

## Functional Points Analysis for CAR: Course Analysis and Registration

Akshat Totla, Juan Piñeros, Liyan Ibrahim

### List of Functionalities

Functionality	Information Domain	Weighting
User requests Course Suggestion generation	EO	Complex
Live Professor Rating	EQ	Simple
User queries to locate specific courses	EI	Simple
Live Class Status	EQ	Simple
User Consults Study Plan and Major Requirements	EO	Simple
Saving and Updating 4 Year Plan	EI	Average
Exporting 4 Year Plan	EO	Average
Verifying Classes	EI	Simple
Admin modifies Courses and Requirements	EI	Average
Periodically updating course database from API	EQ	Complex
Database for course lists, requirements and individual study plans	ILF	Average

### Function Point Calculation

Information Domains	Count	Weighting			Total
		Simple	Average	Complex	
EIs	4	2*3	2*4	0*6	14
EOs	3	1*4	1*5	1*7	16
EQs	3	2*3	0*4	1*6	12
ILFs	1	0*7	1*10	0*15	10
EIFs	0	0*5	0*7	0*10	0
Total Unadjusted Function Points					52

### Processing Complexity Calculation

Complexity Weighting Factor	Description	Value (1-5)
Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?	3
Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?	0
Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?	2
End user efficiency	Was the application designed for end-user efficiency?	3
Complex processing	Does the application have extensive logical or mathematical processing?	2
Installation ease	How difficult is conversion and installation?	1
Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple	1

	organizations?	
Performance	Was response time or throughput required by the user?	0
Distributed data processing	How are distributed data and processing functions handled?	1
Online data entry	What percentage of the information is entered online?	1
Online updating	How many ILF's are updated by On-Line transaction?	1
Reusability	Was the application developed to meet one or many user's needs?	2
Operational ease	How effective and/or automated are start-up, back-up, and recovery procedures?	1
Extensibility	Was the application specifically designed, developed, and supported to facilitate change?	3
<b>Total Processing Complexity (TPC)</b>		21

#### Adjusted Processing Complexity (APC)

$$APC = 0.65 + (0.01 * TPC)$$

$$APC = 0.65 + (0.01 * 21) = 0.86$$

#### Total Adjusted Function Points (TAFP)

$$TAFP = TUFP * APC$$

$$TAFP = 52 * 0.86 = 44.2$$

#### TAFP to LOC

Language	Ratio and FP	LOC per FP	LOC
Python	40% - 17.68	53.33	942.9
JavaScript	40% - 17.68	71.11	1257.2
HTML	20% - 8.84	15	132.6
<b>Total LOC</b>			2332.7

### Estimating the Effort:

Project	a	b
Application Programs	2.4	1.05
Utility Programs	3.0	1.12
System Programs	3.6	1.20

$$\begin{aligned}\text{Effort} &= a + (\text{LOC}/1000)^b \\ &= 2.4 * (2332.7/1000)^{1.05} \\ &= 5.83 \text{ person months}\end{aligned}$$

### Estimating the Schedule Time

Project	c	d
Application Programs	2.5	0.38
Utility Programs	3.0	0.35
System Programs	3.6	1.20

$$\begin{aligned}\text{Time} &= c * \text{Effort}^d \\ &= 2.5 * (5.83)^{0.38} \\ &= 4.88 \text{ months}\end{aligned}$$

### Estimating the number of persons

$$\begin{aligned}&= \text{Effort} / \text{Time} \\ &= 5.83 / 4.88 \\ &= 1.19 \text{ persons}\end{aligned}$$