Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №9**

«Оценка трудоемкости разработки программного продукта»

дисциплина «Технология программирования»

Выполнил.

студент группы 413.1 Лебедев Д.В.

Проверил.

ст. преподавать кафедры ПОиАИС Ураева Е. Е.

Курск, 2025

**Цель работы:**

Научиться определять плановую трудоемкость и время создания программного продукта (ПП) с использованием методов функциональных указателей (Function Points) и вариантов использования (Use Case Points).

1. **Исходные данные**
   1. **Действующие лица системы**

Участник команды – обычный пользователь, имеющий доступ к задачам и проектам своей команды

Руководитель команды – пользователь с расширенным функционалом, имеющий доступ к редактированию и удалению проектов, задач, и участников своей команды.

Администратор – имеет доступ к изменению всех данных всех пользователей всех команд

* 1. **Варианты использования системы**

Таблица 1 – Варианты использования для оценки UCP

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Классификация** |
| 1 | Регистрация | Simple |
| 2 | Авторизация | Simple |
| 3 | Создать команду | Simple |
| 4 | Присоединиться к команде | Simple |
| 5 | Назначить роль | Simple |
| 6 | Редактировать профиль | Average |
| 7 | Отобразить информацию о пользователе | Simple |
| 8 | Удалить пользователя из команды | Simple |
| 9 | Создать проект | Simple |
| 10 | Отобразить информацию о проекте | Simple |
| 11 | Удалить проект | Simple |
| 12 | Создать задачу | Simple |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Классификация** |
| 13 | Отобразить информацию о задаче | Simple |
| 14 | Редактировать задачу | Average |
| 15 | Переместить задачу по Канбану | Simple |
| 16 | Отобразить Гант-диаграмму | Average |
| 17 | Комментировать задачу | Simple |
| 18 | Удалить комментарий | Simple |
| 19 | Редактировать комментарий | Simple |
| 20 | Отобразить системную админ-панель | Complex |
| 21 | Удалить команду | Simple |
| 22 | Редактировать команду | Average |
| 23 | Сменить пригласительный код команды | Simple |

* 1. **Технологический стек**

Backend: Python 3.x

Frontend: Jinja2 flask + Tailwind CSS

База данных: PostgreSQL

Контейнеризация: Docker

1. **Этапы выполнения расчётов**
   1. **Анализ и классификация актёров (UAW)**
   2. **Классификация вариантов использования (UUCW)**

Распределение сложности (методика Карластера):

— Простые (Simple, 5): 18 шт.

— Средние (Average, 10): 4 шт.

— Сложные (Complex, 15): 1 шт.

* 1. **Рассчёт UUCP**
  2. **Технические факторы (TCF)**

Таблица 2 – технические факторы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Технический фактор** | **Вес** | **Оценка** | **Вклад (Вес × Оценка)** |
| T1 | Требуется высокая надежность системы | 1.0 | 4 | 4.0 |
| T2 | Легкость внедрения новых функций | 0.5 | 4 | 2.0 |
| T3 | Сложность внутренней обработки | 1.0 | 3 | 3.0 |
| T4 | Требуется производительность | 1.0 | 3 | 3.0 |
| T5 | Высокий уровень повторного использования компонентов | 1.0 | 4 | 4.0 |
| T6 | Легкость установки | 0.5 | 5 | 2.5 |
| T7 | Легкость использования интерфейса | 0.5 | 4 | 2.0 |
| T8 | Портируемость (работа на разных платформах) | 2.0 | 2 | 4.0 |
| T9 | Легкость изменения системы | 1.0 | 4 | 4.0 |

Продолжение таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Технический фактор** | **Вес** | **Оценка** | **Вклад (Вес × Оценка)** |
| T10 | Параллельная/распределенная обработка | 1.0 | 1 | 1.0 |
| T11 | Специальные требования к безопасности | 1.0 | 4 | 4.0 |
| T12 | Доступность пользователей к обучению работы с системой | 1.0 | 3 | 3.0 |
| T13 | Специальные требования к настройкам и установке | 0.5 | 2 | 1.0 |

* 1. **Факторы окружения (EF)**

Таблица 3 – факторы окружения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Фактор окружения** | **Вес** | **Оценка** | **Вклад (Вес × Оценка)** |
| E1 | Опыт работы с аналогичными системами | 1.5 | 4 | 6.0 |
| E2 | Опыт применения объектно-ориентированного проектирования | 0.5 | 3 | 1.5 |
| E3 | Опыт разработки системы | 1.0 | 4 | 4.0 |
| E4 | Знание технологии (Flask, PostgreSQL, Docker) | 0.5 | 4 | 2.0 |
| E5 | Мотивация команды | 1.0 | 5 | 5.0 |
| E6 | Устойчивость требований | 2.0 | 4 | 8.0 |
| E7 | Четкость процессов разработки | 1.0 | 3 | 3.0 |
| E8 | Использование доп. инструметов | 1.0 | 3 | 3.0 |

Сумма вкладов = 6.0 + 1.5 + 4.0 + 2.0 + 5.0 + 8.0 + 3.0 + 3.0 = 32.5

Теперь считаем сам ECF по формуле:

* 1. **Расчёт UCP**
  2. **Трудоёмкость и длительность по UCP**

Трудоёмкость = UCP × Часовая норма на 1 UCP

Трудоёмкость = 63.38 × 20 = 1267.6 человеко-часов

Длительность в неделях = 1267.6/80 ​= 15.845 недель ≈ 4 месяца

1. **Оценка трудоёмкости методом Function Points**
   1. **«Сырьевые» функциональные точки**

Функциональные точки распределяются по 5 категориям:

Таблица 4 – категории функциональных точек

|  |  |  |  |
| --- | --- | --- | --- |
| Категория | Простая | Средняя | Сложная |
| ILF (внутр. логические файлы) | 7 | 10 | 15 |
| EIF (внешние интерфейсы) | 5 | 7 | 10 |
| EI (ввод данных) | 3 | 4 | 6 |
| EO (вывод данных) | 4 | 5 | 7 |
| EQ (запросы) | 3 | 4 | 6 |

На основе БД:

Таблица 5 – таблица функциональных точек с учетом БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория | Кол-во | Уровень сложности | Вес | Итого |
| ILF (User, Team, Role, Project, Task, Report, Comment, CommentAttachment) | 8 | Средняя | 10 | 8 × 10 = 80 |
| EIF | 0 | — | — | 0 |
| EI (Регистрация, авторизация, создание команды, создание проекта, создание задачи, редактирование задачи и др.) | 12 | Простая/Средняя | 3/4 | 12 × 3.5 = 42 (усредним) |
| EO (Отобразить профиль пользователя, проекта, задачи, отчет) | 6 | Средняя | 5 | 6 × 5 = 30 |
| EQ (Поиск задач, поиск проектов, поиск участников) | 3 | Простая | 3 | 3 × 3 = 9 |

* 1. **Рассчет FP**
  2. **Пересчет в LOC**

Для Python принято:

Тогда:

* 1. **Трудоёмкость и длительность по FP**

T — длительность проекта в месяцах

PM (Person-Months) — трудоёмкость в человеко-месяцах

KSLOC — объём кода в тысячах строк кода (Kilo Source Lines of Code)

Формулы:

где:

a,b,c,d — коэффициенты базовой модели COCOMO.

Теперь пересчитаем по COCOMO:

1. Трудоёмкость:

Посчитаем:

Тогда:

PM ≈ 28.4 человеко-месяцев

1. Длительность:

Посчитаем:

Тогда:

1. **Итоговые результаты**

Таблица 6 — Сводные результаты трудоёмкости и длительности

|  |  |  |
| --- | --- | --- |
| **Метод** | **Трудоёмкость** | **Время разработки** |
| UseCasePoints | 1267.6 ч (15.8 PM) | 4–6 мес |
| FunctionPoints | 28.4 PM | 9.2 мес |

1. **Выводы**

Методы UCP и FP дали сопоставимые оценки; UCP демонстрирует более

короткий календарный срок при команде из пяти разработчиков.

FP даёт более консервативную оценку длительности, так как опирается

на объём кода.

Основной вклад в трудоёмкость вносят операции управления пользователями/группами и обработка заданий.

Перед детальным планированием рекомендуется уточнить оценки с учётом компетенций команды и технических рисков.