УДК 004.41:004.056

А.В. Коваленко

*Кировоградский национальный технический университет, Кировоград*

# МЕТОД УПРАВЛЕНИЯ РИСКАМИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*В данной работе задача управления рисками разработки программного обеспечения при условии ограниченности средств (финансовых, технических и др.) выделенных на устранение ошибок безопасности, рассматривается в виде полумарковской модели принятия решений для управляемого процесса в непрерывном времени с критерием минимума расходов на устранение аномалий. Разработан метод управления рисками разработки программного обеспечения, отличающийся от известных использованием псевдобулевых методов бивалентного программирования с нелинейной целевой функцией и линейными ограничениями для определения оптимальной стратегии устранения эксплуатационных ошибок. В качества примера рассмотрены ситуации возникновения ошибок безопасности программного обеспечения, и определена оптимальная стратегия управления для устранения указанной аномальной ситуации.*

***Ключевые слова:*** *управление рисками, разработка программного обеспечения, псевдобулевые методы бивалентного программирования.*

## Постановка проблемы в общем виде и анализ литературы

Проведенные исследования, а также анализ литературы [1 – 9] показали, что управление риском разработки программного обеспечения (ПО) состоит в заблаговременном выявлении связанных с риском финансовых, технических, психологических, и др. опасностей, и принятии мер по снижению риска путем целенаправленного изменения этих факторов с учетом эффективности принимаемых мер. Управление риском разработки ПО включает систему мероприятий, осуществляемых как до проявления негативного события, так и после его реализации. Однако, как показали исследования, превентивный анализ и учет большинства возможных эксплуатационных ошибок позволит снизить финансовые и др. затраты в жизненном цикле разработки ПО.

Ряд авторов [1-11, 16-20] под термином "управление риском" понимают разработку и обоснование оптимальных программ деятельности, призванных эффективно реализовать решения в области обеспечения безопасности. При этом главным элементом такой деятельности является процесс оптимального распределения ограниченных ресурсов с учетом характерных эксплуатационных, экономических и социальных факторов. Рассматриваемую задачу управления рисками разработки ПО при определенных ограничениях на мероприятия по тестированию качества и безопасности, сформулируем в виде полумарковской модели принятия решений для управляемого марковского процесса в непрерывном времени и дисконтированными доходами (с коэффициентом $0 < \alpha < 1$ в нормальных условиях процесса создания ПО) или расходами (в условиях с отклонениями от плана, связанными с пренебрежением невыявления уязвимостей (ошибок) безопасности). При этом данный вид эксплуатационных рисков отождествляются с последовательно соединенными независимыми элементами, восстанавливаемыми за конечное время.

Оптимальную нерандомизированную стационарную стратегию управления определим с помощью псевдобулевых методов бивалентного программирования, находя все решения системы ограничений. Эти решения определяются на основе алгоритма пересечения решений отдельных неравенств-ограничений, предложенного в работах [12-15] для нахождения базисных решений системы линейных неравенств с булевыми переменными. В таких условиях сформулируем основную задачу. Пусть каждому состоянию $i \in S,\ S = \{0,1,...,N\}$ рассматриваемой системы управления рисками разработки ПО поставлено в соответствие конечное множество $R_i$ решений, элементы которого обозначим как $r = 1,2,...,r_i$. Если система находится в состоянии $i \in S$ и принимается решение $r = R_i$, то ее дальнейшее поведение определяется вероятностным законом:

$$Y_{ij}^{r}(t) = P_{ij}^{(r)} F_{ij}^{(r)}(t), \quad j \in S, \qquad (1)$$

где вероятность перехода системы из состояния в состояние $i$ – $P_{ij}^{(r)}$, а $F_{ij}^{(r)}(t)$ – функция распределения времени пребывания системы в состоянии $i$ при принятии решения $r$ и при условии, что следующий переход произойдет в состояние $j$.

При этом сделаем допущение, что выполнены следующие условия:

Состояние $i = 0$ соответствует нормальному процессу разработки ПО, а $i \neq 0$ – ситуация ошибки безопасности.

Функции $F_{0j}^{(r)}(t)$ и $F_{j0}^{(r)}(t)$, $j \in \tilde{S} = S/\{0\}$, $r \in R_j$, вместе со своими первыми производными непрерывны при $t > 0$, за исключением конечного числа точек, и возрастают в соответствии с экспоненциальным законом распределения.

За единицу времени пребывания в состоянии $i$ в случае принятия решения $r$ тратится в среднем $k_i^{(r)}$ средств (при $i \neq 0$ число $k_i^{(r)}$ отрицательно и равно издержкам системы за единицу времени пребывания в состоянии $i$ при условии выхода из этого состояния с учетом решения $r$).

Величины $\left|k_i^{(r)}\right|$ ограничены при всех $i \in S$, $r \in R_i$, а вероятности $P_i^{(r)}$ удовлетворяют соотношениям:

$$\sum_{j \in S} P_i^{(r)} = 1, \quad P_{ij}^{(r)} \geq 0, \quad i, j \in S, \quad r \in R_i.$$

Таким образом, в каждом состоянии $i \in S$ существует $r_i$ решений из конечного множества $R_i$. Выбор некоторого решения $r$ из этого множества $R_i$ в состоянии $i \in S$ означает задание величин $Y_{ij}^{r}(t)$, $P_{ij}^{(r)}$, $F_{ij}^{(r)}(t)$, $k_i^{(r)}$, $j \in S$.

При $i = 0$, $R_0 = \{0\}$, $P_{0j}^{(r)} \neq 0$, $j \in S$ является вероятностью перехода в состояние $j$. Вероятность $P_{0j}^{(r)} \neq 0$, $j \in S$ вычисляется на практике как доля состояний с ошибками безопасности типа $j$ в общей совокупности уязвимостей безопасности различных типов на основе данных предысторий процесса разработки ПО. Тогда $F_{0j}^{(r)}(t)$ – функция распределения времени тестовой эксплуатации ПО между выявленными ошибками безопасности типа $j$.

При $i = \overline{1, N}$ $\forall r \in R_i$, $P_{i0}^{(r)} = 1$, $P_{ij}^{(r)} = 0$, $j \neq 0$, функция $F_{i0}^{(r)}(t)$ – функция распределения времени устранения уязвимостей безопасности с использованием решения $r$ при ошибке типа $j$.

При условии непрерывности во времени исследуемого процесса будем пользоваться переоценкой экспоненциального вида с нормой $\alpha$, то есть если в некоторый момент времени затраты составляют какую-то единичную величину, то через время $t$ эти затраты уже будут $e^{-\alpha t}$ единичных величин. Тогда если $k_i$ – расход за единицу времени, то суммарный расход за время $t$ имеет вид:

$$\int_0^t k_i e^{-\alpha\tau} d\tau = \frac{k_i}{\alpha}\left(1 - e^{-\alpha\tau}\right). \qquad (2)$$

Обозначим $i_n$ состояние системы после $n$-го перехода, $u_n$ – принятое решение, а $\tau_n$ – время пребывания в этом состоянии ($n = 0,1,2,...$), $i_0$ – начальное состояние. Допустимую стратегию $\beta$ для системы управления разработкой ПО определим как последовательность $\{\beta_0, \beta_1, \beta_2, ...\}$, где $\beta_n(\bullet / z_n)$ – вероятностная мера, сосредоточенная на функции ограничения $U(S)$ на принятые решения (управления), определяемые системой неравенств:

$$\sum_{j \in S} c_{rj} x_{rj} \leq b_r, \quad r \in R = \bigcup_{j \in S} R_j, \qquad (3)$$

и зависящая от истории управляемой системы к моменту $z_n = (i_0, u_0, \tau_0, ..., i_{n-1}, u_{n-1}, \tau_{n-1}, i_n)$ – $n$-го перехода. Мера $\beta_n(\bullet / z_n)$ задает рандомизированное правило выбора решения $u_n$ на основе информации $z_n$. Такую стратегию $\beta$ можно назвать рандомизированной.

Стратегия $\beta$ является марковской, если $\beta_n(\bullet / z_n) = \beta_n(\bullet / i_n)$, где $n = 0,1,2,...$ Марковская стратегия называется стационарной, если $\beta_n(\bullet / i_n) = \beta_n(\bullet / i_n)$. Плотность меры такой стратегии при $i_n = i, u_n = r, (r \in R_i)$ обозначим $d_i^{(r)}$. Если стратегия $\beta$ – марковская стационарная, то управляемый процесс является полумарковским.

Анализ литературы показал, что наиболее популярная информация о полумарковских процессах и управляемых полумарковских моделях с дополнительными расходами и дивидендами изложена в работах [12-15].

Обозначим через $g_i(t, \alpha, \beta)$ суммарный расход системы, управляемой в соответствии со стратегией $\beta$, с нормой переоценки $\alpha$, за время $t$ жизненного цикла разработки ПО. Обязательным условием является то, что процесс начинается в момент $t = 0$ из состояния $i$. Через $\upsilon_i(t, \alpha, \beta) = g_i(t, \alpha, \beta)/t$ обозначим суммарный средний расход системы за время $t$ при тех же условиях.

Пусть $c_{rj}$ – затраты, связанные с реализацией мероприятия $r$ в случае события нарушения безопасности ПО $j$ и $x_{rj}$ – булевая переменная: $x_{rj} = 1$, если $r$ применяется при событии $j$, $x_{rj} = 0$ в противном случае.

Предположим, что общий объем средств, отпущенных для устранения недостатков безопасности ПО (мероприятия типа $r$) ограничен константой $b_r$, т.е. выполняется неравенство (3).

Если затраты $c_{rj}$ позволяют выполнить каждое из ограничений (3), то реализованная на основании

(3) система определяет в пространстве $\Re^d$, $d = \dim R$, некоторое конечное множество дискретных точек. Тогда в соответствии с работами [12-15] существует нерандомизированная стационарная стратегия $\beta^*$, называемая $\beta$ – оптимальной, которая минимизирует суммарный средний расход $\upsilon(\alpha,\beta)$ при произвольной стратегии $\beta$ и норме переоценки $\alpha(\alpha > 0)$. При этом $\upsilon(\alpha,\beta)$ – $(N+1)\times 1$-мерный вектор $(\upsilon_0(\alpha,\beta), \upsilon_1(\alpha,\beta),\ldots,\upsilon_N(\alpha,\beta))$, где

$$\upsilon_i(\alpha,\beta) = \lim_{t\to\infty} \upsilon_i(t,\alpha,\beta), i \in S. \qquad (4)$$

Необходимо найти $\alpha$ – оптимальную нерандомизированную марковскую стационарную стратегию $\beta^*$, которая минимизирует суммарный средний расход $\upsilon(\alpha,\beta)$ при произвольном начальном распределении процесса:

$$y = (y_0, y_1, \ldots, y_N), \qquad (5)$$

$$\sum_{i\in S} y = 1,\ y \ge 0,\ i \in S. \qquad (6)$$

Не уменьшая общности, в качестве начального распределения возьмем вектор $y = (1, 0, \ldots, 0)$, т.е. начальное состояние системы. На основе полумарковской модели принятия решений данную задачу приведем к эквивалентной задаче бивалентного программирования с использованием псевдобулевых методов.

## 1. Оптимизационная стратегия полумарковской модели принятия решений

Вероятности переходов рассматриваемого, для системы разработки ПО, полумарковского процесса принятия решений в моменты скачков из состояния $i$ в состояние $j$ при принятии решения $r \in R_i$ определяется стохастической $(N+1)\times(N+1)$ матрицей $P^{(r)} = \left\{p_{ij}^{(r)}\right\}$, которая задает вложенную цепь Маркова. Элементы $p_{ij}^{(r)}$ $\forall i, j \in S$ и $r \in R_i$ позволяют определять по формуле (1) совместную вероятность $Q_{ij}^{(r)}(t)$ того, что длительность пребывания в состоянии $i$ не превосходит время $t$ из состояния $i$ при $r \in R_i$ процесс переходит в состояние $j$ с вероятностью $p_{ij}^{(r)}$. Функции $Q_{ij}^{(r)}(t)$ в (1) удовлетворяют условиям:

$$Q_{ij}^{(r)}(0) = 0,\ i, j \in S, r \in R_i, \qquad (7)$$

$$\sum_{j\in S} Q_{ij}^{(r)}(\infty) = \sum_{j\in S} p_{ij}^{(r)} = 1,\ i \in S, r \in R_i. \qquad (8)$$

С помощью матрицы $Q_{ij}^{(r)}(t) = \left\{Q_{ij}^{(r)}(t)\right\}$ переходных распределений, определим функцию:

$$H_i^{(r)}(t) = \sum_{j\in S} Q_{ij}^{(r)}(t),\quad i, \in S, r \in R_i, \qquad (9)$$

являющуюся функцией распределения времени пребывания процесса в состоянии $i$ при принятии решения $r \in R_i$. Случайный процесс $(Z_t), t \ge 0$ со значениями $Z_t = i$, если в момент $t$ система находится в состоянии $i$, является полумарковским, и задается величинами $N, y, Q_{ij}^{(r)}(t), i, j \in S, r \in R_i$.

Полумарковский процесс называется регулярным, если за конечный промежуток времени он с вероятностью $p_p = 1$ перейдет в любое состояние не более конечного числа раз. Таким образом, регулярный полумарковский процесс за конечный промежуток времени всегда совершает лишь конечное число переходов. Далее в разделе будем рассматривать только регулярные полумарковские процессы.

В случае одноэлементных множеств решений $R_i$ в результате стандартных для теории восстановления [12-15] рассуждений получаем следующее уравнение восстановления

$$\upsilon_i(t) = (1 - H_i(t))\frac{k_i}{\alpha}\left(1 - e^{-\alpha t}\right) +$$

$$+\sum_{j\in S}\int_0^t \left(\frac{k_i}{\alpha}\left(1 - e^{-\alpha t}\right) + e^{-\alpha t}\upsilon_j(t-\tau)\right) dQ_{ij}(\tau),\ i \in S,$$

где $\upsilon_i(t)$ – краткая запись суммарного среднего расхода $\upsilon_i(t,\alpha,\beta)$ за время $t$.

В случае конечных множеств $R_i$ уравнение восстановления с учетом вероятностей $d_i^{(r)}$ принятия решений $r$ в состоянии $i$ запишем в виде:

$$\upsilon_i(t) = \sum_{r\in R_i} d_i^r\left(1 - H_i^{(r)}(t)\right)\frac{k_i^{(r)}}{\alpha}\left(1 - e^{-\alpha t}\right) +$$

$$+\sum_{j\in S}\sum_{r\in R_i}\int_0^t d_i^r\left(\frac{k_i^{(r)}}{\alpha}\left(1 - e^{-\alpha t}\right) + e^{-\alpha t}\upsilon_j(t-\tau)\right) dQ_{ij}^{(r)}(\tau),\ i \in S, \qquad (10)$$

где $k_i^{(r)}$ – расход системы за единицу времени пребывания в состоянии $i$ при решении $r \in R_i$; $\upsilon_j(t)$ – суммарный средний расход с учетом переоценки (2), при условии, что процесс начинается в момент $t = 0$ из состояния $j$.

Величины $\upsilon_i(\alpha,\beta)$ из (4) можно записать в виде $\upsilon_i(\alpha)$, и для этого уравнения воспользоваться основными положениями уравнения (интеграла) Лапласа-Стилтьеса. В соответствии с [12-15] для лю-

бой функции $F(t)$, производная $F'(t)$ которой является функцией-оригиналом, удовлетворяющей неравенству $F'(t) < Ce^{\alpha t}$ для всех $t < 0$, при всех комплексных $s$, когда $\mathrm{Re}\, s > \alpha$ существует функция:

$$F^*(s) = L_s^* \langle F(t) \rangle = \int_0^\infty e^{-st} dF(t), \qquad (11)$$

то есть функция $e^{-st}$ при $\mathrm{Re}\, s > \alpha$ интегрируема по функции $F(t)$. Функцию $F^*(s)$ называют преобразование Лапласа-Стилтьеса функции $F(t)$.

Из выражений 8 и 9 следует, что $H_i^{(r)}(\infty) = 1$, $i \in S, r \in R_i$, поэтому первая сумма в выражении 10 при $t \to \infty$ обращается в нуль. Интегрируя по частям (11) для $L_s^* \langle F(t) \rangle$, получаем:

$$sL_s^* \langle F(t) \rangle = L_s^* \langle F(t) \rangle - F(0), \qquad (12)$$

где $$F(s) = L_s \langle F(t) \rangle = \int_0^\infty e^{-st} F(t) dt$$ –

преобразование Лапласа функции $F(t)$.

Из (12) при $s \neq 0$ находим

$$L_s \langle F(t) \rangle = \frac{1}{s} \left( L_s^* \langle F(t) \rangle - F(0) \right). \qquad (13)$$

Интегрируем по частям с учетом (9) находим

$$\sum_j \int_0^t \left(1 - e^{-\alpha t}\right) dQ_{ij}^{(r)}(\tau) = \left(1 - e^{-\alpha t}\right) \sum_j dQ_{ij}^{(r)}(\tau) \Big|_0^t - \sum_j \alpha \int_0^t e^{-\alpha t} H_i(\tau) d\tau. \qquad (14)$$

Проводя преобразования, переходя в выражении 14 к пределу $t \to \infty$ и применяя формулу (13) для $s = \alpha, (\alpha > 0)$, с учетом (7) и (8) получим:

$$\sum_j \int_0^t \left(1 - e^{-\alpha t}\right) dQ_{ij}^{(r)}(\tau) = (1 - \alpha) L_{s=\alpha} \left\langle H_i^{(r)}(\tau) \right\rangle = 1 - \alpha \frac{1}{\alpha} L_{s=\alpha}^* \left\langle H_i^{(r)}(\tau) \right\rangle = 1 - h_i^{(r)}(\alpha),$$

где $h_i^{(r)}(\alpha) = L_{s=\alpha}^* \left\langle H_i^{(r)}(t) \right\rangle$.

Применяя к функции:

$$\Phi_i^{(r)}(t) = \int_0^r e^{-\alpha t} \nu_j(t - \tau) dQ_{ij}^{(r)}(t)$$

теорему о предельном переходе в интеграле по параметру, от которого зависят пределы интегрирования и подынтегральная функция [12-15], при $t \to \infty$:

$$\Phi_i^{(r)}(\infty) = \int_0^\infty e^{-\alpha t} \nu_j(\alpha) dQ_{ij}^{(r)}(\tau) = \nu_j(\alpha) q_{ij}^{(r)}(\alpha), \qquad (16)$$

где $q_{ij}^{(r)}(\alpha) = L_{s=\alpha}^* \left\langle Q_i^{(r)}(\alpha) \right\rangle$.

Переходя в выражении 10 к пределу при $t \to \infty$, с учетом 15 и 16 получаем следующее аналитическое выражение:

$$\upsilon_i(t) = \sum_{r \in R_i} d_i^{(r)} \left( \zeta_i^{(r)}(\alpha) \right) + \sum_{j \in S} q_{ij}^{(r)}(\alpha) \nu_j(\alpha), \qquad (17)$$

где $$\zeta_i^{(r)}(\alpha) = \frac{k_i^{(r)}}{\alpha} \left(1 - h_i^{(r)}(\alpha)\right). \qquad (18)$$

Пусть $$\zeta_i(\alpha) = \sum_{r \in R_i} d_i^r \left( \rho_i^{(r)}(\alpha) \right)$$ и

$\Im(\alpha) = \left(\zeta_0(\alpha), \ldots, \zeta_N(\alpha)\right)^T$, $\wp(\alpha) = \left(\nu_0(\alpha), \ldots, \nu_N(\alpha)\right)^T$, (T – символ транспонирования матрицы). Тогда:

$$\wp(\alpha) = \Im_0(\alpha) + q(\alpha) \wp(\alpha) \qquad (19)$$

где $q(\alpha) = \{q_{ij}(\alpha)\}$, $q_{ij}(\alpha) = \sum_{r \in R_i} d_i^{(r)} \left( q_{ij}^{(r)}(\alpha) \right)$.

Из выражения (19) найдем:

$$\wp(\alpha) = \{I - q(\alpha)\}^{-1} \Im_0(\alpha) \qquad (20)$$

Данное выражение справедливо, так как при $\alpha > 0$ матрица $\{I - q(\alpha)\}$ – невырожденная, $I$ – единичная матрица размера $(N \times 1) \times (N \times 1)$.

Умножив обе части равенства 19 слева на вектор $y$, получим следующее:

$$y\upsilon(\alpha) = \sum_{i \in S} \sum_{j \in \tilde{S}} \sum_{r \in R_i} y_i \mu_{ij}(\alpha) \zeta_j^{(r)}(\alpha) d_i^{(r)},$$

$$\{I - q(\alpha)\}^{-1} = \{\mu_{ij}(\alpha)\}. \qquad (21)$$

Величины $\mu_{ij}(\alpha)$ зависят от $d_i^{(r)}$, $r \in R_i$, $i \in S$, так как элементы матрицы $\{I - q(\alpha)\}$ можно выразить через $d_i^{(r)}$, $r \in R_i$, $i \in S$.

Пусть $\left\{d_i^{(r)}\right\}$ $(r \in R_i)$ – нерандомизированная марковская стационарная стратегия системы разработки ПО в состоянии $j$.

$$d_j^{(r)} \in \{0,1\}, \ \sum_{j \in S} d_j^{(r)} = 1 \quad \text{и} \quad x_{00} = 1, \ x_{rj} = d_j^{(r)},$$

$r \in R_i$, $j \in \tilde{S}$. Минимизация расходов (выражение (21)) приводит к следующей задаче оптимизации для булевых переменных $X = \{x_{rj}\}$, $r \in R_i$, $j \in \tilde{S}$:

$$f(\alpha, X) = \sum_{i \in S} \sum_{j \in \tilde{S}} \sum_{r \in R_i} y_i \mu_{ij}(\alpha, X) \zeta_j^{(r)} x_{rj} \to \min, \qquad (22)$$

$$\sum_{r \in R_i} x_{rj} = 1, \ j \in \tilde{S}, \qquad (23)$$

$$\sum_{j \in \tilde{S}} c_{rj} x_{rj} \le b_r, \ r \in R_i, \ j \in \tilde{S}, \qquad (24)$$

$$x_{rj} \in \{0,1\}, \ j \in \tilde{S}, \ r \in R_i. \qquad (25)$$

## 2. Построение оптимальной нерандомизированной марковской стационарной стратегии

Обозначим систему (24), (25) как система С. Она является системой псевдобулевых неравенств.

Подключив далее дополнительное условие (23) можно обозначить систему как $\tilde{C}$, а через $X_r^{(k)} = \left\{ x_{r1}^{(k)}, \ldots, x_{rN}^{(k)} \right\}$, $k = 1, \ldots k_r$ – допустимые решения $r$-го неравенства системы С.

Для построения решений системы $\tilde{C}$ при известных допустимых решениях каждого неравенства (24) применим следующий подход. Решения системы $\tilde{C}$ находятся как $Z = \left\{ s_j \right\}$, $j = 1, \ldots N$, где $s_j$ – множество номеров $r$, для которых допустимо равенство $x_{rj} = 1$. Решения находятся за m шагов, где $m$ – число ограничений 24. В исходном состоянии каждое из множеств $s_j^{(0)}$ вектора $Z^{(o)}$ включает все возможные значения $r \in R_i$. На $r$-м шаге происходит пересечение вектора $Z^{(r-1)}$ с одним из решений $r$-го неравенства. Допуская, что $r$-му неравенству соответствует $r = r_l$, а также, что $\alpha_j$ является $j$-м элементом допустимого решения данного неравенства, $\alpha_j \in \{0, 1, \phi\}$, где $\phi$ – неопределенный булев параметр, называемый в дальнейшем почерком, можно сформулировать такие правила для $r$-го шага алгоритма построения решений системы $\tilde{C}$.

1. Если $\alpha_j$ не фиксировано, то $s_j^{(r)} = s_j^{(r-1)}$.

2. Если $\alpha_j = 1$, то при $r_l \in s^{(r-1)}$ допускаем $s^{(r)} = \{r_l\}$, а при $r_l \overline{\in} s^{(r-1)}$ допускаем $s^{(r)}$ равно пустому множеству.

3. Если $\alpha_j = 0$ то $s^{(r)} = s^{(r-1)} \Big/ \{r_l\}$.

При этом пересечение семейств решений осуществляется с учетом дополнительных ограничений (23).

На m-м шаге алгоритма получается вектор $Z^{(m)} = \left\{ \alpha_1^{(m)}, \ldots, \alpha_N^{(m)} \right\}$, каждая компонента $\alpha_j^{(m)}$, которого является одноэлементным множеством $\{r\}$, $r \in R$, $R = \{1, \ldots, m\}$ и следовательно, $Z^{(m)}$ есть решение системы $\tilde{C}$. Исключение составляют случаи, когда $\tilde{C}$ представляет собой некоторый набор чисел $r$ из множества $R$. В этом случае с помощью сочетания элементов многозначных компонент из вектора $Z^{(m)}$ можно получить несколько решений системы $\tilde{C}$. Затем в результате находим совокупность всех решений системы $\tilde{C}$, из которых выбираем оптимальное решение, доставляющее минимум целевой функции $f(\alpha, X)$. Численная реализация изложенного метода управления рисками безопасности индустрии программного обеспечения для полумарковской модели принятия решений при аномальных ситуациях безопасности представлена в следующем примере.

## 3. Рекомендации по использованию метода управления рисками разработки программного обеспечения

Рассмотрим две возможные ситуации ошибок безопасности ПО:

– ситуация 1 – наличие функциональных ошибок ПО (функциональные возможности ПО, не описанные или не соответствующие описанным в документации, при использовании которых возможно нарушение конфиденциальности, доступности или целостности обрабатываемой информации);

– ситуация 2 – наличие нефункциональных ошибок ПО (избыточные ФО и ИО, переполнение буфера, утечки памяти, ошибки типов данных, ошибки указателей и др.).

Альтернативные мероприятия (упрощенно) в случае первой аномальной ситуации следующие:

- отзыв ПО (для устранения ошибок);
- декомпозиция программы и оценка метрик сложности;
- сигнатурный анализ.

В случае наличия нефункциональных ошибок ПО альтернативные мероприятия следующие:

- отзыв ПО (для устранения ошибок);
- анализ указателей, зависимостей по данным и интервальный анализ.

Предположим, что в обоих ситуациях первое мероприятие (отзыв ПО) осуществляется за одно и то же время $T = 1$ у.е. Поэтому, учитывая малую стоимость соответствующих функций тестирования по сравнению с убытками при простое ПО, можно считать первые мероприятия в обоих случаях одинаковыми по затратам. Представим мероприятия в случае первой аномальной ситуации в виде $R = \{r_1; r_2; r_3\}$, а мероприятия для устранения второй – $R = \{r_1; r_4\}$. Тогда число элементов системы $N = 2$, а общее число различных мероприятий для их устранения $m = 4$.

Также представим описанные аномальные ситуации j по отношению с отдельными модулями и блоками ПО как последовательные соединения (по надежности). Ошибки (отказы) и их устранение не влияют на надежность других модулей и элементов ПО. Время устранения ошибки не зависит от того, выявляются ли другие ошибки в оставшихся модулях. Другими словами, каждая ошибка выявляется и устраняется независимо друг от друга. В этом случае поток ошибок и, устраняющих их, мероприятий есть сумма N независимых процессов устранения с конечным временем восстановления.

Обозначим через $F_j(t)$ функцию распределения времени эксплуатации системы между двумя последовательными аномальными ситуациями типа $j$, а через $G_j^{(k)}(t)$ – функцию распределения времени устранения ошибки после аномальной ситуации типа $j$ при принятии решения $r$. Будем полагать, что $F_j(t)$ и $G_j^{(k)}(t)$ подчиняются экспоненциальным законам с функциями интенсивности соответственно $\lambda_j$ и $\mu_j^{(k)}$:

$$F_j(t)=1-e^{-\lambda_j t},\ G_j^{(k)}(t)=1-e^{-\mu_j^{(r)}t}, \quad (26)$$

где $\lambda_j=1/T_{j1}$; $\mu_j^{(r)}=1/T_{j2}$; $T_{j1}$ – среднее время эксплуатационного этапа жизненного цикла разработки ПО между двумя аномальными ситуациями типа $j$; $T_{j2}$ – среднее время восстановительного этапа жизненного цикла разработки ПО после выявления ошибки типа $j$ при решении $r$. Пусть $d_i^{(k)}$ – нерандомизированная стационарная стратегия системы в состоянии $i$ $(i\in S)$ при решении $k$ (т.е. вероятность принятия решения $r$ в состоянии $i$),

$$d_i^{(r)}\in\{0,1\},\ \sum_{k\in K}d_i^{(r)}\in 1,\ i\in S. \quad (27)$$

Тогда закон распределения эксплуатации и закон распределения устранения ошибок в целом запишем в виде

$$F_j(t)=1-e^{-\lambda t},\ G_j^{(r)}(t)=1-e^{-\mu t},$$

где $\lambda=\sum_{j=1}^{N}\lambda_j$; $\mu=\sum_{j=1}^{N}\sum d_j^{(r)}\mu_j^{(r)}$.

Пусть

$$T_{11}=8\text{ у.е},\quad T_{21}=8\text{ у.е},\quad T_{12}^{(1)}=T_{22}^{(1)}=1\text{ у.е},$$
$$T_{12}^{(2)}=2\text{ у.е},\quad T_{12}^{(3)}=1\text{ у.е},\quad T_{22}^{(4)}=0,5\text{ у.е}. \quad (28)$$

Предположим, что $i=0$ фиксирует нормальное состояние, $i=1$ аномальное состояние (ситуация 1), $i=2$ аномальное состояние (ситуация 2); $S=\{0,1,2\}$, $\tilde{S}=\{1,2\}$. В соответствии с условием (28) и учитывая выражения (25)-(27) получим:

$$F_0(t)=F(t)=1-e^{-0,25t},\ F_j(t)=1-e^{-0,125t},\ j=1,2;$$

$$G_1^{(1)}(t)=1-e^{-t},\ G_1^{(2)}(t)=1-e^{-0,5t},$$

$$G_1^{(3)}(t)=1-e^{-t},\ G_2^{(1)}(t)=1-e^{-t},\ G_2^{(4)}(t)=1-e^{-2t}.$$

Пусть издержки от невыполнения своих функций программной системы в течении времени $T=1$ ч составят 70000 у.е. Обозначим через $c_{k,j}$ затраты на мероприятие $k$ в случае аномальной ситуации $j$. Тогда, пренебрегая затратами на тестирование в мероприятии $r=1$, получаем $c_{1,1}=c_{1,2}=c_1=70000$ у.е. Пусть далее $c_{2,1}=300$, $c_{3,1}=400$, $c_{4,2}=600$ у.е. Будем считать, что в состоянии $i=0$ принято единственное решение $(r=0)$ – продолжить нормальное функционирование, и в этом состоянии задано такое распределение вероятностей:

$$p_{00}^{(0)}=0,7,\ p_{01}^{(0)}=0,1,\ p_{02}^{(0)}=0,2. \quad (29)$$

Функции (1) и (9) запишем как:

$$Q_{00}^{(0)}(t)=0,7\left(1-e^{-0,25t}\right),\ Q_{01}^{(0)}(t)=0,1\left(1-e^{-0,125t}\right);$$

$$Q_{02}^{(0)}(t)=0,2\left(1-e^{-0,125t}\right),\ Q_{10}^{(1)}(t)=\left(1-e^{-t}\right),$$

$$Q_{10}^{(2)}(t)=\left(1-e^{-0,5t}\right),\quad (k=1,2,3),$$

$$Q_{10}^{(3)}(t)=\left(1-e^{-t}\right),\ Q_{11}^{(k)}(t)=Q_{12}^{(k)}(t)=0,$$

$$Q_{20}^{(1)}(t)=\left(1-e^{-t}\right),\ Q_{20}^{(4)}(t)=\left(1-e^{-2t}\right);$$

$$H_0^{(0)}(t)=0,7\left(1-e^{-0,25t}\right)+0,3\left(1-e^{-0,125t}\right),$$

$$H_1^{(0)}(t)=\left(1-e^{-0,25t}\right),\ H_1^{(2)}(t)=\left(1-e^{-0,5t}\right),$$

$$H_1^{(3)}(t)=\left(1-e^{-t}\right),$$

$$H_2^{(1)}(t)=\left(1-e^{-t}\right),\ H_2^{(4)}(t)=\left(1-e^{-2t}\right).$$

С учетом обозначений:

$$x_{00}=1,\ x_{k,j}=d_j^{(k)},\ r\in R_j,\ j\in\tilde{S},$$

матрица $q(\alpha,x)=\left[q_{i,j}(\alpha,x)\right](i,j\in S)$ с элементами $q(\alpha,x)=\sum_{r\in R_j}x_{k,i}q_{i,j}^{(r)}(\alpha)(i,j\in S)$ примет вид

$$q(\alpha,x)=\begin{bmatrix}\dfrac{0,175}{\alpha+0,25} & \dfrac{0,0125}{\alpha+0,125} & \dfrac{0,025}{\alpha+0,125}\\ \dfrac{x_{11}+x_{31}}{\alpha+1}+\dfrac{x_{21}}{\alpha+0,5} & 0 & 0\\ \dfrac{x_{12}}{\alpha+1}+\dfrac{x_{42}}{\alpha+2} & 0 & 0\end{bmatrix}.$$

Найдем определитель матрицы $\left[I-q(\alpha,x)\right]$:

$$D(\alpha,x)=-\frac{0,025}{\alpha+0,125}\left(\frac{x_{12}}{\alpha+1}+\frac{2x_{42}}{\alpha+2}\right)+1-$$
$$-\frac{0,175}{\alpha+0,25}-\frac{0,0125}{\alpha+0,125}\left(\frac{x_{11}}{\alpha+1}+\frac{x_{21}}{\alpha+0,5}\right).$$

В матрице $\left[I-q(\alpha,x)\right]^{-1}=\left[\mu_{i,j}(\alpha,x)\right],(i,j\in S)$;

$$\mu_{00}(\alpha,x)=0,\ \mu_{01}(\alpha,x)=\frac{1}{D(\alpha,x)}\frac{0,0125}{\alpha+0,125},$$

$$\mu_{02}(\alpha,x)=1/D(\alpha,x)\cdot 0,025/(\alpha+0,125),$$

$$\mu_{10}(\alpha,x)=\frac{1}{D(\alpha,x)}\left(\frac{x_{11}+x_{31}}{\alpha+1}+\frac{x_{21}}{\alpha+0,5}\right),$$

$$\mu_{11}(\alpha,x)=\frac{1}{D(\alpha,x)}\left(\begin{array}{c}1-\dfrac{0,175}{\alpha+0,25}-\dfrac{0,0125}{\alpha+0,125}\times\\ \times\left(x_{12}/(\alpha+1)+2x_{42}/(\alpha+2)\right)\end{array}\right),$$

$$\mu_{12}(\alpha,x)=\frac{1}{D(\alpha,x)}\frac{0,025}{\alpha+0,125}\left(\frac{x_{11}+x_{31}}{\alpha+1}+\frac{x_{21}}{\alpha+0,5}\right),$$

$$\mu_{20}(\alpha,x)=\frac{1}{D(\alpha,x)}\left(\frac{x_{12}}{\alpha+1}+\frac{2x_{42}}{\alpha+2}\right),$$

$$\mu_{21}(\alpha,x)=\frac{1}{D(\alpha,x)}\frac{0,0125}{\alpha+0,125}\left(\frac{x_{12}}{\alpha+1}+\frac{2x_{42}}{\alpha+2}\right),$$

$$\mu_{22}(\alpha,x)=\frac{1}{D(\alpha,x)}\left(\begin{array}{c}1-\dfrac{0,0175}{\alpha+0,25}-\dfrac{0,0125}{\alpha+0,125}\times\\ \times\left(\dfrac{x_{11}+x_{31}}{\alpha+1}+\dfrac{x_{21}}{\alpha+0,5}\right)\end{array}\right).$$

В (18) $k_i^{(r)}$ принимают такие значения:

$k_0^{(0)}=70000$ у.е., $k_1^{(1)}=-70000$ у.е., $k_1^{(2)}=-150$ у.е., $k_1^{(3)}=-400$ у.е., $k_2^{(1)}=-70000$ у.е., $k_2^{(4)}=-12000$ у.е.,

а величины $\zeta_i^{(r)}$ записываются в виде:

$$\zeta_0^{(0)}(\alpha)=\frac{70000}{\alpha}\left(1-\frac{0,175}{\alpha+0,25}-\frac{0,0375}{\alpha+0,125}\right),$$

$$\zeta_1^{(1)}(\alpha)=-\frac{70000}{\alpha}\left(1-\frac{1}{\alpha+1}\right),$$

$$\zeta_1^{(2)}(\alpha)=-\frac{150}{\alpha}\left(1-\frac{0,5}{\alpha+0,5}\right),$$

$$\zeta_1^{(3)}(\alpha)=-\frac{400}{\alpha}\left(1-\frac{1}{\alpha+1}\right),$$

$$\zeta_2^{(1)}(\alpha)=-\frac{70000}{\alpha}\left(1-\frac{1}{\alpha+1}\right),$$

$$\zeta_2^{(4)}(\alpha)=-\frac{1200}{\alpha}\left(1-\frac{2}{\alpha+2}\right).$$

Найдем решения системы $\tilde{C}$ с помощью алгоритма пересечения решений отдельных неравенств (24), полагая, что правые части $b_k$ этих неравенств удовлетворяют условиям:

$$c_{11}<b_1<2c_1,\ b_2>c_{21},\ b_3>c_{31},\ b_4>c_{42}. \quad (30)$$

С учетом условий (30) находим следующие решения отдельных неравенств системы $\tilde{C}$:

$r_1=1:1-(1,0);\ 2-(0,1),\ 3-(0,0);$

$r_1=2:1-(1,0);\ 2-(0,0);$

$r_1=3:1-(1,0);\ 2-(0,0);$

$r_1=4:1-(0,1);\ 2-(0,0).$

При наличии $Z^{(0)}=\{\{1,2,3\},\{1,4\}\}$ на последнем шаге алгоритма получаем:

$$Z_{1221}^{(4)}=\{\{1\},\{4\}\},\quad Z_{1222}^{(4)}=\{\{1\},\varnothing\},$$

$$Z_{2121}^{(4)}=\{\{2\},\varnothing\},\quad Z_{2122}^{(4)}=\{\{2\},\{1\}\},$$

$$Z_{2211}^{(4)}=\{\{3\},\varnothing\},\quad Z_{2212}^{(4)}=\{\{3\},\{1\}\},$$

$$Z_{3121}^{(4)}=\{\{2\},\{4\}\},\quad Z_{3122}^{(4)}=\{\{2\},\varnothing\},$$

$$Z_{3211}^{(4)}=\{\{3\},\{4\}\},\quad Z_{3212}^{(4)}=\{\{3\},\varnothing\}.$$

Таким образом, решениями системы $\tilde{C}$ являются векторы: $1-\{\{1\},\{4\}\}$, $2-\{\{2\},\{1\}\}$, $3-\{\{3\},\{1\}\}$, $4-\{\{2\},\{4\}\}$, $5-\{\{3\},\{4\}\}$.

Им соответствуют следующие значения булевых переменных $\{x_{k,j}\}$:

1) $x_{1,1}=1,\ x_{4,2}=1;$
2) $x_{2,1}=1,\ x_{1,2}=1;$
3) $x_{2,1}=1,\ x_{4,2}=1;$
4) $x_{3,1}=1,\ x_{1,2}=1;$
5) $x_{3,1}=1,\ x_{4,2}=1.$

Не указанные переменные в каждом $i$-ом решении равны нулю.

С учетом выражений 28 и 29 $f(\alpha,x)$ при $\alpha=0,1$, что соответствует инфляции, равной 10%, и начальном распределении $y=(1,0,...,0)$ принимает значения: $f_1=-51903$; $f_2=-5329040$; $f_3=-17161$; $f_4=-26939$; $f_5=-3325$/

Таким образом, при $\alpha=0,1$ и $y=(1,0,...,0)$ оптимальной нерандомизированной марковской стационарной стратегией буде стратегия $x_{1,1}=0, x_{2,1}=0,\ x_{3,1}=1, x_{1,2}=0,\ x_{4,2}=1$, соответствующая решению $\{\{3\},\{4\}\}$ системы $\tilde{C}$.

## Выводы

В данной работе усовершенствован метод управления рисками разработки ПО. В основу данного метода была положена полумарковская модель принятия решений для управляемого марковского процесса в непрерывном времени. Отличительной особенностью предложенного метода является использование псевдобулевых методов бивалентного программирования с нелинейной целевой функцией и линейными ограничениями для определения оптимальной стратегии устранения эксплуатационных ошибок. Проведенные исследования показали, что используемые в данной работе теоретические положения в достаточном объеме отражают стандарты и возможности современных методологий тестирования ПО.

В качества примера рассмотрены ситуации возникновения ошибок безопасности ПО, и определена оптимальная стратегия управления для устранения указанной аномальной ситуации. Следует заметить, что представленный в работе метод целесообразно использовать не только при управлении

рисками безопасности ПО, но и при функциональном, нагрузочном, стрессовом и других видах тестирования для предотвращения возможных потерь.

## Список литературы


*1. Krishnan M. Soumya Software Development Risk Aspects and Success Frequency on Spiral and Agile Model / M. Soumya Krishnan // International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 1, January 2015 pp.301-310*

*2. Zeng Y. Risk Management For Enterprise Resource Planning System Implementations in Project-Based Firms : dis. for the degree of PHD / Zeng Yajun, Maryland, 2010 – 210 p.*

*3. Бриткин А. И. Риски, связанные с внедрением технологий, в проектах разработки программного обеспечения / А. Бриткин // Социально-экономические и технические системы. – 2007. – № 8 (42)*

*4. Вишняков Я.Д Общая теория рисков: учеб. пособие для студ. высш. учеб. заведений / Я.Д. Вишняков, Н.Н. Радаев. — М. : Изд. центр «Академия», 2008. – 368 с.*

*5. Шапкин А.С. Теория риска и моделирование рисковых ситуаций / А.С. Шапкин, В.А. Шапкиню.– М.: «Дашкв и К», 2005. – 880 с.*

*6. Boehm B.W. A spiral model of software development and enhancement / Boehm B., Egyed A. // IEEE Computer, May 1988 pp. 61-72*

*7. Исикава К. Японские методы управления качеством / К. Исикава, Сокр.пер. с англ. / Под. Ред. А. В. Гличева. — М: Экономика, 1988. — 214 с.*

*8. В.Д. Ногин. Принятие решений при многих критериях. Учебно-методическое пособие.– СПб.: ЮТАС, 2007. – 104 с.*

*9. Geymayr J. Fault-Tree Analysis: A Knowledge-Engineering Approach / J. Geymayr, N. Ebecken // IEEE Transactions on Reliability. – 1995. – № 44(1), pp. 37 – 45.*

*10. Анализ дерева отказов (Fault tree analysis (FTA)) / Электронный вариант Режим доступа: http://www.statistica.ru/knowledge-clusters/technical-sciences/analiz-dereva-otkazov.*

***11.** Інженерія програмного забезпечення : Навч. посібник / [Смірнов О.А., Коваленко О.В., Мелешко Є.В. та ін.] – К.: РВЛ КНТУ, 2013. – 409 с.*

*12. Будников С.А. Полумарковская модель сложного конфликта радиоэлектронных систем [Текст] / С.А. Будников // V Межд. конф. «Методы и средства управления технологическими процесами», Саранск, 19 – 21 ноября 2009 года. Режим доступа: – http://fetmag.mrsu.ru/2009-2.*

*13. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. Изд. 3-е, испр. и доп. – М.: Комкнига. – 2005.− 400с.*

*14. Semi-Markov risk models for finance, insurance and reliability [Electronic resource] / J. Jacques, M. Raimondo. - Electronic text data. - Boston, Ma: Springer Science + Business Media LLC, 2007.*

*15. К.В. Литвиненко Полумарковский гиперслучайный подход к оценке рисков систем / К.В. Литвиненко // Збірник наукових праць ОДАТРЯ, №1(4). – 2014. – С.77-80.*

*16. Коваленко А.В. Задачи распознавания ситуаций в ERP системах/ А.А. Смирнов, А.В. Коваленко, А.С. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 4(120). – Х.: ХУПС – 2014. – С. 161-164*

*17. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.*

*18. Проблемы анализа и оценки рисков информационной деятельности / А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.*

*19. Метод качественного анализа рисков разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.*

*20. Метод количественной оценки рисков разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Х.: ХУПС. – 2016. – С. 128-133.*


*Надійшла до редколегії 2.06.2016*

**Рецензент:** д-р техн. наук, доц. М.А. Павленко, Харківський університет Повітряних Сил ім. Кожедуба, Харків.

### МЕТОД УПРАВЛІННЯ РИЗИКАМИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

О.В. Коваленко

*У даній роботі завдання управління ризиками розробки програмного забезпечення за умови обмеженості коштів (фінансових, технічних та ін.) виділених на усунення помилок безпеки, розглядається у вигляді напівмарковської моделі прийняття рішень для керованого процесу в безперервному часу з критерієм мінімуму витрат на усунення аномалій. Розроблено метод управління ризиками розробки програмного забезпечення, що відрізняється від відомих використанням псевдобулевих методів бівалентного програмування з нелінійної цільової функцією і лінійними обмеженнями для визначення оптимальної стратегії усунення експлуатаційних помилок. В якості прикладу розглянуті ситуації виникнення помилок безпеки програмного забезпечення, і визначена оптимальна стратегія управління для усунення зазначеної аномальної ситуації.*

***Ключові слова:** управління ризиками, розробка програмного забезпечення, псевдобулеві методи бівалентного програмування.*

### METHOD OF RISK MANAGEMENT SOFTWARE DEVELOPMENT

A.V. Kovalenko

*In this paper, the problem of risk management software development, provided the limited resources (financial, technical, etc.) allocated to the elimination of security bugs, regarded as a semi-Markov decision model for the controlled process in continuous time with a minimum criterion to eliminate anomalies costs. A risk management software development method, which differs from the known methods using pseudo bivalent programming with nonlinear objective function and linear constraints to determine the optimal strategy for elimination of operational errors. For example, consider the situation of occurrence of software security bugs and determined the optimal control strategy to eliminate this abnormal situation.*

***Keywords**: risk management, software development, pseudo-dual-mode programming methods.*