



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ОТЧЕТ

по лабораторной работе №7

«Оценка параметров программного проекта с использованием
метода функциональных точек и модели СОСОМО II»

Студент: Зыкин Д.А.

Группа: ИУ7-83

Преподаватели: Барышникова М.Ю.,
Силантьева А.В.

Москва, 2020 г.

Цель работы: ознакомление с существующими методиками предварительной оценки параметров программного проекта и практическая оценка затрат на примере методики COSOMO II.

Содержание проекта: рассчитать количество функциональных точек для разрабатываемого программного приложения. С этой целью разработать программный инструмент. Произвести оценку трудозатрат и длительности разработки по методике COSOMO II с использованием моделей композиции приложения и ранней разработки архитектуры. Определить среднюю численность команды разработчиков. На основе экспертной оценки стоимости человеко-месяца произвести предварительную оценку бюджета проекта. Дать заключение о применимости метода функциональных точек и модели COSOMO II, а также их сравнения с базовой моделью COSOMO для решения поставленной задачи с учетом своего варианта.

Содержание индивидуального задания:

Компания получила заказ на разработку клиентского мобильного приложения брокерской системы. Программа позволяет просматривать актуальную биржевую информацию, производить сделки и отслеживать их выполнение. Приложение имеет 4 страницы: авторизация, биржевые сводки, заявки, новая заявка. Страница авторизации содержит два поля ввода и одну командную кнопку, а также флажок для запоминания параметров авторизации. Страница биржевых сводок содержит таблицу, содержащую колонки: Ценная бумага, Цена, Изменение, кнопку «Добавить» и диалоговое окно с одним полем для ввода и двумя командными кнопками. Заявки содержат таблицу, содержащую колонки: Тип, Имя бумаги, Цена, Количество. При нажатии на любую строку таблицы появляется контекстное меню с возможностью удалить или изменить заявку. Страница новой заявки позволяет оформить заявку на покупку или продажу ценной бумаги и состоит из 4 полей: Бумага, Цена, Покупка (булева переменная) и кнопка «Оформить».

Характеристики продукта:

1. Обмен данными - 5.
2. Распределенная обработка -5
3. Производительность -3
4. Эксплуатационные ограничения по аппаратным ресурсам – 2
5. Транзакционная нагрузка – 3
6. Интенсивность взаимодействия с пользователем (оперативный ввод данных) – 4
7. Эргономические характеристики, влияющие на эффективность работы конечных пользователей - 1
8. Оперативное обновление – 4
9. Сложность обработки – 4
10. Повторное использование – 0
11. Легкость инсталляции – 1
12. Легкость эксплуатации/администрирования – 2
13. Портруемость – 2
14. Гибкость- 2

Для реализации проекта была сформирована новая команда разработчиков, у отдельных членов которой имеется некоторый опыт создания систем подобного типа. В целях сплочения команды были проведены определенные мероприятия, что обеспечило на старте проекта приемлемую коммуникацию внутри коллектива. Заказчик не настаивает на жесткой регламентации процесса, однако график реализации проекта довольно жесткий. Несмотря на то, что предметная область является для разработчиков относительно новой, анализу архитектурных рисков было уделено лишь некоторое внимание. Организация только начинает внедрять методы управления проектами и формальные методы оценки качества процесса разработки. Надежность и уровень сложности (RCPX) разрабатываемой системы оцениваются как очень высокие, повторного использования компонентов не предусматривается (RUSE). Возможности персонала (PERS) – средние, его опыт работы в разработке систем подобного типа (PREX) низкий. Сложность платформы (PDIF) высокая. Разработка предусматривает очень интенсивное использование инструментальных средств поддержки (FCIL). Заказчик настаивает на жестком графике (SCED).

Описание методики функциональных точек

Этот метод используется для измерения производительности взамен устаревшего линейного подхода, где производительность измерялась количеством строк программного кода. Впервые функциональные точки (function points) были предложены сотрудником IBM Аланом Альбрехтом в 1979 г.

Преимуществом данного метода является то, что поскольку применение функциональных точек основано на изучении требований, то оценка необходимых трудозатрат может быть выполнена на самых ранних стадиях работы над проектом. Для поддержки и развития данного метода в 1986 г. была создана Международная группа пользователей функционального измерения (IFPUG — International Function Point User Group).

Определение числа функциональных точек является методом количественной оценки ПО, применяемым для измерения функциональных характеристик процессов его разработки и сопровождения независимо от технологии, использованной для его реализации.

Трудоемкость вычисляется на основе функциональности разрабатываемой системы, которая, в свою очередь, определяется путем выявления функциональных типов —логических групп взаимосвязанных данных, используемых и поддерживаемых приложением, а также элементарных процессов, связанных с вводом и выводом информации.

Описание методики COSOMO II

COSOMO II является развитием стандартного COSOMO. В методику входят три различные модели оценки стоимости:

- Модель композиции приложения —это модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка

- Модель ранней разработки архитектуры. Эта модель применяется для получения приблизительных оценок проектных затрат периода выполнения проекта перед тем как будет определена архитектура в целом. В этом случае используется небольшой набор новых драйверов затрат и новых уравнений оценки. В качестве единиц измерения используются функциональные точки либо KSLOC
- Постархитектурная модель –наиболее детализированная модель COCOMOII, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчета строк кода, а также новые уравнения

Модель композиции приложения	Модель ранней разработки архитектуры	Постархитектурная модель
Грубые входные данные	Ясно понимаемые особенности проекта	Детальное описание проекта
Оценки низкой точности	Оценки умеренной точности	Высокоточные оценки
Приблизительные требования	Ясно понимаемые требования	Стабилизировавшиеся основные требования
Концепция архитектуры	Ясно понимаемая архитектура	Стабильная базовая архитектура

Рисунок 1. Характеристики моделей оценки

В данной работе рассматриваются модель композиции приложения и модель ранней разработки архитектуры.

Метод функциональных точек

Передачи данных 5

Распределенная обработка данных 5

Производительность 3

Эксплуатационные ограничения 2

Частота транзакций 3

Оперативный ввод данных 4

Эффективность работы конечных пользователей 1

Оперативное обновление 4

Сложность обработки 4

Повторная используемость 0

Легкость инсталляции 1

Легкость эксплуатации 2

Количество возможных установок на различных платформах 2

Простота изменений 2

Рисунок 2. Системные параметры приложения

Интерфейс построено соответствует таблице, приведенной в лекции (рис. 3)

№	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 обработки?
3	Производительность	Нуждается ли пользователь в фиксации времени ответа или производительности?
4	Эксплуатационные ограничения	Насколько сильны эксплуатационные ограничения и каков объем специальных усилий на их преодоление?
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 конечных пользователей	Приложение проектировалось для обеспечения эффективной работы 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	Легкость инсталляции	Насколько трудны преобразование и инсталляция приложения?
12	Легкость эксплуатации	Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления?
13	Количество возможных установок на различных платформах	Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций?
14	Простота изменений (гибкость)	Была ли спроектирована, разработана и поддержана в приложении простота изменений?

Рисунок 3. Таблица определения системных параметров приложения

Таблица в середине формы используется для изменения параметров COSOMO II (рис. 4). Значения колонки Value соответствуют значению параметра. Значения варьируются от 0 до 5, что где 0 – очень низкий уровень, а 5 – сверхвысокий.

Модель ранней разработки архитектуры

PERS Номинальный

RCPX Очень высокий

RUSE Низкий

PDIF Высокий

PREX Низкий

FCIL Очень высокий

SCED Очень низкий

Рассчитать

Рисунок 4. Параметры COSOMO II

Для задания параметров степени в методике COSOMO II используется ряд выпадающих список в правой стороне формы. (рис. 5) Каждый элемент списка построчно соответствует значениям из лекций.

Факторы, влияющие на показатель степени в модели

Новизна (PREC) Наличие некоторого количества прецедентов

Гибкость разработки (FLEX) Некоторое согласование процесса

Разрешение рисков в архитектуре системы (RESL) Малое (20%)

Сплоченность команды (TEAM) Некоторая согласованность

Уровень зрелости процесса разработки (PMAT) Уровень 1 CMM

Рисунок 5. Настройка параметров степени

Чтобы вычислить значения по методике COSOMO II модели ранней разработки архитектуры используется кнопка «Рассчитать».

Применение модели композиции приложения

Для применения модели композиции приложения необходимо оценить количество объектных точек в проектируемом проекте.

По условию задачи, проект состоит из четырех форм, двух отчетов и двух модулей.

Рассмотрим формы:

1. Авторизация. Средняя сложность
2. Биржевые сводки. Средняя сложность
3. Заявки. Средняя сложность
4. Новая заявка. Средняя сложность

Отчеты:

1. Список заявок. Средняя сложность
2. Биржевые сводки. Средняя сложность

Модули:

1. Модуль авторизации
2. Модуль работы с биржей

Подсчитаем количество объектных точек:

2 формы низкой сложности, 2 формы средней сложности, 2 отчета средней сложности, 2 модуля. Итого:

$$2 * 4 + 2 * 5 + 2 * 10 = 38$$

По условию задачи (брокерское приложение) количество объектных точек было оценено в 38 штук.

Применение модели ранней разработки архитектуры

Для применения данной модели необходимо вычислить значение KLOC с помощью методики функциональных точек.

Внешние вводы – это отображение рассчитанных данных: количество DET-элементов, связанных с 1 ссылкой на файл (FTR), равно 4; количество

DET-элементов, связанных с 2 ссылками на файл (FTR), равно 1; количество DET-элементов, связанных более, чем 2 ссылками на файл (FTR), равно 1.

Внешние выводы – это отображение рассчитанных данных: количество DET-элементов, связанных с 1 ссылкой на файл (FTR), больше 20; количество DET-элементов, связанных с 2 ссылками на файл (FTR), равно 1; количество DET-элементов, связанных более, чем 3 ссылками на файл (FTR), равно 1.

Внешние запросы – элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных или обновлением внутренних логических файлов (базы данных). Количество DET-элементов, связанных с 1 ссылкой на файл, равно 15; количество DET-элементов, связанных с 2 ссылками на файл, равно 1; количество DET-элементов, связанных более, чем 3 ссылками на файл, равно 1.

Внутренние логические файлы – это программные модули программы. Количество DET-элементов, связанных с 1 типом элементов-записей (RET), равно 10; количество DET-элементов, связанных с 2-5 типами элементов-записей (RET), равно 5; количество DET-элементов, связанных более чем 5 типом элементов-записей (RET), равно 5.

Внешние интерфейсные файлы используются для взаимодействия различных программ. В данном приложении, количество DET-элементов, связанных с 1 типом элементов-записей (RET), равно 4; DET-элементов, связанных с 2-3 типами элементов-записей (RET), равно 2; DET-элементов, связанных более, чем 3 типом элементов-записей (RET), равно 1.

Внешние вводы EI

	1-4	5-15	>15
0-1	4		
2	1		
>2	1		

Внешние выводы EO

	1-4	5-19	>19
0-1			20
2-3	1		
>3	1		

Внутренние логические файлы ILF

	1-19	20-50	>50
1	10		
2-5	5		
>5	5		

Внешние интерфейсные файлы EIF

	1-19	20-50	>50
1			
2-5	3		
>5			

Внешние запросы EQ

	1-4	5-19	>19
0-1	4		
2-3	2		
>3	1		

Рисунок 6. Ввод данных о ФТ

Применение методики для конкретной задачи

Метод функциональных точек

Передачи данных

Распределенная обработка данных

Производительность

Эксплуатационные ограничения

Частота транзакций

Оперативный ввод данных

Эффективность работы конечных пользователей

Оперативное обновление

Сложность обработки

Повторная используемость

Легкость инсталляции

Легкость эксплуатации

Количество возможных установок на различных платформах

Простота изменений

Рассчитать

Коэфф.корректировки:38.00

Функц.точки с корректировкой:
329.60

Функц.точки без корректировки:
320.00

Внешние вводы EI

	1-4	5-15	>15
0-1	4		
2	1		
>2	1		

Внешние выводы EO

	1-4	5-19	>19
0-1			20
2-3	1		
>3	1		

Внутренние логические файлы ILF

	1-19	20-50	>50
1	10		
2-5	5		
>5	5		

Внешние интерфейсные файлы EIF

	1-19	20-50	>50
1			
2-5	3		
>5			

Внешние запросы EQ

	1-4	5-19	>19
0-1	4		
2-3	2		
>3	1		

Факторы, влияющие на показатель степени в модели

Новизна (PREC)

Наличие некоторого количества прецедентов

Гибкость разработки (FLEX) Некоторые послабления в процессе

Разрешение рисков в архитектуре системы (RESL) Малое (20%)

Сплоченность команды (TEAM) Некоторая согласованность

Уровень зрелости процесса разработки (PMAT) Уровень 1 CMM

P = 1.25

Модель ранней разработки архитектуры

PERS Номинальный

RCPX Очень высокий

RUSE Низкий

PDIF Высокий

PREX Низкий

FCIL Очень высокий

SCED Очень низкий

Рассчитать

Трудозатраты(чел/мес):
183.41

Время(мес):21.51

Модель композиции изображения

%RUSE: 0

Опытность команды\разработчика Низкая

Экранные формы

Простые 0

Средние 4

Сложные 0

Отчеты

Простые 0

Средние 2

Сложные 0

Модули на языках 3 поколения 2

Трудозатраты(чел/мес):5.428571428571429

Время(мес):5.69

Рисунок 7. Результаты работы

Вывод

Методика COSOMO II, как и ее первая версия, позволяет досочно быстро оценить длительность и трудозатраты проекта, основываясь на субъективных данных. В условиях отсутствия объективной информации о предполагаемых трудозатратах особенно важно правильно спрогнозировать характеристики проекта.

Методика функциональных точек позволяет оценить размер программного продукта на этапе его проектирования. С помощью этого подхода можно применять методики COSOMO, которым необходимо знание о размере продукта. Хотя методика функциональных точек является неточной (как и любая методика прогнозирования), этих результатов достаточно для общего понимания объемов работ и получения приблизительных оценок параметров проекта.