Department of Computer Engineering

Academic Term: First Term 2023-24

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

Practical No:	**4**
Title:	**Function Point**
Date of Performance:	17/08/2023
Roll No:	9605
Team Members:	

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:

Department of Computer Engineering

Academic Term: First Term 2022-23

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

FUNCTION POINT

Step-1:
F = 14 * scale
Scale varies from 0 to 5 according to character of Complexity Adjustment Factor (CAF). Below table shows scale:

0 No Influence
1 Incidental
2 Moderate
3 Average
4 Significant

- Essential Step-2: Calculate Complexity Adjustment Factor (CAF). CAF = 0.65 + (0.01 * F)
Step-3: Calculate Unadjusted Function Point (UFP).
Step-4: Calculate Function Point. FP = UFP * CAF

TABLE (Required)
Calculation:
User Input = 10
User Output = 5
User Inquiries = 20
User Files = 6
External Interface = 4
Scale = 3

Signature of the Teacher:

Department of Computer Engineering

Academic Term: First Term 2022-23

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

F = 14 * 3 = 42

CAF = 0.65 + (0.01 * 42) = 1.07

UFP = (15*4) + (6*5) + (25*4) + (7*10) + (3*7) = 60+30+100+70+21=281

Function Point = 281 * 1.07 = 300.67

Unadjusted Function Points (UFP): 281

Complexity Adjustment Factor (CAF): 1.07

Function Points (FP): 300.67

POSTLAB:

1. **Critically evaluate the Function Point Analysis method as a technique for software sizing and estimation, discussing its strengths and weaknesses.**
   Function Point Analysis (FPA) is a widely used technique for software sizing and estimation. It evaluates software based on the functionality it delivers to end-users. While FPA has its strengths, it also comes with several weaknesses. Let's critically evaluate this method:

   Strengths of Function Point Analysis:

   1. User-Centric: FPA focuses on the functionality provided to end-users, making it a user-centric method. This aligns well with the primary goal of software development – delivering value to users.

Signature of the Teacher:

2. Technology Agnostic: FPA is technology-agnostic, which means it doesn't depend on the underlying technology or programming language used. This makes it applicable to various types of software projects.

3. Objective and Repeatable: FPA relies on well-defined rules and guidelines, leading to objective and repeatable results. Different analysts following the same guidelines should arrive at similar estimates.

4. Basis for Benchmarking: FPA can serve as a basis for benchmarking and historical data analysis. Organizations can use past project data to improve future estimates and project planning.

5. Communication and Alignment: FPA encourages communication and alignment between business stakeholders and the development team. It helps clarify requirements and expectations, reducing misunderstandings.

Weaknesses of Function Point Analysis:

1. Complexity: FPA can be complex, particularly for large and complex software systems. Counting function points involves assessing multiple factors, such as user inputs, outputs, inquiries, and interfaces, which can be time-consuming.

2. Subjectivity: While FPA aims to be objective, there's still an element of subjectivity involved, especially in determining the complexity of functions. Different analysts may interpret guidelines differently.

3. Limited for Non-Functional Requirements: FPA primarily focuses on functional requirements (e.g., user interfaces, reports). It may not adequately account for non-functional requirements like performance, security, and scalability.

4. Learning Curve: Training and gaining proficiency in FPA can be time-consuming and costly.

Signature of the Teacher:

Organizations need skilled analysts who understand the nuances of the method.

5. Dependency on Requirements: FPA relies heavily on having well-defined and detailed requirements upfront. In agile or iterative development environments, where requirements evolve, applying FPA can be challenging.

6. Doesn't Consider Developer Productivity: FPA doesn't account for developer productivity or team experience. Two projects with the same function points may have vastly different development efforts if the teams have different skill levels.

7. Influence of External Factors: FPA doesn't consider external factors that can impact software development, such as changes in technology, market conditions, or regulatory requirements.

8. Limited for Early Estimation: FPA is more suited for detailed estimation in later project stages when requirements are well-defined. It may not be ideal for rough, early-stage estimates.

In summary, Function Point Analysis is a valuable technique for software sizing and estimation, especially for projects with well-defined requirements and a focus on user functionality. However, it's not without its challenges, including complexity, subjectivity, and limitations in addressing non-functional requirements. Organizations should carefully consider the context and specific needs of their projects when deciding whether to use FPA as their estimation method.

2. Apply the Function Point Analysis technique to a given software project and determine the function points based on complexity and functionalities.

Certainly, here are the steps to apply the Function Point Analysis (FPA) technique to a software project:

1. Identify Functional User Requirements: Begin by identifying and documenting all the functional user requirements of the software project. These requirements should describe what the software is supposed to do from a user's perspective. This can include user inputs, outputs,

inquiries, and interfaces.

2. Categorize Functions: Categorize the identified functions into five main categories based on their characteristics:
   - External Inputs (EI): User inputs that result in data processing within the software.
   - External Outputs (EO): User-driven data outputs generated by the software.
   - External Inquiries (EQ): User-driven inquiries that result in data retrieval.
   - Internal Logical Files (ILF): Data maintained by the software.
   - External Interface Files (EIF): Data shared with external systems.

3. Assign Weights: Assign complexity weights to each function. FPA typically uses a scale of "Low," "Average," and "High" complexity. These weights are based on the complexity of processing and the number of data elements involved.

4. Count Function Points: Calculate the unadjusted function points for each category by multiplying the number of functions in each category by their respective complexity weights and summing them up.

5. Apply Complexity Adjustment: Assess the general system characteristics (GSCs) that affect the complexity of the project, such as distributed data processing, performance requirements, and transaction rates. Assign complexity factors based on these characteristics and apply them to adjust the function points.

6. Calculate Adjusted Function Points: Multiply the unadjusted function points by the adjustment factor to obtain the adjusted function points (AFP).

7. Use Function Points for Estimation: Function points can be used for various software project management activities, including effort estimation, project scheduling, and resource allocation. You can use historical data or industry benchmarks to estimate the effort required for a project based on the AFP.

Signature of the Teacher:

8. Review and Validate: Finally, review and validate the function point count with relevant stakeholders to ensure accuracy and completeness.

It's important to note that applying FPA effectively requires training and expertise in the method, and there are software tools available that can assist with the process.

Please provide specific details about the software project, such as the number of functions in each category and the complexity of each function, to perform a detailed calculation of function points for your project.

3. **Propose strategies to manage and mitigate uncertainties in function point estimation and how they can impact project planning and resource allocation.**

Certainly, here are strategies to manage and mitigate uncertainties in function point estimation and how they can impact project planning and resource allocation:

1. Historical Data Analysis:
   - Use historical project data as a reference. Analyze past projects with similar characteristics to identify patterns and trends in function point estimation. This can provide a basis for more accurate estimation and reduce uncertainties.

2. Expert Judgment:
   - Engage experienced estimators and domain experts in the estimation process. Their insights and expertise can help identify potential risks and uncertainties, allowing for more informed estimates.

3. Sensitivity Analysis:
   - Perform sensitivity analysis by varying input parameters (e.g., function point counts, complexity factors) to assess the impact on project outcomes. This helps in understanding the range of potential project outcomes and identifying key drivers of uncertainty.

Signature of the Teacher:

4. Monte Carlo Simulation:
   - Use Monte Carlo simulation techniques to model and analyze the impact of uncertainties. By running multiple simulations with different input scenarios, you can estimate the probability distribution of project outcomes, helping in risk assessment and mitigation planning.

5. Buffer Management:
   - Incorporate contingency buffers into project schedules and budgets. These buffers can account for uncertainties in function point estimation. However, it's important to track and manage these buffers to avoid unnecessary project bloat.

6. Risk Identification and Mitigation:
   - Conduct a thorough risk analysis to identify potential risks associated with function point estimation. Develop risk mitigation plans to address these uncertainties. For example, if there's uncertainty about requirements, consider iterative development methodologies.

7. Continuous Monitoring:
   - Continuously monitor and update function point estimates as the project progresses and more information becomes available. Regularly review and refine estimates to reflect the evolving project scope and requirements.

8. Range Estimation:
   - Provide a range of function point estimates (e.g., minimum, most likely, maximum) instead of a single point estimate. This conveys the degree of uncertainty and allows for more flexible project planning.

9. Stakeholder Involvement:
   - Involve project stakeholders, including clients and end-users, in the estimation process. Their insights can help clarify requirements and reduce misunderstandings, ultimately reducing estimation uncertainties.

Signature of the Teacher:

10. Documentation and Assumptions:
    - Clearly document all assumptions made during the estimation process. This helps in identifying potential sources of uncertainty and allows for validation as the project progresses.

11. Review and Quality Assurance:
    - Implement peer reviews and quality assurance processes for function point estimation. Having multiple experts review and validate the estimates can help identify and rectify estimation errors and uncertainties.

12. Agile and Adaptive Approaches:
    - Consider using agile or adaptive project management approaches that accommodate changing requirements and scope. These methodologies are more adaptable to uncertainties and allow for frequent reassessment and adjustment of project plans.

13. Risk Response Planning:
    - Develop a comprehensive risk response plan that outlines strategies for addressing identified risks related to function point estimation. Include contingency plans for potential estimation errors.

Managing uncertainties in function point estimation is essential for effective project planning and resource allocation. By implementing these strategies, you can reduce the impact of estimation uncertainties, enhance project predictability, and improve the allocation of resources to meet project goals and expectations.

Signature of the Teacher: