УДК 004.41:004.056

А.А. Смирнов, А.В. Коваленко

Кировоградский национальный технический университет, Кировоград

ИСПОЛЬЗОВАНИЕ ПСЕВДОБУЛЕВЫХ МЕТОДОВ БИВАЛЕНТНОГО ПРОГРАММИРОВАНИЯ ДЛЯ УПРАВЛЕНИЯ РИСКАМИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В данной работе предлагается использовать псевдобулевые методы бивалентного программирования с нелинейной целевой функцией и линейными ограничениями, для формирования метода управления рисками разработки программного обеспечения, с целью определения оптимальной стратегии устранения эксплуатационных ошибок. При этом предлагается рассматривать задачу управления рисками разработки программного обеспечения, в виде полумарковской модели принятия решений для управляемого процесса в непрерывном времени с критерием минимума расходов на устранение аномалий.

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

Постановка проблемы в общем виде и анализ литературы

Всеобщие процессы глобализации экономических, финансовых, социальных и информационных отношений способствовали развитию направления риск-менеджмента. Однако общемировые финансовые кризисы показали недостаточно внимательное отношение к управлению рисками со стороны большинства представителей руководства организаций, в том числе и в Украине.

В настоящее время в большинстве организаций и предприятий различных форм собственности все больше внимания уделяется вопросам анализа и оценки рисков. Но, несмотря на это проблемы и вопросы, относящиеся к общей теории и методологии анализа, оценки и управления рисками требуют адаптации к подходам и положениям современного менеджмента, учета новых факторов становления и развития технологий, объединения известных «устоявшихся» положений теории рисков с новыми, прогрессирующими подходами анализа и синтеза.

Анализ литературы [1 – 11, 16 – 20] показал, что несмотря на достаточно глубокую историю развития понятия «риск» и попытки ряда известных авторов сконцентрировать свои разработки в область управления рисками отдельных отраслей и направлений деятельности, разработка новых, перспективных научных положений в этой области все же несколько «заужена» финансовой деятельностью. В то же время широкое использование в нашей работе информационных технологий требует повышенного внимания к этому направлению, и соответственно, более глубокого освещения вопросов риск-менеджмента IT-индустрии.

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 работе [12 – 15] для нахождения базисных решений системы линейных неравенств с булевыми переменными.

В таких условиях сформулируем основную задачу. Пусть каждому состоянию $i \in S$, где $S = \{0,1,2,...,N\}$ 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:

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

где $P_{ij}^{(r)}$ – вероятность перехода системы из состояния в состояние i; $F_{ij}^{(r)}(t)$ – 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 $F_{j0}^{(r)}(t)$, $j \in \widetilde{S} = S/\{0\}$, $r \in R_j$, 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 $i \in S$, $r \in R_i$ и вероятности $P_i^{(r)}$ удовлетворяют соотношениям:

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

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 $i = 0$, $R_0 = \{0\}$, вероятность $P_{0j}^{(r)} \neq 0$, $j \in S$ 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. В этом случае $F_{0j}^{(r)}(t)$ – функция распределения времени тестовой эксплуатации ПО между выявленными ошибками безопасности типа j.

При $i = 1,...,N$ для любого $r \in R_i$, $P_{i0}^{(r)} = 1$, $P_{ij}^{(r)} = 0$, $j \neq 0$, функция $F_{i0}^{(r)}(t)$ это 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 α, 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:

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

Обозначим i_n состояние системы после n-го перехода, u_n – принятое решение, а τ_n – время пребывания в этом состоянии ($n`= 0,1,2,...$), i_0 – начальное состояние. Допустимую стратегию β 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:

$$\sum_{j \in S} c_{rj} x_{rj} \leq b_r, \quad r \in R = \underset{j \in S}{U} R_j, \tag{3}$$

и зависящая от истории управляемой системы к моменту n-го перехода

$$z_n = (i_0, u_0, \tau_0, ..., i_{n-1}, u_{n-1}, \tau_{n-1}, i_n).$$

Мера $\beta_n(\bullet / z_n)$ задает рандомизированное правило выбора решения u_n на основе информации z_n. Такую стратегию β можно назвать рандомизированной. Стратегия β 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 β – 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 β, с нормой переоценки α, 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 β^*, называемая β – оптимальной, которая минимизирует суммарный средний расход $\upsilon(\alpha, \beta)$ при произвольной стратегии β и норме переоценки $\alpha(\alpha > 0)$. При этом $\upsilon(\alpha, \beta)$ есть $(N+1)\times 1$-мерный вектор $(\upsilon_0(\alpha, \beta), \upsilon_1(\alpha, \beta), \ldots, \upsilon_N(\alpha, \beta))$, где:

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

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

$$y = (y_0, y_1, \ldots, y_N), \tag{5}$$

$$\sum_{i \in S} y = 1, y \geq 0, i \in S. \tag{6}$$

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.

2. Задача управления рисками разработки программного обеспечения, в виде 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 $P^{(r)} = \{p_{ij}^{(r)}\}$, которая задает так называемую вложенную цепь Маркова. Элементы $p_{ij}^{(r)}$ при любых $i, j \in S$ 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:

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

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

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

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

являющуюся функцией распределения времени пребывания процесса в состоянии i при принятии решения $r \in R_i$.

Случайный процесс $(Z_t), t \geq 0$ 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

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

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:

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

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 выражения 4 можно записать в виде $\upsilon_i(\alpha)$, и для этого уравнения воспользоваться основными положениями уравнения (интеграла) Лапласа-Стилтьеса. В соответствии с работами [12-15] для любой 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 $\operatorname{Re} s > \alpha$ существует функция:

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

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

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

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

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

есть преобразование Лапласа функции $F(t)$.

Из (12) при $s \neq 0$ находим

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

Интегрируем по частям с учетом (9) находим

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

Проводя преобразования, переходя в выражении 14 к пределу $t \to \infty$ и применяя (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 $t \to \infty$ получаем:

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

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:

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

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

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

$$\Im(\alpha) = \left(\zeta_0(\alpha), \ldots, \zeta_N(\alpha)\right)^T, \quad \wp(\alpha) = \left(\nu_0(\alpha), \ldots, \nu_N(\alpha)\right)^T,$$

(T – символ транспонирования матрицы). Тогда:

$$\wp(\alpha) = \Im_0(\alpha) + q(\alpha)\wp(\alpha), \tag{19}$$

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

Из выражения (19) найдем:

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

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 $\{d_i^{(r)}\}$ $(r\in R_i)$ – нерандомизированная марковская стационарная стратегия системы разработки ПО в состоянии j, $d_j^{(r)}\in\{0,1\}$, $\sum_{j\in S} d_j^{(r)}=1$ и $x_{00}=1$, $x_{rj}=d_j^{(r)}$, $r\in R_i$, $j\in\tilde{S}$. Минимизация расходов (21) приводит к задаче оптимизации для булевых переменных $X=\{x_{rj}\}$, $r\in R_i$, $j\in\tilde{S}$:

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

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

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

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

3. Построение оптимальной нерандомизированной марковской стационарной стратегии

Обозначим систему (24), (25) как систему *C*. Она является системой псевдобулевых неравенств.

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

Для построения решений системы $\tilde{C}$ при известных допустимых решениях каждого неравенства (24) применим следующий подход. Решения системы $\tilde{C}$ находятся в виде $Z=\{s_j\}$, $j=1,\ldots N$, 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 $r=r_1$, а также, что α_j является j-м элементом допустимого решения данного неравенства, $\alpha_j\in\{0,1,\varphi\}$, где φ – неопределенный параметр из множества $\{0,1\}$, называемый в дальнейшем почерком, можно сформулировать следующие правила для r-го шага алгоритма построения решений системы $\tilde{C}$.

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

2. Если $\alpha_j=1$, то при $r_1\in s^{(r-1)}$ допускаем $s^{(r)}=\{r_1\}$, а при $r_1\,\overline{\in}\,s^{(r-1)}$ допускаем $s^{(r)}$ равно пустому множеству.

3. Если $\alpha_j=0$, то допускаем $s^{(r)}=s^{(r-1)}/\{r_1\}$.

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

На *m* шаге алгоритма получается вектор $Z^{(m)}=\{\alpha_1^{(m)},\ldots,\alpha_N^{(m)}\}$, 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 $f(\alpha,X)$. Это решение может находиться разными известными методами линейного программирования, или просто путем непосредственного сравнения значений $f(\alpha,X)$ при определении X системы $\tilde{C}$.

Численная реализация изложенного метода управления рисками безопасности индустрии программного обеспечения для полумарковской модели принятия решений при аномальных ситуациях безопасности представлена в дальнейших работах.

Выводы

В данной работе использованы псевдобулевые методы бивалентного программирования с нелинейной целевой функцией и линейными ограничениями для формирования метода управления рисками разработки программного обеспечения, с целью определения оптимальной стратегии устранения эксплуатационных ошибок. При этом, задача управления рисками разработки программного обеспечения, рассматривается в виде полумарковской модели принятия решений для управляемого процесса в непрерывном времени с критерием минимума расходов на устранение аномалий.

Проведенные исследования показали, что используемые в данной работе теоретические положения в достаточном объеме отражают стандарты и возможности современных методологий тестирования ПО.

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. – P.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 – P. 210.*
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.: Изд.-торговая корпорация «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 ресурс]. – 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 [Электронный ресурс] / С.А. Буд- ников // V Межд. конф. «Методы и средства управления технологическими процесами», Саранск, 19 – 21 ноября 2009 года. Режим доступа: – - [http://fetmag.mrsu.ru/2009-2.](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.− 400 с.*
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. *Литвиненко К.В. Полумарковский гиперслучайный подход к оценке рисков систем / К.В. Литвиненко // Збірник наук. праць ОДАТРЯ. – 2014. – №1(4). – 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". – Вип. 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". – Вип. 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". – Вип. 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. – № 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. — 2016. — № 2. — С. 128-133.*

Надійшла до редколегії 2.03.2016

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

ВИКОРИСТАННЯ ПСЕВДОБУЛЕВИХ МЕТОДІВ БІВАЛЕНТНОСТІ ПРОГРАММИРОВАНИЯ ДЛЯ УПРАВЛІННЯ РИЗИКАМИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

О.А. Смірнов, О.В. Коваленко

У даній роботі пропонується використовувати псевдобулеві методи бівалентного програмування з нелінійною цільовою функцією і лінійними обмеженнями, для формування методу управління ризиками розробки програмного забезпечення, з метою визначення оптимальної стратегії усунення експлуатаційних помилок. При цьому пропонується розглядати задачу управління ризиками розробки програмного забезпечення, у вигляді напівмарковських моделі прийняття рішень для керованого процесу в безперервному часу з критерієм мінімуму витрат на усунення аномалій.

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

USING METHODS PSEUDO BIVALENT PROGRAMMING FOR RISK MANAGEMENT SOFTWARE DEVELOPMENT

A.A. Smirnov, A.V. Kovalenko

In this paper we propose to use a pseudo-dual-programming techniques with non-linear objective function and linear constraints for the formation method of risk management software development, in order to determine the optimal strategy for elimination of operational errors. At the same time it is proposed to consider the problem of risk management software development, in the form of a semi-Markov decision model for the controlled process in continuous time with a minimum criterion to eliminate anomalies costs.

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