Интернет-журнал «Науковедение» ISSN 2223-5167 http://naukovedenie.ru/
Том 7, №2 (2015) http://naukovedenie.ru/index.php?p=vol7-2
URL статьи: http://naukovedenie.ru/PDF/74TVN215.pdf
DOI: 10.15862/74TVN215 (http://dx.doi.org/10.15862/74TVN215)

УДК 621.316

Типикина Анна Петровна
ФГБОУ ВПО «Национальный исследовательский университет «МЭИ»
Россия, Москва[1]
Филиал в г. Смоленске
аспирант
E-mail: anna73167@mail.ru

Певцова Людмила Сергеевна
ФГБОУ ВПО «Национальный исследовательский университет «МЭИ»
Россия, Москва
Филиал в г. Смоленске
старший преподаватель
E-mail: plus_energo@mail.ru

Оценка программной надежности микропроцессорных релейных защит

[1] Россия, 214013, г. Смоленск, Энергетический проезд, дом 1

Аннотация. В настоящий момент в электроэнергетике России происходит замена отработавших свой срок электромеханических и статических реле на микропроцессорные. Этот процесс был начат еще в конце прошлого века, и в настоящий момент уже можно говорить о показателях, достигнутых в этой области. В статье приводится анализ причин неправильной работы устройств релейной защиты и автоматики, выявлены основные тенденции и отмечены особенности показателей для микропроцессорных устройств защиты. Полученные выводы позволяют говорить о существенном влиянии на надежность цифровых устройств используемых программных средств и алгоритмов. В связи с этим проведен анализ программного обеспечения как объекта исследования надежности, выявлены сходства и отличия программной и аппаратной части устройств в части свойств надежности. Для расчета программной надежности предлагается метод на основе случайных дискретных Марковских процессов. Приведен пример расчета для простейшего алгоритма трехступенчатой максимальной токовой защиты. Полученные результаты позволяют говорить о применимости метода для анализа чувствительности надежности комплекса к надежности каждого модуля в отдельности. Результатом рассмотрения вышеизложенного явилась выдача рекомендаций для эксплуатационного персонала и проектных организаций.

Ключевые слова: микропроцессорные устройства релейной защиты; показатели надежности; аппаратная надежность; программная надежность; частота появления ошибок в программном обеспечении; дискретный Марковский процесс; максимальная токовая защита.

Ссылка для цитирования этой статьи:

Типикина А.П., Певцова Л.С. Оценка программной надежности микропроцессорных релейных защит // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, №2 (2015) http://naukovedenie.ru/PDF/74TVN215.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/74TVN215

Согласно данным ОРГРЭС [4] доля микропроцессорных релейных защит (МП РЗА) в энергосистеме России продолжает расти. По состоянию на 2013 год на объектах ЕНЭС в эксплуатации находится 23,9 % цифровых устройств РЗА. Для эффективной эксплуатации микропроцессорных терминалов требуется своевременное выявление причин неправильной работы, их анализ и устранение. Так в 2013 основной показатель правильной работы МП РЗА (99,12 %), рассчитанный по инструкции[2], ниже аналогичного для электромеханических устройств (99,27 %). Среди организационных причин неправильной работы существенный процент занимает вина монтажно-наладочных организаций (30,5 %) и заводов-изготовителей (22 %). Т.е. упущения происходят не на этапе эксплуатации (как для электромеханических реле), а на предшествующих. Кроме того, существенный процент относят к «прочим причинам» (14,1 %), т.е. без представления полной информации о конкретных причинах. Это говорит о сложности идентификации причин персоналом служб РЗА и о соответствующих упущениях в эксплуатации.

Если говорить о технических причинах неправильной работы, то стоит отметить высокую надежность аппаратной части МП РЗА по сравнению с прочим оборудованием (устройствами измерительного тракта и элементами оперативных цепей). В тоже время имеет место существенный процент неправильной работы по причину «дефекты разработки и сбои программного обеспечения (ПО)» (11,7 %). В связи с этим можно говорить, что нельзя пренебрегать надежностью программной части при проектировании и эксплуатации.

Актуальной проблема видится и потому, что согласно данным [8] на 1000 строк программного кода больших промышленных программ приходится в среднем от 6 до 16 ошибок. По другим данным этот показатель может доходить до 25 и даже выше для определенного типа программ (для драйверов внешних устройств). Таким образом, пренебрегать их наличием нельзя, поскольку последствия отказов для устройств РЗА могут иметь серьезные последствия.

Современные устройства РЗА с программной логикой представляют собой систему из двух неотделимых друг от друга частей – аппаратной и программной. Аппаратная часть характеризуется надежностью физического элемента и имеет распределенные во времени этапы приработки (этап со сниженной надежностью в начале эксплуатации), нормальной эксплуатации и износа (когда надежность неизбежно снижается). Ненадежность компонентов устройства РЗА связана с их старением (на этапе износа) или ошибками персонала.

Особенность надежности программного обеспечения заключается в том, что она не изнашивается[3], надежность ПО со временем лишь увеличивается при наличии системы исправления ошибок. Причем ненадежность связана с ошибками проектировщиков и иного персонала на различных стадиях разработки ПО. Таким образом, интенсивность отказов для ПО имеет экспоненциальный закон – см. рисунок 1.

[2] СО 153-34.35.516-89. Инструкция по учету и оценке работы релейной защиты и автоматики электрической части энергосистем», ПО «Союзтехэнерго», 1989
[3] ГОСТ 28806-90 «Качество программных средств. Термины и определения», 1990

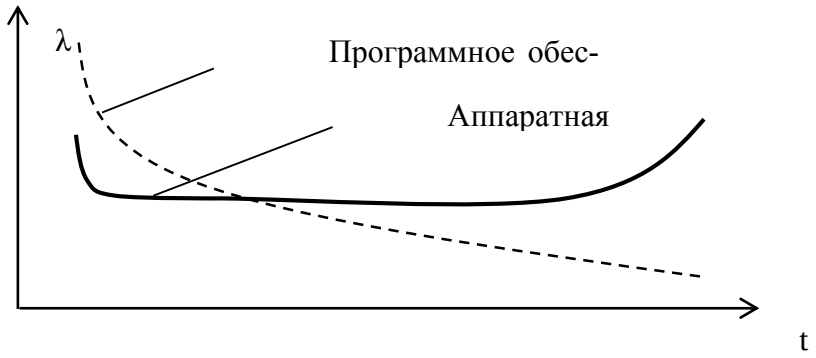

Рисунок 1. *Вид функции интенсивности отказов для ПО и аппаратной части*

В случае отсутствия системы восстановления после отказа по причине ненадёжности ПО, количество и характер этих отказов зависят от способа применения программного средства и от выбираемых вариантов его функционирования, но не зависят от времени. Применительно к системе РЗА – отказ проявляется при формировании непредусмотренного при проектировании набора входных данных.

На надежность ПО влияет используемый язык программирования. Стандарт IEC 61131-3 описывает языки программирования ПЛК. Каждый производитель ПЛК сопровождает свой продукт собственной средой программирования, которая, как правило, не совместима с другими. Это вызывает необходимость введения дополнительных программ конверторов для организации взаимодействия устройств различных производителей (использование одного стандарта обмена данными так же не гарантирует отсутствия этих подпрограмм). А как известно введение дополнительного элемента всегда негативно сказывается на надежности.

В РД[4] указано, что пользователь должен иметь возможность конфигурирования прикладного ПО (выбирать различные варианты взаимодействия с внешними устройствами и режимами объекта защиты, вводить в работу дополнительные функции). Однако возможность выбора конфигурации системного ПО не предусматривает – покупая устройство РЗА пользователь не может воспользоваться операционной системой стороннего производителя. Это в настоящий момент оправдано, т.к. каждый производитель имеет свою аппаратную основу (свой тип микропроцессора, свою архитектуру устройства и т.д.)

Оправдан путь развития открытой аппаратной архитектуры –аппаратная основа является стандартной и предоставляется фирмой-изготовителем, а пользователь имеет возможность выбора алгоритма (т.е. программы) функционирования у любого из производителей. Это позволит комбинировать наиболее надежные и эффективные алгоритмы с высокой аппаратной надежностью. А потребитель сможет выбрать оптимальный вариант по надежностно-ценовым качествам.

При развитии системы РЗА по такому сценарию в процессе эксплуатации важно оценивать функциональную надежность, как совокупность аппаратной, программной надежности, а также точности обеспечения рабочих настроек прибора. Последняя составляющая относится к человеческому фактору и связана, например, с ошибками расчетов уставок или ввода данных.

[4] РД 34.35.310-97 Общие технические требования к микропроцессорным устройствам защиты и автоматики энергосистем

Способов строгой методики оценки программной надежности в настоящий момент нет. При этом расчет надежности должен относиться к разным этапам жизненного цикла программного продукта. При разработке важно определить динамику изменения показателя для определения длительности интервала тестирования (повышение надежности может быть произведено именно на этом этапе - [9]), для проектирования важна статическая характеристика для определения оптимального варианта исполнения и т.д.

В общем случае надежность программного обеспечения определяется числом ошибок, имеющихся в коде. Под ошибкой подразумевается неправильность, погрешность или неумышленное искажение объекта или процесса, что может быть причиной ущерба – риска при функционировании или применении программы.

Ошибка может быть заложена на этапе проектирования – могут быть не предусмотрены некоторые входные данные, которые могут вызвать неправильную работу программы. Т.е. ошибка является не явной. В этом случае вероятность ошибки характеризуется вероятностью возникновения непредусмотренного набора входных данных – вероятностью инициирующего события $P_{ИН.С.}$, и числом неявных ошибок $n_{НЕЯВ.ОШ.}$

Кроме того нельзя исключать наличия ошибок кодирования – явных ошибок (например, вместо положительного знака числа заложен отрицательный). В этом случае ошибка будет возникать каждый раз при запуске модуля. И вероятность неправильной работы по причине явных ошибок определяется числом явных ошибок $n_{ЯВ.ОШ.}$ и частотой запуска модуля ν.

В системе программно-аппаратной надёжности так же выделяют человеческий фактор. В структуре разработки программного обеспечения этому компоненту надежности можно сопоставить этап документирования. Не всегда документация точно и полно отображает все аспекты эксплуатации. В связи с этим могут возникать ошибки по причине человеческого фактора. Конечно, они обусловлены не только недостатками документации, часто сказываются также факторы уровня обучения персонала, сложности интерфейсов и т.д. По типам ошибок в ПО процентные частоты их появления согласно [6] представлены Таблица **1**1**.**

Таблица 1

Частота появления ошибок в ПО

Тип ошибки	Частота появления, %
Не полная или ошибочная спецификация	28
Отклонение от спецификации	12
Пренебрежение правилами программирования	10
Ошибочная выборка данных	10
Ошибочная логика или последовательность операций	12
Ошибочные арифметические операции	9
Нехватка времени для решения	4
Ошибка обработки прерываний	4
Ошибка в исходных данных	3
Неточная запись	8

Как видно большинство ошибок приходится на этап проектирования, из-за неверно составленного ТЗ и спецификации.

Наличие ошибок в ПО не может быть выявлено до проявления отказа (аварийное завершение программы, некорректное действие защиты, зависание и т.д.). Т.е. до тех пор, пока

не будут сформированы входные данные определенного типа. Поскольку формирование таких данных является случайным процессом, то можно предположить, что распределение времени между появлениями отказов (по причине ошибок в ПО) имеют экспоненциальное распределение.

При этом следует учитывать, что в процессе эксплуатации число ошибок в программном обеспечении зачастую остается постоянным. При неправильной работе терминала либо происходит его самостоятельная новая инициализация или персонал осуществляет его перезагрузку. С другой стороны, в ходе профилактик могут быть выявлены программные ошибки [10], и терминал будет отправлен на завод-изготовитель на доработку. Т.е. могут быть рассмотрены две модели: с уменьшающимся и постоянным числом ошибок.

При этом после устранения выявленной ошибки могут автоматически исправиться не выявленные или появиться новые, т.е. после устранения ошибки следует рассматривать новый программный продукт со своими характеристиками.

Модель ошибки сводится к наличию определенных входных данных при которых происходит некорректное срабатывание того или иного модуля – как при последовательном, так и при параллельном исполнении операторов. При параллельном исполнении операторов так же может возникнуть непредвиденная ситуация – ошибка синхронизации. Эта модель представляет ПО в виде черного ящика. Использование ее на практике ограничено, т.к. объем возможных входных данных очень существенен и зачастую не поддается точному моделированию.

Для анализа надежности ПО важно определить набор его состояний и переходов между ними. В этом случае для оценки применим марковский процесс. Переход из состояния в состояние осуществляется под действием потоков управления – внешних потоков (потоков КЗ, повреждаемости защищаемого элемента и т.д.) и внутреннего предопределенного потока управления. Потоки можно признать ординарными, т.к. всегда можно выделить малый период времени, на котором происходит только одно событие. Кроме того для него характерно отсутствие последействия, что следует из предложенной модели. Таким образом, переход из состояния в состояние осуществляется под действием пуассоновского потока, что обуславливает возможность описания модели дискретным марковским процессом.

Перед анализом должны быть приняты некоторые допущения:

1) Программа состоит из n модулей;
2) Время перехода от модуля к модулю случайно;
3) Отказ каждого модуля не приводит к отказу других;
4) Ведется исследование программной надежности без отношения к отказам оборудования.

Программные средства устройства РЗА могут быть классифицированы на следующие группы:

- поддержки и ввода информации – отбор и верификация данных;
- анализа и обработки результатов измерений – цифровая фильтрация и вычисление векторов;
- фиксации пуска РЗА;
- обработка события пуска для каждой ступени;
- фиксация и отображения информации – запись событий, сообщений, осциллограмм, интерфейсы ввода-вывода;

- связи с вышестоящим уровнем управления;
- самодиагностирования;
- база данных – конфигурация, параметры, уставки.

Для функций РЗА значение имеют не все представленные модули. В качестве объекта для анализа выберем функцию трехступенчатой токовой защиты, тогда схема взаимодействия модулей задействованных в исполнении функций релейной защиты будет следующей:

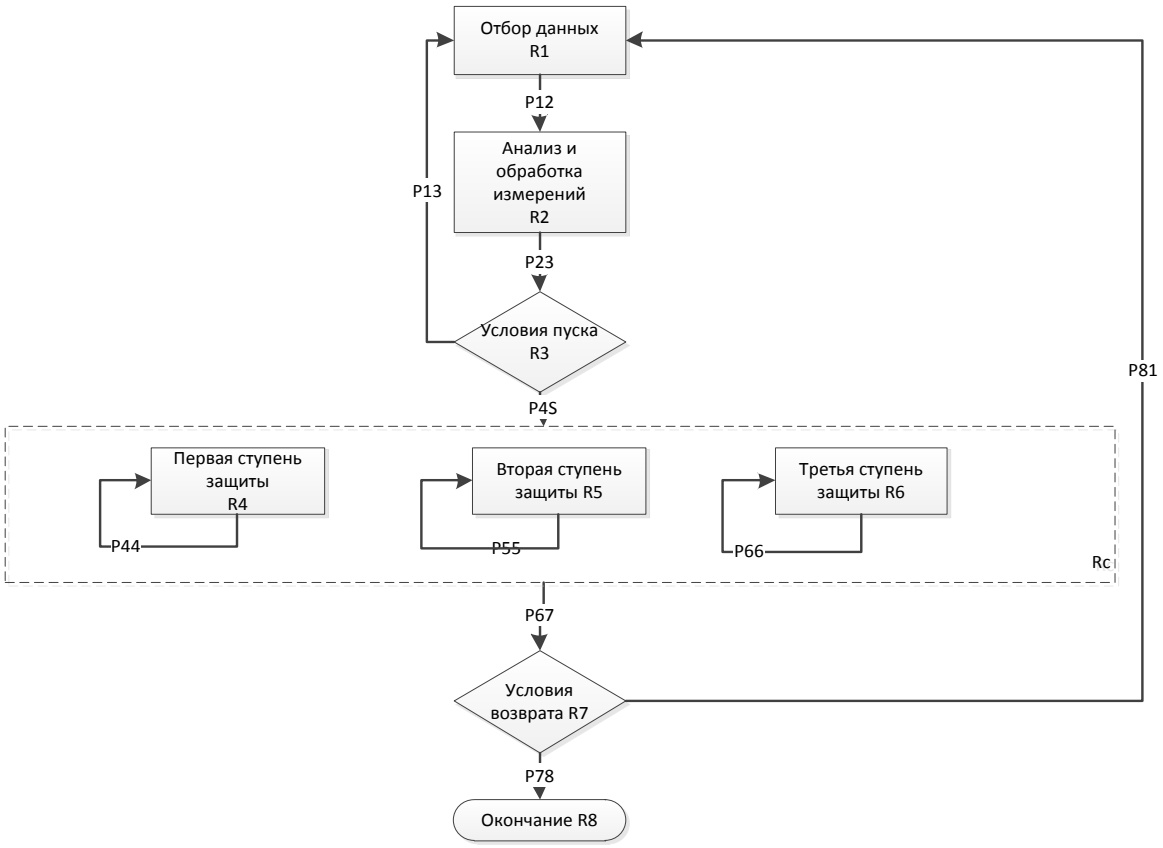

Рисунок 2. *Архитектура модулей РЗА*

Для представленной архитектуры можно выделить параллельную архитектуру для модулей срабатывания 1-ой, 2-ой и 3-ей ступеней защиты. Т.к. программа единожды сравнивает значение измерений со всеми имеющимися в БД уставками, то работу ступеней защиты можно считать параллельной. Наличие перехода на самого себя для блоков ступеней связано с наличие уставки по времени для каждой ступени. Примем надежность модулей согласно [3]:

Таблица 2

Надежность модулей

R_1	R_2	R_3	R_4	R_5	R_6	R_8	R_9
0.986	0.985	0.994	0.988	0.988	0.988	0.985	0.995

Отметим, что надежность модулей в большинстве определяется внутренними характеристиками ПО и основывается на статистических данных. В то время как переходы от модуля к модулю Pij определяются не только характеристиками защиты, но и объекта на котором он установлен. При анализе необходимо выявить наихудшие условия для эксплуатации ПО и опираться в расчетах на полученную цифру.

Для параллельной архитектуры:

$$R_c = R_4 \cdot R_5 \cdot R_6 = 0{,}988^3 = 0.964$$

Таблица 3

Вероятности переходов

P_{12}	1 (т.к. переход единственный)
P_{23}	1 (т.к. переход единственный)
P_{31}	Согласно статистическим данным частота возникновения события требования срабатывания для МТЗ – 25%
P_{3C}	75%
P_{47}	1 – т.к. первая ступень работает без выдержки времени
P_{57}	1 - P_{55} = 1-0,05=0,95
P_{67}	1 - P_{66} = 1-0,1=0,9
P_{C7}	$P_{47} \cdot P_{57} \cdot P_{67}$=1·0,95·0,9=0,855
P_{44}	0 – т.к. первая ступень работает без выдержки времени
P_{55}	Согласно документации уставка по времени в среднем в 1/0,1=10 раз меньше чем для четвертой ступени, то 0,5/10=0,05
P_{66}	Согласно документации уставка по времени в среднем в 1/0,2=5 раз меньше, чем для четвертой ступени, то 0,5/5=0,1
P_{78}	Переход в состояние моделирования или окончания согласно отчету CIGRE [5] и [3] составляет 0,55
P_{71}	1 - P_{89} = 1 – 0,55 = 0,45

[5] Analysis and Guidelines For Testing Numerical Protection Schemes Final Report March CIGRE 34.10, 2000

Тогда получившаяся матрица переходов:

$$M := \begin{pmatrix} 0 & R_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_2 & 0 & 0 & 0 \\ R_3 \cdot 0.75 & 0 & 0 & R_3 \cdot 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_4 \cdot 0.855 & 0 \\ R_5 \cdot 0.45 & 0 & 0 & 0 & 0 & R_5 \cdot 0.55 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$ или, $$M = \begin{pmatrix} 0 & 0.986 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.985 & 0 & 0 & 0 \\ 0.7455 & 0 & 0 & 0.2485 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.82422 & 0 \\ 0.44325 & 0 & 0 & 0 & 0 & 0.54175 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

Тогда суммарная надежность ПО:

$$R_h := L_{1,n} \cdot R_n = 0.57099$$

где

$$L := (I - M)^{-1} = \begin{pmatrix} 5.32507 & 5.25052 & 5.17177 & 1.28518 & 1.05927 & 0.57386 \\ 4.38649 & 5.32507 & 5.2452 & 1.30343 & 1.07431 & 0.58201 \\ 4.45328 & 4.39094 & 5.32507 & 1.32328 & 1.09067 & 0.59087 \\ 1.94544 & 1.9182 & 1.88943 & 1.46952 & 1.21121 & 0.65617 \\ 2.36034 & 2.32729 & 2.29239 & 0.56966 & 1.46952 & 0.79611 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Вероятность перехода в конечное состояние составляет– 57,1%.

Для исследования влияния надежностей модулей на надежность ПО в целом примем надежность всех равной 1 и снизим на 10% надежность каждого при фиксированной надежности остальных. Получим следующую зависимость:

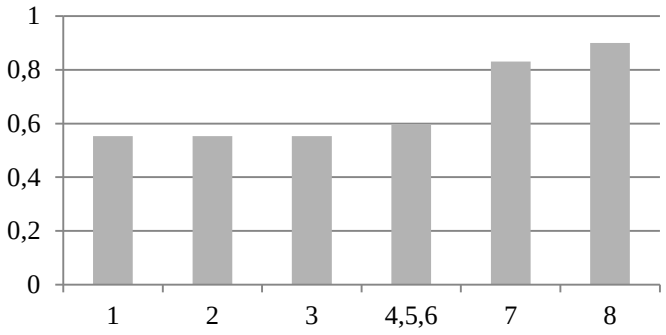

Рисунок 3. *Анализ чувствительности надежности ПО по отношению к Ri*

Как видно, наиболее критичными являются параметры 1-6 модулей. Это кажется очевидным, т.к. модуль измерений и моделирования (первый модуль) предоставляет данных для всех блоков программы. Модуль оценки срабатывания важен для разграничения функций срабатывания и несрабатывания.

Таким образом, для современных устройств РЗА имеет существенное значение функциональная и программная надежность, как ее составляющая, что подтверждается статистическими данными эксплуатации. Для ее оценки можно использовать несколько моделей: например, модель «черного ящика» или дискретный марковский процесс. Последний метод позволяет охарактеризовать надежность модуля в целом, а также критичность каждого из них. Хотелось бы отметить важность учета программной надежности в процессе эксплуатации и ввести требование обязательной «перепрошивки» устройств при проверках и тестированиях. Это требование связано не только с ошибками, обнаруженными в процессе эксплуатации, но и увеличивающимся разнообразием устройств РЗА, взаимодействие с которыми может потребоваться.

ЛИТЕРАТУРА

1. Захаров О.Г. Надежность цифровых устройств релейнойзащиты. Показатели. Требования. Оценки – М.:Инфра-инженерия, 2014.–128с. ISBN978-5-9
2. Шалин А.И. Надежность и диагностика релейной защиты энергосистем. Новосибирск, НГТУ, 2002, 384 с.
3. Li Xiao-hua, Wang Gang, Xiao Lin, Ding Maosheng Reliability Analysis of Digital Protection's Software Based on Architecture IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China, 2005
4. Кузьмичев В.А., Коновалова Е.В., Сахаров С.Н., Захаренков А.Ю. Ретроспективный анализ работы устройств РЗА в ЕНЭС // Релейная защита и автоматизация. 2012. №1 (06). С.60-65.
5. Владимир Гуревич, к. т. н. Новая концепция построения микропроцессорных устройств релейной защиты // Компоненты и технологии, № 6, 2010 с. 93-96
6. Липаев В.В. Функциональная безопасность программных средств. – М.: СИНТЕГ, 2004. – 340
7. Шнеерсон Э.М. Цифровая релейная защита. М.:Энергоатомиздат, 2007, 549 с.
8. Thomas J. Ostrand, Elaine J. Weyuker.The distribution of faults in a large industrial software system , ISSTA '02 Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis
9. М.В. Ширяев, К.Ю. Лавров, О.Н. Андреева. Совершенствование способа повышения надежности специального программного обеспечения встраиваемых систем // Нейрокомпьютеры: разработка, применение, № 5, 2014г. с. 38-45.
10. С.Н. Колмогорцев, А.Н. Владимиров. Программный комплекс для анализа и устранения неисправностей релейной защиты и автоматики // Сборник докладов XXI конференции «Релейная защита и автоматизация энергосистем», 29-31 мая 2012, с. 58-64

Рецензент: Никифоров Владимир Анатольевич, доцент кафедры МИТЭ Смоленский филиал МЭИ, кандидат технических наук, доцент.

Pevtsova Ludmila Sergeevna
Moscow Power Engineering Institute
Russia, Smolensk
E-mail: plus_energo@mail.ru

Evaluating software reliability of digital relays

Abstract. Currently there is a replacement electromechanical and static relay by the digital relays. this process begin at the end of last century and today we can inventory the progress in this field. so reliability analysis of relay protection show that used algorithms have appreciable influence to relia-bility of the digital devices in whole. this paper provides an analysis causes of variable failure, reli-ability analysis software, as well as propose a method for calculating the reliability of software using random discrete markov's circuit.

Keywords: digital relays; hardware reliability; software reliability; reliability indicator; the error rate in the software; discrete markov’s circuit; overcurrent protection.

REFERENCES

1. ZaharovO.G. Nadezhnost' cifrovyh ustrojstv relejnojzashhity. Pokazateli. Trebovanija. Ocenki – M.:Infra-inzhenerija, 2014.–128s. ISBN978-5-9
2. Shalin A.I. Nadezhnost' i diagnostika relejnoj zashhity jenergosistem. Novosibirsk, NGTU, 2002, 384 s.
3. Li Xiao-hua, Wang Gang, Xiao Lin, Ding Maosheng Reliability Analysis of Digital Protection's Software Based on Architecture IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China, 2005
4. Kuz'michev V.A., Konovalova E.V., Saharov S.N., Zaharenkov A.Ju. Retrospek-tivnyj analiz raboty ustrojstv RZA v ENJeS // Relejnaja zashhita i avtomatizacija. 2012. №1 (06). S.60-65.
5. Vladimir Gurevich, k. t. n. Novaja koncepcija postroenija mikroprocessornyh ustrojstv relejnoj zashhity // Komponenty i tehnologii, № 6, 2010 s. 93-96
6. Lipaev V.V. Funkcional'naja bezopasnost' programmnyh sredstv. – M.: SIN-TEG, 2004. – 340
7. Shneerson Je.M. Cifrovaja relejnaja zashhita. M.:Jenergoatomizdat, 2007, 549 s.
8. Thomas J. Ostrand, Elaine J. Weyuker.The distribution of faults in a large industrial software system , ISSTA '02 Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis
9. M.V. Shirjaev, K.Ju. Lavrov, O.N. Andreeva. Sovershenstvovanie sposoba po-vyshenija nadezhnosti special'nogo programmnogo obespechenija vstraivaemyh sistem // Nejrokomp'jutery: razrabotka, primenenie, № 5, 2014g. s. 38-45.
10. S.N. Kolmogorcev, A.N. Vladimirov. Programmnyj kompleks dlja analiza i ustranenija neispravnostej relejnoj zashhity i avtomatiki // Sbornik dokladov XXI konferencii «Relejnaja zashhita i avtomatizacija jenergosistem», 29-31 maja 2012, s. 58-64