

УДК 621.391.7.001

С.Г. АНТОЩУК<sup>1</sup>, Д.А. МАЕВСКИЙ<sup>1</sup>, С.А. ЯРЕМЧУК<sup>2</sup><sup>1</sup>Одесский национальный политехнический университет, Украина<sup>2</sup>Измаильский институт водного транспорта, Украина

## ПРОГНОЗИРОВАНИЕ КОЛИЧЕСТВА ОШИБОК НА ЭТАПЕ ЭКСПЛУАТАЦИИ АДАПТИРУЕМЫХ УЧЕТНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

*В настоящей статье предложен математический метод для прогнозирования количества ошибок в программном обеспечении адаптируемых информационных систем и времени эксплуатации, в течение которого эти ошибки могут быть выявлены. Прогнозирование осуществляется на основании данных по динамике обнаружения ошибок в течение конечного времени эксплуатации. Построен прогнозируемый график зависимости количества найденных ошибок программы от времени. На основе экспериментальных данных об эксплуатации конкретной учетной информационной системы выполнены практические расчеты и анализ полученных результатов. Сделаны выводы о возможности применения данного математического метода для прогнозирования показателей надежности учетных информационных систем.*

**Ключевые слова:** надежность систем, безопасность программ, программные ошибки, программные дефекты, классификация ошибок, транзакции, сложность программ, количество ошибок, среднее время наработки на ошибку.

### Введение

В настоящее время учетные информационные системы (УИС) представляют собой наиболее многочисленный класс информационных систем, используемых в народном хозяйстве. Они используются на предприятиях всех форм собственности и призваны накапливать, обрабатывать и анализировать информацию о финансовом состоянии предприятий. Сама необходимость использования УИС, а также алгоритмы их функционирования напрямую обусловлены действующим законодательством Украины [1]. Многочисленные законодательные и подзаконные акты определяют, как именно выполняемые предприятием хозяйственные операции должны отображаться в учетных регистрах информационной базы УИС.

Основной чертой законодательства Украины в настоящее время является его динамическое изменение [2]. О динамичности изменений законодательной базы свидетельствует, например, следующее. За период из 1991 – 2009 гг., Закон Украины «О системе налогообложения» изменялся 38 раз, Закон Украины «О налоге на добавленную стоимость» – 112 раз, Закон Украины «Об акцизном сборе» – 72 раза, Закон Украины «О налоге на прибыль предприятий» – 124 раза. Некоторые положения действующих правовых актов изменяются на основании других. Так, например, Закон Украины о Государственном бюджете Украины, который прини-

мается в конце каждого года на следующий год, вносит изменения и коррективы в несколько десятков других законов, которые регламентируют ведение бухгалтерского и налогового учета. Данным законом, в частности, корректируются размеры минимальной заработной платы и прожиточного минимума, от которых в свою очередь зависят размеры отчислений в социальные фонды и размеры налоговых льгот.

Задача построения УИС, способных к адаптации при изменении нормативной базы является сложной и слабо формализованной. Для ее решения необходимо обеспечить гибкость алгоритмов функционирования при работе УИС. В идеале, большинство таких алгоритмов функционирования вообще не должно быть жестко прописано в программном коде системы, а пользователь должен иметь возможность легко и оперативно изменять эти алгоритмы, изменяя параметры ее настройки.

В связи с этим автоматически возникает проблема обеспечения надежности и повышения уровня безопасности программного обеспечения УИС на этапе их функционирования.

Под надежностью программного обеспечения (ПО) понимается его способность безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью [3].

Под уровнем безопасности понимается вероятность того, что при заданных условиях в процессе

експлуатації УИС буде отримано функціонально придатний результат. В [4] відмічено, що причини, що призводять до функціонально непридатного результату, можуть бути різними: збої комп'ютерних систем, помилки програмістів і користувачів, помилки в програмному забезпеченні (ПО) УИС. При цьому помилки можуть бути двох типів: передбачені і непередбачені. Перші є результатом зловмисних дій, другі – помилкових дій людини.

Окремою проблемою, що виникає на етапі експлуатації адаптованих УИС, є проблема помилок, які вносяться користувачем в уже налаштовану програмну середовище. Помилки, що виникають в результаті некоректних дій користувача, виходять на перший план і надають вирішальний вплив на рівень безпеки системи. Тому питання класифікації таких помилок і розробка методів прогнозування їх виявлення при експлуатації УИС є актуальними.

## 1. Визначення і класифікація помилок на етапі експлуатації УИС

Життєвий цикл будь-якої інформаційної системи можна розбити на три етапи:

- етап проектування і розробки;
- етап тестування;
- етап експлуатації.

Програмні помилки в неадаптованих системах можуть бути внесені на першому і другому етапі, навіть не зважаючи на те, що другим етапом як раз і передбачено їх виправлення.

Програмний код систем, здатних адаптуватися, змінюється також і на етапі експлуатації. Це призводить до високої ймовірності внесення додаткових помилок в зв'язі з тим, що кваліфікація користувачів, що експлуатують систему, як правило, значно нижче, ніж кваліфікація розробників. Тому адаптовані УИС найбільше піддані негативному впливу програмних помилок.

В літературі можна зустріти цілий ряд синонімів поняття «програмна помилка». Крім власне терміна «помилка» (error), досить часто зустрічаються такі терміни як «проблема» (problem), «дефект» (defect), «неисправність» (fault).

Г. Майерс [5] пропонує наступне визначення «Якщо програма не робить того, чого користувач від неї повністю очікує, значить наявна програмна помилка».

Однак слід відзначити, що визначення помилок як розходження між програмою і її

специфікацією визнається не всіма дослідниками. Так, С. Канер виходить з того, що навіть точно відповідна специфікації програма містить помилки в тому випадку, якщо є помилки і в самій специфікації [6]. А в роботі [7] Б. Бейзером дається таке визначення програмних помилок: "... програмна помилка - це не щось інше, ніж помилка в розробці програмного продукту, який викликає невідповідність очікуваних результатів виконання програмного продукту і фактично отриманих результатів. Дефект може виникнути на стадії кодування, на стадії формулювання вимог або на стадії проектування, причому причиною може бути некоректна конфігурація або дані. Дефектом може бути також щось інше, що не відповідає очікуванням замовника і що може бути, а може і не бути визначено в специфікації програмного продукту".

Б. Бейзер справедливо вказує на суб'єктивний характер програмних помилок: «Не існує ні абсолютного визначення помилок, ні точного критерію наявності їх в програмі. Можливо лише сказати, наскільки програма не справляється зі своєю задачею, - це виключно суб'єктивна характеристика» [7].

На наш погляд, в цьому питанні слід чітко розмежувати причину – від чого виникають програмні помилки, і наслідок – сам факт виникнення таких помилок. Таке розмежування відкриває два напрями дослідження.

Перший напрямок пов'язаний з розробкою методів і підходів до розробки і налагодки ПО УИС, які зводять до мінімуму кількість причин появи помилок і самих помилок.

Другий напрямок пов'язаний з прогнозуванням кількості помилок в уже створених системах, дослідженням динаміки їх внесення в процес адаптації і вивчення наслідків їх появи. При цьому помилка представляє собою вже ставший фактом, причиною якого не є суттєва.

Метою поточної роботи є розробка моделі прогнозування кількості помилок на етапі експлуатації УИС.

Виходячи з цього, будемо вважати, що програмна помилка – це дефект, який не дозволяє користувачеві повністю або частково виконувати всі або деякі функції системи.

Програмна помилка на етапі експлуатації УИС призводить до порушення її спроможності в цілому або певної функції в частині. З точки зору джерел виникнення і факту виявлення можна запропонувати класифікацію програмних помилок, показану на рис. 1.

При класифікації за джерелом слід розрізняти помилки, внесені в програмний код сис-

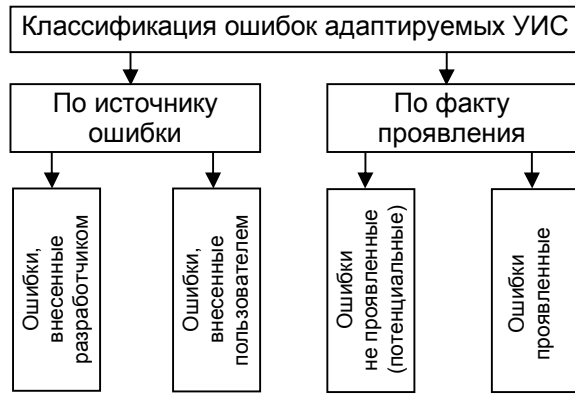


Рис. 1. Классификация ошибок адаптируемых УИС

темы вследствие действий разработчика и вследствие действий пользователя. Вторая группа ошибок присуща только УИС, способным к адаптации, программный код которых тем или иным способом изменяется пользователем. Следует отметить, что не любое, пусть даже некорректное действие пользователя или разработчика приводит к возникновению ошибки. Во-первых, такие некорректные действия могут скомпенсировать друг друга а, во-вторых, одна ошибка может не проявиться потому, что вследствие другой ошибки система никогда не будет выполнять соответствующую часть программного кода.

Как видим, некорректное действие разработчика или пользователя само по себе еще не приводит к ошибке. В этом плане можно считать, что в программный код вносятся не ошибки, а только возможность их проявления. Здесь мы приходим к необходимости классифицировать ошибки по факту их проявления – на не проявленные (потенциальные) и проявленные, то есть уже обнаруженные на этапе эксплуатации.

С понятием потенциальных ошибок тесно связано понятие надежности работы ПО. Степень надежности ПО характеризуется вероятностью работы программного продукта без отказа в течение определенного периода времени. Понятно, что количество потенциальных ошибок напрямую связано с надежностью. Поэтому актуальной является проблема прогнозирования количества потенциальных ошибок в ПО УИС.

## 2. Прогнозирование показателей надежности программного обеспечения УИС

Разные авторы вкладывают разный смысл в понятие надежности программного обеспечения.

Для оценки надежности ПО используется показатель среднего времени наработки на ошибку, т.е. время, в течение которого с большой степенью вероятности будет выявлена хотя бы одна ошибка [8].

В работе [9] предложено характеризовать надежность ПО показателем

$$R = 1 - \frac{k}{n},$$

где  $n$  – общее число запусков программы,  $k$  – число запусков, которые закончились с ошибочным результатом. Здесь под запуском программы понимается ее выполнение от начала и до завершения.

Такой показатель не может характеризовать надежность ПО УИС, так как, во-первых, в нем трудно выделить одну единственную программу, выполняющуюся для обработки запроса пользователя, а, во-вторых, даже одну и ту же программу можно запускать неограниченное количество раз с одними и теми же исходными данными. Если при этом она все время будет выдавать правильный результат, то это не означает, что показатель  $R = 1$ , так как при другом наборе данных результат всегда может быть неправильным.

Согласно [10], наиболее естественной характеристикой надежности можно считать количество ошибок в ПО. В терминах предложенной классификации следует здесь понимать именно потенциальные ошибки, так как уже проявленные исправляются и в дальнейшем на надежность оказывать влияния не могут. Точно оценить этот показатель зачастую не представляется возможным, так как количество потенциальных ошибок невозможно подсчитать напрямую. Даже после продолжительного безотказного периода работы ПО может быть выявлена новая ошибка.

Однако, количество потенциальных ошибок в ПО УИС можно оценить приблизительно. Основой для такой оценки может быть динамика изменения скорости выявления ошибок в процессе эксплуатации УИС. Скорость выявления ошибок (количество ошибок в единицу времени) зависит от двух факторов: количества ошибок и интенсивности работы с системой. Если допустить, что УИС в период своего жизненного цикла эксплуатируется с одинаковой интенсивностью, то скорость выявления ошибок должна уменьшаться с течением времени. По динамике ее изменения можно косвенно судить о количестве ошибок в момент начала эксплуатации УИС и о количестве оставшихся необнаруженными ошибок.

Для характеристики интенсивности работы пользователя с УИС используем понятие «транзакция». Под транзакцией в данном контексте будем

понимать выполнение системой конкретного запроса пользователя с конкретным набором исходных данных. Каждая транзакция может быть либо выполнена корректно, либо в процессе ее выполнения может быть обнаружена ошибка.

Примем следующие допущения:

1. Частота транзакций (количество транзакций в единицу времени) одинакова в течение всего периода эксплуатации УИС. Обозначим частоту транзакций как  $f$ .

2. Обнаруженная во время транзакции ошибка исправляется до начала следующей транзакции.

3. Во время исправления ошибок новые ошибки не вносятся. Это допущение позволяет создать идеальную модель выявления ошибок на этапе эксплуатации УИС, на базе которой возможно построение реальной модели, в которой такое допущение снято.

Обозначим:

$R_0$  – количество потенциальных ошибок в ПО в начале эксплуатации (неизвестная величина);

$x$  – количество выполненных транзакций;

$r(x)$  – текущее количество потенциальных ошибок, оставшееся в ПО после выполнения транзакции с номером  $x$ ;

$S$  – общее количество мест в ПО, в которых теоретически может быть допущена ошибка. В качестве значения этого показателя может быть принято, например, количество операторов во всех программных текстах системы. Этот показатель будем называть сложностью ПО.

Скорость выявления ошибок  $\frac{dr}{dx}$  будет прямо пропорциональна числу имеющихся в ПО к моменту транзакции  $x$  ошибок ( $r$ ) и частоты транзакций  $f$ . С ростом сложности системы скорость выявления ошибок будет уменьшаться, поэтому ее зависимость от  $S$  будет обратно пропорциональной.

С учетом того, что с ростом числа транзакций  $X$  число оставшихся ошибок  $r$  уменьшается, скорость их выявления будет отрицательной.

На основании сказанного выше можно составить следующее уравнение:

$$-\frac{dr}{dx} = \frac{f}{S} \cdot r,$$

или

$$S \cdot \frac{dr}{dx} + f \cdot r = 0 \quad (1)$$

Полученное уравнение (1) является линейным однородным дифференциальным уравнением первого порядка. Его решение в общем виде можно записать как:

$$r(x) = A \cdot e^{\lambda \cdot x}, \quad (2)$$

где  $A$  – неизвестная постоянная интегрирования, а  $\lambda$  – корень характеристического уравнения:

$$\lambda \cdot S + f = 0,$$

откуда:

$$\lambda = -\frac{f}{S}.$$

Постоянную интегрирования  $A$  найдем из уравнения (2), подставив  $x = 0$ :

$$A = r(0) = R_0.$$

Таким образом, количество ошибок, оставшихся в ПО на момент транзакции  $x$ , равно:

$$r(x) = R_0 \cdot e^{-\frac{f}{S} \cdot x}. \quad (3)$$

Если через  $q$  обозначить количество найденных к моменту транзакции  $x$  ошибок, то получим:

$$q(x) = R_0 \cdot \left(1 - e^{-\frac{f}{S} \cdot x}\right). \quad (4)$$

Из (4) следует, что количество ошибок, которое имело место в ПО на начало эксплуатации, может быть определено как:

$$R_0 = \frac{q(x)}{1 - e^{-\frac{f}{S} \cdot x}}. \quad (5)$$

Для вычислений по формуле (5) необходимо знать отношение  $\frac{f}{S}$ .

Его прямое вычисление может быть затруднено, так как зачастую частоту транзакций  $f$  и сложность системы  $S$  точно определить невозможно.

Однако, благодаря свойствам экспоненциальной функции, построив по экспериментальным данным на каком-либо ограниченном отрезке график зависимости  $r(x)$ , можно получить отношение  $\frac{f}{S}$  путем несложных графических построений. Гра-

фик функции (4) показан на рис. 2.

Пусть на основании экспериментальных данных известна зависимость количества найденных ошибок от числа транзакций, выраженная в табличной форме.

На основании данных этой таблицы можно построить участок экспоненты в диапазоне, например, от  $x_1$  до  $x_2$ .

Проведя в любой точке этого участка касательную к экспоненте и отрезка от проекции точки касания до пересечения касательной с асимптотой  $q(x) = R_0$ , получим так называемую длину «подкасательной».

Из курса математического анализа [11] известно, что длина подкасательной к экспоненте одинакова для любой точки касания и равна

$$\Delta x = \frac{1}{\left| -\frac{f}{S} \right|} = \frac{S}{f}.$$

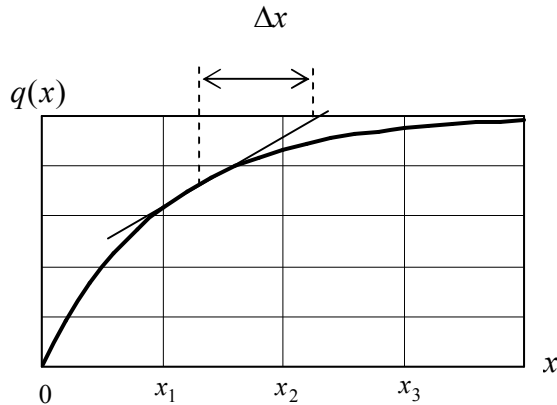


Рис. 2. Зависимость количества найденных ошибок от количества транзакций

Длина подкасательной  $\Delta x$  измеряется в количестве транзакций, которое, зная частоту транзакций  $f$ , можно выразить в единицах времени. Поэтому назовем величину  $\Delta x$  постоянной времени отладки. Физически эта постоянная соответствует времени отладки, в течение которого количество ошибок в ПО уменьшается в  $e$  раз.

Зная величину постоянной времени можно получить отношение  $\frac{f}{S}$  и по формуле (5) рассчитать значение  $R_0$ .

Среднее время наработки на ошибку  $t_1$  получим из выражения (4). Из него находим такое значение  $x_1$ , при котором  $q(x) = 1$ . С учетом того, что  $t_1 = f \cdot x_1$ , имеем:

$$t_1 = -\ln\left(1 - \frac{1}{R_0}\right) \cdot f \cdot \Delta x. \quad (6)$$

### 3. Анализ полученных результатов

В качестве примера определим показатели  $R_0$ ,  $\Delta x$  и  $t_1$  для разработанной с участием авторов УИС учета платежей населения за природный газ. УИС эксплуатируется в малом государственном коммунальном предприятии «Лайф» г. Измаила с 1998 года по настоящее время.

Система выполняет функции учета и анализа платежей, работы с должниками, электронного об-

мена с ПО в государственных структурах – отделом субсидий, Сбербанком, УСЗН, УЖКХ. УИС работает в сетевом режиме (6 рабочих мест), общее количество транзакций в месяц равно 12000. Данные по количеству обнаруженных за первый год эксплуатации ошибок приведены в табл. 1.

По экспериментальным данным таблицы 1 построен график зависимости количества найденных ошибок от количества транзакций  $x$ , показанный на рис. 3.

По графику определена постоянная времени отладки, составляющая 55 200 транзакций, что при указанной частоте транзакций соответствует периоду в 4,6 месяца.

Таблица 1.

Экспериментальные данные

Месяц	Количество выявленных ошибок	
	За тек. месяц	С начала эксплуатации
1	13	13
2	11	24
3	8	32
4	7	39
5	6	45
6	5	50
7	4	54
8	6	60
9	3	63
10	2	65
11	1	66
12	1	67



Рис. 3. Зависимость количества найденных ошибок от количества транзакций по экспериментальным данным

Значение  $R_0$  составило 72 ошибки. Таким образом, было определено количество ошибок в УИС на начало эксплуатации, что с учетом уже обнаруженных 67 ошибок позволяет сделать вывод о количестве еще не выявленных ошибок после годичной эксплуатации.

В итоге мы прогнозируем оставшиеся невыявленными программные ошибки в количестве 5 штук.

Этот показатель надежности вполне реалистичен и подтверждается последующим практическим опытом эксплуатации системы.

Среднее время наработки на ошибку  $t_1$  составило 768 транзакций. Исходя из 12 000 транзакций в месяц на 6-ти рабочих местах на одном рабочем месте за месяц (20 рабочих дней) выполняется 100 транзакций.

Следовательно, первая ошибка в системе будет обнаружена на одном из рабочих мест на восьмой день работы системы.

На практике сообщение о первой найденной ошибке поступило через шесть рабочих дней после начала эксплуатации системы, что хорошо согласуется с полученным теоретическим результатом.

### Заключение

В статье предложена прогнозная модель определения параметров надежности ПО УИС, основывающаяся на динамике выявления ошибок в процессе эксплуатации системы. Ее достоинством является то, что оценка показателей может быть выполнена на основе анализа проявления ошибок в течение небольшого временного периода.

На основе полученных за этот период экспериментальных данных можно оценить такие показатели надежности, как:

- первоначальное количество ошибок в системе. Зная эту характеристику можно судить об эффективности принятых методик тестирования и профессиональных качествах разработчиков;

- количество ошибок, оставшихся невыявленными на данный момент эксплуатации. Эта характеристика позволяет принимать решение о целесообразности дальнейшей эксплуатации системы и оптимальным образом организовать работу с ней;

- время выявления практически всех ошибок позволяет оценить материальные затраты на эксплуатацию системы.

Показано соответствие разработанной модели экспериментальным данным, что позволяет сделать вывод о ее применимости для прогнозирования показателей надежности адаптируемых учетных информационных систем.

### Литература

1. Маевский Д.А. Основные принципы построения учетных систем операционного управления / Д.А. Маевский, Е.Ю. Маевская // *Электромашиностр. и электрооборудов.* – 2008. – Вып. 70. – С. 123-126.

2. Маевский Д.А. Адаптація функціонування облікових інформаційних систем до вимог нормативно-правових актів / Д.А.Маєвський, О.Ю.Маєвська, В.М.Антощук // *Сб. Электромашиностр. и электрооборудов.* – 2009. – Вып. 72. – С. 153-160.

3. Промыслов В.Г. Оценка надежности программного обеспечения на различных этапах жизненного цикла различных программ / В.Г. Промыслов, А.В. Антонов, С.И. Масолкин, А.С. Степаняну // *Труды V международной конференции «Идентификация систем и задачи управления» SCIPRO'06.* Москва, 30 января – 2 февраля 2006 г. – М.: Институт проблем управления им. В.А. Трапезникова РАН. – 2006. – С. 1300-1304.

4. Казарин О.В. Безопасность программного обеспечения компьютерных систем / О.В. Казарин. – М: МГУЛ, 2003. – 212 с.

5. Майерс Г. Искусство тестирования программ / Г. Майерс. – М.: Финансы и статистика, 1982. – 174 с.

6. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ./ С. Канер, Дж. Фолк, Е.К. Нгуен. — К.: ДиаСофт, 2001. – 544 с.

7. Beizer V. *Software System Testing and Quality Assurance*, Van Nostrand Reinhold, New York, New York, 1984.

8. Смагин В.А. Моделирование и обеспечение надёжности программных средств АСУ. / В.А. Смагин, В.С. Солдатенко, В.В. Кузнецов - СПб.: ВИКУ им. А.Ф. Можайского, 1999. – 49 с.

9. Романюк С.Г. Оценка надежности программного обеспечения / С.Г. Романюк // *Открытые системы.* – 1994. – № 4. – С. 95-104.

10. Майерс Г. Надежность программного обеспечения. / Г. Майерс. – М.: Мир, 1980. – 236 с.

11. Решетняк Ю.Г. Курс математического анализа: Ч. I., Кн. I. Введение в математический анализ. Дифференциальное исчисление функций одной переменной. / Ю. Г. Решетняк. – Новосибирск: Ин-т математики, 1999. – 454 с.

Поступила в редакцию 11.01.2010

**Рецензент:** д-р техн. наук, проф. О.В. Дрозд, заведуючий кафедрой СПО и ТДО Одесского национального университета им. Мечникова, Одесса.

### ПРОГНОЗУВАННЯ КІЛЬКОСТІ ПОМИЛОК НА ЕТАПІ ЕКСПЛУАТАЦІЇ АДАПТУЄМИХ ОБЛІКОВИХ ІНФОРМАЦІЙНИХ СИСТЕМ

*С.Г. Антощук, Д.А. Маєвський, С.О. Яремчук*

В статті запропоновано математичний метод для прогнозування кількості помилок в програмному забезпеченні інформаційних систем, що адаптуються, і часу експлуатації, протягом якого ці помилки можуть бути виявлені. Прогнозування здійснюється на підставі даних по динаміці виявлення помилок протягом кінцевого часу експлуатації. Побудовано прогнозований графік залежності кількості знайдених помилок програми від часу. На підставі експериментальних даних про експлуатацію конкретної облікової інформаційної системи виконані практичні розрахунки і аналіз отриманих результатів. Зроблено висновки про можливість застосування даного математичного методу для прогнозування показників надійності облікових інформаційних систем.

**Ключові слова:** надійність систем, безпека програм, програмні помилки, програмні дефекти, класифікація помилок, транзакції, складність програм, кількість помилок, середній час напрацювання на помилку.

### ERRORS AMOUNT PROGNOSTICATION ON THE STAGE OF EXPLOITATION OF THE ADAPTED ACCOUNTING INFORMATION SYSTEMS

*S.G. Antoschuk, D.A. Maevsky, S.A. Yaremchuk*

In this article a mathematical method is offered for prognostication of amount of errors in software of the adapted informative systems and time of exploitation, which these errors can be exposed during. Prognostication is carried out on the basis of information on the dynamics of detection of errors during eventual time of exploitation. The forecast chart of dependence of amount of the found errors of the program is built from time. On the basis of experimental information about exploitation of the concrete registration informative system practical calculations and analysis of the got results are executed. Conclusions are done about possibility of application of this mathematical method for prognostication of reliability of the registration informative systems indexes.

**Keywords:** reliability of the systems, safety of the programs, software failures, programmatic defects, classification of errors, transactions, complication of the programs, amount of errors, mean time of work on an error.

**Антощук Светлана Григорьевна** – д-р техн. наук, проф., директор института компьютерных систем Одесского национального политехнического университета, Одеса, Украина, e-mail: svetlana\_onpu@mail.ru

**Маєвський Дмитрій Андреевич** – канд. техн. наук, доцент, заведуючий кафедрой теоретических основ и общей электротехники Одесского национального политехнического университета, г. Одеса, Украина. e-mail: toe-onpu@ukr.net

**Яремчук Светлана Александровна** - старший преподаватель кафедры информационных управляющих систем и технологий Измаильского института водного транспорта, Измаил, Украина, e-mail: svetlana397@yandex.ru