

# FP Calculator

2021170820  
2021170802  
2021170824  
2021170804  
2021170826

علي يحيي زكريا فهمي  
احمد سامح احمد مختار  
فادي مجدي زكي ابراهيم  
احمد طارق محمد كمال  
كريم فؤاد شهب محمد

SWE  
Team Id : 1

# Table Of Content

Table Of Content .....	2
Introduction .....	2
Problem Statement.....	2
Input Test Data & Output.....	3
Tasks .....	4
Conclusion .....	4
References .....	4

## Introduction

In modern software development, accurate estimation of project size and effort is important for successful planning and execution. Recognizing the challenges regarding this process, we started on the development of a Function Point Calculator App. Motivated by the desire to identify project estimation practices and empower development teams with a reliable tool, this project aims to deliver a robust and user-friendly application. Our objectives focus on the creation of a user-friendly interface for inputting project details, accurate calculation algorithms coping with industry standards, and customization options to provide diverse project requirements. By providing a comprehensive solution that facilitates accurate function point analysis, our vision is seeking to enhance project management practices and contribute to the efficiency and success of software development methods.

## Problem Statement

Facilitating the measurement of software size by decomposing the system into data function types that describe how the system stores information or requests it from external systems, and transaction function types that depict the information flow between the user and the system. A more robust walkthrough anticipating any technological complexity that may occur will result from management and engineers emphasizing the size of software products by outlining each one's complexity.

# Input Test Data & Output

## Calculate UFP

### *inputs*

3 simple X 3 = 9

4 average X 4 = 16

1 complex X 6 = 6

### *outputs*

6 average X 5 = 30

2 complex X 7 = 14

### *files*

5 complex X 15 = 75

### *inquiries*

8 average X 4 = 32

### *interfaces*

3 average X 7 = 21

4 complex X 10 = 40

•Unadjusted function points (UFP) =  $(3 \times 2 + 4 \times 4 + 1 \times 6) + (6 \times 5 + 2 \times 7) + 5 \times 15 + 8 \times 4 + (3 \times 7 + 4 \times 10) = 243$

## Calculate DI & TCF

Complex internal processing = 3

Code has to be reusable = 2

High performance = 4

Multiple sites = 3

Distributed processing = 5

Others has no influence

DI =  $3 + 2 + 4 + 3 + 5 = 17$

TCF =  $(0.65 + 0.01 \times DI)$

=  $(0.65 + 0.01 \times 17)$

= 0.82

Adjusted FP = UFP x TCF

=  $243 \times 0.82 = 199.26$

## Output of our program:

Summary

Summary For All The Calculations

UFP

243

DI

17.00

TCF

0.82

FP

199.26

Please Select Your Code Language To Calculate LOC :

LOC

## Tasks

Tasks	Names
UFP	علي يحيي زكريا فهمي
Summary	احمد سامح احمد مختار
Main Page	فادي مجدي زكي ابراهيم
DI	احمد طارق محمد كمال
GUI and TCF	كريم فؤاد شهاب محمد

## Conclusion

In Conclusion, this project pointed how a certain function type complexity would affect the software size leading to in an unwanted increase in cost and effort. So, a simplified architecture of the system is crucial to a more resilient walkthrough of development mitigating any waste of human power on over engineered implementation. Also, it has highlighted the need to take into consideration any technical complexities that does not relate to the internal system unit like installation ease, transaction rate, etc. Such precautions would save the stakeholders from late delivery of the product if they were aware of it during the planning and analysis phase.

# References

- Lectures
- Labs