

УДК 623.618

А.В. Царевский

ИНТЕГРАЛЬНАЯ ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМОВ ГЕНЕРАЦИИ И ОБРАБОТКИ ПОТОКА ИНФОРМАЦИИ

Царевский Андрей Валентинович, кандидат технических наук, окончил физический факультет Казанского государственного университета. Главный специалист ФНПЦ ОАО «НПО «Марс». Специализируется в области информационного обеспечения автоматизированных систем специального назначения. [e-mail: charevsk@mail.ru].

Аннотация

В статье рассмотрен комплексный анализ топологической сложности алгоритмов генерации тестов и обработки данных в специализированных программных комплексах обмена информацией. Получены интегральные количественные оценки сложности алгоритмов и тестирования для двух структур потока информации.

Ключевые слова: проектирование АСУ, метрики сложности программного обеспечения, проектирование программного обеспечения.

Andrey Valentinovich Tsarevsky, Candidate of Engineering, graduated from the Faculty of Physics at Kazan State University; chief specialist at Federal Research-and-Production Center Open Joint-Stock Company Research-and-Production Association 'Mars'; specializes in the field of information support for special-purpose computer-aided systems. e-mail: charevsk@mail.ru.

Abstract

The article deals with a complex analysis of topological complexity of algorithms for generation of tests and data processing in special-purpose systems for information exchange. The author got integral quantitative measures of algorithm complexity and testing for two structures of in-formation flow.

Key words: design of computer-aided C2 systems, software complexity metrics, software design.

ВВЕДЕНИЕ

функционирование автоматизированных систем специального назначения основано на совокупной обработке огромного количества информации от внешних источников. Результатом обработки является выработка управляющих воздействий. Сбор и предварительную обработку выполняют специализированные программные комплексы обмена информацией [1], отвечающие за качество сбора.

Надежность функционирования комплексов зависит от их сложности. Выбор алгоритмов сбора и обработки связан со способом структуризации потока информации. Именно структура потока предопределяет сложность разрабатываемых комплексов и их тестирование.

В настоящее время единственным вариантом организации потока информации, который используется в рассматриваемых программных комплексах, является представление потока в виде формализованных сообщений (ФС) - двумерная структуризация потока. Формализация сообщения позволяет:

- упорядочить в рамках одного сообщения группу параметров, характеризующих объект;
- компактно представить параметры в сообщении.

Другим вариантом является представление обрабатываемой информации как потока параметров без нарушения компактности их представления - одномерная

структуризация потока. Как будет показано ниже, при определенных условиях, такая организация является предпочтительнее.

Целями статьи являются комплексная оценка и сравнение алгоритмов генерации тестов и алгоритмов обработки информации для двух способов структуризации с применением топологических метрик сложности в обеспечение выбора рациональных проектных и управленческих решений по созданию программного обеспечения (ПО).

ИССЛЕДОВАНИЕ сложности ТОПОЛОГИЧЕСКИМИ МЕТРИКАМИ

Структуризация потоков информации

Введем обозначения: K - количество ФС; M - общее количество параметров в формализованных сообщениях; (M) - одномерная структура потока информации; (M, K) - двумерная структура потока информации.

Графически структуризация (S) показана на рисунке 1.

Варианты исходных данных для проектирования рассматриваемых программных комплексов следующие:

1. Заданы значения параметров M и K . Следует оценить разработку ПО.
2. Задано значение параметра M . Параметр K подлежит определению.
3. Заданы требования к автоматизированным системам специального назначения. Параметры M и K подлежат определению.

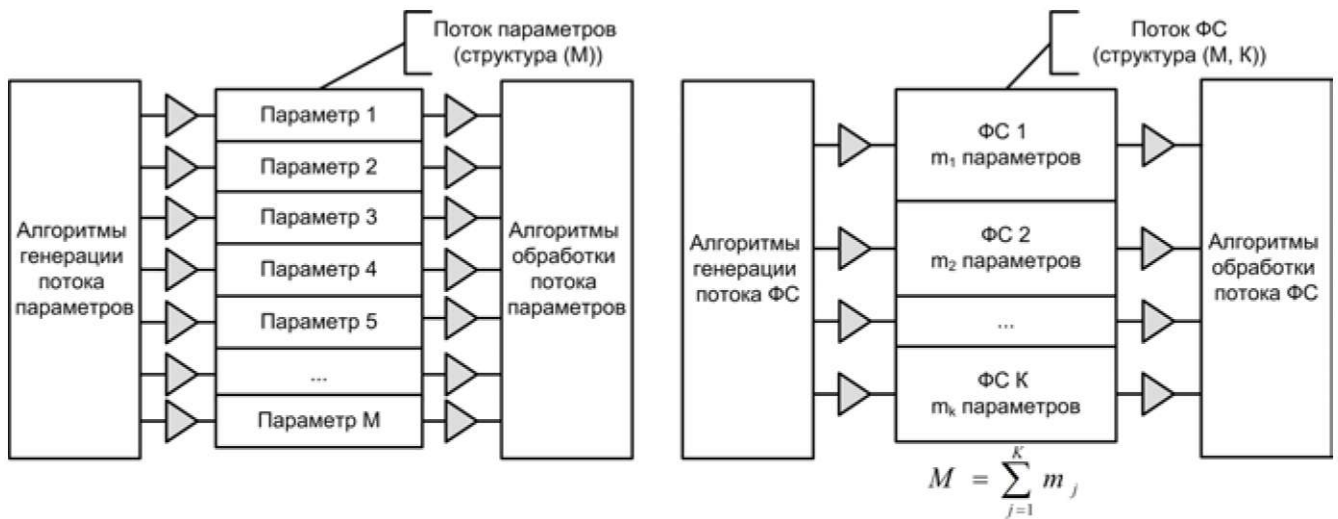


рис. 1. Способы структуризации потоков информации

Топологические метрики для анализа

Общей, инвариантно присущей любому ПО, является его структура. При анализе структурной сложности используются топологические меры [2, 3], т.е. меры, в основе которых лежат топологические характеристики граф-модели программы. Эти меры удовлетворяют следующим требованиям, предъявляемым к показателям: общность применимости, адекватность рассматриваемому свойству, существенность оценки, количественное выражение, малая трудоемкость вычислений, возможность автоматизации оценивания.

Для оценки сложности алгоритмов обработки и тестирования рассматриваемых комплексов воспользуемся топологической мерой Хансена, которая задается в виде пары $\{V, N\}$, где V - цикломатическое число Мак-Кейба (метрика, количественно оценивающая сложность потока управления программы), N - количество блоков (вершин графа), определяющее топологию графа.

Цикломатическая сложность для графов рассчитывается по формуле:

$$V(G) = p + 1,$$

где p - количество предикатных узлов в графе управляющей логики G .

Анализ алгоритмов генерации и обработки потока ФС

При анализе разработанных программных комплексов обмена информацией со структурой потока (M, K) , успешно применяемых на отечественных и экспортных заказах АСУ специального назначения, выделены следующие последовательно выполняемые функциональные блоки (рис. 2):

- «генерация значения параметра» - выработка значения параметра согласно его структуре;
- «передача управления» - подготовка и передача параметра или ФС (структуры из m параметров) на вход блока приема;
- «прием управления» - обеспечение правил приема параметра или ФС;

- «инкапсуляция параметра» - включение параметра в структуру ФС;
- «декапсуляция параметра» - извлечение параметра из структуры ФС;
- «обработка параметра» - приведение параметра к структуре использования;
- «использование параметра» - запись параметра в зоны использования и хранения.

Сложность алгоритмов генерации и обработки одного ФС с m параметрами соответственно равна:

$$A_3(m, 1) = \{V(G_3), N_3\} = \{1, 1 + 2 \cdot m\} = 2 + 2 \cdot m$$

и

$$L_4(m, 1) = \{F(G_4), N_4\} = \{1, 1 + 3 \cdot m\} = 2 + 3 \cdot m,$$

а сложность с M параметрами и K формулярами равна:

$$A_3(M, K) = \sum_{j=1}^K \wedge A_3(m_j) = 2 \cdot K + 2 \cdot M$$

$$L_4(M, K) = \sum_{j=1}^K \wedge A_4(m_j) = 2 \cdot K + 3 \cdot M.$$

Суммарная оценка:

$$A_{34}(M, K) = A_3(M, K) + A_4(M, K) = 4 \cdot K + 5 \cdot M.$$

Суммарная оценка тестирования для одного ФС с m параметрами:

$$C_{34}(m, 1) = \{F(G_3, G_4), N_3 + N_4\} = \{1, 2 + 5 \cdot m\} = 3 + 5 \cdot m,$$

а для M параметров и K формуляров:

$$C_{34}(M, K) = \sum_{j=1}^K C_{34}(m_j, 1) = 3 \cdot K + 5 \cdot M.$$

Интегральная оценка сложности:

$$I_{34}(M, K) = A_{34}(M, K) + C_{34}(M, K).$$

Формулы выведены из предположения, что для генерации и обработки каждого параметра применимы графы G_3 и G_4 .

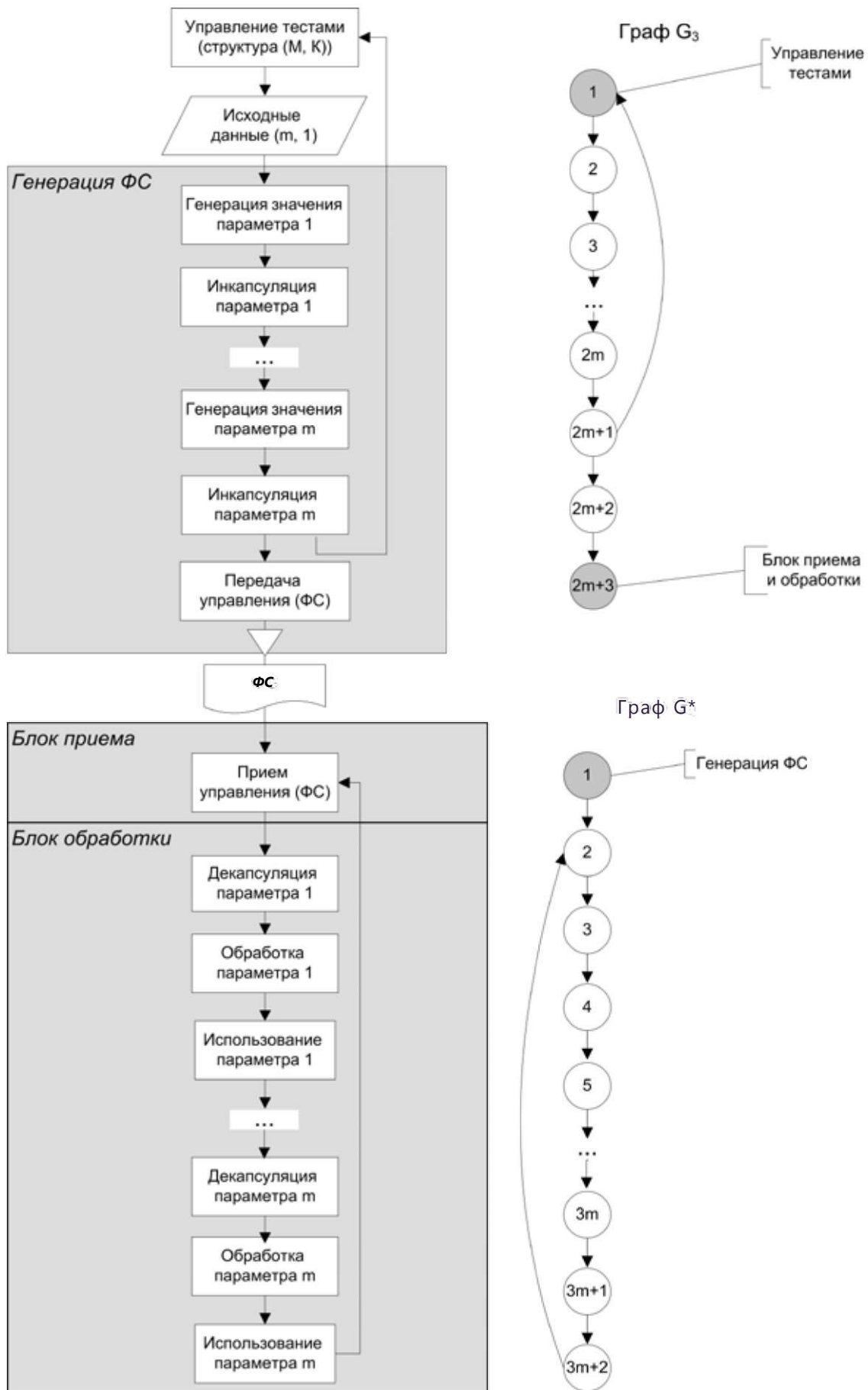


Рис. 2. Алгоритмы и графы управляющей логики при генерации и обработке потока ФС

Построение и анализ алгоритмов генерации и обработки потока параметров

При построении граф-моделей алгоритмов (рис. 3) со структурой потока (M) использованы функциональные блоки алгоритмов обработки потока со структурой (M, K).

Сложность алгоритмов генерации и обработки для одного параметра равна соответственно:

$$A_2(1) = \{V(G_2), N_2\} = \{1, 3\} = 4,$$

для M параметров:

$$A_2(M) = \sum_{i=1}^M A_2(1) = 3 \cdot M$$

$$A_2(M) = \sum_{i=1}^M A_2(1) = 4 \cdot M.$$

Суммарная оценка:

$$A_{12}(M) = 7M.$$

Суммарная оценка сложности тестирования для одного параметра:

$$C_{12}(1) = \{TiG \wedge GJM + NJ\} = \{1, 5\} = 6,$$

для M параметров:

$$C_{12}(M) = \sum_{i=1}^M C_{12}(1) = 6 \cdot M.$$

Интегральная оценка сложности:

$$I_{12}(M) = A_{12}(M) + C_{12}(M).$$

Полученные формальные зависимости позволяют оценить сложность тестирования и обработки с заданной структурой потока информации в обеспечение принятия рациональных проектных решений по созданию ПО.

Сравнительная оценка алгоритмов

Преимущество через интегральные оценки сложности записывается следующим образом:

$$AI = I_{34}(M, K) - I_U(M) = 7K - 3M.$$

Критерий выбора структуры потока по преимуществу M.

1. Предпочтение структуре (M), если $D! > 0$.
2. Предпочтение структуре (M, K), если $D! < 0$.
3. Предпочтение отсутствует, если $D! = 0$.

В таблицах 1 и 2 приведены примеры оценок $D!$ и K для выбора структуры потока при разных значениях M.

таблица 1

Оценки AI и K при M = 500

Параметры оценки	Интервал изменения					
	max	min	max	min	max	min
K	500	216	215	214	213	1
AI	2000	12	5	-2	-9	-1493
Предпочтение структуре	(M)		нет		(M,K)	

ОЦЕНКА ВРЕМЕНИ РАЗРАБОТКИ И ТЕСТИРОВАНИЯ

Обозначим через $T_A(1)$, $T_A(1, 1)$, $T_C(1)$ и $T_C(1, 1)$ время (в рабочих днях) разработки и тестирования программ с минимальной сложностью $A_{12}(1) = 7$, $A_{34}(1, 1) = 9$, $C_{12}(1) = 6$ и $C_{34}(1, 1) = 8$ соответственно.

Время разработки и тестирования единицы сложности соответственно равно:

$$a = \frac{B}{4_4(i, i)} = M \text{ и}$$

$$c = \frac{\Gamma_c(1,1)}{C_{34}(1,1)} = \frac{\Gamma_c(1)}{C_{12}(1)}$$

тогда время разработки будет

$$T_a(M) = <5_{\sigma-A_n(M)} \text{ и}$$

$$T_a(M, K) = <5_{\sigma-A_{34}(M,K)}$$

а время тестирования

$$\Gamma_c(M) = a_c \cdot C_{12}(M) \text{ и}$$

$$T_c(M, K) = o_c \cdot C_{34}(M, K).$$

Преимущество по времени:

$$AT = (T_A(M, K) + T_C(M, K)) - (\Gamma_A(M) + T_C(M)).$$

Критерий выбора структуры потока по преимуществу

AT с величиной допустимого отклонения времени разработки и тестирования $\pm t$:

1. Предпочтение структуре (M), если $AT > t$.
2. Предпочтение структуре (M, K), если $AT < -t$.
3. Предпочтение отсутствует, если $AT \in [-t, t]$.

Из условия $AT = \pm t$ следует, что при

$$K = \frac{\pm t + M \cdot (2 \cdot o_A + o_C)}{4 \cdot o_a + 3 \cdot o_c}$$

изменяется предпочтение структуре потока.

Полученные формальные зависимости позволяют прогнозировать время разработки и тестирования ПО с заданной структурой потока информации в обеспечение принятия необходимых управленческих решений.

В таблицах 3, 4 показаны примеры оценок DT и K, а также динамика их изменения с изменением времени тестирования (например, использование средств автоматизации) при выборе структуры потока. Расчеты проводились при экспертных значениях $T_A(1, 1) = 0,125 \text{ раб. дня (1 час)}$ и $M = 500$ по опыту разработок, а также с отклонением времени разработки и тестирования $t = \pm 3 \text{ раб. дня}$.

таблица 2

Оценки AI и K при M = 800

Параметры оценки	Интервал изменения					
	max	min	max	min	max	min
K	800	344	343	342	341	1
AI	3200	8	1	-6	-13	-2393
Предпочтение структуре	(M)		нет		(M,K)	

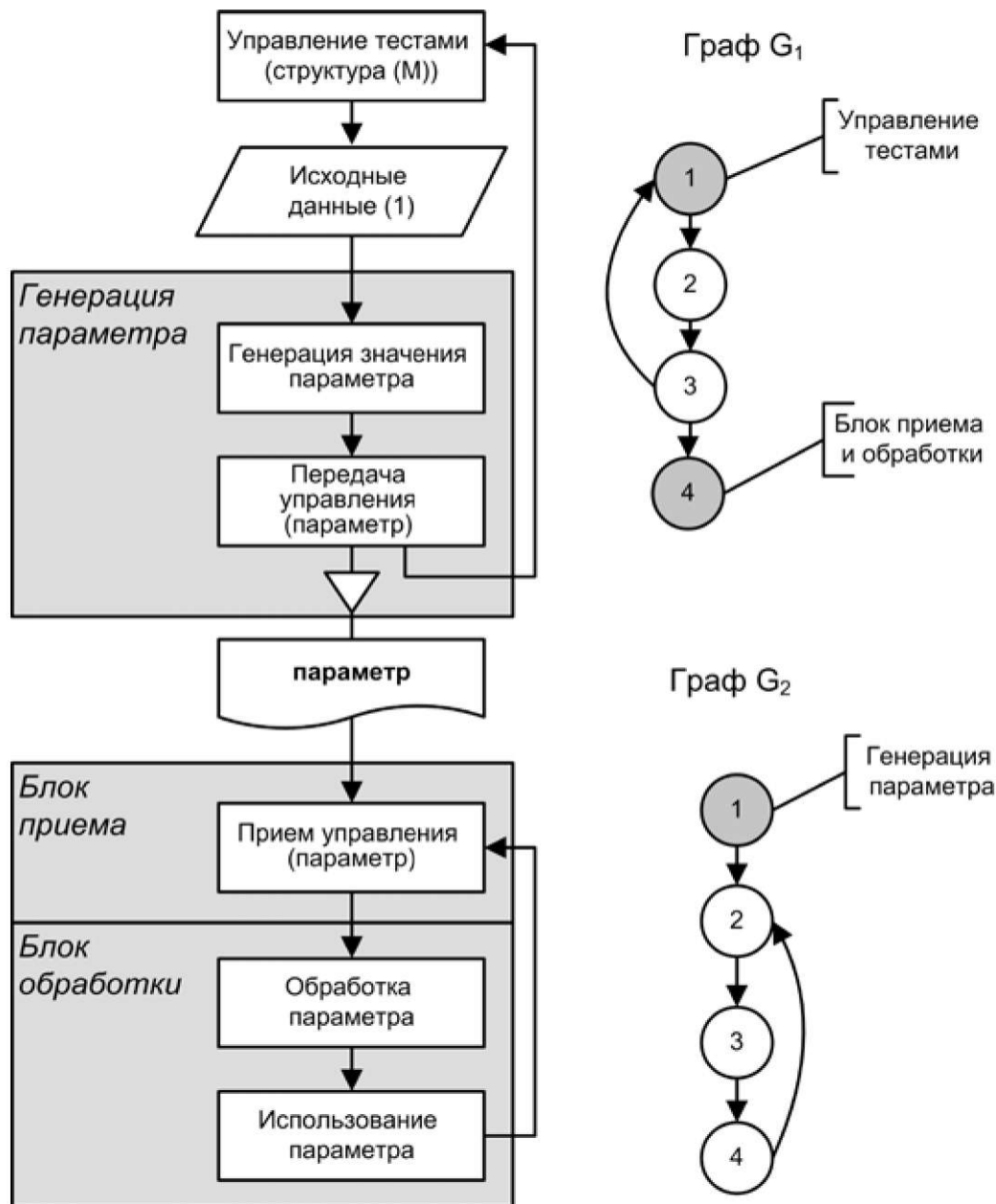


Рис. 3. Алгоритмы и графы управляющей логики при генерации и обработке потока параметров

таблица 3

Оценки AT и K
при $T_c(1, 1) = 0,0625$ раб. дня (30 мин)

Параметры оценки	Интервал изменения					
	max	min	max	min	max	min
K	500	264	263	187	186	1
AT (раб. день)	21	4	3	-3	-4	17
Предпочтение структуре	(M)		нет		(M, K)	

таблица 4

Оценки AT и K
при $T_c(1, 1) = 0,00625$ раб. дня (3 мин)

Параметры оценки	Интервал изменения					
	max	min	max	min	max	min
K	500	307	306	200	199	1
AT (раб. день)	14	4	3	-3	-4	-15
Предпочтение структуре	(M)		нет		(M K)	

ЗАКЛЮЧЕНИЕ

В результате проведенных исследований алгоритмов установлена зависимость интегральной сложности (I) от структуры потока информации (S) и алгоритмов функционирования (G) в обеспечение выбора рациональных проектных и управленческих решений по созданию ПО.

Полученные формальные зависимости позволяют проводить оценку времени разработки программных комплексов в соответствии со структурой потока информации, что необходимо для снижения трудоемкости проектирования, разработки, испытаний и сопровождения, обеспечения высокой надежности создаваемого ПО.

Предложенные критерии выбора структуры потока информации могут быть использованы как самостоятельно, так и в задачах многокритериальной оптимизации про-

ектных решений при разработке программ обработки данных.

СПИСОК ЛИТЕРАТУРЫ

1. Царевский А.В. Аппаратно-программный комплекс обмена информацией для корабельных АСУ // Автоматизация процессов управления. - 2010. - № 2 (20). - С. 84-90.
2. Изосимов А.В., Рыжко А.Л. Метрическая оценка качества программ. - М.: МАИ, 1989.
3. Черноножкин С.К. Меры сложности программ // Системная информатика. - Новосибирск: Наука, 1997. - Вып.5: Архитектурные, формальные и программные модели. - С. 188-227.