верификация кода программных модулей;

– верификация интегрированного программного обеспечения.

3.3. Определение перечня действий, выполняемых для этапов верификации и применяемых для этого методик

Соответствие между этапами верификации ПО, выполняемыми на этих этапах задачами и применяемыми для выполнения задач методиками [8, 9] установлено в табл. 2. Следует отметить, что для реализации указанных методик целесообразно применять апробированные инструментальные средства.

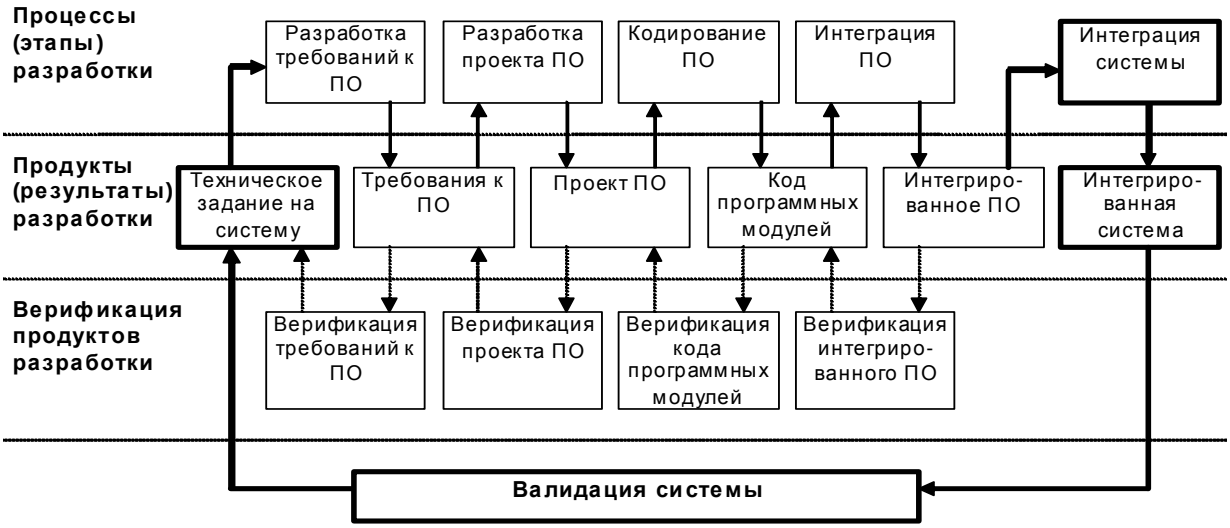

Рис. 3. Структура процессов верификации и валидации программного обеспечения

Таблица 2. Соответствие между этапами верификации ПО, выполняемыми задачами и применяемыми методиками

Этап верификации	Задача верификации	Методика верификации
Верификация требований к ПО	Оценка полноты и корректности требований	Технический обзор документации
	Оценка соответствия техническому заданию на систему	Анализ трассируемости
Верификация проекта ПО	Оценка полноты и корректности проекта	Технический обзор документации
	Оценка соответствия требованиям к ПО	Анализ трассируемости
Верификация кода программных модулей	Оценка полноты и корректности программного кода	Статический анализ
	Автономное тестирование программных модулей	Структурное тестирование по принципу «белого ящика»
	Оценка полноты тестовых проверок	Сквозной просмотр документации
	Оценка соответствия проекту ПО	Анализ трассируемости
Верификация интегрированного ПО	Комплексное тестирование интегрированного ПО	Функциональное тестирование по принципу «черного ящика»
	Оценка полноты тестовых проверок	Сквозной просмотр документации
	Оценка соответствия интегрированного ПО проекту и требованиям	Анализ трассируемости

3.4. Определение мероприятий по устранению недостатков, выявленных в процессе верификации

Все обнаруженные в ходе верификации недостатки должны быть зафиксированы, проанализированы и устранены, после чего необходимо провести повторную проверку ПО. В ходе рассмотрения недостатков они должны быть задокументированы в отчете по верификации ПО, включая следующую информацию:

– каким образом и когда был обнаружен недостаток;

– как недостаток был проанализирован;

– какие требовались изменения для устранения недостатка и какие мероприятия были выполнены;

– на какие из элементов ПО повлияли внесенные изменения и какие версии были затронуты.

Формальное описание процесса устранения недостатков (рис. 4) приведено в стандарте IEEE 1044-1993 «IEEE Standard Classification for Software Anomalies» и включает следующие шаги:

1) обнаружение;

2) анализ;

3) устранение;

4) закрытие проблемы.

Каждый из шагов включает следующие три действия:

1) документирование;

2) классификация (с точки зрения выполняемых шагов);

3) определение последствий.

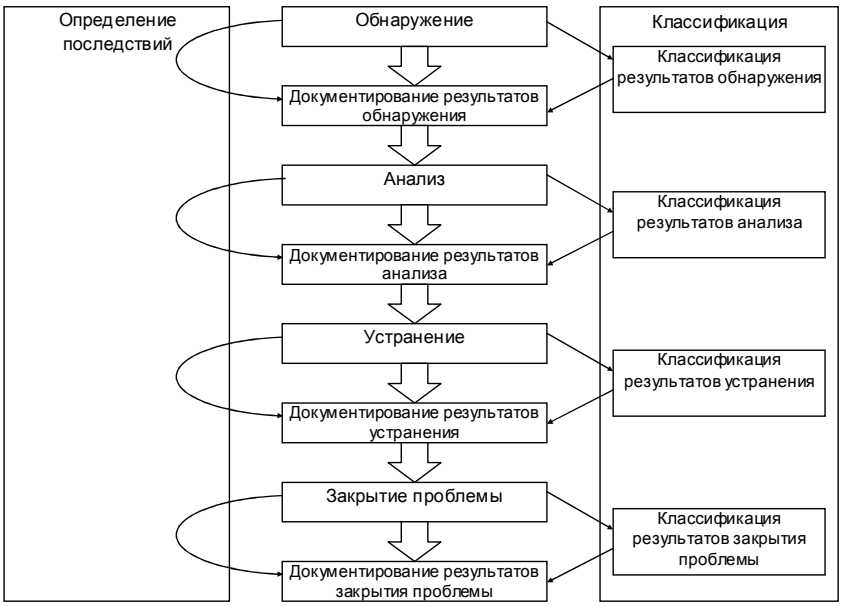

Рис. 4. Структура мероприятий по устранению недостатков, выявленных в процессе верификации

3.5. Определение мероприятий по документированию процесса верификации

Процесс верификации ПО должен быть документирован. Перед началом верификации должен быть выпущен план верификации ПО (рис. 5). Отдельные части плана верификации ПО, имеющие

самостоятельное значение (программы, методики испытаний, планы тестирования и т.п.), могут быть выпущены в виде отдельных документов. Если отдельные части или функции ПО могут быть проверены только в составе системы (в процессе валидации), то это должно быть обосновано и отражено в плане верификации.

По результатам верификации должен быть выпущен отчет по верификации ПО (рис. 6). Отдельные части отчета по верификации ПО, имеющие самостоятельное значение (отчеты по испытаниям, журналы испытаний, протоколы испытаний и т.п.), могут быть выпущены в виде отдельных документов.

Вся документация по разработке и верификации ПО должна быть изложена в доступной форме, понятной специалистам, не участвовавшим в проведении разработки и верификации ПО.

Введение
1. Объект верификации
2. Цель верификации
3. Исходные и нормативные документы
4. Термины и определения
4.1. Принятые сокращения
4.2. Основные термины и определения
5. Стратегия и организация верификации
5.1. Стратегия верификации
5.2. Организация процесса верификации
5.3. Участники верификации и распределение ответственности
6. Методики и средства верификации
6.1. Методики тестирования и анализа
6.2. Инструментальные средства верификации
7. Порядок проведения этапов верификации
8. Отчетность по верификации
8.1.Отчетные документы
8.2. Оценка результатов верификации
9. Администрирование процесса верификации

Рис. 5. Структура плана верификации ПО согласно требованиям стандарта IEEE 1012-1998 «Верификация и валидация программного обеспечения»

Введение
1. Объект верификации
2. Объем и цель верификации
3. Методы и средства, использовавшиеся при верификации
4. Порядок и особенности проведения верификации
5. Отчеты по верификации
5.1. Структура отчетов по верификации
5.2. Соответствие отчетов методикам верификации
6. Анализ результатов верификации
6.1. Результаты испытаний
6.2. Анализ недостатков и принятые меры
6.3. Результаты повторных испытаний
7. Общие выводы по верификации

Рис. 6. Структура отчета по верификации ПО согласно требованиям стандарта IEEE 1012-1998 «Верификация и валидация программного обеспечения»

4. Выводы

В результате проведенных исследований выполнено структурирование процесса верификации ПО. При проведении верификации ПО должны быть учтены следующие аспекты:

– свойства ПО;

– независимость процесса верификации;

– структура жизненного цикла ПО, включая этапы разработки и верификации;

– задачи верификации и используемые для их выполнения методики;

– выявление и устранение недостатков;

– документирование процесса верификации;

– оценка и сертификация процесса верификации.

Разработанный метод комплексной верификации ПО основан на учете требований стандартов по безопасности, интегрирует этапы верификации ПО и их атрибуты, включая задействованный персонал, задачи, методики, устранение недостатков и выпускаемую документацию. Данный метод включает набор действий по анализу объекта верификации, планированию верификации, а также поэтапному проведению верификации.

Действенность метода комплексной верификации ПО была подтверждена в ходе работ по выполнению экспертной оценки информационных и управляющих систем, важных для безопасности АЭС [8].

Дальнейшие исследования целесообразно проводить в следующих направлениях:

– повышение полноты, глубины и достоверности процесса верификации за счет использования инструментальных средств и поддерживаемых ими формальных методов [9];

– реализация процессов анализа и выбора инструментальных средств и формальных методов с точки зрения их применимости для верификации ПО;

– исследование свойств объекта верификации с точки зрения дифференциации методов по типам программируемых компонент, например, для ПЛИС, программируемых логических контроллеров (ПЛК), встроенных микропроцессоров интеллектуальной периферии и т.д.

– сравнительный анализ и адаптация метода комплексной верификации ПО для различных критических отраслей, включая связь, транспорт, комплексы вооружений и т.д.

СПИСОК ЛИТЕРАТУРЫ

1. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. – М.: СИНТЕГ, 2001. – 380 с.
2. Скляр В.В., Харченко В.С., Ястребенецкий М.А. Цифровые информационные и управляющие системы атомных электростанций и ракетно-космических комплексов: Сравнительный анализ, тенденции развития, обеспечение безопасности // Ядерная и радиационная безопасность. – 2004. – Т. 7, № 2. – С. 35–41.
3. Харченко В.С., Скляр В.В., Тарасюк О.М. Безопасность аэрокосмической техники и надежность компьютерных систем // Авиационно-космическая техника и технология. – 2004. – № 1(9). – С.66–80.
4. Lyu M.R. Handbook of Software Reliability Engineering. – McGraw-Hill Company, 1996. – 805 p.
5. Leveson N. Safeware: System Safety and Computers. – Addison-Wesley, 1995. – 431 p.
6. Харченко В.С., Скляр В.В., Тарасюк О.М. Методы моделирования и оценки качества и надежности программного обеспечения. –Харьков: Национальный аэрокосмический ун-т «Харьк. авиац. ин-т», 2004.–159 с.
7. Смит Д., Симпсон К. Функциональная безопасность. Простое руководство по применению стандарта МЭК 61508 и связанных с ним стандартов. – М.: Издательский Дом «Технологии», 2004. – 208 с.
8. Безопасность атомных станций: Информационные и управляющие системы / Ястребенецкий М.А., Васильченко В.Н., Виноградская С.В. и др. – К.: Техніка, 2004. – 472 с.
9. Скляр В.В. Инструментальные средства для статического анализа программного обеспечения: принципы применения, оценки и выбора // Электронное моделирование. – 2006. – Т. 28, № 2. – С. 29–41.