

Длина	Количество слов	Длина	Количество слов	Длина	Количество слов	Длина	Количество слов
0	1	8	4 476	16	10 401 496	24	1 000 867 498
1	4	9	11 896	17	26 831 306	25	130 718 312
2	12	10	31 528	18	68 290 046	26	1 353 842
3	32	11	83 508	19	169 729 186	27	3 454
4	88	12	221 108	20	403 331 722	28	714
5	236	13	582 828	21	873 779 504	29	146
6	632	14	1 529 508	22	1 558 150 656	30	12
7	1 688	15	3 998 452	23	1 853 591 734	–	–
$ B_0(2,5,6) = 5^{14} = 6\,103\,515\,625$							

Группа $B_0(2,5,5)$. Аналогичным образом вычислим функцию роста группы $B_0(2,5,5)$ (табл. 5).

Справедлива следующая теорема.

Теорема 5. В алфавите A группа $B_0(2,5,5)$ имеет диаметр Кэли, равный 20, а функция ее роста задана табл. 5.

Доказательство теоремы следует из вычислений функции роста группы $B_0(2,5,5)$ по алгоритму перечисления элементов группы.

Группа $B_0(2,5,6)$. Функция роста группы $B_0(2,5,6)$ вычислена в работе [2] (табл. 6).

Библиографические ссылки

1. Hawas G., Wall G., Wamsley J. The Two Generated Burnside Group of Exponent Five // Bull. Austral. Math. Soc. 1974. № 10. P. 459–470.
2. Sims C. Fast Multiplication and Growth in Groups // Proc. of the 1998 Intern. Symposium on Symbolic and Algebraic Computation. Rostock, Germany, 1998. P. 165–170.
3. Sims C. Computation with Finitely Presented Groups. Cambridge : Cambridge Univ. Press, 1994.
4. Кузнецов А. А., Кузнецова А. С. Быстрое умножение элементов в конечных двупорожденных группах периода пять // Прикл. дискрет. математика. 2012. № 4 (17). С. 54–60.

A. A. Kuznetsov, A. S. Kuznetsova

COMPUTER MODELING OF FINITE TWO-GENERATOR GROUPS OF EXPONENT 5

Let $B_0(2,5,k)$ be the largest finite, provided by two, Burnside group of period of the 5^{th} step of nilpotency k , and $\{a_1, a_2\}$ be generators of this group. In this article the growth functions of the groups $B_0(2,5,k)$, relative to $A = \{a_1, a_1^{-1}, a_2, a_2^{-1}\}$, are calculated for cases $k = 1, 2, 3, 4, 5$.

Keywords: Burnside groups, growth function, Cayley diameter.

© Кузнецов А. А., Кузнецова А. С., 2012

УДК 004.4

В. В. Кукарцев, Д. А. Шеенок

ОЦЕНКА ЗАТРАТ НА МОДЕРНИЗАЦИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КРИТИЧЕСКИХ ПО НАДЕЖНОСТИ СИСТЕМ

Предложен подход для оценки затрат на модернизацию программного обеспечения с учетом различных этапов жизненного цикла и введения программной избыточности. Проведены расчеты затрат на модернизацию программного обеспечения реальной системы.

Ключевые слова: программная избыточность, трудозатраты, программное обеспечение, декомпозиция.

При разработке высокобюджетных программных проектов важную роль играет оценка затрат. Расхождение планируемых затрат с фактическими может привести к серьезным финансовым потерям и даже

банкротству компании-разработчика. Существует множество простых и сложных моделей и основанных на них методик оценки будущей стоимости проекта, но они в совокупности имеют следующие недостатки.

Во-первых, при оценке трудозатрат не различаются явным образом затраты персонала компании-разработчика на этапах проектирования, разработки, тестирования, документирования, а также затраты на руководителей, на организацию всего жизненного цикла проекта.

Во-вторых, расчет трудозатрат базируется на оценке объема кода программного продукта, что при современных технологиях проектирования и разработки не совсем адекватно отражает трудоемкость технологического процесса разработки.

В-третьих, не учитывается явным образом трудоемкость модернизации части системы, что заставляет рассматривать каждую новую доработку системы как отдельный новый проект. Современные сложные программные комплексы имеют большую иерархическую программную архитектуру, и доработка одного программного компонента обязательно влечет за собой изменения в других, связанных с ним компонентах. Поэтому при модернизации системы должна учитываться глубина доработки.

В-четвертых, информация, необходимая для расчета многих параметров, может отсутствовать на ранних этапах проекта, а их предполагаемая оценка может быть ошибочной.

В-пятых, в существующих моделях не учитываются затраты на повышение надежности наиболее важных компонентов путем введения программной избыточности. Снижение вероятности сбоя получают за счет одновременной работы нескольких версий одного и того же компонента, реализованных разными командами разработчиков или с использованием разных языков программирования. Таким образом снижается вероятность того, что на одних и тех же входных данных откажут все версии компонента. Для одновременной работы нескольких версий необходима разработка среды их исполнения. В среду исполнения входит алгоритм голосования, который выбирает наиболее приемлемые выходные данные от всех версий [1]. Введение программной избыточности необходимо для обеспечения безопасности функционирования критически важных информационных систем, связанных с управлением технологическими процессами или финансовыми расчетами.

Следовательно, необходима разработка методики оценки трудозатрат на модернизацию уже существующего программного продукта, которая поможет менеджерам проектов уменьшить ошибку оценки и снизить финансовые риски компании.

При расчете размера оцениваемой системы будут учитываться объектные указатели (компоненты) системы, аналогично модели СОМОМО II [2].

Рассмотрим основные этапы методики оценки затрат на модернизацию программного обеспечения критических по надежности систем.

1. *Анализ технических требований заказчика.* Заказчик передает свои задокументированные требования на доработку системы для анализа компании разработчика или аналитики выявляют требования заказ-

чика в ходе интервью. После проведенного анализа формируется проект технического задания на модернизацию системы.

2. *Декомпозиция задач.* На этапе проектирования архитектуры системы задачи модернизации декомпозируются на множество простых. Затем задачи сопоставляются с доработкой уже существующих компонентов системы или разработкой новых.

3. *Группировка компонентов.* Компоненты, которые необходимо разработать или модернизировать, группируются на типы по аналогичному назначению и сложности.

4. *Определение избыточности программных компонентов.* Для уменьшения вероятности сбоя в наиболее важных компонентах определяется количество версий, которые будут реализованы или разными командами разработчиков, или с использованием различных технологий. Также проводится оценка трудозатрат на разработку алгоритма голосования для данного компонента.

5. *Оценка трудозатрат на разработку компонентов.* Вычисляется среднее время разработки компонента каждого типа, на основании предыдущих работ или с помощью экспертной оценки. Если тип компонентов ранее не разрабатывался, то время определяется экспертно, исходя из анализа подобных типов компонентов.

6. *Определение глубины модернизации компонентов.* Определяется глубина доработки компонентов (весовые коэффициенты) в процентах.

7. *Вычисление трудоемкости разработки.* Суммируется трудоемкость разработки компонентов каждого типа. Если компонент подлежит модернизации, то трудоемкость разработки этого компонента умножается на весовой коэффициент. Также учитываются затраты на разработку алгоритма голосования данного компонента и на его версии, если компонент реализуется с применением программной избыточности.

8. *Определение затрат на этапы жизненного цикла.* На основании предыдущих работ вычисляется соотношение среднего времени, затраченного на другие этапы жизненного цикла, и среднего времени разработки. Вычисленные коэффициенты (соотношения) умножаются на трудоемкость этапа разработки. Таким образом, получается трудоемкость этапов анализа или проектирования, разработки, тестирования и работ по организации процесса.

9. *Вычисление финансовых затрат на модернизацию проекта.* Трудоемкость каждого этапа умножается на среднюю норму оплаты труда сотрудника, работающего на данном этапе жизненного цикла. Затем полученные суммы складываются.

Для проведения расчетов по приведенной методике используются следующие параметры:

M – множество типов компонентов;

i – тип компонента, $i \in M$;

N – множество новых и подлежащих доработке компонентов i -го типа;

j – номер компонента i -го типа, $j \in N$;

T_i – трудоемкость разработки нового или уже имеющегося компонента i -го типа, человеко-часов;

w_{ij} – весовой коэффициент модернизации существующего j -го компонента i -го типа (для новых компонентов равен $w_{ij} = 1$, если нет j компонента i -го типа, то $w_{ij} = 0$);

r_{ij} – количество версий в компоненте, если вводится программная избыточность (если программная избыточность не вводится, то $r_{ij} = 1$);

T_{Rij} – трудоемкость разработки алгоритма голосования для j -го компонента i -го типа (если программная избыточность не вводится, то $T_{Rij} = 0$);

T_m – общие трудозатраты на управление и организацию проекта, человеко-часов;

w_m – весовой коэффициент, определяющий процент трудоемкости организации от трудоемкости разработки;

C_m – стоимость оплаты одного человеко-часа менеджера проекта, руб.;

T_a – общая трудоемкость работы аналитиков и проектировщиков, человеко-часов;

w_a – весовой коэффициент, определяющий процент трудоемкости анализа и проектирования от трудоемкости разработки, человеко-часов;

C_a – стоимость одного человеко-часа аналитика или проектировщика, руб.;

T_{dev} – общая трудоемкость этапа разработки, человеко-часов;

C_{dev} – стоимость оплаты одного человеко-часа разработчика, руб.;

T_t – общая трудоемкость этапа тестирования;

w_t – весовой коэффициент, определяющий процент трудоемкости анализа от трудоемкости разработки;

C_t – стоимость оплаты одного человеко-часа тестировщика, руб.;

T_{doc} – общая трудоемкость документирования доработок системы, человеко-часов;

w_{doc} – весовой коэффициент, определяющий процент трудоемкости документирования от трудоемкости разработки;

C_t – стоимость оплаты одного человеко-часа специалиста по технической документации, руб.;

T_p – общая трудоемкость проекта, человеко-часов;

C_p – общая стоимость проекта, руб.

Трудоемкость разработки рассчитывается по матрице w_{ij} и с учетом введения программной избыточности следующим образом:

$$T_{dev} = \sum_{i=1}^M \sum_{j=1}^N (w_{ij} T_i + (r_{ij} - 1) T_i + T_{Rij}).$$

Трудоемкости остальных этапов жизненного цикла рассчитываются по соответствующим весовым коэффициентам из трудоемкости разработки:

$$T_m = w_m T_{dev},$$

$$T_a = w_a T_{dev},$$

$$T_t = w_t T_{dev},$$

$$T_{doc} = w_{doc} T_{dev}.$$

Общая трудоемкость рассчитывается по формуле

$$T_p = (1 + w_m + w_a + w_t + w_{doc}) T_{dev}.$$

Формула расчета общей стоимости в детализированном виде выглядит следующим образом:

$$C_p = (w_m C_m + w_a C_a + C_{dev} + w_t C_t + w_{doc} C_{doc}) \times \sum_{i=1}^M \sum_{j=1}^N (w_{ij} T_i + (r_{ij} - 1) T_i + T_{Rij}).$$

Для проверки корректности оценки по предложенной методике был проведен анализ оценочных и фактических данных на шести примерах модернизации системы. Расхождение фактической и прогнозируемой стоимости вычислялось по формуле

$$\text{Расхождение} = \frac{\left| \text{Стоимость}_{\text{прогнозируемая}} - \text{Стоимость}_{\text{фактическая}} \right|}{\text{Стоимость}_{\text{фактическая}}}.$$

На основании данных по расхождениям была произведена оценка показателя PRED(L). Он отражает процент оценок, отклонение которых от фактических значений меньше L [3]. Показатель рассчитывается по следующей формуле:

$$\text{PRED}(L) = \frac{100}{n} \sum_{i=1}^n \begin{cases} 1, & \text{Расхождение} \leq L, \\ 0, & \text{иначе} \end{cases}$$

где n – общее количество оценок.

В таблице приведены значения PRED(L) для параметра L , равного 20, 25 и 30 %.

Таблица

L	PRED(L)
PRED(20)	83 %
PRED(25)	83 %
PRED(30)	100 %

Согласно этим данным, расчеты по предложенной методике достаточно точны, что позволяет принять решение о ее использовании в ИТ-компаниях, занимающихся разработкой и поддержкой программного обеспечения сложных информационно-управляющих систем.

Предложенная методика оценки затрат на модернизацию программного обеспечения позволяет учитывать трудозатраты персонала на различных этапах жизненного цикла программного обеспечения; определять размер системы исходя из количества объектных указателей; учитывать глубину доработки архитектурных компонентов программной системы; использовать для расчета коэффициентов экспертную оценку, если нет актуальных данных по предыдущим работам; учитывать трудозатраты на введение программной избыточности.

Библиографические ссылки

1. Царев Р. Ю. Система многоатрибутивного формирования мультиверсионных программных средств

отказоустойчивых систем управления : дис. канд. ... техн. наук. Красноярск, 2003.

2. Алиев Х. Р. Эффективная модель оценки разработки программного обеспечения [Электронный ресурс] // Исследовано в России. 2008. Т. 11, Вып. 3.

С. 338–364. URL: <http://zhurnal.ape.relarn.ru/articles/2008/030.pdf> (дата обращения: 29.09.2012).

3. Глазова М. А. Моделирование стоимости разработки проектов в ИТ-компаниях : дис. ... канд. экон. наук. М., 2008.

V. V. Kukartsev, D. A. Sheenok

COST ESTIMATION FOR MODERNIZATION OF CRITICAL RELIABILITY SYSTEMS SOFTWARE

The authors present the approach of estimation of cost of software upgrade with the account of different stages of life cycle and putting into effect software redundancy. The calculations of the cost of modernization of the software of the real system are made.

Keywords: software redundancy, laboriousness, software, decomposition.

© Кукарцев В. В., Шеенок Д. А., 2012

УДК 004.052.2

В. А. Кулягин

АНАЛИЗ ПРИМЕНЕНИЯ МОДЕЛИ ОЦЕНКИ НАДЕЖНОСТИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Рассматривается аспект применения предложенной ранее модели оценки надежности автоматизированных систем управления предприятием на проекте автоматизации торгового предприятия. Цель – дать оценку надежности системы в виде определенного показателя и рассчитать его на каждом этапе жизненного цикла разработки, экспериментально подтвердить зависимости параметров, присутствующих в формулах, оценить эффективность предложенного метода оценки надежности.

Ключевые слова: модели оценки надежности программного обеспечения, автоматизированная система управления предприятием.

Проекты автоматизации торговли отличаются от проектов автоматизации иных видов деятельности. При автоматизации торговли возрастает значимость факторов производительности и надежности системы автоматизации вследствие больших объемов данных, которыми оперирует торговая компания, а также высокого уровня оперативности ввода данных в систему и их анализа. Автоматизированные системы управления предприятием (АСУП) в торговле достаточно часто требуют интеграции с разнообразным компьютеризированным торговым оборудованием: сканерами штрих-кодов, принтерами этикеток (штрих-кодов), терминалами сбора данных и т. д. При автоматизации торговли акцент всегда делается на автоматизацию незначительного количества формализованных бизнес-процессов (закупки, хранения, обработки заказов и т. д.), в то время как, например, бухгалтерский учет в торговой компании достаточно простой. Практически всегда при внедрении новой системы автоматизации торговли или сложных функций существующих систем автоматизации вопрос о «ручном» вводе данных в новую систему не стоит – большой объем данных требует максимального использования автоматических средств. И наконец, непрерывный цикл

работы торговой компании существенно усложняет планомерное внедрение системы автоматизации. В связи с приведенными причинами возникает интерес к исследованию оценки и обеспечения надежности применительно к этому классу АСУП.

Для оценки надежности АСУП торговых организаций предлагаются модели, описанные в [1]. Если входная надежность компонентов задана в виде значений вероятности безотказной работы, то используется базовая модель надежности. Если же она задана в виде значений вероятности успешного срабатывания, то используется модифицированная модель оценки надежности на основе статических вероятностей компонентов. Надежность автоматизированной системы управления предприятием в торговле $R_{АСУП,t}$ за рабочее время t :

$$R_{АСУП,t} = \prod_{l=1}^N R_{АСУП_{l,l}},$$

$$R_{АСУП_{l,l}} = \prod_{i=K_{l-1}+1}^{K_l} R_{it},$$

$$R_{it} = R_i,$$