

УДК 004.052.42

МЕТОДИКА ЧИСЛЕННОГО АНАЛИЗА ЭФФЕКТИВНОСТИ ОТЛАДКИ ПРОГРАММНЫХ СРЕДСТВ

А.И. Данилов^a, А.А. Данилов^b

^a Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург, 197198, Российская Федерация

^b ООО «Нокиа Солюшнз энд Нетворкс», Санкт-Петербург, 192019, Российская Федерация

Адрес для переписки: Andrey.Danilov.aad@mail.ru

Информация о статье

Поступила в редакцию 30.03.17, принята к печати 18.04.17

doi: 10.17586/2226-1494-2017-17-3-543-551

Язык статьи – русский

Ссылка для цитирования: Данилов А.И., Данилов А.А. Методика численного анализа эффективности отладки программных средств // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 3. С. 543–551. doi: 10.17586/2226-1494-2017-17-3-543-551

Аннотация

Предмет исследования. Исследована нестационарная модель надежности программ и на ее основе предложена методика численного анализа эффективности отладки программных средств. Моделирование эффективности процессов отладки программ позволяет комплексно планировать их финальное качество, затрачиваемые ресурсы и необходимое время на различных этапах реализации проектов. **Методы.** Методика основана на предложенной усовершенствованной численной модели отладки программ. Процесс обнаружения ошибок аппроксимируется экспоненциальным законом, а процесс устранения – двухфазным обобщенным распределением Кокса. Процесс отладки программ после аппроксимации представляется марковской системой обслуживания с дискретным множеством состояний и непрерывным временем. Предусмотрена возможность использования вероятностей обнаружения ошибок для каждого модуля при их тестировании. Приведены модифицированный размеченный граф и система дифференциальных уравнений, численное решение которой позволяет вычислить частные показатели целевого эффекта процессов отладки программных средств: вероятности пребывания системы в определенных состояниях, функции распределения времени обнаружения и устранения ошибок, математические ожидания случайных величин, количество обнаруженных или устраненных ошибок. Для комплексного (с учетом затрачиваемых ресурсов) оценивания эффективности этих процессов использован обобщенный показатель – вероятность достижения цели операции (отладки). **Основные результаты.** Разработанная методика применена при исследовании эффективности реализованного проекта. Приведены результаты расчетов как частных показателей целевого эффекта отладки программ, так и комплексного показателя эффективности. Определено требуемое время отладки для достижения заданных показателей качества программ. Выполнен анализ влияния целевого эффекта и времени на эффективность отладки (на вероятность достижения цели операции). **Практическая значимость.** Разработанная методика позволяет учитывать оценки надежности для каждого модуля в отдельности, а аппроксимация двухфазным обобщенным распределением Кокса снимает ограничения на использование законов распределений времени исправления ошибок. Это обобщает известные модели, упрощает подготовку исходных данных, повышает точность моделирования процессов отладки программ и позволяет учитывать состоятельность (мощность) тестов, искать пути повышения показателей надежности программ путем формирования тестов, обнаруживающих ошибки с высокой вероятностью. Предложенная методика позволяет не только рассчитать частные показатели надежности программных средств, но и вычислить комплексный показатель эффективности процессов отладки, выработать практические рекомендации по эффективной организации этих процессов.

Ключевые слова

модель, эффективность, программные средства, ошибка, вероятность, распределение Кокса

NUMERICAL ANALYSIS METHODS OF SOFTWARE TEST EFFICIENCY

A.I. Danilov^a, A.A. Danilov^b

^a Mozhayskiy Military Space Academy, Saint Petersburg, 197198, Russian Federation

^b Saint Petersburg Nokia Solutions and Networks, Saint Petersburg, 192019, Russian Federation

Corresponding author: anna_hab_2016_2@mail.ru

Article info

Received 30.03.17, accepted 18.04.17

doi: 10.17586/2226-1494-2017-17-3-543-551

Article in Russian

For citation: Danilov A.I., Danilov A.A. Numerical analysis methods of software test efficiency. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2017, vol. 17, no. 3, pp. 543–551 (in Russian). doi: 10.17586/2226-1494-2017-17-3-543-551

Abstract

Subject of Research. A nonstationary software testing model is studied. Numerical analysis methods of software testing efficiency based on this model are developed. Modeling of software testing efficiency enables to plan comprehensively the final quality, resources and time required at various project implementation stages. **Methods.** The technique is based on the proposed improved numerical model for software testing. The process of errors detecting is approximated by the exponential law and the process of elimination by generalized two-phase Cox distribution. The software debugging process after approximation is described by Markovian queue with a discrete set of states and continuous time. The possibility is provided to use the probability of errors detection for each module during their testing. The paper presents the modified marked graph and the system of differential equations; its numerical solution gives the possibility to calculate specific indicators for target effect of software debugging process: probability of certain system states, the time distribution function for errors detection and elimination, the mathematical expectation of random variables and the number of detected or corrected errors. The probability of operating goal achievement (testing) is used as an overall index for the integrated effectiveness evaluating of these processes (including required resources). **Main Results.** The developed methodology is applied for effectiveness research of the actual project. The private indicators of target effect and integrated efficiency indicator for testing are calculated. The required testing time for specified software quality indicators achievement is identified. The analysis of target effect and time influence on testing effectiveness is performed (on the probability of operation goal achieving). **Practical Relevance.** The suggested methodology enables to take into account the reliability assessment for each module separately. The Cox approximation removes restrictions on the usage of arbitrary time distribution for fault resolution duration. That generalizes well-known models, simplifies the initial data preparation, improves the accuracy of software test process modeling and helps to take into account the viability (power) of the tests. With these models we can search for the ways of software reliability improvement by generating tests that detect errors with high probability. This methodology gives the possibility to calculate not only the private reliability software indicators, but the integrated indicator of software testing process effectiveness and to develop practical recommendations for effective organization of these processes.

Keywords

model, effectiveness, software, error, probability, Cox distribution

Введение

Обеспечение требуемого финального качества (и в первую очередь надежности) программных средств (ПС), сокращение затрат и сроков на их разработку могут быть достигнуты не только за счет использования более совершенных технологий, сред разработки и средств автоматизации отладки, но и путем более эффективного планирования и организации проведения работ при реализации проектов. Следовательно, планирование и распределение ресурсов программных проектов являются важными управленческими задачами. Для решения этих задач могут быть использованы средства математического моделирования надежности ПС и эффективности процессов их разработки (отладки). К настоящему времени в научной литературе предложено большое количество различных моделей для оценивания и предсказания надежности ПС на различных этапах их жизненного цикла. Подробный обзор и описание таких моделей даны в [1–12]. Анализ показывает, что все известные модели в совокупности имеют ряд недостатков: модели не обладают достаточной общностью (универсальностью); описывают процессы обнаружения, но не устранения ошибок; моделируют процессы обнаружения и исправления всех прогнозируемых ошибок, хотя не все эти ошибки могут быть допущены (найжены) с единичной вероятностью, т.е. модели не учитывают вероятностный характер обнаружения ошибок и состоятельность (мощность) используемых тестов; для оценивания надежности ПС и эффективности процессов их отладки используют частные показатели. Однако при реализации проектов далеко не безразлично, какой ценой, за какое время и к какому сроку получен целевой эффект (финальное качество ПС). Чаще всего именно расход ресурсов и времени определяют стратегию реализации проектов.

Целью настоящей работы является разработка методики численного анализа эффективности отладки ПС. Эта методика базируется на предлагаемой обобщенной динамической модели процессов отладки ПС, в которой предусматривается возможность учитывать вероятностные параметры обнаружения ошибок для каждого модуля при тестировании программ или состоятельность используемых тестов. Для реализации общих (не экспоненциальных) предположений о характере распределения интервалов времени исправления найденных ошибок (с целью повышения точности моделирования) используется аппроксимация по методу Кокса. Наряду с частными показателями, в качестве комплексного показателя эффективности (ПЭ) процессов отладки ПС используется вероятность достижения цели.

Таким образом, объектом исследования в предлагаемой работе является ПС на этапе проектирования, реализации или тестирования (отладки). Предмет исследования и разработки – методика численного анализа эффективности отладки ПС.

Методика численного анализа эффективности отладки программных средств*1. Определение вероятности обнаружения ошибок при тестировании программного модуля.*

При задании исходных данных для подобных моделей распространенным подходом является использование собранной статистики уже завершенных проектов. Однако для учета особенностей конкретного проекта широко используются различные программные метрики, характеризующие программный продукт (его структуру, сложность, интерфейсы, факторы разработки и др.). Так, с точки зрения совре-

менного объектно-ориентированного программирования целесообразно использовать наборы метрик Чайдамбера–Кемерера и Бансия–Дэвиса [13, 14]. Значения метрик собираются из документов проектирования. На основе полученных метрик решается задача классификации модулей. Так, вероятность наличия ошибки в модуле ω_i можно оценить с использованием логистической регрессии и выразить формулой [13, 14]

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

где β_j – параметры логистической регрессии, которые определяются методом максимального правдоподобия; x_j – значение метрики сложности.

По принятому в [2, 9] подходу модели строятся на результате двоичной классификации модулей ПС (надежные – ненадежные). Новизна предлагаемой модели заключается в отказе от двоичной классификации модулей и использовании в ней оценки надежности для каждого модуля.

2. *Аппроксимация произвольных законов распределения интервалов времени исправления найденных ошибок двухэтапным распределением Кокса.*

Как показано в [15], произвольное распределение случайной величины можно аппроксимировать распределением фазового типа (гиперэкспоненциальным, Эрланга или Кокса). Эта аппроксимация позволяет свести случайный процесс к марковскому, получить и численно решить систему дифференциальных уравнений, которая полностью описывает поведение соответствующей модели. Диаграмма двухфазного обобщенного распределения Кокса (C_2) представлена на рис. 1.

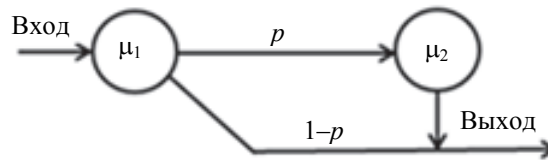


Рис. 1. Двухфазное распределение Кокса

Оно представляет собой смесь двух экспоненциальных фаз с интенсивностями μ_1 и μ_2 . При этом после прохождения первой фазы с некоторыми вероятностями p и $1-p=p'$ выбираются продолжение процесса (следующая фаза) или завершение процесса. Моменты распределения C_2 с вероятностью p являются моментами свертки показательного распределенных задержек в фазах, а с дополнительной к ней – моментами задержки в первой фазе. Таким образом, для расчета параметров C_2 по трем начальным моментам имеем следующую систему уравнений:

$$\begin{cases} \frac{1-p}{\mu_1} + p \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} \right) = f_1 \\ \frac{1-p}{\mu_1^2} + p \left(\frac{1}{\mu_1^2} + \frac{1}{\mu_1 \mu_2} + \frac{1}{\mu_2^2} \right) = f_2 \\ \frac{1-p}{\mu_1^3} + p \left(\frac{1}{\mu_1^3} + \frac{1}{\mu_1^2 \mu_2} + \frac{1}{\mu_1 \mu_2^2} + \frac{1}{\mu_2^3} \right) = f_3, \end{cases} \quad (1)$$

где $f_i = g_i / i!$, $i = \overline{1,3}$; g_i – i -й начальный момент исходного распределения.

Решая систему уравнений (1), можно получить параметры аппроксимирующего распределения C_2 :

$$\begin{cases} p = \mu_2 (f_1 \mu_1 - 1) / \mu_1 \\ \mu_1 = (\mu_2 f_1 - 1) / (\mu_2 f_2 - f_1) \\ \mu_2 = (f_1 f_2 - f_3 \pm \sqrt{D}) / (2(f_2^2 - f_1 f_3)), \end{cases} \quad (2)$$

где $D = (f_1 f_2 - f_3)^2 - 4(f_2^2 - f_1 f_3)(f_1^2 - f_2)$.

В простейшем случае при $p = 1$ и аппроксимации по двум начальным моментам из (1) получаем

$$\mu_{1,2} = (f_1 \pm \sqrt{4f_2 - 3f_1^2}) / (2(f_1^2 - f_2)). \quad (3)$$

3. *Обобщенная динамическая модель процессов отладки ПС.*

Рассмотрим следующую стратегию распределения времени и ресурсов между этапами тестирования и исправления найденных ошибок (другие стратегии и модели опубликованы в [2–9]). Обнаруженные ошибки устраняются последовательно по мере их выявления, а тестирование во время исправления найденных ошибок не приостанавливается.

После аппроксимации представим процессы обнаружения и исправления ошибок марковским процессом с дискретным множеством состояний и непрерывным временем. Диаграмма (размеченный граф) состояний и переходов таких процессов показана на рис. 2.

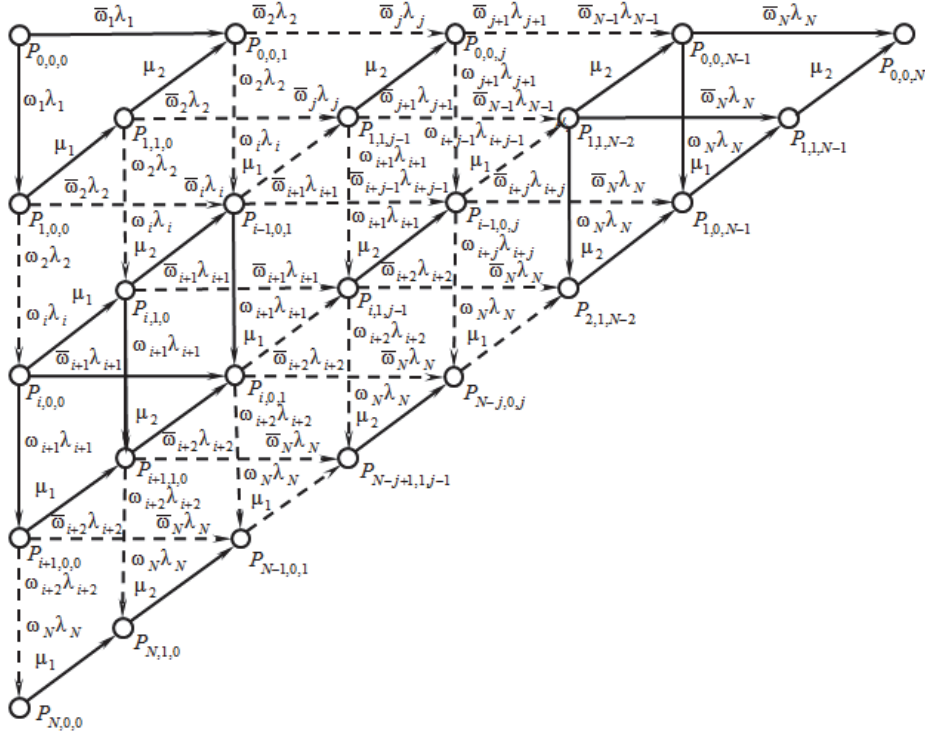


Рис. 2. Граф переходов между состояниями марковской системы

При составлении графа учитывалось, что ПС может содержать N ошибок. Вероятность обнаружения (необнаружения) i -ой ошибки равна ω_i ($\bar{\omega}_i = 1 - \omega_i$). Время обнаружения (необнаружения) ошибок распределено по экспоненциальному закону с интенсивностями $\omega_i \lambda_i$ ($\bar{\omega}_i \lambda_i$), зависящими от номера ошибки. Произвольное распределение времени устранения всех ошибок аппроксимируется двухэтапным распределением Кокса с параметром $p=1$ ($1-p=p'=0$) и интенсивностями экспоненциальных фаз μ_1, μ_2 .

Состояния (i, k, j) такой системы в каждый момент времени будем характеризовать количеством обнаруженных, но еще не исправленных ошибок i , числом устраненных (сумма найденных и исправленных ошибок) j ($i, j = 0, N$) и фазой k ($k=0,1$) распределения Кокса длины интервалов времени исправления ошибок. Вероятности пребывания системы в этих состояниях обозначим $P_{i,k,j}(t)$. Переход системы из состояния (i, k, j) в состояние $(i+1, k, j)$ означает, что при тестировании была обнаружена $(i+1)$ -я ошибка. Переход из состояния (i, k, j) в состояние $(i, k, j+1)$ означает, что при тестировании подтверждено отсутствие ошибки или она не найдена. Переход из состояния $(i, 0, j)$ в состояние $(i-1, 0, j+1)$, означает, что была исправлена $(j+1)$ -я обнаруженная ошибка, устранение которых описывается двухэтапным распределением Кокса с интенсивностями μ_1, μ_2 . В таком представлении общее число состояний системы вычисляется по формуле $N_c = (N+1)^2$. Процессы в рассматриваемой марковской цепи описываются системой из N_c дифференциальных уравнений:

$$\begin{aligned} \frac{dP_{i,k,j}(t)}{dt} = & \delta(i-k)P_{i-1,k,j}(t)\omega_{i+j}\lambda_{i+j} - \delta(N-i-j)P_{i,k,j}(t)\lambda_{i+j+1} - \\ & -\delta(i)\delta(1-k)P_{i,k,j}(t)\mu_1 + \delta(k)[P_{i,0,j}(t)p\mu_1 - P_{i,1,j}(t)\mu_2] + \delta(j)\delta(1-k)[P_{i+1,1,j-1}(t)\mu_2 + \\ & + P_{i,0,j-1}(t)(1-\omega_{i+j})\lambda_{i+j} + P_{i+1,0,j-1}(t)(1-p)\mu_1] + \delta(j)\delta(k)P_{i,1,j-1}(t)(1-\omega_{i+j})\lambda_{i+j}, \end{aligned} \quad (4)$$

где $\delta(m) = \begin{cases} 1, & \text{если } m > 0 \\ 0, & \text{если } m \leq 0 \end{cases}$; $i = \overline{0, N}$; $j = \overline{0, N-i}$; $k=0, 1$; значения параметров C_2 : p, μ_1, μ_2 вычисляются по формулам (1)–(3).

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

$P_{i,k,j}(0) = \begin{cases} 1, & \text{если } i+k+j=0 \\ 0, & \text{если } i+k+j \neq 0 \end{cases}$, можно найти численное решение соответствующей задачи Коши для произвольного значения времени t . Используя решение предложенной системы (4), можно получить ряд важных частных вероятностных показателей процесса испытаний и состояния ПС.

4. *Вычисление частных вероятностных показателей эффективности процесса испытаний и состояния ПС.*

Вероятность того, что в процессе испытаний было найдено ровно i ошибок (сумма найденных, но не исправленных и устраненных ошибок):

$$R_i(t) = \sum_{j=0}^i P_{i-j,0,j}(t) + \sum_{j=1}^i P_{i-j+1,j-1}(t). \quad (5)$$

Математическое ожидание числа найденных ошибок:

$$N_R(t) = \sum_{i=1}^N iR_i(t). \quad (6)$$

Вероятность того, что в процессе испытаний было устранено (исправлено или отсутствовало) ровно j ошибок:

$$P_j(t) = \sum_{i=0}^{N-j} P_{i,0,j}(t) + \sum_{i=1}^{N-j} P_{i,1,j}(t). \quad (7)$$

Математическое ожидание числа устраненных ошибок:

$$N_P(t) = \sum_{j=1}^N jP_j(t). \quad (8)$$

Функция $F_i(t)$ распределения времени устранения не менее i ошибок:

$$F_i(t) = \sum_{l=i}^N P_l(t), i = \overline{0, N}. \quad (9)$$

Время отладки, которое требуется, чтобы устранить N_{TP} ошибок с заданной вероятностью P_{TP} :

$$T_{N_{TP}, P_{TP}} = t | F_{N_{TP}}(t) \geq P_{TP}, \quad (10)$$

где N_{TP} – требуемое количество ошибок (N); P_{TP} – требуемая вероятность устранения N_{TP} ошибок.

Вероятность безотказной работы $P(t, \tau)$ в течение интервала τ после испытаний в течение времени t :

$$P(t, \tau) = 1 - \sum_{j=0}^{N-1} P_{0,0,j}(t)(1 - e^{-\omega_j + \lambda_j + \tau}). \quad (11)$$

5. *Вычисление обобщенного показателя эффективности отладки программных средств.*

Приведенные вероятностные показатели качества процессов отладки ПС можно отнести к частным показателям целевого эффекта. При комплексном оценивании эффективности этих процессов необходимо использовать обобщенный показатель [16], который называется вероятностью достижения цели операции $P_{ДЦ}$ и в общем случае вычисляется по формуле:

$$P_{ДЦ} = P(\hat{Y}_{<n>} \in \{\hat{Y}_{<n>}^A\}) = P(\hat{Y}_{<n>} \underset{<}{\geq} \hat{Z}_{<n>}) = \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_n \Phi_{\hat{Y}_{<n>}}(\mathbf{Z}_{<n>}) dF_{\hat{Z}_{<n>}}(\mathbf{Z}_{<n>}), \quad (12)$$

где $\hat{Y}_{<n>}$ – n -мерный ($n=n_1+n_2+n_3$) вектор показателя качества результатов операции, который должен учитывать минимум три аспекта: результативность – $\mathbf{Y}_{<n_1>}^{(1)}$, ресурсоемкость – $\mathbf{Y}_{<n_2>}^{(2)}$ и оперативность – $\mathbf{Y}_{<n_3>}^{(3)}$;

$\hat{Z}_{<n>}$ – n -мерный вектор требований к вектору $\hat{Y}_{<n>}$ (или область допустимых значений вектора $\hat{Y}_{<n>}$).

Как видно из (12), для вычисления вероятности $P_{ДЦ}$ необходимо знать интегральный закон распределения $\Phi_{\hat{Y}_{<n>}}(\mathbf{Y}_{<n>})$ случайного вектора $\hat{Y}_{<n>}$ и закон распределения $F_{\hat{Z}_{<n>}}(\mathbf{Z}_{<n>})$ случайного вектора $\hat{Z}_{<n>}$. Для получения значений $P_{ДЦ}$ могут использоваться следующие методы: аналитический, численный, статистических испытаний, статистического имитационного моделирования. Аналитическое построение функции $\Phi_{\hat{Y}_{<n>}}(\mathbf{Y}_{<n>})$ распределения показателя $\hat{Y}_{<n>}$ и вычисление $P_{ДЦ}$ представляет трудоемкую задачу. Решение этой задачи может быть упрощено в случае, если известны функциональные связи между компонентами вектора $\hat{Y}_{<n>}$ (операционный функционал – совокупность операционной функции и функции связи). Операционная функция (ОФ) – это соотношение, описывающее зависимость целевого эффекта от расходуемых операционных ресурсов и времени. Функция связи (ФС) – это балансные соотношения между различными характеристиками результатов операции (в частности, между ресурсами различных видов). Для нашего случая, не нарушая общности рассуждений, ограничимся симплексной канонической формой представления показателя качества результатов операции, при которой размер-

ность вектора $\hat{Y}_{\langle n \rangle}$ равна трем, т.е. $n=3$. Тогда: $\hat{Y}_{\langle 3 \rangle} = \langle \hat{y}_1, \hat{y}_2, \hat{y}_3 \rangle = \langle \hat{\vartheta}, \hat{r}, \hat{\tau} \rangle$, а его компоненты связаны монотонными зависимостями. В общем случае целевой эффект ϑ , вычисленный по формулам (5)–(11), связан с ресурсами такой зависимостью ОФ: $\vartheta = R(r, \tau)$. Пусть целевой эффект ϑ и расход операционного времени τ связаны с расходом ресурсов r (выделенных на отладку программ денежных средств) ФС, для которых существуют обратные функции: $\vartheta = R(r) \Rightarrow r=R^{-1}(\vartheta)$; $\tau=S(r) \Rightarrow r=S^{-1}(\tau)$.

Очевидно, что в нашем случае функции (вероятностные показатели) R и S – монотонно возрастающие. Это положение можно аргументировать так. При фиксированной технологии, чем больше в ходе операции расходуются ресурсы (денежные средства), тем большее время можно уделить отладке ПС. Пусть известна функция распределения $F_r(r)$ количества расходуемых ресурсов (денежных средств) r , которая является генеральной компонентой. При указанных предположениях найдем функцию распределения вектора $\hat{Y}_{\langle 3 \rangle}$:

$$\begin{aligned} \Phi_{\hat{Y}_{\langle 3 \rangle}}(\mathbf{Y}_{\langle 3 \rangle}) &= \Phi_{\langle \hat{\vartheta}, \hat{r}, \hat{\tau} \rangle}(\langle \vartheta, r, \tau \rangle) = P[(\hat{\vartheta} \geq \vartheta) \cap (\hat{r} \leq r) \cap (\hat{\tau} \leq \tau)] = \\ &= P[(R(\hat{r}) \geq \vartheta) \cap (\hat{r} \leq r) \cap (S(\hat{r}) \leq \tau)] = P[(\hat{r} \geq R^{-1}(\vartheta)) \cap (\hat{r} \leq r) \cap (\hat{r} \leq S^{-1}(\tau))] = \\ &= P[R^{-1}(\vartheta) \leq \hat{r} \leq \min\{r, S^{-1}(\tau)\}] = F_r[\min\{r, S^{-1}(\tau)\}] - F_r[R^{-1}(\vartheta)]. \end{aligned}$$

Если известной генеральной компонентой будет операционное время с функцией распределения $F_{\tau}(\tau)$, то целевой эффект ϑ и расход ресурсов r будут связаны с расходом операционного времени τ ФС, для которых существуют обратные функции: $\vartheta = R(\tau) \Rightarrow \tau=R^{-1}(\vartheta)$; $r=S(\tau) \Rightarrow \tau=S^{-1}(r)$. Тогда функция распределения вектора $\hat{Y}_{\langle 3 \rangle}$ может быть вычислена по следующей формуле:

$$\begin{aligned} \Phi_{\hat{Y}_{\langle 3 \rangle}}(\mathbf{Y}_{\langle 3 \rangle}) &= \Phi_{\langle \hat{\vartheta}, \hat{r}, \hat{\tau} \rangle}(\langle \vartheta, r, \tau \rangle) = P[(\hat{\vartheta} \geq \vartheta) \cap (\hat{r} \leq r) \cap (\hat{\tau} \leq \tau)] = \\ &= P[(R(\hat{\tau}) \geq \vartheta) \cap (S(\hat{\tau}) \leq r) \cap (\hat{\tau} \leq \tau)] = P[(\hat{\tau} \geq R^{-1}(\vartheta)) \cap (\hat{\tau} \leq S^{-1}(r)) \cap (\hat{\tau} \leq \tau)] = \\ &= P[R^{-1}(\vartheta) \leq \hat{\tau} \leq \min\{S^{-1}(r), \tau\}] = F_{\tau}[\min\{S^{-1}(r), \tau\}] - F_{\tau}[R^{-1}(\vartheta)]. \end{aligned} \tag{13}$$

Таким образом, методика численного оценивания эффективности отладки программных средств состоит из реализации пяти предложенных пунктов. Для получения методики численного анализа эффективности отладки программных средств необходимо к указанной методике добавить анализ чувствительности и влияния ПЭ к изменениям параметров, определяющих условия задачи. Эта задача решается методами теории чувствительности.

Вычислительный эксперимент

Предложенная методика позволяет не только рассчитать частные показатели надежности ПС, представленные формулами (5)–(11), но и вычислить комплексный ПЭ процессов отладки ПС $P_{\text{дц}}$, выработать практические рекомендации по эффективной организации отладки программ. Этого можно достичь с помощью математического аппарата теории чувствительности (коэффициентов чувствительности и влияния), проанализировав зависимость ПЭ от параметров, определяющих условия задачи.

Для иллюстрации таких возможностей модели приведем результаты расчета некоторых частных показателей процесса испытаний и состояния ПС. Расчеты выполнены по формуле (9) для следующих исходных данных. Предполагается, что изначально в ПС содержится $N = 10$ ошибок со средними длительностями интервалов времени тестирования, равными 0,3; 0,3; 0,3; 0,7; 0,7; 1; 2; 3; 5; 10 (ч) соответственно. Распределение времени устранения каждой ошибки одинаково и аппроксимируется распределением Кокса с интенсивностями экспоненциальных фаз $\mu_1 = 1,2375$, $\mu_2 = 0,591$. Значения вероятностей обнаружения ошибок ω_i , $i = \overline{1, N}$, указаны непосредственно на графике. На рис. 3 представлены графики функций распределения времени устранения не менее i , ($i=5-10$) ошибок для рассматриваемой модели. Приведенные графики функций дают возможность оценить время отладки $T_{N_{\text{тр}}, P_{\text{тр}}}$, которое требуется для устранения заданного количества ошибок $N_{\text{тр}}$ с заданной вероятностью $P_{\text{тр}}$. Время $T_{N_{\text{тр}}, P_{\text{тр}}}$ устранения ошибок в данном случае зависит от значения вероятностей обнаружения ошибок. Получены данные о зависимости вероятности исправления всех $N=10$ ошибок от различных значений коэффициента вариации v распределения интервалов времени устранения ошибок. Эти данные позволили рассчитать значения требуемого времени отладки по заданной вероятности при $v = 0,5; 0,75; 1,0$, которое представлено в таблице.

Расчеты показывают, что переход от экспоненциальной модели надежности ПС к неэкспоненциальной дает преимущество в точности прогноза. Эта информация необходима руководителям проекта для планирования сроков сдачи ПС для определения требуемых трудозатрат и ресурсов, формирования команд программистов и тестеров.

Для иллюстрации возможностей модели по исследованию эффективности процессов отладки ПС конкретизируем условия задачи. Пусть целевой эффект $\hat{\vartheta}$ есть вероятность устранения не менее i программных ошибок, связанная со временем отладки τ соотношением (9) $\vartheta = F_i(\tau) = \sum_{l=1}^N P_l(\tau) \Rightarrow \tau = F_i^{-1}(\vartheta)$. Функции распределения этой вероятности для $i=5$ и 10 ошибок представлены на рис. 3.

Вероятность отсутствия ошибки	Требуемое время отладки, ч		
	Значения коэффициента вариации		
	$v=0,5$	$v=0,75$	$v=1,0$
0,7	30,7	31,9	32,3
0,8	33,7	35,5	36,4
0,9	41,1	41,7	42,2
0,95	48,4	48,6	49,1

Таблица. Расчет времени отладки по вероятности отсутствия ошибок

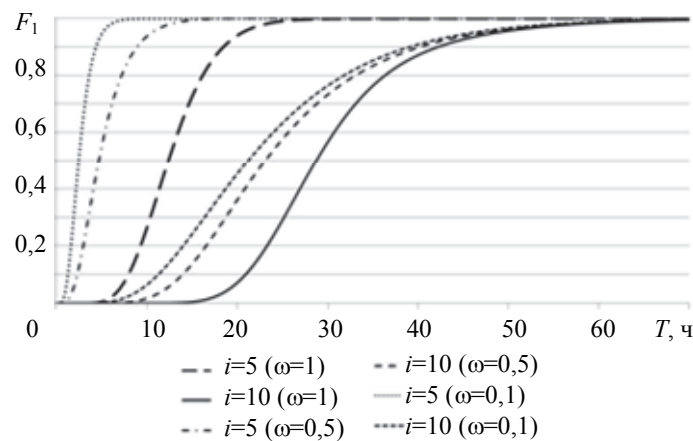


Рис. 3. Функции распределения времени устранения ошибок

Продолжительность времени отладки ПС $\hat{\tau}$ от момента начала отладки t' до момента ее окончания t'' , связанного с израсходованием выделенных на отладку ресурсов (денежных средств) или достижением заданной вероятности целевого эффекта, случайна и подчинена равномерному закону распределения на интервале $\hat{\tau} \in [t', t'']$, т.е.

$$F_i(\tau) = \frac{\tau - t'}{t'' - t'} \Pi(\tau; t', t'') + \Delta(\tau - t''),$$

где $\Delta(x)$ – «селектор луча» – индикатор полубесконечного интервала $[0, \infty)$; $\Pi(\tau; t', t'')$ – «селектор интервала» – индикатор интервала $[t', t'']$.

Расходуемые в ходе отладки ПС ресурсы r (денежные средства) пропорциональны времени отладки, т.е. $r = a\tau$. Таким образом, известной генеральной компонентой будет операционное время с функцией распределения $F_i(\tau)$. Цель отладки ПС достигается, если $[(\hat{\vartheta} \geq \vartheta_T) \cap (\hat{r} \leq r_{II}) \cap (\hat{\tau} \leq \tau_D)] \equiv U$, т.е. одновременно выполняются три указанных события. Здесь ϑ_T – требуемый целевой эффект; r_{II} – предельные затраты ресурсов; τ_D – директивное время.

По формуле (13) найдем закон распределения вектора $\hat{Y}_{\langle 3 \rangle}$

$$\begin{aligned} \Phi_{\hat{Y}_{\langle 3 \rangle}}(\mathbf{Y}_{\langle 3 \rangle}) &= \Phi_{\langle \hat{\vartheta}, \hat{r}, \hat{\tau} \rangle}(\langle \vartheta, r, \tau \rangle) = P[(\hat{\vartheta} \geq \vartheta) \cap (\hat{r} \leq r) \cap (\hat{\tau} \leq \tau)] = \\ &= P[(F_i(\hat{\tau}) \geq \vartheta) \cap (a\hat{\tau} \leq r) \cap (\hat{\tau} \leq \tau)] = P[(\hat{\tau} \geq F_i^{-1}(\vartheta)) \cap (\hat{\tau} \leq \frac{r}{a}) \cap (\hat{\tau} \leq \tau)] = \\ &= P[F_i^{-1}(\vartheta) \leq \hat{\tau} \leq \min\{\frac{r}{a}, \tau\}] = F_i[\min\{\frac{r}{a}, \tau\}] - F_i[F_i^{-1}(\vartheta)]. \end{aligned}$$

С учетом равномерного закона распределения $F_i(\tau)$ и области допустимых значений вектора $\hat{Y}_{\langle 3 \rangle}$, которая определяется декартовым (прямым) произведением трех множеств $[\hat{z}_1, \infty) \times (-\infty, \hat{z}_2] \times (-\infty, \hat{z}_3]$, получаем выражение для закона распределения вектора $\hat{Y}_{\langle 3 \rangle}$:

$$\begin{aligned} \Phi_{\hat{Y}_{\langle 3 \rangle}}(\mathbf{Y}_{\langle 3 \rangle}) = & \Phi_{\langle \hat{\theta}, \hat{r}, \hat{\tau} \rangle}(\langle \theta, r, \tau \rangle) = \left[\frac{\min\{\frac{r}{a}, \tau\} - t'}{t'' - t'} - \frac{F_i^{-1}(\theta) - t'}{t'' - t'} \right] \Pi(\theta; F_i(t'), F_i(t'')) \times \Pi(r; at', at'') \times \\ & \times \Pi(\tau; t', t'') + \left[\frac{\min\{\frac{r}{a}, \tau\} - t'}{t'' - t'} \right] \Delta(F_i(t'') - \theta) \times \Pi(r; at', at'') \times \Pi(\tau; t', t'') + \left[\frac{\tau - t'}{t'' - t'} - \frac{F_i^{-1}(\theta) - t'}{t'' - t'} \right] \times \\ & \times \Pi(\theta; F_i(t'), F_i(t'')) \times \Delta(r - at'') \times \Pi(\tau; t', t'') + \left[\frac{r - at'}{a(t'' - t')} - \frac{F_i^{-1}(\theta) - t'}{t'' - t'} \right] \Pi(\theta; F_i(t'), F_i(t'')) \times \\ & \times \Pi(r; at', at'') \times \Delta(\tau - t'') + \frac{\tau - t'}{t'' - t'} \Delta(F_i(t'') - \theta) \times \Delta(r - at'') \times \Pi(\tau; t', t'') + \frac{r - at'}{a(t'' - t')} \Delta(F_i(t'') - \theta) \times \\ & \times \Pi(r; at', at'') \times \Delta(\tau - t'') + \frac{F_i^{-1}(\theta) - t'}{t'' - t'} \Pi(\theta; F_i(t'), F_i(t'')) \times \Delta(r - at'') \times \Delta(\tau - t'') + \\ & + \Delta(F_i(t'') - \theta) \times \Delta(r - at'') \times \Delta(\tau - t''). \end{aligned}$$

Определим выражение для $P_{\text{ДЦ}}$ и вычислим ее при следующих значениях параметров: $\theta_T = 0,8$; $r_{\text{П}} = 200$ [тыс. руб.]; $\tau_{\text{Д}} = 60$ [час]; $a=3$ [тыс. руб./час]; $t' = 0$ [час]; $t'' = 70$ [час].

Для получения значения обратной функции целевого эффекта $F_i^{-1}(\theta_T)$ воспользуемся графиком рис. 3, построенным при $i=10$, $\omega=0,5$.

$$P_{\text{ДЦ}} = \Phi_{\langle \hat{\theta}, \hat{r}, \hat{\tau} \rangle}(\langle \theta_T, r_{\text{П}}, \tau_{\text{Д}} \rangle) = \Phi_{\langle \hat{\theta}, \hat{r}, \hat{\tau} \rangle}(\langle 0,8; 200; 60 \rangle) = \frac{\min\{\frac{200}{3}, 60\}}{70} - \frac{32}{70} = 0,40.$$

Выполним анализ влияния аспектов θ_T и $\tau_{\text{Д}}$ на эффективность отладки ПС:

$$P_{\text{ДЦ}}(\theta_T, \tau_{\text{Д}}) = \left[\frac{\tau_{\text{Д}} - t'}{t'' - t'} - \frac{F_i^{-1}(\theta_T) - t'}{t'' - t'} \right] = \frac{\tau_{\text{Д}}}{70} - \frac{F_i^{-1}(\theta_T)}{70}.$$

Вычислим коэффициенты чувствительности $h_{\tau_{\text{Д}}}^{P_{\text{ДЦ}}}$ и влияния $V_{\tau_{\text{Д}}}^{P_{\text{ДЦ}}}$:

$$h_{\tau_{\text{Д}}}^{P_{\text{ДЦ}}} = \frac{dP_{\text{ДЦ}}(\theta_T, \tau_{\text{Д}})}{d\tau_{\text{Д}}} = \frac{1}{70} = 0,014 \left[\frac{1}{\text{час}} \right]; V_{\tau_{\text{Д}}}^{P_{\text{ДЦ}}} = h_{\tau_{\text{Д}}}^{P_{\text{ДЦ}}} \tau_{\text{Д}} = 0,86.$$

Для вычисления $h_{\theta_T}^{P_{\text{ДЦ}}}$ и $V_{\theta_T}^{P_{\text{ДЦ}}}$ необходимо показатель целевого эффекта $F_i^{-1}(\theta)$ выровнять (аппроксимировать) теоретической функцией распределения и найти ее обратную функцию. Аппроксимировать функции можно стандартным способом, например, с помощью процедуры «построение линии тренда» EXCEL, а затем проинтегрировать показатель $P_{\text{ДЦ}}(\theta_T, \tau_{\text{Д}})$ по параметру θ_T .

$$h_{\theta_T}^{P_{\text{ДЦ}}} = \frac{dP_{\text{ДЦ}}(\theta_T, \tau_{\text{Д}})}{d\theta_T}; V_{\theta_T}^{P_{\text{ДЦ}}} = h_{\theta_T}^{P_{\text{ДЦ}}} \theta_T.$$

Заключение

Использование в предложенной методике оценки надежности для каждого модуля упрощает задание исходных данных, повышает точность моделирования, позволяет учитывать состоятельность (мощность) тестов, используемых для отладки программ, искать пути повышения характеристик надежности программных средств путем формирования тестов, обнаруживающих ошибки с высокой вероятностью. Аппроксимация произвольных законов распределений интервалов времени исправления ошибок распределением Кокса обобщает известные марковские модели, повышает точность моделирования процессов отладки программ и делает предложенную модель более универсальной. Методика позволяет не только рассчитать частные показатели надежности программ, но и вычислить комплексный показатель эффективности отладки программных средств, выработать практические рекомендации по эффективной организации испытаний программ.

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

References

1. Smagin V.A. *Fundamentals of Software Reliability Theory*. St. Petersburg, Military Space Academy named A.F. Mozhaitsky, 2009, 355 p. (In Russian).

- моделей нестационарных систем обслуживания. СПб.: Лань, 1999. 64 с.
3. Данилов А.И., Данилов А.А. Нестационарные модели процессов испытаний программных средств в условиях риска // II Всерос. науч.-практ. конф. «Современные проблемы создания и эксплуатации вооружения, военной и специальной техники». СПб., 2014. С. 199–202.
 4. Хомоненко А.Д., Данилов А.И., Данилов А.А. Нестационарные модели стратегий испытаний программных средств при вероятностных параметрах обнаружения ошибок // Информационно-управляющие системы. 2015. № 4. С. 50–58.
 5. Хомоненко А.Д., Данилов А.И., Данилов А.А. Динамические модели испытаний программных средств // 18 Международная конференция по мягким вычислениям и измерениям. СПб., 2015. Т. 1. С. 239–242.
 6. Данилов А.И., Данилов А.А. Динамические модели испытаний программных средств с двумя типами ошибок // Труды военно-космической академии имени А.Ф. Можайского. 2015. № 4. С. 12–21.
 7. Хомоненко А.Д., Данилов А.И., Данилов А.А., Герасименко П.В. Нестационарные модели отладки программ с распределением Кокса длительности исправления ошибок // Международная конференция по мягким вычислениям и измерениям. СПб., 2016. Т. 1. С. 163–166.
 8. Хомоненко А.Д., Данилов А.И., Данилов А.А. Динамические модели отладки программ с вероятностным обнаружением ошибок и распределением Эрланга длительности их исправления // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16. № 4. С. 655–662. doi: 10.17586/2226-1494-2016-16-4-655-662
 9. Бубнов В.П., Тырва А.В., Бурцева К.И. Нестационарная модель надежности программных средств с распределением Кокса длин интервалов времени исправления ошибок // Вестник ВЭЛНИИ. 2010. № 1(59). С. 143–152.
 10. Moranda P., Jelinski Z. Final Report on Software Reliability Study. McDonnell Douglas Astronautics Company, MADC Report Number 63921, 1972.
 11. Musa J.D., Iannino A., Okumoto K. Software Reliability: Measurement, Prediction, Application. NY: McGraw-Hill, 1987.
 12. Littlewood B. The Littlewood-Verrall model for software reliability compared with some rivals // Journal of Systems and Software. 1980. V. 1. N 3. P. 251–258. doi: 10.1016/0164-1212(79)90025-6
 13. Chidamber S.R., Kemerer C.F. A metrics suite for object oriented design // IEEE Transactions on Software Engineering. 1994. V. 20. N 6. P. 476–493. doi: 10.1109/32.295895
 14. El Emam K., Melo W., Machado J.C. The prediction of faulty classes using object-oriented design metrics // Journal of Systems and Software. 2001. V. 56. N 1. P. 63–75.
 15. Cox D.R. A use of complex probabilities in the theory of stochastic processes // Mathematical Proceedings of the Cambridge Philosophical Society. 1955. V. 51. N 2. P. 313–319. doi: 10.1017/S0305004100030231
 16. Петухов Г.Б., Якунин В.И. Методологические основы внешнего проектирования целенаправленных процессов и целеустремленных систем. М.: АСТ, 2006. 504 с.
 2. Bubnov V.P., Safonov V.I. *Dynamic Modeling of Non-Stationary Service Systems*. St. Petersburg, Lan' Publ., 1999, 64 p. (In Russian).
 3. Danilov A. I., Danilov A. A. Transient models of the software testing processes in risk conditional. *II Vseros. Nauch.-Prakt. Konf. Sovremennye Problemy Sozdaniya i Ekspluatatsii Vooruzheniya, Voennoi i Spetsial'noi Tekhniki*. St. Petersburg, 2014, pp. 199–202. (In Russian).
 4. Khomonenko A.D., Danilov A.I., Danilov A.A. Non-stationary models of software testing strategies with probabilistic parameters for fault detection. *Informatsionno-Upravlyayushchie Sistemy*, 2015, no. 4, pp. 50–58. (In Russian)
 5. Khomonenko A.D., Danilov A.I., Danilov A.A. Dynamic software testing models. *Proc. 18th Int. Conf. on Soft Computing and Measurements, SCM*. St. Petersburg, 2015, vol. 1, pp. 239–242. (In Russian)
 6. Danilov A.I., Danilov A.A. Dynamic software testing models with two types of faults. *Proceedings of the Military Space academy named after A.F. Mozhaisky*, 2015, no. 647, pp. 12–21. (In Russian).
 7. Khomonenko A.D., Danilov A.I., Gerasimenko P.V., Danilov A.A. Nonstationary software testing models with cox distribution for fault resolution duration. *Proc. 19th Int. Conf. on Soft Computing and Measurements, SCM 2016*. St. Petersburg, 2016, pp. 209–212.
 8. Khomonenko A.D., Danilov A.I., Danilov A.A. Dynamic software testing models with probabilistic parameters for fault detection and Erlang distribution for fault resolution duration. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2016, vol. 16, no. 4, pp. 655–662. doi: 10.17586/2226-1494-2016-16-4-655-662
 9. Bubnov V.P., Tyrva A.V., Burtseva K.I. Non-stationary software reliability model with Coxian distribution of errors detection time interval lengths. *Vestnik VEINII*, 2010, no. 1, pp. 143–152. (In Russian)
 10. Moranda P., Jelinski Z. *Final Report on Software Reliability Study*. McDonnell Douglas Astronautics Company, MADC Report Number 63921, 1972.
 11. Musa J.D., Iannino A., Okumoto K. *Software Reliability: Measurement, Prediction, Application*. NY, McGraw-Hill, 1987.
 12. Littlewood B. The Littlewood-Verrall model for software reliability compared with some rivals. *Journal of Systems and Software*, 1980, vol. 1, no. 3, pp. 251–258. doi: 10.1016/0164-1212(79)90025-6
 13. Chidamber S.R., Kemerer C.F. A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 1994, vol. 20, no. 6, pp. 476–493. doi: 10.1109/32.295895
 14. 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, no. 1, pp. 63–75.
 15. Cox D.R. A use of complex probabilities in the theory of stochastic processes. *Mathematical Proceedings of the Cambridge Philosophical Society*, 1955, vol. 51, no. 2, pp. 313–319. doi: 10.1017/S0305004100030231
 16. Petukhov G.B., Yakunin V.I. *Methodological Basis for External Design of Targeted Processes and Dedicated Systems*. Moscow, AST Publ., 2006, 504 p. (In Russian).

Авторы

Данилов Анатолий Исаевич – кандидат технических наук, доцент, доцент, Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург, 197198, Российская Федерация, anna_hab_2016_2@mail.ru

Данилов Андрей Анатольевич – ведущий инженер-программист, ООО «Нокиа Солюшнз энд Нетворкс», Санкт-Петербург, 192019, Российская Федерация, Andrey.Danilov.aad@mail.ru

Authors

Anatoly I. Danilov – PhD, Associate professor, Associate professor, Mozhayskiy Military Space Academy, Saint Petersburg, 197198, Russian Federation, anna_hab_2016_2@mail.ru

Andrey A. Danilov – Senior software engineer, Saint Petersburg Nokia Solutions and Networks, Saint Petersburg, 192019, Russian Federation, Andrey.Danilov.aad@mail.ru