Государственная корпорация по атомной энергии «Росатом»

Акционерное общество   
«Ордена Ленина Научно-исследовательский и конструкторский институт энерготехники имени Н.А. Доллежаля»   
(АО «НИКИЭТ»)

КОМПЛЕКС РАСПРЕДЕЛЕННЫХ СРЕДСТВ СЕТЕВОЙ ОБРАБОТКИ (КРОСС) **Программный комплекс КРОСС**

**План верификации**

RU.589.08624303.00100-01 91 01

**(на магнитном носителе)**

**Листов 49**

**АННОТАЦИЯ**

Данный документ содержит план верификации программного обеспечения программного комплекса КРОСС.

**СОДЕРЖАНИЕ**

[1. Введение 5](#_Toc88822551)

[2. Объект верификации 6](#_Toc88822552)

[3. Стратегия и организация верификации 15](#_Toc88822553)

[3.1. Стратегия верификации 15](#_Toc88822554)

[3.2. Организация верификации 17](#_Toc88822555)

[3.3. Участники верификации и распределение ответственности 19](#_Toc88822556)

[4. Методы и средства верификации 20](#_Toc88822557)

[4.1. Методы анализа и просмотра 20](#_Toc88822558)

[4.2. Методы тестирования 21](#_Toc88822559)

[4.2.1. Тестирование методом «белого ящика» 22](#_Toc88822560)

[4.2.2. Тестирование методом «черного ящика» 22](#_Toc88822561)

[4.3. Инструментальные средства разработки и верификации ПО 24](#_Toc88822562)

[5. Порядок проведения этапов верификации 27](#_Toc88822563)

[5.1. Этап 1. Верификация требований к ПО КРОСС 28](#_Toc88822564)

[5.1.1. Задачи 28](#_Toc88822565)

[5.1.2. Методы и критерии 28](#_Toc88822566)

[5.1.3. Входы и выходы 29](#_Toc88822567)

[5.2. Этап 2. Верификация проекта ПО КРОСС 30](#_Toc88822568)

[5.2.1. Задачи 30](#_Toc88822569)

[5.2.2. Методы и критерии 30](#_Toc88822570)

[5.2.3. Входы и выходы 31](#_Toc88822571)

[5.3. Этап 3. Верификация исходных кодов ПО КРОСС 32](#_Toc88822572)

[5.3.1. Задачи 32](#_Toc88822573)

[5.3.2. Методы и критерии 32](#_Toc88822574)

[5.3.3. Входы и выходы 34](#_Toc88822575)

[5.4. Этап 4. Функциональное тестирование и верификация результатов тестирования ПО КРОСС 34](#_Toc88822576)

[5.4.1. Задачи 34](#_Toc88822577)

[5.4.2. Методы и критерии 35](#_Toc88822578)

[5.4.3. Входы и выходы 36](#_Toc88822579)

[6. Отчетность по верификации 38](#_Toc88822580)

[7. Анализ результатов верификации 41](#_Toc88822581)

[7.1. Анализ результатов верификации требований к ПО 41](#_Toc88822582)

[7.2. Анализ результатов верификации требований к ПО 42](#_Toc88822583)

[7.3. Анализ результатов верификации исходных кодов 43](#_Toc88822584)

[7.4. Анализ результатов верификации на этапе функционального тестирования 44](#_Toc88822585)

[Перечень сокращений 47](#_Toc88822586)

[Перечень ссылочных документов 48](#_Toc88822587)

# **Введение**

Настоящий план предназначен для определения стратегии и организации мероприятий по верификации, необходимых для достижения требуемого качества при разработке программного обеспечения, комплекс распределенных средств сетевой обработки (краткое название КРОСС). ПО КРОСС - программный комплекс, предназначенный для создания на его основе автоматизированных систем контроля и управления технологическим процессом, программно-технических комплексов вывода информации и диагностики систем безопасности, устройств комплексной функциональной проверки оборудования и испытательных стендов.

Определяемые данным планом мероприятия по верификации программного обеспечения КРОСС должны обеспечить как реализацию всех требований, предъявляемых к программному обеспечению, так и выявление и устранение всех несоответствий на ранних стадиях проектирования.

План разработан в соответствии с требованиями ГОСТ Р МЭК 61513 [1], ГОСТ Р МЭК 62138 [2] и устанавливает объем, методы и средства проверки, критерии оценки соответствия характеристик и функций программного обеспечения КРОСС.

## 

# **Объект верификации**

Объектом верификации по настоящему плану является программный комплекс КРОСС, который верифицируется в процессе разработки, а затем в процессе сопровождения.

КРОСС должен состоять из нескольких отдельных программных комплексов, что должно позволить решать практически любые задачи верхнего уровня (ВУ) автоматизированной системы управления технологическим процессом (АСУ ТП).

Программный комплекс должен иметь модульную, конфигурируемую структуру и развитые инструментальные средства разработки, что определяет широкую область применения и высокую скорость разработки.

Программный комплекс КРОСС должен базироваться на концепциях, направленных на повышение надежности выполнения заявленных функций:

* применение дублированной или многократно резервированной отказоустойчивой структуры;
* наличие развитых средств самодиагностики;
* разделение каналов обмена информацией на каналы передачи данных с детерминированной нагрузкой и вспомогательные каналы;
* детерминированное выполнение функций межзадачного и сетевого обмена.

Для достижения устойчивости к резкому увеличению нагрузки при аварийных и переходных режимах объекта управления, в программном комплексе должны быть реализованы:

* распределение функций обработки;
* независимая работа источников и потребителей информации;
* широковещательный режим передачи данных.

Данные решения должны позволить использовать программный комплекс КРОСС в качестве основы для построения автоматизированных систем контроля и управления объектами со свойствами повышенной опасности.

Программный комплекс КРОСС должен выполнять следующие функции:

* прием, предварительная обработка, проверка достоверности и хранение данных о технологических параметрах;
* проведение оперативных и неоперативных расчетов параметров технологического процесса;
* передача данных и команд в системы контроля и управления нижнего уровня;
* отображение измеренных и расчетных параметров;
* обмен информацией с внешними системами;
* сигнализация отклонений значений технологических параметров от разрешенных пределов;
* регистрация и архивирование технологической информации;
* автоматизированное формирование и печать твердых копий выходных документов системы;
* оперативная диагностика выполнения основных функций и состояния оборудования.

Программный комплекс КРОСС должен состоять из следующих комплексов:

* комплекс программного обеспечения распределенной базы данных реального времени;
* комплекс системного программного обеспечения функций архивирования;
* комплекс интерфейсного программного обеспечения;
* комплекс системного программного обеспечения функции отображения на рабочей станции (РС);
* комплекс системного программного обеспечения сервера вычислений;
* комплекс специального программного обеспечения функциональных проверок;
* комплекс программного обеспечения генерации и печати отчётов;
* комплекс программного обеспечения вспомогательных функций.

Программные комплексы и их компоненты должны иметь возможность их распределения по нескольким отдельным компьютерам (контроллерам), связанным общими либо выделенными каналами передачи данных (сетями данных и управления), далее называемым «узлы КРОСС». Должна быть предусмотрена возможность отдельной настройки и конфигурирования каждой компоненты комплекса для каждого узла системы, включая полное резервирование функций.

Комплекс программного обеспечения распределенной базы данных реального времени должен выполнять функции:

* создание и поддержка базы данных с информацией о технологических параметрах в оперативной памяти на каждом из узловкомплекса КРОСС;
* приема из информационной сети, предварительной обработки и проверки достоверности данных о технологических параметрах;
* передача данных и команд по информационной сети между узлами КРОСС.

В состав комплекса должны быть включены модули:

* модуль распределенной базы данных реального времени (РБДРВ);
* модуль асинхронного доступа к данным для программного комплекса КРОСС;
* модуль удаленного доступа к информации РБДРВ (сервер).

Комплекс системного программного обеспечения функций архивирования должен выполнять функции:

* архивирование технологической информации;
* экспорт архивной информации для использования сторонними по отношению к КРОСС системами и для долговременного хранения на сменных носителях;
* предоставление архивной информации пользователям.

В состав комплекса должны быть включены модули:

* модуль архивирования в РБДРВ;
* модуль ведения временных архивов;
* модуль экспорта архивов в текстовые и бинарные форматы;
* модуль отображения значений архивных параметров в виде графиков;
* модуль предоставления архивных данных в текстовом виде;
* модуль пошагового доступа к архивам;
* модуль проигрывания архивов;
* модуль доступа к архивным данным.

Комплекс интерфейсного программного обеспечения должен выполнять функции:

* обмен информацией с внешними системами;
* обмен информацией с «низовой» измерительной и исполнительной аппаратурой.

В составе комплекса должны быть, как минимум, разработаны следующие модули:

* модуль программного обеспечения шлюза с СЦК «СКАЛА-МИКРО»;
* модуль обмена по последовательному интерфейсу;
* модуль обмена по интерфейсу ТПТС;
* модуль обмена по интерфейсу MODBUS;
* модуль обеспечения работы с картами ввода/вывода.

Комплекс системного программного обеспечения функции отображения на рабочей станции (РС) должен выполнять функции:

* предоставление пользователю информации, содержащейся в распределенной базе данных технологических параметров КРОСС;
* предоставление информации о работоспособности аппаратных средств системы;
* предоставление информации о функционировании программных элементов комплекса.

В состав комплекса должны быть включены модули:

* модуль отображения текущей информации о технологических параметрах;
* модуль разработки и отображения видеокадров;
* модуль отображения списков сигнализаций;
* модуль отображения сообщений о ходе технологического процесса;
* модуль отображения графиков технологических параметров в текущем времени.

Комплекс системного программного обеспечения сервера вычислений должен выполнять функции:

* выполнение расчетных задач стандартных алгоритмов;
* выполнение прикладных расчетных задач.

В состав комплекса должны быть включены модули:

* модуль сторожевого таймера;
* модуль стандартных алгоритмов обработки;
* модуль прикладных программ;
* модуль системной библиотеки.

Комплекс специального программного обеспечения функциональных проверок должен выполнять функции:

* имитация с нормированной точностью электрических сигналов датчиков технологических параметров, соответствующих различным режимам работы и передачи этих сигналов в аппаратуру;
* прием, отображение и архивирование результатов измерений, вычислений и проверок, проводимых с применением комплекса.

В состав комплекса должны быть включены модули:

* модуль имитации электрических сигналов;
* модуль приема и отображения тестовых сигналов.

Комплекс программного обеспечения генерации и печати отчётов должен выполнять функции:

* формирование шаблонов отчетов;
* редактирование шаблонов отчетов;
* загрузка шаблонов отчетов на сервер генерации и печати отчетов;
* генерация отчетов по заданным шаблонам;
* печать готовых отчетов;
* задание и управление расписанием генерации отчетов;
* задание событий для генерации отчетов.

В состав комплекса должны быть включены модули:

* модуль генерации отчетов;
* модуль менеджера отчетов.

Комплекс программного обеспечения вспомогательных функций должен выполнять функции:

* проведение оперативной диагностики выполнения основных функций и состояния оборудования;
* поддержания единого времени для всех рабочих станций и серверов комплекса технических средств КРОСС, объединенных в локальную вычислительную сеть;
* предоставления обслуживающему персоналу возможности в интерактивном режиме просматривать и изменять любые поля в РБДРВ на любой рабочей станции.

В состав комплекса должны быть включены модули:

* модуль синхронизации времени;
* модуль диагностики;
* модуль консольных утилит для работы с РБДРВ.

ПО КРОСС участвует в выполнении функций категории В согласно   
ГОСТ Р МЭК 61226 [3].

# **Стратегия и организация верификации**

## **Стратегия верификации**

Целями верификации ПО КРОСС являются:

* создание высоконадежного ПО за счет наиболее раннего выявления ошибок проектирования и кодирования ПО, обнаружения потенциально опасных и (или) нестабильных в работе проектных решений и способов кодирования ПО;
* доказательство того, что результат разработки ПО соответствует, предъявленным к нему требованиям.

При выборе стратегии верификации ПО КРОСС учитывались требования ГОСТ Р МЭК 62138 [2], который устанавливает требования к процессу верификации ПО, выполняющего функции категорий В.

Процесс верификации проводится сверху вниз, начиная от общих требований, заданных в техническом задании. Цели верификации ПО достигаются посредством последовательного выполнения следующих мероприятий:

* просмотр и анализ требований к ПО - цель просмотра и анализа обнаружить и зарегистрировать ошибки, которые могли быть внесены в процессе разработки требований к ПО;
* просмотр и анализ проекта ПО - цель просмотра и анализа обнаружить и зарегистрировать ошибки, которые могли быть внесены во время разработки проекта ПО и что проект ПО соответствует требованиям к ПО;
* просмотр и анализ исходного кода - цель этих просмотра и анализа выявление и регистрация ошибок, которые могли быть внесены в процессе кодирования ПО. Просмотр и анализ подтверждают, что выходные результаты кодирования являются точными, полными и могут быть верифицированы. Прежде всего проверяются корректность кода по отношению к требованиям к ПО и архитектуре ПО и соответствие стандартам на кодирование;
* функциональное тестирование ПО КРОСС - тестирование проводится методом «черного ящика». В ходе данного тестирования проверяются все функциональные требования системы. Функциональное тестирование нацелено на выявление следующих ошибок: неверная или пропущенная функциональность, ошибки интерфейса, ошибки в структурах данных или ошибки доступа к базам данных, проблемы снижения производительности, ошибки загрузки, ошибки многопользовательского доступа, ошибки инициализации и завершения, проблемы сохранения резервных копий и способности к восстановлению работы, проблемы безопасности;
* анализ тестового покрытия – в ходе анализа должно быть подтверждено, что тестовые наборы основаны на требованиях к ПО, что все требования к ПО были протестированы. Должны быть разработаны тестовые наборы для проверки функционирования в области допустимых значений и для проверки на устойчивость к ошибкам входных данных. Так же должны быть разработаны тестовые наборы для проверки устойчивости ПО при возникновении ошибок и возникновении неисправности ТС.

На всех этапах разработки ПО КРОСС должна осуществляться проверка согласованности результатов, полученных на каждом этапе разработки с требованиями, установленными на предыдущих этапах, и получение данных, обеспечивающих анализ и оценку показателей качества реализации требований.

## **Организация верификации**

Верификация должна проводиться компетентными лицами, не принимавшими участие в разработке ПО КРОСС.

Верификация ПО должна проводиться на каждом этапе жизненного цикла ПО в соответствии с данным Планом.

Все обнаруженные в ходе верификации несоответствия должны быть устранены до начала следующего этапа разработки ПО.

Верификация ПО должна проводиться параллельно с процессом разработки с тем, чтобы можно было своевременно зарегистрировать и исправить обнаруженные ошибки.

Каждый этап верификации должен завершаться оформлением отчета, в котором должны быть указаны:

* объект верификации;
* оцениваемые показатели;
* порядок проведения проверки (методика верификации данного этапа, выполняемые действия, контролируемые параметры и данные);
* используемые программно-технические средства;
* критерии принятия решения, о соответствии проверяемого ПО и (или) документации предъявленным требованиям;
* результаты проверки ПО и (или) документации с указанием соответствия (несоответствия) предъявленным требованиям;
* выводы об успешном выполнении этапа верификации, либо о необходимости доработки ПО и (или) документации и проведении повторной верификации, включая, при необходимости, предыдущие этапы;
* перечень выявленных ошибок и недостатков (при обнаружении таковых), которые должны быть устранены на данном этапе проектирования, оформленный в виде приложения к протоколу верификации;
* перечень выявленных ошибок и недостатков (при обнаружении таковых), которые затрагивают предыдущие этапы и требуют проведения повторной верификации, а также перечень функций, верификация которых невозможна на данном этапе и переносится на последующие этапы, оформленные в виде приложений к протоколу верификации;
* заключение о возможности перехода к верификации следующего этапа.

Если при устранении несоответствия требуется корректировка, то откорректированный вариант документа или программного продукта должен быть повторно предъявлен на верификацию.

Верификация следующего этапа возможна только после устранения всех обнаруженных несоответствий и проведения повторной верификации измененных фрагментов ПО.

## **Участники верификации и распределение ответственности**

Верификацию ПО КРОСС проводит верификационная группа из числа сотрудников отдела компьютерных и интегрированных систем.

Верификацию требований к ПО КРОСС проводят сотрудники отдела компьютерных и интегрированных систем не участвующие в разработке ТЗ на ПО КРОСС.

Верификацию проекта ПО КРОСС проводят сотрудники отдела компьютерных и интегрированных систем не участвующие в разработке проекта ПО КРОСС.

Верификацию исходных кодов ПО КРОСС проводят сотрудники отдела компьютерных и интегрированных систем не участвующие в кодировании ПО КРОСС.

Верификацию результатов функционального тестирования ПО КРОСС проводят сотрудники отдела компьютерных и интегрированных систем не участвующие в разработке проекта ПО КРОСС и не участвующие в кодировании ПО КРОСС.

# **Методы и средства верификации**

## **Методы анализа и просмотра**

Методы анализа и просмотра (обзора) должны применяться для верификации проектной документации, исходных кодов и тестового покрытия. Анализ должен обеспечить доказательство корректности разрабатываемых компонентов ПО и детально исследовать функциональность, эффективность, прослеживаемость и надежность компонентов ПО.

Анализ трассируемости должен подтвердить полноту реализации на текущем этапе всех требований предыдущего этапа разработки.

Просмотр (обзор) должен обеспечить качественную оценку степени корректности компонентов ПО и может включать методы инспекции и проверки. Метод инспекции заключается в рассмотрении документации на предмет отсутствия ошибок и аномалий. Метод проверки – в установлении соответствия документации, разработанной на предыдущем этапе проектирования.

Просмотры и анализы должны проводиться сверху вниз, начиная от общих требований до детальных требований на модули и их взаимодействие.

Просмотры и анализы должны обеспечивать оценку точности, полноты и верифицируемости требований, архитектуры ПО, а также исходных кодов программ.

## **Методы тестирования**

Методы тестирования должны применяться для верификации ПО на этапах автономного и функционального тестирования.

Реализация тестирования делится на три этапа:

1. Создание тестового набора путем ручной разработки или автоматической генерации для конкретной среды тестирования.
2. Прогон программы на тестах с оформлением протокола тестирования.
3. Оценка результатов выполнения программы на наборе тестов с целью принятия решения о продолжении или остановке тестирования.

Тестирование должно обеспечить проверку внутренней непротиворечивости и полноты реализации требований и подтвердить соответствие испытываемых компонентов ПО исходным функциональным требованиям, а также выявить все несоответствия и различия между ожидаемыми и полученными результатами для последующего исследования и устранения.

Помимо тестирования общего функционирования должно быть предусмотрено стрессовое тестирование для критических ситуаций (вне области допустимых требованиями входных данных), важных для обеспечения качества и надежности функционирования ПО КРОСС.

### Тестирование методом «белого ящика»

Тестирование методом «белого ящика» включает в себя стратегию модульного тестирования, при котором тестирование ведется на модульном уровне и работы по тестированию направлены на исследование внутреннего устройства модуля. На этом уровне тестирования проверяется управляющая логика, проявляющаяся на модульном уровне. В ходе модульного тестирования каждый модуль тестируется как на соответствие требованиям, так и на отсутствие проблемных участков программного кода, могущих вызвать отказы и сбои в работе ПО. Тестовые наборы разрабатываются для того, чтобы все ветви в данном модуле были проверены хотя бы один раз, все логические решения рассмотрены во всевозможных условиях, циклы были выполнены с использованием верхних и нижних границ и проконтролированы структуры внутренних данных. Проверяется обработка как верных, так и неверных входных данных, а так же выдача сообщений об ошибках.

### Тестирование методом «черного ящика»

Тестирование «чёрного ящика»  это метод тестирования функционального поведения программы с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из требований к ПО. При тестировании функционального поведения используются следующие методы:

1. Эквивалентное разбиение. Тестирование с использованием подмножества входных данных. Применяют три основных типа эквивалентных классов: значения внутри границы диапазона, за границей диапазона и на границе.
2. Функциональное тестирование. Функциональное тестирование проверяет ПО в отношении функциональных требований с целью обнаружения несоответствия требованиям к ПО.
3. Тестирование безопасности. Тестирование безопасности включает в себя проверку работы механизмов доступа к системе и к данным. Для этого разрабатываются тестовые процедуры, которые пытаются преодолеть защиту системы. Во время тестирования проверяется степень безопасности и ограничения доступа, выявляя таким образом соответствие установленным требованиям к безопасности и всем применяемым правилам по безопасности ПО.
4. Тестирование перегрузок. При тестировании перегрузок выполняется проверка ПО без учета ограничений архитектуры с целью выявления технических ограничений ПО. Эти тесты проводятся на пике обработки транзакций и при непрерывной загрузке большого объема данных. Тестирование перегрузок измеряет пропускную способность ПО и его эластичность на всех аппаратных платформах. От системы требуется обработка огромного количества данных или выполнение большого числа функциональных запросов в течение короткого периода времени.
5. Тестирование производительности. Тесты производительности проверяют, удовлетворяет ли программное приложение требованиям по производительности. Применяя тестирование производительности, можно замерить и составить отчеты по таким показателям, как скорость передачи входных и выходных данных, общее число действий по вводу и выводу данных, среднее время, затрачиваемое базой данных на отклик на запрос, и интенсивность использования центрального процессора.

## **Инструментальные средства разработки и верификации ПО**

Инструментальные средства разработки ПО КРОСС (САПР, транслятор, компилятор) должно использоваться из числа ранее применяемых на предприятии для аналогичного ПО.

Инструментальные средства верификации (тестовое ПО, стенды проверки блоков связи с объектом) должно использоваться из числа проверенных и уже примененных на предприятии.

Инструментальные средства тестирования (тестовое ПО, макеты или имитаторы) должно использоваться из числа проверенных и уже примененных на предприятии для тестирования подобного ПО. При необходимости инструментальные средства тестирования должны быть модифицированы по отдельным требованиям и проверены.

Используемые для подтверждения правильности функционирования аппаратные и программные средства специальной верификации не подвергаются. Документально должно подтверждаться, что программные и аппаратные средства соответствуют своему назначению.

Структура и состав средств, используемых в процессе верификации должны быть уточнены в процессе проведения верификации каждого компонента ПО и приведены в соответствующих программах и методиках.

Список инструментального ПО используемого при разработке ПО КРОСС приведен в табл. 1.

Т а б л и ц а 1

| Название ПО | Версия | Описание |
| --- | --- | --- |
| Разработка ПО КРОСС | | |
| GCC | 4.7.2 | Компилятор С/С++ |
| GNU Make | 3.81 | Программа управления сборкой |
| QMake | 2.01а | Программа управления сборкой |
| Qt | 4.8.7 | Базовые библиотеки |
| Qt Creator | 3.6.0 | Среда разработки |
| Eclipse-CDT | 4.5.2 | Среда разработки |
| Тестирование ПО КРОСС | | |
| Cppcheck | 1.71 | Статический анализатор кода для языка C/C++ |

*Окончание таблицы 1*

| Название ПО | Версия | Описание |
| --- | --- | --- |
| Valgrind | 3.11.0 | Отладчик использования памяти |
| gcov | 4.7.2 | Анализатор покрытия кода |
| GNU gprof | 2.22 | Профайлер |
| Среда контроля версий ПО КРОСС | | |
| SCM Mercurial | 2.2.2 | Среда контроля версий |

# **Порядок проведения этапов верификации**

На основании рекомендаций ГОСТ Р ИСО/МЭК 12207 [4] для разработки программного обеспечения КРОСС в соответствии с V-моделью жизненного цикла программных средств, жизненный цикл разрабатываемого ПО должен разбиваться на следующие этапы:

* разработка требований к ПО КРОСС;
* разработка проекта ПО КРОСС;
* кодирование;
* функциональное тестирование.

Работы по верификации программного обеспечения КРОСС, разделяются на следующие этапы:

* этап 1 – верификация требований к ПО КРОСС;
* этап 2 – верификация проекта ПО КРОСС;
* этап 3 – верификация исходных кодов ПО КРОСС;
* этап 4 – функциональное тестирование и верификация результатов тестирования ПО КРОСС.

## **Этап 1. Верификация требований к ПО КРОСС**

### Задачи

Установление соответствия требований ТЗ на ПО КРОСС требованиям нормативных документов.

Подтверждение что функции, выполняемые ПО КРОСС определены, функциональные, технические требования и требования по безопасности описаны.

Обнаружение ошибок в требованиях, которые могли быть допущены в процессе разработки требований к ПО КРОСС.

Регистрация обнаруженных ошибок.

Проверка корректирующих действий проводимых по исправлению обнаруженных ошибок.

Проверка внесенных исправлений в ТЗ на ПО КРОСС, в случае необходимости.

### Методы и критерии

Методом проведения процедур верификации на данном этапе является экспертная оценка формулируемых целей и задач исходя из нормативных документов, выбранных для проведения процесса верификации.

В качестве критерия для проводимых оценок выступает требование максимального удовлетворения рекомендациям по проведению процедур верификации, изложенным в стандарте ГОСТ Р МЭК 62138 [2].

В ходе верификации требований ТЗ на ПО КРОСС методом просмотра и анализа проверяется:

* отсутствие противоречий между требованиями ТЗ на ПО КРОСС, предъявляемыми к разра­батываемому ПО, и требованиями действующих НД;
* отсутствие противоречий между отдельными требованиями ТЗ на ПО КРОСС;
* полнота требований ТЗ на ПО КРОСС – оценивается наличие требований для всех режимов работы ПО и наличие требований по взаимодействию с внешними программными средствами;
* точность требований ТЗ на ПО КРОСС – оценивается четкость формулировок требований ТЗ на ПО, не допускающая их неоднозначную трактовку;
* верифицируемость требований ТЗ на ПО – оценивается, что каждое требование может быть верифицировано;
* реализуемость требований ТЗ на ПО - оценивается, что каждое требование может быть реализуемо.

### Входы и выходы

Входными данными процесса верификации на этапе верификации требований к ПО КРОСС являются:

* план верификации программного обеспечения КРОСС;
* техническое задание.

Выходные результаты процесса верификации на этапе верификации требований к ПО КРОСС содержатся в следующих документах:

* отчет по верификации требований к программному обеспечению КРОСС;
* запросы на изменение, сформированные в ходе верификации требований к ПО КРОСС.

## **Этап 2. Верификация проекта ПО КРОСС**

### Задачи

Обнаружение ошибок в проекте ПО КРОСС которые могут быть допущены в процессе проектирования ПО.

Регистрация обнаруженных ошибок.

Проверка корректирующих действий проводимых по исправлению обнаруженных ошибок.

Проверка внесенных исправлений в проект ПО и в ТЗ на ПО КРОСС, в случае необходимости.

### Методы и критерии

В ходе верификации проекта ПО КРОСС методом просмотра и анализа проверяется:

* отсутствие несоответствий между требованиями ТЗ на ПО КРОСС и требованиями проекта ПО;
* отсутствие противоречий между отдельными требованиями проекта ПО КРОСС;
* точность требований проекта ПО КРОСС – оценивается четкость формулировок требований проекта ПО, не допускающая их неоднозначную трактовку;
* совместимость с объектным компьютером: гарантировать, что не существует никаких противоречий между архитектурой ПО и программно-аппаратными возможностями объектного компьютера;
* верифицируемость требований проекта ПО – оценивается, что каждое требование может быть верифицировано;
* трассируемость требований проекта ПО – оценивается, что требования технического задания были реализованы в требованиях проекта ПО.
* реализуемость требований проекта ПО - оценивается, что каждое требование может быть реализовано.

### Входы и выходы

Входными данными процесса верификации на этапе верификации проекта ПО КРОСС являются:

* план верификации программного обеспечения КРОСС;
* техническое задание;
* описание программы.

Выходные результаты процесса верификации на этапе верификации проекта ПО КРОСС содержатся в следующих документах:

* отчет по верификации проекта программного обеспечения КРОСС;
* запросы на изменение, сформированные в ходе верификации проекта ПО КРОСС.

## **Этап 3. Верификация исходных кодов ПО КРОСС**

### Задачи

Обнаружение ошибок в исходном коде ПО КРОСС которые могут быть допущены в процессе кодирования ПО.

Регистрация обнаруженных ошибок.

Проверка корректирующих действий проводимых по исправлению обнаруженных ошибок.

Проверка внесенных исправлений в проект ПО, в ТЗ на ПО КРОСС и в исходный код в случае необходимости.

Подтверждение, что результаты процесса кодирования точны, полны и верифицируемы.

### Методы и критерии

Методом просмотра и анализа проведения процедур верификации на данном этапе является независимый сквозной аудит текстов исходных кодов программ из состава ПО КРОСС и экспертная оценка выпущенной программной документации.

В качестве критерия для проводимой верификации выступают формальные правила написания программ на языках программирования Си и Си++, отсутствие синтаксических и логических ошибок в текстах программ, соответствие разработанного программного кода требованиям проекта.

В ходе верификации исходного кода ПО КРОСС методом просмотра и анализа проверяется:

* исходный код читаем, понятен и пригоден к проверке;
* исходный код соответствует требованиям проекта и технического задания;
* исходный код не содержат потенциально опасных участков;
* исходный код не содержат вредоносный код;
* исходный код не содержат недокументированных функций;
* исходный код согласуется с потоком данных и потоком управления, которые определены в архитектуре ПО;
* исходный код не содержит операторов и структур, которые не могут быть проверены, и для испытаний в него не требуется вносить изменения;
* правильность и непротиворечивость исходного кода, включая оценку конкуренции в использовании ресурсов, обработки исключительных ситуаций, использования неинициализированных переменных или констант, неиспользуемых переменных или констант, а также нарушения целостности данных.

### Входы и выходы

Входными данными процесса верификации на этапе верификации исходных кодов ПО КРОСС являются:

* план верификации программного обеспечения КРОСС;
* техническое задание;
* описание программы;
* текст программы.

Выходные результаты процесса верификации на этапе верификации исходного кода ПО КРОСС содержатся в следующих документах:

* отчет по верификации исходного кода программного обеспечения КРОСС;
* запросы на изменение, сформированные в ходе верификации исходного кода ПО КРОСС.

## **Этап 4. Функциональное тестирование и верификация результатов тестирования ПО КРОСС**

### Задачи

Проверка что существуют тестовые наборы для каждого требования к ПО.

Проверка что тестовые наборы разработаны так, чтобы верифицировать корректность функционирования и сформировать условия, которые выявляют потенциальные ошибки.

Функциональное тестирование ПО КРОСС, в ходе которого должно быть продемонстрировано, что программный комплекс КРОСС выполняет требования ТЗ.

Обнаружение ошибок в коде ПО КРОСС которые могут быть допущены в процессе разработки ПО.

Регистрация обнаруженных ошибок.

Проверка корректирующих действий проводимых по исправлению обнаруженных ошибок.

Проверка внесенных исправлений в проект ПО, в ТЗ на ПО КРОСС и в исходный код в случае необходимости.

Проверка результатов тестирования на предмет, что испытания выполнены в полном объеме в соответствии с тестовыми процедурами и результаты испытаний корректны.

### Методы и критерии

В ходе верификации тестовых наборов для функционального тестирования компонентов ПО КРОСС методом анализа и обзора проверяется:

* соответствие требований ТЗ и методов испытаний;
* наличие тестовых наборов для каждого требования к ПО;
* наличие тестовых наборов для тестирования на устойчивость к ошибкам входных данных;
* наличие тестовых наборов, разработанных на основе требований к ПО с учетом потенциальных источников ошибок.

Функциональное тестирование осуществляется методом «черного ящика». В качестве критерия для функционального тестирования выступает:

* ПО КРОСС функционирует в соответствии с требованиями ТЗ;
* компоненты ПО КРОСС полностью и правильно интегрированы в программный комплекс;
* ПО КРОСС правильно взаимодействуют с техническими средствами;
* ПО КРОСС устойчиво функционируют при отказах и сбоях технических средств, ошибках во входных данных;
* все данные, которыми обменивается ПО КРОСС, соответствуют проектным спецификациям структур данных.

### Входы и выходы

Входными данными процесса верификации на этапе функционального тестирования компонентов ПО КРОСС являются:

* план верификации программного обеспечения КРОСС;
* техническое задание;
* тестовые процедуры и тестовые наборы для функционального тестирования.

Выходные результаты процесса верификации на этапе функционального тестирования компонентов ПО КРОСС содержатся в следующих документах:

* протоколы функционального тестирования;
* отчет по верификации на этапе функционального тестирования ПО КРОСС;
* запросы на изменение, сформированные в ходе функционального тестирования и верификации ПО КРОСС.

# **Отчетность по верификации**

Отчетная документация по верификации программного обеспечения КРОСС должна включать:

* отчеты по выполнению верификации на каждом из этапов;
* запросы на изменение, сформированные в ходе верификации на каждом из этапов;
* протоколы или отчеты автономного тестирования программного обеспечения;
* протоколы функционального тестирования программного обеспечения.

Отчет по верификации требований должен включать следующие пункты:

* задачи верификации на этапе разработки требований к ПО;
* методы и критерии верификации;
* результаты верификации;
* заключение.

Отчет по верификации проекта ПО должен включать следующие пункты:

* задачи верификации на этапе разработки проекта ПО;
* методы и критерии верификации;
* результаты верификации;
* заключение.

Отчет по верификации исходных кодов должен включать следующие пункты:

* задачи верификации на этапе кодирования ПО;
* методы и критерии верификации;
* результаты верификации;
* заключение.

Отчет по верификации на этапе функционального тестирования должен включать следующие пункты:

* задачи верификации на этапе функционального тестирования;
* методы и критерии верификации;
* результаты верификации;
* заключение.

Запрос на изменение должен включать следующие пункты:

* номер запроса;
* автор запроса;
* описание проблемы (описание причин возникновения изменений, влияние предлагаемых изменений, к чему может привести невыполнение предлагаемых изменений);
* решение о корректирующих действиях;
* отметка о реализации корректирующих действий;
* краткий отчет о проверке внесенных изменений;
* статус (подан/утвержден/отложен/отклонен/закрыт).

Протоколы функционального тестирования программного обеспечения должны включать следующие пункты:

* объект испытаний;
* цель испытаний;
* конфигурацию технических средств, примененную при тестировании;
* дополнительные данные, связанные со временем, последовательностью событий и т.д.;
* отчеты о несоответствиях заданию;
* заключение о соответствии критериям приемки.

# **Анализ результатов верификации**

## **Анализ результатов верификации требований к ПО**

Все мероприятия, проводимые в ходе верификации требований, должны быть зафиксированы в отчете по верификации требований. Должен быть проведен анализ, что все требуемые мероприятия выполнены.

Все обнаруженные в ходе верификации требований несоответствия должны быть зафиксированы в отчете по верификации требований.

На каждое несоответствие необходимо сформировать Запрос на изменение в системе управления проектами Redmine.

Сформированный Запрос на изменение должен быть проанализирован и вынесено решение о корректирующих действиях по устранению данного несоответствия.

После устранения несоответствий выявленных в ходе верификации требований к ПО необходимо провести повторную верификацию скорректированных требований.

Результаты повторной верификации должны быть зафиксированы в отчете по верификации требований.

Результаты повторной верификации должны быть проанализированы, в случае отсутствия несоответствий принимается решение о переходе к следующему этапу разработки ПО.

## **Анализ результатов верификации требований к ПО**

Все мероприятия, проводимые в ходе верификации проекта ПО, должны быть зафиксированы в отчете по верификации проекта ПО. Должен быть проведен анализ, что все требуемые мероприятия выполнены.

Все обнаруженные в ходе верификации проекта ПО несоответствия должны быть зафиксированы в отчете по верификации проекта ПО.

На каждое несоответствие необходимо сформировать Запрос на изменение в системе Redmine.

Сформированный Запрос на изменение должен быть проанализирован и вынесено решение о корректирующих действиях по устранению данного несоответствия.

В случае если корректирующие действия затрагивают только проект ПО, после устранения несоответствий выявленных в ходе верификации проекта ПО необходимо провести повторную верификацию скорректированных требований проекта ПО.

В случае если корректирующие действия затрагивают и требования ТЗ и требования проекта ПО, после устранения несоответствий выявленных в ходе верификации проекта ПО необходимо провести повторную верификацию скорректированных требований ТЗ и проекта ПО.

Результаты повторной верификации должны быть зафиксированы в отчете по верификации проекта ПО.

Результаты повторной верификации должны быть проанализированы, в случае отсутствия несоответствий принимается решение о переходе к следующему этапу разработки ПО.

## **Анализ результатов верификации исходных кодов**

Все мероприятия, проводимые в ходе верификации исходных кодов ПО, должны быть зафиксированы в отчете по верификации исходных кодов. Должен быть проведен анализ, что все требуемые мероприятия выполнены.

Все обнаруженные в ходе верификации исходных кодов ошибки и несоответствия должны быть зафиксированы в отчете по верификации.

На все обнаруженные ошибки и несоответствия необходимо сформировать Запрос на изменение в системе Redmine.

Сформированный Запрос на изменение должен быть проанализирован и вынесено решение о корректирующих действиях по устранению данной ошибки или несоответствия.

В случае если корректирующие действия затрагивают только исходный код, необходимо провести повторную верификацию измененного исходного кода.

В случае если корректирующие действия затрагивают исходный код и требования проекта ПО, необходимо провести повторную верификацию измененного исходного кода и скорректированных требований проекта ПО.

В случае если корректирующие действия затрагивают требования ТЗ, требования проекта ПО и исходный код, необходимо провести повторную верификацию скорректированных требований ТЗ и проекта ПО и измененного исходного кода.

Результаты повторной верификации должны быть зафиксированы в отчете по верификации проекта ПО.

Результаты повторной верификации должны быть проанализированы, в случае отсутствия несоответствий принимается решение о переходе к следующему этапу разработки ПО.

## **Анализ результатов верификации на этапе функционального тестирования**

Все мероприятия, проводимые в ходе верификации на этапе функционального тестирования, должны быть зафиксированы в отчете по верификации. Должен быть проведен анализ, что все требуемые мероприятия выполнены.

Функциональное тестирование должно быть выполнено в соответствии с заранее определенными тестовыми наборами и тестовыми процедурами.

Результаты функционального тестирования должны быть зафиксированы в протоколах тестирования.

Протоколы тестирования должны быть проанализированы на предмет:

* все тестовые процедуры выполнены;
* результаты всех тестов зафиксированы.

Должен быть проведен анализ тестового покрытия, в случае определения необходимости в дополнительном тестировании необходимо сформировать Запрос на изменение в системе Redmine. Данный запрос должен быть проанализирован и вынесено решение о создании дополнительных тестовых процедур и/или тестовых наборов и проведении дополнительного тестирования. По результатам дополнительного тестирования оформляется протокол.

Все обнаруженные в ходе верификации на этапе функционального тестирования ошибки и несоответствия должны быть зафиксированы в отчете по верификации.

На все обнаруженные ошибки и несоответствия необходимо сформировать Запрос на изменение в системе Redmine.

Сформированный Запрос на изменение должен быть проанализирован и вынесено решение о корректирующих действиях по устранению данной ошибки или несоответствия.

В случае если корректирующие действия затрагивают исходный код, требования проекта ПО или требования ТЗ необходимо провести повторную верификацию измененного исходного кода, проекта ПО и требований ТЗ.

Результаты повторной верификации должны быть зафиксированы в отчете по верификации на этапе функционального тестирования.

Результаты повторной верификации должны быть проанализированы, в случае отсутствия несоответствий принимается решение о переходе к следующему этапу разработки ПО.

**Перечень сокращений**

|  |  |  |
| --- | --- | --- |
| АСУ ТП | – | автоматизированная система управления технологическим процессом |
| АЭС | – | атомная электростанция |
| ВУ | – | верхний уровень |
| КРОСС | – | комплекс распределенных средств сетевой обработки |
| НД | – | нормативная документация |
| ПО | – | программное обеспечение |
| ПТК | – | программно-технический комплекс |
| РБДРВ | – | распределенная база данных реального времени |
| РС | – | рабочая станция |
| САПР | – | система автоматизированного проектирования |
| СЦК | – | система централизованного контроля |
| ТЗ | – | техническое задание |
| ТПТС | – | типовые программно-технические средства |
| ТС | – | технические средства |

**Перечень ссылочных документов**

1. ГОСТ Р МЭК 61513-2020 «Атомные станции. Системы контроля и управления, важные для безопасности. Общие требования»

2. ГОСТ Р МЭК 62138-2010 «Атомные электростанции. Системы контроля и управления, важные для безопасности. Программное обеспечение компьютерных систем, выполняющих функции категории В и С»

3. ГОСТ Р МЭК 61226-2011 «Атомные электростанции. Системы контроля и управления, важные для безопасности. Классификация функций контроля и управления»

4. ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц)  в  документе | Номер  документа | Входящий номер сопрово-дительного документа и дата | Подпись | Дата |
| изме-ненных | заме-  ненных | новых | аннулиро-ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |