**PROJECT SIZE ESTIMATION DOCUMENT**

**Function Point:**

Function Point Analysis was developed first by Allan J. Albrecht in the mid 1970s. It was an attempt to overcome difficulties associated with lines of code as a measure of software size, and to assist in developing a mechanism to predict effort associated with software development.

Function points are a unit measure for software much like an hour is to measuring time, miles are to measuring distance or Celsius is to measuring temperature. Function Points are an ordinal measure much like other measures such as kilometers, Fahrenheit, hours, so on and so forth.

**FP Analysis for E-Logistics Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **Count** | **Simple** | **Average** | **High** |
| External Input | 8 | 3 | 4 | 6 |
| External Output | 8 | 4 | 5 | 7 |
| Files | 10 | 7 | 10 | 15 |
| Interfaces | 10 | 5 | 7 | 10 |
| External Inquires | 7 | 3 | 4 | 6 |

We are considering the complexity of project to be **Average**

UFP stands for Unadjusted Function Point

***UFP*** = 8\*4 + 8\*5 + 10\*10 + 7\*10+ 6\*7 = ***284***

The value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. The degrees of influence range on a scale of zero to five, from no influence to strong influence.

Value Adjustment Factors:

Data Communications 4

Distributed Functions 2

Performance 4

Heavily Used Configuration 4

Transaction Rate 4

Online Data Entry 5

End User Efficiency 4

Online Update 4

Complex Processing 4

Reusability 2

Installation Ease 1

Operational Ease 1

Multiple Sites 1

Facilitate Change 3

***DI*** = Sum of Influence = ***43***

***VAF*** = 0.65 + 0.01 \* DI = 0.65 + 0.01\*43 = ***1.08***

***FP = UCF \* VAF =*** 284 \* 1.08 = ***306.72 FP/PM***

***LOC = FP \* 53 = 16256.16***

***KLOC = 16K***

Thus E-logistics Management has **16K** (kilo) lines of code

**Development Time Calculation for E-logistics Management**

**COCMO MODEL**:

Barry Boehm wrote 'Software Engineering Economics', published in 1981, he introduced an empirical effort estimation model (COCOMO - **CO**nstructive **CO**st **MO**del) that is still referenced by the software engineering community.

**COCOMO Development Modes**:

1. **Organic** - small relatively small, simple software projects in which small teams with good application experience work to a set of flexible requirements.
2. **Embedded** - the software project has tight software, hardware and operational constraints.
3. **Semi-detached** - an intermediate (in size and complexity) software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modes** | **a** | **b** | **d** |
| Organic (KLOC < 20K) | 2.4 | 1.05 | 0.38 |
| Semi-Detached  (20K < KLOC < 200K) | 3.6 | 1.20 | 0.32 |
| Embedded (KLOC > 200K) | 3.0 | 1.12 | 0.35 |

***Development Time = Ed \* 2.5***

Where,

***E = a \* (KLOC)b***

Now, KLOC = 16K

Hence,

**E** = 2.4 \* 161.05 = **44 Person Months**

Therefore,

**DT** = 2.5 \* Ed = 2.5 \* 440.38 = **10.5 months**

Development Time for Online Plant Nursery is **10.5 Months**