

УДК 621.039:004.05

В.И. ДУЖИЙ, А.В. ШОСТАК, И.В. ДУЖИЙ

*Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Украина***ОЦЕНКА ПОКАЗАТЕЛЯ ВЕРСИОННОЙ ИЗБЫТОЧНОСТИ
ПРИ ПРОЕКТИРОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Описаны типовые структуры графовой модели двухверсионного проекта и их параметры. Предложена аналитическая зависимость для показателя версионной избыточности, учитывающая длину минимальной версии проекта. Выполнено исследование влияния на показатель диверсности величины перекрытия версий проекта. Отмечено, что при увеличении перекрытия версий проекта показатель диверсности линейно уменьшается. Приведены примеры графовых моделей двухверсионных проектов и определены показатели диверсности для них. Даны практические рекомендации для разработки двухверсионных проектов.

Ключевые слова: двухверсионный проект, графовая модель, показатель диверсности.

Введение

К современным системам критического применения, являющимся сложными программно-аппаратными комплексами, предъявляются требования высокой надежности [1]. Одним из способов обеспечения требуемой надежности является использование многоверсионного программного обеспечения (ПО) [2-4]. Как показано в работах [4,5] многоверсионный проект ПО должен включать несколько версий, работающих одновременно и позволяющих компенсировать их ошибки. При разработке многоверсионного проекта ПО необходимо решить следующие задачи: генерация множества версий, анализ качественных показателей сгенерированных версий и всего проекта. В литературе описаны метрики, позволяющие выполнить анализ многоверсионных систем для оценки их надежности и безопасности [6]. Недостатком такого способа оценивания систем является то, что он применяется к уже разработанным системам и не в состоянии повлиять на их надежность-стоимостные характеристики.

В работе [7] предложена классификация типов многоверсионности, учитывающая этапы жизненного цикла, на котором вносится версионная избыточность. В этой работе предлагается разделить многоверсионность на внешнюю и внутреннюю, а формирование многоверсионного проекта выполнять в два этапа - формирование внешней и внутренней версионной избыточности проекта. В этой же работе предложен метод генерации множества версий проекта на основе внешней многоверсионности, и представление его в виде графовой модели. В этом методе графовая модель рассматривается как нисходя-

щий ярусный граф элементов технологий, которые будут использованы в данном проекте. В работе [7] описан алгоритм генерации множества версий, учитывающий совместимость различных элементов решений.

В работе [8] проанализированы классификационные схемы видов версионной избыточности, учитывающие возможности современных компьютерных технологий, а также обобщены модели многоверсионных систем при использовании нескольких видов версионной избыточности.

Наиболее простой реализацией многоверсионного проекта является использование двух версий. Такие двухверсионные проекты имеют меньшую стоимость и рекомендуются к использованию соответствующими нормативными документами [9]. По этим причинам в данной статье будут исследованы двухверсионные проекты ПО.

Целью данной работы является выбор и обоснование показателя версионной избыточности при проектировании ПО. Использование данного показателя позволяет уменьшить затраты на генерацию и выбор двухверсионного проекта путем оценки его характеристик еще на этапе проектирования. Каждый проект должен быть представлен в виде своей абстракции - графовой модели.

В данной работе будет рассмотрен показатель версионной избыточности с точки зрения внешней многоверсионности.

Для достижения поставленной цели необходимо решить следующие задачи:

- выявить типовые структуры графовых моделей, представляющих двухверсионный проект;
- предложить аналитические зависимости для определения количественных значений для показа-

телей качества двухверсионного проекта ПО;

- исследовать факторы, влияющие на показатель качества двухверсионного проекта ПО.

Решение поставленных задач дает возможность сформировать множество двухверсионных проектов, удовлетворяющих заданным показателям качества, что позволит уменьшить затраты на проектирование двухверсионных систем.

1. Оценка показателя качества двухверсионного проекта

В результате анализа графовых моделей двухверсионных проектов были выявлены типовые структуры трех видов, которые представлены на рис. 1. Каждую из этих структур далее будем называть соответственно: тип 1, тип 2 и тип 3 (рис. 1 а, б, в).

В графовой модели каждая вершина обозначает некоторый элемент решения, являющийся одним из вариантов технологии, используемой в данной версии проекта. Так, вершина h_0 может означать один из вариантов аппаратного обеспечения, вершина o_0 – одну из операционных систем, l_0 – один из языков программирования, используемого в проекте. Вершины d_0 и d_1 в графовой модели означают различные среды разработки ПО, а вершины m_0 и m_1 – различные виды ПО промежуточного уровня, которые используются в версиях проекта. Ребра графа означают совместимость различных элементов решений между собой.

Графовая модель типа 1 состоит из двух ветвей, не имеющих общих вершин. Графовая модель типа 2 состоит из двух ветвей, в которых вершины одного уровня совпадают. В графовой модели третьего типа совпадают лишь некоторые вершины, представленные в каждой версии.

Структура типа 1 (рис. 1а) соответствует двухверсионному проекту, в котором различные элементы решений (вершины графа) выбираются из аналогичных технологий (уровней проекта). Это могут быть различные аппаратные средства, разные операционные системы, разные языки программирования и т.д. Такой проект является композицией двух совершенно независимых элементов технологий, которые будут использовать неодинаковые элементы решений. Следует ожидать от такого проекта максимальной внешней диверсности при максимальных затратах на технологическую оснастку проекта.

Структура типа 2 (рис. 1б) соответствует двухверсионному проекту, в котором используются одинаковые элементы решений из однотипных технологий. В таком проекте требуемая задача решается на однотипных компьютерах, использующих одну и ту

же операционную систему, с использованием одинаковых языков программирования и т.д. двумя группами разработчиков. Такой проект позволяет получить минимальную внешнюю диверсность при минимальных затратах на технологическую оснастку проекта.

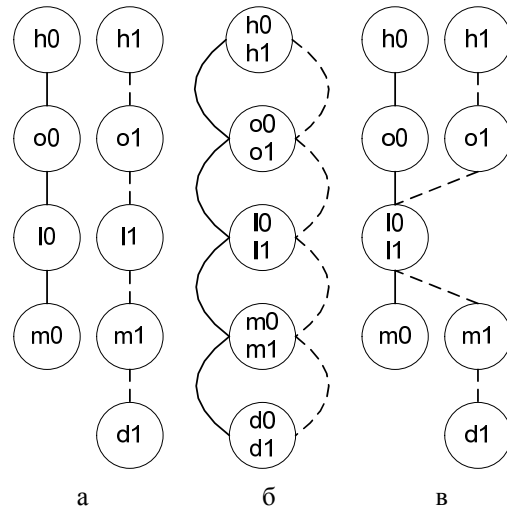


Рис. 1. Типовые структуры графовой модели двухверсионного проекта

Структура типа 3 (рис. 1в) соответствует промежуточному варианту двухверсионного проекта, использующему некоторые общие элементы решений в обеих версиях проекта. Общим элементом может быть, например, одинаковый язык программирования в обеих версиях проекта. Версионная избыточность этого проекта не может быть максимальной, но и стоимость обеспечения такой технологической цепочки не достигает максимума.

Будем вычислять показатель версионной избыточности, основываясь на следующих параметрах графовой модели:

- количестве версий проекта - M ;
- количестве уровней проекта - L (level);
- количестве вершин в каждой версии - n_i (nodes),
- количестве общих вершин в проектах - k ;
- весе вершины в версии проекта - p_i .

В приведенных структурах принято следующее допущение: для простоты все вершины каждой версии проекта имеют один и тот же индекс. Так, все вершины первой версии проекта имеют индекс ноль, а второго – единица.

На рис. 2 приведены графовые модели двух проектов, использующих две версии ($M=2$). Количество уровней L показывает число технологий, используемых в данном проекте. При этом элементы одной технологии, но принадлежащие разным версиям, должны располагаться на одном уровне, на-

пример m_0 и m_1 . В графовых моделях, приведенных на рис. 2, первый проект имеет шесть уровней ($L=6$) (рис. 2а), а второй – пять уровней ($L=5$) (рис. 2б).

Количество вершин в каждой версии проекта показывает число различных элементов технологий образующих каждую версию проекта. Так, в первом проекте (рис. 2а) первая версия содержит шесть вершин $h_0 \rightarrow o_0 \rightarrow l_0 \rightarrow m_0 \rightarrow d_0 \rightarrow s_0$, а вторая версия – пять $h_1 \rightarrow o_1 \rightarrow l_0 \rightarrow m_1 \rightarrow d_1$. Под перекрытием версий k будем понимать количество общих вершин в обеих версиях проекта, или, другими словами, в обеих ветвях графа. В проекте, представленном на рис. 2а, имеется одна общая вершина l_0 , поэтому $k=1$.

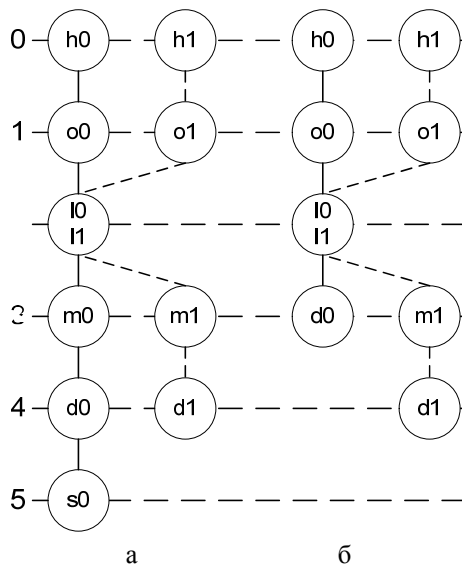


Рис. 2. Пример графовой модели для определения ее параметров

Вес вершины p показывает значимость элемента технологии (вершины) для данного проекта. В простейшем случае вес каждой вершины равен единице. Таким образом, для оценки показателя версии избыточности используются следующие параметры:

- для структуры типа 3 (рис. 2а): $M = 2$; $L = 6$; $n_0 = 6$; $n_1 = 5$; $k = 1$;

- для структуры типа 3 (рис. 2б): $M = 2$; $L = 5$; $n_0 = 4$; $n_1 = 5$; $k = 1$.

Показатель версии избыточности должен отражать степень разнообразия версий, входящих в двухверсионный проект. Для измерения показателя версии избыточности двухверсионного проекта введем понятие "показатель диверсности" ID , который показывает степень различия двух версий одного проекта.

Для оценки показателей версии избыточности проекта предлагается использовать аналитическую зависимость, учитывающую длину минимальной версии n_{min} . Показатель версии избыточности предлагается определять по формуле:

$$ID = (M - 1) + \frac{n_{min} - k}{n_{min}} \quad (1)$$

Формулу (1) можно преобразовать к другому виду:

$$ID = M - \frac{k}{n_{min}} \quad (2)$$

Используя формулу (1), рассчитаем показатель версии избыточности ID для проекта, представленного на рис. 2а.

$$ID_1 = (2 - 1) + \frac{5 - 1}{5} = 1,8.$$

Для проекта, представленного на рис. 2б, показатель версии избыточности ID будет иметь следующее значение:

$$ID_2 = (2 - 1) + \frac{4 - 1}{4} = 1,75.$$

Первый проект (рис. 2а) имеет большее значение показателя диверсности ($ID=1,8$), чем второй проект (рис. 2б), имеющий значение $ID=1,75$.

Таким образом, можно сделать вывод о том, что большее значение показателя версии избыточности ID свидетельствует о более высоком качестве первого проекта с точки зрения внешней многоверсионности.

2. Анализ показателя версии избыточности ID

Исследуем влияние величины перекрытия версий k на показатель версии избыточности $ID = f(k)$ в двухверсионном проекте.

Определим ограничения для данного случая. При значении k равно нулю имеем две полностью независимые версии, а максимальное значение величины перекрытия версий k не превосходит длины минимальной версии n_{min} ($0 \leq k \leq n_{min}$). Минимальная версия проекта должна содержать хотя бы одну вершину и не может превосходить длины максимальной версии ($1 \leq n_{min} \leq n_{max}$).

Определим значения показателя диверсности ID , которые характерны для предельных значений величины перекрытия версий k :

1) $k = 0$ - минимальное перекрытие версий, то есть две версии проекта полностью независимы.

Имеет место графовая модель типа 1 (рис. 1, а) и значение показателя диверсности, вычисленное по формуле (2), равно:

$$ID = M - \frac{k}{n_{min}} = M - \frac{0}{n_{min}} = M.$$

Показатель диверсности в этом случае является максимальным и не зависит от длины минимальной версии n_{min} :

$$k = 0, \forall n_{min}, ID = ID_{max} = M = 2.$$

2) $k = k_{\max} = n_{\min}$ - максимальное перекрытие версий, то есть проект содержит две полностью перекрывающиеся версии. Имеет место графовая модель типа 2 (рис. 1, б) и значение показателя диверсности, вычисленное в соответствии с формулой (2), равно:

$$ID = M - \frac{n_{\min}}{n_{\min}} = M - 1.$$

Показатель диверсности в этом случае является минимальным и также не зависит от длины минимальной версии n_{\min} :

$$k = n_{\min}, \forall n_{\min}, ID = ID_{\min} = M - 1 = 1.$$

Для данного случая более точно было бы сказать, что все вершины более короткой версии совпадают с некоторыми вершинами более длинной версии. При этом показатель версионной избыточности не зависит от длины минимальной версии. Этот факт иллюстрирует рис. 3, на котором приведены графовые модели двухверсионных проектов, удовлетворяющие условию k_{\max} . Из рис. 3 видно, что перекрытие версий максимальное, и, хотя длина минимальной версии проектов различна, показатель диверсности для всех трех графовых моделей будет одинаковым и иметь минимальное значение $ID_{\min} = 1$.

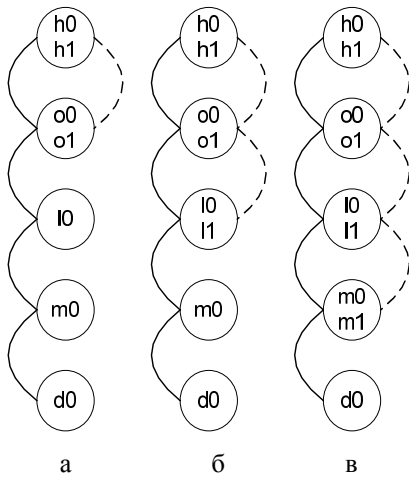


Рис. 3. Виды графовых моделей для диверсных проектов, имеющих максимальное перекрытие версий k

Проведенный анализ показывает, что для предельных значений величины перекрытия k показатель версионной избыточности *не зависит* от длины более короткой версии. Этот факт может считаться недостатком показателя диверсности ID , определяемого по формуле (1).

Полученная на основании (2) аналитическая зависимость позволяет сделать вывод о том, что максимальную диверсность будут иметь проекты, состоящие из двух независимых версий, а минимальную – проекты, состоящие из двух полностью

перекрывающихся версий. Эти данные полностью согласуются с практикой разработки двухверсионных проектов.

Такое поведение объясняется тем, что при использовании двух независимых версий общие ошибки в соответствующих элементах решений маловероятны и могут быть обнаружены и блокированы правильной работой другой версии. При использовании полностью перекрывающихся версий ошибки в технологической оснастке проекта могут быть общими для двух версий, что может привести к неверной работе проекта. Практика разработки двухверсионных проектов подтверждает эти факты.

Зависимость показателя диверсности от величины перекрытия проектов k для проектов с различной длиной минимальной версии n_{\min} приведено в табл. 1. На рис. 4 результаты, представленные в табл. 1, изображены в виде графиков. При построении графиков были приняты следующие значения параметров: $n_{\min} = 3 \dots 12$, $k = 0 \dots 12$.

Таблица 1

Зависимость показателя диверсности ID от величины перекрытия k для различных значений длины минимального проекта n_{\min}

	$n_{\min} = 3$	$n_{\min} = 5$	$n_{\min} = 7$	$n_{\min} = 10$	$n_{\min} = 12$
k	ID_0	ID_1	ID_2	ID_3	ID_4
0	2,00	2,00	2,00	2,00	2,00
1	1,67	1,80	1,86	1,90	1,92
2	1,33	1,60	1,71	1,80	1,83
3	1,00	1,40	1,57	1,70	1,75
4		1,20	1,43	1,60	1,67
5		1,00	1,29	1,50	1,58
6			1,14	1,40	1,50
7			1,00	1,30	1,42
8				1,20	1,33
9				1,10	1,25
10				1,00	1,17
11					1,08
12					1,00

Анализ графиков, представленных на рис. 4, подтверждает тот факт, что с увеличением числа общих вершин в проекте ($k \uparrow$) значение показателя диверсности *линейно уменьшается*. При этом *скорость уменьшения* показателя диверсности ID тем больше, чем меньше длина минимальной версии:

$$\Delta ID_1 / \Delta k > \Delta ID_2 / \Delta k, n_{1\min} < n_{2\min}.$$

Кроме того, изучение графиков (рис. 4) позволяет сделать следующие выводы:

- при заданной длине минимальной версии n_{\min} показатель диверсности ID будет больше у того проекта, который имеет меньшую величину перекрытия версий k :

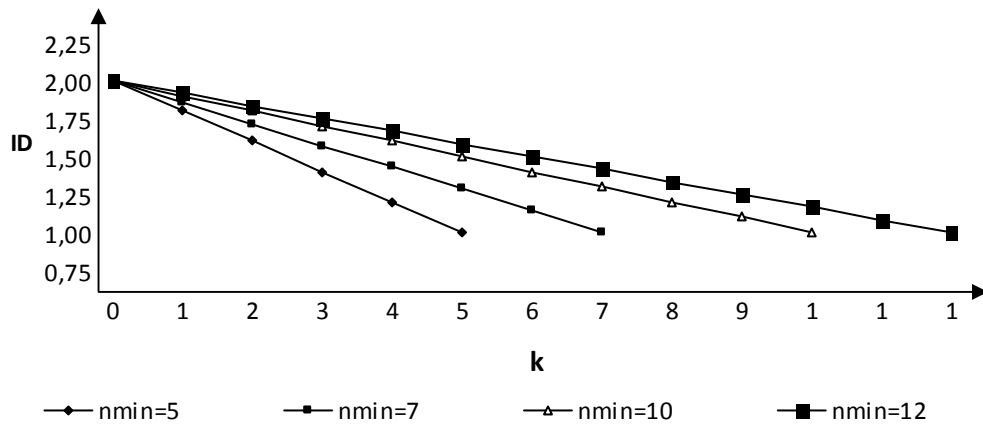


Рис. 4. График изменения показателя версионной избыточности ID в зависимости от количества общих вершин k

Таблица 2

Оценка показателя версионной избыточности диверсного проекта

№	Графовая модель	Показатель ID	№	Графовая модель	Показатель ID	№	Графовая модель	Показатель ID
1		2,0	3		1,6	5		1,2
2		1,8	4		1,4	6		1,0

- при заданной величине перекрытия k показатель диверсности ID будет больше у того проекта, который имеет большую длину минимальной версии n_{min} :

$$ID_1 > ID_2, k_1 < k_2, \forall n_{min}$$

$$ID_1 > ID_2, n_{min1} > n_{min2}, \forall k$$

В табл. 2 представлены различного вида графовые модели и показатели диверсности, вычисленные для них. Максимальное значение, равное 2, показатель диверсности ID принимает при отсутствии перекрытия ветвей проекта, т.е. при $k = 0$. Это означает, что обе версии проекта не имеют общих вершин (модель типа 1, рис. 1, а). В табл. 2 этот случай соответствует примеру 1. Минимальное значение, равное 1, показатель диверсности ID принимает при максимальном перекрытии ветвей проекта, т.е. при $k = k_{max}$. Это означает, что обе версии проекта совпадают (модель типа 2, рис. 1, б). В табл. 2 этот случай соответствует примеру 6.

Проведенный анализ показывает, что при наличии общих вершин в двух версиях проекта (модель

типа 3, рис. 1, в) показатель версионной избыточности ID *зависит* от длины более короткой версии и находится в диапазоне от единицы до двух. По графику (рис. 4) видно, что с увеличением числа общих вершин в обеих версиях проекта k значение показателя версионной избыточности ID линейно уменьшается с двух до единицы. В табл. 2 этот случай соответствует примерам 2-5, которые представляют графовые модели различного вида типа 3.

Проведенный анализ позволяет сделать вывод о том, перекрытие версий уменьшает показатель диверсности проекта ($ID < 2$). Это объясняется тем, что общие ошибки в элементах решений могут, хотя и не обязательно, оказать влияние на правильную работу проекта ПО. Этот факт также подтверждается практикой разработки двухверсионных проектов. Чтобы уменьшить зависимость показателя диверсности ID от величины перекрытия, следует увеличить длину более короткой версии проекта.

Предложенная в статье аналитическая зависимость (1) позволяет дать количественную оценку показателя диверсности, которая подтверждается

практикой проектирования диверсных систем, что позволяет сделать вывод о том, что данная формула адекватно описывает показатель диверсности ID.

Можно дать следующие рекомендации по практическому использованию полученных результатов для разработки двухверсионных проектов:

1. По возможности следует использовать две независимые версии в проекте, это позволит получить максимальную диверсность $ID = 2$. Против использования таких проектов говорит высокая стоимость их технологического обеспечения.

2. Следует избегать использования в проектах полностью перекрывающиеся версии, поскольку они имеют минимальную диверсность $ID = 1$. В пользу таких проектов, однако, говорит их невысокая стоимость.

3. Если все же избежать перекрытия версий в проекте нельзя, например, по экономическим соображениям, следует стремиться использовать проекты ПО, в которых более короткая версия имеет значительную длину. Это позволит снизить влияние общих вершин на уменьшение показателя диверсности ID в соответствии с формулой (1).

4. При использовании перекрывающихся версий в проекте, используя приведенную аналитическую зависимость, можно определить, какие элементы решений целесообразнее перекрывать, с учетом их влияния на показатель диверсности и на стоимостные характеристики проекта.

Заключение

В данной работе решена задача выбора и обоснования показателя версионной избыточности при проектировании ПО. Для решения этой задачи был разработан формальный метод оценки показателя диверсности ID двухверсионных проектов. Это позволяет уменьшить затраты на генерацию и выбор двухверсионного проекта путем оценки его характеристик на ранних этапах проектирования.

Получены следующие **научные результаты**:

- выполнен анализ структур двухверсионных проектов и выявлены три типа базовых структур графовой модели проектов ПО;

- описаны параметры графовых моделей двухверсионных проектов;

- предложен показатель версионной избыточности двухверсионных проектов ID, учитывающий длину минимальной версии;

- исследовано влияние величины перекрытия версий на показатель версионной избыточности.

Недостатком предложенного показателя диверсности является то, что при отсутствии перекрытия версий показатель диверсности не зависит от длины более короткой версии.

Практическим результатом являются рекомендации, которые можно использовать при разработке двухверсионных проектов ПО.

Дальнейшая модификация показателя диверсности проекта должна быть направлена на учет в показателе диверсности различной значимости вариантов решений версий, входящих в проект программного обеспечения.

Литература

1. Аїзенберг Я.Е. Сопоставление принципов обеспечения безопасности систем управления ракето-носителями и атомными электростанциями. / Я.Е. Аїзенберг, М.А. Ястребенецький // *Космічна наука та технологія*. – 2002. – № 1. – С. 55-60.

2. Littlewood B. Discussion of Practices for Enhancing Diversity in Software Designs / B. Littlewood, L.A. Strigini // *Technical report LS_DI_TR-04*. - version 1.1d. - 23 November, 2000. – 58 p.

3. Sklyar V. A Method of Multiversion Technologies Choice on Development of Fault-Tolerant Software Systems. / V. Sklyar, V. Kharchenko // *Proceeding of Workshop on Methods, Models and Tools for Fault Tolerance*. – Oxford, UK. – 3 July, 2007. – P. 148-157.

4. Головкин Б.А. Многовариантное программирование и его применение. / Б.А. Головкин // *Автоматика и телемеханика*. – М.: Наука, 1986. – № 7. – С. 5-39.

5. Preckshot G.G. Method for Performing Diversity and Defense-in-Depth Analyses of Reactor Protection Systems. / G.G. Preckshot. - Lawrence Livermore National Laboratory, California USA December, 1994. – 45 p.

6. Харченко В.С. Метрики диверсности: Классификация, анализ и применение для оценки надежности и безопасности компьютерных систем управления / В.С. Харченко, И.В. Пискачева, В.В. Скляр // *Открытые информационные и компьютерные интегрированные технологии*. – Х.: НАКУ «ХАИ», 2001. – № 9. – С. 194-214.

7. Дужий В.І. Розробка багатоверсійної ієрархії рішень при проектуванні програмного забезпечення / В.І. Дужий, І.В. Дужий, А.В. Шостак // *Радіоелектронні і комп'ютерні системи*. - 2007. - № 8 (27). – С. 173-176.

8. Сиора А.А. (n,m)-версионные системы: таксономия, модели и технологии / А.А. Сиора, В.В. Скляр, В.С. Харченко // *Вісник Харківського національного університету*. – 2008. – № 833. – С. 231-246

9. Скляр В.В., Харченко В.С. Анализ и гармонизация регулирующих требований к программному обеспечению информационных и управляющих систем АЭС с учетом изменений международной нормативной базы/ В.В. Скляр, В.С. Харченко // *Ядерная и радиационная безопасность*. - 2004. – Т. 7. – № 4. – С. 34-47.

Поступила в редакцію 12.02.2009

Рецензент: д-р техн. наук, проф., зав. кафедрой компьютерных систем и сетей В.С. Харченко, Харьковский национальный аэрокосмический университет им. Н.Е. Жуковского, Харьков, Украина.

ОЦІНКА ПОКАЗНИКА ВЕРСІЙНОЇ НАДЛИШКОВОСТІ ПРИ ПРОЕКТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В.І. Дужий, А.В. Шостак, І.В. Дужий

Описані типові структури графової моделі двоверсійного проекту та їх параметри. Запропонована аналітична залежність для показника версійної надлишковості, яка враховує довжину найменшої версії проекту. Виконано дослідження впливу на показник диверсності величини перекриття версій проекту. Відзначено, що збільшення перекриття версій проекту показник диверсності лінійно зменшується. Наведені приклади графових моделей двоверсійних проектів та визначені показники диверсності для них. Надані практичні рекомендації для розробки двоверсійних проектів.

Ключові слова: двоверсійний проект, графова модель, показник диверсності.

EVALUATION OF VERSION REDUNDANCY INDEX DURING SOFTWARE DESIGN PROCESS

V.I. Duzhyi, A.V. Shostak, I.V. Duzhyi

The paper describes type structures of the graph model of a two-version project and their parameters. As a result of the research, analytical dependency for the version redundancy has been introduced, which takes into account the minimal project version duration. It has additionally been studied how the value of project version overlapping affects the diversity index. It has been noted that with the increase of the project version overlapping the diversity index shows a linear decrease. The research uses examples of graph models of two-version projects and determines diversity indexes for them. The practical recommendations for development of two-version projects are formulated.

Key-words: two-version project, graph model, diversity index.

Дужий Вячеслав Игоревич – старший преподаватель кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: V.Duzhy@csac.khai.edu.

Шостак Анатолий Васильевич – канд. техн. наук, доцент, доцент кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: A.Shostak@csac.khai.edu.

Дужий Игорь Вячеславович – студент 4 курса факультета радиотехнических систем летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина.