



УДК 343.983.25

**МЕТОДИКА ОЦЕНКИ ТРУДОЕМКОСТИ РАЗРАБОТКИ
НА ПЛАТФОРМЕ «1С: ПРЕДПРИЯТИЕ 8»**

Александр Владимирович Мельников*,
Федор Александрович Музалевский**,
Роман Александрович Солодуха***

* Центральный филиал Российского государственного университета правосудия имени В. М. Лебедева, Воронеж, Россия, meln78@mail.ru

** ООО «РТМ ТЕХНОЛОГИИ», Воронеж, Россия, f.muzalevsky@rtmtech.ru

*** Воронежский государственный университет инженерных технологий, Воронеж, Россия, standartal@list.ru

Аннотация. Одним из наиболее частых вопросов, которые ставятся перед экспертами в рамках арбитражных процессов, является определение объема и стоимости фактически выполненных работ. Цель статьи – разработка методики оценки трудозатрат программиста в экспертизах по ИТ-контрактам, связанным с разработкой на платформе «1С: Предприятие 8».

Выполнен обзор имеющихся методов оценки трудоемкости автоматизированных информационных систем. Предложена методика, использующая метод экспертных оценок и метод PERT. Исходными данными являются тип и количество объектов конфигурации, соответствующее им количество строк кода. Учитываются как созданные объекты, так и модифицированные объекты типовой конфигурации. Выполнена модификация метода PERT, позволяющая уменьшить вычислительную сложность без снижения точности оценки.

Приведены экспертные оценки трудоемкости разработки всех объектов конфигурации. Описана программная реализация предложенной методики в части автоматизации процедуры извлечения сведений о типе / количестве объектов конфигурации и соответствующего им количества строк кода.

Методы исследования: теория вероятностей (в части параметров бета-распределения), теория принятия решений (в части экспертного оценивания), системный подход (в качестве методологической базы).

Полученный результат в виде количественной экспертной методики предполагается к практическому применению при производстве судебных экспертиз и внесудебных исследований в качестве типовой экспертной методики.

Ключевые слова: компьютерно-техническая экспертиза, оценка трудозатрат, экспертная оценка, экспертная методика, «1С: Предприятие», PERT, ИТ-контракт

Для цитирования: Мельников А. В., Музалевский Ф. А., Солодуха Р. А. Методика оценки трудоемкости разработки на платформе «1С: Предприятие 8» // Судебная экспертиза. 2026. № 1 (85). С. 124–137.



**TECHNIQUE FOR ASSESSING THE COMPLEXITY
OF DEVELOPMENT ON THE "1C: ENTERPRISE 8" PLATFORM**

Alexander Vladimirovich Melnikov**, *Fedor Alexandrovich Muzalevskii,
*Roman Alexandrovich Solodukha******

* Central branch of the Russian State University of Justice named after V. M. Lebedev, Voronezh, Russia, meln78@mail.ru

** LLC "RTM TECHNOLOGY", Voronezh, Russia, f.muzalevsky@rtmtech.ru

*** Voronezh State University of Engineering Technologies, Voronezh, Russia, standartal@list.ru

Abstract. One of the most frequent expertise questions in court of arbitration is the estimation of the volume and cost of the actually performed work. The purpose of the article is to develop a technique for programmer's labor evaluating in projects based on "1C: Enterprise 8" platform.

An overview of the available methods for assessing the complexity of automated information systems has been performed. A technique using the expert estimations method and the PERT method is proposed. The source data are the type and number of configuration objects, and the corresponding number of lines of code. Both created objects and modified objects of a typical configuration are taken into account. A modification of the PERT method which makes it possible to reduce the computational complexity without reducing the accuracy of the estimation has been performed.

Expert estimates of the project complexity of all configuration object types are given. A software implementation of the proposed technique is described. It automates the procedure for extracting the data of type/number configuration objects and the corresponding number of lines of code.

Research methods: probability theory (concerning in beta distribution parameters), decision theory (concerning in expert estimation), a system approach (as a methodological basis).

The result obtained in the form of a quantitative expert technique. It might be used for practical application in the forensic and extrajudicial investigations as a typical expert technique.

Keywords: computer-technical expertise, labor cost assessment, expert estimation, expert technique, "1C: Enterprise", PERT, IT contract

For citation: Melnikov A. V., Muzalevskii F. A., Solodukha R. A. Technique for assessing the complexity of development on the "1C: Enterprise 8" platform. Forensic Examination, 124–137, 2026. (In Russ.).

При ответе на вопрос суда об объеме и стоимости фактически выполненных работ по ИТ-контракту эксперт, как правило, сталкивается с проблемами отсутствия информации в части временных затрат, численности и квалификации специалистов, участвовавших в разработке, используемых инструментов и сторонних компонентов. В ходе внедрения «коробочного»¹ программного обеспе-

¹ Тиражируемый, готовый к применению неопределенным кругом покупателей программный продукт.



чения неизвестен объем собственного кода, использованного при доработке или модификации, а также отсутствует детальная техническая документация на такие изменения.

Чаще всего в качестве объекта исследования эксперту предоставляют доступ к продуктивной системе, хранилищу конфигураций или коду проекта, а также общую информацию о функционале из договора или технического задания. Эти ограничения существенно затрудняют проведение экспертного исследования.

Анализ судебной практики показал, что суды часто сталкиваются с недостатком доказательств реального объема работ. Примеры споров, связанных с оказанием услуг по разработке, адаптации, модификации программ «1С: Предприятие»:

- АС Москвы, А40-128923/2024, оспаривание факта получения услуг;
- АС СПб и ЛО, А56-99631/2024, отказ от оплаты услуг;
- АС Новосибирской области, А45-35555/2019, претензии к объему работ;
- АС Мурманской области, А42-8668/2023, сомнения в реальности оказания услуг.

Однако на текущий момент не существует типовых экспертных методик проведения такого рода исследований, что усложняет решение задач, необходимых для установления юридически значимых фактов в суде.

Предлагается методика судебной компьютерной экспертизы для ответа на два основных вопроса в рамках разработки на базе «1С: Предприятие»:

- установление объема фактически выполненных работ;
- определение стоимости разработки.

Методика дает экспертам инструмент для независимой количественной оценки, снижая зависимость от субъективных показаний сторон, позволяет объективно оценить, были ли выполнены заявленные в актах работы, и обосновать расчет задолженности. Следует отметить, что данные вопросы актуальны не только в арбитражных спорах, но и в гражданских (взыскание неосновательного обогащения по договору) или уголовных делах (завышение стоимости госконтракта).

Среди существующих методов оценки трудоемкости [1; 2] можно выделить несколько основных. Метод экспертных оценок [3], как частный случай – метод оценки по аналогии [4]. Модель COCOMO II (Constructive Cost Model) [5–7] оценивает трудозатраты на разработку программного обеспечения, настраиваясь по результатам реальных проектов и используя KSLOC – единицу измерения объема программного кода, обозначающую тысячу строк исходного текста программы. Метод оценки функциональных точек (Function Point Analysis) [8] измеряет объем разработки по количеству и сложности функций, таких как ввод, вывод, логические файлы и интерфейсы, с последующим пересчетом функциональных точек в человеко-часы с помощью коэффициентов. Метод декомпозиции работ (Work Breakdown Structure) [3] базируется на иерархической модели, которая разбивает проект на более мелкие управляемые элементы: фазы, задачи, подзадачи и пакеты работ; оценки этих элементов затем суммируются для получения общей оценки. Анализ методов показал, что все они в той или иной мере опираются на экспертное оценивание, различаясь уровнем детализации.



Наиболее востребованным инструментом для планирования и управления проектами является метод критического пути (CPM – Critical path method). Он позволяет определить минимальное время выполнения работ при заданных оценках их длительности. Однако в реальной практике точные оценки времени исполнения задач часто затруднены из-за неопределенности и изменчивости факторов. Для учета неопределенности в оценках длительности задач разработан метод PERT (Program Evaluation and Review Technique)¹ [8; 9]. PERT предполагает использование трех вариантов оценки времени выполнения каждой задачи: оптимистичной, пессимистичной и наиболее вероятной, что позволяет получить более реалистичную и статистически обоснованную оценку.

В отсутствие сведений о трудозатратах специалистов на реализацию проекта единственным реалистичным подходом является моделирование процесса разработки, включающее декомпозицию задач и специалистов, экспертную оценку трудозатрат на реализацию каждой задачи с дальнейшей корректировкой оценки по PERT.

Модификация PERT

PERT является одним из методов, приведенных в материалах Учебного центра ИС: ИТС для расчета времени при разработке программных продуктов на базе «ИС: Предприятие». В рамках этого метода предполагается получение трех экспертных оценок сроков, необходимых для выполнения задачи одним условным разработчиком².

Первая оценка O_t – оптимистичное время (англ. optimistic time), предполагающее отсутствие рисков и задержек при выполнении работ. Вторая P_t – пессимистичное время (англ. pessimistic time), с учетом всех возможных рисков и проблем при реализации проекта. Третья оценка M_t – наиболее вероятное время (англ. most likely time).

Использование этих трех точек позволяет определить ожидаемое время (англ. expected time) E_t :

$$E_t = \frac{(O_t + 4 * M_t + P_t)}{6}, \quad (1)$$

В формуле (1) используется β -распределение³, относящееся к двухпараметрическому семейству абсолютно непрерывных распределений.

В общем виде распределение случайной величины T задается плотностью вероятности f_T :

$$f_T(t) = \frac{1}{B(\alpha, \beta)} t^{\alpha-1} (1-t)^{\beta-1}, \quad (2)$$

где $\alpha, \beta > 0$ произвольные фиксированные параметры;

$B(\alpha, \beta) = \int_0^1 t^{\alpha-1} (1-t)^{\beta-1} dt$ – бета-функция.

¹ Program Evaluation Research Task. Summary report. Phase 1 / U.S. Department of the Navy. Washington (DC): Government Printing Office, 1958.

² Там же.

³ Там же.



Если T случайная величина, имеющая β -распределение, распределенная с плотностью (2) на интервале $(0; 1)$, тогда мода, обозначенная как $\text{mod}(T)$:

$$\text{mod}(T) = \frac{\alpha - 1}{\alpha + \beta - 2}, \quad \alpha > 1, \beta > 1. \quad (3)$$

Для решения прикладных задач используют линейную функцию от случайной величины T , имеющей β -распределение с плотностью (2):

$$T^* = O_t + hT, \quad h > 0. \quad (4)$$

Плотность случайной величины T^* положительна при $O_t < t < P_t$, где $P_t = O_t + h$.

Определение параметров β -распределения (α, β) в методе PERT осуществляется с использованием трех экспертных оценок (O_t, P_t, E_t), что является избыточным. Приняв некоторые допущения и типизировав выполнение работ на практике, в работе А. И. Орлова [10] доказано, что для PERT используют комбинацию параметров β -распределения: $\alpha = 2$, а $\beta = 3$, в этом случае $\text{mod}(T) = 0,33$. В качестве ожидаемого времени выполнения работы (E_t) будем использовать математическое ожидание $M(T)$.

В основу метода PERT положено предположение, что оптимистичное время (O_t) будет при $P(T < t) = 0,166667$, пессимистичное время (P_t) при $P(T < t) = 0,833333$. Тогда при $\alpha = 2, \beta = 3$ и $P(T < t) = 0,166667$: $t_{\min} = 0,1905$; при $\alpha = 2, \beta = 3$ и $P(T < t) = 0,833333$: $t_{\max} = 0,6111$.

Пусть t_m – наиболее вероятное время выполнения работ, t_{\min} – продолжительность работ при наиболее благоприятном исходе, t_{\max} – продолжительность работ при крайне неблагоприятном стечении обстоятельств. Используя нормировку, можно определить соотношение между изучаемыми t_i (t_{\min}, t_m, t_{\max}):

$$t_{\min}^* = \frac{t_{\min}}{t_{\max}} = 0,3117; \quad t_m^* = \frac{t_m}{t_{\max}} = 0,5454; \quad t_{\max}^* = 1, \quad t_o^* = \frac{t_o}{t_{\max}} = 0,654. \quad \text{Таким}$$

образом, зная параметры (α, β) β -распределения и любое из значений времени (O_t, P_t, E_t), в методе PERT можно восстановить оставшиеся.

Сформируем приближенные формулы для использования в практике экспертизы. При $\alpha = 2, \beta = 3$ эксперты оценивают (M_t) и приблизительно вычисляется ожидаемое время (E_t). Также можно определить оптимистичное и пессимистичное время выполнения работ из формул (5):

$$E_t = 1,1 * M_t; \quad P_t = 1,9 * M_t; \quad O_t = 0,7 * M_t. \quad (5)$$

Проведем верификацию полученных формул, сравнив с решением, согласно формуле (1). Например, если наиболее вероятное время выполнения работ $M_t = 2$ (час.), то ожидаемое время $E_t = 2,2$ (час.). Для оптимистичного и пессимистичного времени выполнения работ $P_t = 3,8$ (час.), $O_t = 1,4$ (час.):

$$E_t = \frac{(O_t + 4 * M_t + P_t)}{6} = \frac{1,4 + 4 * 2 + 3,8}{6} = 2,2 \text{ (час.)}.$$

Если $M_t = 10$ (час.), то ожидаемое время $E_t = 11$ (час.). Для оптимистичного и пессимистичного времени выполнения работ $P_t = 19$ (час.), $O_t = 7$ (час.):



$$E_t = \frac{(o_t + 4 * M_t + P_t)}{6} = \frac{7 + 4 * 10 + 19}{6} = 11 \text{ (час.)}.$$

Таким образом, при использовании предложенной модификации PERT и формул (5) снижается вычислительная сложность задачи, так как эксперту необходимо оценить одно время вместо трех. Естественно предположить, что наиболее адекватной экспертной оценкой является наиболее вероятное время выполнения работ, и далее по экспертной оценке наиболее вероятного времени (M_t) рассчитывается ожидаемое время (E_t). На завершающем этапе ожидаемое время используется в качестве исходных данных для применения затратного метода [11] оценки стоимости результатов работ.

Экспертная оценка трудозатрат на разработку объекта метаданных

Конфигурация «1С: Предприятие 8» включает 47 типов объектов. При планировании разработки целесообразно осуществлять декомпозицию задач с учетом структуры этих объектов и объема программного кода, который необходимо добавить в конфигурацию.

Для оценки трудозатрат каждый объект характеризуется двумя показателями:

- нормой времени на разработку одного объекта соответствующего типа, выраженной в часах ($t_{но}$);
- нормой времени на разработку 100 уникальных строк кода модуля, также выраженной в часах ($t_{нк}$).

Уникальные строки кода модуля – это строки программного кода, которые не повторяются в пределах модуля и отражают реальный объем написанной логики¹. Учет уникальных строк кода позволяет точнее оценивать трудоемкость разработки модулей и корректно распределять ресурсы при планировании.

Основное требование к получению валидных экспертных оценок состоит в минимизации ошибок оценивания. Однако достижение этой цели традиционно связано с существенным увеличением объема выборки (числа экспертов), что ведет к возрастанию затрат и длительности экспертизы. В связи с этим целесообразно применить метод последовательного анализа Вальда, позволяющий в среднем сокращать требуемый размер выборки примерно в 2–2,5 раза [12].

Последовательный анализ основан на выборочной схеме, задающей правило принятия решения о продолжении или прекращении отбора на каждом шаге. Такое решение в произвольный момент зависит от всех наблюдений, полученных до этого момента. Следовательно, для последовательности экспертных

¹ Учет уникальных, а не всех строк исходного кода осуществляется по двум причинам:

1. Язык программирования 1С содержит стандартные программные конструкции со стандартным оформлением в коде (Цикл – КонецЦикла, Если – Иначе – КонецЕсли и т. д.). Строки, состоящие из зарезервированных слов, могут составлять значительную часть кода. Их вклад в трудоемкость ничтожен.

2. Стиль написания программ может быть разным. Часто используется неоптимальное структурирование, следствием чего является дублирование функциональности в коде (дублирование участков кода). Трудоемкостью копия можно пренебречь.

Таким образом, для корректного определения трудоемкости все дублирования строк устраняются.



оценок объем выборки (число привлеченных экспертов), при котором осуществляется остановка, является функцией ранее полученных значений оценок.

В работе С. В. Бухарина и А. В. Мельникова [13] предложена модифицированная процедура последовательного анализа, включающая два этапа. На первом этапе выполняется проверка статистических гипотез при заранее установленном «защитном» минимальном объеме выборки, что препятствует преждевременной остановке эксперимента. На втором этапе проводится адаптация состава группы экспертов на основе анализа корреляционных связей между векторами их оценок, что позволяет оптимизировать дальнейший набор респондентов и повысить информативность итоговых оценок.

Для формирования экспертных оценок наиболее вероятного времени выполнения работ (M_i) была собрана фокус-группа в количестве $n = 10$ человек из лиц, имеющих стаж разработки на платформе «1С» не менее пяти лет. После применения модифицированного метода последовательного анализа согласованная группа определена из $n^* = 6$ экспертов. Коэффициент конкордации $W = 0.72$, что соответствует высокой согласованности группы. Согласованность определялась при оценке пяти наиболее часто используемых объектов метаданных – справочники, документы, отчеты, обработки, регистры накопления.

Опрос согласованной экспертной группы стал основой для получения значений норм времени на разработку одного объекта указанного типа ($t_{но,i}$) и ста уникальных строк кода модуля ($t_{нк,i}$), где единицей измерения времени является час. Результат после обработки оценок экспертов представлен в таблице (см. табл.) (оценены все объекты метаданных, кроме «Языки», которые определяют поставкой и не модифицируются, так как не являются объектом разработки).

Экспертные оценки трудоемкости разработки

Тип объекта конфигурации	Папка объекта при выгрузке конфигурации в файлы	i	$t_{но,i}$ (час.)	$t_{нк,i}$ (час.)
1. Общие				
1.1. Подсистемы	Subsystems	1	0,25	–
1.2. Общие модули	CommonModules	2	–	1
1.3. Параметры сеанса	SessionParameters	3	0,1	–
1.4. Роли	Roles	4	0,1	–
1.5. Общие реквизиты	CommonAttributes	5	0,1	–
1.6. Планы обмена	ExchangePlans	6	10	2
1.7. Критерии отбора	FilterCriteria	7	0,25	1,5
1.8. Подписки на события	EventSubscriptions	8	0,25	–
1.9. Регламентные задания	ScheduledJobs	9	0,5	–
1.10. Боты	Bots	10	–	1,5
1.11. Функциональные опции	FunctionalOptions	11	0,5	–
1.12. Параметры функциональных опций	FunctionalOptionsParameters	12	0,1	–



Окончание таблицы

Тип объекта конфигурации	Папка объекта при выгрузке конфигурации в файлы	i	тно, (час.)	тнк, (час.)
1.13. Определяемые типы	DefinedTypes	13	0,25	–
1.14. Хранилище настроек	SettingsStorages	14	0,25	1
1.15. Общие команды	CommonCommands	15	–	1,5
1.16. Группы команд	CommandGroups	16	0,1	–
1.17. Общие формы	CommonForms	17	1	1,5
1.18. Общие макеты	CommonTemplates	18	0,5	–
1.19. Общие картинки	CommonPictures	19	0,25	–
1.20. Пакеты XDTO	XDTOPackages	20	10	–
1.21. WEB-сервисы	WebServices	21	10	–
1.22. HTTP-сервисы	HTTPServices	22	10	–
1.23. WS-ссылки	WSReferences	23	1	–
1.24. Сервисы интеграции	IntegrationServices	24	–	2
1.25. Элементы стиля	StyleItems	25	0,25	–
1.26. Стили	Styles	26	1	–
1.27. Языки	Languages	27	–	–
2. Константы	Constants	28	0,1	–
3. Справочники	Catalogs	29	1	1
4. Документы	Documents	30	4	1,5
4.1. Нумераторы	DocumentNumerators	31	0,5	–
4.2. Последовательности	Sequences	32	0,25	1
5. Журналы документов	DocumentJournals	33	0,5	1
6. Перечисления	Enums	34	0,25	1
7. Отчеты	Reports	35	1,5	1,5
8. Обработки	DataProcessors	36	1,5	2
9. Планы видов характеристик	ChartsOfCharacteristicTypes	37	2	1,5
10. Планы счетов	ChartsOfAccounts	38	10	–
11. Планы видов расчета	ChartsOfCalculationTypes	39	0,5	1,5
12. Регистры сведений	InformationRegisters	40	2	1,5
13. Регистры накопления	AccumulationRegisters	41	2	1,5
14. Регистры бухгалтерии	AccountingRegisters	42	2	1,5
15. Регистры расчета	CalculationRegisters	43	2	1,5
16. Бизнес-процессы	BusinessProcesses	44	1	1,5
17. Задачи	Tasks	45	1	1,5
18. Внешние источники данных	ExternalDataSources	46	1	–



Учет объектов метаданных и строк кода

Подсчет количества объектов конфигурации и сбор кода модулей можно осуществить вручную через интерфейс Конфигуратора. Однако этот вариант довольно трудоемкий, потенциально не лишенный ошибок, поэтому целесообразно данный процесс автоматизировать. Следует отметить, что конфигурация или расширение конфигурации «1С» может поступить на исследование в нескольких формах. В ряде случаев форма представления оговаривается заранее, но, как правило, эксперт на это влияния не имеет. Варианты представления конфигурации для экспертизы:

1. Предпочтительный вариант – каталог с набором файлов конфигурации или архив каталога. Выгрузка конфигурации или расширения конфигурации в файлы (Конфигурация / Выгрузка конфигурации в файлы...) появилась в функционале платформы 8.3.10. Анализ структуры каталога выгрузки позволяет разработать алгоритм автоматизированного подсчета объектов и сбора кода модулей.

2. В файлах конфигурации .cf и .cfe – развертывание на платформе «1С» с последующей выгрузкой в каталог.

3. В формате хранилища – подключение хранилища с загрузкой конфигурации на платформе «1С» с последующей выгрузкой в каталог. Альтернативный вариант для хранилища – преобразование в формат гит-репозитория с использованием gitsync¹. Данный вариант удобен при анализе истории разработки, так как эксперт избавлен от необходимости откатывать хранилище на нужную дату средствами «1С» с последующей выгрузкой в каталог.

4. В составе информационной базы данных .dt – развертывание информационной базы на платформе «1С» с последующей выгрузкой в каталог.

Структура каталога выгрузки следующая: в корне файлы Configuration.xml (словарь «тип объекта верхнего уровня – имя объекта») и ConfigDumpInfo.xml (словарь «тип объекта – имя объекта» для объектов метаданных, в том числе подчиненных). Объекты метаданных одного типа хранятся в одной папке. Названия папок приведены в табл. 1, тип объекта в разделе ChildObjects (Configuration.xml) совпадает с названием папки, но записывается в единственном числе (папка – Tasks, тип объекта – Task).

Сбор данных для исследования предлагает два этапа (при $i = 1..n$, $n = 47$ – количество типов учитываемых объектов метаданных):

1. Парсинг файла Configuration.xml для определения количества объектов метаданных каждого типа (M_i).

2. Сбор содержимого файлов с расширением .bsl (модули «1С» любого уровня) отдельно по каждой папке, соответствующей типу объектов метаданных. Устранение комментариев, лишних пробелов, пустых строк, дубликатов строк – получение множества уникальных строк исходного кода. Множество уникальных строк исходного кода в i -м модуле (S_i) можно записать как $S_i = \{S_1, S_2, S_3, \dots, S_m\}$, где S_j – j -я уникальная строка исходного кода, $m_i = |S_i|$ – общее количество уникальных строк в i -м модуле.

¹ URL: <https://github.com/oscript-library/gitsync> (дата обращения: 11.12.2026).



Тогда общие трудозатраты разработки, в терминах PERT – наиболее вероятное время разработки (M_{P_t}), можно рассчитать по формуле:

$$M_{P_t} = \sum_{i=1}^{46} \left(t_{но,i} M_i + \frac{t_{нк,i}}{100} m_i \right), \quad (6)$$

где $t_{но,i}$ – норма времени на разработку i -го объекта конфигурации;
 $t_{нк,i}$ – норма времени на разработку ста уникальных строк кода модуля i -го объекта конфигурации;

M_i – количество объектов метаданных каждого типа;
 m_i – общее количество уникальных строк в модулях, соответствующих всем экземплярам i -го типа метаданных.

Отдельно следует рассмотреть ситуацию, когда осуществлена доработка (модификация) типовой конфигурации. В этом случае необходимо определить:

1. Множества имен объектов метаданных по каждому типу метаданных:
 - для исследуемой конфигурации – ИК_ M_i ,
 - для типовой конфигурации, на основании которой доработана исследуемая – ТК_ M_i .
2. Множества уникальных строк исходного кода по каждому типу метаданных:
 - для исследуемой конфигурации – ИК_ m_i ,
 - для типовой конфигурации, на основании которой доработана исследуемая – ТК_ m_i .

Тогда количество новых объектов каждого типа определяется как мощность множества $| \text{ИК}_M \setminus \text{ТК}_M |$ новых или измененных уникальных строк кода – $| \text{ИК}_m \setminus \text{ТК}_m |$. Общие трудозатраты на модификацию типовой конфигурации, в терминах PERT – наиболее вероятное время доработки ($M_{д_t}$), можно рассчитать по формуле:

$$M_{д_t} = \sum_{i=1}^{46} \left(t_{но,i} | \text{ИК}_M \setminus \text{ТК}_M | + \frac{t_{нк,i}}{100} | \text{ИК}_m \setminus \text{ТК}_m | \right). \quad (7)$$

После получения оценок наиболее вероятного времени выполнения работ, M_{P_t} или $M_{д_t}$, осуществляется расчет ожидаемого времени E_t по формуле (5) при использовании в качестве исходных данных для затратного метода оценки стоимости результатов работ.

Программная реализация

Предложенная методика определения уникальных строк и объектов метаданных, присутствующих в выгрузке конфигурации (расширении конфигурации) платформы «1С: Предприятие 8», реализована в виде программы¹ на языке Python. Программа состоит из трех консольных приложений:

– parser_Configuration. Анализирует содержимое файла Configuration.xml из папки выгрузки конфигурации (расширения конфигурации) «1С: Предприятие 8». Данный файл хранит информацию об объектах метаданных конфигурации.

¹ Определение объема разработки на платформе 1С: Предприятие 8 / Р. А. Солодуха, Ф. А. Музалевский, А. В. Мельников. Свидетельство о государственной регистрации программы для ЭВМ № 2025663035 от 25.05.2025.



Программа формирует файл в формате MS Excel со списком тип / имя объекта метаданных, количеством объектов метаданных по типам;

– parser_Folder. Проходит по папкам выгрузки конфигурации (расширения конфигурации), если их название присутствует в словаре (формируется на основании технической документации о платформе «1С: Предприятие 8»). Папки имеют название, идентичное имени типа объекта метаданных. Если объект метаданных содержит модуль, то программный код модуля находится в соответствующей папке, в файле с расширением .bsl. Программа находит файлы модулей, построчно считывает код, фиксирует ошибки считывания. Формирует выходной файл в формате Excel со списками «тип объекта / количество уникальных строк» и «тип объекта / количество неучтенных строк». Также формируется папка, содержащая текстовые файлы с уникальными строками модулей по каждому объекту метаданных;

– addon_calc. Программа сравнивает две конфигурации. Предполагается, что одна конфигурация типовая (typical), другая – с наработками (actual). Входными данными являются файлы с уникальными исходными строками и файлы формата Excel, полученные программами parser_Configuration и parser_Folder. С помощью операции вычитания множеств для каждого объекта метаданных формируются файлы с уникальными строками, имеющимися в actual, но отсутствующими в typical. Также формируется файл в формате Excel с информацией по каждому типу объекта метаданных: о количестве привнесенных объектов; количестве уникальных строк кода.

Итак, авторами предложена и математически обоснована методика оценки трудоемкости разработки на платформе «1С: Предприятие 8» на базе модифицированного метода PERT и анализа конфигурации. Значения норм времени на разработку одного объекта указанного типа ($t_{но,i}$) и ста уникальных строк кода модуля ($t_{нк,i}$) получены опросом согласованной экспертной группы. Для применения в затратном методе данные временные оценки пересчитываются в ожидаемое время.

Процедура извлечения количества объектов метаданных каждого типа и количества уникальных строк кода автоматизирована, в том числе, с учетом доработки конфигурации поставщика.

Представленная методика может быть использована в рамках производства судебных экспертиз по оценке объема и / или стоимости разработки на платформе «1С: Предприятие» в отсутствие данных о фактических трудозатратах.

Список источников

1. Анализ методик оценки трудоемкости разработки программного обеспечения / А. В. Будник, О. В. Домакур, Е. С. Романова, Т. Л. Труханович // Веснік сувязі. 2020. № 4 (162). С. 36–41.
2. Глазова М. А. Системы оценки стоимости проектов по разработке программного обеспечения // Прикладная информатика. 2008. № 3. С. 12–27.
3. Макконнелл С. Сколько стоит программный проект. Санкт-Петербург: Рус. ред.: Питер, 2007. 297 с.
4. Angelis L., Stamelos I. A. A simulation tool for efficient analogy based cost estimation // Empirical Software Engineering. 2000. Vol. 5, № 1. P. 35–68.



5. Бакулина А. А., Школов Д. С. Оценка стоимости программного обеспечения с помощью моделей POPOMO II и SLIM // Образование и право. 2018. № 5. С. 124–127.
6. Тютюнников Н. Н. Оценка трудозатрат на создание программных средств для ранней стадии проектирования по модели COCOMO II // Экономика труда и управление персоналом. 2014. № 37. С. 129–134.
7. COCOMO II Model Definition Manual / B. Boehm [et al.]. Los Angeles (CA): University of Southern California: Center for Software Engineering, 2000.
8. Garmus D., Herron D. Function point analysis: measurement practices for successful software projects. Boston (MA): Addison – Wesley, 2001.
9. Kerzner H. Project management: a systems approach to planning, scheduling, and controlling. 12th ed. Hoboken (NJ): John Wiley Sons, 2017. URL: <https://cmls-global.com/wp-content/uploads/2021/07/Project-Management-A-Systems-Approach-to-Planning-Scheduling-and-Controlling.pdf> (дата обращения: 15.12.2025).
10. Орлов А. И. Вероятностный метод оценки времени выполнения работ и бета-распределения // Контроллинг. 2023. № 89. С. 54–61.
11. Методология оценки нематериальных активов / А. Ю. Дорофеев, В. В. Филатов, В. М. Медведев [и др.] // Науковедение: интернет-журн. 2015. № 7 (1). С. 1–28.
12. Вальд А. Последовательный анализ. Москва: Физматгиз, 1960. 328 с.
13. Бухарин С. В., Мельников А. В. Комбинированный метод последовательного анализа оценок экспертизы // Вестник Воронежской государственной технологической академии. Серия «Информационные технологии, моделирование и управление». 2011. № 2 (48). С. 47–51.

References

1. Budnik A. V., Domakur O. V., Romanova E. S., Trukhanovich T. L. Analysis of labor cost technique for software engineering. Bulletin of connections, 36–41, 2020. (In Russ.).
2. Glazova M. A. Estimation of software project systems. Applied informatics, 12–27, 2008. (In Russ.).
3. McConell S. Software estimation: demystifying the black art. Saint Petersburg: Russkaya redakcia; Piter; 2007: 297. (In Russ.).
4. Angelis L., Stamelos I. A. A simulation tool for efficient analogy based cost estimation. Empirical Software Engineering, 35–68, 2000. (In Eng.).
5. Bakulina A. A., Shkolov D. S. Software valuation with the models of POPOMO II and SLIM. Education and law, 124–127, 2018. (In Russ.).
6. Tutunnikov N. N. Labor cost estimation for early stage of software engineering by COCOMO II model. Labor economy and personnel management, 129–134, 2014. (In Russ.).
7. Boehm B. (et al.) COCOMO II Model Definition Manual. Los Angeles (CA): University of Southern California; Center for Software Engineering; 2000. (In Eng.).
8. Garmus D., Herron D. Function point analysis: measurement practices for successful software projects. Boston (MA): Addison – Wesley, 2001. (In Eng.).
9. Kerzner H. Project management: a systems approach to planning, scheduling, and controlling. 12th ed. Hoboken (NJ): John Wiley Sons; 2017. Available from: <https://cmls-global.com/wp-content/uploads/2021/07/Project-Management-A-Systems->



Approach-to-Planning-Scheduling-and-Controlling.pdf. Accessed: 15 December 2025. (In Eng.).

10. Orlov A. I. Probabilistic method for estimating work completion time and beta-distribution. *Controlling*, 54–61, 2023. (In Russ.).

11. Dorofeev A. U., Filatov V. V., Medvedev V. M. (et al.) The methodology for the valuation of intangible assets. *Naukovedenie. Internet-magazine*, 1–28, 2015. (In Russ.).

12. Vald A. *Sequential analysis*. Moscow: Physmatgiz; 1960: 328. (In Russ.).

13. Bukharin S. V., Melnikov A. V. Combine method for sequential analysis of expertise estimation. *Bulletin of the Voronezh State Technological Academy. Series: Information technologies, management and modeling*, 47–51, 2011. (In Russ.).

Мельников Александр Владимирович,

заведующий кафедрой правовой информатики,
информационного права и естественнонаучных дисциплин
Центрального филиала
Российского государственного университета правосудия
имени В. М. Лебедева,
доктор технических наук, доцент;
meln78@mail.ru

Музалевский Федор Александрович,

ведущий эксперт компьютерно-технического направления
ООО «РТМ ТЕХНОЛОГИИ»,
кандидат физико-математических наук;
f.muzalevsky@rtmtech.ru

Солодуха Роман Александрович,

доцент кафедры информационных технологий,
моделирования и управления
Воронежского государственного университета
инженерных технологий,
кандидат технических наук, доцент;
standartal@list.ru

Melnikov Alexander Vladimirovich,

head of the chair of legal informatics,
information law and natural sciences
of the Central branch of the Russian State University of Justice
named after V. M. Lebedev,
doctor of technical sciences, docent;
meln78@mail.ru

Muzalevskii Fedor Alexandrovich,

leading expert in computer department
of the LLC "RTM TECHNOLOGY",



candidate of physical-mathematical sciences;
f.muzalevsky@rtmtech.ru

Solodukha Roman Alexandrovich,
associate professor at the chair of information technology,
modeling and control,
Voronezh State University of Engineering Technologies,
candidate of technical sciences, docent;
standartal@list.ru

Статья поступила в редакцию 16.02.2026; одобрена после рецензирования
20.02.2026; принята к публикации 20.02.2026.

The article was submitted 16.02.2026; approved after reviewing 20.02.2026; ac-
cepted for publication 20.02.2026.

* * *