

УДК 004.519.217

Д. А. МАЕВСКИЙ, Е. Ю. МАЕВСКАЯ, О. П. ЖЕКОВ

*Одесский национальный политехнический университет, Украина***ВЕРИФИКАЦИЯ МОДЕЛЕЙ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

В работе описана верификация существующих моделей надежности программного обеспечения и их сравнение с новой моделью на основе теории динамики программных систем. Для верификации использован статистически значимый ряд наблюдений за процессом выявления дефектов в пятидесяти разнообразных программных системах. Проведено сравнение точности оценивания девяти наиболее широко применяемых моделей надежности с новой моделью, которая базируется на теории динамики программных систем. Показано, что модель динамики программных систем обеспечивает в 2,7 раза большую точность оценивания надежности, чем существующие модели надежности.

Ключевые слова: программное обеспечение, надежность, модель надежности, динамика программных систем, программные дефекты, потоки дефектов.

Введение

Проблема обеспечения и прогнозирования надежности программного обеспечения информационных систем является одной из наиболее актуальных проблем современной программной инженерии. Цена программной ошибки сегодня может измеряться уже не миллионами долларов, а миллионами человеческих жизней. Современные информационные технологии настолько проникли во все сферы нашей жизни, что сегодня основная часть человечества стала заложником нами же созданных программных систем. Этим системам сегодня доверено огромное количество функций по управлению транспортом, связью, энергетикой, экономикой, обороной и так далее. Сегодня, чтобы парализовать жизнь большого города, достаточно сбоя в системе, управляющей работой светофоров на его оживленных транспортных магистралях. Значительно более тяжелыми могут быть последствия компьютерных ошибок и потери работоспособности систем мобильной связи или энергетических систем. А отказ системы управления ядерным реактором на ближайшей АЭС может привести к последствиям, значимым для всего континента.

Поэтому создание надежных компьютерных систем и поддержание их надежности в процессе эксплуатации – не просто актуальная, а жизненно важная задача. Оценке и прогнозированию надежности компьютерных систем посвящено сравнительно молодое направление теории надежности – надежность программного обеспечения (ПО). Его задачей является разработка теоретических основ надежности ПО, моделей, методов и практических технологий для определения надежности программных средств.

Современное состояние проблемы

В настоящее время насчитывается около двадцати различных моделей надежности программного обеспечения (МНПО). Такое обилие моделей привело к необходимости их классификации. На сегодня создано несколько классификационных схем. Наиболее распространенная классификация предложена Мусой и Окумото [1]. В ней выделяют такие классификационные признаки:

1. **Время модели.** Определяет систему отсчета времени, применяемую в модели – текущее астрономическое (календарное время) время или процессорное время, затраченное на работу с данной ПС к моменту выявления дефекта.

2. **Категория модели.** Определяет количество дефектов, которое может быть выявлено при бесконечно большом времени исследования. По этому признаку модели делятся на конечные и бесконечные.

3. **Тип модели.** Определяет распределение вероятности наступления случайных событий – выявления дефекта. В моделях надежности используются два типа распределения – *распределение Пуассона и биномиальное распределение.*

4. **Класс модели.** Этот признак применяется только для категории конечных моделей и определяет вид функции, описывающей закон изменения интенсивности проявления дефектов.

5. **Семейство.** Этот признак применяется только для бесконечных моделей и имеет тот же смысл, что и класс для конечных.

Для оценки точности моделирования показателей надежности было проведено исследование наиболее распространенных моделей, принадлежащих различным ветвям рассмотренной классификации.

Кратко рассмотрим характеристики этих моделей.

1. Модель Джелинского-Моранды [2]. Время – астрономическое, категория – конечная; тип – биномиальная; класс – экспоненциальная. Предположения: интенсивность отказов пропорциональна текущему количеству дефектов в программе и остается постоянной в промежутке между любыми двумя соседними моментами выявления дефектов; выявление всех дефектов в программе равновероятно и независимо; все дефекты имеют одинаковую степень важности; время до выявления следующего дефекта программы распределено экспоненциально.

2. Модель Гела-Окумото [3]. Время – астрономическое, категория – конечная; тип – Пуассоновская; класс – экспоненциальная. Предположения: все дефекты ПС взаимно независимы; обнаруженные дефекты устраняются немедленно; процесс обнаружения дефектов представляет собой поток однородных событий и имеет распределение Пуассона.

3. Модель Шнайдевинда [4]. Время – астрономическое, категория – конечная; тип – Пуассоновская; класс – экспоненциальная. Основная отличительная идея этой модели заключается в том, что интенсивность отказов, определенная в более позднее время, считается более точной для предсказания дальнейшего развития процесса, нежели интенсивность, измеренная на ранних стадиях. Предположения: количество ошибок в данном интервале не зависит от количества ошибок в других интервалах; количество обнаруженных ошибок уменьшается от интервала к интервалу; интенсивность отказов пропорциональна числу имеющихся ошибок в данный момент времени.

4. Модель Мусы [5]. Время – процессорное, категория – конечная; тип – Пуассоновская; класс – экспоненциальная. Предположения: процесс выявления дефектов является Пуассоновским процессом; интенсивность выявления дефектов пропорциональна количеству еще не выявленных дефектов.

5. Модель распределения Вейбулла [6]. Время – процессорное, категория – конечная; тип – биномиальная; класс – экспоненциальная. Предположения: в начальный момент наблюдений в ПС находится конечное количество дефектов; время до выявления дефекта случайно с вероятностью, подчиняющейся распределению Вейбулла.

6. S-образная модель [7]. Время – астрономическое, категория – конечная; тип – Пуассоновская; класс – гамма-распределение. Предположения: процесс выявления дефектов является Пуассоновским процессом; при выявлении дефекта он немедленно устраняется без внесения новых.

7. Модель Дюзна [8]. Время – астрономическое, категория – бесконечная; тип – Пуассоновская; семейство – гамма-распределение. Предположения:

выявление кумулятивных дефектов представляет собой Пуассоновский процесс с функцией распределения $\mu(t) = \alpha \cdot t^\beta$, где α и β – положительные числа.

8. Геометрическая модель Моранды [9]. Время – астрономическое, категория – бесконечная; тип – Пуассоновская; семейство – экспоненциальная. Предположения: интенсивность отказов представляет собой геометрическую прогрессию $\lambda(t) = D \cdot \varphi^{i-1}$ со знаменателем $0 < \varphi < 1$; вероятность выявления каждого отдельного дефекта имеет экспоненциальный закон распределения.

9. Логарифмическая модель Мусы-Окумото [10]. Время – астрономическое, категория – бесконечная; тип – Пуассоновская; семейство – экспоненциальная. Предположения: интенсивность отказов уменьшается со временем по экспоненциальному закону; процесс выявления дефектов является Пуассоновским процессом.

10. Модель динамики программных систем. Основы теории динамики программных систем (ДПС) разработаны в [12] как принципиально новый детерминированный подход к определению показателей надежности с учетом влияния вторичных дефектов. ДПС отличается от существующей теории надежности ПО тем, что базируется на общей теории динамики систем, а не на теории вероятностей, и рассматривает процессы проявления дефектов в программной системе (ПС) не как случайный процесс, а как результат действия детерминированных потоков дефектов.

В основу ДПС положены следующие предположения:

Предположение 1. ПС является открытой неравновесной системой, которая взаимодействует со своим окружением по законам неравновесных процессов.

Предположение 2. ПС характеризуется специальной функцией состояния – количеством дефектов, которые находятся в ней.

Предположение 3. Исчезновение и появление дефектов в ПС является результатом совместимого действия прямого (выходного) и обратного (входного) потоков дефектов.

Предположение 4. Интенсивность потока пропорциональна количеству дефектов, образующих поток.

Предположение 5. (Принцип равноправия). Все дефекты равноправны и участвуют в образовании потока одинаковым образом, независимо от причины возникновения, локализации, типа дефекта и возможных последствий его проявления.

Предположение 6. (Принцип непрерывности). Функция является дифференцированной на всей

области определения.

Описание процесса верификации

В качестве исходных данных для верификации перечисленных моделей надежности использованы данные о процессе выявления дефектов в пятидесяти разнотипных программных системах. Все они написаны на разных языках программирования и имеют разное функциональное назначение. Сведения об этих системах приведены в таблице 1. Все использованные ряды являются неоднородными: в силу вносимых в системы изменений закон изменения кривых обнаружения дефектов меняется во вре-

мени. Поэтому, для повышения точности, каждый из рядов разбивался на участки и неизменным законом выявления дефектов.

Всего насчитывается 522 интервала с одинаковым законом изменения количества выявленных дефектов во времени. По всем этим интервалам для десяти исследуемых моделей выполнено 5220 оценок надежности и определена их точность. Точность определялась по критерию среднеквадратичного отклонения (СКО) наблюдаемых и вычисленных по модели значений дефектов и дисперсии значений СКО для разных программных систем.

Таблица 1

Сведения о рассматриваемых программных системах

№ ряда	Источник данных	Сведения о системе и языке программирования	Кол-во точек
1	[13]	ОС «Андроид», вер. 2.3	765
2	[14], глава 4, файл Csr1.dat	Нет данных	397
3	[14], глава 4, файл Csr2.dat	Нет данных	129
4	[14], глава 4, файл Csr3.dat	Нет данных	104
5	[14], глава 4, файл SS3.dat	Нет данных	278
6	[14], глава 4, файл Sys1.dat	Нет данных	136
7	[14], глава 7, файл Sys1.dat	Нет данных	136
8	[14], глава 7, файл Sys2.dat	Нет данных	86
9	[14], глава 7, файл Sys3.dat	Нет данных	207
10	[14], глава 7, файл J1.dat	Нет данных	62
11	[14], глава 7, файл J2.dat	Нет данных	181
12	[14], глава 7, файл J3.dat	Нет данных	41
13	[14], глава 7, файл J4.dat	Нет данных	114
14	[14], глава 7, файл J5.dat	Нет данных	73
15	[14], глава 8, файл 8.txt	Мультипроц. система	186
16	[14], глава 9, файл Odc1.dat	Крупный IBM проект	1207
17	[14], глава 9, файл Odc3.dat	Нет данных	400
18	[14], глава 10, файл S2.dat	Нет данных	54
19	[14], глава 10, файл S27.dat	Нет данных	41
20	[14], глава 10, файл SS4.dat	Нет данных	197
21	[14], глава 10, файл SS1.dat	Язык ассемблера	81
22	https://github.com/AArnott/dotnetopenid	C#	61
23	https://github.com/activescaffold/active_scaffold	Ruby	119
24	https://github.com/adamzap/landslide	Python	101
25	https://github.com/AFNetworking/AFNetworking	Objective-C	186
26	https://github.com/ai/r18n	Ruby	57
27	https://github.com/akzhan/jwysiwyg	JavaScript	270
28	https://github.com/alankligman/gladius	JavaScript	91
29	https://github.com/AlanQuatermain/AQGridView	Objective-C	107
30	https://github.com/alecgorge/jsonapi	Java	115
31	https://github.com/alohaeditor/Aloha-Editor	JavaScript	404
32	https://github.com/amatsuda/kaminari	Ruby	194
33	https://github.com/andreasgal/B2G	Rust	142
34	https://github.com/andreasronge/neo4j	Ruby	138
35	https://github.com/andrewplummer/Sugar	JavaScript	91
36	https://github.com/andris9/Nodemailer	JavaScript	62
36	https://github.com/andymatuschak/Sparkle	Objective-C	134
38	https://github.com/antirez/hiredis	C	78
39	https://github.com/apneadiving/Google-Maps-for-Rails	Ruby	143
40	https://github.com/Araq/Nimrod	Nimrod	97
41	https://github.com/arsduo/koala	Ruby	162
42	https://github.com/asual/jquery-address	JavaScript	112
43	https://github.com/away3d/away3d-core-fp11	ActionScript	205
44	https://github.com/BaseXdb/basex	Java	346
45	https://github.com/Baystation12/Baystation12	Нет данных	304
46	https://github.com/bbatsov/ruby-style-guide	Ruby	68
47	https://github.com/benbarnett/jquery-Animate-Enhanced	JavaScript	66
48	https://github.com/bengottlieb/Twitter-OAuth-iPhone	Objective-C	96
49	https://github.com/benoitc/unicorn	Python	290
50	https://github.com/binaryage/totalfinder-i18n	Ruby	175

Значения СКО вычислялись по формуле:

$$СКО = \frac{\sum_{k=1}^n (f_{ok} - f_{ck})^2}{n}, \quad (1)$$

где f_{ok} – наблюдаемое количество дефектов в k -й точке ряда, f_{ck} – вычисленное по модели количество дефектов в k -й точке ряда, n – количество точек в ряде.

Значения дисперсии вычислялись по формуле:

$$\sigma^2 = \sum_{i=1}^N (x_i - \mu) \cdot p(x_i), \quad (2)$$

где N – количество исследованных программных систем; x_i – среднее значение СКО для i -й системы, μ – математическое ожидание x_i , $p(x_i)$ – вероятность приобретения случайной величиной x значения x_i . Для расчета дисперсии использован пакет статистических функций Microsoft Excel.

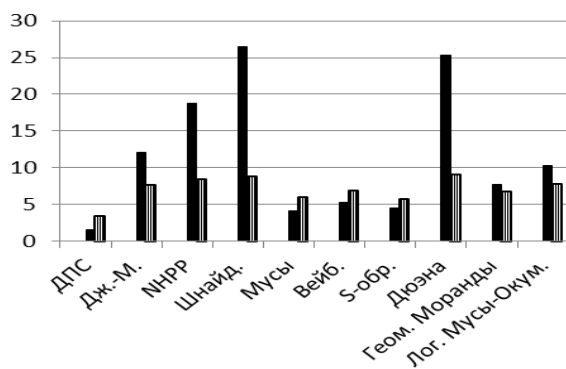


Рис. 1. Гистограмма значений СКО (сплошная линия) и логарифма дисперсии (штриховая линия)

Результаты приведены в таблице 2. Кроме этого, в таблице 2 приведено количество необработанных каждой моделью рядов (в %), минимальное и максимальное значения СКО, показанные моделью по всем рядам, среднее значение СКО по модели, а также логарифм дисперсии.

Из анализа таблицы 2 видно, что лишь две из исследованных девяти МНПО – модель ДПС и S-образная модель смогли выполнить оценку надеж-

ности для всех 522 временных интервалов. Наихудшей по этому показателю оказалась модель логарифмическая Мусы-Окумото, которая не смогла обработать почти 71% интервалов. Наилучшей по значению критерию СКО и дисперсии оказалась модель ДПС. Она показала в 2,7 раза более точные результаты, чем модель Мусы, которая заняла второе место по точности. По значению дисперсии модель ДПС показала на два и более порядка меньшие значения, чем остальные исследованные модели.

Результаты верификации в графическом виде представлены на рисунке 1.

а этом рисунке сплошной линией показаны значения СКО, а штриховой – логарифмы значений дисперсии. Логарифмическая шкала для дисперсии использована потому, что ее абсолютные значения от модели к модели разнятся на порядки.

Выводы

Таким образом, результаты верификации показывают, что модель ДПС на сегодняшний день является лучшей по точности оценивания надежности моделью. Низкие значения дисперсии говорят о том, что модель ДПС показывает одинаково точные результаты для всех исследованных программных систем, и поэтому может считаться универсальной моделью.

Литература

1. Musa, J. D. *Software Reliability Models: Concepts, Classification, Comparisons, and Practice [Text]* / J. D. Musa, K. Okumoto // *Electronic Systems Effectiveness and Life Cycle Costing, NATO ASI Series, F3, Springer-Verlag, Heidelberg.* – P. 395–4245.
2. Moranda, P. B. *Final Report of Software Reliability Study [Text]* / P. B. Moranda, J. Jelinski // *McDonnell Douglas Astronautics Company.* – MDC Report № 63921, dec. – 1972. – 51 p.
3. Goel, A. L. *Time-Dependent Error-Detection Rate Model for Software and Other Performance Measures [Text]* / A. L. Goel, K. Okumoto // *IEEE Transactions on Reliability.* – V. R-28, № 5, August. – 1979. – P. 206 – 211.

Таблица 2

Результаты верификации

Показатели	ДПС	Дж.-Мор.	NHPP	Шнайд.	Мусы	Вейб.	S-обр.	Дюэна	Геом. Моран.	Лог. Мусы-Окум
Необр. рядов, %	0,00	54,21	15,90	36,78	55,56	0,96	0,00	28,16	9,39	70,88
Мин. СКО	0,00	0,23	0,02	0,00	0,01	0,00	0,00	0,09	0,14	0,00
Макс. СКО	54,54	379,59	800,88	928,06	246,34	559,70	245,97	996,76	556,26	417,84
Ср. СКО	1,51	12,14	18,72	26,47	4,11	5,29	4,46	25,29	7,78	10,32
$\lg(\sigma^2)$	3,37	7,59	8,47	8,75	5,91	6,86	5,75	9,09	7,27	7,72

4. Schneidewind, N. F. *Software Reliability Model with Optimal Selection of Failure Data [Text]* / N. F. Schneidewind // *IEEE Transactions on Software Engineering*. – Vol. 19, No. 11. Nov. – 1993. – P. 1095–1104
5. Musa, J. D. *Validity of Execution time theory of software reliability [Text]* / J. D. Musa // *IEEE Trans. on reliability*. – 1979. – № 3. – P. 199–205.
6. Quadri, S. M. K. *Software Reliability Growth Modeling with New Modified Weibull Testing-effort and Optimal Release Policy [Text]* / S. M. K. Quadri, N. Ahmad // *IJCA*. – Vol. 6. – 2010. – № 12. – P. 1–10.
7. Yamada, S. *S-Shaped Reliability Growth Modeling for Software Error Detection [Text]* / S. Yamada, M. Ohba, S. Osaki // *IEEE Transactions on Reliability*. – Vol. R-32, No. 5, Dec. – 1983. – P. 475–478.
8. Duan, J. T. *Learning Curve Approach to Reliability Monitoring [Text]* / J. T. Duan // *IEEE Trans. on Aero-space*. – 1964. – Vol. 2. – P. 563–566.
9. Moranda, P. B. *Event-Altered Rate Models for General Reliability Analysis [Text]* / P. B. Moranda // *IEEE Transactions on Reliability*. – Vol. R-28, No. 5. – 1979. – P. 376–381.
10. Musa, J. D. *A Logarithmic Poisson Time Model for Software Reliability Measurement [Text]* / J. D. Musa, K. Okumoto // *Proc. Sevent International Conference on Software Engineering*. – Florida : Orlando. – 1984. – P. 230–238.
11. Маєвський, Д. А. *Анализ моделей надежности программного обеспечения гарантоспособных информационных систем [Текст]* / Д. А. Маєвський, С. А. Яремчук // *Електромашинобудування та електрообладнання*. – 2010. – № 76. – С. 68–79.
12. Maevsky, D. A. *A New Approach to Software Reliability [Text]* / Dmitry A. Maevsky // *Lecture Notes in Computer Science : Software Engineering for Resilient Systems*. – № 8166. – Berlin : Springer, 2013. – P. 156–168.
13. *Android – An Open Handset Alliance Project [Электронный ресурс]*. – Режим доступа: <http://code.google.com/p/android/issues/list>. – 10.03.2014.
14. Lyu, M. R. *Handbook of Software Reliability Engineering [Text]* / M. R. Lyu. – New York : McGraw-Hill Company, 1996. – 805 p.

Поступила в редакцію 10.03.2014, рассмотрена на редколлегии 24.03.2014

Рецензент: д-р техн. наук, проф., зав. каф. комп'ютерних систем і мереж В. С. Харченко, Національний аерокосмічний університет ім. Н. Е. Жуковського «ХАІ», Харків, Україна.

ВЕРИФІКАЦІЯ МОДЕЛЕЙ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Д. А. Маєвський, О. Ю. Маєвська, О. П. Жеков

У роботі описана верифікація існуючих моделей надійності програмного забезпечення та їх порівняння з новою моделлю, яку побудовано на основі теорії динаміки програмних систем. Для верифікації використаний статистично значимий ряд спостережень за процесом виявлення дефектів в п'ятдесяти різноманітних програмних системах. Проведено порівняння точності оцінювання дев'яти найширше вживаних моделей надійності з новою моделлю, яка базується на теорії динаміки програмних систем. Показано, що модель динаміки програмних систем забезпечує в 2,7 разів більш велику точність оцінювання, ніж існуючі моделі надійності.

Ключові слова: програмне забезпечення, надійність, модель надійності, динаміка програмних систем, програмні дефекти, потоки дефектів.

VERIFICATION OF THE SOFTWARE RELIABILITY MODELS

D. A. Maevsky, E. J. Maevskaya, O. P. Zhekov

The article describes the verification of existing software reliability models and their comparison with the new model based on the theory of the dynamics of software systems. For verification we used a statistically significant number of observations for the detection of defects in the process of fifty diverse software systems. A comparison of the estimation accuracy of the nine most widely used models of reliability with a new model that is based on the theory of the dynamics of software systems. It is shown that the model of the dynamics of software systems provides 2.7 times more accurate estimation of reliability than the existing reliability models.

Keywords: software reliability, reliability model, the dynamics of software systems, software defects, defect fluxes.

Маєвський Дмитрій Андреевич – д-р техн. наук, доцент, завідувач кафедри теоретических основ і общей електротехники Одесского національного політехнічного університета, г. Одеса, Україна, e-mail: Dmitry.A.Maevsky@gmail.com.

Маєвська Елена Юрьевна – канд. техн. наук, доцент, доцент кафедри теоретических основ і общей електротехники Одесского національного політехнічного університета, г. Одеса, Україна, e-mail: ifs@ukr.net.

Жеков Олег Прокопьевич – ассистент кафедри теоретических основ і общей електротехники Одесского національного політехнічного університета, г. Одеса, Україна, e-mail: vokeg@rambler.ru.