

Министерство образования и науки РФ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

Миньков С.Л.

## ПРОГРАММНАЯ ИНЖЕНЕРИЯ

### Лабораторный практикум Часть 2



Томск 2014

**Миньков С.Л.**

Программная инженерия. Лабораторный практикум. Часть 2: учебное пособие  
– Томск: ТУСУР, 2014. – 40 с.

Содержит методические указания к выполнению лабораторных работ по дисциплине «Программная инженерия» для направления подготовки бакалавров 230700 «Прикладная информатика». Рекомендуется для всех форм обучения, в том числе с применением дистанционных технологий.

© Миньков С.Л., 2014

Иллюстрация на обложке: [http://ru.123rf.com/photo\\_7066570\\_software-word-collage-on-white-background.html](http://ru.123rf.com/photo_7066570_software-word-collage-on-white-background.html)

## СОДЕРЖАНИЕ

Лабораторная работа 1 АВТОМАТИЗАЦИЯ РАСЧЕТА ТЕХНИКО-ЭКОНОМИЧЕСКОГО ОБОСНОВАНИЯ ПРОЕКТА .....	5
1.1 Оценка конкурентоспособности проекта в сравнении с аналогом .....	5
1.2 Планирование комплекса работ по разработке темы и оценка трудоемкости ...	7
1.3 Расчет затрат на разработку проекта .....	10
1.4 Расчет эксплуатационных затрат .....	13
1.5 Расчет показателей экономической эффективности.....	16
Задание.....	18
Лабораторная работа 2. КОНСТРУКТИВНАЯ МОДЕЛЬ СТОИМОСТИ СОСОМО.....	19
2.1. СОСОМО .....	19
2.2. СОСОМО II.....	22
Задание.....	27
Приложение (справочное) Некоторые примеры автоматизации расчета СОСОМО.....	29
1. Интерфейс онлайн-калькулятора .....	29
2. Интерфейс программы-калькулятора Costar 7.0 .....	30

## **ВВЕДЕНИЕ**

Экономика разработки программных продуктов – одна из составляющих программной инженерии. В главе 23 «Оценка стоимости программного продукта» классического учебника И. Соммервилла «Инженерия программного обеспечения» [1] подчеркивается, что менеджеры проекта должны уметь отвечать на вопросы:

- 1) Какие затраты необходимы для выполнения проекта?
- 2) Сколько это займет времени?
- 3) Какова стоимость выполнения проекта?

Во второй части лабораторного практикума по дисциплине «Программная инженерия», предназначенного для студентов, обучающихся по направлению подготовки бакалавров «Прикладная информатика», предлагаются лабораторные работы по изучению методов оценки трудовых и временных затрат на разработку программных продуктов (в частности, модели Б. Бозма) с дальнейшей автоматизацией расчета этих затрат и оценки экономической эффективности проекта.

При составлении лабораторного практикума использован опыт преподавания дисциплин «Разработка и применение программных средств и информационных технологий» и «Программная инженерия» студентам факультета систем управления Томского государственного университета систем управления и радиоэлектроники и факультета инновационных технологий Национального исследовательского Томского государственного университета.

# Лабораторная работа 1

## АВТОМАТИЗАЦИЯ РАСЧЕТА ТЕХНИКО-ЭКОНОМИЧЕСКОГО ОБОСНОВАНИЯ ПРОЕКТА

*Цель работы: изучить структуру технико-экономического обоснования эффективности проекта по разработке автоматизированной информационной системы на основе сравнения с аналогом и автоматизировать процесс расчета ТЭО.*

Технико-экономическое обоснование (ТЭО) – это анализ, расчет, оценка экономической целесообразности осуществления предлагаемого проекта, в данном случае – проекта по разработке автоматизированной информационной системы. ТЭО основано на сопоставительной оценке затрат и результатов, установлении эффективности использования, срока окупаемости вложений.

Технико-экономическое обоснование является необходимым для каждого инвестора исследованием, в ходе подготовки которого проводится ряд работ по изучению и анализу всех составляющих инвестиционного проекта и разработке сроков возврата вложенных в бизнес средств.

В качестве примера рассмотрено технико-экономическое обоснование разработки автоматизированной системы контроля договоров на поставку материально-технических ресурсов на предприятии ООО «Стройсервис» [2].

### 1.1 Оценка конкурентоспособности проекта в сравнении с аналогом

В качестве программы-аналога для сравнения при разработке проекта принята программа «Dogo Pro» (производитель – компания «InterBuild»).

Эта разработка принята в качестве базового варианта исходя из трех факторов:

- 1) смежный профиль;
- 2) соответствие требованиям технического задания проекта;
- 3) доступность для исследования и сравнения с разрабатываемым проектом реальной версии программы.

Для оценки конкурентоспособности разрабатываемого продукта необходимо провести анализ и сравнение с выбранным аналогом по функциональному назначению, основным техническим и эксплуатационным параметрам, областям применения. Подобный анализ осуществляется с помощью оценки эксплуатационно-технического уровня разрабатываемого продукта.

Эксплуатационно-технический уровень (ЭТУ) разрабатываемого продукта – это обобщенная характеристика его эксплуатационных свойств, возможностей, степени новизны, являющихся основой качества продукта. Для определения ЭТУ продукта можно использовать индекс эксплуатационно-технического уровня  $J_{ЭТУ}$ , который рассчитывается как сумма частных

индексов, куда входят показатели качества программного продукта. Для учета значимости отдельных параметров применяется балльно-индексный метод.

Тогда

$$J_{\text{ЭТУ}} = \sum_{j=1}^n B_j \times X_j, \quad (1.1)$$

где  $J_{\text{ЭТУ}}$  – комплексный показатель качества продукта по группе показателей;

$n$  – число рассматриваемых показателей;

$B_j$  – коэффициент весомости  $j$ -го показателя в долях единицы, назначаемый в соответствии с потребностями организации-заказчика программного продукта;

$X_j$  – экспертная оценка  $j$ -го показателя качества по выбранной шкале оценивания.

В таблице 2.1 представлены результаты расчета балльно-индексным методом при пятибалльной шкале оценивания. Показатели качества выбираются в соответствии с деревом характеристик качества программного изделия (стандарт «ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению» (рис. 1.1).

Таблица 1.1 – Расчет показателя качества балльно-индексным методом

Показатели качества	Коэффициент весомости, $B_j$	Проект		Аналог	
		$X_j$	$B_j \times X_j$	$X_j$	$B_j \times X_j$
1. Удобство работы (пользовательский интерфейс)	0,14	4	0,56	2	0,28
2. Новизна (соответствие современным требованиям)	0,1	4	0,4	3	0,3
3. Соответствие профилю деятельности заказчика	0,2	4	0,8	2	0,4
4. Ресурсная эффективность	0,05	4	0,2	4	0,2
5. Надежность (защита данных)	0,13	3	0,39	3	0,39
6. Скорость доступа к данным	0,1	4	0,4	4	0,4
7. Гибкость настройки	0,06	3	0,18	3	0,18
8. Обучаемость персонала	0,13	5	0,65	1	0,13
9. Соотношение стоимость/возможности	0,09	4	0,36	2	0,18
Обобщенный показатель качества $J_{\text{ЭТУ}}$		$J_{\text{ЭТУ1}}=3,94$		$J_{\text{ЭТУ2}}=2,46$	

Отношение двух найденных индексов называют коэффициентом технического уровня (КТУ)  $A_k$  первого программного продукта по отношению ко второму:

$$A_k = \frac{J_{\text{ЭТУ1}}}{J_{\text{ЭТУ2}}}. \quad (1.2)$$

В нашем случае  $A_k = \frac{3,94}{2,46} = 1,60. \quad (1.3)$

Так как КТУ больше 1, то разработка проекта с технической точки зрения оправдана.

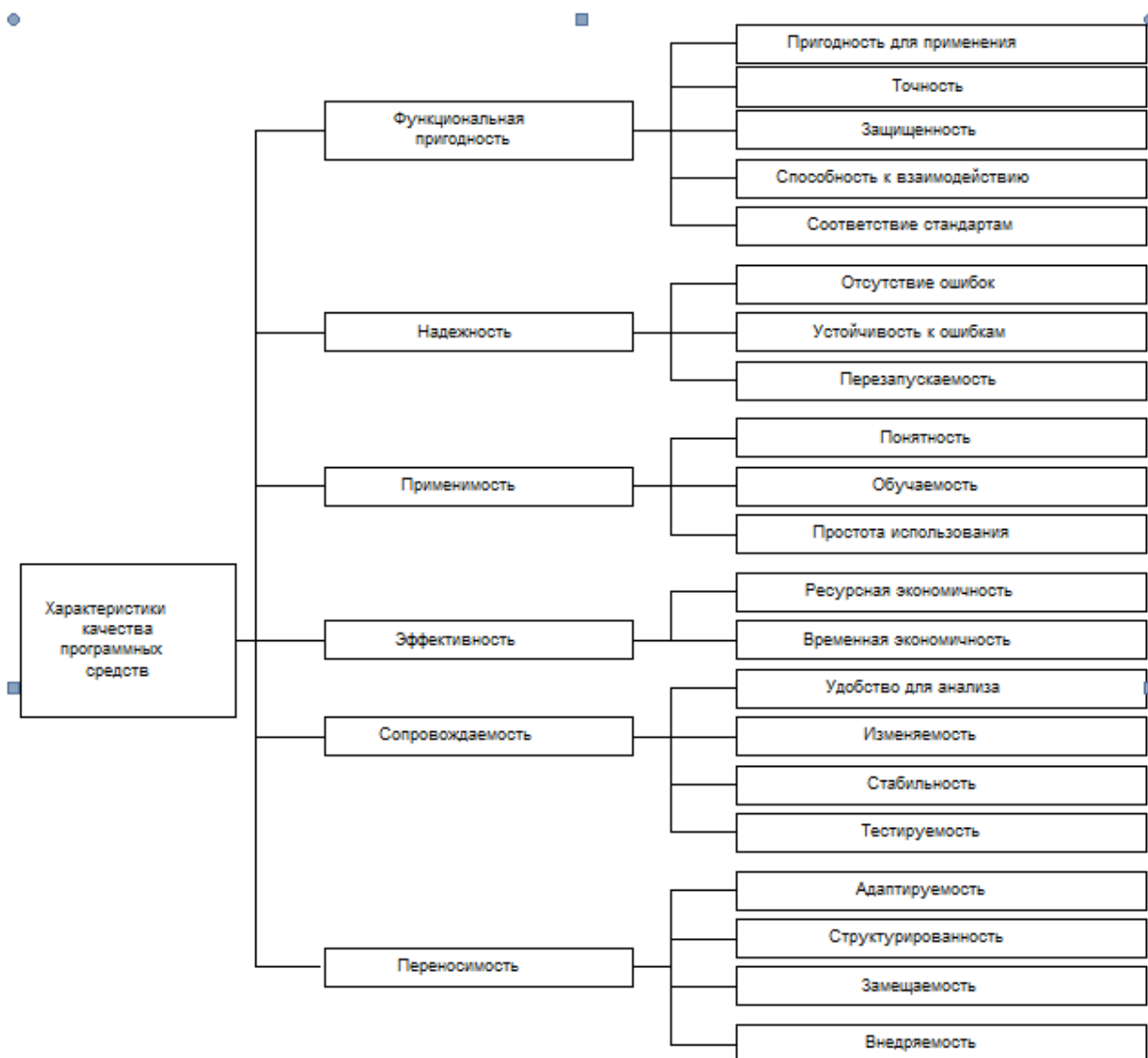


Рисунок 1.1 - Дерево характеристик качества программного изделия

## 1.2 Планирование комплекса работ по разработке темы и оценка трудоемкости

Для разработки было задействовано два человека: руководитель проекта и исполнитель (инженер-программист). Руководитель выполняет постановку задачи, курирует ход работ и дает необходимые консультации при разработке системы. Исполнитель отвечает за проектирование информационного обеспечения, разработку структур баз данных, реализацию вычислительных алгоритмов в виде заверченного продукта, разработку интерфейсных блоков и отладку программы.

Выбор комплекса работ по разработке проекта производится в соответствии со стандартом «ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств»,

устанавливающим стадии разработки программных продуктов, и приведен в таблице 1.1.

Таблица 1.2 – Комплекс работ по разработке проекта

Содержание работ	Исполнители	Длительность, дни	Загрузка	
			дни	%
<b>1. Подготовка процесса разработки и анализ требований</b>				
1.1 Исследование и обоснование разработки				
1.1.1 Постановка задачи	Руководитель Программист	3	1 3	33 100
1.1.2 Сбор исходных данных	Руководитель Программист	14	5 14	35 100
1.2 Поиск аналогов и прототипов				
1.2.1 Анализ существующих методов решения задачи и программных средств	Руководитель Программист	6	0 6	0 100
1.2.2 Обоснование принципиальной необходимости разработки	Руководитель Программист	2	1 2	50 100
1.3 Анализ требований				
1.3.1 Определение и анализ требований к проектируемой программе	Руководитель Программист	3	1 3	33 100
1.3.2 Определение структуры входных и выходных данных	Руководитель Программист	5	1 5	20 100
1.3.3 Выбор технических и программных средств реализации	Руководитель Программист	3	1 3	33 100
1.3.4 Согласование и утверждение технического задания	Руководитель Программист	3	1 3	33 100
<b>Итого по этапу 1</b>	Руководитель Программист	<b>39</b>	<b>11</b> <b>39</b>	28 100
<b>2. Проектирование</b>				
2.1 Проектирование программной архитектуры	Руководитель Программист	3	0 3	0 100
2.2 Техническое проектирование компонентов программы	Руководитель Программист	7	0 7	0 100
<b>Итого по этапу 2</b>	Руководитель Программист	<b>10</b>	<b>0</b> <b>10</b>	0 100
<b>3. Программирование и тестирование программных модулей</b>				
3.1 Программирование модулей в выбранной среде программирования	Руководитель Программист	13	0 13	0 100
3.2 Тестирование программных модулей	Программист Программист	21	0 21	0 100
3.3 Сборка и испытание программы	Руководитель Программист	5	2 5	40 100
3.4 Анализ результатов испытаний	Руководитель Программист	5	1 5	20 100
<b>Итого по этапу 3</b>	Руководитель	<b>44</b>	<b>3</b>	7



	Программист		<b>44</b>	100
<b>4. Оформление рабочей документации</b>				
4.1 Проведение расчетов показателей безопасности жизнедеятельности	Руководитель	3	0	0
	Программист		3	100
4.2 Проведение экономических расчетов	Руководитель	4	0	0
	Программист		4	100
4.3 Оформление пояснительной записки	Руководитель	15	5	33
	Программист		15	100
<b>Итого по этапу 4</b>	Руководитель	<b>22</b>	<b>5</b>	<b>23</b>
	Программист		<b>22</b>	<b>100</b>
<b>Итого по проекту</b>	Руководитель	<b>115</b>	<b>19</b>	<b>17</b>
	Программист		<b>115</b>	<b>100</b>

На основе данных таблицы 1.2 разработан календарный график выполнения работ (таблица 1.3), показывающий последовательность и взаимосвязь выполнения комплекса работ (с учетом воскресенья; можно брать без учета воскресенья и праздничных дней).

Таблица 1.3 - Календарный график выполнения работ

Содержание работы	Исполнители	Длительность, дни	График работ	
			Начало	Конец
1 Постановка задачи	Руководитель	1	21.01.2013	21.01.2013
	Программист	3	21.01.2013	23.01.2013
2 Сбор исходных данных	Руководитель	5	24.01.2013	28.01.2013
	Программист	14	24.01.2013	06.02.2013
3 Анализ существующих методов решения задачи и программных средств	Руководитель	0	-	-
	Программист	6	07.02.2013	12.02.2013
4 Обоснование принципиальной необходимости разработки	Руководитель	1	13.02.2013	13.02.2013
	Программист	2	13.02.2013	14.02.2013
5 Определение и анализ требований к программе	Руководитель	1	15.02.2013	15.02.2013
	Программист	3	15.02.2013	17.02.2013
6 Определение структуры входных и выходных данных	Руководитель	1	18.02.2013	18.02.2013
	Программист	5	18.02.2013	22.02.2013
7 Выбор технических средств и программных средств реализации	Руководитель	1	23.02.2013	23.02.2013
	Программист	3	23.02.2013	25.02.2013
8 Согласование и утверждение технического задания	Руководитель	1	26.02.2013	26.02.2013
	Программист	3	26.02.2013	28.02.2013
9 Проектирование программной архитектуры	Руководитель	0	-	-
	Программист	3	01.03.2013	03.03.2013
10 Техническое	Руководитель	0	-	-

проектирование компонентов программы	Программист	7	04.03.2013	10.03.2013
11 Программирование модулей в выбранной среде программирования	Руководитель	0	-	-
	Программист	13	11.03.2013	23.03.2013
12 Тестирование программных модулей	Руководитель	0	-	-
	Программист	21	24.03.2013	13.04.2013
13 Сборка и испытание программы	Руководитель	2	14.04.2013	15.04.2013
	Программист	5	14.04.2013	18.04.2013
14 Анализ результатов испытаний	Руководитель	1	19.04.2013	19.04.2013
	Программист	5	19.04.2013	23.04.2013
15 Проведение расчетов показателей безопасности жизнедеятельности	Руководитель	0	-	-
	Программист	3	24.04.2013	26.04.2013
16 Проведение экономических расчетов	Руководитель	0	-	-
	Программист	4	27.04.2013	30.04.2013
17 Оформление пояснительной записки	Руководитель	5	01.05.2013	05.05.2013
	Программист	15	01.05.2013	15.05.2013

### 1.3 Расчет затрат на разработку проекта

Капитальные вложения, связанные с автоматизацией обработки информации, рассчитываются по формуле

$$K = K_{\Pi} + K_p, \quad (1.4)$$

где  $K_{\Pi}$  - капитальные вложения на проектирование, руб.;

$K_p$  - капитальные вложения на реализацию проекта, руб.

Предпроизводственные затраты представляют собой единовременные расходы на разработку обеспечивающих или функциональных систем или элементов на всех этапах проектирования, а также затраты на их усовершенствование, т.е. на проведение обследования и обработку материалов исследования, разработку технического задания, разработку технического и рабочего проекта системы и ее опытного внедрения. Сюда включаются затраты на разработку алгоритмов и программ, стоимость разработок по привязке типовых проектных решений (ТПР) и пакетов прикладных программ (ППП) к конкретному объекту автоматизации.

Суммарные затраты на проектирование системы и ее разработку и отладку на компьютере определяются по формуле

$$K_{\Pi} = ((1 + W_d)(1 + W_c) + W_n) \sum_{i=1}^m Z_{oi} + C_M + M_e, \quad (1.5)$$

где  $m$  – количество работников, участвующих в разработке проекта;

$Z_{oi}$  – затраты на основную заработную плату работника  $i$ -й категории, руб.;

$W_d$  – коэффициент, учитывающий дополнительную заработную плату в долях к основной заработной плате ( $W_d = 0,4$  и состоит из коэффициента отпускных, равного  $0,1$ , и районного коэффициента –  $0,3$  для Томска);

$W_c$  – коэффициент, учитывающий отчисления на социальные нужды, в долях к сумме основной и дополнительной заработной платы разработчиков: *страховые взносы в Пенсионный фонд – 0,22, страховые взносы в ФСС – 0,029, страховые взносы в ФФОМС – 0,051, страховые взносы на производственный травматизм – 0,002; итого  $W_c=0,302$  (в соответствии с Федеральным законом 379-ФЗ от 03.12.2011г.).*

$W_n$  – коэффициент, учитывающий накладные расходы организации, в долях к основной заработной плате разработчиков (*принимается по фактическим данным,  $W_n = 0,6$* );

$C_M$  – затраты на материалы;

$M_в$  – затраты на использование машинного времени.

Затраты на основную заработную плату работника  $i$ -й категории:

$$Z_{oi} = Z_{днi} t_i, \quad (1.6)$$

где  $Z_{днi}$  – среднедневная заработная плата работника  $i$ -й категории, руб./дн.;

$t_i$  – количество дней, отработанных работником  $i$ -й категории.

Затраты времени на разработку системы по каждому исполнителю принимаются, исходя из его загрузки по календарному графику выполнения работ (табл. 1.3).

Расчет основной заработной платы разработчиков проекта приведен в таблице 1.4 из расчета, что в месяце в среднем 21 рабочий день.

Таблица 1.4 – Основная заработная плата разработчиков

Должность	Должностной оклад, руб.	Средняя дневная ставка, руб.	Затраты времени на разработку, человеко-дней	ОЗП, руб.
Руководитель	19000	904,76	19	17190,48
Программист	7000	333,33	115	38333,33
Итого				55523,81

*Примечание. С сентября 2013 г. в бюджетных образовательных учреждениях введены следующие должностные оклады профессорско-преподавательского состава: преподаватель 14000 руб., доцент, канд.н. – 19000 руб., профессор, докт.н. – 24000 руб.*

Ввиду того, что проектируемая информационная система должна быть запрограммирована и отлажена с помощью компьютеров, к суммарным затратам на разработку добавляются затраты на использование машинного времени, исчисляемые как:

$$M_в = t_{мв} S_{мч} K_м, \quad (1.7)$$

где  $t_{мв}$  – машинное время компьютера, необходимое для разработки программного продукта;  $t_{мв} = 460$  час.;

$S_{мч}$  – стоимость 1 часа машинного времени:  $S_{мч}=20$  руб./час.; (*рассчитать или использовать среднюю стоимость платного доступа к ПК в вашем городе*);

$K_м$  – коэффициент мультипрограммности (*показывает долю машинного времени, отводимого непосредственно на работу над проектом*);  $K_м=1$ .

Материалы, приобретенные в процессе выполнения работы, и их стоимость приведены в таблице 1.5.

Таблица 1.5 - Затраты на материалы

Материалы	Единица измерения	Требуемое количество	Цена за единицу, руб.	Сумма, руб.
Тетрадь общая	шт.	1	10	10
Компакт-диск CD-RW	шт.	2	35	70
Тонер для лазерного принтера	шт.	1	1000	1000
Бумага офисная	пачка	1	120	120
Итого				1200

Таким образом, капитальные вложения на проектирование равны:

$$K_{\pi} = (17190,48 + 38333,33) \times ((1+0,4) \times (1 + 0,302) + 0,6) + 460 \times 20 \times 1 + 1200 \text{ руб.} = 144923,09 \text{ руб.}$$

Смета затрат на разработку представлена в таблице 1.6.

Таблица 1.6 – Затраты на разработку

Статьи затрат	Сумма, руб.
Основная заработная плата	55523,81
Дополнительная зарплата	22209,52
Отчисления на социальные нужды	23475,47
Затраты на материалы	1200
Затраты на машинное время	9200
Накладные расходы организации	33314,29
ИТОГО	144923,09

Капитальные вложения на реализацию проекта:

$$K_p = K_o + K_{зд} + K_{пп} + K_{св} + K_{иб} + K_{пк}, \quad (1.8)$$

где  $K_o$  – затраты на основное и вспомогательное оборудование, руб.;

$K_{зд}$  – затраты на строительство, реконструкцию здания и помещений, руб.;

$K_{пп}$  – затраты на приобретение типовых разработок, пакетов, руб.;

$K_{св}$  – затраты на прокладку линий связи, руб.;

$K_{иб}$  – затраты на создание информационной базы, руб.;

$K_{пк}$  – затраты на подготовку и переподготовку кадров, руб.

В связи с тем, что для внедрения системы, рассматриваемой в данном проекте, не было затрат связанных с прокладкой линии связи, затрат на основное и вспомогательное оборудование, затрат на реконструкцию и строительство зданий, то данные затраты для внедрения системы не учитывают. Также не принимаются в расчет затраты по подготовке и переподготовке кадров, затраты на создание информационной базы и затраты на приобретение типовых разработок.

Таким образом, при внедрении системы, рассматриваемой в данном проекте, затраты на его реализацию определяются затратами на оборудование и материалы. В оборудование и материалы входит компьютер на базе процессора Pentium-4. Стоимость компьютера 22500 руб.

Тогда затраты на основное и вспомогательное оборудование составят

$$K_o = \sum_{j=1}^n C_{bj} Q_j Y_j, \quad (1.9)$$

где  $C_{bj}$  – балансовая стоимость  $j$ -го вида оборудования, руб. (при  $n=1$   $C_{b1}=22500$  руб.);

$Q_j$  – количество единиц  $j$ -го оборудования, руб. (1 шт.);

$Y_j$  – коэффициент загрузки  $j$ -го вида оборудования при обработке информации по решению задач предметной области:

$$Y_j = \frac{T_j}{\Phi_{\text{эф}j}}, \quad (1.10)$$

где  $\Phi_{\text{эф}j}$  – эффективный годовой фонд времени работы технического средства  $j$ -го вида, час./год.

Время работы технического средства  $j$ -го вида по решению  $s$  задач, час./год:

$$T_j = \sum_{k=1}^s t_{kj} \times U_k, \quad (1.11)$$

где  $t_{kj}$  – трудоемкость однократной обработки информации по  $k$ -й задаче на  $j$ -м виде технических средств, часов машинного времени ( $t_{kj}=6$ );

$U_k$  – частота (периодичность) решения  $k$ -й задачи, дней /год ( $U_k=247$ ).

Затраты на реализацию:

$$K_p = 22500 \times 1 \times 6 \times 247 / (247 \times 8) \text{ руб.} = 16875 \text{ руб.}$$

Таким образом, суммарные затраты на разработку проекта:

$$K = K_n + K_p = 144923,09 + 16875 \text{ руб.} = 161798,09 \text{ руб.}$$

Суммарные затраты, связанные с внедрением аналога складываются из следующих затрат:

–затраты на приобретение программного продукта (37300 руб.);

–затраты по оплате услуг на установку и сопровождение продукта (12000 руб.);

–затраты на основное и вспомогательное оборудование (22500 руб.) (предполагается, что для внедрения аналога понадобится такой же компьютер, что и для проектируемой системы);

–затраты на подготовку пользователя (оплата курсов повышения квалификации, командировочные расходы и пр.) (9000 руб.).

(Постарайтесь расписать эти статьи расходов и обосновать их).

Итого суммарные затраты, связанные с внедрением аналога составят 80800 руб.

#### 1.4 Расчет эксплуатационных затрат

К эксплуатационным затратам относятся затраты, связанные с обеспечением нормального функционирования проекта. Эти затраты называют также текущими затратами. Это могут быть затраты на ведение информационной базы, эксплуатацию комплекса технических средств, эксплуатацию систем программно-математического обеспечения, реализацию

технологического процесса обработки информации по задачам, эксплуатация системы в целом.

Текущие затраты рассчитываются по формуле

$$Z_{\text{тек}} = Z_{\text{зп}} + C_a + Z_э + C_{\text{рем}} + Z_m + Z_n, \quad (1.12)$$

где  $Z_{\text{зп}}$  – затраты на зарплату основную и дополнительную с отчислениями во внебюджетные фонды, руб.;

$C_a$  – амортизационные отчисления от стоимости оборудования и устройств системы, руб.;

$Z_э$  – затраты на силовую энергию, руб.;

$C_{\text{рем}}$  – затраты на текущий ремонт оборудования и устройств системы, руб.;

$Z_m$  – затраты на материалы и машинные носители, руб.;

$Z_n$  – накладные расходы информационного отдела, руб.

Эксплуатацию разработанной системы осуществляют специалисты. Затраты на заработную плату основную и дополнительную с отчислениями на социальные нужды производственного персонала рассчитываются по формуле

$$C_{\text{зп}} = \sum_{i=1}^m (t_i z_i (1 + W_d)(1 + W_c)), \quad (1.13)$$

где  $t_i$  – время эксплуатации системы  $i$ -м работником, дни;

$z_i$  – среднедневная заработная плата  $i$ -го работника, руб./день.

Данные расчета заработной платы специалистов приведены в таблицах 1.7 и 1.8.

Таблица 1.7 – Данные по заработной плате специалистов (для проекта)

Должность	Должностной оклад, руб.	Средняя дневная ставка, руб./день	Затраты времени на эксплуатацию, человеко-дней	Фонд заработной платы, руб.
Сотрудник отдела МТС	15000	714,29	40	52080,0
Программист	10000	476,19	20	17360,0
Итого (с учетом начисл.)				69440,0

$$C_{\text{зп1}} = (40 \times 714,29 + 20 \times 476,19) \times 1,4 \times 1,302 \text{ руб.} = 69440,0 \text{ руб. (за год).}$$

Таблица 1.8 – Данные по заработной плате специалистов (для продукта-аналога)

Должность	Должностной оклад, руб.	Средняя дневная ставка, руб./день	Затраты времени на эксплуатацию, человеко-дней	Фонд заработной платы, руб.
Сотрудник отдела МТС	15000	714,29	60	78120
Программист	10000	476,19	40	21706,2
Итого (с учетом начисл.)				112840

$$C_{\text{зп2}} = (60 \times 714,29 + 40 \times 476,19) \times 1,4 \times 1,302 = 112840,0 \text{ руб. (за год).}$$

Сумма амортизационных отчислений рассчитывается следующим образом:

$$C_a = \sum_{j=1}^n \frac{C_{bj} a_j g_j t_j}{F_{эфj}}, \quad (1.14)$$

где  $C_{bj}$  – балансовая стоимость  $j$ -го вида оборудования, руб.;

$t_j$  – время работы  $j$ -го вида оборудования, час;

$F_{эфj}$  – эффективный фонд времени работы оборудования в год, час;

$a_j$  – норма годовых амортизационных отчислений для  $j$ -го вида оборудования;

$g_j$  – количество единиц оборудования  $j$ -го вида.

Эффективный фонд времени работы оборудования можно вычислить по формуле

$$F_{эф} = D_p \times H_3, \quad (1.15)$$

где  $D_p$  – количество рабочих дней в году.  $D_p = 247$  (в соответствии с производственным календарём на 2013 год);

$H_3$  – норматив среднесуточной загрузки, час./день,  $H_3 = 8$ .

Таким образом, эффективный фонд времени работы оборудования составит

$$F_{эф} = 247 \times 8 = 1976 \text{ час.}$$

Данные для расчета:

$a_j = 0,2$  (используется ускоренная амортизация – 20-30 %);

$g_j = 1$ ;

$t_j$  (для проекта) =  $(40 + 20) \times 8 = 480$  час.;

$t_j$  (для аналога) =  $(60 + 40) \times 8 = 800$  час.;

$C_{b1} = 22500$  руб.;  $C_{b2} = 22500$  руб.

Сумма амортизационных отчислений для проекта составит

$$C_{a1} = (22500 \times 0,2 \times 1 \times 480) / 1976 \text{ руб.} = 1093,12 \text{ руб.}$$

Сумма амортизационных отчислений для аналога составит

$$C_{a2} = (22500 \times 0,2 \times 1 \times 800) / 1976 \text{ руб.} = 1821,86 \text{ руб.}$$

Затраты на силовую энергию рассчитываются по формуле

$$Z_3 = \sum_{j=1}^n N_j t_j g_j T_3, \quad (1.16)$$

где  $N_j$  – установленная мощность  $j$ -го вида технических средств, кВт;

$t_j$  – время работы  $j$ -го вида технических средств, час;

$g_j$  – коэффициент использования установленной мощности оборудования;

$T_3$  – тариф на электроэнергию, руб./кВт ч.

В настоящее время тариф на электроэнергию на данной территории составляет 2,6 руб./кВт ч (указать конкретно, каждый год тариф меняется для каждой территории), установленная мощность для компьютера равна 0,4 кВт (укажите суммарную мощность используемого оборудования), таким образом затраты на силовую энергию для проекта составят  $Z_3 = 0,4 \times 1 \times 480 \times 2,6$  руб. = 499,20 руб., для аналога составят  $Z_3 = 0,4 \times 1 \times 800 \times 2,6$  руб. = 832,00 руб.

Затраты на текущий ремонт оборудования рассчитываются по формуле

$$Z_{\text{рем}} = \sum_{j=1}^n \frac{C_{pi} C_{bj} T_{pi}}{F \Delta \Phi_j}, \quad (1.17)$$

где  $C_{pi}$  – норматив затрат на ремонт ( $C_{pi} = 0,05$ ).

Затраты на текущий ремонт оборудования составят:

– для проекта  $Z_{\text{рем1}} = (0,05 \times 22500 \times 480) / 1976 = 273,28$  руб.,

– для аналога  $Z_{\text{рем2}} = (0,05 \times 22500 \times 800) / 1976 = 455,47$  руб.

Затраты на материалы, потребляемые в течение года, составляют 1 % от балансовой стоимости основного оборудования и равны 225 руб. ( $22500 \times 0,01$ ) для проекта и аналога.

Накладные расходы включают затраты на содержание административного и управленческого персонала, на содержание помещения и т.д. Норматив накладных расходов составляет 20 % от прямых затрат, включающих первые пять статей затрат, представленных в таблице 1.9.

Накладные расходы для проекта:

$Z_{\text{н1}} = (69440,00 + 1093,12 + 499,20 + 273,28 + 225) \times 0,2 = 14306,12$  руб.

Накладные расходы для аналога:

$Z_{\text{н2}} = (112840,00 + 1821,86 + 832,00 + 455,47 + 225) \times 0,2 = 23234,87$  руб.

Таблица 1.9 – Годовые эксплуатационные затраты

Статьи затрат	Затраты на проект, руб.	Затраты на аналог, руб.
Основная и дополнительная зарплата с отчислениями во	69440,00	112840,00
Амортизационные отчисления	1093,12	1821,86
Затраты на электроэнергию	499,20	832,00
Затраты на текущий ремонт	273,28	455,47
Затраты на материалы	225,00	225,00
Накладные расходы	14306,12	23234,87
<b>Итого</b>	<b>85836,72</b>	<b>139409,19</b>

## 1.5 Расчет показателей экономической эффективности

Оценка экономической эффективности вариантов проектных решений элементов АИС основывается на расчете показателей сравнительной экономической эффективности капитальных вложений. Годовой экономический эффект от использования разрабатываемой системы определяется по разности приведенных затрат на базовый (аналог) и новый (проектируемый) варианты в расчете на годовой объем выпуска:

$$\Delta = (Z_1 \times A_k - Z_2) \times N, \quad (1.18)$$

где  $Z_1, Z_2$  – приведенные затраты на единицу работ, выполняемых с помощью базового и проектируемого вариантов процесса обработки информации, руб.;

$A_k$  – коэффициент эксплуатационно-технической эквивалентности, или технического уровня,  $A_k = 1,60$  (формула (1.3));



$N$  – объем работ, выполняемых с помощью разрабатываемого продукта (примем равным 1).

Приведенные затраты  $Z_i$  на единицу работ, выполняемых по базовому и разрабатываемому вариантам, рассчитываются по формуле

$$Z_i = C_i + E_n \times K_i, \quad (1.19)$$

где  $C_i$  – себестоимость (текущие эксплуатационные затраты единицы работ), руб.;

$E_n$  – нормативный коэффициент экономической эффективности ( $E_n = 0,33$ );

$K_i$  – суммарные затраты, связанные с внедрением нового проекта.

Затраты на единицу работ по аналогу:

$$Z_1 = 139409,19 + 0,33 \times 80800 = 166073,19 \text{ руб.}$$

Затраты на единицу работ по проекту:

$$Z_2 = 85836,72 + 0,33 \times 161798,09 = 139230,08 \text{ руб.}$$

Экономический эффект от использования разрабатываемой системы:

$$\mathcal{E} = 166073,19 \times 1,60 - 139230,08 = 126757,06 \text{ руб.}$$

Сводные данные по расчету экономического эффекта приведены в таблице 1.10.

Таблица 1.10 – Экономический эффект

Характеристика	Значение	
	продукт-аналог	разрабатываемый продукт
Себестоимость (текущие эксплуатационные затраты), руб.	139409,19	85836,72
Суммарные затраты, связанные с внедрением проекта, руб.	80800,00	161798,09
Приведенные затраты на единицу работ, руб.	166073,19	139230,08
Экономический эффект от использования разрабатываемой системы, руб.	126757,06	

После определения годового экономического эффекта необходимо рассчитать срок окупаемости затрат на разработку продукта по формуле

$$T_{ок} = K/\mathcal{E}. \quad (1.20)$$

Срок окупаемости составит:  $T_{ок} = 161798,09 / 126757,06 = 1,28$  года.

Затем рассчитаем фактический коэффициент экономической эффективности разработки ( $E_f$ ) и сопоставим его с нормативным значением коэффициента эффективности капитальных вложений  $E_n = 0,33$ :

$$E_f = 1/T_{ок} = 1/1,28 = 0,78. \quad (1.21)$$

Фактический коэффициент экономической эффективности разработки получился больше, чем нормативный, поэтому разработка и внедрение разрабатываемого продукта является эффективной.

Таким образом, в ходе проделанной работы найдены все необходимые данные, доказывающие целесообразность и эффективность данной разработки. Приведем эти данные в сводной таблице 1.11.

Таблица 1.11 – Результаты экономического обоснования проекта

<b>Характеристика проекта</b>	<b>Значение</b>
Затраты на разработку и внедрение проекта, руб.	161798,09
Общие эксплуатационные затраты, руб.	85836,72
Экономический эффект, руб.	126757,06
Коэффициент экономической эффективности	0,78
Срок окупаемости, лет	1,28

В предлагаемых методических указаниях приведен расчет для модельных условий. В каждом конкретном случае следует учесть конкретные обстоятельства и сроки выполнения работы, действующие цены и нормативы.

### **Задание**

1. Выбрав инструментальную среду программирования, автоматизировать весь расчетный процесс технико-экономического обоснования, создав удобный пользовательский интерфейс, позволяющий проводить параметрические исследования.

2. Обратить особое внимание на то, чтобы значения для каждого входного данного для расчета ТЭО вводились в программе только один раз. Все используемые коэффициенты вводить в формулы не константами, а переменными, включив их в список входных данных.

В календарном графике работ (таблица 1.3) в качестве временных исходных данных вводить только дату начала работы.

3. Весь расчетный процесс предлагается разбить на пять расчетных блоков:

- 1) расчет КТУ;
- 2) расчет план-графика работ;
- 3) расчет затрат на разработку проекта и внедрение аналога;
- 4) расчет эксплуатационных затрат;
- 5) расчет показателей экономической эффективности.

4. Изменяя входные данные, проанализировать степень их влияния на итоговые результаты (табл. 1.11).

## Лабораторная работа 2. КОНСТРУКТИВНАЯ МОДЕЛЬ СТОИМОСТИ СОСОМО

*Цель работы: изучить алгоритмы различных уровней конструктивных моделей Б. Боэма и автоматизировать процесс расчета экономических показателей проекта по разработке программного продукта.*

### 2.1. СОСОМО

СОСОМО (Constructive Cost Model) – это конструктивная модель стоимости, разработанная в начале 80-х годов Барри Боэмом для оценки трудоемкости разработки программных продуктов<sup>1</sup>. Она основана на статистическом анализе фактических данных по выполнению 63 проектов в компании TRW Aerospace, где Барри Боэм был директором отдела исследований программного обеспечения и технологий. Анализировались проекты объемом от 2 до 100 тысяч строк кода, на языках программирования от ассемблеров до высокоуровневого языка PL/1, основанные на каскадной модели жизненного цикла разработки ПО.

Модель состоит из иерархии трех последовательно детализируемых и уточняемых уровней [3]. На каждом уровне все проекты разбиваются на три группы по уровню сложности:

- 1) распространенный тип (organic projects);
- 2) встроенный тип (embedded projects);
- 3) полунезависимый тип (semidetached projects).

**Распространенный тип** характеризуется тем, что проект выполняется небольшой группой специалистов, имеющих опыт в создании подобных изделий и опыт применения технологических средств. Условия работы стабильны, и изделие имеет относительно невысокую сложность.

**Встроенный тип** характеризуется очень жесткими требованиями на программный продукт, интерфейсы, параметры ЭВМ. Как правило, у таких изделий высокая степень новизны и планирование работ осуществляется при недостаточной информации, как о самом изделии, так и об условиях работы. Встроенный проект требует больших затрат на изменения и исправления.

**Полунезависимый тип** занимает промежуточное положение между распространенным и встроенным – это проекты средней сложности. Исполнители знакомы лишь с некоторыми характеристиками (или компонентами) создаваемой системы, имеют средний опыт работы с подобными изделиями, изделие имеет элемент новизны. Только часть требований к изделию жестко фиксируется, в остальном разработки имеют степени выбора.

Тип той или иной группы можно рассматривать как один из параметров модели СОСОМО.

---

<sup>1</sup> Опубликована в книге *Barry Boehm. Software Engineering Economics* (Englewood Cliffs, NJ: Prentice-Hall, 1981).

Рассмотрим уровни модели.

### 2.1.1 Базовый уровень (Basic COCOMO)

Модель этого уровня – двухпараметрическая. В качестве параметров выступают тип проекта и объем программы (число строк кода).

Уравнения базового уровня модели имеют вид:

$$PM = a_i \times (SIZE)^{b_i},$$
$$TM = c_i \times (PM)^{d_i},$$

где

*PM* (People×Month) – трудоемкость (чел.×мес.);

*TM* (Time at Month) – время разработки в календарных месяцах;

*SIZE* – объем программного продукта в тысячах строк исходного текста (Kilo of Source Line of Code – *KSLOC*).

Коэффициенты  $a_i$ ,  $b_i$ ,  $c_i$  и  $d_i$  выбираются из табл. 2.1.

Таблица 2.1. Значения коэффициентов базовой уровня модели COCOMO в зависимости от типа проекта

Тип проекта	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
Распространенный	2,4	1,05	2,5	0,38
Полунезависимый	3,0	1,12	2,5	0,35
Встроенный	3,6	1,20	2,5	0,32

Модель этого уровня подходит для ранней быстрой приблизительной оценки затрат, но точность её весьма низка, т.к. не учитываются такие факторы, как квалификация персонала, характеристики оборудования, опыт применения современных методов разработки программного обеспечения и современных инструментальных сред разработки и др.

### 2.1.2 Промежуточный уровень (Intermediate COCOMO)

На этом уровне базовая модель уточнена за счет ввода дополнительных 15 «атрибутов стоимости» (или факторов затрат) *Cost Drivers* ( $CD_k$ ), которые сгруппированы по четырем категориям:

– **Характеристики продукта (Product Attributes):**

- Требуемая надежность ПО (Required Software Reliability);
- Размер БД приложения (Size of Application Database);
- Сложность продукта (Complexity of the Product);

– **Характеристики аппаратного обеспечения (Hardware Attributes):**

- Ограничения быстродействия при выполнении программы (Run-Time Performance Constraints);
- Ограничения памяти (Memory Constraints);
- Неустойчивость окружения виртуальной машины (Volatility of the Virtual Machine Environment);
- Требуемое время восстановления (Required Turnabout Time);

- **Характеристики персонала (Personnel Attributes):**
  - Аналитические способности (Analyst Capability);
  - Способности к разработке ПО (Software Engineer Capability);
  - Опыт разработки (Applications Experience);
  - Опыт использования виртуальных машин (Virtual Machine Experience);
  - Опыт разработки на языках программирования (Programming Language Experience);
- **Характеристики проекта (Project Attributes):**
  - Использование инструментария разработки ПО (Use of Software Tools);
  - Применение методов разработки ПО (Application of Software Engineering Methods);
  - Требования соблюдения графика разработки (Required Development Schedule).

Значения каждого атрибута выбирается из табл. 2.2 в соответствии с его степенью значимости (рейтингом) в конкретном проекте.

Таблица 2.2. Значения атрибутов стоимости в зависимости от их уровня

Атрибуты стоимости, $CD_k$	Рейтинг					
	Очень низкий	Низкий	Средний	Высокий	Очень высокий	Критический
<b>Характеристики продукта</b>						
1. Требуемая надежность ПО	0,75	0,88	1,00	1,15	1,40	n/a
2. Размер БД приложения	n/a	0,94	1,00	1,08	1,16	n/a
3. Сложность продукта	0,70	0,85	1,00	1,15	1,30	1,65
<b>Характеристики аппаратного обеспечения</b>						
4. Ограничения быстродействия при выполнении программы	n/a	n/a	1,00	1,11	1,30	1,66
5. Ограничения памяти	n/a	n/a	1,00	1,06	1,21	1,56
6. Неустойчивость окружения виртуальной машины	n/a	0,87	1,00	1,15	1,30	n/a
7. Требуемое время восстановления	n/a	0,87	1,00	1,07	1,15	n/a
<b>Характеристики персонала</b>						
8. Аналитические способности	1,46	1,19	1,00	0,86	0,71	n/a
9. Опыт разработки	1,29	1,13	1,00	0,91	0,82	n/a
10. Способности к разработке ПО	1,42	1,17	1,00	0,86	0,70	n/a
11. Опыт использования виртуальных машин	1,21	1,10	1,00	0,90	n/a	n/a
12. Опыт разработки на языках программирования	1,14	1,07	1,00	0,95	n/a	n/a
<b>Характеристики проекта</b>						
13. Применение методов разработки ПО	1,24	1,10	1,00	0,91	0,82	n/a

14. Использование инструментария разработки ПО	1,24	1,10	1,00	0,91	0,83	n/a
15. Требования соблюдения графика разработки	1,23	1,08	1,00	1,04	1,10	n/a

Примечание: n/a (not available) – данные отсутствуют, т.е. соответствующий уровень не оценивается

**Формула промежуточного уровня модели** имеет вид

$$PM = EAF \times a_i \times (SIZE)^{b_i},$$

где  $PM$  – трудоемкость (чел.×мес.);

$SIZE$  – объем программного продукта в тысячах строк исходного текста (Kilo of Source Line of Code –  $KSLOC$ ).

$EAF$  (Effort Adjustment Factor) – произведение выбранных атрибутов стоимости из табл. 2:  $EAF = \prod_{k=1}^{15} CD_k$ .

Коэффициенты модели  $a_i$  и  $b_i$  выбираются из табл. 2.3.

Таблица 2.3. Значения коэффициентов промежуточного уровня модели COCOMO в зависимости от типа проекта

Тип проекта, $i$	$a_i$	$b_i$
1. Распространенный	3,2	1,05
2. Полунезависимый	3,0	1,12
3. Встроенный	2,8	1,20

Время разработки рассчитывается по той же формуле, что и для базовой модели.

### 2.1.3 Детальный уровень (Advanced COCOMO)

Повышает точность оценки за счет иерархической декомпозиции создаваемого ПО и учета стоимостных факторов на каждом уровне иерархии и по фазам работ (здесь не рассматривается).

## 2.2. COCOMO II

В 1997 методика была усовершенствована и получила название COCOMO II. Калибровка параметров производилась уже по 161 проекту разработки ПО.

Различаются две стадии оценки проекта: *предварительная* оценка на начальной фазе (Early Design) и *детальная* оценка после проработки архитектуры (Post Architecture).

Формула оценки трудоемкости проекта в чел.×мес. имеет вид:

$$PM = EAF \times A \times (SIZE)^E,$$

где  $E = B + 0,01 \times \sum_{j=1}^5 SF_j$ ;

$B = 0,91$ ;  $A = 2,94$  для предварительной оценки;  $A = 2,45$  для детальной оценки<sup>1</sup>;

$SF_j$  – факторы масштаба (Scale Factors) (табл. 5);

$SIZE$  – объем программного продукта в тысячах строк исходного текста (KSLOC – Kilo of Source Line of Code);

$EM_j$  – множители трудоемкости<sup>2</sup> (Effort Multiplier).  $n=7$  – для предварительной оценки (табл. 6),  $n=17$  – для детальной оценки (табл. 7);

$EAF$  (Effort Adjustment Factor) – произведение выбранных множителей трудоемкости:  $EAF = \prod_{k=1}^n EM_k$ .

## 2.2.1 Факторы масштаба (Scale Factors)

В методике COCOMO II используются пять факторов масштаба  $SF_j$ , описание которых приведено в табл.4.

Таблица 4. Описание уровней значимости факторов масштаба

$SF_j$	Описание	Уровень значимости фактора					
		Очень низкий	Низкий	Средний	Высокий	Очень высокий	Критический
1. PREC. Precedentedness.	Прецедентность, наличие опыта аналогичных разработок	опыт в продукте и платформе отсутствует	продукт и платформа не много знакомы	некоторый опыт в продукте и платформе присутствует	продукт и платформа в основном известны	продукт и платформа в большей степени знакомы	продукт и платформа полностью знакомы
2. FLEX. Development Flexibility	Гибкость процесса разработки	процесс строго детерминирован	допускаются некоторые компромиссы	значительная жесткость процесса	относительная жесткость процесса	незначительная жесткость процесса	определены только общие цели
3. RESL. Architecture / Risk Resolution	Архитектура и разрешение рисков	риски известны / проанализированы на 20%	риски известны / проанализированы на 40%	риски известны / проанализированы на 60%	риски известны / проанализированы на 75%	риски известны / проанализированы на 90%	риски разрешены на 100%

<sup>1</sup> Приведены параметры модели COCOMO II.2000 (Version 2.1). Источник: COCOMOII: Model Definition Manual. – USC, Center for Software Engineering. – URL:

<http://csse.usc.edu/csse/TECHRPTS/2000/usccse2000-500/usccse2000-500.pdf>

<sup>2</sup> Иное название: Cost Driver (CD) – атрибут стоимости (фактор затрат).

4. TEAM. Team Cohesion	Сработанность команды	формальные взаимодействия	тяжелое взаимодействие до некоторой степени	чаще всего коллективная работа	в основном коллективная работа	высокая степень взаимодействия	полное доверие, взаимозаменяемость и взаимопомощь
5. PMAT. Process Maturity	Зрелость процессов	СММ Уровень 1 (ниже среднего)	СММ Уровень 1 (выше среднего)	СММ Уровень 2	СММ Уровень 3	СММ Уровень 4	СММ Level 5

*Примечание. СММ (Capability Maturity Model) — пятиуровневая модель зрелости возможностей компании-разработчика ПО, предложенная SEI (Software Engineering Institute, США).*

Эти факторы применяются **на обеих стадиях оценки** проекта.

Числовые значения фактора масштаба в зависимости от оценки его уровня, приведены в таблице 5.

Таблица 5. Значение фактора масштаба в зависимости от оценки его уровня

Фактор масштаба, $SF_j$	Оценка уровня фактора					
	Very Low	Low	Nominal	High	Very High	Extra High
1. PREC	6,20	4,96	3,72	2,48	1,24	0,00
2. FLEX	5,07	4,05	3,04	2,03	1,01	0,00
3. RESL	7,07	5,65	4,24	2,83	1,41	0,00
4. TEAM	5,48	4,38	3,29	2,19	1,10	0,00
5. PMAT	7,80	6,24	4,68	3,12	1,56	0,00

### 2.2.2 Множители трудоемкости (Effort Multipliers)

Количество и значения множителей трудоёмкости отличаются для разных стадий оценки проекта.

1. Стадия **предварительной оценки трудоемкости программного проекта (Early Design)**. Для этой оценки необходимо оценить для проекта уровень **семи множителей** трудоемкости  $EM_j$ :

– **параметры персонала:**

1. PERS (Personnel Capability) – квалификация персонала (Extra Low – аналитики и программисты имеют низшую квалификацию, текучесть больше 45%; Extra High – аналитики и программисты имеют высшую квалификацию, текучесть меньше 4%);

2. PREX (Personnel Experience) – опыт персонала (Extra Low – новое приложение, инструменты и платформа; Extra High – приложение, инструменты и платформа хорошо известны);

– **параметры продукта:**

3. RCPX (Product Reliability and Complexity) – сложность и надежность продукта (Extra Low – продукт простой, специальных требований по



надежности нет, БД маленькая, документация не требуется; Extra High – продукт очень сложный, требования по надежности жесткие, БД сверхбольшая, документация требуется в полном объеме);

4. RUSE (Developed for Reusability) – разработка для повторного использования (Low – не требуется; Extra High – предполагается переиспользование в других продуктах);

– **параметры платформы:**

5. PDIF (Platform Difficulty) – сложность платформы разработки (Extra Low – специальные ограничения по памяти и быстродействию отсутствуют, платформа стабильна; Extra High – жесткие ограничения по памяти и быстродействию, платформа нестабильна);

– **параметры проекта:**

6. FCIL (Facilities)– оборудование (Extra Low – инструменты простейшие, коммуникации затруднены; Extra High – интегрированные средства поддержки жизненного цикла, интерактивные мультимедиа коммуникации);

7. SCED (Required Development Schedule) – требуемое выполнение графика работ (Very Low – 75% от номинальной длительности; Very High – 160% от номинальной длительности).

Значения множителей трудоемкости в зависимости от их уровня приведены в табл. 6.

Таблица 6. Значения множителей трудоемкости в зависимости от оценки их уровня (Early Design)

№	Множитель трудоёмкости, $EM_i$	Оценка уровня множителя трудоемкости						
		Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
1	PERS	2,12	1,62	1,26	1,00	0,83	0,63	0,50
2	PREX	1,59	1,33	1,22	1,00	0,87	0,74	0,62
3	RCPX	0,49	0,60	0,83	1,00	1,33	1,91	2,72
4	RUSE	n/a	n/a	0,95	1,00	1,07	1,15	1,24
5	PDIF	n/a	n/a	0,87	1,00	1,29	1,81	2,61
6	FCIL	1,43	1,30	1,10	1,00	0,87	0,73	0,62
7	SCED	n/a	1,43	1,14	1,00	1,00	n/a	n/a

Примечание: n/a (not available) – данные отсутствуют, т.е. соответствующий уровень не оценивается

2. Стадия **детальной оценки после проработки архитектуры (Post Architecture)**. Для этой оценки необходимо оценить для проекта уровень **семнадцати множителей** трудоемкости  $EM_j$ :

– **параметры персонала:**

- 1) Analyst Capability (ACAP) – возможности аналитика;
- 2) Applications Experience (AEXP) – опыт разработки приложений;
- 3) Programmer Capability (PCAP) – возможности программиста;
- 4) Personnel Continuity (PCON) – продолжительность работы персонала;
- 5) Platform Experience (PEXP) – опыт работы с платформой;

6) Language and Tool Experience (LTEX) – опыт использования языка программирования и инструментальных средств.

– **параметры продукта:**

7) Required Software Reliability (RELY) – требуемая надежность программы;

8) Database Size (DATA) – размер базы данных;

9) Software Product Complexity (CPLX) – сложность программы;

10) Required Reusability (RUSE) – требуемая возможность многократного использования;

11) Documentation Match to Life-Cycle Needs (DOCU) – соответствие документации потребностям жизненного цикла.

– **параметры платформы:**

12) Execution Time Constraint (TIME) – ограничения времени выполнения;

13) Main Storage Constraint (STOR) – ограничения памяти;

14) Platform Volatility (PVOL) – изменяемость платформы.

– **параметры проекта:**

15) Use of Software Tools (TOOL) – использование инструментальных программных средств;

16) Multisite Development (SITE) – многоабонентская (удаленная) разработка;

17) Required Development Schedule (SCED) – требуемое выполнение графика работ.

Значения множителей трудоемкости в зависимости от их уровня приведены в табл. 7.

Таблица 7. Значения множителей трудоемкости в зависимости от оценки их уровня (Post Architecture)

№	Effort Multiplier, $EM_j$		Very Low	Low	Nominal	High	Very High	Extra High
	<i>Personnel Factors</i>							
1	ACAP	Analyst Capability	1,42	1,29	1,00	0,85	0,71	n/a
2	AEXP	Applications Experience	1,22	1,10	1,00	0,88	0,81	n/a
3	PCAP	Programmer Capability	1,34	1,15	1,00	0,88	0,76	n/a
4	PCON	Personnel Continuity	1,29	1,12	1,00	0,90	0,81	n/a
5	PEXP	Platform Experience	1,19	1,09	1,00	0,91	0,85	n/a
6	LTEX	Language and Tool Experience	1,20	1,09	1,00	0,91	0,84	n/a
<i>Product Factors</i>								
7	RELY	Required Software Reliability	0,84	0,92	1,00	1,10	1,26	n/a
8	DATA	Database Size	n/a	0,23	1,00	1,14	1,28	n/a
9	CPLX	Software Product Complexity	0,73	0,87	1,00	1,17	1,34	1,74
10	RUSE	Required Reusability	n/a	0,95	1,00	1,07	1,15	1,24
11	DOCU	Documentation Match to Life-	0,81	0,91	1,00	1,11	1,23	n/a

Cycle Needs								
<i>Platform Factors</i>								
12	TIME	Execution Time Constraint	n/a	n/a	1,00	1,11	1,29	1,63
13	STOR	Main Storage Constraint	n/a	n/a	1,00	1,05	1,17	1,46
14	PVOL	Platform Volatility	n/a	0,87	1,00	1,15	1,30	n/a
<i>Project Factors</i>								
15	TOOL	Use of Software Tools	1,17	1,09	1,00	0,90	0,78	n/a
17	SITE	Multisite Development	1,22	1,09	1,00	0,93	0,86	0,80
16	SCED	Required Development Schedule	1,43	1,14	1,00	1,00	1,00	n/a

Примечание: n/a (not available) – данные отсутствуют, т.е. соответствующий уровень не оценивается

### 2.2.3 Оценка длительности проекта

Время разработки проекта  $TM$  в методике COCOMO II для обоих уровней рассчитывается по формуле:

$$TM = SCED \times C \times (PM_{NS})^{D+0,2 \times (E-B)},$$

где  $C = 3,67$ ;  $D = 0,28$ ;

$PM_{NS}$  – рассчитанная трудоемкость проекта без учета множителя  $SCED$ , определяющего сжатие расписания.

Остальные параметры определены выше.

#### Задание

1. Используя выбранную инструментальную среду разработки, создать программное приложение «Калькулятор COCOMO» для расчета трудоёмкости и сроков разработки программного продукта на основе различных конструктивных моделей стоимости COCOMO с удобным пользовательским интерфейсом.

2. Исследовать влияние объема программного кода ( $SIZE$ ) на трудоемкость ( $PM$ ) и время разработки проекта ( $TM$ ) для разных уровней и версий COCOMO.

2. Получить значения  $PM$  и  $TM$  по всем моделям для одного и того же значения параметра  $SIZE$ , выбрав номинальный (средний) уровень сложности проекта. Повторить расчет для простого проекта и проекта, имеющего высокую степень новизны.

3. Результаты исследований оформить графически.

## ЛИТЕРАТУРА

1. Соммервилл, Иан. Инженерия программного обеспечения, 6-е изд. : Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 624 с.
2. Миньков С.Л. Технико-экономическое обоснование выполнения проекта: методическое пособие. – Томск: ТУСУР, 2014. – 30 с. [Электронный ресурс]. – Режим доступа:  
[http://asu.tusur.ru/learning/spec080801/d26/s080801\\_d26\\_work.docx](http://asu.tusur.ru/learning/spec080801/d26/s080801_d26_work.docx)
- 3 Миньков С.Л. Разработка и применение ППП в экономике: Учебное пособие. – Томск: ТМЦДО, 2002. – 231 с.

## Приложение (справочное)

### Некоторые примеры автоматизации расчета COCOMO

#### 1. Интерфейс онлайн-калькулятора

Copyright © 2008 Ray Madachy (Naval Postgraduate School)

На сайте Центра системного и программного инжиниринга Университета Южной Каролины (США) (USC Center for Systems and Software Engineering) можно посмотреть интерфейс интернет-калькулятора **COCOMO Suite of Constructive Cost Models** (<http://csse.usc.edu/tools/COCOMOSuite.php>).

The screenshot shows the web interface for the COCOMO Suite of Constructive Cost Models. The browser window title is "COCOMO Suite of Constructive Cost Models - Windows Internet Explorer". The address bar shows the URL "http://csse.usc.edu/tools/COCOMOSuite.php". The page features the USC CSSE logo and a navigation menu. The main content area is titled "COCOMO Suite of Constructive Cost Models". It includes a "Model(s)" dropdown menu set to "COCOMO", with "Monte Carlo Risk" and "Auto Calculate" options both set to "Off". Under "Software Size", the "Sizing Method" is "Source Lines of Code". There are input fields for "New", "Reused", and "Modified" software size, with associated percentage fields for "Design Modified", "Code Modified", "Integration Required", "Assessment and Assimilation", "Software Understanding", and "Unfamiliarity". Below are sections for "Software Scale Drivers" and "Software Cost Drivers", each with multiple "Nominal" dropdown menus for various project factors like "Precedentedness", "Architecture / Risk Resolution", "Process Maturity", "Development Flexibility", "Team Cohesion", "Required Software Reliability", "Analyst Capability", "Time Constraint", "Data Base Size", "Programmer Capability", "Storage Constraint", "Product Complexity", "Personnel Continuity", "Platform Volatility", "Developed for Reusability", "Application Experience", "Platform Experience", "Documentation Match to Lifecycle Needs", "Language and Toolset Experience", "Multisite Development", and "Required Development Schedule".

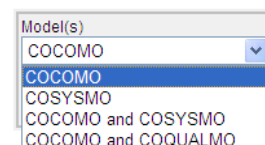
Рис. 1. Интерфейс интернет-калькулятора «COCOMO Suite of Constructive Cost Models»

В правом верхнем углу расположено окно выбора модели.

Здесь:

COSYSMO – Constructive Systems Engineering Cost Model,  
COQUALMO – Constructive Quality Model.

На сайте <http://csse.usc.edu/csse/tools/> находятся ссылки на другие программные продукты по расчету моделей COCOMO, разработанные в USC CSSE.



## 2. Интерфейс программы-калькулятора Costar 7.0

Copyright © 2011 Softstar Systems

Программа-калькулятор Costar 7.0 разработана компанией Softstar (<http://www.softstarsystems.com/>) на основе модели СОСОМО II для автоматизации оценки стоимости разработки программных продуктов.

Ниже приведены примеры интерфейса этой программы для уровня детальной оценки после проработки архитектуры (Post Architecture).

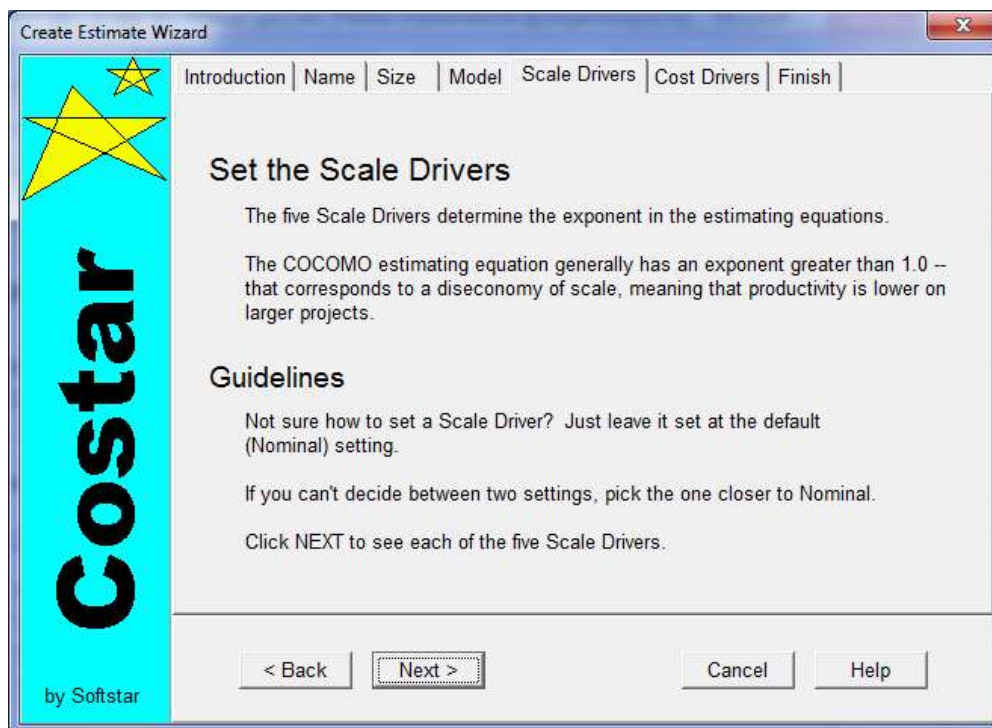


Рис. 2. Установка 5-ти факторов масштаба (Scale Drivers)

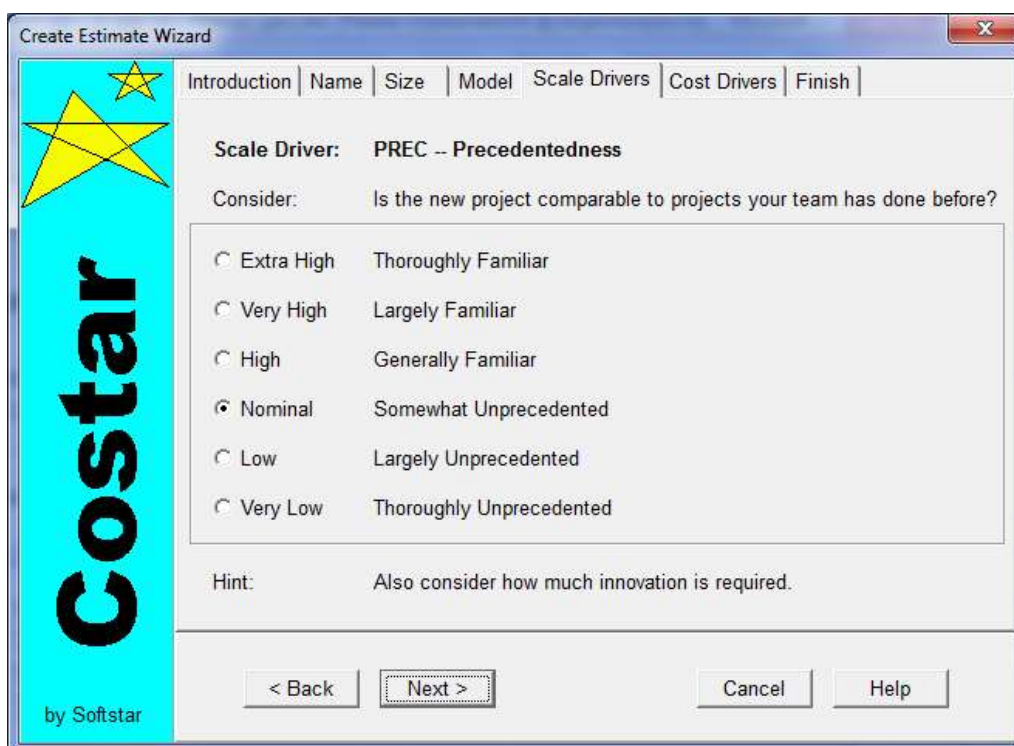


Рис. 3. Выбор уровня фактора масштаба «Прецедентность» (Precedentedness, PREC )

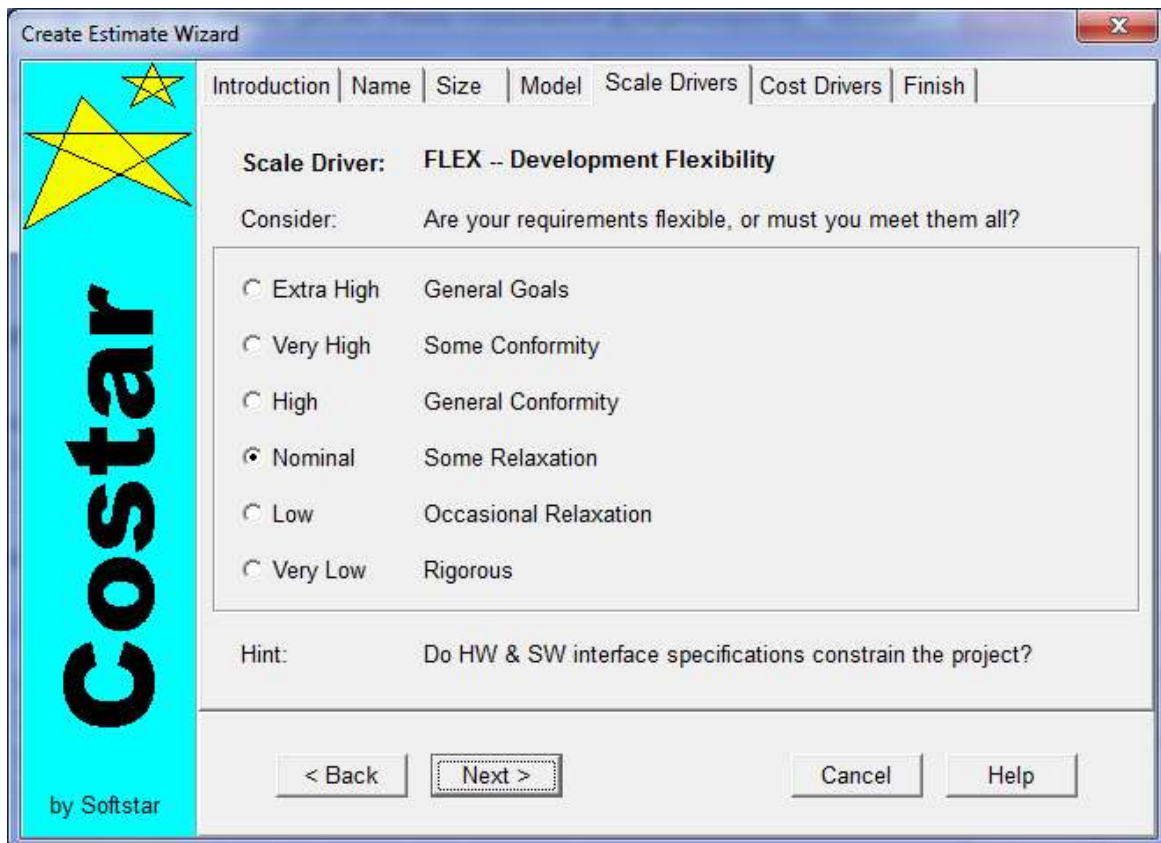


Рис. 4. Выбор уровня фактора масштаба «Гибкость разработки» (Development Flexibility, FLEX.)



Рис. 5. Выбор уровня фактора масштаба «Архитектура/Разрешение рисков» (Architecture / Risk Resolution, RESL)

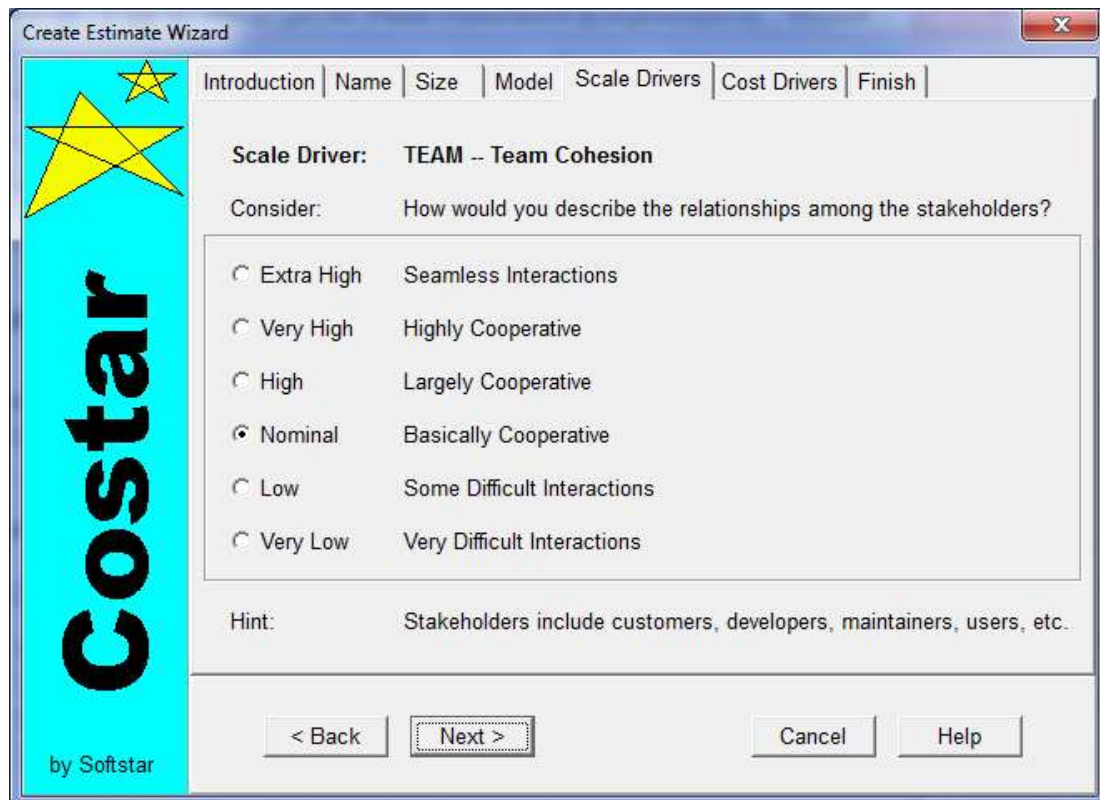


Рис. 6. Выбор уровня фактора масштаба «Сработанность команды» (Team Cohesion, TEAM)

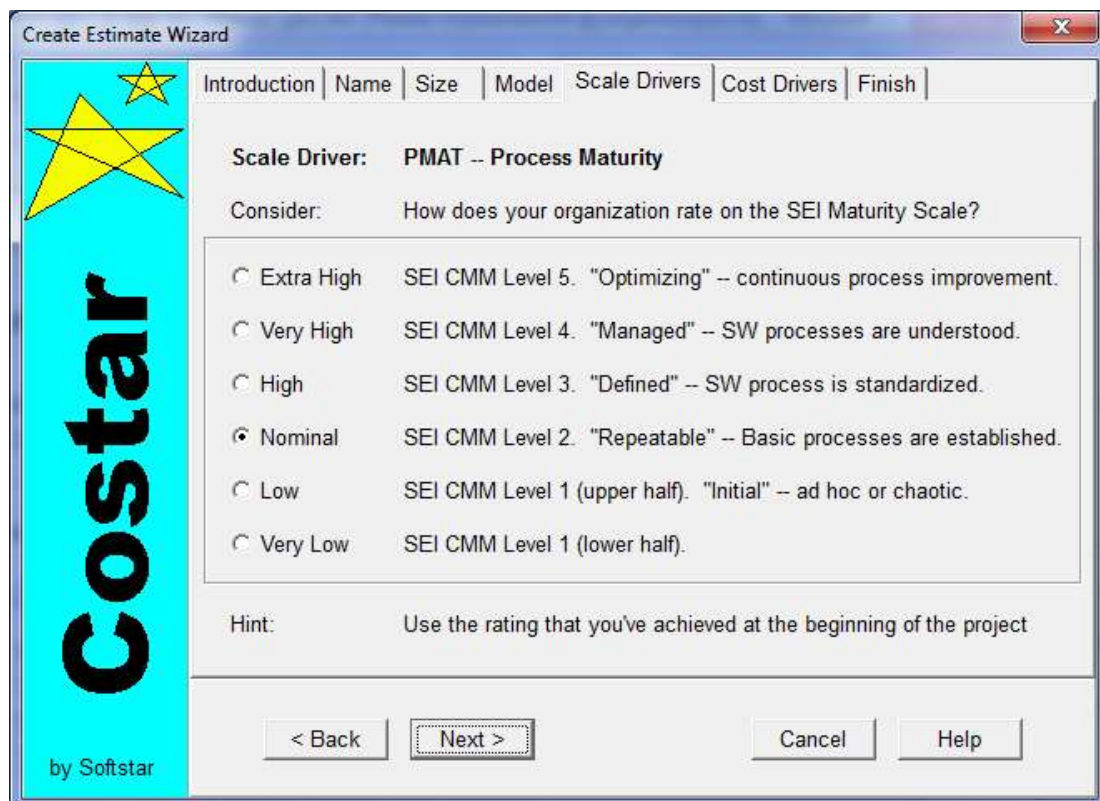


Рис. 7. Выбор уровня фактора масштаба «Зрелость процессов» (Process Maturity, PMAT)



## Cost Drivers (Effort Multipliers)

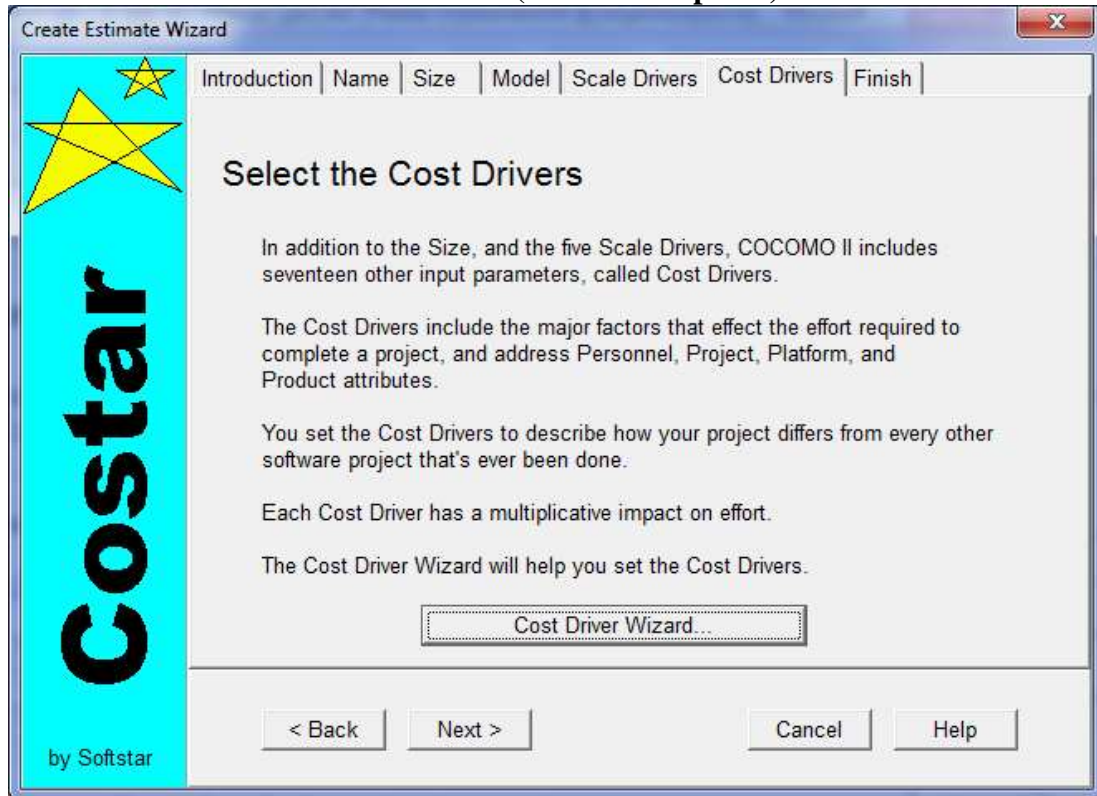


Рис. 8. Установка 17-ти факторов затрат (Cost Drivers, или Effort Multipliers)

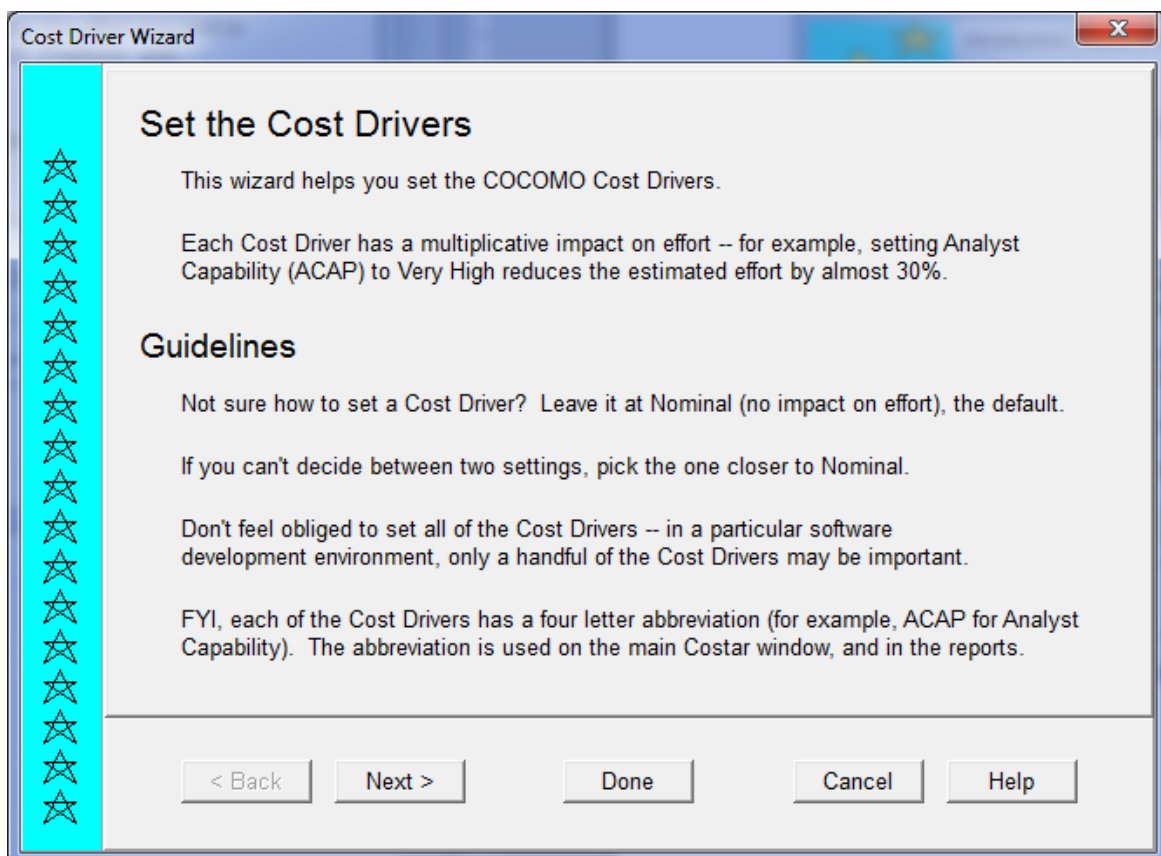


Рис. 9. Методические рекомендации по выбору факторов затрат. Например, «если вы не уверены в выборе фактора, оставьте его значение по умолчанию (Nominal)»

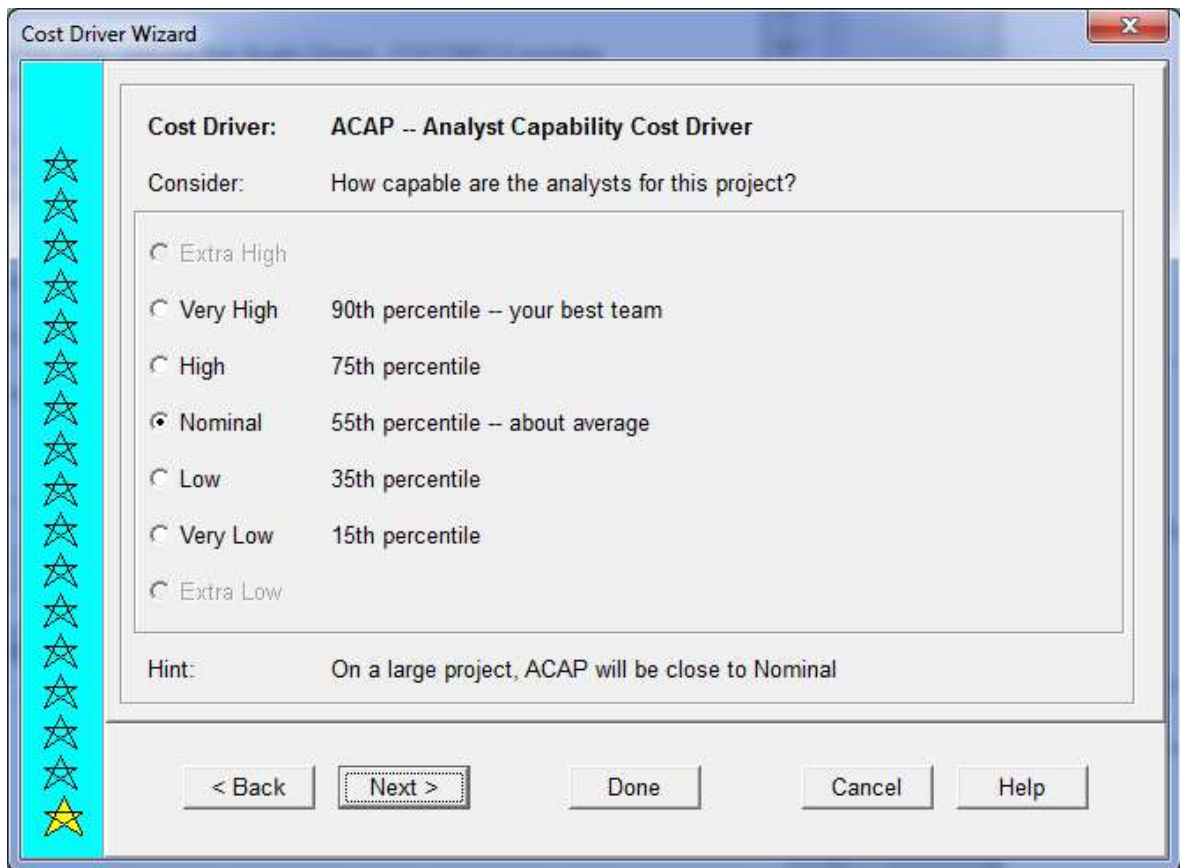


Рис. 10. Выбор уровня фактора затрат ACAP

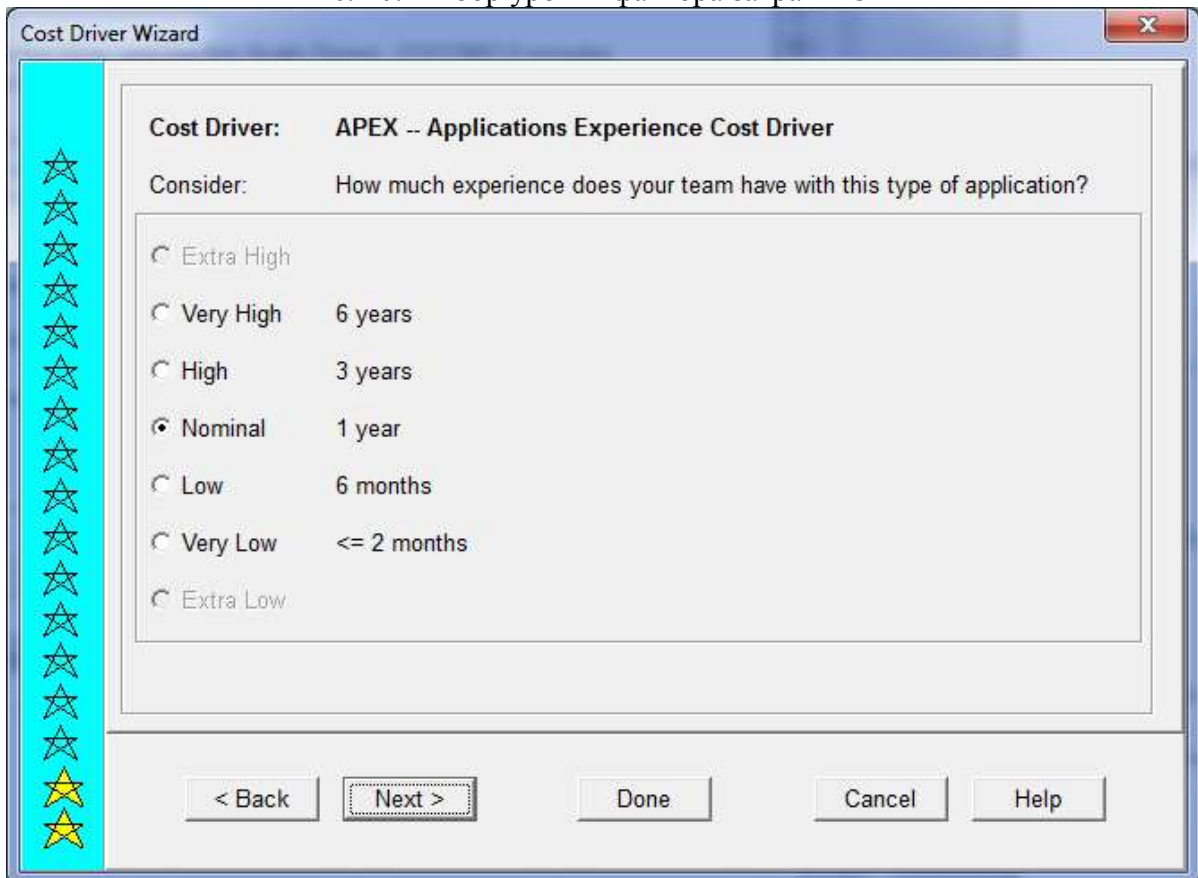


Рис. 11. Выбор уровня фактора затрат APEX

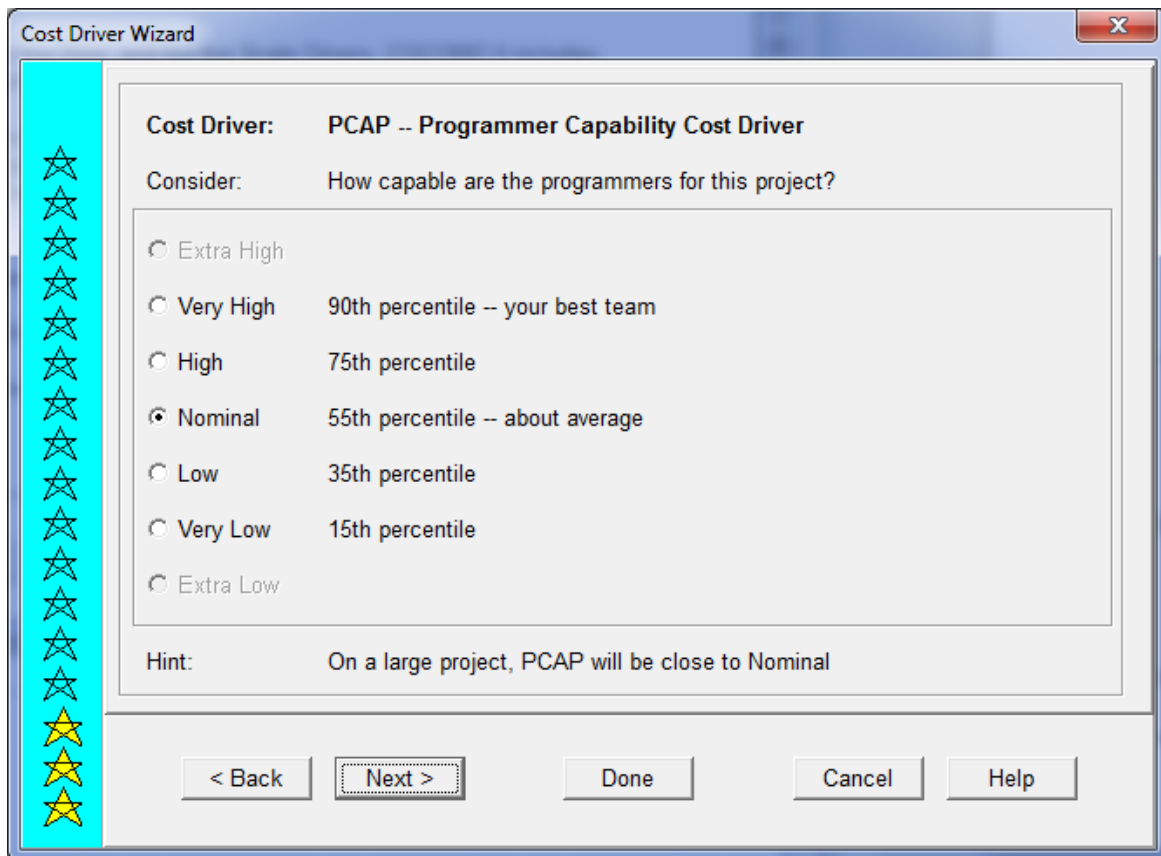


Рис. 12. Выбор уровня фактора затрат PCAP

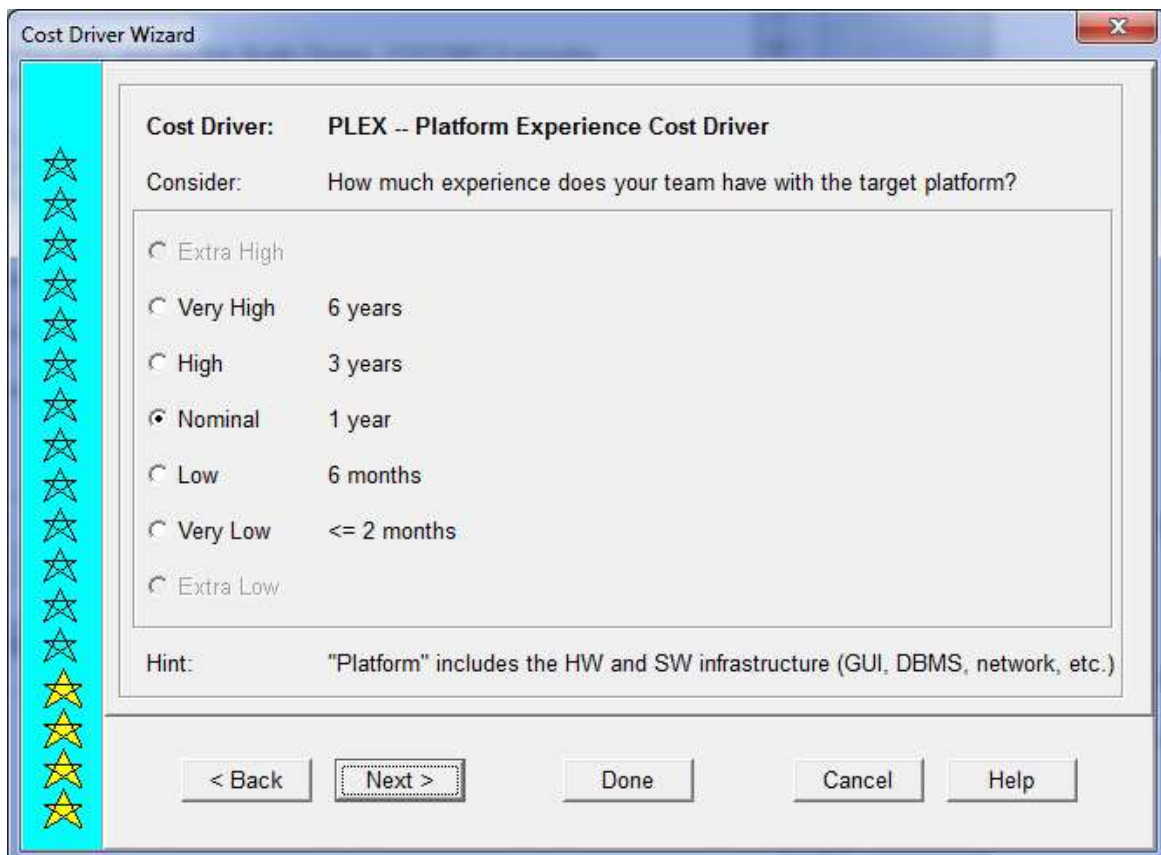


Рис. 13. Выбор уровня фактора затрат PLEX

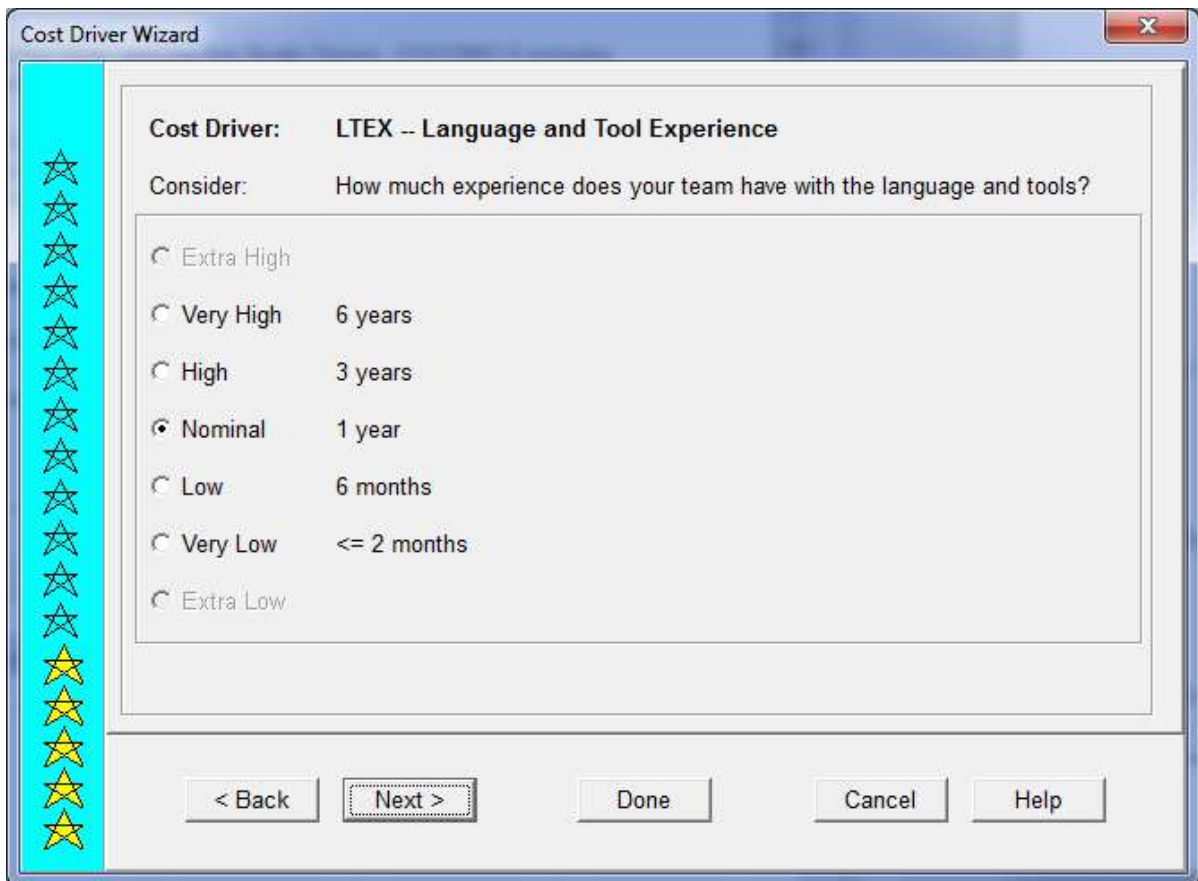


Рис. 14. Выбор уровня фактора затрат LTEX

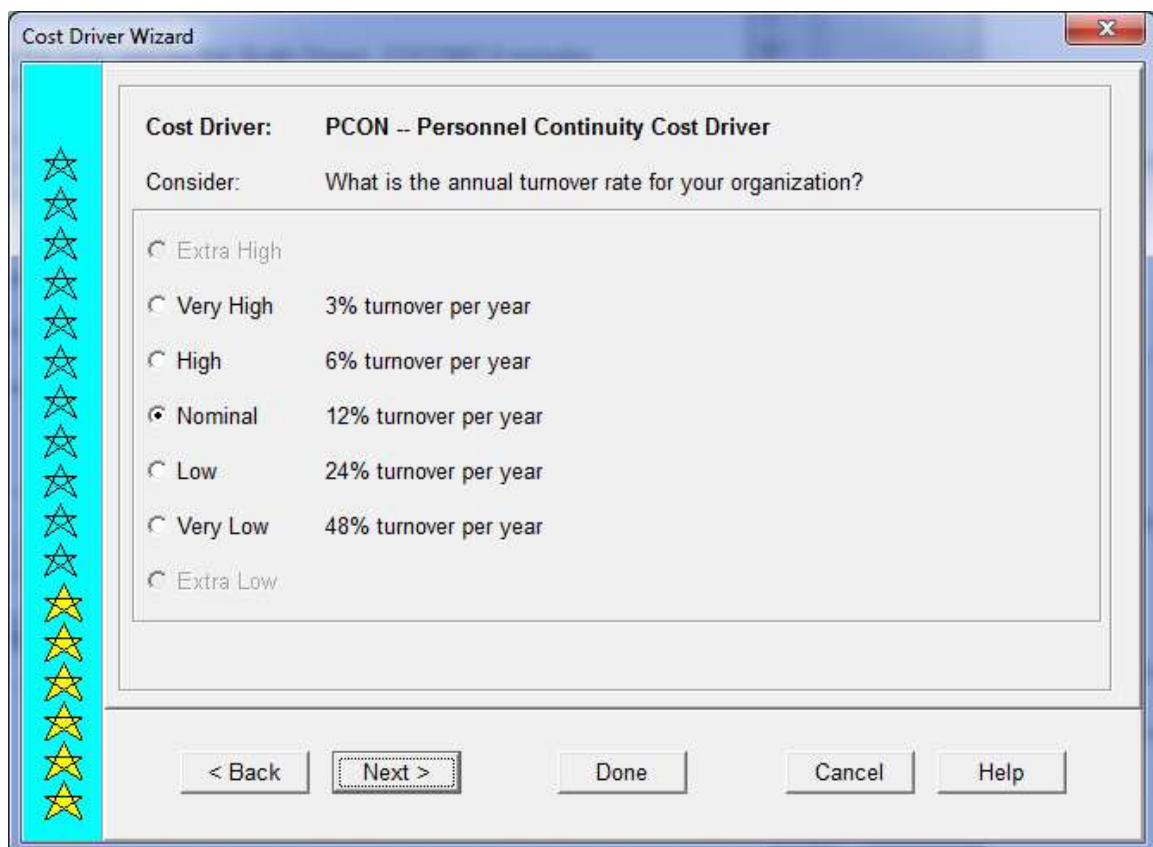


Рис. 15. Выбор уровня фактора затрат PCON

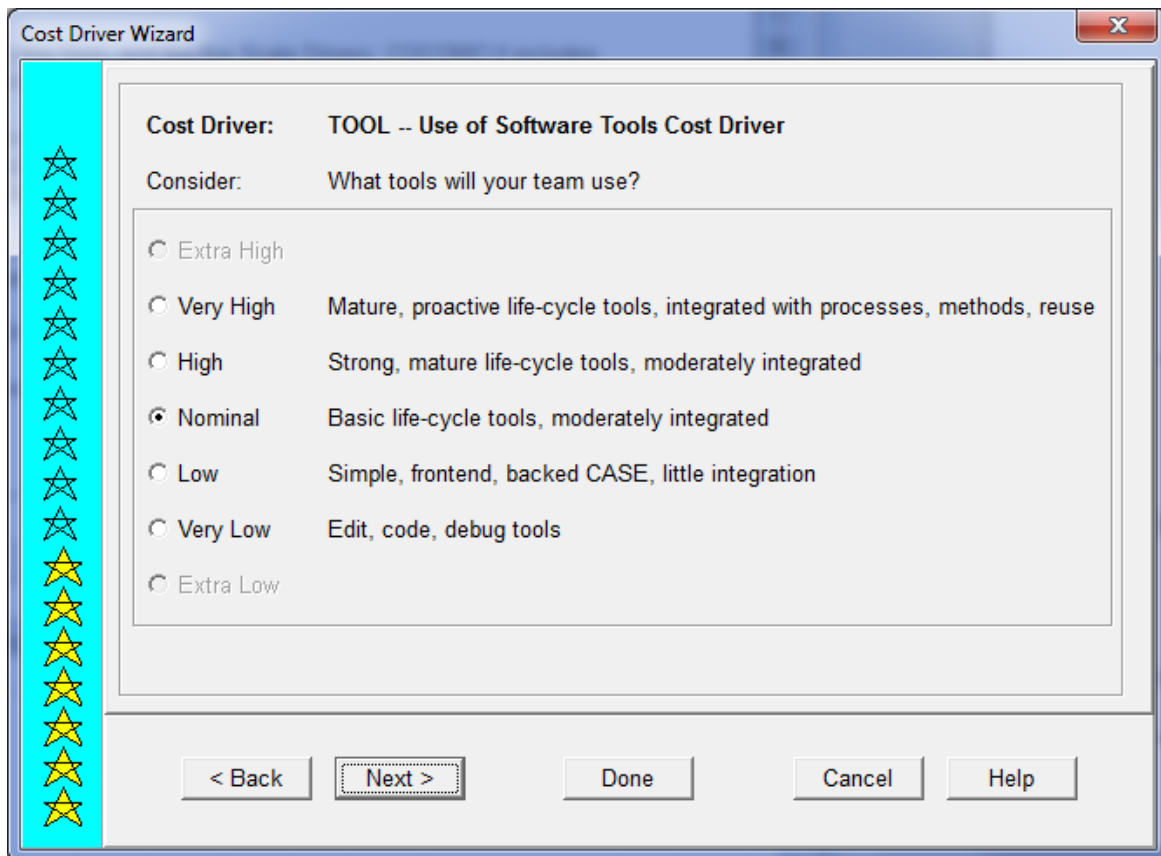


Рис. 16. Выбор уровня фактора затрат TOOL

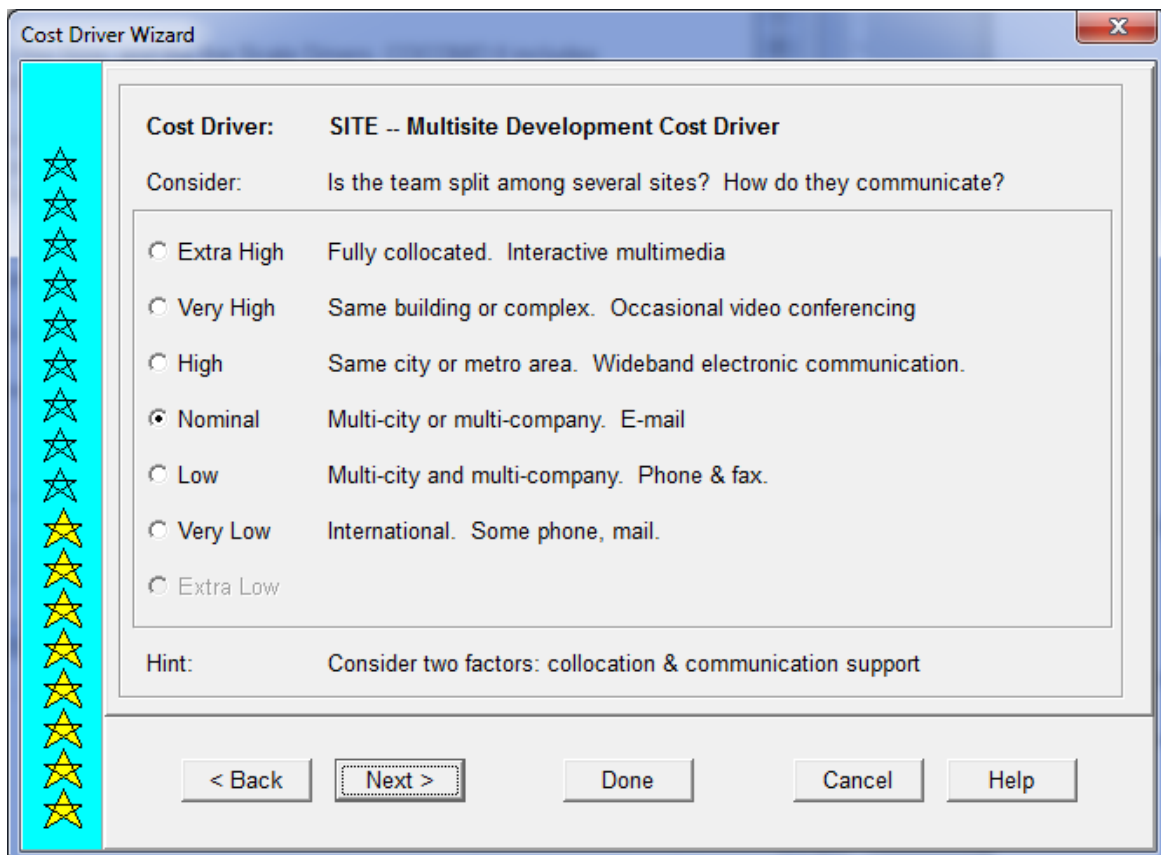


Рис. 17. Выбор уровня фактора затрат SITE

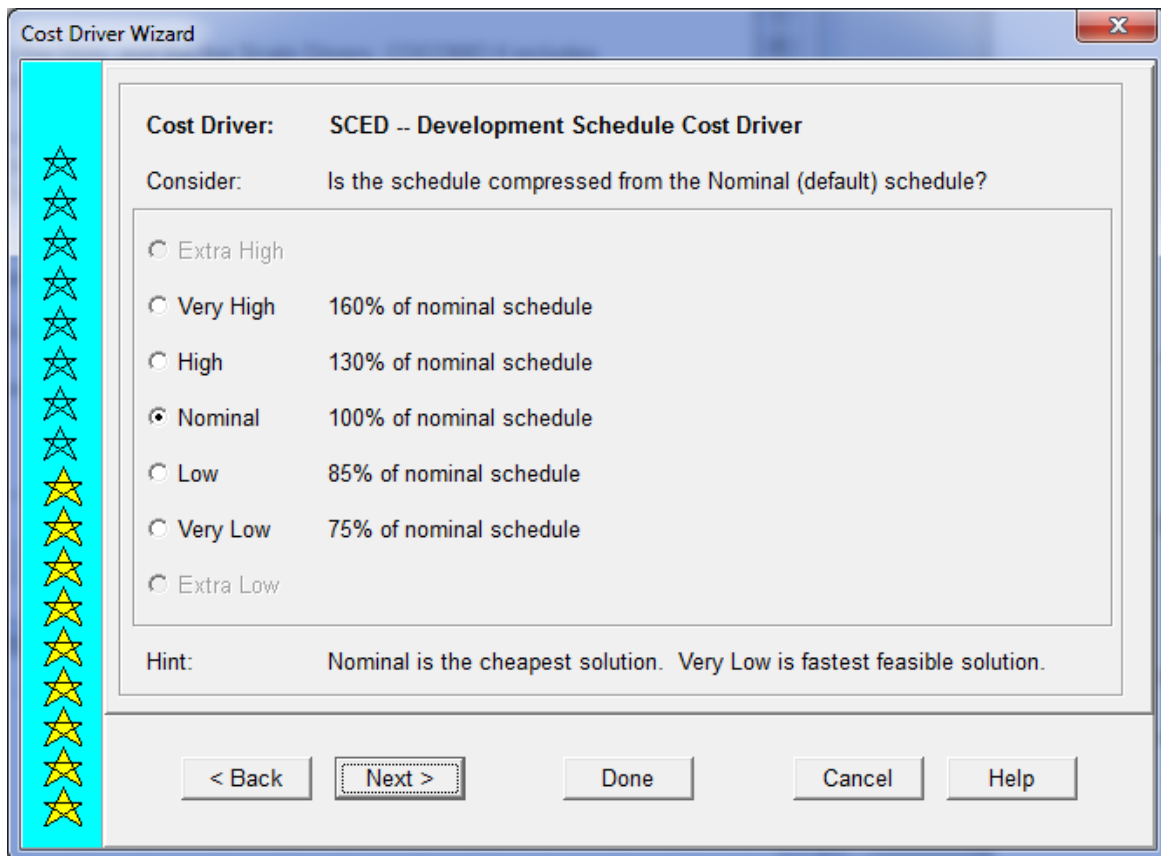


Рис. 18. Выбор уровня фактора затрат SCED

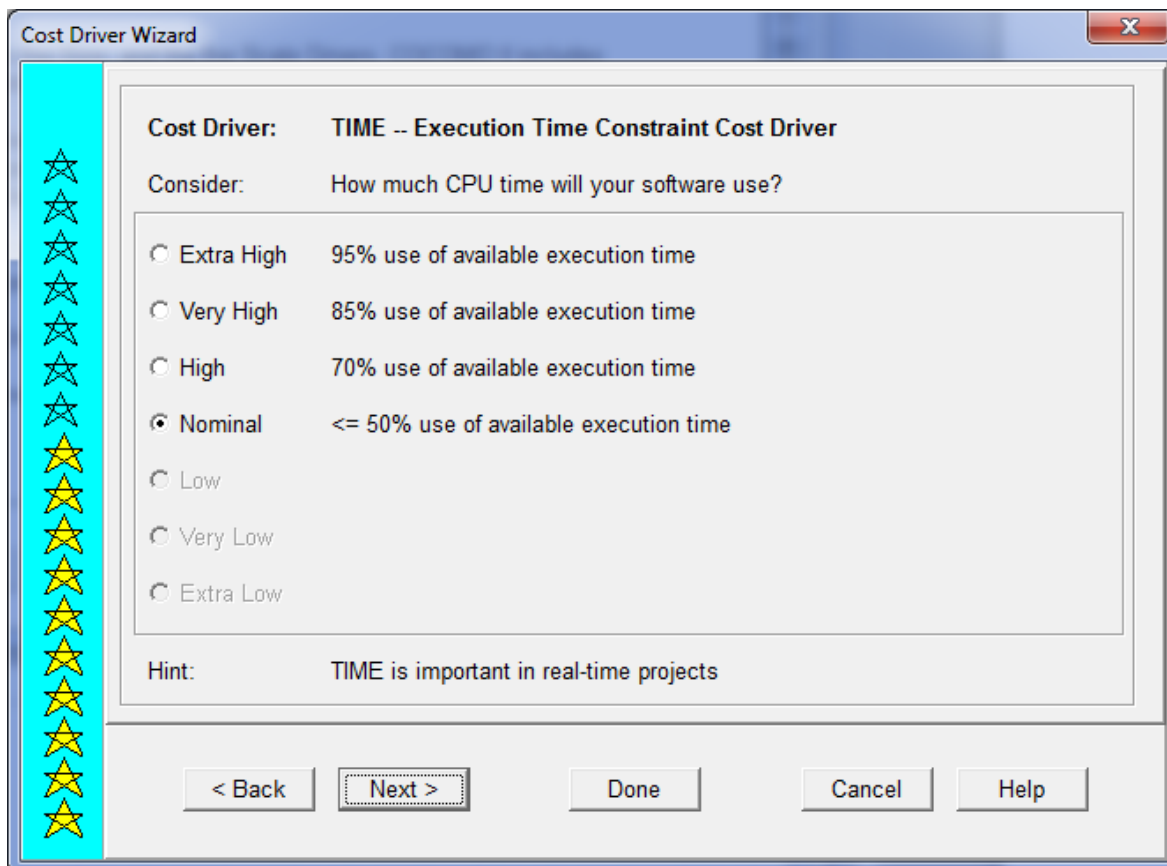


Рис. 19. Выбор уровня фактора затрат TIME

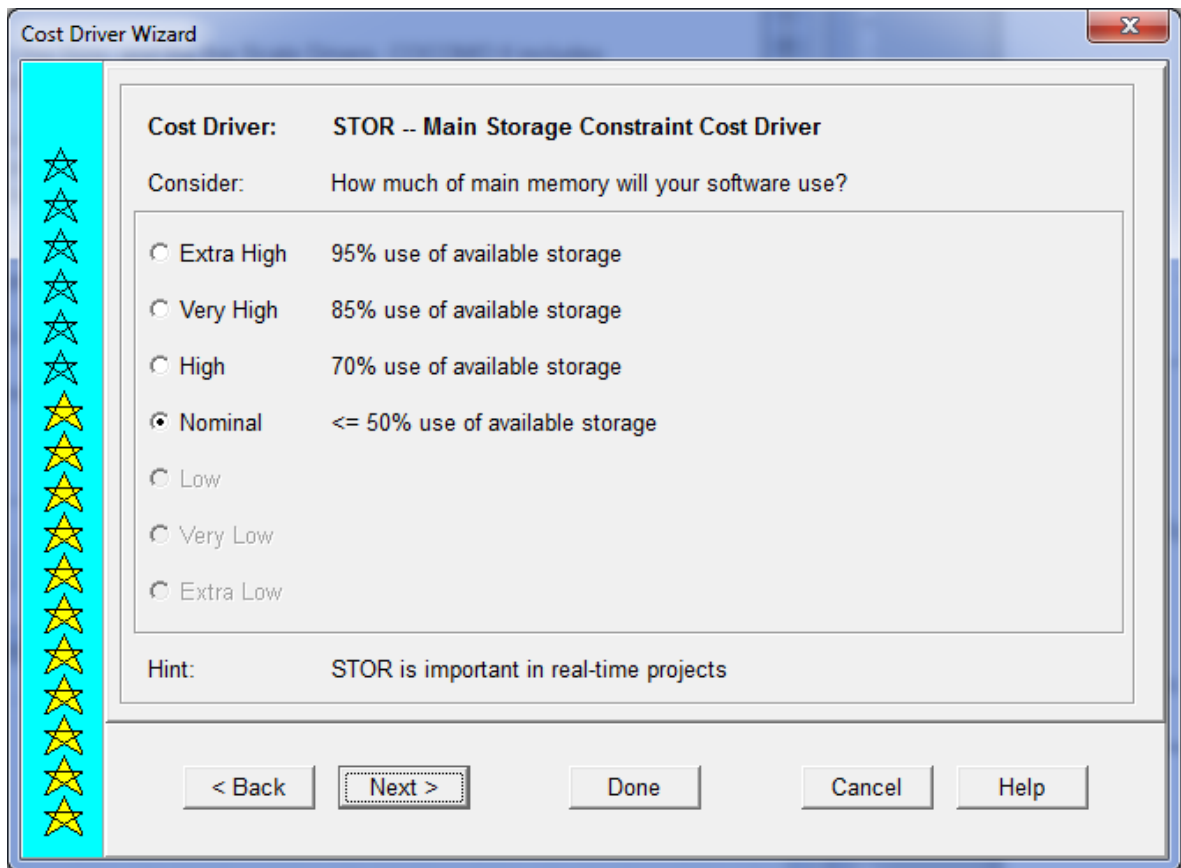


Рис. 20. Выбор уровня фактора затрат STOR

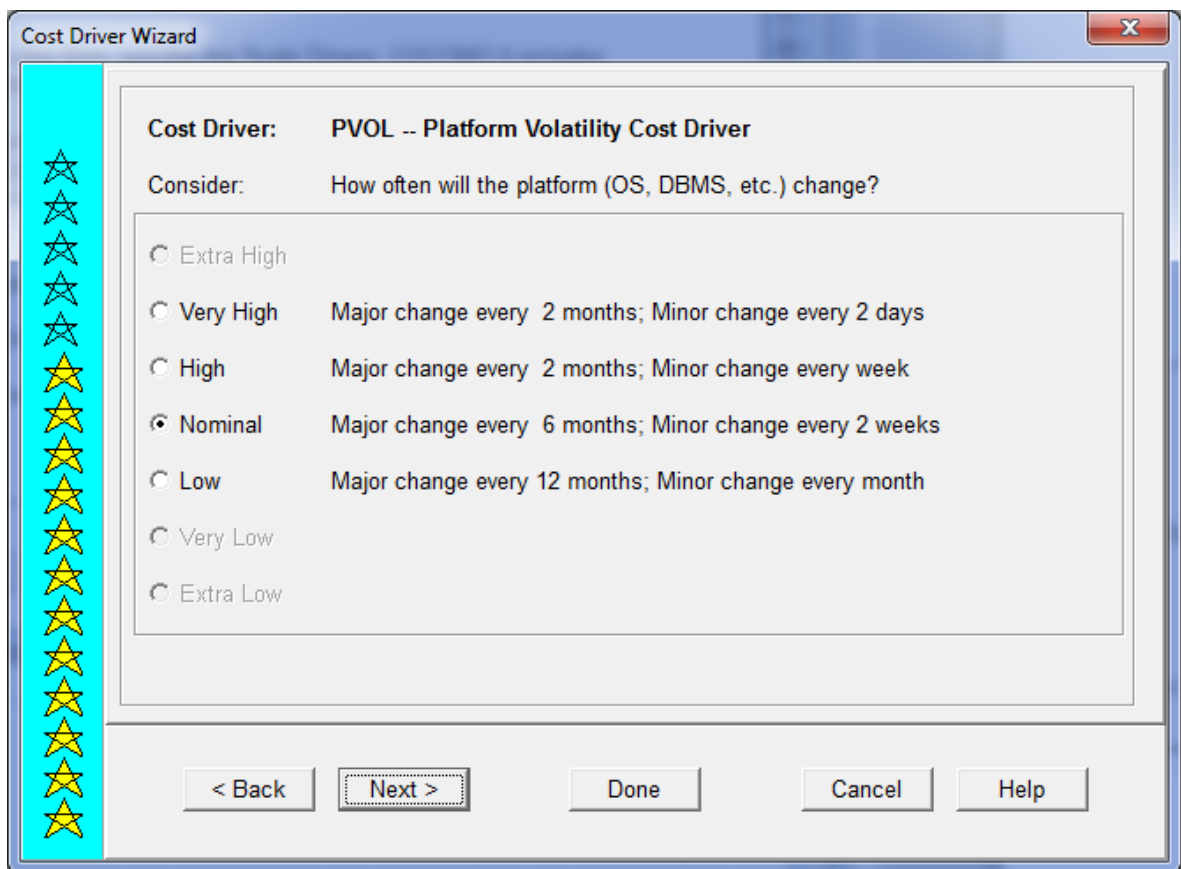


Рис. 21. Выбор уровня фактора затрат PVOL

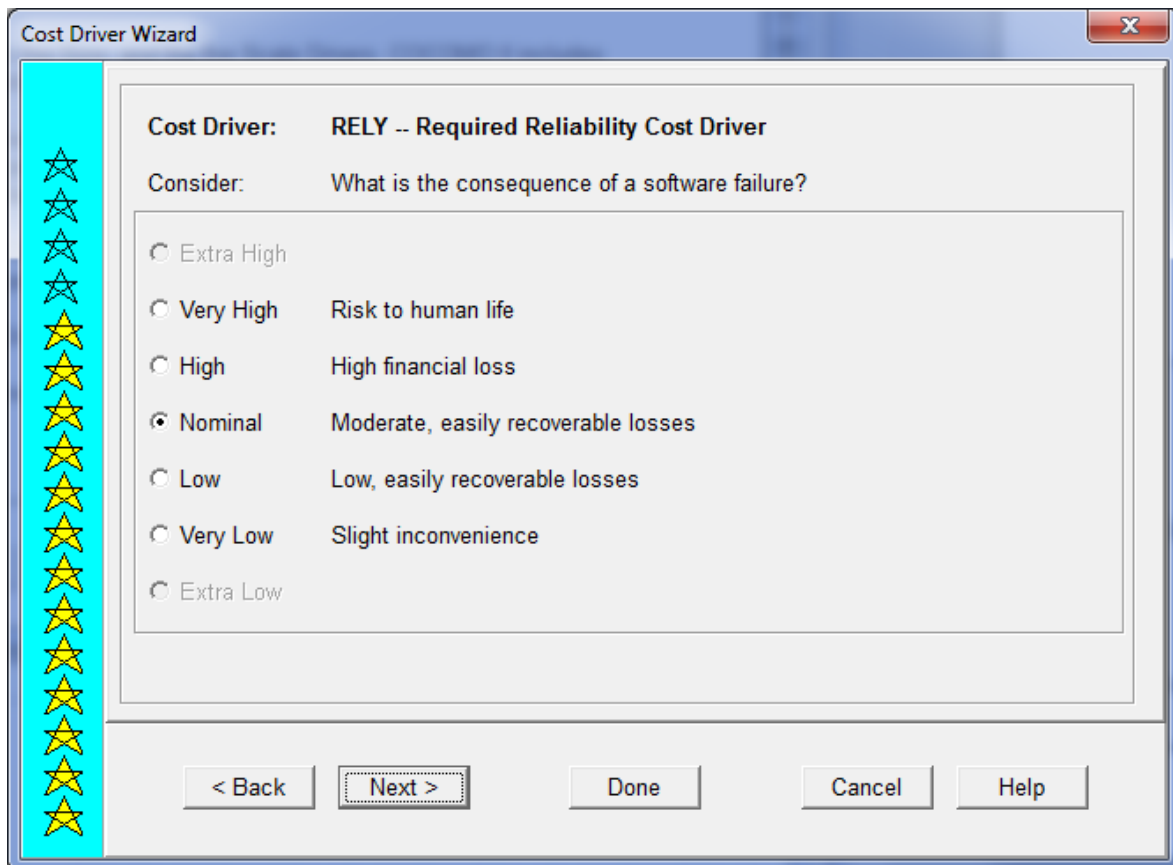


Рис. 22. Выбор уровня фактора затрат RELY

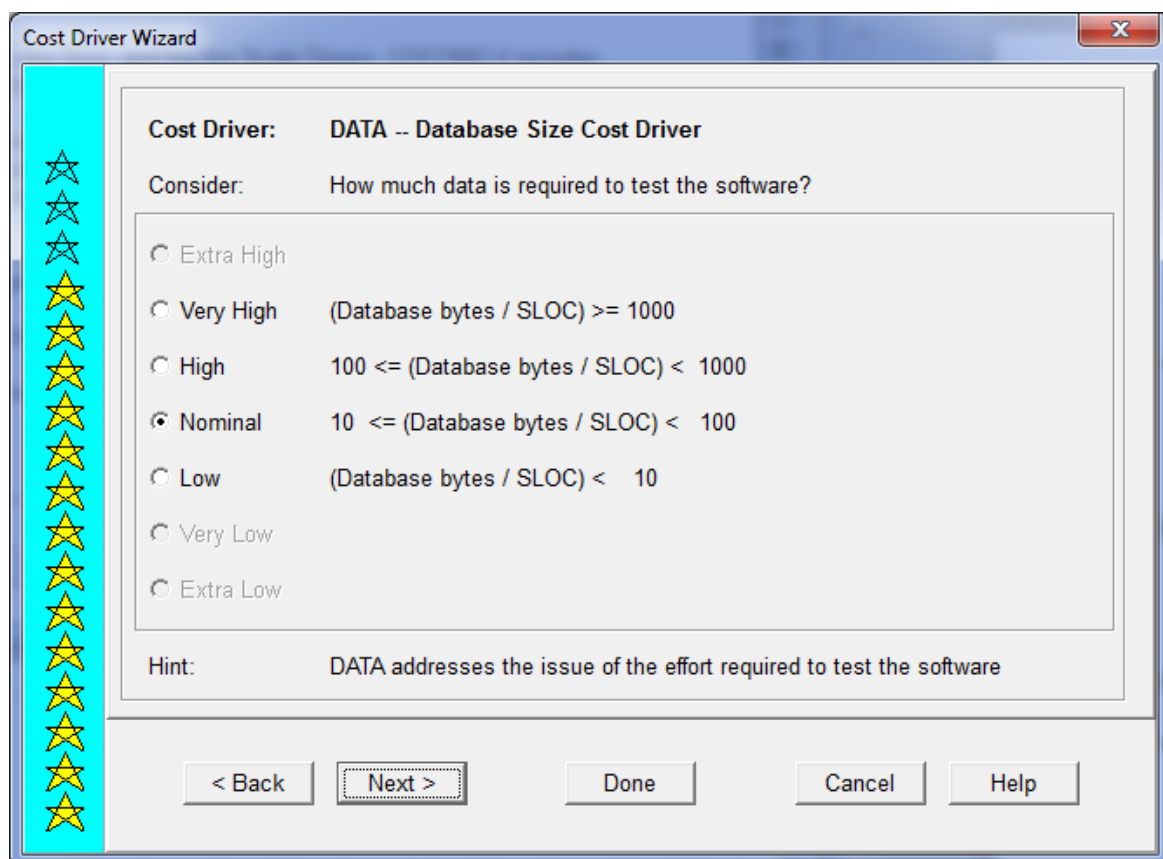


Рис. 23. Выбор уровня фактора затрат DATA



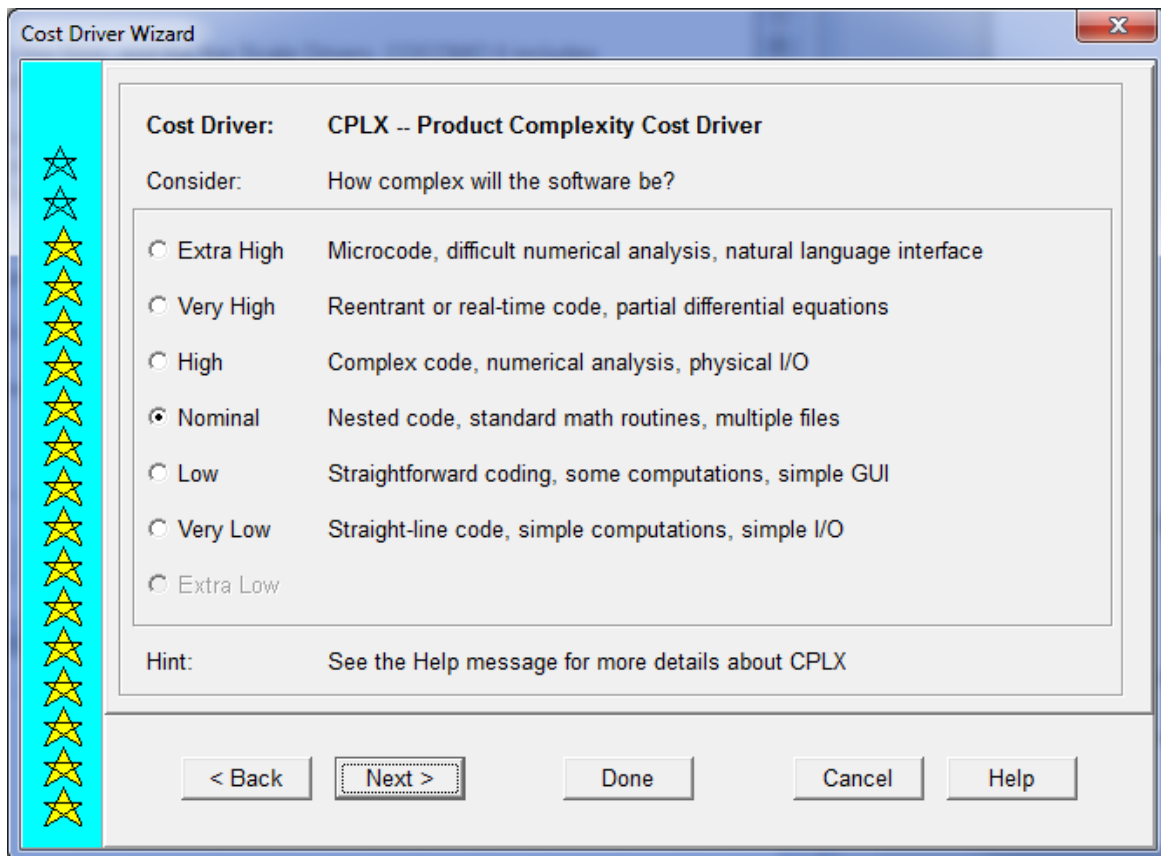


Рис. 24. Выбор уровня фактора затрат CPLX

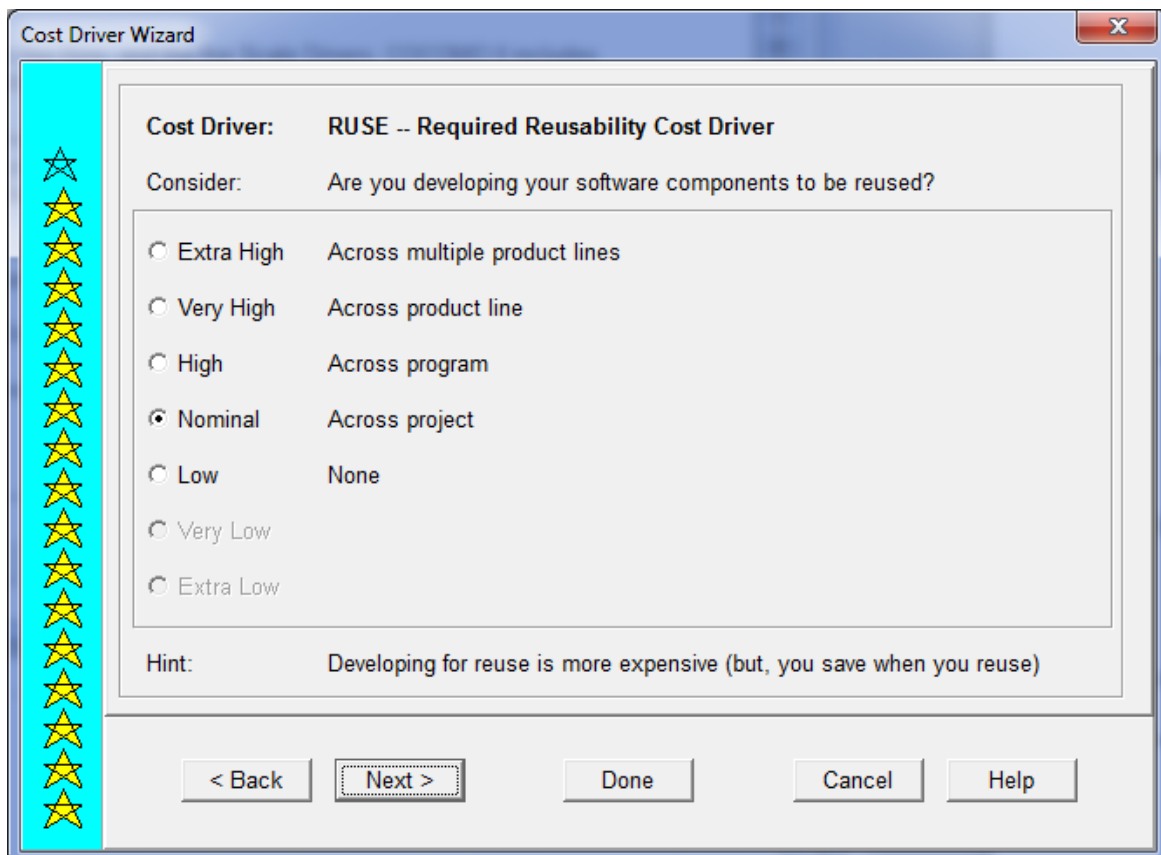


Рис. 25. Выбор уровня фактора затрат RUSE

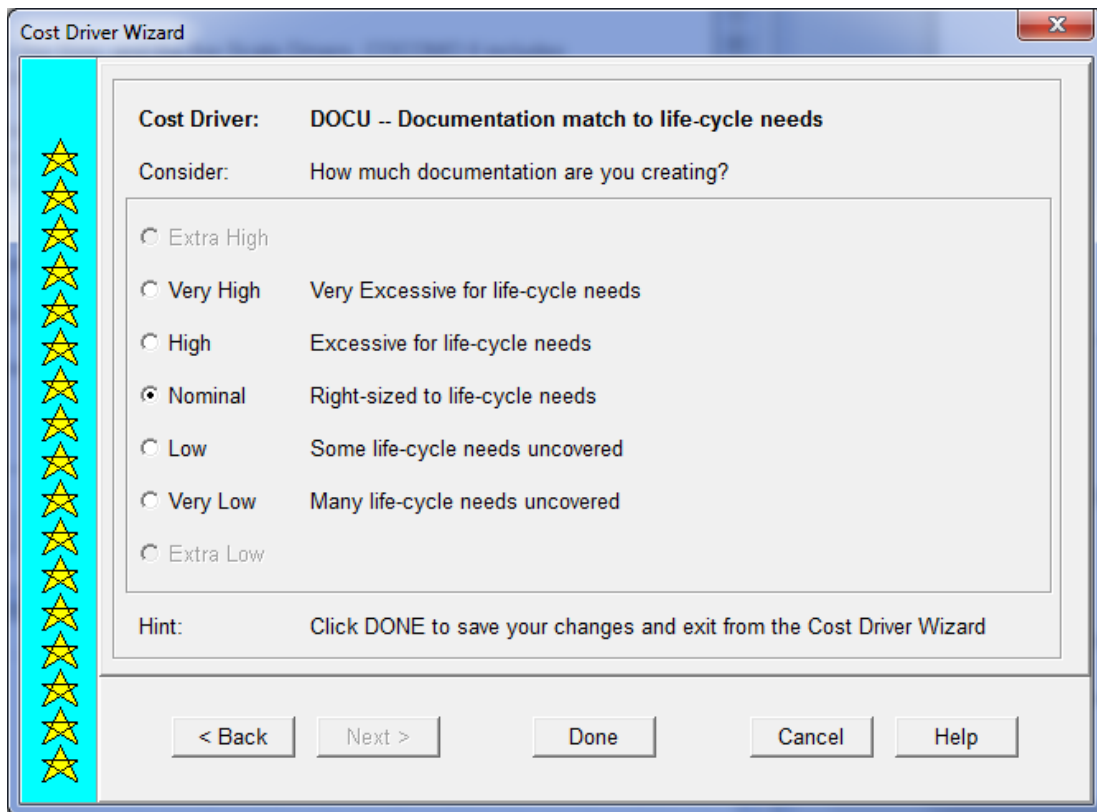


Рис. 26. Выбор уровня фактора затрат DOCU

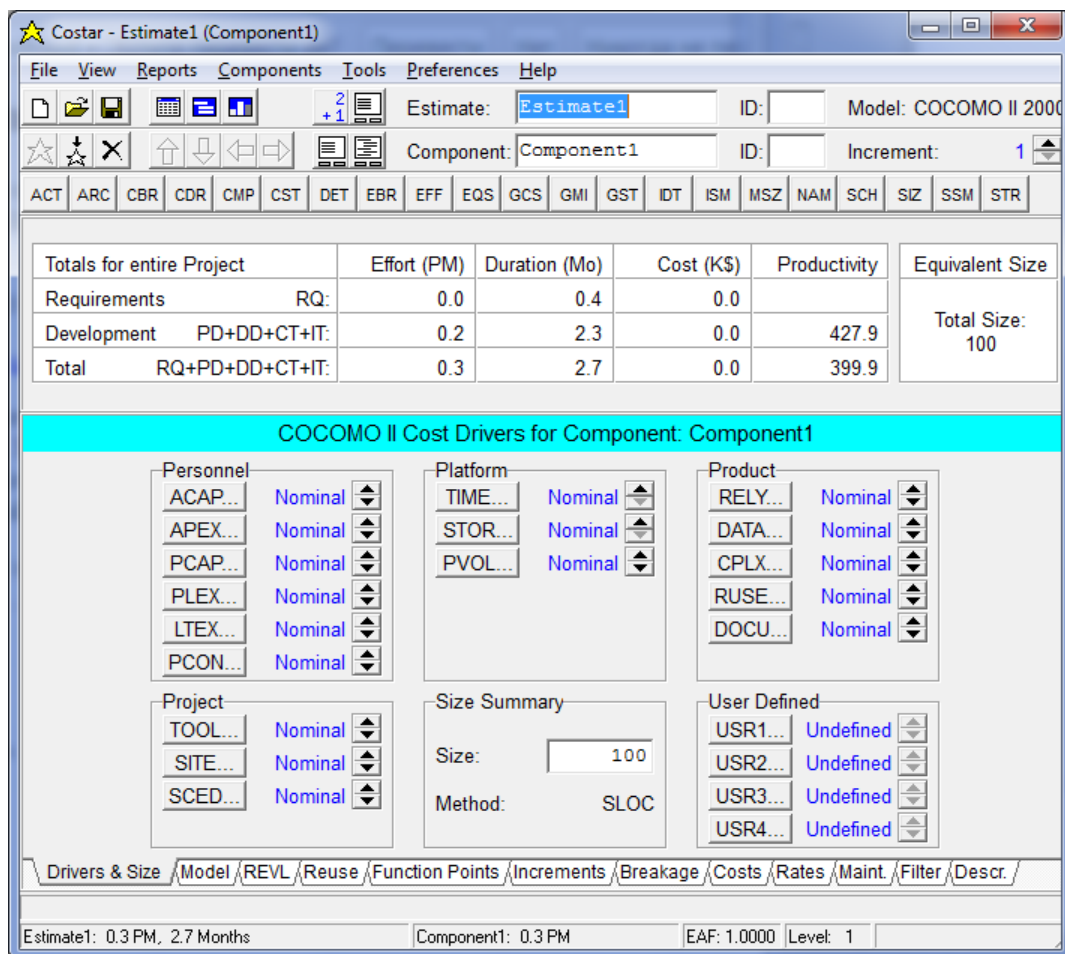


Рис. 27. Оценка трудоемкости и времени выполнения проекта объемом в 100 SLOC