

# ОЦЕНКА КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОНИТОРИНГА И УПРАВЛЕНИЯ АВТОНОМНЫМИ ЭЛЕКТРОЭНЕРГЕТИЧЕСКИМИ СИСТЕМАМИ

УДК 004.05

## **ДОРОГАНЬ Ольга Игоревна**

аспирант кафедры теоретической электротехники и электронных систем  
Национального университета кораблестроения им. адм. Макарова.

**Научные интересы:** информационное обеспечение электроэнергетических систем, имитационное моделирование в электроэнергетике.

## **РЯБЕНЬКИЙ Владимир Михайлович**

д.т.н., профессор, заведующий кафедрой теоретической электротехники  
и электронных систем Национального университета кораблестроения им. адм. Макарова.

**Научные интересы:** автоматизация электроэнергетических систем, компьютеризированные системы управления.

## **УШКАРЕНКО Александр Олегович**

к.т.н., доцент кафедры теоретической электротехники и электронных систем  
Национального университета кораблестроения им. адм. Макарова.

**Научные интересы:** автоматизация электроэнергетических систем, информационное обеспечение электроэнергетических систем.

## **ВВЕДЕНИЕ**

Современные системы управления автономными электроэнергетическими системами (АЭЭС) являются распределенными и условно разделены на три иерархических уровня: аппаратура и программное обеспечение верхнего уровня, координирующие работу всей системы и содержащие сервер сбора данных и автоматизированное рабочее место оператора, совокупность территориально распределенных аппаратных средств автоматизации на нижнем уровне и коммуникационная инфраструктура в качестве элемента промежуточного уровня. При проектировании программного обеспечения (ПО) верхнего уровня одним из ключевых вопросов является его качество. Общепринятой признана модель качества ПО SquaRE, основанная на стандартах ISO9126 и ISO14598. В соответствии с этой моде-

лью оценка качества является трехуровневой и состоит в определении для каждого типа ПО необходимых характеристик, для каждой из характеристик – атрибутов и для каждого из атрибутов – метрик. Стандарты также регламентируют основные характеристики для всех типов ПО: функциональность, практичность, эффективность, надежность, сопровождаемость и мобильность.

**Целью работы** является разработка методики тестирования и оценки основных показателей качества специализированного программного обеспечения для мониторинга и управления автономными электроэнергетическими установками в соответствии с международными стандартами, что позволит получить состав атрибутов качества ПО и количественные значения метрик, конкретизировать направление первоочередных изменений и оценить результаты их выполнения,

определить показатели эффективности и надежности специализированного программного обеспечения.

### ОСНОВНОЙ МАТЕРИАЛ

В качестве атрибутов функциональности программного обеспечения можно выделить пригодность и точность выполнения функций, способность к взаимодействию со сторонними ПО и аппаратными средствами автоматизации [1]. В работе [2] приведены методы оценки функциональности ПО и обоснована эффективность использования тестирования с этой целью.

Основой для проведения тестирования функциональности ПО являются требования, которым оно должно удовлетворять. В работе [3] приведены требования к ПО для мониторинга и управления АЭС, для описания которых были использованы средства UML: диаграммы вариантов использования, коопераций и последовательностей. Все требования разделены на три группы по действующим лицам, которые их выполняют, – Оператор наладки системы, Оператор управления электростанцией и Аналитик. Задачей Оператора наладки системы является создание мнемосхемы электростанции и настройка свойств ее компонентов, поэтому к требованиям соответствующей группы относятся наличие визуальных метафор элементов электростанции (дизель-генераторный агрегат (ДГА), автоматический выключатель, нагрузка, система синхронизации генераторов и т.д.) и возможность выполнения с ними действий (перемещение, настройка взаимодействий, изменение свойств). Оператор управления электростанцией должен выполнять мониторинг электроэнергетических процессов в системе и иметь возможность изменения структуры электростанции – выполнять подключение/отключение нагрузки к шинам главного распределительного щита (ГРЩ), пуск/остановку и подключение на параллельную работу ДГА. Часть действий при этом выполняются в автоматическом режиме. Также должна присутствовать информационная поддержка Оператора (разнообразные средства для отображения электрических параметров системы и средства для оценки возможных последствий подключения нагрузки). К действиям Аналитика относятся анализ информационных потоков для расчета временных характеристик системы, анализ режимов работы ДГА с целью оптимизации их использования и автома-

тизированное тестирование ПО. Требования к отображению мнемосхемы электростанции определяют существование двух режимов работы программного обеспечения: режим конструктора схемы (составление мнемосхемы и настройка сетевых свойств ее компонентов) и режим мониторинга параметров и управления электроэнергетической системой.

ПО для мониторинга и управления электроэнергетической системой является составной частью программно-аппаратного комплекса, поэтому для его тестирования необходимо наличие моделей, имитирующих поведение аппаратных средств и объекта автоматизации как единого целого. В работе [4] приведены модели контроллера автоматизации, системы синхронизации и распределения нагрузок, которые и предлагается использовать с этой целью.

Функциональное тестирование обычно разделяют на несколько уровней: модульное, интеграционное и системное [5]. Процедура тестирования состоит в выполнении следующей последовательности действий:

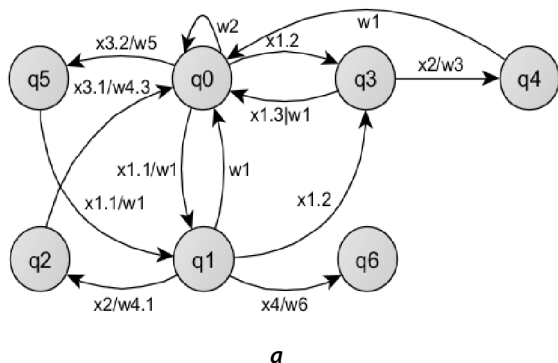
- определение критерия полноты тестирования;
- составление полного набора тестовых ситуаций;
- оформление отчета с информацией о результатах тестирования.

Результатом проведения тестирования должны быть значения метрик атрибутов функциональности – пригодности, точности и способности к взаимодействию. Поскольку тестирование ПО является многоступенчатым процессом, в работе предлагается создание таблицы, заполнение которой будет происходить после прохождения каждого этапа тестирования.

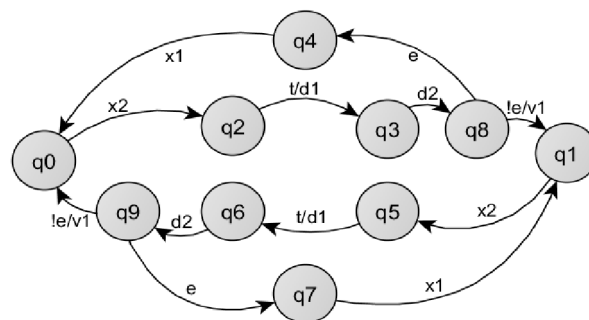
Задачей модульного тестирования является определение метрик атрибутов функциональности отдельных компонентов ПО. Для большинства компонентов программного обеспечения поведение в режиме конструктора значительно отличается от поведения в режиме управления электростанцией и реализовано (спроектировано и закодировано) в виде двух отдельных конечных автоматов. Приведенный подход позволяет изоморфно перейти к автомату, который моделирует поведение компонента, и в качестве критерия полноты тестирования применить покрытие его переходов [5]. Использование двух автоматов позволяет отделить расчет метрик пригодности (полнота тестового покрытия для автомата, моделирующего поведение

в режиме конструктора схемы) и способности к взаимодействию (для поведения в режиме управления электростанцией). Подобный подход приемлем только для компонентов, которые принимают участие в обмене данными с аппаратными средствами автоматизации и не требуют обязательного информационного обмена с другими компонентами (ДГА, Защита ДГА, Синхронизация, Кнопка, Автоматический выключатель, Светодиод). Для компонентов, которые используются лишь для визуальной полноты мнемосхемы и не требуют связи с другими компонентами (Асинхронный двигатель, Трансформатор), определяется только пригодность.

Значение точности как степени соответствия срабатывания устройств установленным граничным значениям на этапе модульного тестирования может быть рассчитано лишь для компонента Защита ДГА.



а



б

Рисунок 1 – Поведение компонента Автоматический выключатель

При тестировании используются те же самые автоматы, что и при описании и кодировании поведения компонента; верификация изменения состояний выполняется с помощью встроенных средств компилятора для отладки программ. Полное покрытие автомата, приведенного на рис. 1, а, происходит при генерации последовательности сообщений:

$x1.1 \rightarrow x2 \rightarrow x3.1 \rightarrow x1.2 \rightarrow$   
 $x2 \rightarrow w1 \rightarrow w2 \rightarrow x3.2 \rightarrow x1.1 \rightarrow$   
 $\rightarrow x1.2 \rightarrow x1.3 \rightarrow x1.1 \rightarrow w1$  ;  
 $\rightarrow x1.2 \rightarrow x2 \rightarrow w1 \rightarrow x1.1 \rightarrow x4$

для автомата, приведенного на рис. 1, б:

$x2 \rightarrow t \rightarrow d2 \rightarrow e \rightarrow x1 \rightarrow x2 \rightarrow t \rightarrow d2 \rightarrow !e \rightarrow$   
 $\rightarrow x2 \rightarrow t \rightarrow d2 \rightarrow e \rightarrow x1 \rightarrow x2 \rightarrow t \rightarrow d2 \rightarrow !e$

Следует отметить, что генерация отдельных сообщений (получение данных  $d2$ , ошибка  $e$  или отсутствие ошибки  $!e$ ) возможна только при использовании

Таким образом, в качестве критерия полноты модульного тестирования всех компонентов, для которых оно проводится, определяется полнота покрытия переходов автоматов; полный набор тестовых ситуаций составляется на основе условий переходов автоматов и результаты тестирования заносятся в таблицу. Поскольку данный этап тестирования относится к отладочному, переход к добавлению функциональности и проведение следующего этапа тестирования происходит только при условии 100% покрытия тестовых ситуаций.

В качестве примера рассматривается тестирование компонента Автоматический выключатель. На рис. 1 приведены автоматы, описывающие поведение компонента в режиме конструктора схемы (а) и в режиме управления электростанцией (б).

контроллера автоматизации или его имитационной модели.

Следующим уровнем тестирования является интеграционное тестирование, цель которого состоит в верификации взаимодействий между компонентами. Использование аппарата паттерновых сетей, в которых автоматы, реализующие поведение компонентов, представлены в виде образующих, а каждый из видов информационного обмена между компонентами – в виде отношения связи, позволяет формализовать описание взаимосвязей между компонентами. Таким образом, в процессе интеграционного тестирования для каждой пары взаимодействующих компонентов в качестве критерия полноты тестирования можно использовать покрытие отношений связи. Поскольку данный уровень тестирования, также как и модульное тестирование, относится к отладочному, то он может быть завершен только при 100% покрытии тестовых ситуа-

ций. Результаты тестирования (доля верифицированных отношений связи для каждого из компонентов) заносятся в таблицу.

Завершающий уровень тестирования – системный. Работа электростанции – это определенная последовательность работы ДГА и исполнительных механизмов в соответствии с технологическим процессом. Целью системного тестирования является проверка работоспособности ПО при управлении электростанцией, поэтому в качестве критерия полноты тестирования можно выделить покрытие требований к ПО. Перед составлением набора тестовых ситуаций необходимо определиться с классами эквивалентных ситуаций. В работе предлагается использовать следующее множество неэквивалентных ситуаций:

- использование компонента каждого типа в режиме конструктора схемы;
- пуск и остановка ДГА;
- подключение и отключение нагрузки от шин ГРЩ;

- отображение параметров ДГА;
- срабатывание каждой из функций защит ДГА;
- синхронизация ДГА с шинами ГРЩ;
- распределение нагрузки между параллельно работающими ДГА.

Для осуществления полного тестирования ПО набор ситуаций должен включать все перечисленные неэквивалентные ситуации.

ПО для мониторинга и управления электростанцией является дискретно-событийной системой, поэтому в качестве способа составления тестовых ситуаций могут быть использованы конечные автоматы [6]. Однако, ПО также является и составляющей программно-аппаратного комплекса, поэтому для возможности определения функций, которые подлежат проверке, и моделей и средств ПО, которые для этого используются, в работе предлагается задавать тестовую последовательность в виде R-схемы (рис. 2) [7, 8].

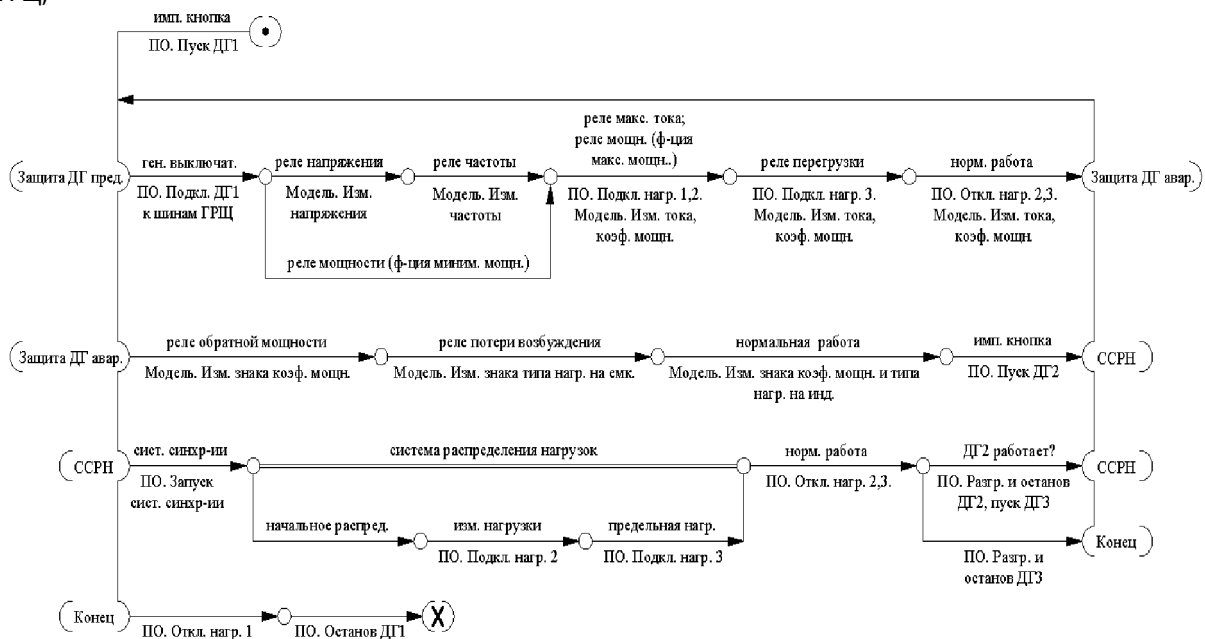


Рисунок 2 – Тестовая последовательность действий для валидации ПО

На рис. 2 использованы следующие обозначения: ССРН – система синхронизации и распределения нагрузки, ДГ – дизель-генератор, ГРЩ – главный распределительный щит, ПО – программное обеспечение.

Для автоматизации и ускорения процесса тестирования предлагается реализация R-схемы в виде

управляющего скрипта, который в качестве схемы имитации поддерживает принцип особых моментов [9]. Для ПО скрипт может быть представлен в виде дополнительного диалогового окна; запуск скрипта будет инициировать выполнение внесенных в него действий с определенными временными. Для интер-

претации результатов тестирования все информационные уведомления и управляющие действия заносятся в файл протокола. После окончания тестирования данные из протокола анализируются, и на их основе рассчитываются пригодность (доля корректно выполненных функций от всех проверяемых), способность к взаимодействию (доля корректно отправленных и обработанных запросов) и точность для каждого компонента (усредненное значение временной рассогласованности срабатывания защит (в относительных единицах) для Защиты ДГА; рассогласование процентов неравномерностей распределения нагрузки для Распределения нагрузки). Полученные данные заносятся в таблицу.

К задачам системного тестирования также относятся функции проверки средств, необходимых для выполнения действий Аналитиком – анализ содержимого пакетов обмена данными, статистики нагрузки на информационный канал, графиков работы силовых агрегатов и электрических нагрузок. Их проверка возможна только после определенного времени работы ПО в режиме управления электростанцией, и единственным атрибутом функциональности, который может быть получен в результате этой проверки, является пригодность.

Для расчета числовых значений пригодности, способности к взаимодействию и точности для каждого компонента и каждого из средств для обеспечения работы Аналитика определяются весовые коэффициенты. Далее с помощью аддитивной свертки и нормализации рассчитываются значения атрибутов:

$$M = \frac{\sum_{i=1}^{15} k_{\delta.i.i} \cdot k_{\delta.\delta.i} \cdot \alpha_i}{\sum_{i=1}^{15} \alpha_i}, \quad (1)$$

где  $M$  – значение атрибута;  $k_{m.n.i}$  – значение тестового покрытия;  $k_{p.m.i}$  – результаты тестирования;  $\alpha_i$  – весовой коэффициент метрики.

Для ПО для мониторинга и управления электростанцией по результатам тестирования были рассчитаны следующие значения атрибутов: пригодность – 0,91; способность к взаимодействию – 0,86; точность – 0,81. Анализируя таблицу с результатами всех этапов тестирования можно сделать выводы, что повысить значения атрибутов «пригодность» и «способность к взаимо-

действию» можно путем усиления системного тестирования, а улучшение точности связано с изменением алгоритмов работы ПО.

Значение функциональности рассчитывается по аналогичному алгоритму: атрибутам назначаются весовые коэффициенты и с помощью свертки и нормализации рассчитывается числовое значение. Задав коэффициенты для пригодности и способности к взаимодействию как 1, а для точности – как 0,8, можно получить значение функциональности – 0,86.

*Практичность.* Оценка практичности программного обеспечения состоит в расчете значений трех ее атрибутов: эргономичности, понятности и эффективности освоения. Для каждого из атрибутов также выделяются подхарактеристики – показатели, которые оцениваются численно. Для ПО для мониторинга и управления автономной электростанцией можно выделить следующие показатели: само-описательность интерфейса, простота выполнения частых операций, легкость выполнения сложных операций, приемлемость задержек, отображение графических компонентов и цветовая гамма, информационная полнота второстепенных диалоговых окон (атрибуты эргономичности, далее обозначенные как 1.1 – 1.6); интуитивная понятность, соответствие поведения ПО ожидаемому, удобство использования библиотеки компонентов (атрибуты понятности, далее обозначенные как 2.1 – 2.3); легкость повторного создания мнемосхемы, легкость восстановления навыков работы с ПО, удобство определения некорректных настроек обмена данными (атрибуты эффективности усвоения, далее обозначенные как 3.1 – 3.3).

В работе [10] приведены методы оценки метрик практичности; среди них как наиболее широко используемые определены тестирование, экспертная оценка и опрос. Для ПО для мониторинга и управления автономной электростанцией в качестве основы для оценки практичности выбран опрос, для проведения которого были привлечены 10 пользователей. На основе данных, полученных в результате опроса, определены нормированные средние статистические значения показателей практичности:

$$S_j = \frac{1}{k} \cdot \frac{\sum_{i=1}^m \alpha_{i,j}}{m}, \quad (2)$$

где  $k$  – нормировочный коэффициент (максимальная оценка – 10);  $\alpha_{ij}$  – оценка  $j$ -того показателя  $i$ -м пользователем;  $m$  – количество опрошенных пользователей.

Далее с помощью введения весовых коэффициентов важности и аддитивной свертки выполняется расчет значений атрибутов:

$$A_j = \frac{1}{\sum_{i=1}^{m_j} p_{ij}} \cdot \sum_{i=1}^{m_j} (S_{ij} \cdot p_{ij}^{(n)}), \quad (3)$$

где  $p_{ij}^{(n)}$  – весовой коэффициент  $i$ -того показателя  $j$ -того атрибута;  $S_{ij}$  – нормированное среднее статистическое значение  $i$ -того показателя  $j$ -того атрибута;  $m_j$  – количество показателей  $j$ -того атрибута.

Аналогичным образом (введением коэффициентов важности и использованием аддитивной свертки) можно получить численное значение практичности.

Для визуализации результатов расчета предлагается использовать радиальную метрическую диаграмму [13, 14], с помощью которой удобно отображать «покрытие» требований к показателям (рис. 3).

Представление результатов оценки практичности с помощью одного числа не выявляет возможные проблемные места [10], однако оно позволяет на разных этапах разработки и внедрения ПО определять необходимость дальнейших изменений и результатов их выполнения. Конкретизировать направление первоочередных изменений в ПО можно рассматривая результаты расчетов в «обратном» направлении. Для этого необходимо выбрать атрибуты с наиболее значимыми весами, а потом среди выбранных – с наименьшими рассчитанными значениями:

$$\min_{A_j} \max_{p_j} \langle A_j, p_j^{(a)} \rangle, \quad (4)$$

и далее выбрать наименьшие по значению из наибольших по весу показателей атрибутов, которые были выбраны на предыдущем этапе:

$$\min_{S_{ij}} \max_{p_{ij}} \langle S_{ij}, p_{ij}^{(n)} \rangle. \quad (5)$$

Результатом проведения описанной процедуры будет множество показателей, в направлении которых улучшение ПО является наиболее важным. Для облегчения расчетов минимаксных критериев можно считать значения атрибутов и показателей с учетом

«обратного» веса, рассчитанного как алгебраическое дополнение до 2 «истинного» веса:

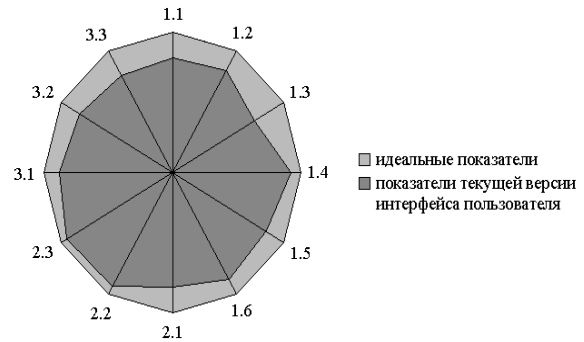


Рисунок 3 – «Покрытие» требований к показателям практичности

$$A_j^* = (2 - p_j) \cdot A_j, \quad (6)$$

$$S_{ij}^* = (2 - p_{ij}) \cdot S_{ij}. \quad (7)$$

Тогда определение наиболее критичных показателей для улучшения практичности ПО будет соответствовать процедуре определения наименьших значений дополненных атрибутов, а потом – среди наименьших значений дополненных показателей:

$$\min_{p_{ij}} \min_{A_j} \langle A_j, \{p_{ij}\} \rangle. \quad (8)$$

Рассмотрим анализ результатов расчета практичности ПО для мониторинга и управления энергосистемой. После расчетов были получены диаграмма «покрытия» требований и численное значение практичности – 0,86. Например, это значение и покрытие были выявлены неудовлетворительными, поэтому в ПО решено внести изменения. Область изменений (конкретнее – показатели практичности) необходимо определять с учетом значимости отдельных направлений. Для этого из весовых коэффициентов атрибутов выбирается наибольший – 1,0, что расшифровывается как «для данного ПО эргономичность является наиболее важной». Далее из весовых коэффициентов показателей, которые являются составляющими эргономичности, выбирается наибольший – 1,0. Выбранному весу соответствуют два показателя – «простота выполнения частых операций» и «легкость выполнения сложных операций». Из последних выбирается тот, у которого полученное в результате опроса среднее статистическое значение наименьшее («легкость выполнения сложных операций») – именно в этом направлении

изменения являются наиболее важными. Результатом для рассматриваемой ситуации будет показатель «удобство определения некорректных настроек обмена данными» – изменения в этом направлении также будут значительно влиять на улучшение практичности.

Упростить указанную последовательность действий можно рассчитав дополненные значения атрибутов и показателей (значения «дополняются» для минимизации влияния на результат атрибутов и показателей с малыми весами). После их расчета выбирается наименьшее дополненное значение атрибута – 0,83 (атрибут «эффективность освоения») и для соответствующего атрибута – наименьшее дополненное значение показателя – 0,8 (показатель «удобство определения некорректных настроек обмена данными»). Для определения второго по необходимости изменений направления проводится аналогичная процедура, но при исключенном атрибуте «эффективность освоения». Результатом выполнения последней будет показатель «легкость выполнения сложных операций».

После выбора показателей, в направлении которых необходимо проводить первоочередные изменения, интерфейс ПО корректируется и далее снова проводится опрос и его результаты обрабатываются приведенным способом. Этот процесс повторяется до тех пор, пока «покрытие» требований и значение практичности не станут удовлетворительными.

**Эффективность.** Эффективность ПО включает два атрибута: эффективность по времени и эффективность по ресурсам. ПО для мониторинга и управления электростанцией принадлежит к системам управления реального времени, поэтому эффективность по времени является одним из ключевых показателей его качества. В работе [13] приведен алгоритм расчета временных характеристик рассматриваемой системы. Требования ПО для мониторинга и управления электростанцией к ресурсам являются минимальными (для надежной работы ПО потребуется не менее 15 Мб RAM и не менее 1 Мб HDD).

**Надежность.** Надежность ПО состоит из трех атрибутов: безотказности, стойкости к ошибкам и восстанавливаемости.

Показателями безотказности ПО являются математическое ожидание и среднеквадратичное отклонение времени наработки до возникновения отказа, ус-

ловная функция надежности (вероятность того, что случайная наработка до последующего отказа будет более чем заданная) [14]. Расчет перечисленных показателей ведется на основе полученных в результате функционального тестирования промежутков времени между последовательными отказами. Статистические оценки численных характеристик случайного времени между отказами получены в соответствии с выражениями:

$$m_{\Delta t}^* = \frac{1}{n_i} \sum_{i=1}^{n_i} \Delta t^{(i)} = \frac{1}{10} \cdot 89,08 = 8,908 \text{ (ч)}, \quad (9)$$

$$[\sigma_{\Delta t}^2]^* = \frac{1}{n_i - 1} \sum_{i=1}^{n_i} [\Delta t^{(i)} - m_{\Delta t}^*]^2 = \frac{1}{9} \cdot 33,75 = 3,75 \text{ (ч}^2\text{)}. \quad (10)$$

Математическое ожидание следующего (в данном примере – одиннадцатого) отказа:

$$m_{t_n} = m_{\Delta t} \frac{n \cdot (n+1)}{2} = 8,908 \cdot 66 \approx 588 \text{ (ч.);} \quad (11)$$

среднеквадратического отклонение времени наработки:

$$\sigma_{t_n} = \sigma_{\Delta t} \sqrt{\frac{1}{6} n \cdot (n+1) \cdot (n+2)} = \sqrt{3,75} \cdot \sqrt{7,7} = \epsilon \text{ (ч);} \quad (12)$$

условная функция надежности:

$$p^{(n)}(\tau) = 0,5 - \hat{O} \left( \frac{\tau - n \cdot m_{\Delta t}}{\sigma_{\Delta t} \cdot \sqrt{n}} \right) = 0,5 - \hat{O} \left( \frac{\tau - 97,9}{6,423} \right) \quad (13)$$

где  $\tau$  – заданная наработка;  $\Phi(u)$  – интеграл вероятности.

Как пример использования функции надежности рассматривается промежуток времени, вероятность наработки которого составляет более 85%:

$$0,85 = 0,5 - \hat{O} \left( \frac{\tau - 97,988}{6,423} \right); \quad \tau \approx 91,3 \text{ (ч)}. \quad (14)$$

Таким образом, с вероятностью 85% можно утверждать, что программное обеспечение будет безотказно работать более чем 91,3 часа. Увеличить это значение можно за счет дальнейшего сбора статистических данных про отказы, которые приведут к перерасчету статистических показателей качества.

Стойкость ПО к ошибкам состоит в его способности выполнять функции при аномальных условиях. Для определения аномальных условий необходимо выде-

лить прецеденты, которые требуют наличия сторонних программных и/или аппаратных средств. Выделенные прецеденты, их требования к наличию сторонних

средств и реакция программного обеспечения для мониторинга и управления АЭС приведены в табл. 1.

Таблица 1 –

### Реакция ПО на аномальные условия

Действия пользователя	Программные и/или аппаратные средства	Реакция ПО на сбой
прецеденты, связанные с обменом данными	аппаратн. средства автоматизации	индикация отсутствия связи и исключение запроса из очереди
прогнозирование результата подключения нагрузки	Matlab Engine	информационное уведомление и автоматич. отклонение действия
сохранение/восстановление мнемосхемы и среды ПО; анализ данных про загруженность канала связи; автоматизация тестирования	средства операционной системы для обеспечения доступа к файлам	информационное уведомление
накопление и анализ данных про работу ДГА и состояние нагрузок	СУБД PostgreSQL	информационное уведомление; возможность изменения настроек; атомарность содержимого таблиц

Восстанавливаются ПО для мониторинга и управления АЭС состоит в восстановлении мнемосхемы и настроек компонентов. Это требование является одним из главных требований к ПО, поэтому оно проверяется при проведении функционального тестирования.

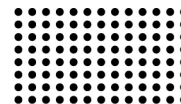
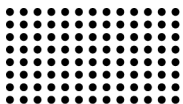
*Сопровождаемость.* Атрибутами сопровождаемости являются простота анализа, изменяемость, стабильность и проверяемость. Использование автоматического подхода и аппарата паттерновых сетей при проектировании и реализации ПО позволяет достаточно просто вносить изменения в поведение компонентов системы (с помощью добавления новых состояний, входных и выходных воздействий к автомату) и добавлять новые компоненты (с помощью создания новых образующих с соответствующими отношениями связи к паттерновой сети). Использование указанных методов также обуславливает высокий уровень простоты анализа и проверяемости новых функциональных возможностей ПО.

### ВЫВОДЫ

Современный процесс разработки ПО является итеративным и включает несколько стадий: анализ

требований, разработка архитектуры, кодирование, тестирование и отладка, документирование, внедрение и сопровождение. На этапе тестирования и отладки проводится оценка соответствия ПО определенным нормам качества. Расчет основных для рассматриваемой категории ПО показателей качества определяет необходимость возвращения к более ранним этапам разработки (анализу требований – кодированию) или перехода к следующим (документирование и далее). В работе рассмотрена методика количественной оценки основных показателей качества специализированного программного обеспечения для мониторинга и управления автономными электроэнергетическими системами – функциональности, практичности и надежности. Использование приведенной методики позволило получить состав атрибутов программного обеспечения и количественные значения метрик, конкретизировать направления первоочередных изменений и оценить результаты их выполнения, определить показатели надежности программного обеспечения. На основе анализа всех этапов тестирования установлено, что повысить функциональность ПО можно усилив системное тестирование и изменив некоторые алгоритмы работы программного обеспечения.





**ЛИТЕРАТУРА:**

1. Danilina T.G. Otsenka kachestva programmnogo obespecheniya v sootvetstvii s mezhdunarodnymi standartami /T.G. Danilina //Radioelektronni i kompyuterni sistemi. – 2012. – №7 (59). – S.266-269.
2. Kulyamin V.V. Obzor metodov postroyeniya pokryvayushchikh naborov /V.V. Kulyamin, A.A. Petukhov //Programmirovaniye. – 2011. – №3. – S.3-11.
3. Ryabenkiy V.M. Struktura spetsializovanogo programmnogo zabezpechennya dlya keruvannya avtonomnimi yelektroyenergetichnimi ustanovkami [Tekst] /Ryabenkiy V.M., Ushkarenko O.O., Dorogan O.I. //Vestnik KhNTU. –2012. – №1 (44) – S.113-116.
4. Dorogan O.I. Modelirovaniye apparatno-programmnykh sredstv avtomatizatsii elektroenergeticheskikh ustanovok /Dorogan O.I. //Avtomatika-2013. Materiali XX mizhnarodnoi konferentsii z avtomatichnogo upravlinnya. – Mikolaiv: NUK, 2013. – S.322.
5. Kurs lektsiy «Tekhnologicheskiye aspekty kontrolya kachestva programmnykh sistem» [Elektronnyy resurs]. – Rezhim dostupa: <http://panda.ispras.ru/~kulyamin/mbt-course.html>
6. Kulyamin V.V. Komponentnaya arkhitektura sredy dlya testirovaniya na osnove modeley /V.V. Kulyamin //Programmirovaniye. – 2010. – №5. – S.54-75.
7. Kharchenko S.L. Yazyk proyektirovaniya tekhnicheskogo zadaniya sistemy upravleniya /S.L. Kharchenko //Vostochno-Evropeyskiy zhurnal peredovykh tekhnologiy. – 2011. – №3 (49). – S.10-16.
8. Yelektronniy fond Glushkova [Elektronniy resurs]. – Rezhim dostupu: [http://glushkov.org/?page\\_id=195](http://glushkov.org/?page_id=195)
9. Olzoyeva S.I. Raspredelennoye modelirovaniye v zadachakh razrabotki ASU /S.I. Olzoyeva. – Ulan-Ude, 2005. – 209 s.
10. Pogrebnyak T.P. Otsenka kachestva interfeysov programnykh sistem: model i elementy metodiki /T.P. Pogrebnyak, A.A. Orekhova //Radioelektronni i kompyuterni sistemi. – 2009. – №7 (41). – S.251-254.
11. Tarasyuk O.M., Kharchenko V.S. Dinamicheskkiye radialnyye metricheskkiye diagrammy v zadachakh upravleniya kachestvom programmnogo obespecheniya //Zb. nauk. prats. in-tu problem modelyuvannya v yenergetitsi im. G.E. Pukhova. Vip. 22. – K: NANU, IPME, 2003. – S.202-205.
12. Kharchenko V.S. Primeneniye dinamicheskikh radialnykh metricheskikh diagramm dlya upravleniya mnogoversionnymi programmnyimi proyektami /V.S. Kharchenko, O.M. Tarasyuk, A.V. Volkovoy, Yu.A. Belyy //Radioelektronni i kompyuterni sistemi. – 2005. – №2 (10). – S.63-68.
13. Dorogan O.I. Doslidzhennya vplivu strukturi yelektroyenergetichnoi ustanovki na kharakteristiki informatsiynikh potokiv v merezhi keruvannya /Dorogan O.I. //Vestnik KhNTU. – 2013. – №1 (46) – S.366-371.
14. Shishmarev V.Yu. Nadezhnost tekhnicheskikh sistem: uchebn. dlya stud. vyssh. ucheb. zavedeniy /V.Yu. Shishmarev. – Moskva: Izdatelskiy tsentr «Akademiya», 2010. – 304 s.

**Рецензент:** *д.т.н., проф. Ходаков В.Е.,  
Херсонский национальный технический университет, Херсон.*