

УДК 004.41:004.056

А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский

Кировоградский национальный технический университет, Кировоград

МЕТОД КОЛИЧЕСТВЕННОЙ ОЦЕНКИ РИСКОВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Данная статья посвящена разработке метода количественной оценки рисков разработки программного обеспечения. Его отличительной особенностью является комплексное использование методики «Анализа дерева отказов» и способа оценки показателя чистой приведенной стоимости проекта разработки программного обеспечения с учетом негативных факторов возможного невыявления уязвимостей безопасности программного обеспечения.

Ключевые слова: оценка рисков, разработка программного обеспечения.

Введение

Постановка проблемы исследования. Авторами разработан комплекс методов качественного анализа и количественной оценки рисков разработки программного обеспечения, что позволило решить противоречие, возникающих при разработке программного обеспечения (ПО), и заключающееся в пренебрежении фирмами-разработчиками ПО факторов уязвимости безопасности программного обеспечения.

В качестве решения указанной проблемы предложено использование разработанных методов качественного анализа и количественной оценки рисков разработки программного обеспечения.

Анализ литературы [1 – 16] и проведенные исследования показали, что общая последовательность оценки рисков чаще всего включает в себя следующие действия:

1. Выявление источников и причин риска разработки ПО, этапов и работ, при выполнении которых возникает риск.
2. Идентификация всех возможных рисков, свойственных рассматриваемому проекту.
3. Документирование результатов и их последующая приоритизация.
4. Оценка уровня отдельных рисков и риска проекта в целом, определяющая его экономическую целесообразность.
5. Определение допустимого уровня риска разработки ПО.
6. Разработка мероприятий по снижению риска.

В соответствии с данным алгоритмом оценка риска подразделяется на три взаимно дополняющих направления: качественный (этапы 1, 2, 3) и количественный анализ (этапы 4, 5) рисков разработки ПО, а также управление (этап 6).

Данная статья посвящена разработке метода количественной оценки рисков разработки программного обеспечения.

Основной раздел

Исследование дерева рисков разработки программного обеспечения

Как указано выше для эффективного управления проектами нужно не только идентифицировать риски, но и оценивать их количественно. При этом особенности современных фирм-разработчиков ПО как надсистем, особенности отдельных этапов разработки ПО, бизнес-процессов и их групп как подсистем, определяют ряд проблем. К этим проблемам можно отнести:

- отсутствие статистических данных об удачных и неудачных проектах внедрения систем, особенно на операционном уровне;
- отсутствие статистических данных о провалах безопасности при эксплуатации ПО;
- уникальность каждого проекта внедрения;
- долгосрочность подобных проектов;
- высокую стоимость подобных проектов;
- значительную составляющую несистемных факторов риска, связанных с внутренними факторами фирмы-разработчика ПО.

Учитывая приведенные факторы можно отметить, что для оценки рисков разработки ПО можно использовать три основных подхода:

- формализованное описание неопределенности рисков разработки ПО;
- корректировку показателей проекта путем замены их проектных значений на ожидаемые;
- проверку устойчивости.

Формализованная оценка неопределенности, которая возникает в процессе реализации проектов, при отсутствии статистических данных, может опираться на два метода: экспертных оценок и нечетких множеств.

Проведенные исследования показали, что использование субъективно-аксиологической вероятности (экспертных оценок) является вынужденным отступлением науки перед наращиванием несистемных факторов риска разработки ПО, но это требует

последующей верификации модели и вычисленных показателей риска. В этой связи целесообразным представляется переход от субъективных экспертных методов к методам, которые используют теорию нечетких множеств.

Корректировка показателей проекта (процесса разработки ПО) путем замены их проектных значений на значения с учетом рисков вызывает дополнительные сложности, связанные с неопределенностью всех факторов, влияющих финансовые, имиджевые и другие приобретения и потери.

Для преодоления этих проблем можно использовать методы, которые опираются на описание бизнес-процессов и позволяют выявлять изменения отдельных их параметров, связанных с разработкой, внедрением и эксплуатацией ПО.

Проведенные исследования показали, что адекватным инструментом для таких исследований, является «Анализ дерева отказов» (*Fault Tree Analysis, FTA*), предложенный в работах [9 – 11].

FTA методика описана в нескольких отраслевых и государственных стандартах: *NUREG CPH-0492* для атомной энергетики [9 – 11]. Ориентированная на космос версия этого стандарта используется *NASA*, стандарт *SAE ARP4761* для гражданской авиационной отрасли [9 – 11], *MIL-HDBK-338* – для военных систем. *IEC* стандарт предназначен для межотраслевого использования и был принят в качестве европейского стандарта *EN 61025* [9 – 11].

Анализ данного подхода количественной оценки рисков показал целесообразность использования графической модели *FTA* в терминах математической логики. Это поможет формализовать условия влияния факторов риска в различных их комбинациях на конечные показатели проекта разработки ПО.

Пример дерева рисков разработки ПО приведен на рис. 1. На этой схеме группы рисков формируются с учетом разработанной классификации рисков разработки ПО и результатов качественной оценки ранга рисков.

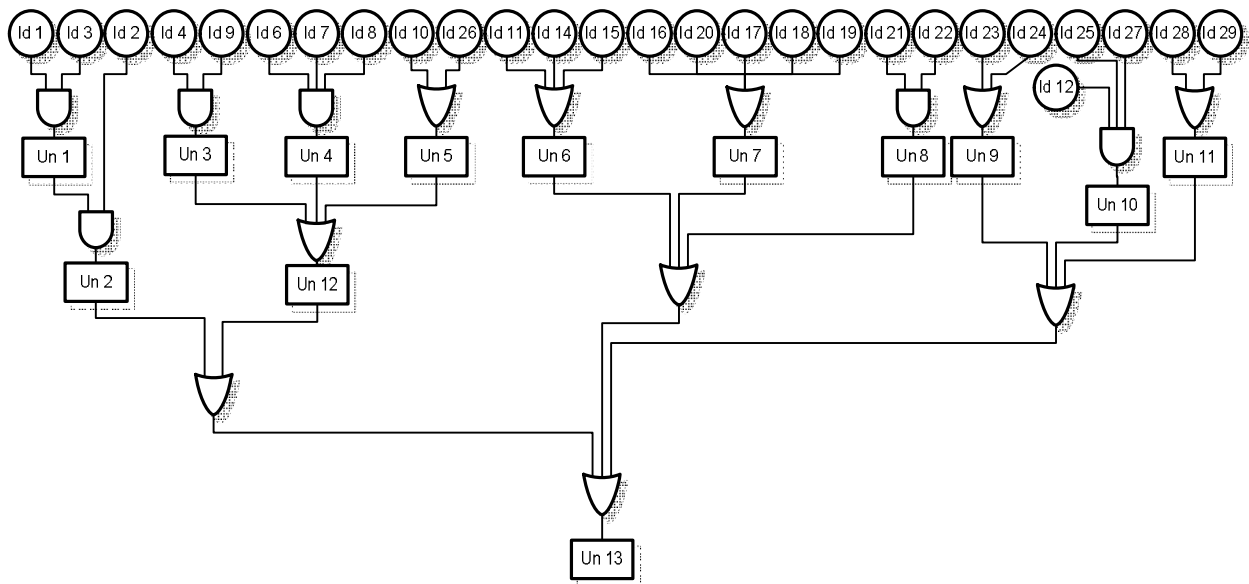


Рис. 1. Пример дерева рисков разработки ПО

Кроме того, для наглядности оценки групп рисков предлагается использовать логические элементы «*and*» и «*or*», которые позволяют использовать методы математической логики для расчета коэффициентов групп рисков и общего риска.

Очевидно, что для этой схемы общий коэффициент рисков разработки ПО можно рассчитать по формуле:

$$P(\text{Un } 13) = P(\text{Un } 2) + P(\text{Un } 12) + P(\text{Un } 6) + P(\text{Un } 7) + P(\text{Un } 8) + P(\text{Un } 9) + P(\text{Un } 10) + P(\text{Un } 11), \quad (1)$$

где

$$P(\text{Un } 2) = P(\text{Un } 1) \cdot P(\text{Id } 2) - \text{коэффициент рис-}$$

ка выбора неправильной методики разработки ПО;

$$P(\text{Un } 1) = P(\text{Id } 1) \cdot P(\text{Id } 3) - \text{коэффициент риска}$$

выбора неправильной методики разработки ПО;

$P(\text{Un } 12) = P(\text{Un } 3) + P(\text{Un } 4) + P(\text{Un } 5)$ – коэффициент риска неадекватного менеджмента проекта;

$P(\text{Un } 3) = P(\text{Id } 4) \cdot P(\text{Id } 9)$ – коэффициент риска пренебрежения топ-менеджментом разработки ПО;

$P(\text{Un } 4) = P(\text{Id } 6) \cdot P(\text{Id } 7) \cdot P(\text{Id } 8)$ – коэффициент риска неадекватного менеджмента активной стадии разработки ПО;

$$P(\text{Un } 5) = P(\text{Id } 10) + P(\text{Id } 26) - \text{коэффициент}$$

риска неадекватного операционного и социального менеджмента;

$P(Un\ 6) = P(Id\ 11) + P(Id\ 14) + P(Id\ 15)$ – коэффициент риска несоответствия профессионального уровня участников проекта;

$P(Un\ 7) = P(Id\ 16) + P(Id\ 17) + P(Id\ 18) + P(Id\ 19) + P(Id\ 20)$ – коэффициент

технологических рисков разработки ПО;

$P(Un\ 8) = P(Id\ 21) \cdot P(Id\ 22)$ – коэффициент

риска несоответствия сложности ПО уровню подготовки эксплуатанта;

$P(Un\ 9) = P(Id\ 23) + P(Id\ 24)$ – коэффициент

риска безопасности эксплуатации ПО;

$P(Un\ 10) = P(Id\ 12) \cdot P(Id\ 25) \cdot P(Id\ 27)$ – коэф-

фициент риска неадекватной коммуникации участников проекта;

$P(Un\ 11) = P(Id\ 28) + P(Id\ 29)$ – коэффициент

наступления правовых рисков.

Значения приведенных выше коэффициентов, полученные в результате качественного анализа рисков разработки ПО приведены в табл. 1.

Таблица 1

Значения элементов
дерева рисков разработки ПО

Элемент дерева рисков	Коэффициент	Элемент дерева рисков	Коэффициент
Id 1	0,08	Id 16	0,06
Id 2	0,05	Id 17	0,05
Id 3	0,06	Id 18	0,03
Id 4	0,05	Id 19	0,07
Id 5	0,01	Id 20	0,03
Id 6	0,07	Id 21	0,04
Id 7	0,07	Id 22	0,03
Id 8	0,05	Id 23	0,05
Id 9	0,06	Id 24	0,07
Id 10	0,06	Id 25	0,04
Id 11	0,04	Id 26	0,05
Id 12	0,04	Id 27	0,05
Id 13	0,02	Id 28	0,02
Id 14	0,04	Id 29	0,03
Id 15	0,6		

Используя выражение 1. и приведенные в табл. 1 значения коэффициентов можно определить, что общий коэффициент рисков разработки ПО $P(Un\ 13)=0,558$.

Следует заметить, что пренебрежение риском невыявления уязвимостей безопасности ПО ($P(Id\ 24)$) может снизить точность количественной оценки рисков разработки ПО до 13% ($P(Un\ 13)=0,488$), а пренебрежение коэффициентом риска безопасности экс-

плуатации ПО ($P(Un\ 9)$) снизит точность оценки до 22% ($P(Un\ 13)=0,438$). Это подтверждает необходимость учета рисков, невыявления уязвимостей безопасности ПО и неадекватного взаимовлияние эксплуатируемого и внедряемого ПО.

Оценка показателя чистой приведенной стоимости проекта разработки ПО

Еще одной особенностью проектов разработки ПО является их долгосрочность и необходимость учета информации, возникающей на очередных стадиях принятия решений. Для решения этой задачи в [1 – 3], например, предлагается, используя модульность программного обеспечения, оценивать их инвестиционную привлекательность с учетом разработанных компонент.

Проведенные исследования позволили найти такой подход вероятностной оценки рисков, который позволил использовать основные положения нечетко-множественной теории при оценке ключевых показателей результативности проекта. Возможность его использования оценим на примере показателя стоимости C_{NVP} (*Net Present Value*).

Анализ ряда работ, связанных с экономической теорией рисков показал, что расчета показателя стоимости часто используется классический метод определения чистой приведенной стоимости проекта:

$$C_{NVP} = \sum_{i=1}^n \frac{D_i}{(1+r)^i} - \sum_{i=1}^n \frac{C_i}{(1+r)^i}, \quad (2)$$

где n – количество периодов реализации проекта; D_i – финансовый поток доходов от проекта в период i ; C_i – финансовый поток расходов на проект в период i ; r – ставка дисконтирования.

Однако с учетом того, что проект разработки ПО является долгосрочным и зависит от многих факторов, существует необходимость рассматривать показатели возврата инвестиций в эти проекты совместно с процессами развития проекта. Кроме этого важно учитывать факторы динамического изменения рисковых компонентов в процессе проектирования, кодирования, тестирования и эксплуатации на всех «витках спирали» разработки программного обеспечения.

Проведенные исследования показали, что с учетом дополнительных факторов риска, а также (8) для расчета C_{NVP} целесообразно использовать следующее выражение:

$$C_{NVP} = \sum_{i=1}^n \frac{(B - AC_i)}{(1+r)^i} - \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} C_i^{(k)} \right)}{(1+r)^i} - \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} e^{-\lambda \times QC_i^{(k)}} \right)}{(1+r_i)^i}, \quad (3)$$

где V – финансовые инвестиций, поступающих в процессе разработки ПО в период i ;

AC_i – текущие затраты на поддержание и развитие системы разработки ПО в период i ;

ℓ – количество дополнительных видов расходов на приобретение, настройку и модернизацию технической, технологической, программной и других составляющих в процессе разработки программного обеспечения;

$QC_i^{(k)}$ – расходы на учет безопасности и тестирование уязвимости ПО;

λ – параметр влияния факторов безопасности и тестирование уязвимости ПО на чистую приведенную стоимость проекта;

$C_i^{(k)}$ – дополнительные расходы в процессе разработки ПО.

Отличительной особенностью математического выражения (3) является введение дополнительных составляющих

$$\sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} C_i^{(k)} \right)}{(1+r)^i}; \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} e^{-\lambda \times QC_i^{(k)}} \right)}{(1+r_1)^i},$$

характеризующих учет дополнительных расходов на аппаратную и программную модернизацию фирмы, усовершенствование системы ее управления, а также учет безопасности и тестирования уязвимости программного обеспечения.

Оценка эффективности проекта заключается в сравнении C_{NVP} с некоторым значением $C_{NVP}^{Preference}$, которое определяет минимально допустимый уровень приведенной стоимости проекта разработки ПО.

С учетом неопределенности составляющих, на основе теории нечетких множеств можно рассматривать треугольные нечеткие значения составляющих для выражения (3):

$\overline{B}_i = (B_i^{(min)}, \overline{B}_i, B_i^{(max)})$ если невозможно определить финансовые доходы, которые возникнут от модификации в процессе разработки ПО технологических этапов;

$\overline{AC}_i = (AC_i^{(min)}, \overline{AC}_i, AC_i^{(max)})$ если существует неопределенность расходов, необходимых для поддержки и развития системы разработки программного обеспечения;

$\overline{C}_i^{(k)} = (C_i^{(k)(min)}, \overline{C}_i^{(k)}, C_i^{(k)(max)})$ если существует неопределенность в отношении дополнительных затрат k -го предназначения.

Кроме того, подобным же образом нужно представить коэффициент дисконтирования:

$$\overline{r}_i = (r_i^{(min)}, \overline{r}_i, r_i^{(max)}) \text{ если инвестор не может}$$

оценить стоимость капитала, который будет использоваться в проекте.

Для приведения формулы (3) к виду, который может использоваться для вычислений, воспользуемся сегментным способом, приведенным в работе [3].

Если выбрать фиксированный уровень принадлежности нечетких чисел α (ординату функции принадлежности нечетких чисел), то можно применить операции нечеткой арифметики, которые позволят превратить выражение (3) в систему уравнений:

$$[C_{NVP1}, C_{NVP2}] = \begin{cases} \sum_{i=1}^n \frac{(B_{i1} - AC_{i1})}{(1+r_1)^i} - \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} C_{i1}^{(k)} \right)}{(1+r_1)^i} - \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} e^{-\lambda \times C_{i3}^{(k)}} \right)}{(1+r_1)^i}, \\ \sum_{i=1}^n \frac{(B_{i2} - AC_{i2})}{(1+r_2)^i} - \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} C_{i2}^{(k)} \right)}{(1+r_2)^i} - \sum_{i=1}^n \frac{\left(\sum_{k=1}^{\ell} e^{-\lambda \times C_{i3}^{(k)}} \right)}{(1+r_1)^i}. \end{cases} \quad (4)$$

Наиболее общий метод расчета функции риска для $NVP - R_{NVP}$, который представляется нечетким числом обоснован в [3]. Используются следующие соотношения:

$$R_{NVP} = \begin{cases} 0, & C_{NVP} < C_{NVP}^{(min)}; \\ R \cdot \left(1 + \frac{1-\alpha}{\alpha} \cdot \ln(1-\alpha) \right), & C_{NVP}^{(min)} \leq C_{NVP} < C_{NVP}^{(av)}; \\ 1 - (1-R) \cdot \left(1 + \frac{1-\alpha}{\alpha} \cdot \ln(1-\alpha) \right), & C_{NVP}^{(av)} \leq C_{NVP} < C_{NVP}^{(max)}; \\ 1, & C_{NVP} \geq C_{NVP}^{(max)}, \end{cases} \quad (5)$$

где

$$R = \begin{cases} \frac{C_{NVP} - C_{NVP}^{(min)}}{C_{NVP}^{(max)} - C_{NVP}^{(min)}}, & C_{NVP} < C_{NVP}^{(max)}; \\ 1, & C_{NVP} \geq C_{NVP}^{(max)}; \end{cases} \quad (6)$$

$$\alpha = \begin{cases} 0, & C_{NVP} < C_{NVP}^{(min)}; \\ \frac{C_{NVP} - C_{NVP}^{(min)}}{C_{NVP}^{(av)} - C_{NVP}^{(min)}}, & C_{NVP}^{(min)} \leq C_{NVP} < C_{NVP}^{(av)}; \\ \frac{C_{NVP}^{(max)} - C_{NVP}}{C_{NVP}^{(max)} - C_{NVP}^{(av)}}, & C_{NVP}^{(av)} \leq C_{NVP} < C_{NVP}^{(max)}; \\ 1, & C_{NVP} \geq C_{NVP}^{(max)}. \end{cases} \quad (7)$$

где $C_{NVP}^{(av)}$ – задает средний уровень нечеткого числа.

Исследуем и проведем оценку рисков разработки ПО при следующих условиях: $n=3$; $B_1=15$; $B_2=16$; $B_3=17$; $r=(0,1; 0,125; 0,15)$; $AC_1=AC_2=AC_3=1$; $\ell = 3$ (1 – расходы на приобретение, настройку и модернизацию программного обеспечения в ходе реализации проекта, 2 – расходы на модернизацию аппаратного обеспечения фирмы и реорганизацию системы управления, которая необходима после модернизации программного обеспечения и оборудования, 3 – расходы на учет безопасности программного обеспечения; и дополнительное тестирование его уязвимости);

$$C_{i1} = (1,1; 1,2; 1,3); C_{i2} = (0,5; 0,6; 0,7);$$

$$C_{i3} = (0,1; 0,2; 0,3), \lambda = 1, C_{NVPreference} = 0.$$

На рис. 2 представлены графики зависимости $[C_{NVP1}, C_{NVP2}]$ от значения B финансовых инвестиций, поступающих в процессе разработки программного обеспечения в период i (графики CN3 и CN4 иллюстрирует зависимость при отсутствии дополнительных затрат на учет безопасности и тестирование уязвимости ПО).

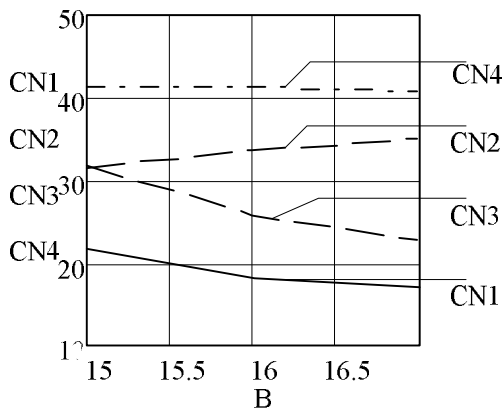


Рис. 2. Графики зависимости $[C_{NVP1}, C_{NVP2}]$ от значения B

Как видно из этого рисунка введение дополнительных расходов в процессе разработки программного обеспечения до 1,5 раз повышает чистую приведенную стоимость проекта.

Результаты расчета R_{NVP} для уровней принадлежности α от 0,1 до 1 с шагом 0,1 могут быть представлены в нечеткой треугольной форме (рис. 3).

Риск проекта разработки программного обеспечения определяется вероятностью попадания точки в область, ограниченную графиком нечеткого числа при $R_{NVP} < 0$.

Эта вероятность определяется отношением площади под этой частью графика к площади под всем графиком.

Исходя из этого с помощью выражения:

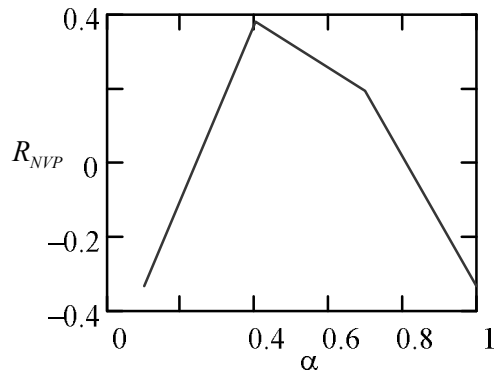


Рис. 3. Треугольная форма нечеткого значения R_{NVP} для рассматриваемого примера

$$P_{NVP} = \begin{cases} \frac{(R_{NVP_{min}})^2}{(R_{NVP_{av}} - R_{NVP_{min}}) \times (R_{NVP_{max}} - R_{NVP_{min}})}, & \text{если } R_{NVP_{av}} > 0; \\ 1 - \frac{(R_{NVP_{max}})^2}{(R_{NVP_{max}} - R_{NVP_{av}}) \times (R_{NVP_{max}} - R_{NVP_{min}})}, & \text{если } R_{NVP_{av}} < 0; \end{cases} \quad (8)$$

определим уровень риска для рассматриваемого примера проекта разработки ПО.

Для приведенного примера проекта разработки ПО уровень риска составит 0,713.

Таким образом, предложенный общий способ оценки показателя чистой приведенной стоимости проекта разработки ПО является средством к преодолению проблем, связанных с неэффективным завершением проектов разработки ПО. Способ предлагает рассматривать проект комплексно, с учетом необходимости учета безопасности и тестирование уязвимости ПО, с использованием инструментов, которые позволяют преодолеть сложность, неопределенность и долгосрочность проектов.

Для преодоления проблем отсутствия статистических данных и повышения достоверности экспертной оценки предлагается использовать методы теории нечетких множеств.

Учитывая все описанные выше этапы количественной оценки рисков разработки ПО можно отметить, что отличительной особенностью разработанного метода является комплексное использование методики «Анализа дерева отказов» и способа оценки показателя чистой приведенной стоимости проекта разработки ПО с учетом негативных факторов возможного невыявления уязвимостей безопасности ПО.

Результаты количественной оценки рисков разработки ПО могут быть использованы в качестве входных данных при управлении рисками непосредственно на последующих этапах разработки программного обеспечения; (прототипирование, кодирование, тестирование и т.д.).

Выводы

В ходе решения поставленной задачи разработан метод количественной оценки рисков разработки ПО. Его отличительной особенностью является комплексное использование методики «Анализа дерева отказов» и способа оценки показателя чистой приведенной стоимости проекта разработки ПО с учетом негативных факторов возможного невыявления уязвимостей безопасности ПО.

Использование усовершенствованной методики «Анализа дерева отказов» позволит до 22% повысить точность количественной оценки рисков разработки ПО. В то же время использование способа оценки показателя чистой приведенной стоимости проекта разработки ПО позволяет рассматривать проект комплексно, с учетом необходимости учета безопасности и тестирование уязвимости ПО, с привлечением инструментов, которые позволяют преодолеть сложность, неопределенность и долгосрочность проектов.

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 – Pp. 210.
3. Бриткин А.И. Риски, связанные с внедрением технологий, в проектах разработки программного обеспечения / А.И. Бриткин // *Социально-экономические и технические системы*. – 2007. – № 8 (42). – С. 24-26.
4. Вишняков Я.Д. Общая теория рисков: учеб. пособие для студ. высш. учеб. заведений / Я.Д. Вишняков, Н.Н. Радаев. 2-е изд., испр. – М.: Издательский центр «Академия», 2008. – 368 с.
5. Шапкин А.С. Теория риска и моделирование рисков ситуаций / А.С. Шапкин, В.А. Шапкину. – М.: Издат.-торговая корпорация «Дашкв и К», 2005. – 880 с.

6. Boehm B.W. A spiral model of software development and enhancement / Boehm B., Egedy 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. Ногин В.Д. Принятие решений при многих критериях: учеб.-метод. пособ. / В.Д. Ногин. – СПб. Издательство «ИЮТАС», 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 / [О.А. Смірнов, О.В. Коваленко, С.В. Мелешко та ін.]. – К.: РВЛ КНТУ, 2013. – 409 с.

12. Доренський О.П. Формалізація процесу зміни станів програмних об'єктів складних систем на основі формального апарату скінчених автоматів Мура / О.П. Доренський, О.А. Смірнов // *Зв'язок*. – 2014. – № 3 (109). – С. 27-31.

13. Dorensky O. Development of the theoretical bases of logical domain modeling of a complex software system / Oleksandr Dorensky, Alexey Smirnov // *International Journal of Computational Engineering Research (IJCER)*. – India: Delhi, 2014. – Vol. 4, Issue 4. – P. 19-23.

14. Лысенко И.А. Исследование уровней тестирования программного обеспечения инфотелекоммуникационных систем / И.А. Лысенко, А.А. Смирнов, Е.В. Мелешко // *Наука і техніка Повітряних Сил Збройних Сил України*. – Х.: ХУПС, 2014. – Вип. 4(17). – С. 79-81.

15. Лысенко И.А. Исследование процесса разработки программного обеспечения инфотелекоммуникационных систем / И.А. Лысенко, А.А. Смирнов, Л.И. Полищук // *Система озброєння і військова техніка*. – Х.: ХУПС, 2014. – Вип. 4(40). – С. 103-106.

16. Лысенко И.А. Исследование алгоритма выявления вида неучтенных тестовых случаев в процессе проектирования тестовых наборов / И.А. Лысенко, А.А. Смирнов // *Научно-виробничий журнал "Зв'язок"*. – К.: ДУТ, 2014. – № 2 (108). – С. 153-156.

Поступила в редколлегию 8.02.2016

Рецензент: д-р техн. наук, проф. И.В. Рубан, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

МЕТОД КІЛЬКІСНОЇ ОЦІНКИ РИЗИКІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

О.А. Смірнов, О.В. Коваленко, Н.М. Якименко, О.П. Доренський

Дана стаття присвячена розробці методу кількісної оцінки ризиків розробки програмного забезпечення. Його відмінною рисою є комплексне використання методики «Аналізу дерева відмов» і способу оцінки показника чистої приведеної вартості проекту розробки програмного забезпечення з урахуванням негативних чинників можливого виявлення вразливостей безпеки програмного забезпечення.

Ключові слова: оцінка ризиків, розробка програмного забезпечення.

METHOD OF QUANTITATIVE RISK ASSESSMENT SOFTWARE DEVELOPMENT

A.A. Smirnov, A.V. Kovalenko, N.M. Yakimenko, O.P. Dorensky

This article focuses on the development of a method of quantitative evaluation of software development risks. Its distinguishing feature is the integrated use of the method "Analysis of fault tree" and the method of estimating the net present value of a software development project, taking into account the possible negative factors of not detecting software security vulnerabilities.

Keywords: risk assessment, software development.