CS342 Software Engineering

Dr. Ismail Hababeh

German-Jordanian University

Lecture 8

Software Project Estimation

Software Project Estimation

- Software Project plan includes estimating:
  - Cost
  - Duration
  - Effort
  - Staff scheduling
  - Risk management (identification, analysis and planning)
  - Quality Assurance, configuration management, and other resources.

Project Estimation Techniques

- Empirical Techniques: Derived from prior experience, experiments and observations rather than theory. (e.g. Expert judgment, Delphi)
- Heuristics Techniques: A mathematical formula is used to estimate the resource $\boxed{\text{Estimated Resource} = C_1 \cdot e^{D}{}_1 + C_2 \cdot e^{D}{}_2 + \ldots}$

  where C1 , C2, D1, D2, … are constants (historical values).

  e is the characteristics of the software (historical data).
- Analytical Techniques: A scientific basis estimation techniques

Example: Halstead’s software science, useful for estimating maintenance efforts, and starts with basic assumption about the project to derive the required results.

Software Estimation and Measurement

- Software project functionality size is estimated and measured by:
- Lines of code (LOC)
  - Disadvantages:
    - Measures the code only
    - Could not measure the software functionality.
- Functional Points
  - Estimate and measure the software functionality

Functional Points Analysis

- Estimation technique for a software project in terms of function size of the product called (Function Point).
- Estimation and measurement are independent of technology.
- Measurements are more significant than opinions.

Software Project Functional Points

- Standard software measurement ISO/IEC 20926:2003.
- Consider all software functionalities provided to the customer.
- Performed based on delivered product.
  - Measure the amount of business functionality.
  - Measure the information system (as a product) provides to the user.

Functional Points Advantages

- Reduces the risk of
  - Project termination
  - Unstable requirements
  - Poor quality
  - Cost and schedule overruns

Functional Points Key Factors

Functional Points Counts are key factors to

- Data Measurement and Analysis
- Requirements Management
- Project Planning
  - Determine the estimates of Effort and Cost
  - Establish the Budget and Schedule

Functional Points Key Factors

- Project Monitoring and Control
- Integrated Project Management
  - Use organizational process assets for planning project activities
- Quantitative Project Management
  - Manage Project Performance
  - Select Measures and Analytic Techniques

Functional Points Estimation Components

Functional Points are estimated by computing the total Adjusted Function Points*:

Total Adjusted Function Points = Adjusted Processing Complexity * Unadjusted Function Points.

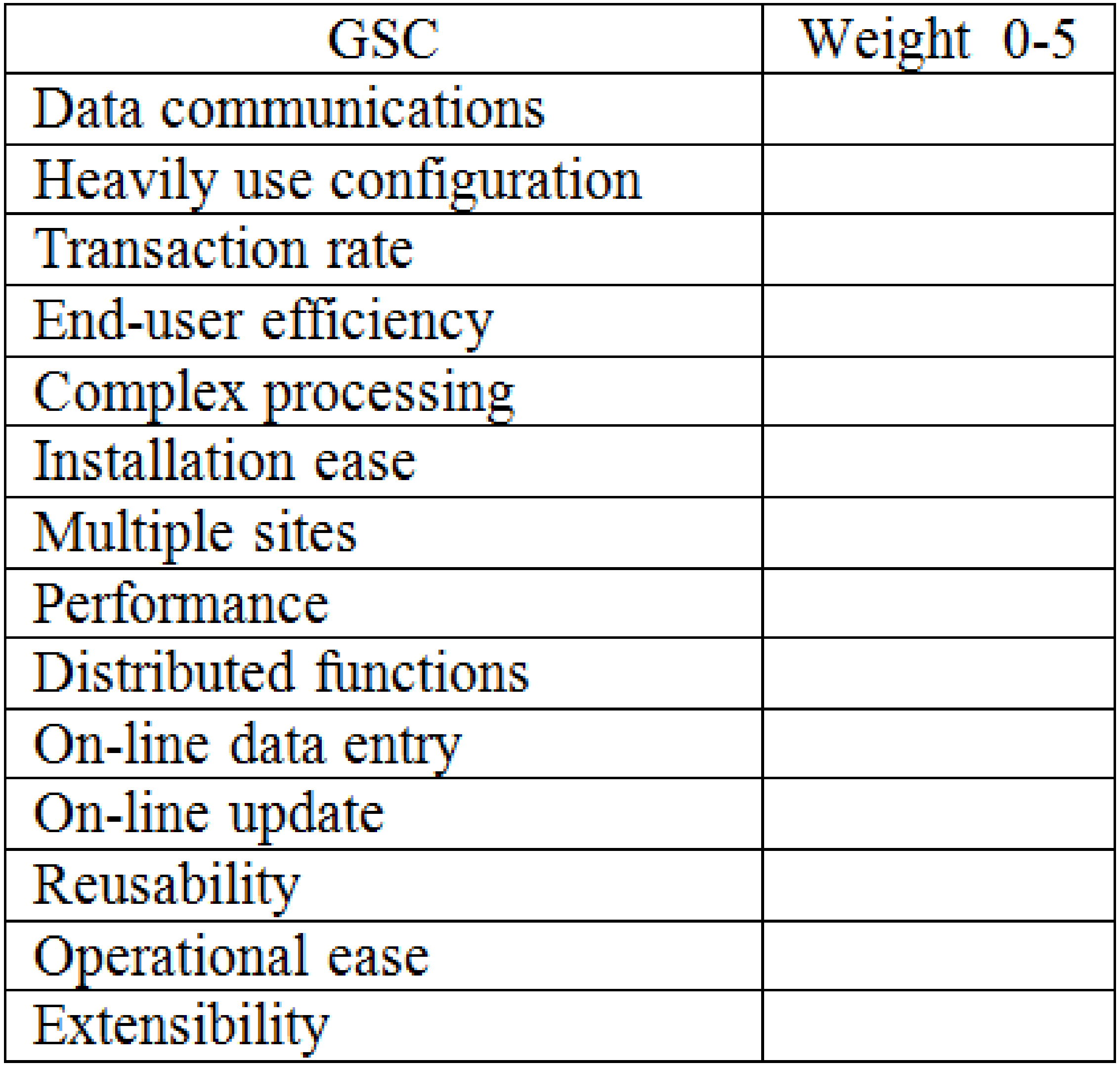
*Source: https://www.wiley.com/college/info/dennis241008/resources/Resources/proj3.htm

Adjusted Processing Complexity

Processing Complexity is based on 14 General System Characteristics (GSC) that effect the application processing complexity.

The influence of each characteristic is rated on the scale (0 – 5) where 0 indicates no effect on processing complexity, and 5 means the highest effect.

General System Characteristics



GSC	Weight 0-5
Data communications	
Heavily use configuration	
Transaction rate	
End-user efficiency	
Complex processing	
Installation ease	
Multiple sites	
Performance	
Distributed functions	
On-line data entry	
On-line update	
Reusability	
Operational ease	
Extensibility	

Adjusted Processing Complexity

Adjusted Processing Complexity is computed as follows:

Adjusted Processing Complexity =

0.65 + (0.01 * Total GSC Processing Complexity)

Unadjusted Functional Points

- Unadjusted Functional Points is based on the complexity of the following factors: Inputs, Outputs, Queries, Interfaces, and Files.
- The factors complexity is categorized in 3 levels; low, medium, and high.
- Each complexity level is rated on a scale that reflects the influence on the application processing.

Unadjusted Functional Points Complexity

Unadjusted Functional Points complexity scale is described as follows:

- Input (3 - 6)
- Output (4 - 7)
- Queries (3 - 6)
- Interfaces (5 - 10)
- Data Files (7 - 15)

Unadjusted Functional Points Computation

Unadjusted Functional Points is computed as follows:

Unadjusted Functional Points =

Σ (Input counts * scale

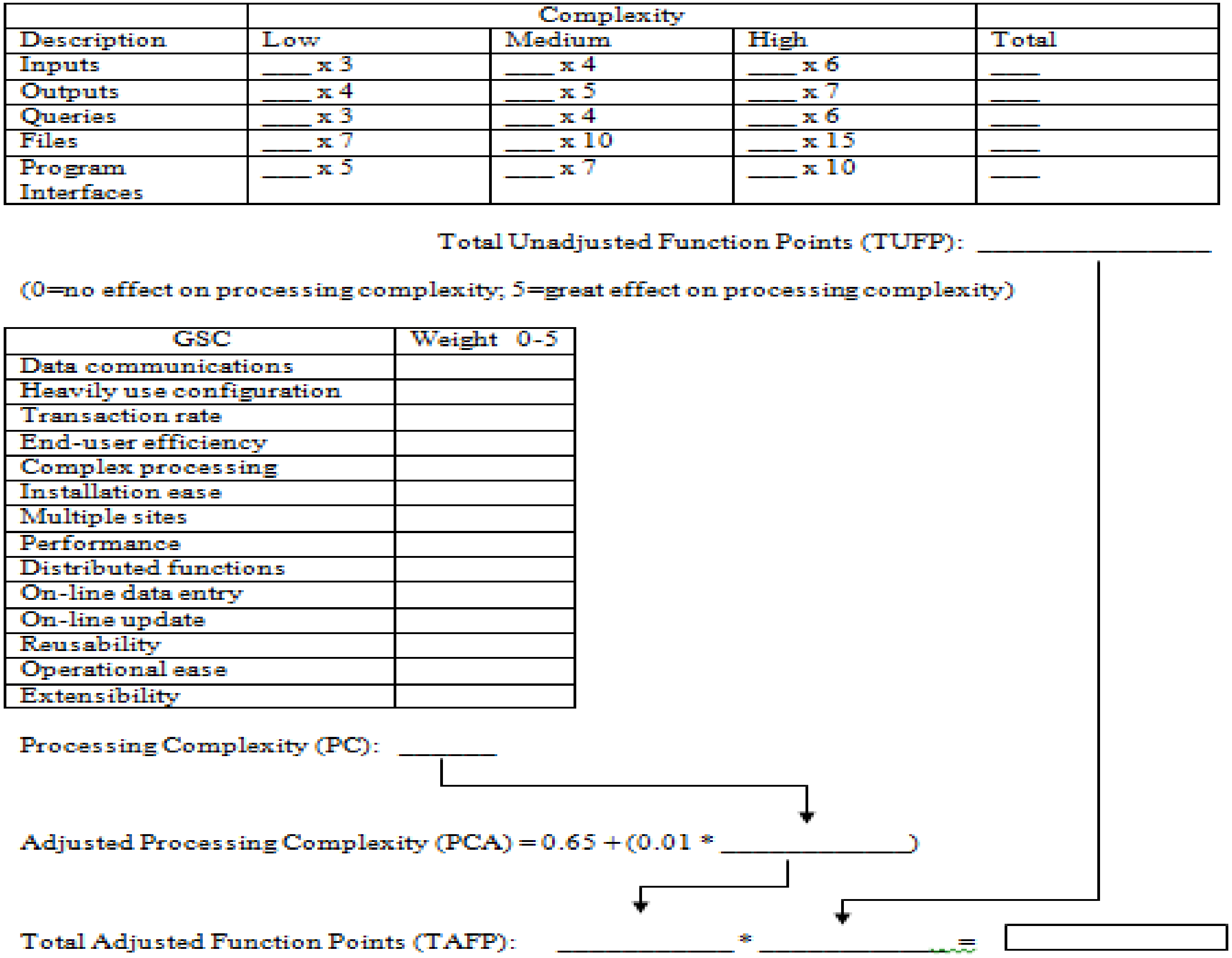
+ Output counts * scale

+ Queries counts * scale

+ Interfaces counts * scale

+ Files counts * scale)

Functional Points Estimation Summary



	Complexity			
Description	Low	Medium	High	Total
Inputs	____ x 3	____ x 4	____ x 6	____
Outputs	____ x 4	____ x 5	____ x 7	____
Queries	____ x 3	____ x 4	____ x 6	____
Files	____ x 7	____ x 10	____ x 15	____
Program Interfaces	____ x 5	____ x 7	____ x 10	____

Total Unadjusted Function Points (TUFP): ________________

(0=no effect on processing complexity; 5=great effect on processing complexity)

GSC	Weight 0-5
Data communications	
Heavily use configuration	
Transaction rate	
End-user efficiency	
Complex processing	
Installation ease	
Multiple sites	
Performance	
Distributed functions	
On-line data entry	
On-line update	
Reusability	
Operational ease	
Extensibility	

Processing Complexity (PC): ________

Adjusted Processing Complexity (PCA) = 0.65 + (0.01 * ______________)

Total Adjusted Function Points (TAFP): ____________ * ______________ =

Functional Points Estimation - Example

Consider a software project with the following functional units :

- Number of user inputs = 50
- Number of user outputs = 40
- Number of user queries = 35
- Number of external interfaces = 4
- Number of user files = 6

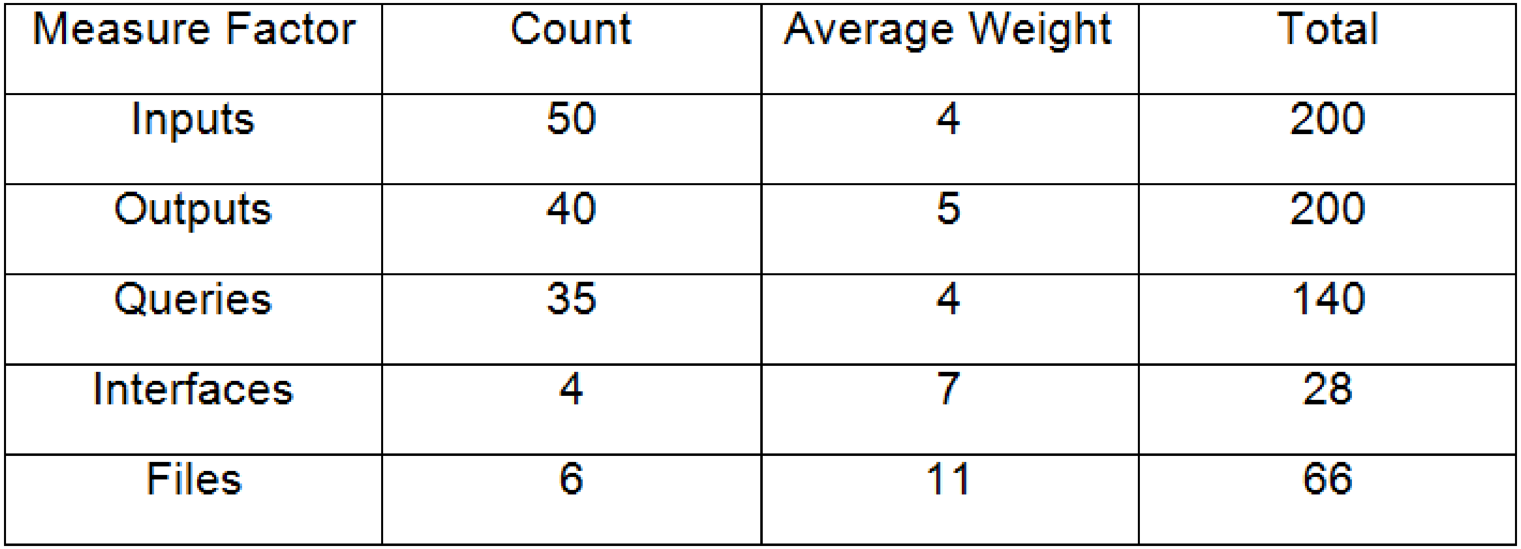
Compute the project function points estimation assuming the average for all complexity adjustment and GSC characteristic weight factors.

Functional Points Estimation - Example

Averages of Unadjusted Function Points factors are computed as follows:

- Average = { Input (3 - 6) = 4,
  Output (4 - 7) = 5,
  Queries (3 - 6) = 4,
  Interfaces (5 - 10) = 7,
  Files (7 - 15) = 11}
- Average of GSC characteristic weight (0 – 5) = 3.

Functional Points Estimation - Example



Measure Factor	Count	Average Weight	Total
Inputs	50	4	200
Outputs	40	5	200
Queries	35	4	140
Interfaces	4	7	28
Files	6	11	66

Functional Points Estimation - Example

Function Points = Σ (0.65 + (0.01 * (14 * 3)) * (200 + 200 + 140 + 28 + 66))

= 634 * (0.65 + 0.42)

= 634 * (1.07)

= 678

Functional Points Disadvantages

- Not applicable to all types of software.
- Doesn’t include all functional characteristics of real-time software.
- Depends on estimation equation values (like 0.65, 0.01, GSC influence values) but not metrics.
- Designed for business information systems where data dimension is predefined.
- Inappropriate for many engineering and embedded systems.

Functional Points Alternatives

- Feature Points
  - Suitable for real-time, process-control and embedded systems.
  - Counts a new software characteristics; like algorithms.

Functional Points Alternatives

- 3D Function Points
- Suitable for software applications that process function and control tasks.
- Based on 3 basic metrics:
  - Data dimension
  - Functional dimension
  - Control dimension
- The influence values of the characteristics are computed to provide an indication of the functionality.