

МОДЕЛЬ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.И. Вершина, Т.Н. Семерюк, Б.Т. Солдатов

Запорожский национальный технический университет
69063, Запорожье, Жуковского 64,
тел. (380)(612) 69 8508, E-mail: rej@mail.ru

Предлагаемая модель разработки программного обеспечения, основанная на допущении, что вероятность качественного обеспечения в неограниченно малом промежутке времени пропорциональна этому промежутку. Рассмотрены условия, при которых процесс разработки программ можно описать как марковский.

The model of process of development of the software based on the assumption is offered, that the probability of qualitative development of the program in infinitesimal an interval of time is proportional to this interval. The conditions are considered, at which the process of development of the programs can be described as markovskiy

Модели и процессы жизненного цикла программного обеспечения

Качество разработки программного обеспечения зависит от многих факторов. Поиск новых методов повышения качества программного обеспечения является актуальной задачей. Эта проблема связана с необходимостью создания математической модели. Предлагается один из возможных путей создания этой модели.

Процесс разработки программ можно разбить на два этапа, первый из которых состоит в разработке определенной последовательности операторов, а второй – в проверке полученных результатов поставленным требованиям. В случае отсутствия соответствия результатов поставленным требованиям, осуществляется доработка разработанной программы.

Результаты программирования зависят от многих субъективных и объективных факторов и, в общем случае, представляют собой случайный процесс. Если считать, что существует оценка качества программирования как вероятность того, что программа соответствует предъявляемым к ней требованиям, то процесс разработки программ можно описывать вероятностными характеристиками. Это позволяет поставить задачу построения математической модели процесса разработки программного обеспечения, как случайного процесса.

Пусть вероятность $\Delta k(t)$ разработки элемента программного обеспечения в малом промежутке времени Δt пропорциональна величине этого промежутка,

$$\Delta k(t) = k(t + \Delta t) - k(t) = [1 - k(t)]\lambda \Delta t, \quad (1)$$

где $k(t)$ - вероятность того, что элемент программного обеспечения за время разработки t соответствует предъявляемым требованиям;

λ - коэффициент пропорциональности, который отражает интенсивность разработки программного обеспечения.

Переходя к пределу $\Delta t \rightarrow 0$, получаем дифференциальное уравнение

$$dk(t) = [1 - k(t)]\lambda dt, \quad (2)$$

решение которого имеет вид

$$k(t) = 1 - e^{-\lambda t}. \quad (3)$$

Плотность распределения времени разработки элемента программного обеспечения определяется выражением

$$p_k(t) = \frac{dk(t)}{dt} = \lambda e^{-\lambda t}, \quad (4)$$

т. е. подчиняется экспоненциальному закону.

Разработка программы представляет собой совокупность α элементов программы, это приводит к композиции законов распределения. Так как показательная плотность является частным случаем гамма-распределения [1, 2], а семейство гамма-плотностей замкнуто относительно операции свертки, то плотность распределения времени на разработку программы имеет вид

$$p(t) = \frac{\lambda^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\lambda t}, \quad (5)$$

где $\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt$ - гамма-функция Эйлера.

Для произвольного времени T , затраченного на разработку программы, вероятность $K(T)$ того, что программа соответствует поставленным требованиям, определяется выражением

$$K(T) = \int_0^T p(t) dt = \frac{\lambda^\alpha}{\Gamma(\alpha)} \int_0^T t^{\alpha-1} e^{-\lambda t} dt. \quad (6)$$

Ожидаемое время на разработку равно

$$T_{yc} = \int_0^{\infty} t p(t) dt = \frac{\lambda^\alpha}{\Gamma(\alpha)} \int_0^{\infty} t^\alpha e^{-\lambda t} dt = \frac{\alpha}{\lambda}. \quad (7)$$

Как правило, время на разработку T_0 ограничено и оно пропорционально ожидаемому времени на разработку T_{yc} :

$$T_0 = \mu T_{yc} = \mu \frac{\alpha}{\lambda}, \quad (8)$$

где μ - коэффициент пропорциональности.

В этом случае имеем

$$K(T_0) = \int_0^{\frac{\mu\alpha}{\lambda}} p(t) dt = \frac{\lambda^\alpha}{\Gamma(\alpha)} \int_0^{\frac{\mu\alpha}{\lambda}} t^{\alpha-1} e^{-\lambda t} dt. \quad (9)$$

После внесения λ^α под знак интеграла, замены переменной $z = \lambda * t$ и изменения пределов интегрирования получаем

$$K(T_0) = \frac{1}{\Gamma(\alpha)} \int_0^{\mu\alpha} z^{\alpha-1} e^{-z} dz = K(\mu, \alpha). \quad (10)$$

Из этого следует, что вероятность качественной разработки в случае выделения времени на создание программы пропорционально ожидаемому времени \bar{T} не зависит от значения λ .

Если за заданное время T_0 результат разработки не удовлетворяет поставленным требованиям, то необходимо выделить дополнительное время T_δ . Это время разумно также выделять пропорционально ожидаемому времени теперь уже на доработку. Если считать, что производимые доработки в одинаковой мере затрагивают каждый элемент программного обеспечения, то количество элементов α останется прежним, а изменится значение λ . В соответствии с полученным выражением (10), вероятность качественной доработки за дополнительное время не изменится. Можно также сказать, что время, выделяемое на разработку определенного объема программного обеспечения, определяется заданным уровнем качества.

Оценим ожидаемое дополнительное время T_δ , которое необходимо выделить на проведение доработок:

$$T_\delta = \int_{T_0}^{\infty} (t - T_0) p(t) dt = \frac{\lambda^\alpha}{\Gamma(\alpha)} \int_{T_0}^{\infty} (t - T_0) t^{\alpha-1} e^{-\lambda t} dt = \frac{\lambda^\alpha}{\Gamma(\alpha)} \left[\int_{T_0}^{\infty} t^\alpha e^{-\lambda t} dt - T_0 \int_{T_0}^{\infty} t^{\alpha-1} e^{-\lambda t} dt \right]. \quad (11)$$

Используя для первого слагаемого в квадратных скобках метод интегрирования по частям, вводя функции $u = t^\alpha$ и $dv = e^{-\lambda t} dt$, для которых имеем $du = \alpha t^{\alpha-1} dt$ и $v = \int e^{-\lambda t} dt = -\frac{1}{\lambda} e^{-\lambda t}$, с учетом того,

что $\frac{\lambda^\alpha}{\Gamma(\alpha)} \int_{T_0}^{\infty} t^{\alpha-1} e^{-\lambda t} dt = 1 - K(T_0)$, получаем

$$T_{\delta} = \frac{\lambda^{\alpha}}{\Gamma(\alpha)} \left[-t^{\alpha} \frac{1}{\lambda} e^{-\lambda t} \Big|_{T_0}^{\infty} + \frac{\alpha}{\lambda} \int_{T_0}^{\infty} t^{\alpha-1} e^{-\lambda t} dt - T_0 \int_{T_0}^{\infty} t^{\alpha-1} e^{-\lambda t} dt \right] = \frac{\lambda^{\alpha}}{\Gamma(\alpha)} T_0^{\alpha} \frac{1}{\lambda} * e^{-\lambda T_0} + \left(\frac{\alpha}{\lambda} - T_0 \right) [1 - K(T_0)]. \quad (12)$$

Подставляя T_0 из формулы (8) в данное выражение, получаем

$$T_{\delta} = \frac{\lambda^{\alpha}}{\Gamma(\alpha)} \frac{1}{\lambda} \left(\mu \frac{\alpha}{\lambda} \right)^{\alpha} e^{-\lambda \mu \frac{\alpha}{\lambda}} + (1 - \mu) \frac{\alpha}{\lambda} [1 - K(\mu \frac{\alpha}{\lambda})] = \frac{1}{\lambda \Gamma(\alpha)} (\mu \alpha)^{\alpha} e^{-\mu \alpha} + (1 - \mu) \frac{\alpha}{\lambda} [1 - K(\mu, \alpha)]. \quad (13)$$

Будем считать, что дополнительное время выделяется также пропорционально ожидаемому времени:

$$T_{0\delta} = \mu T_{\delta}. \quad (14)$$

Отсюда отношение δ дополнительного времени на доработку программы ко времени, первоначально выделенному, равно

$$\delta = \frac{T_{0\delta}}{T_0} = \frac{\mu \left\{ \frac{1}{\lambda \Gamma(\alpha)} (\mu \alpha)^{\alpha} e^{-\mu \alpha} + (1 - \mu) \frac{\alpha}{\lambda} [1 - K(\mu, \alpha)] \right\}}{\mu \frac{\alpha}{\lambda}} = \frac{(\mu \alpha)^{\alpha}}{\Gamma(\alpha + 1)} e^{-\mu \alpha} + (1 - \mu) [1 - K(\mu, \alpha)]. \quad (15)$$

Из полученного выражения следует, что отношение дополнительного времени на разработку программного обеспечения к первоначально выделяемому времени будет постоянной величиной, если считать, что $\mu = const$ и $\alpha = const$.

Аналогичные рассуждения приводят к выражению для качества проверки результатов разработки программного обеспечения как вероятности $A(T_n)$ того, что в результате проверки программы в течение времени T_n получен достоверный результат о качестве разработки:

$$A(T_n) = \int_0^{T_n} p(t) dt = \frac{\lambda_n^{\alpha}}{\Gamma(\alpha)} \int_0^{T_n} t^{\alpha-1} e^{-\lambda_n t} dt. \quad (16)$$

Следует отметить, что на практике вероятности качественной проверки правильно разработанной программы и программы с ошибками дает различные значения, поэтому целесообразно для проверки программы с ошибками ввести вероятность $\bar{A}(T_n)$. Величина $A(T_n)$ близка к единице, т.е. при проверке правильной программы получается достоверный результат.

Ожидаемое время на проверку программы определяется выражением

$$T_{ож.н} = \int_0^{\infty} t p(t) dt = \frac{\lambda_n^{\alpha}}{\Gamma(\alpha)} \int_0^{\infty} t^{\alpha} e^{-\lambda_n t} dt = \frac{\alpha}{\lambda_n}, \quad (17)$$

где λ_n - интенсивность проверки программы.

Время T_n , выделяемое на проверку программы, берется по аналогии с (8) пропорционально ожидаемому времени $T_{ож.н}$:

$$T_n = \mu_n T_{ож.н}, \quad (18)$$

где μ_n - коэффициент пропорциональности при выделении времени на проверку программы.

Качество проверки разработанной программы определяется выражением

$$\bar{A}(T_n) = \frac{1}{\Gamma(\alpha)} \int_0^{\mu_n \alpha} z^{\alpha-1} e^{-z} dz. \quad (19)$$

Отношение дополнительного времени на проверку доработанной программы ко времени, первоначально выделенному, равно

$$\delta_n = \frac{(\mu_n \alpha)^\alpha}{\Gamma(\alpha + 1)} e^{-\mu_n \alpha} + (1 - \mu_n)[1 - A(\mu_n, \alpha)]. \quad (20)$$

Следует отметить, что особенности гамма-распределения нашли место, например, для описания творческой деятельности человека. Так, в работе [6] предложено теоретико-вероятностное описание научного труда, исходя из случайного распределения знаний и проблем в тезаурусе ученого. При этом делаются допущения, которые позволяют рассматривать порождение знаний и проблем как дискретный процесс Пуассона. После обобщения автор приходит к гамма-распределению, которое достаточно хорошо согласуется с экспериментальными данными науковедения о производительности научного труда.

Вышеприведенные допущения могут быть положены в основу описания процесса разработки программного обеспечения с помощью цепей Маркова.

Марковская модель разработки программ представлена следующими состояниями процесса обучения:

1. исходное состояние;
2. результат качественной разработки программного обеспечения;
3. наличие ошибок в программном обеспечении;
4. качество программы после проведения ее проверки;
5. отсутствия соответствия программы поставленным требованиям после проверки.

Если вероятности переходов между состояниями являются постоянными величинами, то такой процесс можно описать поглощающей цепью Маркова, так как существуют состояния 4 и 5, которыми завершается процесс разработки программы.

Структура матрицы переходов для поглощающей цепи Маркова имеет вид [7]:

$$P = \begin{bmatrix} Q & R \\ O & E \end{bmatrix}, \quad (21)$$

где Q - подматрица, описывающая поведение процесса до попадания в поглощающее состояние; R - подматрица переходов в поглощающие состояния; O, E - нулевая и единичные подматрицы.

Для программного обеспечения состояния 4 и 5 являются поглощающими.

Матрицы Q и R соответственно равны:

$$Q = \begin{bmatrix} 0 & K_0 & 1 - K_0 \\ 1 - A & 0 & 0 \\ \bar{A} & 0 & 0 \end{bmatrix}; \quad R = \begin{bmatrix} 0 & 0 \\ A & 0 \\ 0 & 1 - \bar{A} \end{bmatrix}, \quad (22)$$

где K_0 - вероятность качественной разработки;

A и \bar{A} - вероятности качественной проверки программы соответствующей и несоответствующей поставленным требованиям.

Использование фундаментальной матрицы $N = (E - Q)^{-1}$ позволяет получить ряд важнейших характеристик процесса разработки программ:

$$N = \frac{1}{1 - (1 - K_0)\bar{A} - K_0(1 - A)} \begin{bmatrix} 1 & K_0 & 1 - K_0 \\ 1 - A & 1 - (1 - K_0)\bar{A} & (1 - K_0)(1 - A) \\ \bar{A} & K_0\bar{A} & 1 - K_0(1 - A) \end{bmatrix}. \quad (23)$$

Элемент $n_{i,j}$ матрицы N дает ожидаемое количество моментов времени, которое проводит процесс в состоянии j до попадания в поглощающее состояние при условии, что он начался в состоянии i .

Матрица $B = N * R$ позволяет оценить вероятность попадания в соответствующее поглощающее состояние:

$$B = \frac{1}{1 - (1 - K_0)\bar{A} - K_0(1 - A)} \begin{bmatrix} K_0 A & (1 - K_0)(1 - \bar{A}) \\ [1 - (1 - K_0)\bar{A}]A & (1 - K_0)(1 - A)(1 - \bar{A}) \\ K_0 \bar{A} A & [1 - K_0(1 - A)](1 - \bar{A}) \end{bmatrix}. \quad (24)$$

Для предлагаемой модели представляют интерес элементы $n_{1,1}$ и $b_{1,1}$ матриц N и B соответственно:

$$n_{1,1} = \frac{1}{1-F}; \quad b_{1,1} = \frac{K_0 A}{1-F}, \quad (25)$$

где $F = (1 - K_0)\bar{A} + K_0(1 - A)$.

Значение элемента $b_{1,1}$ соответствует вероятности качественной разработки K на выходе процесса разработки:

$$K = b_{1,1} = \frac{K_0 A}{1-F}. \quad (26)$$

Величина $n_{1,1}$ определяет ожидаемое количество интервалов времени, необходимых для разработки программного обеспечения.

Для оценки общего ожидаемого времени на разработку рассмотрим вероятности следующих событий:

- вероятность завершения процесса разработки за выделенное время T_0 $G_0 = K_0 A + (1 - K_0)(1 - \bar{A})$, где первое слагаемое определяет вероятность того, что программа разработана и ее проверка дала положительный результат, а второе слагаемое – это вероятность того, что программа содержит ошибки, а проверка не выявила это;

- вероятность выделения первого дополнительного времени на разработку программы как события, противоположного завершению процесса за выделенное время $F_1 = F = 1 - G_0$;

- вероятность завершения процесса разработки после выделения первого дополнительного времени $G_1 = F G_0$;

- вероятность выделения второго дополнительного времени на разработку программы $F_2 = F^2$;

- вероятность завершения процесса разработки после выделения второго дополнительного времени $G_2 = F^2 G_0$;

Таким образом, вероятность выделения j -го дополнительного времени на разработку определяется выражением

$$F_j = F^j, \quad (27)$$

а вероятность завершения процесса разработки G_j после выделения j -го дополнительного времени получается умножением (27) на G_0 :

$$G_j = G_0 F^j. \quad (28)$$

Для ограниченного количества n моментов времени, выделяемых на разработку программы, вероятность того, что процесс будет завершен, определяется суммой

$$G_{(n)} = G_0 + G_0 F + G_0 F^2 + \dots = G_0 \prod_{j=0}^n F^j = G_0 \frac{1 - F^{n+1}}{1 - F}. \quad (29)$$

Предел этой вероятности при $n \rightarrow \infty$ равен единице:

$$G = \lim_{n \rightarrow \infty} [G_0 \frac{1 - F^{n+1}}{1 - F}] = \frac{G_0}{1 - F} = 1. \quad (30)$$

Проводя аналогичные рассуждения, получим выражения для вероятности того, что и результат ее проверки разработанной программы дал положительный результат:

$$K_{(n)} = K_0 A \frac{1 - F^{n+1}}{1 - F}, \quad (31)$$

а при $n \rightarrow \infty$

$$K = \frac{K_0 A}{1 - F}. \quad (32)$$

Последнее выражение совпадает с (26).

Оценим время, затрачиваемое на разработку программы.

Затраченное на создание программы время $T_{(n)}$ определяется суммой основного времени на первоначальную разработку и n раз выделяемого дополнительного времени:

$$T_{(n)} = T_0 + T_1 + \dots + T_i + \dots + T_n, \quad (33)$$

где T_i - выделяемое i -ое дополнительное время.

В соответствии с выражением (15) отношение $\frac{T_{i+1}}{T_i} = \delta$, откуда получаем

$$T_i = \delta^i T_0 \quad (34)$$

и

$$T_{(n)} = T_0 \sum_{i=0}^n \delta^i = T_0 \frac{1 - \delta^{n+1}}{1 - \delta}. \quad (35)$$

Ожидаемое время $\bar{T}_{(n)}$ при количестве выделяемых дополнительных моментов времени, не превышающих n , с учетом выражений (28) и (35), равно

$$\bar{T}_{(n)} = \sum_{i=0}^n T_{(i)} G_i = \frac{T_0 G_0}{1 - \delta} \sum_{i=0}^n F^i (1 - \delta^{i+1}) = \frac{T_0 G_0}{1 - \delta} \left[\frac{1 - F^{n+1}}{1 - F} - \delta \frac{1 - (\delta F)^{n+1}}{1 - \delta F} \right]. \quad (36)$$

При неограниченном количестве выделяемых дополнительных интервалов времени получим значение ожидаемого времени \bar{T} на создание программы:

$$\bar{T} = \lim_{n \rightarrow \infty} \left\{ \frac{T_0 G_0}{1 - \delta} \left[\frac{1 - F^{n+1}}{1 - F} - \delta \frac{1 - (\delta F)^{n+1}}{1 - \delta F} \right] \right\} = \frac{T_0}{1 - \delta F}. \quad (37)$$

Таким образом, гипотеза о том, что вероятность качественной разработки элемента программного обеспечения в малом промежутке времени пропорциональна величине этого промежутка, приводит к экспоненциальному закону распределения времени на разработку этого элемента. Композиция экспоненциальных законов при постоянстве интенсивности разработки приводит к гамма-распределению времени на разработку программного обеспечения в целом. К такому результату приводит процесс проверки программного обеспечения. Выделение времени на разработку и проверку пропорционально ожидаемому времени приводит к постоянству качества и позволяет описывать разработку программ как марковский процесс. Подобный подход использовался при создании модели процесса обучения [8].

1. Феллер В. Введение в теорию вероятностей и ее приложения. В 2-х томах, Пер. с англ. - М.: Мир, 1984. - 1266 с.
2. Венцель Е.С. Теория вероятностей. - М.: Наука, 1964. - 576 с.
3. Справочник по специальным функциям // Под ред. М. Абрамовица и И. Стиган. - М.: Наука, 1979. - 832 с.
4. Янке Е., Эмде Ф., Лёви Ф. Специальные функции. - М.: Наука, 1977. - 344 с.
5. Гмурман В.Е. Руководство к решению задач о теории вероятностей и математической статистике. - М.: Высшая школа, 1975. - 336 с.
6. Muller Fritz. Ein Versuch zur wahrscheinlichkeitstheoretischen Erklärung der wissenschaftlichen Produktivität. - Elektron. Informationsverarb. und Kybern., 1974, 10, №1, P. 53-64.
7. Кемени Дж., Снелл Дж. Конечные цепи Маркова. - М.: Наука, 1970. - 272 с.
8. Вершина А.И., Солдатов Б.Т. Моделирование процесса обучения // «Радиоэлектроника, информатика, управління». - Запоріжжя: ЗДТУ. - 2003. - №1. - С. 65 - 72.