DOI: 10.15514/ISPRAS-2019-31(5)-7

Методы оценки надежности программных и технических систем

[1,2] *Е.М. Лаврищева, ORCID: 0000-0002-1160-1077 <lavr@ispras.ru>*
[1,3] *С.В. Зеленов, ORCID: 0000-0003-0446-0541 <zelenov@ispras.ru>*
[4] *Н.В.Пакулин, ORCID: 0000-0003-1266-7737<nikolay@paxdatatech.com>*
[1]*Институт системного программирования им. В.П. Иванникова РАН,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25
[2] *Московский физико-технический институт,*
141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9
[3] *НИУ Высшая школа экономики,*
101978, Россия, г. Москва, ул. Мясницкая, д. 20
[4] *Pax Datatech PTE. Ltd.,*
051531, Сингапур, Hong Lim Complex, Upper Cross Street, 531A

Аннотация. Определяются основные методы обеспечения и оценки надежности и безопасности программно-технических систем в процессах их жизненного цикла, а также сбора сведений о возникающих в системах ошибках, дефектах и отказах для последующих изменений. Рассматривается стандартная модель надежности и дается характеристика базовых показателей, среди которых присутствует показатель надежности; функциональность и безопасность составляют основу измерения надежности. Приводится классификация моделей надежности, дается характеристика моделей оценочного типов, используемых при проверке показателей надежности компонентов программно-технических систем. Обсуждаются экспериментальные результаты применения оценочных моделей надежности к разным размерам программных компонентов программно-технических систем и приводится оценка результатов измерения показателя надежности на этих компонентах с учетом плотности дефектов, интенсивности отказов и восстанавливаемости. Отмечается важность обеспечения надежности и безопасности (dependability and safety) систем в рамках новых стандартов интеллектуальных систем и Интернета вещей.

Ключевые слова: надежность; ошибка; дефект; отказ; безопасность; тестирование; надежность; риск; умные компьютеры.

Для цитирования: Лаврищева Е.М., Зеленов С.В., Пакулин Н.В. Методы оценки надежности программных и технических систем**.** Труды ИСП РАН, том 31, вып. 5, 2019 г., стр. 95-108. DOI: 10.15514/ISPRAS-2019-31(5)-7

Благодарности. Работа поддержана грантом РФФИ 19-01-00206.

Methods for assessing the reliability of software and hardware systems

[1,2] *E.M. Lavrischeva, ORCID: 0000-0002-1160-1077 <lavr@ispras.ru>*
[1,3] *S.V. Zelenov, ORCID: 0000-0003-0446-0541 <zelenov@ispras.ru>*
[4] *N.V. Pakulin, ORCID: 0000-0003-1266-7737<nikolay@paxdatatech.com>*
[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *Moscow Institute of Physics and Technology (State University),*
9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia
[3] *National Research University, Higher School of Economics*
20, Myasnitskaya Ulitsa,Moscow, 101978, Russia
[4] *Pax Datatech PTE. Ltd.,*
531A Upper Cross Street, Hong Lim Complex, Singapore, 051531

Abstract. At first, the paper introduces the main methods for ensuring and assessing the reliability and safety of software and hardware systems in the processes of their life cycle, as well as collecting information about errors that occur in systems, defects and failures for subsequent changes. Then we consider a standard reliability model and provide a characteristic of basic indicators that include a reliability indicator; functionality and safety form the basis for measuring reliability. After this, the paper demonstrates the classification of reliability models, and shows the characteristics of the evaluation types used in the verification of reliability indicators of components of software and hardware systems. Next, we discuss the experimental results of applying the estimated reliability models to different sizes of software components of software and hardware systems as well as the results of measuring the reliability indicator on these components taking into account the density of defects, failure rates, and recovery. Finally, we note the importance of ensuring the reliability and safety (dependability and safety) of systems in the framework of the new standards of intelligent systems and the Internet of things.

Keywords: reliability; error; defect; failure; security; fault; completeness; testing; reliability; risks; smart computers

For citation: Lavrischeva E.M., Zelenov S.V., Pakulin N.V. Methods for assessing the reliability of applied systems. Trudy ISP RAN/Proc. ISP RAS, vol. 31, issue 5, 2019, pp. 95-108 (in Russian). DOI: 10.15514/ISPRAS-2019-31(5)-7

Acknowledgements. This work was supported by RFBR, project 19-01-00206.

1. Введение

Надежность систем – это теоретическая и прикладная наука, способствующая созданию надежных и безопасных систем и проведению измерения отдельных показателей качества (функциональность, надежность, безопасность и др.) технических и программных систем (ПТС). Методы надежности позволяют прогнозировать, измерять и оценивать качество продукта, доводя ошибки, дефекты и интенсивность отказов в компонентах систем к минимуму [1-9].

В проблеме обеспечения надёжности ПТС имеется некоторые общие положения – это возникновение случайных явлений и способы их анализа с применением теории вероятности. К системам реального времени, включая радарные системы, системы безопасности, медицинские, производственные системы и оборудование, предъявляются высокие требования к надежности (недопустимость ошибок, достоверность, защищенность и др.).

Надежность систем зависит от числа оставшихся и не устраненных ошибок. Чем интенсивнее проводится эксплуатация, тем интенсивнее выявляются ошибки и быстрее растет надежность системы [1-14]. В проблеме обеспечения надёжности ПТС лежат

случайные явления (аварии, киберугрозы, пожары и др.), которые требуют принятия серьезных мер по прогнозированию рисков и мер противодействия возникающим угрозам.

Согласно ФЗ «О безопасности», ГОСТ Р 22.0.02 и недопустимости риска ГОСТ Р 51898-2002, ГОСТ 1.1-2002 и др. в области системной инженерии требуется обеспечение качественных систем; функциональных и программно-технических интерфейсов разных видов систем; безопасности и защищенности общественных и государственных систем от внутренних и внешних угроз; работоспособности и конкурентности компьютерных систем с прогнозированием рисков и принятием сбалансированных мер, упреждающих противодействиям авариям, сбоям, киберугроз и др. [15-20].

Одним из важных условий безопасной работоспособности компьютерных систем является надежность, которая зависит от оставшихся и не устраненных ошибок в отдельных ее компонентах. Чем интенсивнее проводится эксплуатация, тем интенсивнее выявляются ошибки и быстрее растет надежность системы.

Для оценки надежности систем используются собранные статистические данные о вероятности и времени безотказной работы; отказы и частота (интенсивность) отказов, защита от искажений программ и данных, прогнозирования характеристик надежности на процессах жизненного цикла (ЖЦ) с учетом функциональности и безопасности.

В ГОСТ 27.002-2015 надежность определяется как свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, хранения и транспортирования.

Главными источниками информации об ошибочных ситуациях для оценки надежности являются процессы ЖЦ (задания требований, проектирование, тестирование, эксплуатация и испытание), например, стандарты ЖЦ (ISO/IEC 15846-1998, 15939:2002 и др.) [17-20].

На всех процессах ЖЦ могут возникать случайные ситуации:

- *отказ ПС (failure)* – это переход системы из рабочего состояния в нерабочее;
- *дефект (fault)* – это последствие выполнения элемента программы, приводящее к некоторому событию, например, в результате неверной интерпретации его компьютером или человеком; дефекты в программе, не выявленные в результате проверок, является источником потенциальных ошибок и отказов системы;
- *ошибка (error)* как следствие недостатка в описании одной из программ или при принятии неверных решений.

На процессах ЖЦ набирается статистика отказов, их интенсивность и обнаруживаются дефекты в компонентах системы. Проявляющиеся дефекты требуют создания средств защиты от случайных искажений программ и данных. Они оцениваются после проведения испытания системы и влияют на получение количественного коэффициента надежности системы [10-17].

Надежность функционирования системы характеризуется устойчивостью, восстанавливаемостью и работоспособностью системы. Непредусмотренные при проектировании ошибки и дефекты влияют на безопасность функционирования системы. Кроме того, возникают угрозы (отказы, катастрофические явления), которые влияют на функциональную безопасности системы и показатели надежности системы.

При обеспечении безопасности системы учитываются только те отказы, которые могут привести к катастрофическим последствиям и ущербам (например, пожар, взрыв, разрушение зданий и др.). Оценка надежности и безопасности функционирования системы от метрик стандарта качества (внешние, внутренние, эксплуатационные). Они сравниваются с требованиями на систему и используются при сертификации готовой системы.

Для оценки надежности и функциональной безопасности следует использовать стандарт ISO 15408 - 2008, IEC 61508, согласно которым требуется проведение защиты аппаратуры от всех видов угроз и устранение возникающих неисправностей при дефектах, сбоях и отказах.

2. Методы обеспечения безопасного и надежного функционирования систем

Процесс возникновения ошибок и отказов в ПТС определяется временем их возникновения или частотой, числом и их интенсивностью. В связи с этим поиск случайных величин осуществляется стохастическими методами или вероятностными. Если случайные величины распределены по показательному, эрланговскому или гиперэрланговскому законам, то поведение системы описывается Марковским процессом.

В основном модели оценки надежности основываются на статистике отказов и распределении интенсивности выявленных отказов в ПТС. Некоторые модели надежности исходят из предположения, что найденные дефекты устраняются немедленно (или временем их устранения можно пренебречь) и при этом новые дефекты не вносятся. При этом количество дефектов в системе уменьшается, а надежность возрастает и такие модели получили название *моделей роста надежности*.

К наиболее распространенным методам обеспечения безотказной и безопасной работы систем относятся модели Джелинского-Моранды, Нельсона, Мусы, Вейса и др. [1-3]. Сформировалась классификация моделей надежности: прогнозирующие, измерительные и аналитические.

Большинство моделей оценки надежности базируются на дефектах, статистике отказов и распределении их интенсивности в ПС. Найденные дефекты устраняются немедленно и новые дефекты не вносятся. В результате количество дефектов в ПС уменьшается, а надежность возрастает. Такие модели получили название моделей роста надежности.

Известна следующая классификация моделей надежности: прогнозирующие, измерительные и оценочные.

Прогнозирующие модели надежности основаны на измерении технических характеристик создаваемой программы: длина, сложность, число циклов и степень их вложенности, количество ошибок на страницу операторов программы и др. (например, модель Мотли–Брукса).

Измерительные модели предназначены для измерения надежности ПО, работающего с заданной внешней средой и следующими ограничениями: при измерении свойств надежности ПС не изменяется; обнаруженные ошибки не исправляются; оценка надежности проводится для зафиксированной конфигурации ПО. Примером таких моделей является модель Нельсона и Рамамурти–Бастани и др.

Оценочные модели основываются на серии тестовых прогонов и проводятся на ЖЦ тестирования ПС. Эти модели могут быть без подсчета ошибок позволяют спрогнозировать количество ошибок, оставшихся в программе. К таким моделям относятся модели Джелински и Моранды, Шика Вулвертона и Литвуда–Вералла. К моделям с подсчетом отказов на заданных интервалах времени относятся: модели Шика–Вулвертона, Шумана, Пуассоновская модель и др.

При внесении изменений в программу проводится повторное тестирование и оценка надежности. Этот подход базируется на тестировании и генерации множества тестовых выборок из входного распределения (например, модель Нельсона и др.)

По фактору распределения интенсивности отказов модели надежности подразделяются на экспоненциальные, логарифмические, геометрические, байесовские и др. В них проблема

надежности ПО рассматривается как способность системы быть работоспособной (dependability) с атрибутами (свойствами):

- availability – готовность к использованию;
- reliability – готовность к непрерывному функционированию;
- safety – безопасность для окружающей среды (для внешнего окружения). Способность не вызывать катастрафических последствий в случае отказа;
- confidentability – секретность, сохранение секретности информации;
- integrity – способность к сохранению информации, устойчивость к её самопроизвольному изменению;
- aintainability – эксплуатационные способности ПО, простота выполнения операций обслуживания (например, устранение ошибок, восстановление после ошибки и т.п.);
- security – готовность, сохранность и скрытность (confidentability);
- failure – отказ, отклонение поведения системы от предписанного, т.е. когда система перестаёт выполнять предписанные ей функции;
- error – ошибка, состояние системы, которое вызывает отказ (mistake);
- fault – отказ, в случае причины ошибки, что вызывает её;

Достижение работоспособности обеспечивается многими методами, которые включают:

- fault preventation – предотвращение отказа,
- remival fault – устранение отказа,
- fault tolerance – возможность выполнения ПО при наличии ошибки,
- fault forecasting – возможность появления отказа и его последствия для оценки.

Различие между fault и failure не критическое и поэтому используется термин defect может означать либо fault (причина), либо failure (действие).

Таким образом, оценка надежности системы осуществляется по тем моделям надежности, которые соответствуют типу анализируемой системы. Если обнаружены ошибки и внесены необходимые изменения в нее, проводятся такие действия:

- протоколирование отказов в ходе функционирования ПС и измерение надежности функционирования, а также использование результатов измерений при определении потерь надежности в период времени эксплуатации;
- анализ частоты и серьезности отказов для определения порядка устранения соответствующих ошибок;
- оценка влияния функционирования ПС на надежность в условиях усовершенствования технологии или использования новых инструментов разработки ПС.

Таким образом, показано, что надежность является одной и главных характеристик современных ПТС, для которой разработано большое количество моделей для разных ее видов и типов. Рассмотрены основные базовые понятия надежности, обеспечивающие оценку надежности по соответствующим моделям надежности ПТС, основанным на времени функционирования и/или количестве отказов (ошибок), полученных в программах в процессе их тестирования или эксплуатации.

3. Эксперименты по применению моделей надежности.

Были проведены эксперименты для моделей *оценочного* типа, которые базируются на пуассоновских процессах (модели Мусы, Гоэла–Окомото, S-образные и др.). Эксперименты проводились для небольших (1), средних (2), больших (3) и очень больших (4) проектов [2-5, 15].

Таблица 1. Характеристики моделей надежности Пуассоновского типа для вычисления интенсивности и количества отказов
Table 1. Characteristics of Poisson type reliability models for calculating the intensity and number of failures

Название модели	Функции интенсивности отказов $\lambda(t)$	Функция количества отказов $\mu(t)$
Модель Гоэла-Окумото	$\lambda(t) = Nb\ exp(-bt)$	$\mu(t) = N(1 - exp(-bt))$
Модель Мусы	$\lambda(t) = \beta_0 \beta_1\ exp(-\beta_1 t)$	$\mu(t) = \beta_0(1 - exp(-\beta_1 t))$
S –подобная модель	$\lambda(t) = \alpha \beta^2 t\ exp(-\beta t)$	$\mu(t) = \alpha\{1 - (1 + \beta t) exp(-\beta t)\}$
Модель Шнайдевинда	$\lambda(t) = \alpha_0\ exp(-\beta t)$	$\mu(t) = \alpha_0/\beta(1 - exp(-\beta t))$
Общая модель пуассоновского процесса	$\lambda(t) = \alpha \beta^{n+1} t^n\ exp(-\beta t)$	$\mu(t) = \alpha\left(n^1 - \sum b\beta^{n-1}/(n-1)^1 t^n exp(-\beta t)\right)$

В табл. 1 представлены практически полученные значения функций интенсивности отказов $\lambda(t)$ и количество отказов $\mu(t)$ для проектов (1-4) с помощью названных в табл. 1 моделей надежности. В них значения α и β находятся в следующих соотношениях: $N = \alpha, \beta = \alpha, b = \beta, \beta_1 = \beta, \alpha_0 = \alpha\beta$. Параметр n зависит от процесса тестирования и его значений:

- n=0 для небольшого проекта, в котором разработчик является также тестером (модели Мусы, Гоэло-Окомото и др.);
- n=1 для среднего проекта, в котором тестирование и проектирование ПО исполняются несколькими разработчиками из одной рабочей группы (S- образная модель);
- n=2 для большого проекта, в котором группы разработчиков работают параллельно;
- n=3 для очень большого проекта, в котором группы тестирования и разработки работают независимо друг от друга.

На основе проведенных экспериментальных данных получены функции о количестве отказов $\mu(t)$ и интенсивности отказов $\lambda(t)$ на выходных данных и значениях параметра n, который показывает вид функций $\mu(t)$ при разных значений n = 0, 1, 2, 3.

Наибольшее приближение достигается при n=3, а наименьшее при n = 0 (модель Мусы, Гоэло-Окомото и др.). По этим моделям надежность стремиться к 1. Одним из недостатков является форма кривой интенсивности выявленных отказов или неисправностей (экспоненциальная) и строго спускается при $t > 0$. Это свидетельствует о том, что при тестировании проведено недостаточно экспериментов или мало найдено ошибок, когда интенсивность отказов была близка 0. В системе могут оставаться ошибки и их поиск требует больше времени.

При применении S-образной модели, функция интенсивности $\lambda(t)$ выявления ошибок в зависимости от времени работы имеет вид: $\lambda(t) = \alpha \beta^2 t\ exp(-\beta t)$, где α – общее количество дефектов, обнаруженных от начала и до конца тестирования; β – скорость изменения функции интенсивности при выявлении отказов.

Таблица 2. Статистические данные для μ(t) при n=3, 2, 1 и данных t_2
Table 2. Statistics for μ(t) with n=3, 2, 1 and t_2 data

Статистические показатели отклонений	Разница функций *t_2 - μ_3*	Разница функций *t_2 - μ_2*	Разница функций *t_2 - μ_1*	Разница функций *t_2 - μ*
Среднее	16.13522	16.22889	19.88387	58.93807

Статистические показатели отклонений	Разница функций $t\_2 - \mu\_3$	Разница функций $t\_2 - \mu\_2$	Разница функций $t\_2 - \mu\_1$	Разница функций $t\_2 - \mu$
Медианное	15.27700	14.11600	16.0000	60.89700
Максимум	33.58100	54.23600	49.10800	88.80200
Минимум	4. 848000	-1.280000	41.75000	15.96200
Среднеквадратическое отклонение	8.374089	17.37143	14.07056	23.63765

Введение в формулу параметра в степени 1 модели Мусы и Гоэла–Окомото дает изменение формы кривой так, что она сначала растет, а потом спадает. Практика применения этих моделей в ПТС привела к уточнению функции интенсивности при введении дополнительного параметра n: $\lambda(t) = \alpha\beta^{n+1}t^n exp(-\beta t)$, где n отражает сложность и размер проекта системы. Это позволяет более точно определить форму кривой интенсивности с учетом получаемых практических результатов.

Результат экспериментов подтверждается соответствующими статистическими данными (табл. 2), которые задают разницу между выходными данными (t_2) и соответствующими значениями функции $\mu(t)$ при значениях n = 0, 1, 2, 3. На основе экспериментальных данных α, β, n, приведены значения функций $\mu(t)$ и $\lambda(t)$ при n = 3, 2, 1, полученные при использовании методов оценки надежности Мусы, Мусы-Окомото и Шнайдевинда.

4. Процессы ЖЦ для разработки компонентов систем

Многие современные системы, работающие в реальном времени (авиа, космические, и др.) требуют высокой надежности (недопустимость ошибок, отказов, аварий и др.), которая в значительной степени зависит от оставшихся и не устраненных ошибках в процессе разработки ПТС.

На надежность ПО также влияют угрозы, приводящие к неблагоприятным последствиям, риску нарушения безопасности и способности компонентов системы сохранять устойчивость в процессе ее эксплуатации. При этом риск уменьшает надежность и безопасность системы, если проявляются внешние угрозы [4-7, 20-27].

В связи со сказанным особую важность приобретает задача применения ЖЦ при создании качественных компонентов, включая процессы:

- системный анализ и разработка требований к системе и характеристик качества функционирования;
- проектирование архитектуры (модели) системы и отдельных компонентов;
- реализация компонентов и их конфигурация в систему;
- тестирование компонентов и системы из компонентов;
- испытание системы;
- сопровождение системы.

На процессе анализа и разработки требований определяются функции системы и спецификация основных требований к системе с заданием метрик для оценки надежности [4, 19], в терминах интенсивности отказов или вероятности безотказного функционирования. Разработчики системы формируют:

- приоритеты функций системы по критерию важности их реализации;
- параметры среды функционирования и интенсивности выполнения функций и их отказов;
- входные и выходные данные для каждого функционального компонента системы;
- категории отказов и их интенсивность при выполнении функций в заданное время.

На процессе проектирования определяются:

- размеры информационной и алгоритмической сложности всех типов проектируемых компонентов;
- виды дефектов, свойственные всем типам компонентов системы;
- стратегии функционального тестирования компонентов по принципу «черного ящика» с помощью тестов для выявления дефектов и интеграционного тестирования.

Для достижения надежного продукта проводится анализ:

- вариантов архитектуры системы на соответствие требованиям к надежности;
- анализ рисков, отказов и деревьев ошибок для критических компонентов с целью обеспечения отказоустойчивости и восстанавливаемости системы;
- прогнозирование показателей размера системы, чувствительности к ошибкам, степени тестируемости, оценки риска и сложности системы;
- прогнозирование количества и плотности дефектов для проведения процесса измерения надежности.

На процессах реализации и тестирования системы проектные спецификации функций компонентов системы переводятся в коды и подготавливаются наборы тестов для автономного и комплексного тестирования компонентов и систему. При проведении автономного тестирования обеспечение надежности состоит в предупреждении появления дефектов в компонентах и создании эффективных методов защиты от них. Все последующие процессы разработки (например, верификация) не могут обеспечить надежность систем, а лишь способствуют повышению уровня надежности за счет обнаружения ошибок с помощью тестов различных категорий и их исправления.

На процессе испытаний проводится системное тестирование для установления соответствия внешних спецификаций функций целям системы. Испытание проводится в реальной среде функционирования или на испытательном стенде с помощью наборов данных для имитации функций компонентов. При подготовке к испытаниям изучается «история» тестирования (таблица сведений об ошибках и отказах) на процессах ЖЦ, использование ранее разработанных тестов для составления специальных тестов испытаний с целью [2]:

- управления ростом надежности при неоднократном исправлении и регрессионном тестировании ПТС;
- принятия решения о степени готовности системы и возможности ее передачи в эксплуатацию;
- оценки надежности по результатам системного тестирования и испытаний по соответствующим *моделям надежности*, подходящих для заданных целей системы.

На процессе сопровождения оценка надежности ПС проводится путем:

- протоколирования отказов в ходе работы системы, измерения надежности функционирования и использования результатов измерений для определения потерь (снижения) надежности в период времени эксплуатации;
- анализа частоты и серьезности отказов для определения порядка их устранения;
- оценки влияния функционирования системы на надежность в условиях усовершенствования технологии и применения новых инструментов разработки и тестирования.

На этапах жизненного цикла измерения надежности выполняются и решаются следующие задачи [5, 6]: определение функции распределения надежности по компонентам, прогнозирование плотности дефектов, прогнозирование надежности и оценки надежности. Рассмотрим эти задачи.

Задача распределения надежности по компонентам на процессах ЖЦ состоит в парном сравнения компонентов системы и построении квадратной матрицы $A(n \times n)$ из элементов вида:

$$a_{11} = a_{22} = \cdots = a_{nn} = 1, a_{ij} = \frac{1}{a_{ji}}, i, j = 1, \ldots, n, i \neq j;\ n = k, l, m,$$

где n – количество сравниваемых компонентов, k, l, m – количество функций и модулей соответственно. Матрица включает относительный вес w_i-го компонента, который вычислялся по формулам:

$$w_i = \frac{\sum_{j=1}^{n} a_{ji}}{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}}, \sum_{i=1}^{n} w_i = 1$$

В случае больших размеров матрицы в целях получаются более точные оценки компонентов иерархии, вычисляется собственный вектор и собственные значения матрицы. В них используются следующие данные: λ_{max}– максимальное собственное значение матрицы A n–порядка, w_i – коэффициент относительного веса элементов матрицы A, $W = (w_1, w_2, \ldots, w_n)$ – собственный вектор, которому соответствует λ_{max}. Общность решения задачи сравнения устанавливается соотношением $\alpha = \sum_{i=1}^{n} w_i$ и значением $\sum_{i=1}^{n} w_i = 1$. Если матрица A имеет $n-1$ собственных значений λ, равных нулю и $\lambda_{max} = n$, то она является согласованной. При этом индекс согласованности CI и коэффициент согласованности CR вычисляется по формулам:

$$CI = \frac{\lambda_{max} - n}{n-1}, CR = \frac{CI}{E(CI)},$$

где $E(CI)$ – математическое ожидание матрицы парных сравнений $A(n \times n)$.

Критерий приемлемости парного сравнения элементов в матрицах размером $n \geq 3$ получен такой: $CR \leq 0{,}05$ и $CR < 0{,}1$ для $n > 5$. По результатам сравнения формируется квадратная матрица $F(k \times k)$. Аналогично проводится сравнение компонентов ПТС. В результате сравнения получается k матриц. Возможный порядок каждой матрицы – i, а максимальный каждой из них – m.

Инструментом для сравнения является ExpertChoice входной матрицы A, которая автоматически вычисляет собственный вектор W, собственное значение λ_{max} и коэффициент согласованности CR. Для вычисления λ_{max} и W используются соответствующие функции пакета MATLAB 6.5.

Результаты сравнений заносятся в форму, содержащую перечень весовых коэффициентов программ, критерии, индексы и коэффициенты согласованности. Они предоставляются в виде готовых результатов обработки матриц. Полученные весовые коэффициенты синтезируются с помощью пакета MATLAB 6.5. Результаты отображаются в виде отчета о распределении надежности по объектам системы.

Прогнозирование плотности дефектов на процессе ЖЦ проводится по модели RLM (Rome Laboratory Model) с помощью следующих действий.

1) Анализ значений параметров модели прогнозирования, включая остаток дефектов от предыдущего этапа работ с ПО системы, и используется для целевого распределения значения надежности.
2) Сравнение прогнозируемого значения надежности с распределенным значением.
3) Корректировки переменных параметров и учет текущего состояния системы.
4) Оценка параметров модели прогнозирования надежности.
5) Прогнозирование плотности дефектов.
6) Определение значений (допусков) для оценок результатов прогнозирования и анализа альтернатив.
7) Расчет прогнозного значения надежности системы.

Расчет плотности дефектов делается с помощью модели RLM и путем прогнозирования плотности дефектов по формуле:

$$D_0 = \prod_{i=1}^{9} K_i,$$

где K_i – модификатор плотности дефектов D_0, с учетом пороговых значений данных о плотности дефектов.

Для каждой анализируемой системы результаты сравниваются с полученными по модели RLM. При проверке оказалось, что для ПО объемом 10-25 KSLOC погрешность прогнозирования плотности дефектов примерно составила 30-35%. Это объясняется некоторыми ограничениями системы Hugin Lite 6.5. Эти результаты используются при прогнозировании надежности компонентов.

Прогнозирование надежности компонентов реализуется по следующей формуле модели надежности:

$$R_i = exp\left[-D_i I_i \cdot \left(1 - exp\left(\frac{\rho_i K}{I_i \varphi_i} \cdot t\right)\right)\right],$$

где ρ_i – параметр среды эксплуатации i-го компонента, φ_i – характеристика среды разработки, I_i – оцененный размер начального кода, а D_i – прогнозируемая плотность дефектов в системе. Коэффициент дефектов K – константа, полученная для всех объектов ПТС, а значения ρ_i и φ_i – взяты при первоначальном прогнозировании надежности, которые не изменяются во время разработки компонентов системы.

Измерение надежности системы выполняется согласно классификации дефектов (Orthogonal Defect Classification), в соответствии с которой каждый выявленный дефект использует параметры: тип дефекта, триггер дефекта, влияние дефекта. Эти параметры используются одной или двумя подходящими моделями надежности, из выше приведенных, в целях проведения оценки прогнозного значения надежности отдельных компонентов и системы в целом. В модели оценивания надежности (стандарт ICO/IEC 9126) заданы *атрибуты*, которые определяют способность системы преобразовывать исходные данные в результаты при условиях, зависящих от периода жизни системы (износ и старение не учитываются). Снижение надежности компонентов может происходить из-за ошибок проектирования.

К атрибутам надежности относятся следующие:

Безотказность – свойство системы функционировать без отказов (программ или оборудования). Если компонент содержит дефект, то во множестве $D = \{De | e \in L\}$ всех дефектов, можно выделить подмножество $E \subseteq D$, для которых результаты не соответствуют функции F^m, заданной в требованиях на разработку. Вероятность p безотказного выполнения компонента на De, случайно выбранном из D среди равновероятных, равна $1 - card\{E\}/card\{D\}$.

Отказ (failure) показывает отклонение поведения системы от предписанного выполнения предписанных ей функций. Появление отказа может быть причиной ошибки (fault/error), вызывающей его. Если ошибка сделана человеком, то используется термин mistake. Когда различие между fault и failure не критично используется термин defect, означающий либо fault (причина), либо failure (действие). Так как существует большое разнообразие видов отказов (внезапные, постепенные, сбои и др.), то определяется наработка на отказ, среднее время между появлением угроз и делается оценка ущерба, которая наносится соответствующими угрозами. Вычисление среднего времени T наработки на отказ согласно стандарта реализуется по формуле:

$$T = \sum_{i=1}^{De} \nabla t_i^E \Big/ N,$$

где ∇t_i^E – интервал времени безотказной работы компонента i –го отказа; N – количество отказов в системе.

Устойчивость к ошибкам и отказам, которая показывает на способность системы выполнять функции при аномальных условиях (сбоях аппаратуры, ошибках в данных и интерфейсах, нарушениях в действиях исполнителей и др.) можно вычислить по формуле: $\Upsilon = N^v/N$, где N^v – количество разных типов отказов, для которых предусмотрены средства восстановления; N – общее количество всех отказов в системе.

Восстанавливаемость показывает способность возобновить функционирование системы после отказов для повторного исполнения и можно определить по формуле

$$T = \sum_{i=1}^{De} \nabla t_i^b \Big/ D,$$

где ∇t_i^b – время восстановления работоспособности компонента после i – отказа; D – количество дефектов и отказов в системе.

Количественная оценка надежности складывается согласно стандарту OSI/IEC 9126 из четырех атрибутов надежности системы по формуле:

$$Qv(reali) = \sum_{j=1}^{4} a_j\, m_j w_j,$$

где a_j – атрибуты надежности, m_j – метрики (внешние, внутренние), w_j – весовые коэффициенты.

Если оценка надежности получена очень малая, то требуется устранить обнаруженные ошибки и тогда повторно измеряется надежность до возрастания надежность системы до требуемого уровня. Чем интенсивнее проводится эксплуатация, тем интенсивнее выявляются ошибки, и тем самым обеспечивается рост надежности.

Важным фактором, влияющим на оценку надежности ПО, являются – угрозы, приводящие к снижению безопасности системы и ущербности всей системы. В связи с возникающими киберугрозами в сети Интернет, в практике работы с системами появляются более изощренные методы борьбы с такими угрозами для обеспечения функциональной безопасности действующих разного рода прикладных систем [17, 19].

5. Перспективные задачи обеспечения безопасности и надежности

Согласно ФЗ «О безопасности» (Федеральный закон от 28.12.2010 N 390-ФЗ (ред. от 05.10.2015)), ГОСТ Р 22.0.02, ГОСТ Р 51898-2002, ГОСТ 1.1-2002 и др. требуется обеспечение: функциональных и программно-технических интерфейсов при проектировании разных видов систем; безопасности и защищенности общества и государства от внутренних и внешних угроз; работоспособности и конкурентоспособности компьютерных систем, требующих прогнозирования рисков в сравнении с допустимыми рисками и принятие сбалансированных мер, противодействующих киберугрозам.

Основными направлениями развития являются:

- умные сети и города / технологии;
- интеллектуальные и онтологические системы;
- передовое производство компьютерных систем;
- робототехника и автономные системы - облачные вычисления;
- открытые данные и большие данные (Big Data);
- электронное управление;
- обмен метаданными по интероперабельным активам многократного использования и др.

Одним из важных направлений решения поставленных задач является безопасность и надежность систем, создаваемых на основе современных парадигм программирования.

6. Заключение

Отдельные аспекты теории надежности компонентов систем начали исследоваться в проекте РФФИ № 16-01-00352-18. В нем реализованы методы создания отдельных компонентов систем с учетом особенностей моделей вариабельности; проведена экспериментальная конфигурационная сборка варианта ОС Linux и отработан подход к созданию систем и сайтов из готовых ресурсов и сервисов Интернета [1, 2, 20-27]. Готовые ресурсы описывались на языках программирования (C, C++, Basic, Java, Python и др.), а их интерфейсы на языке WSDL. Они трансформировались к выходному коду и могли обрабатывать передаваемые данные (простые, структурные и неструктурированные типы данных) в распределенной среде Интернет. Использовался формальный аппарат отображения (mapping) неструктурных типов GDT стандарта ISO/IEC 11404–2007 к фундаментальным типам данных [24] для сред VS.Net, IBMWebSphere, Linux и др. Обработка неструктурированных данных Web-приложений проводилась разными методами, в том числе Web Content Mining и ЯП среды W3C. К компонентам применялись средства конфигурации, верификации, тестирования и сбора данных для проведения оценки надежности программ, работающих с большими данными.

Исследования теоретических и практических основ обеспечения качества и безопасности технических и программных систем выполняются в новом проекте РФФИ №19-01-00206/19 «Модели, методы и средства надежности технических и программных систем», как дальнейшее развитие проекта 00352. В данной статье проведено исследование моделей надежности, качества и обеспечения безопасности компьютерных систем с учетом известных в России и за рубежом работ [4-7, 10-19]. Согласно стандарту ЖЦ разработка системы проводится путем проектирования компонентов, а их также верификации и тестирования со сбором данных о наличии ошибок и отказов. Дана характеристика процессов ЖЦ, на которых выполняется для компонентов сбор данных об ошибках, отказах и дефектах. Приведена таблица эксперимента по измерению надежности с использованием оценочных моделей (Мусы, Окомота, Шнайдервинда и др.) для малых, средних и больших программ с учетом данных собранных на процессах ЖЦ по распределению и прогнозированию надежности компонентов. Приведена стандартная модель оценки надежности и дана характеристика модели измерения надежности систем с учетом прогнозирования надежности и плотности дефектов с атрибутами и метриками стандарта изменения надежности систем. Определены перспективы развития безопасных и надежных систем.

Список литературы / References

[1]. Лаврищева Е.М., Пакулин Н.В. Модели и методы надежности технических и программных систем. Материалы Пятой научно-практической конференции OS DAY, 2018 / Lavrishcheva E.M., Pakulin N.V. Models and methods of reliability of technical and software systems. In Proc. of the Fifth Scientific and Practical Conference OS DAY, 2018. Available at: http://0x1.tv/20180517F (in Russian), accessed 26.10.2019.

[2]. Лаврищева Е.М., Пакулин Н.В., Рыжов А.Г., Зеленов С.В. Анализ методов оценки надежности оборудования и систем. Практика применения методов. Труды ИСП РАН, том 30, вып. 3, 2018 г., стр. 99-120 / Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue. 3, 2018, pp. 99-120 (in Russian). DOI: 10.15514/ISPRAS-2018-30(3)-8

[3]. Андон Ф.И., Коваль Г.И. и др. Основы инженерии качества программных систем. К.: Наукова думка, 2007, 670 стр. / Andon F. I. et al. Foundation of quality engineering software system. K.: Naukova Dumka, 2007, 670 p. (in Russian).

[4]. Липаев В.В. Надежность и функциональная безопасность комплексов программ реального времени. Москва, ЗАО «Светлица», 2013, 193 стр. / Lipaev V.V. Reliability and functional safety of software systems real time. Moscow, Svetlitsa, 2013, 193 p. (in Russian).
[5]. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Основы индустрии программных продуктов. К.: Наукова думка, 2009, 372 стр. / Lavrisheva E. M., Grishchenko V. N. Assembly programming. Foundation of software industries. K.: Naukova Dumka, 2009, 372 p. (in Russian).
[6]. Р.Т. Фатрелл, Д.Ф. Шафер, Л.И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. Москва, Санкт-Петербург, Киев, Вильямс, 2003, 1125 стр. / Robert T. Futrell, Donald F. Shafer, Linda Isabell Shafer. Quality Software Project Management. Prentice Hall, 2002, 1680 p.
[7]. Липаев В.В. Методы обеспечения качества крупномасштабных программных систем. М.: СИНТЕГ, 2003, 510 стр. / Lipaev V. V. Methods of quality assurance of large-scale software systems. M.: SINTEG, 2003, 510 p. (in Russian).
[8]. Буренин П.В., Девянин П.Н. Лебеденко Е.В., Проскурин В.Г., Цибцуля А.Н. Безопасность операционной системы общего назначения Astra Linux Special Edition. Москва, Горячая линия, 2018, 311 стр. / Burenin P.V., Devyanin P.N., Lebedenko E.V., Proskurin V.G., Tsibtsulya A.N. Security of the Astra Linux Special Edition general purpose operating system. Moscow, Goryatchaya Linaya, 2018, 311 p. (in Russian).
[9]. Зеленов С.В., Зеленова С.А. Моделирование программно-аппаратных систем и анализ их безопасности. Труды ИСПРАН, том 29, вып. 5, 2017 г., стр. 257-282 / Zelenova S.A., Zelenov S.V. Modeling and Risk Analysis of Hardware-Software Systems. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017. pp. 257-282 (in Russian). DOI: 10.15514/ISPRAS-2017-29(5)-13.
[10]. Musa J.D. Okumoto K.A. Logarithmic Poisson Time Model for Software Reliability Measurement. In Proc. of the 7th International Conference on Software Engineering, 1984, стр. 230–238.
[11]. Shanthikumar J.G. Software realibility models: A Review. Microelectronics Reliability, vol. 23, issue 5, 1983, pp. 903-943
[12]. Yamada S., Ohba M., Osaki S. S–shaped software reliability grows modeling for software error detection. IEEE Transactions on Reliability, vol. R–32, № 5, 1983, стр. 475–478.
[13]. Chulani S. Constructive quality modeling for defect density prediction: COQUALMO. In Proc. of the International Symposium on Software Reliability Engineering, 1999, pp. 3-5.
[14]. Duval P., Matyas R., Grover A. Continuous integration improving Software quality and reducing risk. Addison Wesley, 2009, 691 p.
[15]. Горбенко А.В., Засуха С.А., Рубан В.И., Тарасюк О.М., Харченко В.С. Безопасность ракетно-космической техники и надежность компьютерных систем: 2000-е годы. Авиационно-космическая техника и технология, №1 (78), 2011, стр. 9-20 / Gorbenko A.V., Zasukha S.A., Ruban V.I., Tarasyuk O.M., Kharchenko V.S. The safety of rocket and space technology and the reliability of computer systems: the 2000s. Aerospace Engineering and Technology, №1 (78), 2011, pp. 9-20 (in Russian).
[16]. A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, 2004, pp. 11- 33.
[17]. Костогрызов А.И., Липаев В.В. Сертификация качества функционирования автоматизированных информационных систем. Москва, Изд-во "Вооружение. Политика. Конверсия", 1996, 275 стр. / Kostogryzov A.I., Lipaev V.V. Certification of the functioning quality of automated information systems. Moscow, Publishing House "Arms. Politics. Conversion", 1996, 275 p.
[18]. И.С. Захаров, М.У. Мандрыкин, В.С. Мутилин, Е.М. Новиков, А.К. Петренко, А.В. Хорошилов. Конфигурируемая система статической верификации модулей ядра операционных систем. Программирование, том 41, №1, 2015, стр. 44-67 / I.S. Zakharov, M.U. Mandrykin, M.U. Mandrykin, V.S. Mutilin, E.M. Novikov, A.K. Petrenko, A.V. Khoroshilov. Programming and Computer Software, vol. 41, №1, 2015, pp. 49–64
[19]. Абросимов Н.В., Костогрызов А.И. и др. Безопасность России. Правовые, социально-экономические и научно-технические аспекты. Техногенная, технологическая и техносферная безопасность. М.: МГОФ «Знание», 2018, 1016 стр. / Abrosimov N.V., Kostogryzov A.I. et al. Security of Russia. Legal, socio-economic and scientific-technical aspects. Technogenic, technological and technosphere safety. M, Znanie, 2018, 1016 p. (in Russian).

[20]. Кулямин В.В., Лаврищева Е.М., Мутилин В.С., Петренко А.К. Верификация и анализ вариабельных операционных систем. Труды ИСП РАН, том 28, вып. 3, 2017, стр. 189-208 / Kuliamin V.V., Lavrischeva E.M., Mutilin V.S., Petrenko A.K. Verification and analysis of variable operating systems. Trudy ISP RAN/Proc. ISP RAS, vol. 28, issue 3, 2016, pp. 189-208 (in Russian). DOI: 10.15514/ISPRAS-2016-1(2)-12.
[21]. Lavrischeva E.M., Mutilin V.S., Ryzhov A.G. Designing variability models for software, operating systems and their families. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017, pp. 93-110. DOI: 10.15514/ISPRAS-2017-29(5)-6.
[22]. Kozin S.V., Mutilin V.S. Static Verification of Linux Kernel Configurations. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 4, 2017, pp. 217-230. DOI: 10.15514/ISPRAS-2017-29(4)-14.
[23]. Козин С.В. Конфигурационная сборка варианта ядра Linux для прикладных систем. Труды ИСПРАН, том 30, вып. 6, 2018 г., стр. 161-170 / Kozin S.V. Linux kernel configuration build for application systems. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 6, 2018, pp. 161-170 (in Russian). DOI: 10.15514/ISPRAS-2018-30(6)-9.
[24]. Лаврищева Е.М., Рыжов А.Г. Подход к созданию систем и сайтов из готовых ресурсов. Труды XX Всероссийской научной конференции «Научный сервис в сети Интернет», 2018, стр. 321-346 / Lavrishcheva E.M., Ryzhov A.G. Approach to creating systems and websites from ready-made resources.: In Proc. of the XX All-Russian Scientific Conference on Scientific service on the Internet, 2018, p. 321-346 (in Russian).
[25]. Лаврищева Е.М. Компонентная теория и коллекция технологий для разработки индустриальных приложений из готовых ресурсов. Труды 4-й научно–практической конференции «Актуальные проблемы системной и программной инженерии», 2015 г., стр. 101–119 / Lavrischeva E.M. Component theory and collection technology for development of industry application from ready resources. In Proc. of the 4th scientific and practical conference on Actual Problems of System and Software Engineering, 2015, pp. 101-119 (in Russian).
[26]. E.M. Lavrischeva. The Scientific basis of software engineering. International Journal of Applied and Natural Sciences, vol. 7, 2018, pp. 15-32.
[27]. Lavrischeva E.M. Science of the computer programs and systems in XX-XXI centuries: past, present, future. European Journal of Mathematics and Computer Science, vol. 5, no. 1, 2018, стр. 67-92.

Информация об авторах / Information about authors

Екатерина Михайловна ЛАВРИЩЕВА, доктор физико-математических наук, профессор, главный научный сотрудник ИСП РАН, профессор МФТИ. Научные интересы: технология программирования, программная инженерия.

Ekaterina Mikhailovna LAVRISCHEVA, Doctor of Physics and Mathematics, Professor, Chief Researcher at ISP RAS, Professor at MIPT. Research interests: programming technology, software engineering.

Сергей Вадимович ЗЕЛЕНОВ, кандидат физико-математических наук, старший научный сотрудник ИСП РАН, доцент ВШЭ. Научные интересы: модели программно-аппаратных систем, надежность программного обеспечения.

Sergey Vadimovich ZELENOV, Candidate of Physics and Mathematics, Senior Researcher at ISP RAS, Associate Professor at HSE. Research interests: models of software and hardware systems, software reliability.

Николай Витальевич ПАКУЛИН, кандидат физико-математических наук, технический директор компании Pax Datatech PTE, Ltd. Научные интересы: сетевые технологии, блокчейн.

Nikolai Vitalievich PAKULIN, Candidate of Physics and Mathematics, Technical Director of Pax Datatech PTE, Ltd. Research: network technologies, blockchain.