

НЕСТАЦИОНАРНЫЕ МОДЕЛИ СТРАТЕГИЙ ИСПЫТАНИЙ ПРОГРАММНЫХ СРЕДСТВ ПРИ ВЕРОЯТНОСТНЫХ ПАРАМЕТРАХ ОБНАРУЖЕНИЯ ОШИБОК

А. Д. Хомоненко^а, доктор техн. наук, профессор

А. И. Данилов^б, канд. техн. наук, доцент

А. А. Данилов^в, ведущий инженер-программист

^аПетербургский государственный университет путей сообщения Императора Александра I, Санкт-Петербург, РФ

^бВоенно-космическая академия им. А. Ф. Можайского, Санкт-Петербург, РФ

^вЗАО «Нокиа Солюшенз энд Нетвокс», Санкт-Петербург, РФ

Постановка проблемы: современные стандарты разработки программных средств требуют планировать сроки и ресурсы, необходимые для успешной реализации проектов. Важным аспектом проектов являются работы по обеспечению заданных показателей качества разрабатываемых программных средств. Моделирование процессов испытаний программных средств позволяет планировать ресурсы и финальное качество на ранних этапах реализации проектов. Известные модели испытаний программных средств не учитывают вероятностный характер нахождения и исправления ошибок. Модели строятся на результате двоичной классификации модулей программных средств на потенциально приводящие к отказу с вероятностью, равной единице, и надежные. **Цель исследования:** усовершенствование существующих моделей испытаний программных средств с учетом вероятностного характера процессов поиска и устранения ошибок. **Результаты:** разработаны три динамические модели процессов (стратегий) испытаний программных средств, в которых предусматривается возможность использования вероятностей обнаружения ошибок для каждого программного модуля. Для всех стратегий испытаний построены модифицированные размеченные графы и системы дифференциальных уравнений. Найдены численные решения систем дифференциальных уравнений, вероятностные показатели процессов испытаний: вероятности пребывания системы во всех состояниях, функции распределения времени обнаружения и устранения ошибок, математические ожидания случайных величин, количество обнаруженных и устраненных ошибок. Проведено сравнение стратегий по показателям качества. Разработанные модели позволяют использовать оценки надежности для каждого модуля в отдельности. Это повышает точность моделирования процессов отладки программ и позволяет учитывать состоятельность (мощность) тестов, искать пути повышения характеристик надежности программных средств путем формирования тестов, обнаруживающих ошибки с высокой вероятностью.

Ключевые слова — модель, программные средства, ошибка, вероятность, стратегия испытаний, граф.

Введение

Сокращение сроков и затрат на разработку программных средств (ПС) и обеспечение их качества (надежности) могут быть достигнуты не только за счет применения более совершенных технологий, сред разработки и средств автоматизации тестирования (отладки), но и путем грамотного и эффективного планирования проведения работ и разделения ресурсов при разработке (испытании) ПС.

Для решения поставленных задач могут быть использованы средства математического моделирования надежности ПС и процессов их разработки (испытаний). К настоящему времени создано большое количество различных моделей для оценивания и предсказания надежности ПС на различных этапах их жизненного цикла. Подробный обзор и описание таких моделей приведены в работах [1–4]. С методической точки зрения эти модели надежности подразделяют на три класса: эмпирические, статистические и вероятностные. В рамках каждого класса модели надежности могут отличаться применимостью на различных фазах и стадиях жизнен-

ного цикла ПС и определяемыми показателями надежности.

Эмпирические модели являются наиболее простыми. Они основаны на анализе накопленной информации о функционировании разработанных ПС (модель М. Холстеда, модель фирмы IBM). В то же время эмпирические модели достаточно грубы. Часто ошибки в расчетах различаются примерно на порядок с фактическими результатами. Кроме того, эти модели оперируют только с понятием корректности ПС, определяемой числом программных ошибок, но не отражают динамики вычислительного процесса при эксплуатации средств вычислительной техники.

В статистических моделях оценка числа ошибок в ПС связывается с числом ошибок в ней пропорциональными соотношениями, определяемыми простыми статистическими методами на основе интуитивных допущений. При этом не формулируется никаких предположений о законах распределения случайных величин, а источником информации является только исследуемое ПС (модели Миллса, Коркорэна, Бейзина, Б. Руднера и др.). Основная область их использования — завершающие стадии фазы разработки ПС.

Наиболее представительная группа моделей — вероятностные. В нее включены наиболее сложные модели, описывающие случайные процессы обнаружения и проявления программных дефектов или ошибок. Они основаны на описании случайных процессов появления ошибок и характеризуют процесс отладки. В состав параметров моделей входит исходное спрогнозированное число ошибок N , и поэтому модели описывают только свойство корректности. Обобщенная вероятностная модель основана на том, что в зависимости от времени отладки или испытаний скорость выявления ошибок уменьшается (модели Дж. Муса, Л. И. Волкова, Э. Нельсона, Джелинского — Моранды, Ямады — Охбы — Осаки, S -образные модели роста надежности и др.).

Следовательно, все эти модели в совокупности обладают рядом недостатков, ограничивающих их использование: модели не обладают достаточной общностью; описывают процессы обнаружения, но не устранения ошибок, что делает невозможным их применение собственно для планирования испытаний; модели, представленные в работах [2, 3], не учитывают вероятностный характер обнаружения ошибок и состоятельность (мощность) используемых тестов.

Целью настоящей работы является разработка динамических моделей процессов (стратегий) испытаний (тестирования и исправления ошибок) ПС, в которых предусматривается возможность учитывать вероятностные параметры обнаружения ошибок при тестировании программ или состоятельность используемых тестов. Названные параметры обнаружения ошибок и состоятельность используемых тестов могут быть определены, к примеру, с помощью подходов, рассмотренных в работе [5].

При задании исходных данных для динамических моделей процессов испытаний ПС распространенным подходом является использование собранной статистики уже завершенных проектов. Для учета особенностей конкретного проекта широко применяются различные программные метрики, характеризующие программный продукт (его структуру, сложность, интерфейсы, факторы разработки и др.). Так, с точки зрения современного объектно-ориентированного программирования целесообразно использовать наборы метрик Чайдамбера — Кемерера и Бансия — Дэвиса [6–8] (длина наибольшего пути иерархии наследования, взвешенное количество методов класса, количество классов-наследников и др.). Значения метрик собираются из документов проектирования — диаграмм универсального языка моделирования UML. На основе полученных метрик решается задача классификации модулей. Так, вероятность наличия ошибки в модуле ω_i можно оценить с помощью

логистической регрессии и выразить формулой [6, 7]

$$\omega_i = 1 / \left(1 + e^{-(\beta_0 + \sum \beta_j x_j)} \right), \quad (1)$$

где x_j — значение метрики сложности; β_j — коэффициент, который находится методом максимального правдоподобия.

По принятому в работах [2, 3] подходу модели строятся на результате двоичной классификации модулей ПС на классы: потенциально приводящих к отказу ($\omega_i = 1$) и надежных ($\omega_i = 0$). Новизна предлагаемых моделей заключается в отказе от двоичной классификации модулей и использовании в них оценки надежности для каждого модуля.

Модели испытаний программных средств

В работе [9] рассмотрена нестационарная модель операции по испытанию отдельного программного модуля, содержащего единственную ошибку, которую с определенными вероятностями можно найти и исправить или отсутствие которой можно подтвердить. На базе этой модели предложены четыре обобщенные модели процессов (стратегий) испытаний ПС, детальное описание и экспериментальное исследование из которых получила только одна (базовая) модель с одновременным поиском и исправлением ошибок (тестирование во время исправления ошибок не приостанавливается).

Рассмотрим три стратегии распределения времени и ресурсов между этапами тестирования и исправления найденных ошибок.

Распределения временных интервалов между моментами обнаружения ошибок во время тестирования описываются экспоненциальными законами. Предполагается, что ПС могут содержать не более N ошибок, которые обнаруживаются с интенсивностями $\{\omega_1 \lambda_1, \omega_2 \lambda_2, \dots, \omega_N \lambda_N\}$, зависящими от номера ошибки и значений соответствующих вероятностей их обнаружения. В случае подтверждения отсутствия ошибки осуществляется переход по соответствующим дугам с интенсивностями $\{(1 - \omega_1) \lambda_1, (1 - \omega_2) \lambda_2, \dots, (1 - \omega_N) \lambda_N\}$. На рис. 1–3 используются обозначения $(1 - \omega_i) = \omega_i$. Время устранения всех обнаруженных ошибок распределено по экспоненциальному закону с интенсивностью μ .

Стратегия 1. Этап тестирования продолжается, пока не будут определены все отсутствующие и обнаружены имеющиеся ошибки. Затем выявленные ошибки исправляются.

Представим процессы обнаружения и исправления ошибок марковским процессом с дискретным множеством состояний и непрерывным временем. Диаграмма (размеченный граф) состояний и переходов таких процессов представлена на рис. 1.

Состояния (i, j) такой системы в каждый момент времени будем характеризовать количест-

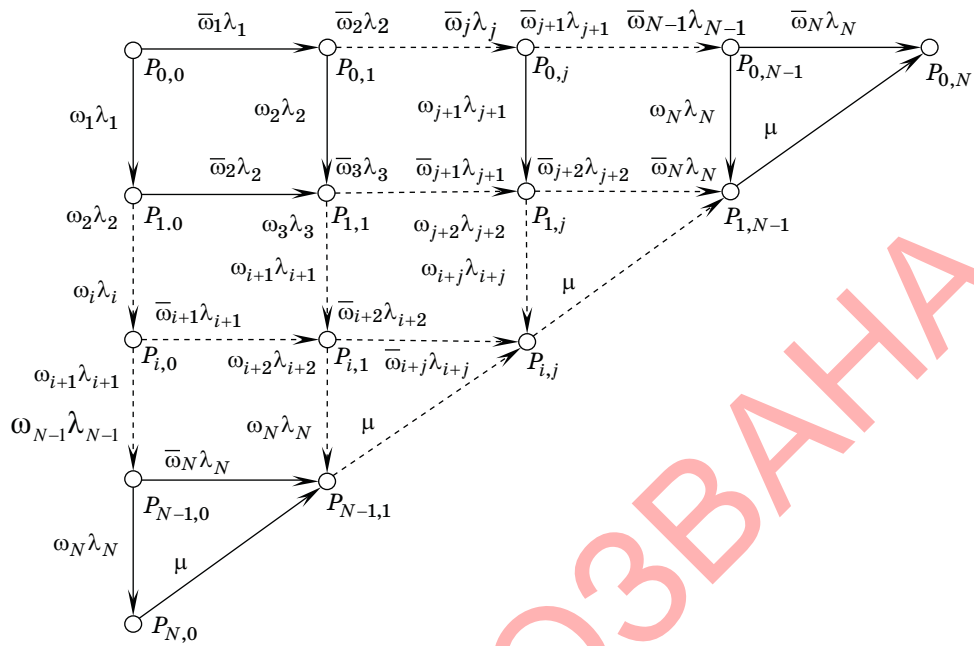


Рис. 1. Граф переходов для стратегии 1

вом обнаруженных, но еще не исправленных ошибок i и числом устраненных (сумма отсутствующих и исправленных ошибок) j ($i, j = 0, 1, \dots, N$). Вероятности пребывания системы в этих состояниях обозначим $P_{i,j}(t)$.

Переход системы из состояния (i, j) в состояние $(i + 1, j)$ означает, что при тестировании была обнаружена $(i + j + 1)$ -я ошибка. Переход из состояния (i, j) в состояние $(i, j + 1)$ означает, что при тестировании подтверждено отсутствие ошибки. Для последних состояний в каждом столбце графа переход системы из состояния (i, j) в состояние $(i - 1, j + 1)$ означает, что была исправлена $(j + 1)$ -я обнаруженная ошибка.

В таком представлении общее число состояний системы вычисляется по формуле

$$N_c = (N + 1)(N + 2)/2. \quad (2)$$

Процессы в рассматриваемой марковской цепи описываются системой из N_c дифференциальных уравнений

$$\begin{aligned} \frac{dP_{i,j}(t)}{dt} = & \delta(i)P_{i-1,j}(t)\omega \times; \\ & \times \lambda_{i+j} - \delta(N-i-j)P_{i,j}(t)\lambda_{i+j+1} + \\ & + \delta(j)P_{i,j-1}(t)(1 - \omega_{i+j})\lambda_{i+j} + \\ & + \delta((i+j+1-N)j)P_{i+1,j-1}(t)\mu - \\ & - \delta((i+j+1-N)i)P_{i,j}(t)\mu, \end{aligned} \quad (3)$$

где $\delta(m) = \begin{cases} 1, & \text{если } m > 0 \\ 0, & \text{если } m \leq 0 \end{cases}$ — функция Хэвисайда;

$i = 0, 1, \dots, N; j = 0, 1, \dots, N - i$.

Для каждого момента времени t должно соблюдаться условие нормировки вида $\sum_{i=0}^N \sum_{j=0}^{N-i} P_{i,j}(t) = 1$. Задав начальные условия к системе уравнений в виде

$$P_{i,j}(0) = \begin{cases} 1, & \text{если } i + j = 0; \\ 0, & \text{если } i + j \neq 0, \end{cases}$$

можно найти численное решение соответствующей задачи Коши для произвольного значения времени t .

Используя решение предложенной системы (3), можно получить ряд важных вероятностных показателей процесса испытаний и состояния ПС (так же, как и для базовой модели).

Вероятность того, что в процессе испытаний было найдено ровно i ошибок, представляет собой сумму найденных, но не исправленных и устраненных ошибок:

$$R_i(t) = \sum_{j=0}^i P_{i-j,j}(t), \quad i = \overline{0, N}. \quad (4)$$

Тогда математическое ожидание числа найденных ошибок вычисляется по формуле

$$N_R(t) = \sum_{i=1}^N iR_i(t). \quad (5)$$

Вероятность того, что в процессе испытаний было устранено (исправлено или отсутствовало) ровно j ошибок, может быть вычислена по формуле

$$P_j(t) = \sum_{i=0}^{N-j} P_{i,j}(t), \quad j = \overline{0, N}. \quad (6)$$

Математическое ожидание числа устраненных ошибок

$$N_P(t) = \sum_{j=1}^N jP_j(t). \quad (7)$$

Соответственно, среднее число $N_{RL}(t)$ ошибок, не обнаруженных к моменту времени t , и среднее число $N_{PL}(t)$ ошибок, не устраненных к моменту времени t , определяются по формулам

$$N_{RL} = N - N_R(t); \quad (8)$$

$$N_{PL} = N - N_P(t). \quad (9)$$

Вероятность безотказной работы в течение интервала τ после испытаний в течение времени t

$$P(t, \tau) = 1 - \sum_{j=0}^{N-1} R_j(t) (1 - e^{-\lambda_{i+1}\tau}). \quad (10)$$

Функция распределения времени устранения не менее i ошибок

$$F_i(t) = \sum_{l=i}^N P_l(t), \quad i = \overline{0, N}. \quad (11)$$

Модель позволяет оценить время отладки, которое требуется, чтобы устранить $N_{тр}$ ошибок с заданной вероятностью $P_{тр}$, как

$$T_{N_{тр}, P_{тр}} = t | F_{N_{тр}}(t) \geq P_{тр}. \quad (12)$$

Стратегия 2. Этап тестирования продолжается, пока не будет обнаружена ошибка или не определено отсутствие очередной ошибки. После обнаружения ошибки тестирование приостанавливается, ошибка исправляется. После исправления ошибки или ее необнаружения вновь начинается тестирование. Диаграмма (размеченный граф) переходов, представляющая данную стратегию отладки, представлена на рис. 2. Количество состояний такой системы $N_c = 2N + 1$.

Система дифференциальных уравнений (2) примет соответствующий вид:

$$\begin{aligned} \frac{dP_{i,j}(t)}{dt} = & -\delta(N-j-Ni)P_{i,j}(t)\lambda_{j+1} + \\ & + \delta(i)[P_{i-1,j}(t)\omega_{j+1}\lambda_{j+1} - P_{i,j}(t)\mu_{j+1}] + \\ & + \delta(j(1-i))[P_{i+1,j-1}(t)\mu_j + P_{i,j-1}(t)(1-\omega_j)\lambda_j], \quad (13) \end{aligned}$$

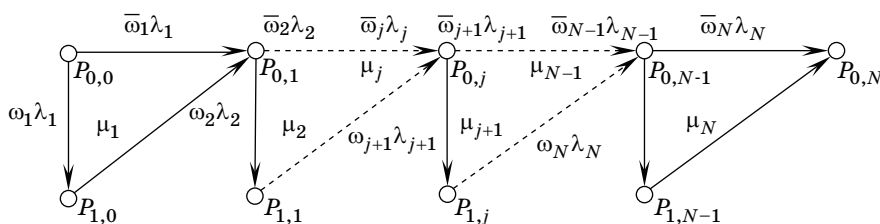
где $i = \overline{0, 1}; j = \overline{0, 1, \dots, N-i}$.

Отметим, что при этой стратегии, кроме однородного показательного распределения интервалов времени исправления найденных ошибок (μ постоянно), допускается процесс, при котором μ зависит от номера обнаруженной ошибки, что и отражено на рис. 2 и в системе дифференциальных уравнений (13).

Характерной особенностью процессов отладки ПС является наличие в них ошибок, исправление которых является условием для отыскания и устранения последующих ошибок. Для моделирования таких процессов испытаний необходимо рассмотреть стратегию, отличную от стратегии 1 наличием двух и более пакетов ошибок (стратегия 3). Не нарушая общности рассуждений, ограничимся двумя пакетами ошибок. Обоснуем это допущение ссылками на источники. Так, в работах [1, 10] все дефекты в программах делятся на явные и неявные, а все ошибки делятся на первичные и вторичные. «При отладке и тестировании программ обычно сначала обнаруживаются вторичные ошибки, т. е. последствия и результаты проявления некоторых исходных дефектов, которые следует квалифицировать как первичные ошибки (по причине вложенных аномалий). Локализация и корректировка первичных ошибок приводят к устранению ошибок, первоначально обнаруживаемых в результатах функционирования программ» [10].

В модели Ямады — Нарихиса — Охтера [1] ошибки также делятся на два вида. К ошибкам первого вида относятся легко обнаруживаемые ошибки. К ошибкам второго вида относятся трудно обнаруживаемые ошибки.

Стратегия 3. Этап тестирования продолжается, пока не будет обнаружено или подтверждено отсутствие N_x первичных ошибок вместе взятых (ошибок первого, блокирующего пакета). После этого тестирование приостанавливается, обнаруженные первичные ошибки полностью исправляются с интенсивностями μ_1 . Затем таким же образом отладка продолжается для вторичных ошибок (второго пакета), а исправление найденных ошибок осуществляется с интенсивностями μ_2 . Размеченный граф, моделирующий стратегию 3 отладки, представлен на рис. 3.



■ Рис. 2. Граф переходов для стратегии 2

Количество состояний такой системы

$$N_c = ((N_x + 1)(N_x + 2)/2 + (N - N_x + 1)(N - N_x + 2)/2) - 1. \quad (14)$$

Система дифференциальных уравнений, описывающих данный размеченный граф, в общем виде может быть представлена как

$$\begin{aligned} \frac{dP_{i,j}(t)}{dt} = & \delta(N_x + 1 - i - j) [\delta(i)P_{i-1,j}(t)\omega_{i+j}, \lambda_{i+j} - \\ & - \delta(N_x - i - j)P_{i,j}(t)\lambda_{i+j+1} + \delta(j)P_{i,j-1}(t)(1 - \omega_{i+j})\lambda_{i+j} + \\ & + \delta(i + j + 1 - N_x) [\delta(j)P_{i+1,j-1}(t)\mu_1 - \delta(i)P_{i,j}(t)\mu_1]] + \\ & + \delta(j + 1 - N_x) [\delta(j - N_x)P_{i,j-1}(t)(1 - \omega_{i+j})\lambda_{i+j} + \\ & + \delta(j + 1 - N) [\delta(N - N_x - i)P_{i+1,j-1}(t)\mu_2 - \\ & - \delta(i)P_{i,j}(t)\mu_2] + \delta(i)P_{i-1,j}(t)\omega_{i+j})\lambda_{i+j} - \\ & - \delta(N - i - j)P_{i,j}(t)\lambda_{i+j+1}]. \end{aligned} \quad (15)$$

Базовую модель [9] можно рассматривать в качестве стратегии 0, так как тестирование и исправление ошибок в этом случае идет параллельно (тестирование во время исправления ошибок не приостанавливается).

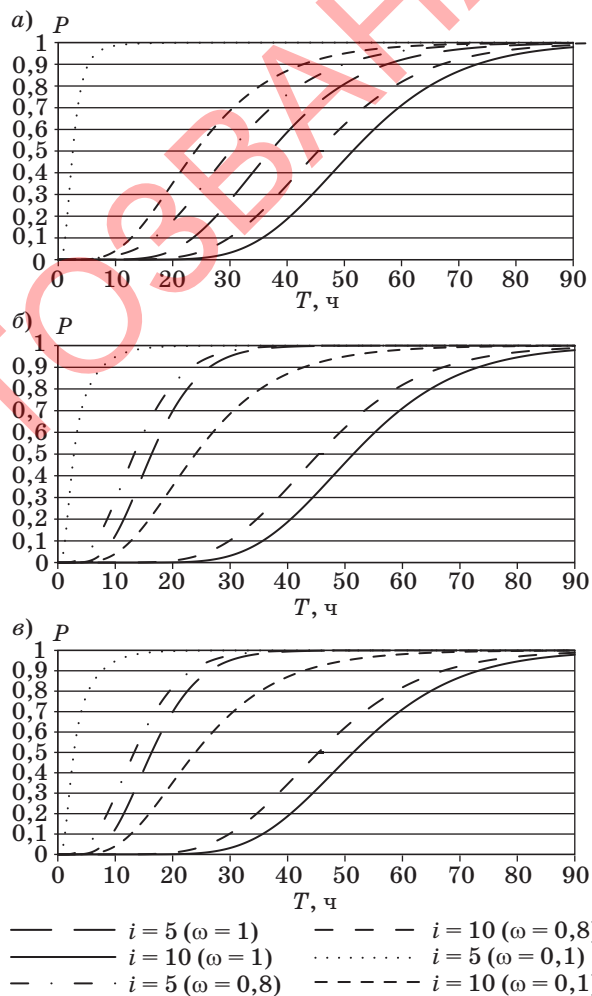
Вычислительный эксперимент

Предложенные модели позволяют не только рассчитать показатели надежности ПС, представленные формулами (4)–(12) для каждой стратегии в отдельности, но и выработать практические рекомендации по организации испытаний программ с целью улучшить характеристики процессов поиска и устранить ошибки. Этого можно достичь сравнением и выбором лучшей для заданного показателя качества стратегии испытаний.

Для иллюстрации таких возможностей моделей приведем результаты расчета некоторых показателей процесса испытаний и состояния ПС. Для всех моделей (стратегий) расчет выполнен при следующих исходных данных. Предполагается, что изначально в ПС содержится $N = 10$ ошибок со средними длительностями интервалов времени тестирования, соответственно равными 0,3; 0,3; 0,3; 0,7; 0,7; 1; 2; 3; 5; 10 ч. Длительность интервала времени исправления ошибок равна 3 ч. Значения вероятностей обнаружения ошибок $\omega_i, i = 0, 1, \dots, N$ указываются непосредственно на графиках и в таблицах, так как этот параметр для данной работы наиболее интересен. На рис. 4, а–в представлены графики функций распределения времени устранения не менее $i (i = 5, 10)$ ошибок для моделей (стратегий) 1, 2, 3 соответственно. С их помощью можно оценить время отладки $T_{N_{тр}, P_{тр}}$, которое требуется

для устранения заданного количества ошибок $N_{тр}$ с заданной вероятностью $P_{тр}$ по формуле (12). Время $T_{N_{тр}, P_{тр}}$ исправления всех ошибок не зависит от выбора стратегии отладки, но зависит от значения вероятностей обнаружения ошибок $\omega_i, i = 0, 1, \dots, N$. Пример такой оценки приведен в табл. 1.

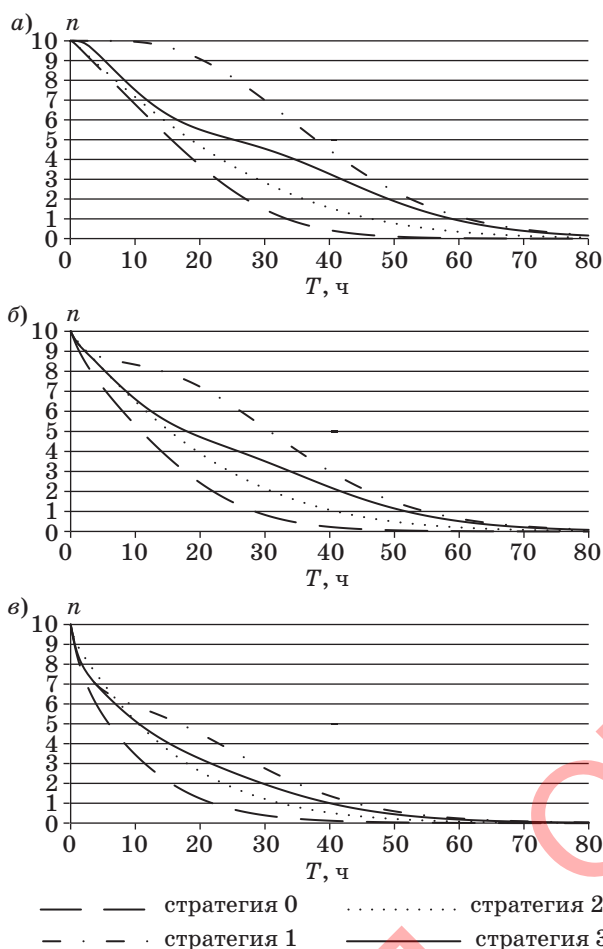
Проведено сравнение стратегий по среднему числу n неустранимых ошибок (рис. 5, а–в) при



■ Рис. 4. Функции распределения времени для стратегии 1 (а), 2 (б) и 3 (в) ($N_x = 5$)

■ Таблица 1. Расчет времени отладки по вероятности отсутствия ошибок

Вероятность $P_{тр}$ отсутствия ошибки	Требуемое время отладки $\tau_{тр}$, ч, при вероятности обнаружения ошибок		
	$\omega_i = 1$	$\omega_i = 0,8$	$\omega_i = 0,1$
0,8	64,9	58,8	35,4
0,9	73,6	67,1	42,9
0,95	81,0	74,7	50,0
0,98	89,9	84,5	59,4



■ Рис. 5. Сравнение стратегий при $\omega = 1$ (а), $\omega = 0,8$ (б) и $\omega = 0,5$ (в)

значения вероятностей их обнаружения $\omega_i = 1; 0,8; 0,5, i = 0, 1, \dots, N$, соответственно. Эти графики могут использоваться для расчета числа не устраненных в программе ошибок к моменту времени τ с момента начала отладки. Пример такого расчета приведен в табл. 2.

Представленные результаты дают наглядную картину того, как соотносятся между собой рассмотренные стратегии отладки ПС по приведенным показателям и как влияют на эти показатели значения вероятностей обнаружения ошибок. Окончательный выбор модели для оценивания характеристик надежности исследуемых ПС производится как с учетом особенностей конкретной стадии их жизненного цикла, так и с учетом вычислительных, экономических, временных, квалификационных и других возможностей пользователей и разработчиков.

Обратим внимание на устойчивость решения построенных дифференциальных уравнений марковского процесса. Решение уравнений для всех вариантов стратегий, как и в работах [11–13], выполнялось методом Рунге — Кутты. При этом

■ Таблица 2. Расчет числа ошибок, не устраненных при отладке

Время τ , ч	Среднее число неустраненных ошибок			
	Стратегия 0	Стратегия 1	Стратегия 2	Стратегия 3 ($N_x = 5$)
Вероятность обнаружения ошибок $\omega_i = 1$				
10	6,7	9,9	7,1	7,5
20	3,6	9,0	4,6	5,5
40	0,4	4,4	1,5	3,2
60	0,0	1,1	0,3	0,9
80	0,0	0,2	0,0	0,2
Вероятность обнаружения ошибок $\omega_i = 0,8$				
10	5,2	8,3	6,4	6,5
20	2,4	7,1	3,8	4,7
40	0,2	3,0	1,0	2,2
60	0,0	0,6	0,2	0,5
80	0,0	0,0	0,0	0,0
Вероятность обнаружения ошибок $\omega_i = 0,5$				
10	3,4	5,9	5,1	5,1
20	1,2	4,4	2,5	3,2
40	0,1	1,3	0,5	1,0
60	0,0	0,2	0,0	0,2
80	0,0	0,0	0,0	0,0

обеспечивались достаточно приемлемые характеристики устойчивости решения, обусловленные тем, что при используемом методе нумерации состояний соответствующие матрицы состояний являются нижними треугольными. Особенности программной реализации численно-аналитического метода решения систем дифференциальных уравнений для моделей нестационарных систем обслуживания, в том числе с позиций обеспечения устойчивости, рассматриваются в статье [14].

Заключение

Использование в предложенных моделях оценки надежности для каждого модуля, с одной стороны, повышает точность моделирования процессов отладки ПС. С другой стороны, достоинством предложенных моделей, в отличие от известных, является то, что они позволяют учитывать состоятельность (мощность) тестов, используемых для отладки ПС, т. е. предполагают различную степень доверия к различным тестам. Это, в свою очередь, позволяет искать пути повышения характеристик надежности ПС путем формирования тестов, обнаруживающих ошибки с высокой вероятностью. Предложенные модели целесообразно

использовать для оценивания текущего уровня показателей процесса испытаний (состояния надежности) ПС и прогнозирования динамики их изменения в процессе испытаний на различных этапах жизненного цикла. На этапе проектирования могут быть получены первые грубые прогнозы показателей процесса испытаний и состояния ПС на основе рассчитанных к этому моменту характеристик сложности проекта и статистики о процессе обнаружения и исправления ошибок, собранных ранее при разработке подобных про-

ектов. Такая оценка может быть уточнена позже на этапе испытания ПС, когда уже получены более точные данные о вероятностях обнаружения, распределении интервалов времени обнаружения и исправления программных ошибок.

Предложенные нестационарные модели позволяют выбрать наилучшую стратегию испытаний ПС исходя из наличия или отсутствия необходимых ресурсов, времени и статистических данных, распределить работы групп тестеров и разработчиков.

Литература

1. Смагин В. А. Основы теории надежности программного обеспечения. — СПб.: ВКА им. А. Ф. Можайского, 2009. — 355 с.
2. Бубнов В. П., Сафонов В. И. Разработка динамических моделей нестационарных систем обслуживания. — СПб.: Лань, 1999. — 64 с.
3. Тырва А. В., Хомоненко А. Д. Метод планирования тестирования сложных программных комплексов на этапах проектирования и разработки // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. 2009. Т. 4. № 82. С. 125–131.
4. Бубнов В. П., Тырва А. В., Хомоненко А. Д. Обоснование стратегии отладки программ на основе нестационарной модели надежности // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. 2010. Т. 2. № 97. С. 85–92.
5. Карпов Ю. Г. Model Checking. Верификация параллельных и распределенных программных систем. — СПб.: БХВ-Петербург, 2010. — 560 с.
6. Chidamber S. R., Kemerer C. F. A Metrics Suite for Object Oriented Design // IEEE Transactions on Software Engineering. 1994. Vol. 20. Iss. 6. P. 476–493.
7. El-Emam K., Melo W., Machado J.C. The Prediction of Faulty Classes Using Object-oriented Design Metrics // Journal of Systems and Software. 2001. Vol. 56. P. 63–75.
8. Basili V. R., Briand L. C. A Validation of Object-oriented Design Metrics as Quality Indicators // IEEE Transactions on Software Engineering. 1996. Vol. 22. Iss. 10. P. 751–761.
9. Данилов А. И., Данилов А. А. Нестационарные модели процессов испытаний программных средств в условиях риска // Вторая Всерос. науч.-практ. конф. «Современные проблемы создания и эксплуатации вооружения, военной и специальной техники»: сб. ст. — СПб., 2014. С. 199–202.
10. Липаев В. В. Надежность программного обеспечения АСУ. — М.: Энергоиздат, 1981. — 247 с.
11. Новиков И. С. Методы расчета количественных показателей надежности сложных программных комплексов на стадии проектирования и разработки // Тр. СПИИРАН. 2008. Вып. 6. С. 86–111.
12. Bubnov V. P., Tyrva A. V., Khomonenko A. D. Model of Reliability of the Software with Coxian Distribution of Length of Intervals between the Moments of Detection of Errors // Proc. of 34th Annual IEEE Computer Software and Applications Conf. (COMPSAC 2010), Seoul, Korea, 19–23 July 2010. P. 238–243.
13. Bubnov V. P., Khomonenko A. D., Tyrva A. V. Software Reliability Model with Coxian Distribution of Length of Intervals between Errors Detection and Fixing Moments // Proc. of 35th Annual IEEE Computer Software and Applications Conf. Workshops (COMPSACW 2011), Munich, 18–22 July 2011. P. 310–314.
14. Бубнов В. П., Еремин А. С., Сергеев С. А. Особенности программной реализации численно-аналитического метода расчета моделей нестационарных систем обслуживания // Тр. СПИИРАН. 2015. № 1(38). С. 218–232.

UDC 004.052.42

doi:10.15217/issn1684-8853.2015.4.50

Non-stationary Models of Software Testing Strategies with Probabilistic Parameters for Fault Detection

Khomonenko A. D.^a, Dr. Sc., Tech., Professor, komon@mail.ru

Danilov A. I.^b, PhD Tech., Associate Professor, Andrey.Danilov.aad@mail.ru

Danilov A. A.^c, Senior Software Engineer, Andrey.Danilov.aad@mail.ru

^aPetersburg Transport State University, 9, Moskovsky Pr., 190031, Saint-Petersburg, Russian Federation

^bA. F. Mozhaiskii Military Space Academy, 13, Zhdanovskaia St., 197198, Saint-Petersburg, Russian Federation

^cSaint-Petersburg Nokia Solutions and Networks, 12, Sedova St., 192019, Saint-Petersburg, Russian Federation

Problem definition: The modern software development standards demand to specify the schedule and resources for successful project implementation. An important aspect is providing the predefined quality level of the software. Modeling of software testing helps to plan the resources and final quality at early stages of the project execution. The available software testing models do not take into account the probabilistic nature of error detection and resolution. Usually models are based on a binary classification of software modules which are divided into three classes: potentially faulty, definitely faulty and fault-free ones. **Purpose:** The existing software testing models should be improved taking into account the probabilistic nature of error detection and resolution. **Results:** Three dynamic models of processes (strategies) are developed for software testing, using error detection probability for each software module. For each strategy, modified labeled graphs are built, along with differential equation systems and their numerical solutions. This helped to compute probabilistic characteristics of the test processes and states: probability states, distribution functions for fault detection and elimination, mathematical expectations of random variables, amount of the detected or fixed errors. The strategies have been compared by their quality indexes. The proposed models allow us to use the reliability estimates for each individual module. This improves the accuracy of software test process modeling and helps to take into account the viability (power) of the tests. Using these models, we can search for ways to improve software reliability by generating tests which detect errors with the highest probability.

Keywords — Dynamic Process Models, Software Testing, Error, Probability, Software Testing Strategies, Labeled Graphs.

References

- Smagin V. A. *Osnovy teorii nadezhnosti programmno obespecheniia* [Theory of Software Reliability]. Saint-Petersburg, VKA im. A. F. Mozhaiskogo Publ., 2009. 355 p. (In Russian).
- Bubnov V. P., Safonov V. I. *Razrabotka dinamicheskikh modelei nestatsionarnykh sistem obsluzhivaniia* [Dynamic Modelling of Non-stationary Queueing Systems]. Saint-Petersburg, Lan' Publ., 1999. 64 p. (In Russian).
- Tyrva A. V., Khomonenko A. D. Method for Planning the Testing of Complex Software Systems at the Stages of Design and Development. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikatsii. Upravlenie*, 2009, no. 4(82), pp. 125–131 (In Russian).
- Bubnov V. P., Tyrva A. V., Khomonenko A. D. Rationale for the Strategy of Debugging Programs on the Basis of Time-dependent Reliability Models. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikatsii. Upravlenie*, 2010, no. 2(97), pp. 85–92 (In Russian).
- Karpov Iu. G. *Model Checking. Verifikatsiia parallel'nykh i raspredelennykh programmnykh sistem* [Verification of Parallel and Distributed Software Systems]. Saint-Petersburg, BKhV-Peterburg Publ., 2010. 560 p. (In Russian).
- Chidamber S. R., Kemerer C. F. A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*, 1994, vol. 20, iss. 6, pp. 476–493.
- El-Emam K., Melo W., Machado J. C. The Prediction of Faulty Classes Using Object-oriented Design Metrics. *Journal of Systems and Software*, 2001, vol. 56, pp. 63–75.
- Basili V. R., Briand L. C. A Validation of Object-oriented Design Metrics as Quality Indicators. *IEEE Transactions on Software Engineering*, 1996, vol. 22, iss. 10, pp. 751–761.
- Danilov A. I., Danilov A. A. Transient Models of the Software Testing Processes in Risk Conditional. *Vtoraiia Vserossiiskaia nauchno-prakticheskaiia konferentsiia "Sovremennye problemy sozdaniia i ekspluatatsii vooruzheniia, voennoi i spetsial'noi tekhniki"* [Proc. of the Second All-Russian Scientific-Practical Conf. "Modern Problems of Weapons, Military and Special Equipment Creation and Operation"]. Saint-Petersburg, 2014, pp. 199–202 (In Russian).
- Lipaev V. V. *Nadezhnost' programmno obespecheniia ASU* [Software Reliability of Automatic Control Systems]. Moscow, Energoizdat Publ., 1981. 247 p. (In Russian).
- Novikov I. S. Calculation Methods for Quantitative Indicators of Complex Software System Reliability at the Design and Development Stages. *Trudy SPIIRAN*, 2008, no. 6, pp. 86–111 (In Russian).
- Bubnov V. P., Tyrva A. V., Khomonenko A. D. Model of Reliability of the Software with Coxian Distribution of Length of Intervals between the Moments of Detection of Errors. *Proc. of 34th Annual IEEE Computer Software and Applications Conf. (COMPSAC 2010)*, Seoul, Korea, 19–23 July 2010, pp. 238–243.
- Bubnov V. P., Khomonenko A. D., Tyrva A. V. Software Reliability Model with Coxian Distribution of Length of Intervals between Errors Detection and Fixing Moments. *Proc. of 35th Annual IEEE Computer Software and Applications Conf. Workshops (COMPSACW 2011)*, Munich, 18–22 July 2011, pp. 310–314.
- Bubnov V. P., Eremin A. S., Sergeev S. A. Software Implementation Details for Numerically-Analytical Method of Non-Stationary Queueing System Model Calculations. *Trudy SPIIRAN*, 2015, no. 1(38), pp. 218–232 (In Russian).