

# Practical Use of Function Point Analysis

A practical course that gives participants an understanding of the fundamental principles of software measurement and an ability to utilise those principles to understand, manage and measure software requirements.

Participants will discuss relationships between functional size, qualitative requirements, development and operational capability, and work effort. They will establish the groundwork for repeatable, defined, proven measurement and estimating practices.

The course is intended as an introduction for people making functional size measurements of both proposed and implemented software applications. It actively involves participants in making estimates and measurements of functional size at various stages in the software lifecycle, for a variety of scenarios, including new developments and enhancement projects.

Based on a customer oriented view of software it provides a quantitative approach to negotiating and managing requirements.

## Objectives

By attending this course, participants will ...

- ◆ understand the need for a repeatable and standardised measure of functional size
- ◆ appreciate the desirable characteristics of a software metric
- ◆ learn how the required functional size and qualitative performance of a product are related to capability, resources and effort
- ◆ understand how to derive a requirements based measure of functional size at minimal cost during a project
- ◆ learn how to measure functional size for new developments, enhancement projects and existing applications - irrespective of development method and technology.
- ◆ learn how to use a catalogue of logical transactions as a tool for requirements management and functional sizing from project conception through to support

## Syllabus

- ◆ **Introduction**  
The participants' goals. Objectives of the course, the course format and structure.
- ◆ **Foundation & Principles**  
Why measure? Problems with projects and the commitment process. Characteristics of measures. Using a logical transaction

catalogue. FPA work breakdown structure. Definitions. History and uses of FPA.

- ◆ **Sizing Software**  
FPA and the software product lifecycle. Customer and purpose. The application boundary. How to make the base counts. Logical transactions. Data element types. Calculating functional size.
- ◆ **Adjusting for Quality**  
Factors that introduce complexity. How to evaluate and adjust the problem size.
- ◆ **Choosing the Boundary**  
Functionality delivered to users. Different views, different sizes. Partitioned systems. Reused components.
- ◆ **Measure for a Purpose**  
Original development. Preventative, corrective, adaptive and perfective maintenance. The application size and the size of a project.
- ◆ **Sizing Installed Applications**  
Physical files and logical models. Deriving an equivalent logical model. Sizing batch flows. Sizing GUIs and windows.
- ◆ **Using FPA in Estimating**  
Early size estimation. Worktime Distribution Model. Effort, duration, and staffing profiles.
- ◆ **Workshop**  
Practical application of the techniques to the participants own project material. Resolution of issues.
- ◆ **Summary**  
Conclusions, questions, answers and action plans.

## Who should attend?

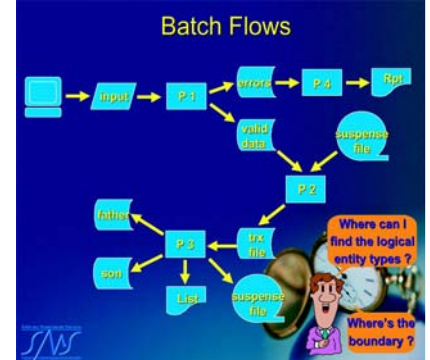
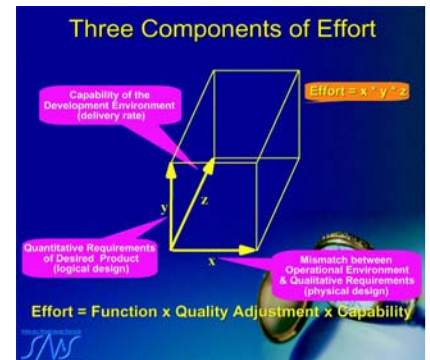
This course is designed for:

- ◆ Project managers and team leaders who are responsible for delivering quality software products on time, within budget
- ◆ Software engineers who wish to gain better control over functional requirements and requests for change
- ◆ Business users involved in specifying, agreeing and procuring software.

## Course Materials

Participants will be provided with a reference manual that includes:

- ◆ Explanatory diagrams and text
- ◆ Case studies, exercises and worked solutions



Work Type & Size

	Trigger	Change of size ?	Impact
Origination	commission	✓	functional change
Corrective	reactive	✗	defect removal
Adaptive	reactive	✓	functional change
Perfective	proactive	✗	qualitative improvement

