Class: T.E /Computer Sem – V / Software Engineering

Practical No:	4
Title:	**Calculating function points of the Project**
Date of Performance:	**14th August 2023**
Roll No:	**9539**
Team Members:	**Crystal Fernandes**

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Class: T.E /Computer Sem – V / **Software Engineering**

Signature of the Teacher:

PROJECT TITLE- NGOFINDER APPLICATION

Volunteer Registration Information (User Input = 1, Weight = 6):
Number of user inputs: 1 (for personal details, location, skills, etc.)
Weight for this category: 6 (moderate complexity)

NGO Registration Information (User Input = 1, Weight = 6):
Number of user inputs: 1 (for organizational details, mission, etc.)
Weight for this category: 6 (moderate complexity)

Volunteer Preferences (User Input = 1, Weight = 6):
Number of user inputs: 1 (for preferences)
Weight for this category: 6 (moderate complexity)

Volunteer Availability (User Input = 1, Weight = 6):
Number of user inputs: 1 (for availability)
Weight for this category: 6 (moderate complexity)

NGO Event Details (User Input = 1, Weight = 6):
Number of user inputs: 1 (for event details)
Weight for this category: 6 (moderate complexity)

Existing Data (User Input = 0, Weight = 0):
Number of user inputs: 0 (no explicit user inputs)
Weight for this category: 0 (no effort required)

Volunteer-NGO Match (User Output = 1, Weight = 4):
Number of user outputs: 1 (for volunteer-NGO matching)
Weight for this category: 4 (moderate complexity)

Event Schedule (User Output = 1, Weight = 4):
Number of user outputs: 1 (for event scheduling)
Weight for this category: 4 (moderate complexity)

Volunteer Notifications (User Output = 1, Weight = 4):
Number of user outputs: 1 (for volunteer notifications)
Weight for this category: 4 (moderate complexity)

NGO Notifications (User Output = 1, Weight = 4):
Number of user outputs: 1 (for NGO notifications)
Weight for this category: 4 (moderate complexity)

Volunteer Ratings and Feedback (User Output = 1, Weight = 4):
Number of user outputs: 1 (for volunteer ratings and feedback)
Weight for this category: 4 (moderate complexity)

Admin Reports (User Output = 0, Weight = 0):
Number of user outputs: 0 (no explicit user outputs)

Weight for this category: 0 (no effort required)

CalculatING the Component Assessment Factor (CAF):
CAF = (User Input Weight * User Input Count) + (User Output Weight * User Output Count)
CAF = (6 * 6) + (4 * 6)
CAF = 36 + 24
CAF = 60

Calculating the Function Points (FP):
F = 14 * Scale Factor (provided as 4)
F = 14 * 4
F = 56

CAF in the formula for calculating Function Points (FP):
FP = Total Count * (0.65 + (0.01 * F))
FP = 60 * (0.65 + (0.01 * 56))
FP = 60 * (0.65 + 0.56)
FP = 60 * 1.21
FP = 72.6 (rounded to one decimal place)

The Function Point for NGO Finder Application is 72.6

POSTLAB

a) Critically evaluate the Function Point Analysis method as a technique for software sizing and estimation, discussing its strengths and weaknesses:

Strengths:
- Independence from Implementation Technology: Function Point Analysis (FPA) focuses on software functionality and user interactions rather than the underlying technology. This makes it technology-independent and suitable for various software development approaches.
- User-Centric: FPA emphasizes the user's perspective, which aligns with the user's needs and expectations. This user-centric approach improves communication between development teams and stakeholders.
- Standardized Measurement: FPA provides a standardized method for measuring software size, making it easier to compare and estimate projects consistently.
- Estimation Accuracy: When used correctly, FPA can result in reasonably accurate software size and effort estimates, which can aid in project planning and resource allocation.
- Risk Identification: The FPA process encourages thorough analysis of software requirements, helping to identify potential risks and issues early in the project lifecycle.

Weaknesses:

- Complexity: FPA can be complex and time-consuming, especially for large and complex software systems. It may require a skilled analyst to perform the assessment accurately.
- Subjectivity: FPA assessments can be subjective, as different analysts may assign different weights and counts to the same components. This subjectivity can lead to variability in estimates.
- Learning Curve: Training is required to become proficient in FPA, which can be a barrier for small teams or organizations with limited resources.
- Limited Scope: FPA focuses primarily on functional aspects of software and may not fully account for non-functional requirements or architectural complexities.
- Dependency on Documentation: FPA relies heavily on comprehensive and accurate documentation of software requirements. If requirements are poorly documented or change frequently, FPA estimates may become less reliable.

b) Apply the Function Point Analysis technique to a given software project and determine the function points based on complexity and functionalities:

Function Point Analysis requires a detailed examination of the software project's functional and non-functional requirements, user interactions, and data flows. I have previously applied FPA to estimate function points for the "NGO Finder" application based on the provided requirements.

c) Propose strategies to manage and mitigate uncertainties in function point estimation and how they can impact project planning and resource allocation:

- Managing uncertainties in function point estimation is crucial for project planning and resource allocation. Here are some strategies to address uncertainties:
- Requirement Elicitation and Documentation: Invest time and effort in thorough requirement elicitation and documentation. Clear and complete requirements reduce uncertainty and improve estimation accuracy.
- Historical Data Analysis: Use historical data from previous projects to calibrate and validate FPA estimates. Analyzing past projects can reveal patterns and trends that help refine future estimates.
- Expert Review: Involve experienced FPA analysts who can provide expertise and guidance during the estimation process. Peer reviews can help identify errors and improve estimates.
- Sensitivity Analysis: Perform sensitivity analysis to assess the impact of varying assumptions and inputs on the final estimate. This helps identify critical factors affecting estimation uncertainty.
- Buffer Management: Include contingency reserves in project planning to account for estimation uncertainties. These buffers can be used to accommodate unexpected changes or challenges.
- Progressive Elaboration: Apply progressive elaboration by updating and refining estimates as the project progresses and more information becomes available. This approach adapts to changing circumstances.
- Communication and Transparency: Maintain open communication with stakeholders regarding estimation uncertainties. Transparent communication builds trust and allows for better decision-making.

- Scenario Planning: Develop different estimation scenarios based on optimistic, pessimistic, and most likely assumptions. This approach helps assess potential project outcomes under varying conditions.