

Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

К. А. АМЕЛИЧЕВА

**«Оценка трудоемкости разработки программного
продукта»**

Методические указания к выполнению практико-ориентированного
задания
по дисциплине «Управление программными проектами»

Калуга, 2019

Оглавление

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	4
1. Методы оценки трудоемкости создания ПО	4
1.1 Методика функциональных указателей	6
1.2 Методика оценки трудоемкости разработки ПО на основе вариантов использования.....	17
1.2.1 Определение весовых показателей действующих лиц	17
1.2.2 Определение весовых показателей вариантов использования..	17
1.2.3 Определение технической сложности проекта.....	19
1.2.4 Определение уровня квалификации разработчиков	19
1.2.5 Оценка трудоемкости проекта	20
1.3 Модель композиции приложения	22
1.4 Методы, основанные на экспертных оценках	25
ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ	26
Пример 1. Рассмотрим порядок выполнения практико- ориентированного задания на примере:	26
Пример 2. Трудоемкость разработки для учебного заведения системы регистрации студентов	30
Пример 3. Трудоемкость создания программного продукта «Учет доверенностей» согласно методике композиции приложения	33
Варианты	36
ФОРМА ОТЧЕТА ПО ПРАКТИКО-ОРИЕНТИРОВАННОМУ ЗАДАНИЮ	37
Список литературы.....	38
Приложения.....	39
Приложение А. Коэффициенты регулировки сложности	39
Приложение Б. Пример отчета по лабораторной работе	44
Приложение Б. Бланк доверенности (Типовая межотраслевая форма N М- 2а).....	48

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения практико-ориентированного задания является формирование практических навыков оценки трудоемкости создания программного обеспечения.

Основными задачами практико-ориентированного задания являются изучение методов расчета трудозатрат на разработку программных продуктов на этапе планирования работ.

. Результатами работы являются:

- Подготовленный отчет, содержащий подробное изложение этапов выполнения практико-ориентированного задания

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1. Методы оценки трудоемкости создания ПО

Методы оценки трудозатрат на создание ПО основаны на анализе статистических данных о ранее выполненных проектах, при этом эмпирически определяется зависимость трудоемкости проекта от какого-нибудь количественного показателя, характеризующего программный продукт (ПП). Проводится оценка этого показателя для данного проекта, после чего на основе применяемой модели прогнозируются будущие трудозатраты.

Большинство моделей для определения трудоемкости разработки программного проекта может быть сведено к функции пяти основных параметров [1]:

- размера конечного проекта (для компонентов, написанных вручную), который обычно измеряется числом строк исходного кода или количеством функциональных точек, необходимых для реализации данной функциональности;
- особенностей процесса, используемого для получения конечного проекта, в частности его способность избегать непроизводительных видов деятельности (переделок, бюрократических проволочек, затрат на взаимодействие);
- возможностей персонала, участвующего в разработке программного проекта, в особенности его профессионального опыта и знания предметной области проекта;
- среды, которая состоит из инструментов и методов, используемых для эффективного выполнения разработки программного проекта и автоматизации процесса;
- требуемого качества проекта, включающего в себя его функциональные возможности, производительность, надежность и адаптируемость.

Наиболее влиятельный фактор оценки трудоемкости в этих моделях – размер программного проекта. Процедура оценки трудоемкости разработки программного проекта состоит из следующих действий:

- оценка размера разрабатываемого проекта;

- оценка трудоемкости в человеко-часах (чел.-ч);
- оценка продолжительности проекта в календарных месяцах;
- оценка стоимости проекта.

Оценка размера проекта базируется на знании требований к системе. Для такой оценки существуют два основных способа:

1. По аналогии. Если в прошлом приходилось иметь дело с подобным проектом и его оценки известны, то можно, отталкиваясь от них, приблизительно оценить свой проект.
2. Путем подсчета размера по определенным алгоритмам на основе исходных данных (предъявленным требованиям к программному продукту).

Проблемы оценки размера программного проекта:

- проблема может быть недостаточно хорошо понята разработчиками и (или) заказчиками из-за того, что некоторые факты были упущены или искажены;
- недостаток или полное отсутствие исторических данных не позволяет создать базу для оценок в будущем;
- проектирующая организация не располагает стандартами, с помощью которых можно выполнять процесс оценивания (либо в случае наличия стандартов их никто не придерживается); в результате наблюдается недостаток совместимости при выполнении процесса оценивания;
- менеджеры проектов полагают, что было бы неплохо фиксировать требования в начале проекта, заказчики же считают, что не стоит тратить время на разработку спецификации требований.

Основными единицами измерения размера программного проекта являются:

- количество строк кода (LOC – Lines of Code);
- функциональные (объектные) указатели (FP – Function Points).

Количество строк кода – исторически самая известная и до недавнего времени распространенная единица измерения.

В настоящее время наиболее часто применяются метрики ПП, основанные на измерении функциональной полезности продукта, это так называемые функционально-ориентированные метрики.

Следует отметить, что применяемые при оценке трудоемкости модели содержат ряд параметров, численные значения которых определяют величину трудозатрат и являются нормативами (стандартами), принятыми на предприятии – разработчике ПП. Эти численные значения зависят от:

- квалификации разработчиков;
- используемых инструментальных средств;
- накопленного на предприятии опыта производства ПО.

Поэтому при применении соответствующей методики оценке трудоемкости изготовления ПО необходимо обоснованно задать численные значения соответствующим параметрам (коэффициентам) модели.

1.1 Методика функциональных указателей

Область применения метода функциональных указателей – коммерческие информационные системы.

Достоинства функционально-ориентированных метрик:

- не зависят от языка программирования;
- легко вычисляются на любой стадии проекта.

Недостаток функционально-ориентированных метрик: результаты основаны на субъективных данных, используются не прямые, а косвенные измерения.

В методике функциональных указателей [3] для определения объема работ используется 5 информационных характеристик:

внешний ввод – элементарный процесс, перемещающий данные из внешней среды в приложение. Данные могут поступать с экрана ввода или из другого приложения. Данные могут использоваться для обновления внутренних логических файлов. Данные могут содержать как управляющую, так и деловую информацию. Управляющие данные не модифицируют внутренние логические файлы;

внешний вывод – элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду. Кроме того, в этом процессе могут обновляться внутренние логические файлы. Выводы означают отчеты, экраны, распечатки, сообщения об ошибках или выходные файлы, посылаемые другим приложениям. Отчеты и файлы создаются на основе внутренних логических файлов и внешних интерфейсных файлов. Дополнительно этот процесс может использовать вводимые данные: критерии поиска либо параметры, не поддерживаемые внутренними логическими файлами. Вводимые данные носят временный характер;

внешний запрос – элементарный процесс, работающий как с вводимыми, так и с выводимыми данными. Его результат – данные, возвращаемые из внутренних логических файлов и внешних интерфейсных файлов. Входная часть процесса не модифицирует внутренние логические файлы, а выходная часть не несет данных, вычисляемых приложением (в этом и состоит отличие запроса от вывода);

внутренний логический файл – распознаваемая пользователем группа логически связанных данных, которая размещена внутри приложения и обслуживается через внешние вводы;

внешний интерфейсный файл – распознаваемая пользователем группа логически связанных данных, которая размещена внутри другого приложения и поддерживается им. Внешний файл данного приложения является внутренним логическим файлом в другом приложении.

Вводы, выводы и запросы относят к категории транзакция. Транзакция – это элементарный процесс, различаемый пользователем и перемещающий данные между внешней средой и программным приложением.

Оценка числа функциональных точек (ФТ) для программного продукта выводится на основе данных, которые определяются в результате анализа информационной области программного изделия и изучения особенностей его будущего функционирования.

Порядок расчета трудоемкости разработки ПО:

- определение количества и сложности функциональных информационных характеристик;
- определение количества связанных с каждой информационной характеристикой элементарных данных (DET), элементарных записей (RET) и файлов типа ссылок (FTR);
- определение сложности (в зависимости от количества DET, RET и FTR);
- подсчет количества функциональных точек приложения;
- подсчет количества функциональных точек с учетом общих характеристик системы оценка трудоемкости разработки (с использованием различных статистических данных).

Для транзакций ранжирование основано на количестве ссылок на файлы и количестве типов элементов данных. Для файлов ранжирование основано на количестве типов элементов-записей и типов элементов данных, входящих в файл.

Тип элемента-записи – подгруппа элементов данных, распознаваемая пользователем в пределах файла.

Тип элемента данных – уникальное не рекурсивное (неповторяемое) поле, распознаваемое пользователем. В качестве примера рассмотрим табл. 1.

Таблица 1. Отчет радиостанции

День	Уровень активности дня недели, хиты	В процентах от суммы хитов	Сеансы пользователя
Понедельник	1887	16,41	201
Вторник	1547	13,45	177
Среда	1975	17,17	195
Четверг	1591	13,83	191
Пятница	2209	19,21	200
Суббота	1286	11,18	121
Воскресенье	1004	8,73	111
Сумма по рабочим дням	9209	80,08	964

Сумма по выходным дням	2290	19,91	232
---------------------------	------	-------	-----

В этой таблице 10 элементов данных: День, Хиты, проценты от Суммы хитов, Сеансы пользователя, Сумма хитов (по рабочим дням), Сумма процентов от Суммы хитов (по рабочим дням), Сумма сеансов пользователя (по рабочим дням), Сумма хитов (по выходным дням), Сумма процентов от Суммы хитов (по выходным дням), Сумма сеансов пользователя (по выходным дням). Отметим, что поля День, Хиты, В процентах от Суммы хитов, Сеансы пользователя имеют рекурсивные данные, которые в расчете не учитываются.

Примеры определения элементов данных для различных характеристик приведены в табл. 2, а табл. 3 содержит правила учета элементов данных из графического интерфейса пользователя (GUI).

Таблица 2. Правила учета элементов данных для транзакций

Информационная характеристика	Элементы данных
Внешние Вводы	Поля ввода данных, сообщения об ошибках, вычисляемые значения, кнопки
Внешние Выводы	Поля данных в отчетах, вычисляемые значения, сообщения об ошибках, заголовки столбцов, которые читаются из внутреннего файла
Внешние Запросы	Вводимые элементы: поле, используемое для поиска, щелчок мыши.

Таблица 3. Правила учета элементов данных из графического интерфейса пользователя

Элемент данных	Правило учета
Группа радиокнопок	Так как в группе пользователь выбирает только одну радиокнопку, все радиокнопки группы считаются одним элементом данных
Группа флажков (переключателей)	Так как в группе пользователь может выбрать несколько флажков, каждый

Элемент данных	Правило учета
	флажок считают элементом данных
Командные кнопки	Командная кнопка может определять действие добавления, запроса. Кнопка ОК может вызывать транзакции (различных типов). Кнопка Next может быть входным элементом запроса или вызывать другую транзакцию. Каждая кнопка считается отдельным элементом данных
Списки	Список может быть внешним запросом, но результат запроса может быть элементом данных внешнего ввода

Например, GUI для обслуживания клиентов может иметь поля Имя, Адрес, Город, Страна, Почтовый Индекс, Телефон, Email. Таким образом, имеется 7 полей или семь элементов данных. Восьмым элементом данных может быть командная кнопка (добавить, изменить, удалить). В этом случае каждый из внешних вводов Добавить, Изменить, Удалить будет состоять из 8 элементов данных (7 полей плюс командная кнопка).

Обычно одному экрану GUI соответствует несколько транзакций. *Типичный экран включает несколько внешних запросов, сопровождающих внешний ввод.*

Рассмотрим порядок учета сообщений. В приложении с GUI генерируются 3 типа сообщений: сообщения об ошибке, сообщения подтверждения и сообщения уведомления.

Сообщения *об ошибке* (например, Требуется пароль) и сообщения *подтверждения* (например, Вы действительно хотите удалить клиента?) указывают, что произошла ошибка или что процесс может быть завершен.

Эти сообщения не образуют самостоятельного процесса, они являются частью другого процесса, то есть считаются элементом данных соответствующей транзакции.

С другой стороны, *уведомление* является независимым элементарным процессом. Например, при попытке получить из банкомата сумму денег, превышающую их количество на счете, генерируется сообщение «Не хватает средств для завершения транзакции». Оно является результатом чтения информации из

файла счета и формирования заключения. Сообщение уведомления рассматривается как внешний вывод.

Данные для определения ранга и оценки сложности транзакций и файлов приведены в табл. 4 – 8 (числовая оценка указана в круглых скобках). Например, внешнему вводу, который ссылается на 2 файла и имеет 7 элементов данных, по табл. 4 назначается средний ранг и оценка сложности 4.

Таблица 4. Ранг и оценка сложности внешних вводов

Ссылки на файлы	Элементы данных		
	1-4	5-15	>15
0-1	Низкий (3)	Низкий (3)	Средний (4)
2	Низкий (3)	Средний (4)	Высокий (6)
> 2	Средний (4)	Высокий (6)	Высокий (6)

Таблица 5. Ранг и оценка сложности внешних выводов

Ссылки на файлы	Элементы данных		
	1-4	5-19	>19
0-1	Низкий (4)	Низкий (4)	Средний (5)
2-3	Низкий (4)	Средний (5)	Высокий (7)
> 3	Средний (5)	Высокий (7)	Высокий (7)

Таблица 6. Ранг и оценка сложности внешних запросов

Ссылки на файлы	Элементы данных		
	1-4	5-19	>19
0-1	Низкий (3)	Низкий (3)	Средний (4)
2-3	Низкий (3)	Средний (4)	Высокий (6)
> 3	Средний (4)	Высокий (6)	Высокий (6)

Таблица 7. Ранг и оценка сложности внутренних логических файлов

Типы элементов- записей	Элементы данных		
	1-19	20-50	>50
0-1	Низкий (7)	Низкий (7)	Средний (10)
2-5	Низкий (7)	Средний (10)	Высокий (15)
> 5	Средний (10)	Высокий (15)	Высокий (15)

Таблица 8. Ранг и оценка сложности внешних интерфейсных файлов

Типы элементов-записей	Элементы данных		
	1-19	20-50	>50
0-1	Низкий (5)	Низкий (5)	Средний (7)
2-5	Низкий (5)	Средний (7)	Высокий (10)
> 5	Средний (7)	Высокий (10)	Высокий (10)

Отметим, что, если во внешнем запросе ссылка на файл используется как на этапе ввода, так и на этапе вывода, она учитывается только один раз. Такое же правило распространяется и на элемент данных (однократный учет).

После определения всех информационных характеристик программного продукта и их сложности приступают к расчету метрики – количества функциональных указателей FP (Function Points) [3].

Исходные данные для расчета сводятся в табл. 9.

Таблица 9. Исходные данные для расчета FP-метрик

Имя характеристики	Ранг, сложность, количество			
	Низкий	Средний	Высокий	Итого
Внешние вводы	<input type="checkbox"/> x3= __	<input type="checkbox"/> x4= __	<input type="checkbox"/> x6= __	= <input type="checkbox"/>
Внешние выводы	<input type="checkbox"/> x4= __	<input type="checkbox"/> x5= __	<input type="checkbox"/> x7= __	= <input type="checkbox"/>
Внешние запросы	<input type="checkbox"/> x3= __	<input type="checkbox"/> x4= __	<input type="checkbox"/> x6= __	= <input type="checkbox"/>
Внутр. логические файлы	<input type="checkbox"/> x7= __	<input type="checkbox"/> x10= __	<input type="checkbox"/> x15= __	= <input type="checkbox"/>
Внешн. интерфейс. файлы	<input type="checkbox"/> x5= __	<input type="checkbox"/> x7= __	<input type="checkbox"/> x10= __	= <input type="checkbox"/>
Общее количество рангов				= <input type="checkbox"/>

В таблицу заносится количественное значение характеристики каждого вида (по всем уровням сложности). Места подстановки значений отмечены прямоугольниками (прямоугольник играет роль метки-заполнителя). Количественные

значения характеристик умножаются на числовые оценки сложности. Полученные в каждой строке значения суммируются, давая полное значение для данной характеристики. Эти полные значения затем суммируются по вертикали, формируя общее количество.

Количество функциональных указателей вычисляется по формуле

$$FP = \text{Общее количество рангов} \times \left(0,65 + 0,01 \times \sum_{i=1}^{14} F_i \right), \quad (1)$$

где F_i – коэффициенты регулировки сложности (табл. 10), принимающие целые значения: 0 – 5 в зависимости от сложности реализации соответствующей характеристики ПП (Приложение А).

Таблица 10. Коэффициенты регулировки сложности

№	Системный параметр	Описание
1	Передачи данных	Сколько средств связи требуется для передачи или обмена информацией с приложением или системой?
2	Распределенная обработка данных	Как обрабатываются распределенные данные и функции
3	Производительность обработки?	Нуждается ли пользователь в фиксации времени ответа или производительности?
4	Эксплуатационные ограничения	Должны ли учитываться некоторые ограничения, связанные с безопасностью или временем реакции, конкретные требования к процессору со стороны компонентов приложения?
5	Частота транзакций	Как часто выполняются транзакции? (каждый день, каждую неделю, каждый месяц)
6	Оперативный ввод данных	Какой процент информации надо вводить в режиме онлайн?
7	Эффективность работы конечного	Приложение проектировалось для обеспечения эффективной работы

№	Системный параметр	Описание
	пользователя	конечного пользователя?
8	Оперативное обновление	Как много внутренних файлов обновляется в онлайн-транзакции?
9	Сложность обработки	Выполняет ли приложение интенсивную логическую или математическую обработку?
10	Повторная используемость	Приложение разрабатывалось для удовлетворения требований одного или многих пользователей?
11	Простота установки	Насколько трудны преобразование и инсталляция приложения?
12	Простота эксплуатации	Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления?
13	Разнообразные условия размещения	Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций?
14	Простота изменений	Была ли спроектирована, разработана и поддержана в приложении простота изменений?

Полученная FP-оценка пересчитывается в LOC-оценки V

$$V = K_{яз} \times FP, \quad (2)$$

где коэффициент $K_{яз}$ зависит от языка программирования, используемого для реализации ПО, берется из табл. 11.

Таблица 11. Пересчет FP-оценок в LOC-оценки

Язык программирования	Количество операторов на один FP (Кяз)
Ассемблер	320
C	128
Фортран	106
Паскаль	90
C++	64
Java	53
Ada 95	49
Visual C++	34
Delphi Pascal	29
Smalltalk	22
Perl	21
HTML3	15
Access	38
ANSI SQL	13
C++	53
Data base default	40
Delphi 5	18
Excel 5	6
FoxPro 2.5	34
Oracle Developer	23
PowerBuilder	16
Visual Basic 6	24
HTML 4	14
Java 2	46
1C	10

Для пересчета объема программы в условных строках V в трудозатраты T используется промежуточная модель COCOMO, в соответствии с которой номинальную трудоемкость (без учета коэффициентов затрат труда, стоимостных факторов и сложности) можно вычислить по формуле

$$T = N1 \times KSLOC^{N2}, \quad (3)$$

где

$$KSLOC \text{ (тыс. строк)} = V / 1000$$

Значения N1 и N2 определяются по табл. 12 [3].

Таблица 12. Коэффициенты N1 и N2

Тип ПО	N1	N2
Распространенное	3,2	1,05
Полунезависимое	3,0	1,12
Встроенное	2,8	1,20

Распространенное ПО – ПО небольшого объема (не более 50 KSLOC), разрабатываемое относительно небольшой группой опытных специалистов в стабильных условиях.

Полунезависимое ПО – ПО среднего объема (не более 300 KSLOC), разрабатываемое неоднородной группой специалистов средней квалификации.

Встроенное ПО – ПО с жесткими ограничениями (система резервирования авиабилетов, система управления воздушным движением и т.п.).

Время разработки вычисляется по формуле

$$t_{\text{разр}} = 2,5 \times T^{N3}, \quad (4)$$

где значения N3 берутся из табл. 13 [3].

Таблица 13. Коэффициенты N1 и N2

Тип ПО	N3
Распространенное	0,38
Полунезависимое	0,35
Встроенное	0,32

Рекомендуемое правило распределения затрат проекта – 40–20–40:

- на анализ и проектирование приходится 40% затрат (из них на планирование и системный анализ – 5%);
- на кодирование – 20%;
- на тестирование и отладку – 40%.

1.2 Методика оценки трудоемкости разработки ПО на основе вариантов использования

Методика основана на выделении действующих лиц и вариантов использования ПП.

1.2.1 Определение весовых показателей действующих лиц

Все действующие лица системы делятся на три типа: простые, средние и сложные.

Простое действующее лицо представляет внешнюю систему с четко определенным программным интерфейсом (API).

Среднее действующее лицо представляет либо внешнюю систему, взаимодействующую с данной системой посредством протокола наподобие TCP/IP, либо личность, пользующуюся текстовым интерфейсом (например, ASCII-терминалом).

Сложное действующее лицо представляет личность, пользующуюся графическим интерфейсом (GUI).

Подсчитанное количество действующих лиц каждого типа n_i умножается на соответствующий весовой коэффициент k_{ai} (табл. 14), затем вычисляется общий весовой показатель A

$$A = \sum n_i \times k_{ai}, \quad (5)$$

Таблица 14. Весовые коэффициенты действующих лиц

Тип действующего лица	Весовой коэффициент k_{ai}
Простое	1
Среднее	2
Сложное	3

1.2.2 Определение весовых показателей вариантов использования

Все варианты использования делятся на три типа: простые, средние и сложные, в зависимости от количества транзакций в потоках событий (основных и альтернативных). В данном случае

под транзакцией понимается атомарная последовательность действий, которая выполняется полностью или отменяется. Подсчитанное количество вариантов использования каждого типа m_i умножается на соответствующий весовой коэффициент k_{vi} (табл. 15, 16), затем вычисляется общий весовой показатель UC (6)

$$UC = \sum m_i \times k_{vi}, \quad (6)$$

Другой способ определения сложности вариантов использования заключается в подсчете количества классов анализа m_i , участвующих в их реализации, с учетом их весовых коэффициентов k_{vi} (табл. 16).

Таблица 15. Весовые коэффициенты вариантов использования

Тип варианта использования	Описание	Весовой коэффициент (k_{vi})
Простой	3 или менее транзакций	5
Средний	От 4 до 7 транзакций	10
Сложный	Более 7 транзакций	15

Таблица 16. Весовые коэффициенты вариантов использования

Тип варианта использования	Описание	Весовой коэффициент (k_{vi})
Простой	Менее 5 классов	5
Средний	От 5 до 10 классов	10
Сложный	Более 10 классов	15

Обобщенный весовой показатель UUCP (unadjusted use case points) вычисляется по формуле

$$UUCP = A + UC. \quad (7)$$

1.2.3 Определение технической сложности проекта

Техническая сложность проекта (TCF – technical complexity factor) вычисляется с учетом показателей технической сложности T_i (табл. 17).

Таблица 17. Показатели технической сложности

Показатель	Описание	Вес
T1	Распределенная система	2
T2	Высокая производительность (пропускная способность).	1
T3	Работа конечных пользователей в режиме онлайн	1
T4	Сложная обработка данных	1
T5	Повторное использование кода	1
T6	Простота установки	0,5
T7	Простота использования	0,5
T8	Переносимость	2
T9	Простота внесения изменений	1
T10	Параллелизм	1
T11	Специальные требования к безопасности	1
T12	Непосредственный доступ к системе со стороны внешних пользователей	1
T13	Специальные требования к обучению пользователей	1

Каждому показателю T_i присваивается значение в диапазоне от 0 до 5 (0 означает отсутствие значимости показателя для данного проекта, 5 – высокую значимость). Значение TCF вычисляется по следующей формуле:

$$TCF = 0,6 + \left(0,01 \times \left(\sum T_i \times \text{Вес}_i \right) \right). \quad (8)$$

1.2.4 Определение уровня квалификации разработчиков

Уровень квалификации разработчиков (EF – environmental factor) вычисляется с учетом следующих показателей F_i (табл. 18).

Таблица 18. Показатели уровня квалификации разработчиков

Показатель	Описание	Вес
F1	Знакомство с технологией	1,5
F2	Опыт разработки приложений	0,5
F3	Опыт использования объектно-ориентированного подхода	1
F4	Наличие ведущего аналитика	0,5
F5	Мотивация	1
F6	Стабильность требований	2
F7	Частичная занятость	-1
F8	Сложные языки программирования	-1

Каждому показателю F_i присваивается значение в диапазоне от 0 до 5. Для показателей F1 – F4 0 означает отсутствие, 3 – средний уровень, 5 – высокий уровень. Для показателя F5 0 означает отсутствие мотивации, 3 – средний уровень, 5 – высокий уровень мотивации. Для F6 0 означает высокую нестабильность требований, 3 – среднюю, 5 – стабильные требования. Для F7 0 означает отсутствие специалистов с частичной занятостью, 3 – средний уровень, 5 – все специалисты с частичной занятостью. Для показателя F8 0 означает простой язык программирования, 3 – среднюю сложность, 5 – высокую сложность.

Значение EF вычисляется по следующей формуле:

$$EF = 1,4 + \left(-0,03 \times \left(\sum F_i \times \text{Вес}_i \right) \right). \quad (9)$$

1.2.5 Оценка трудоемкости проекта

С учетом показателей технической сложности (TCF) и уровня квалификации разработчиков (EF) окончательное значение указателя вариантов использования UCP (use case points) определяется по формуле

$$UCP = UUCP \times TCF \times EF. \quad (10)$$

Пересчет величины UCP в человеко-часы требует определения количества человеко-часов, приходящихся на одну

UCP (коэффициент K_{UCP}). Этот показатель зависит от принятой на предприятии методики определения вариантов использования (каким критериям должен удовлетворять отдельно выделяемый вариант использования), квалификации разработчиков, используемых инструментальных средств, накопленного на предприятии опыта производства ПО. Его значение является стандартом данного предприятия.

В работе [1] в качестве начального значения предлагается использовать 20 человеко-часов на одну UCP (коэффициент $K_{UCP} = 20$, эта величина отражает норматив, принятый на конкретном предприятии/фирме). Эта величина может уточняться с учетом опыта разработчиков. Приведем пример возможного уточнения.

Рассмотрим показатели F1 – F8 и определим, сколько показателей F1 – F6 имеют значение меньше 3 и сколько показателей F7, F8 имеют значение больше 3. Если общее количество меньше или равно 2, следует использовать 20 чел.-ч. на одну UCP, если 3 или 4 – 28. Если общее количество равно 5 или более, следует внести изменения в сам проект, в противном случае риск провала слишком высок.

Для системы регистрации получаем 28 чел.-ч на одну UCP, таким образом, общее количество человеко-часов на весь проект равно $56,56 \times 28 = 1583,68$, что составляет 40 недель при 40-часовой рабочей неделе. Допустим, что команда разработчиков состоит из четырех человек, и добавим 3 недели на различные непредвиденные ситуации, тогда в итоге получим 13 недель на весь проект.

Опытные данные компании Rational Проект среднего размера (приблизительно 10 разработчиков, более чем 6 – 8 месяцев) может включать приблизительно 30 вариантов использования. Это соответствует тому, что средний вариант использования содержит 12 UCP, и каждая UCP требует 20 – 30 ч. Это означает общую трудоемкость 240 – 360 чел.-ч на вариант использования. Таким образом, 30 вариантов использования потребуют приблизительно 9000 чел.-ч (10 разработчиков в течение 6 месяцев). Однако прямой пропорции не существует: очень большой проект со 100

разработчиками и сроком 20 месяцев не начнется с 1000 вариантов использования из-за проблем размерности.

Использование описанной выше методики для простых и сложных систем хорошо согласуется с опытными данными компании Rational (приблизительно 150 – 350 ч на один вариант использования).

Самая простая система (весовой показатель $UC = 5$, $A = 2$, $UUCP = 7$) дает (при 20 чел.-ч на UCP) приблизительно 140 чел.-ч. Сложная система (весовой показатель $UC = 15$, $A = 3$, $UUCP = 18$) дает приблизительно 360 чел.-ч.

Полное время разработки вычисляется по формуле (4) с учетом данных таблицы 13.

1.3 Модель композиции приложения

Модель композиции является одной из конструктивных моделей стоимости COSOMO II.

Параметры данной модели определялись на основе статистического анализа реальных результатов большого количества проектов.

Модель композиции используется на ранней стадии разработки ПО, когда:

- рассматривается макетирование пользовательских интерфейсов;
- обсуждается взаимодействие ПО и компьютерной системы;
- оценивается производительность;
- определяется степень зрелости технологии.

Модель композиции приложения ориентирована на применение объектных указателей.

Объектный указатель – средство косвенного измерения ПО, для его расчета определяется количество экранов (как элементов пользовательского интерфейса), отчетов и компонентов, требуемых для построения приложения. Как показано в табл. 19, каждый объектный экземпляр (экран, отчет) относят к одному из трех уровней сложности. Здесь места подстановки измеренных и вычисленных значений отмечены прямоугольниками

(прямоугольник играет роль метки-заполнителя). В свою очередь, сложность является функцией от параметров клиентских и серверных таблиц данных (табл. 20 и 21), которые требуются для генерации экрана и отчета, а также от количества представлений и секций, входящих в экран или отчет.

Таблица 19. Оценка количества объектных указателей

Тип объекта		Кол-во	ВЕС			Итого
			Низкий	Средний	Высокий	
1	Экран	<input type="checkbox"/>	<input type="checkbox"/> x1	<input type="checkbox"/> x2	<input type="checkbox"/> x3	= <input type="checkbox"/>
2	Отчет	<input type="checkbox"/>	<input type="checkbox"/> x2	<input type="checkbox"/> x5	<input type="checkbox"/> x8	= <input type="checkbox"/>
3	3GL-компонент	<input type="checkbox"/>			<input type="checkbox"/> x10	= <input type="checkbox"/>
Объектные указатели (OP)						= <input type="checkbox"/>

Таблица 20. Оценка сложности экрана

Экраны	Количество серверных (срв) и клиентских (клт) таблиц данных		
Количество представлений	Всего < 4 (<2 срв, <3 клт)	Всего < 8 (2-3 срв, 3-5 клт)	Всего > 8 (>3 срв, >5 клт)
<3	Простой	Простой	Средний
3-7	Простой	Средний	Сложный
>8	Средний	Сложный	Сложный

Таблица 21. Оценка сложности отчета

Отчеты	Количество серверных (срв) и клиентских (клт) таблиц данных		
Количество представлений	Всего < 4 (<2 срв, <3 клт)	Всего < 8 (2-3 срв, 3-5 клт)	Всего > 8 (>3 срв, >5 клт)
0 или 1	Простой	Простой	Средний
2 или 3	Простой	Средний	Сложный
>4	Средний	Сложный	Сложный

После определения сложности количество экранов, отчетов и компонентов взвешивается в соответствии с табл. 19. Количество

объектных указателей определяется перемножением исходного числа объектных экземпляров на весовые коэффициенты и последующим суммированием промежуточных результатов.

Для учета реальных условий разработки вычисляется процент повторного использования программных компонентов %REUSE и определяется количество новых объектных указателей NOP:

$$NOP = (\text{Объектные указатели}) \times [(100 - \%REUSE)/100]. \quad (11)$$

Для оценки трудозатрат, основанной на величине NOP, надо знать скорость разработки продукта PROD. Эту скорость определяют по табл. 22, учитывающей уровень опытности разработчиков и зрелость среды разработки.

Таблица 22. Оценка скорости разработки

Опытность/возможности разработчика	Зрелость/возможности среды разработки	PROD
Очень низка	Очень низка	4
Низкая	Низкая	7
Номинальная	Номинальная	13
Высокая	Высокая	25
Очень высокая	Очень высокая	50

Проектные трудозатраты T оцениваются по формуле

$$T = NOP / PROD \text{ [чел. –мес]},$$

где PROD — производительность разработки, выраженная в терминах объектных указателей.

Полное время разработки вычисляется по формуле (4) с учетом данных таблицы 13.

В более развитых моделях дополнительно учитывается множество масштабных факторов, формирователей затрат, процедур поправок.

1.4 Методы, основанные на экспертных оценках

Подходы, основанные на экспертных оценках, применяются при отсутствии дискретных эмпирических данных. Они используют опыт и знания экспертов-практиков в различных областях. На основе экспертных оценок были разработан метод Дельфи.

Метод Дельфи используется для принятия решений по спорным вопросам. На предварительном этапе участники дискуссии должны без обсуждения с другими ответить на ряд вопросов, относительно их мнения по спорному вопросу. Затем ответы обобщаются, табулируются и возвращаются каждому участнику дискуссии для проведения второго этапа, на котором участникам снова предстоит дать свою оценку спорного вопроса, но на этот раз, располагая мнениями других участников, полученными на первом этапе. Второй этап завершается сужением и выделением круга мнений, отражающих некоторую общую оценку проблемы.

Изначально в методе Дельфи коллективное обсуждение не использовалось, обсуждение между этапами метода было впервые применено в обобщенном методе Дельфи.

Контрольные вопросы

1. Назовите методы оценки трудоемкости создания ПО.
2. Какие факторы наиболее значительно влияют на трудоемкость создания ПО?
3. Назовите два основных способа оценки размера программного проекта.
4. Какие виды информационных характеристик выделяют в методике функциональных указателей?
5. Какие факторы учитываются в методике оценки трудоемкости разработки ПО на основе вариантов использования?
6. Какие объектные указатели выделяются в модели композиции приложения?

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Пример 1. Рассмотрим порядок выполнения практико-ориентированного задания на примере:

Задание. Необходимо оценить трудоемкость создания в среде Delphi 5 программного продукта «Учет доверенностей» (ПП Учет доверенностей), предназначенного для автоматизации процесса учета на предприятии доверенностей на получение материальных ценностей (прил. В), *используя методику функциональных указателей.*

Исходные данные.

Функции ПП Учет доверенностей:

7. ввод и накопление информации о выданных доверенностях;
8. формирование отчетов об использовании доверенностей;
2. визуализация результатов отчетов.

Задачи ПП Учет доверенностей:

- Оформление новых доверенностей;
- Просмотр информации об использованных доверенностях;
- Просмотр информации о неиспользованных доверенностях;
- Ведение справочников материально ответственных лиц, материалов и организаций;
- Формирование отчетов по использованию доверенностей;
- Ведение журналов доверенностей.

Ввод и накопление информации. Данные о выданных и использованных доверенностях вводятся с клавиатуры. В базе данных должна храниться информация обо всех выданных доверенностях и об их использовании.

Состав входной информации:

- отчет о выданных доверенностях;
- отчет об использованных доверенностях;
- отчет об неиспользованных доверенностях;
- справка «Должники по доверенностям»;
- справка о материально ответственных лицах.

Структура и состав баз данных. База данных программы включает в себя следующие таблицы (табл. 23).

Таблица 23. Состав базы данных

№	Наименование	Количество полей
1	Выданные доверенности	10-15
2	Паспорта	<10
3	Организации	<10
4	Материальные ценности	<10

Расчет трудозатрат на разработку программного продукта «Учет доверенностей»

Выделенные согласно методике информационные характеристики показаны в таблице 24.

Таблица 24. Исходные данные для расчета

Наименование	Число элементов данных	Ранг
Внешние вводы:		
1. Экран для ввода, редактирования и просмотра информации о выданных доверенностях.	15	3
2. Экран для работы со справочником доверенных лиц.	< 10	3
3. Экран для работы со справочником организаций	< 10	3
4. Экран для работы со справочником материалов	< 10	3
Внешние выходы:		
1. Отч. Доверенности	< 10	4
2. Отч. Доверенность	< 10	4
3. Отч. Должники	< 10	4
3. Отч. Исползованные	< 10	4
4. Отч. Возвращенные	< 10	4
5. Экран О программе	< 10	4
Внешние запросы:		
1. Запр. Должники	< 10	3

Наименование	Число элементов данных	Ранг
2. Запр. Использованные	< 10	3
3. Запр. Возвращенные	< 10	3
Внутренние логические файлы:		
1. Табл. Доверенности	10-15	7
2. Табл. Дов. лица	< 10	7
3. Табл. Организации	< 10	7
4. Табл. Материалы	< 10	7
Общее количество рангов		73

Количество функциональных указателей вычисляется по формуле (1) с учетом данных таблиц 24 и 25.

$$FP = \text{Общее кол.} - \text{во рангов} \times (0,65 + 0,01 \times \sum F_i),$$

$$FP = 73 \times (0,65 + 0,01 \times 1) = 48,18,$$

т. е. $FP = 48,18$.

Таблица 25. Значения системных параметров приложения

№	Системный параметр	Значение (F_i)
1	Передача данных	0
2	Распределенная обработка данных	0
3	Производительность обработки	0
4	Эксплуатационные ограничения	0
5	Частота транзакций	0
6	Оперативный ввод данных	0
7	Эффективность работы	1
8	Оперативное обновление	0
9	Сложность обработки	0
10	Повторная используемость	0
11	Простота установки	0
12	Простота эксплуатации	0
13	Разнообразные условия размещения	0
14	Простота изменений	0
	Итого ($\sum F_i$)	1

Полученная FP-оценка пересчитывается в LOC-оценку V (табл. 11), учитывая, что ПП создается с использованием среды Delphi 5

$$V = K_{\text{яз}} \times FP = 48,18 \times 18 = 0,867 \text{ KSLOC}.$$

Согласно формуле (2) и табл. 12 трудоемкость создания ПП составляет

$$T = N1 \times KSLOC^{N2} = 3,0 \times 0,867^{1,12} = 2,56 \text{ чел. –мес.}$$

Время разработки ПП составляет согласно (4) и табл. 13 составляет

$$t_{\text{разр}} = 2,5 \times T^{N3} = 2,5 \times 2,56^{0,35} = 3,47 \text{ мес.}$$

Результаты расчета:

- 1) Трудозатраты на разработку проекта составят 2,6 чел.-мес.
- 2) Время разработки составит 3,47 мес.

Расчет трудоемкости разработки с использованием среды 1С.

LOC-оценка V при использовании среды 1С

$$V = K_{\text{яз}} \times FP = 48,18 \times 10 = 0,482 \text{ KSLOC}.$$

Согласно формуле (2) и табл. 12 трудоемкость создания ПП составляет

$$T = N1 \times KSLOC^{N2} = 3,0 \times 0,482^{1,12} = 1,33 \text{ чел. –мес.}$$

Время разработки ПП составляет согласно (4) и табл. 13 составляет

$$t_{\text{разр}} = 2,5 \times T^{N3} = 2,5 \times 1,33^{0,35} = 2,76 \text{ мес.}$$

Пример 2. Трудоемкость разработки для учебного заведения системы регистрации студентов

Задание. Необходимо оценить трудоемкость разработки для учебного заведения системы регистрации студентов на дополнительные платные курсы [2].

Исходные данные

Система должна позволять:

студентам – регистрироваться на курсы и просматривать свои таблицы успеваемости с персональных компьютеров, подключенных к локальной сети университета;

профессорам – указать курсы, которые они будут читать, и предоставлять оценки за курсы.

В настоящее время в университете функционирует база данных, содержащая всю информацию о курсах (каталог курсов).

При регистрации на курсы в начале каждого семестра студенты запрашивают каталог курсов, содержащий список курсов, предлагаемых в данном семестре, с указанием имени профессора, наименования кафедры и требований к предварительному уровню подготовки (прослушанным курсам).

Студент может выбрать четыре курса в предстоящем семестре и указать два альтернативных курса на тот случай, если какой-либо из выбранных им курсов окажется уже заполненным или отмененным. После завершения процесса регистрации студента регистратор направляет информацию в расчетную систему. В конце семестра студенты могут просмотреть свои таблицы успеваемости.

Расчет трудозатрат на разработку программного продукта «Система регистрации»

Список действующих лиц показан в таблице 26.

Таблица 26. Действующие лица (actors)

Действующее лицо	Тип	Весовой коэффициент k_{ai}
Студент	Сложное	3
Профессор	Сложное	3
Регистратор	Сложное	3

Расчетная система	Простое	1
Каталог курсов	Простое	1

Таким образом, общий весовой показатель равен (5)

$$A = 2 \times 1 + 3 \times 3 = 11.$$

Для системы регистрации сложность вариантов использования определяется следующим образом (табл. 27).

Таблица 27. Варианты использования

Вариант использования	Тип	Весовой коэффициент k_{vi}
Войти в систему	Простой	5
Зарегистрироваться на курсы	Средний	10
Просмотреть таблицу успеваемости	Простой	5
Выбрать курсы для преподавания	Средний	10
Проставить оценки	Простой	5
Ввести информацию о профессорах	Простой	5
Ввести информацию о студентах	Простой	5
Закрыть регистрацию	Средний	10

Таким образом, общий весовой показатель равен (6)

$$UC = 5 \times 5 + 10 \times 3 = 45.$$

В результате получаем показатель UUCP (7)

$$UUCP = A + UC = 56.$$

Определим показатели технической сложности системы регистрации (табл. 28).

Вычислим показатель технической сложности проекта TCF (8)

$$TCF = 0,6 + (0,01 \times 40) = 1,0$$

Таблица 28. Показатели технической сложности системы регистрации

Показатель	Вес	Значение	Значение с учетом веса
T1	2	4	8
T2	1	3	3
T3	1	5	5
T4	1	1	1
T5	1	0	0
T6	0,5	5	2,5
T7	0,5	5	2,5
T8	2	0	0
T9	1	4	4
T10	1	5	5
T11	1	3	3
T12	1	5	5
T13	1	1	1
Итого			40

Определим показатели уровня квалификации разработчиков системы регистрации (табл. 29).

Таблица 29. Показатели уровня квалификации разработчиков

Показатель	Вес	Значение	Значение с учетом веса
F1	1,5	1	1,5
F2	0,5	1	0,5
F3	1	1	1
F4	0,5	4	2
F5	1	5	5
F6	2	3	6
F7	-1	0	1,5
F8	-1	3	0,5
Итого			13

Вычислим показатель уровня квалификации разработчиков EF (9)

$$EF = 1,4 + (-0,03 \times 13) = 1,01.$$

В результате получаем окончательное значение UCP (use case points):

$$UCP = UUCP \times TCF \times EF = 56 \times 1,0 \times 1,01 = 56,56$$

Используя данные п. 1.2.5 для системы регистрации, получаем, что на реализацию одной UCP, требуется 28 чел.-ч. Таким образом, общее количество человеко-часов на весь проект равно

$$T = 56,56 \times 28 = 1583,68 \text{ чел.} - \text{ч} = 9,43 \text{ чел.} - \text{мес.}$$

Время разработки ПП составляет согласно (4) и табл. 13

$$t_{\text{разр}} = 2,5 \times T^{N3} = 2,5 \times 9,43^{0,35} = 5,48 \text{ мес.}$$

Результаты расчета:

- 1) Трудозатраты на разработку проекта составят 9,43 чел.-мес.
- 2) Время разработки составит 5,48 мес.

Пример 3. Трудоемкость создания программного продукта «Учет доверенностей» согласно методике композиции приложения

Задание. Необходимо оценить трудоемкость создания программного продукта «Учет доверенностей» (ПП Учет доверенностей), предназначенного автоматизации процесса учета на предприятии доверенностей на получение материальных ценностей (прил. В), используя методику композиции приложения.

Исходные данные

Использовать исходные данные примера 1.

Расчет трудоемкости разработки по методике композиции приложения

Определим количество экранов (как элементов пользовательского интерфейса) и отчетов (табл. 30).

Таблица 30. Количество экранов и отчетов

Наименование	Число представлений	Число таблиц	Ранг
Экраны:			
1. Экран для ввода, редактирования и просмотра информации о выданных доверенностях.	1	<4	1
2. Экран для работы со справочником доверенных лиц.	1	<4	1
3. Экран для работы со справочником организаций	1	<4	1
4. Экран для работы со справочником материалов	1	<4	1
5. Экран Отч. Доверенности	1	<4	1
6. Экран Отч. Доверенность	1	<4	1
7. Экран Отч. Должники	1	<4	1
8. Экран Отч. Исползованные	1	<4	1
9. Экран Отч. Возвращенные	1	<4	1
Отчеты:			
1. Отч. Доверенности	1	<4	2
2. Отч. Доверенность	1	<4	2
3. Отч. Должники	1	<4	2
4. Отч. Исползованные	1	<4	2
5. Отч. Возвращенные	1	<4	2
Общее количество рангов (объектных указателей)			19

Таким образом, количество объектных указателей NOP равно

$$NOP = 19.$$

Предположим номинальную зрелость среды разработки (табл. 22), т. е. $PROD = 13$ получим, что проектные трудозатраты T оцениваются в

$$T = NOP / PROD = 19 / 7 = 2,71 \text{ [чел. – мес]}.$$

Соответственно время разработки ПП составляет согласно (4) и табл. 13 составляет

$$t_{\text{разр}} = 2,5 \times T^{N3} = 2,5 \times 2,71^{0,35} = 3,54 \text{ мес.}$$

Результаты расчета:

- 1) Трудозатраты на разработку проекта составят 2,71 чел. · мес.
- 2) Время разработки составит 3,54 мес.

Варианты

Необходимо оценить трудоемкость создания в среде 1С программного продукта:

1. Форма «Товарно транспортная накладная»;
2. Форма «Поступление товаров»
3. Форма «Акт выполненных работ»
4. Форма «Приходный кассовый ордер»
5. Форма «Расходная Накладная»
6. Форма «Расходный кассовый ордер»
7. Форма «Авансовый отчет»
8. Форма «Поступление на расчетный счет»
9. Форма «Поступление на расчетный счет»
10. Форма «Счет фактура»

ФОРМА ОТЧЕТА ПО ПРАКТИКО-ОРИЕНТИРОВАННОМУ ЗАДАНИЮ

На выполнение практико-ориентированного задания отводится 6 академических часов: 5 часов на выполнение и сдачу практико-ориентированного задания и 1 час на подготовку отчета.

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

- Структура отчета (на отдельном листе(-ах)): титульный лист, оглавление, цели и задачи выполнения практико-ориентированного задания, формулировка задания (вариант), этапы анализа предметной области (пример формы рассматриваемого процесса), подробное изложение этапов оценки трудоемкости на основе трех рассмотренных методов (записывается расчетная формула и численное выражение, полученное после постановки численных значений входящих в формулу параметров и переменных); вывод, содержащий итоговое изложение полученных трех результатов; список литературы.

Список литературы

1. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем: учебник для студ. экон. вузов, обучающихся по спец. "Прикл. информатика (по обл.)" и "Прикл. математика и информатика" / Вендров А. М. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 543 с.: ил.
2. Вендров, А. М. Практикум по проектированию программного обеспечения экономических информационных систем: учебное пособие для вузов / Вендров А. М. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 191 с.: ил.
3. Орлов, С. А. Технологии разработки программного обеспечения. Современный курс по программной инженерии: учебник для вузов / Орлов С. А., Цилькер Б. Я. – 4-е изд. – Санкт-Петербург [и др.]: Питер, 2012. – 608 с.: ил. (Стандарт третьего поколения).

Приложения

Приложение А. Коэффициенты регулировки сложности

Значения F_i	F_n
	F1 – Передача данных
0	Полностью пакетная обработка на локальном ПК
1	Пакетная обработка, удаленный ввод данных или удаленная печать
2	Пакетная обработка, удаленный ввод данных и удаленная печать
3	Сбор данных в режиме «онлайн» или дистанционная обработка, связанная с пакетным процессом
4	Несколько внешних интерфейсов, один тип коммуникационного протокола
5	Несколько внешних интерфейсов, более одного типа коммуникационного протокола
	F2 – распределенная обработка данных
0	Передача данных или процессов между компонентами системы отсутствует
1	Приложение готовит данные для обработки на ПК конечного пользователя
2	Данные готовятся для передачи, затем передаются и обрабатываются на другом компоненте системы (не на ПК конечного пользователя)
3	Распределенная обработка и передача данных в режиме «онлайн» только в одном направлении
4	Распределенная обработка и передача данных в режиме «онлайн» в обоих направлениях
5	Динамическое выполнение процессов в любом подходящем компоненте системы
	F3 – Производительность
0	К системе не предъявляется специальных требований, касающихся производительности
1	Требования к производительности определены, но не требуется никаких специальных действий
2	Время реакции или пропускная способность являются критическими в пиковые периоды. Не требуется никаких специальных решений относительно использования ресурсов процессора. Обработка может быть завершена в течение следующего рабочего дня
3	Время реакции или пропускная способность являются критическими в обычное рабочее время. Не требуется никаких специальных решений относительно использования ресурсов процессора. Время обработки ограничено взаимодействующими системами

Значения F_i	F_n
4	То же, кроме того, пользовательские требования к производительности достаточно серьезны, чтобы ее необходимо было анализировать на стадии проектирования
5	То же, кроме того, на стадиях проектирования, разработки и (или) реализации для удовлетворения пользовательских требований к производительности используются специальные средства анализа
	F4 – Эксплуатационные ограничения
0	Какие-либо явные или неявные ограничения отсутствуют
1	Эксплуатационные ограничения присутствуют, но не требуют никаких специальных усилий
2	Должны учитываться некоторые ограничения, связанные с безопасностью или временем реакции
3	Должны учитываться конкретные требования к процессору со стороны конкретных компонентов приложения
4	Заданные эксплуатационные ограничения требуют специальных ограничений на выполнение приложения в центральном или выделенном процессоре
5	То же, кроме того, специальные ограничения затрагивают распределенные компоненты системы
	F5 – Частота транзакций
0	Пиковых периодов не ожидается
1	Ожидаются пиковые периоды (ежемесячные, ежеквартальные, ежегодные)
2	Ожидаются еженедельные пиковые периоды
3	Ожидаются ежедневные пиковые периоды
4	Высокая частота транзакций требует анализа производительности на стадии проектирования
5	То же, кроме того, на стадиях проектирования, разработки и (или) внедрения необходимо использовать специальные средства анализа производительности
	F6 – Ввод данных в режиме «онлайн»
0	Все транзакции обрабатываются в пакетном режиме
1	От 1% до 7% транзакций требуют интерактивного ввода данных
2	От 8% до 15% транзакций требуют интерактивного ввода данных
3	От 16% до 23% транзакций требуют интерактивного ввода данных
4	От 24% до 30% транзакций требуют интерактивного ввода данных
5	Более 30% транзакций требуют интерактивного ввода данных

Значения F_i	F_n
	F7 – Эффективность работы конечных пользователей¹
0	Ни одной из перечисленных функциональных возможностей
1	От одной до трех функциональных возможностей
2	От четырех до пяти функциональных возможностей
3	Шесть или более функциональных возможностей при отсутствии конкретных пользовательских требований к эффективности
4	То же, кроме того, пользовательские требования к эффективности требуют специальных проектных решений для учета эргономических факторов (например, минимизации нажатий клавиш, максимизации значений по умолчанию, использования шаблонов)
5	То же, кроме того, пользовательские требования к эффективности требуют применения специальных средств и процессов, демонстрирующих их выполнение
	F8 – Онлайнное обновление
0	Отсутствует
1	Онлайнное обновление от одного до трех управляющих файлов. Объем обновлений незначителен, восстановление несложно
2	Онлайнное обновление четырех или более управляющих файлов. Объем обновлений незначителен, восстановление несложно
3	Онлайнное обновление основных внутренних логических файлов
4	То же, плюс необходимость специальной защиты от потери данных
5	То же, кроме того, большой объем данных требует учета затрат на процесс восстановления. Требуются автоматизированные процедуры восстановления с минимальным вмешательством

¹ Эффективность работы конечных пользователей определяется наличием следующих функциональных возможностей.

- Средства навигации (например, функциональные клавиши, динамически генерируемые меню).
- Меню.
- Онлайнные подсказки и документация.
- Автоматическое перемещение курсора.
- Скроллинг.
- Удаленная печать.
- Предварительно назначенные функциональные клавиши.
- Выбор данных на экране с помощью курсора.
- Использование видеоэффектов, цветового выделения, подчеркивания и других индикаторов.
- Всплывающие окна.
- Минимизация количества экранов, необходимых для выполнения бизнес-функций.
- Поддержка двух и более языков.

Значения F_i	F_n
	оператора
	F9 – Сложная обработка²
0	Ни одной из перечисленных функциональных возможностей
1	Любая одна из возможностей
2	Любые две из возможностей
3	Любые три из возможностей
4	Любые четыре из возможностей
5	Все пять возможностей
	F10 – Повторное использование
0	Отсутствует
1	Повторное использование кода внутри одного приложения
2	Не более 10% приложений будут использоваться более чем одним пользователем
3	Более 10% приложений будут использоваться более чем одним пользователем
4	Приложение оформляется как продукт и (или) документируется для облегчения повторного использования. Настройка приложения
5	выполняется пользователем на уровне исходного кода. То же, с возможностью параметрической настройки приложений
	F11 – Простота установки
0	К установке не предъявляется никаких специальных требований.
1	Для установки требуется специальная процедура
2	Заданы пользовательские требования к конвертированию (переносу существующих данных и приложений в новую систему) и установке, должны быть обеспечены и проверены соответствующие руководства. Конвертированию не придается важное значение
3	То же, однако конвертированию придается важное значение.
4	То же, что и в случае 2, плюс наличие автоматизированных средств конвертирования и установки
5	То же, что и в случае 3, плюс наличие автоматизированных средств конвертирования и установки
	F12 – Простота эксплуатации
0	К эксплуатации не предъявляется никаких специальных требований, за исключением обычных процедур резервного

² Сложная обработка характеризуется наличием у приложения следующих функциональных возможностей:

- повышенная реакция на внешние воздействия и (или) специальная защита от внешних воздействий;
- экстенсивная логическая обработка;
- экстенсивная математическая обработка;
- обработка большого количества исключительных ситуаций;
- поддержка разнородных типов входных/выходных данных.

Значения F_i	F_n
	копирования
1-4	Приложение обладает одной, несколькими или всеми из перечисленных далее возможностей. Каждая возможность, за исключением второй, обладает единичным весом: 1) наличие процедур запуска, копирования и восстановления с участием оператора; 2) то же, без участия оператора; 3) минимизируется необходимость в монтировании носителей для резервного копирования; 4) минимизируется необходимость в средствах подачи и укладки бумаги при печати
5	Вмешательство оператора требуется только при запуске и завершении работы системы. Обеспечивается автоматическое восстановление работоспособности приложения после сбоев и ошибок
	F13 – Количество возможных установок на различных платформах
0	Приложение рассчитано на установку у одного пользователя
1	Приложение рассчитано на много установок для строго стандартной платформы (технические средства + программное обеспечение)
2	Приложение рассчитано на много установок для платформ с близкими характеристиками
3	Приложение рассчитано на много установок для различных платформ
4	То же, что в случаях 1 или 2, плюс наличие документации и планов поддержки всех установленных копий приложения
5	То же, что в случае 3, плюс наличие документации и планов поддержки всех установленных копий приложения
	F14 – Простота изменений (Гибкость)³
0	Ни одной из перечисленных возможностей
1	Любая одна из возможностей
2	Любые две из возможностей
3	Любые три из возможностей
4	Любые четыре из возможностей
5	Все пять возможностей

³ Гибкость характеризуется наличием у приложения следующих возможностей:

- поддержка простых запросов, например, логики и (или) в применении только к одному ILF (вес – 1);
- поддержка запросов средней сложности, например, логики и (или) в применении более чем к одному ILF (вес – 2);
- поддержка сложных запросов, например, комбинации логических связей и (или) в применении к одному или более ILF (вес – 3);
- управляющая информация хранится в таблицах, поддерживаемых пользователем в интерактивном режиме, однако эффект от ее изменений проявляется на следующий рабочий день;
- то же, но эффект проявляется немедленно (вес – 2).

Приложение Б. Пример отчета по лабораторной работе

Задание. Оценить трудоемкость создания в среде Delphi 5 программного продукта «Учет доверенностей» (ПП Учет доверенностей), предназначенного для автоматизации процесса учета на предприятии доверенностей на получение материальных ценностей (прил. В), используя методику функциональных указателей.

Исходные данные.

Функции ПП Учет доверенностей:

1. ввод и накопление информации о выданных доверенностях;
2. формирование отчетов об использовании доверенностей.
3. визуализация результатов отчетов.

Задачи ПП Учет доверенностей:

- Оформление новых доверенностей;
- Просмотр информации об использованных доверенностях;
- Просмотр информации о неиспользованных доверенностях;
- Ведение справочников материально ответственных лиц, материалов и организаций;
- Формирование отчетов по использованию доверенностей;
- Ведение журналов доверенностей.

Ввод и накопление информации. Данные о выданных и использованных доверенностях вводятся с клавиатуры. В базе данных должна храниться информация обо всех выданных доверенностях и об их использовании.

Состав входной информации:

- отчет о выданных доверенностях;
- отчет об использованных доверенностях;
- отчет об неиспользованных доверенностях;
- справка «Должники по доверенностям»;
- справка о материально ответственных лицах.

Структура и состав баз данных. База данных программы включает в себя следующие таблицы (табл. Б.1).

Таблица Б.1. Состав базы данных

№	Наименование	Количество полей
1	Выданные доверенности	10-15
2	Паспорта	< 10
3	Организации	< 10
4	Материальные ценности	< 10

Расчет трудозатрат на разработку программного продукта «Учет доверенностей»

Выделенные согласно методике информационные характеристики показаны в таблице Б.2, а значения системных параметров приложения в таблице Б.3.

Таблица Б.2. Исходные данные для расчета

Наименование	Число элементов данных	Ранг
Внешние вводы:		
1. Экран для ввода, редактирования и просмотра информации о выданных доверенностях.	15	3
2. Экран для работы со справочником доверенных лиц.	< 10	3
3. Экран для работы со справочником организаций	< 10	3
4. Экран для работы со справочником материалов	< 10	3
Внешние выходы:		
1. Отч. Доверенности	< 10	4
2. Отч. Доверенность	< 10	4
3. Отч. Должники	< 10	4
6. Отч. Использованные	< 10	4
7. Отч. Возвращенные	< 10	4
8. Экран О программе	< 10	4
Внешние запросы:		
1. Запр. Должники	< 10	3
2. Запр. Использованные	< 10	3
3. Запр. Возвращенные	< 10	3
Внутренние логические файлы:		

Наименование	Число элементов данных	Ранг
1. Табл. Доверенности	10-15	7
2. Табл. Дов. лица	< 10	7
3. Табл. Организации	< 10	7
4. Табл. Материалы	< 10	7
Общее количество рангов		73

Таблица В.3. Значения системных параметров приложения

№	Системный параметр	Значение (F_i)
1	Передача данных	0
2	Распределенная обработка данных	0
3	Производительность обработки	0
4	Эксплуатационные ограничения	0
5	Частота транзакций	0
6	Оперативный ввод данных	0
7	Эффективность работы	1
8	Оперативное обновление	0
9	Сложность обработки	0
10	Повторная используемость	0
11	Простота установки	0
12	Простота эксплуатации	0
13	Разнообразные условия размещения	0
14	Простота изменений	0
	Итого ($\sum F_i$)	1

Количество функциональных указателей вычисляется по формуле (1).

$$FP = \text{Общее кол.} - \text{во рангов} \times \left(0,65 + 0,01 \times \sum F_i \right),$$

$$FP = 73 \times (0,65 + 0,01 \times 1) = 48,18,$$

т. е. $FP = 48,18$.

Полученная FP-оценка пересчитывается в LOC-оценку V (табл. 11), учитывая, что ПП создается с использованием среды Delphi 5

$$V = K_{\text{яз}} \times FP = 48,18 \times 18 = 0,867 \text{ KSLOC}.$$

Согласно формуле (2) и табл. 12 трудоемкость создания ПП составляет

$$T = N1 \times KSLOC^{N2} = 3,0 \times 0,867^{1,12} = 2,56 \text{ чел. –мес.}$$

Время разработки ПП составляет согласно (4) и табл. 13 составляет

$$t_{\text{разр}} = 2,5 \times T^{N3} = 2,5 \times 2,56^{0,35} = 3,47 \text{ мес.}$$

Результаты расчета:

- 1) Трудозатраты на разработку проекта составят 2,6 чел.-мес.
- 2) Время разработки составит 3,47 мес.

Приложение Б. Бланк доверенности (Типовая межотраслевая форма N М-2а)

Организация _____

Форма по ОКУД

по ОКПО

Коды
0315002

Доверенность № _____

Дата выдачи " ____ " _____ 20 ____ г.

Доверенность действительна по " ____ " _____ 20 ____ г.

(наименование потребителя и его адрес)

(наименование плательщика и его адрес)

Счет № _____ в _____
(наименование банка)

Доверенность выдана _____
(должность) (фамилия, имя, отчество)

Паспорт: серия _____ № _____

Кем выдан _____

Дата выдачи " ____ " _____ 20 ____ г.

На получение от _____
(наименование поставщика)

материальных ценностей по _____
(наименование, номер и

дата документа)

Оборотная сторона формы № М-2а

Номер по порядку	Материальные ценности	Единица измерения	Количество (прописью)
1	2	3	4

Подпись лица, получившего доверенность
удостоверяем.

Руководитель

(подпись)

(расшифровка подписи)

М.П.

Главный бухгалтер

(подпись)

(расшифровка подписи)