

# International Software Benchmarking Standards Group

## GLOSSARY OF TERMS

for

## SOFTWARE PROJECT DEVELOPMENT AND ENHANCEMENT

## Table of Contents

Introduction.....	3
CONTEXT.....	3
PURPOSE OF DOCUMENT.....	3
Part A: Terms .....	4
ACTIVITY .....	4
ADJUSTED FUNCTION POINTS (AFP) .....	4
APPLICATION GROUP.....	4
APPLICATION TYPE.....	4
ARCHITECTURE.....	7
BUSINESS AREA TYPE.....	7
CASE (COMPUTER AIDED SOFTWARE ENGINEERING).....	7
CLIENT ROLES .....	8
CLIENT SERVER .....	8
CLIENT/SERVER DESCRIPTION.....	8
COST.....	8
COUNT APPROACH .....	8
DATA QUALITY RATING.....	9
DEFECT .....	9
DEGREE OF CONFIDENCE.....	10
DEGREE OF CUSTOMISATION.....	10
DEVELOPMENT METHODOLOGY .....	10
DEVELOPMENT PLATFORM .....	10
DEVELOPMENT TECHNIQUE.....	10
DEVELOPMENT TYPE.....	10
FUNCTIONAL SIZE .....	11
FUNCTIONAL SIZE MEASUREMENT (FSM) .....	11
FUNCTIONAL SIZE UNIT (FSU).....	11
FUNCTIONAL SIZING TECHNIQUE .....	11
IMPLEMENTATION DATE .....	11
INTENDED MARKET .....	12
LANGUAGE TYPE.....	12
LIFE CYCLE ACTIVITIES .....	12
MAXIMUM TEAM SIZE .....	12
METHODOLOGY USED .....	12
METHODOLOGY ACQUISITION.....	12

NORMALISED WORK EFFORT .....	12
ORGANISATION TYPE .....	12
PACKAGED SOFTWARE CUSTOMISATION .....	12
PRIMARY PROGRAMMING LANGUAGE .....	13
PRODUCTIVITY .....	13
PROJECT .....	13
PROJECT ACTIVITY SCOPE.....	13
PROJECT EFFORT BREAKDOWN.....	13
PROJECT ELAPSED TIME.....	15
PROJECT ID .....	15
PROJECT INACTIVE TIME .....	15
PROJECT LIFE CYCLE .....	15
PROJECT WORK EFFORT.....	15
RATING .....	15
REUSABLE / CUSTOM CRITERIA.....	15
SOFTWARE SIZE .....	16
SERVER ROLES .....	16
STAGED DEVELOPMENT .....	16
SUMMARY WORK EFFORT .....	16
TARGET PLATFORM .....	16
TASK .....	17
TIME RECORDING METHODS.....	17
TYPE OF SERVER.....	17
UNADJUSTED FIUNCTION POINT RATING .....	17
UNRECORDED EFFORT .....	17
USER BASE .....	18
VALUE ADJUSTMENT FACTOR (VAF).....	18
WEB DEVELOPMENT.....	18
WORK EFFORT BREAKDOWN .....	18
Part B: Metrics.....	20
DEFECT DENSITY .....	20
MANPOWER DELIVERY RATE.....	20
PROJECT DELIVERY RATE, (PDR).....	20
SPEED OF DELIVERY .....	20

## Introduction

### CONTEXT

When the International Software Benchmarking Standards Group was formed in 1994, one of the objectives written into the original Charter was:

*“To develop the profession of software measurement by establishing a common vocabulary and understanding of terms.”*

Consistent with this objective, the ISBSG has defined terms and metrics for the purposes of:

- ❖ assisting in the collection of project data into the Repository
- ❖ standardising the way the collected data is analysed and reported

What follows is a consolidated list of ISBSG definitions and terms used. We hope this will help us to meet our Charter. We would appreciate any comments you may have to assist us in meeting this objective.

### PURPOSE OF DOCUMENT

This document provides definitions of terms used in ISBSG documents. This includes Project Data collection forms, publications and data releases. Some of these terms refer to items no longer collected by the ISBSG but which may be found in documents or analysis of earlier data.

Where appropriate these definitions have been adjusted to align with international standards:

*ISO/IEC 14143-1:1998 Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts*

*ISO/IEC 20926:2003 Software engineering -- IFPUG 4.1 Unadjusted functional size measurement method -- Counting practices manual*

*ISO/IEC 20968:2002 Software engineering -- Mk II Function Point Analysis -- Counting Practices Manual*

*ISO/IEC 24765, Software and Systems Engineering Vocabulary*

The Glossary is divided into two parts:

Part A Terms

Part B Metrics

## **Part A: Terms**

### ***ACTIVITY***

Refers to the 6 types of work undertaken during software development and maintenance, namely:

Plan, Specify, Design, Build, Test, and Implement

In earlier ISBSG publications these activities were frequently referred to as phases. For the contents of these activities see PROJECT EFFORT BREAKDOWN.

### ***ADJUSTED FUNCTION POINTS (AFP)***

A software size based on the functional size multiplied by the technical complexity adjustment. The resultant adjusted size is reported in adjusted function points (AFP) and applies to IFPUG, FiSMA, NESMA and MARK II Functional Size Measurement methods.

### ***APPLICATION GROUP***

The application group defines the purpose of the software. The four application groups are:

#### **Business Application**

Software that can enable a computer to be used to perform practical business operations or decision making. 'Business' may encompass activities in the public and/or private sectors.

#### **Real Time Application**

Changes Software that may gather data and control other software or hardware devices. This is all subject to specific timing constraints for the required responses to events.

#### **Mathematically Intensive Application**

Software with processing logic that is dominated by mathematical operations. Examples include mathematically intensive, scientific or engineering computations, processing of audio/video images.

#### **Infrastructure Software**

Software that enables application software to execute on computer hardware. Infrastructure software typically executes in one or more layers of a software architecture between the application software layer and the hardware. Examples include operating system software, middleware, device drivers, dbms, word-processing/spreadsheet software.

### ***APPLICATION TYPE***

How the application is meeting the business area requirements. Classification of an application as a type is according to its primary intended use. Application types are grouped in the following 4 categories: Business Applications, Real-Time Applications, Mathematically-Intensive Applications, and Infrastructure Software.

***Business Applications:***

**Catalogue or register of things or events**

**Customer billing**

**Customer relationship management**

**Data warehouse system**

**Decision Support**

These are interactive information systems that employ database technology to support “what if” scenario testing. These applications are primarily used by middle managers to provide them with information, which is tailored to support semi-structured and unstructured decisions. These applications provide the user with ad-hoc, interactive reporting which is based upon an analytical model that is continually refined in order to solve business problems, e.g. share portfolio management.

**Document management**

**Electronic Data Interchange**

The transmission of business data and documents between organisations or hardware using an OSI protocol.

**Executive Information System**

These systems commonly answer a query by combining information from both internal and external databases. The information is required by top executives to identify problems, opportunities, planning and critical success factor information for the company.

**Financial transaction process & accounting**

**Job, case, incident, project management**

**Logistic or supply planning & control**

**Management Information System**

Provides users with predefined management reports via a reporting system where the user selects the criteria from a limited selection and can usually store the criteria. The report information assists with performance management of a department or business.

**Management or performance reporting**

**Office Information System**

These combine text processing, image processing, telecommunications and other technologies to develop computer-based information systems that collect, process, store and transmit information in the form of electronic office communications, e.g. Word processing, electronic mail, desktop publishing, voice mail, teleconferencing, facsimile and image processing.

**Online analysis and reporting**

**Reservation system (e.g. airline, hotel)**

**Stock control & order processing**

**Trading**

**Transaction/production system**

These systems can be batch or on-line and process business transactions in a logical sequence within a business area. The system consists of a set of inputs to which the transaction/production system adds value and outputs customer or corporate requirements. Examples include: payroll, order entry/processing, general ledger, inventory and case management.

**Workflow support & management**

*Real-Time Applications:*

**Automatic Data Logging**

**Embedded software for simple device control**

**Command & control system (e.g. military, air traffic, police)**

**Complex process control (e.g. oil refinery, steel manufacture)**

Systems that make routine decisions to control operational processes by automatically adjusting physical processes. The software is written for the purpose of controlling, monitoring or manipulating devices such as instruments, e.g. temperature control, manufacturing process control, device or instrument control.

**Fault Tolerance**

A continuously available hardware platform and operating system for critical on-line applications. These systems are able to continue running even when errors are occurring. The foundation of the continuous processing is hardware-based fault tolerance, achieved through:

1. Self checking logic on each major circuit board to detect failures
2. Duplicate boards to continue processing in the event of board failure
3. Hardware components can be upgraded on-line without interruptions

Used for specialised application in telecommunications and retail, travel, banking etc.

**Robot control**

**Telecom & network management**

Software, which monitors and reports on the status of all components of telecommunication networks including communication links and nodes.

**Transportation control (includes avionics, signalling)**

***Mathematically-Intensive Applications:***

**3D modelling or animation**

**Artificial Intelligence**

Used to solve a particular problem by modelling the skill, judgement and expertise of human beings in solving problems in a particular domain. This includes knowledge based systems artificial neural network technology and natural language systems.

**Geographic or spatial information system**

**Image, video or sound processing**

**Mathematical modelling**

**Scientific/ engineering application**

**Statistical analysis**

***Infrastructure Software:***

**Data or database management**

**Device or interface driver**

**Graphics & publishing tools or system**

**Operating system or software utility**

**Personal productivity (e.g. word processor, spreadsheet)**

**Software development tool**

***ARCHITECTURE***

The organisational structure of a system and its implementation guidelines. This derived attribute for the project indicates if the application is Stand alone, Multi-tier, Client server, or Multi-tier with web public interface.

***BUSINESS AREA TYPE***

The business area within the organisation that the application will be supporting.

***CASE (COMPUTER AIDED SOFTWARE ENGINEERING)***

The use of computer software to assist in completing tasks defined within a systems development life cycle methodology.

CASE may be used across the entire project life cycle or used to assist with specific parts of the cycle. The three categories are:



### **Upper CASE**

Environment independent and generally used to perform analysis, e.g. logical data modelling, process modelling, data flow diagramming etc.

### **Lower CASE**

Environment dependent and generally used to assist in physical design and construction of software, e.g. physical data base design, code generation etc.

### **Integrated CASE**

Fully integrating upper and lower CASE, e.g. logical models are converted to physical models, which in turn generate database tables and code.

### **Other**

Any not specified by the above definitions.

### ***CLIENT ROLES***

The roles performed by the computers that provide interface to the software's external users.

### ***CLIENT SERVER***

Indicator of whether the application or product requires more than one computer to operate different components or parts of it.

### ***CLIENT/SERVER DESCRIPTION***

A description of the architecture of the client/server software application or product.

### ***COST***

The price paid, (either through money, time or labour, etc.), to acquire, produce, accomplish or maintain the product. The following methods of collecting COST are believed to be the most common:

#### **Cost Recorded**

The daily recording of all COST incurred by each person on project related tasks.

#### **Cost Derived**

It is possible to derive the COST where it has not been collected on a daily basis as in COST RECORDED.

### ***COUNT APPROACH***

A description of the method used to size the project software. For most projects in the ISBSG repository this is the Functional Size Measurement Method (FSM Method) used to measure the functional size (e.g. IFPUG, MARK II, NESMA, COSMIC, FiSMA etc.). For projects using Other Size Measures (e.g. LOC etc.) this is a short name for that method, and in data releases the size data is not included with sizes measured by an FSM Method but rather is in a section "Size Other than FSM".

Where the Functional Size Measurement Method is IFPUG and the count standard is lower than IFPUG 4 this is indicated by IFPUG (<4), where the count standard is IFPUG 4 or higher this is indicated by IFPUG (4).

#### ***DATA QUALITY RATING***

This field contains an ISBSG rating code of A, B, C or D applied to the project data by the ISBSG quality reviewers to denote the following:

- A** = The data submitted was assessed as being sound with nothing being identified that might affect its integrity
- B** = The submission appears fundamentally sound but there are some factors which could affect the integrity of the submitted data
- C** = Due to significant data not being provided, it was not possible to assess the integrity of the submitted data
- D** = Due to one factor or a combination of factors, little credibility should be given to the submitted data

#### ***DEFECT***

A problem, which if not corrected, could cause an application to either fail or to produce incorrect results. There can be three categories:

##### **Minor Defect**

A minor defect does not make the application unusable in any way, (e.g. a modification is required to a screen field or report).

##### **Major Defect**

A major defect causes part of the application to become unusable.

##### **Extreme Defect**

A failure of some part of an application that causes the application to become totally unusable.

The following information has been collected in relation to application defects both within the project duration (defects and hours per project activity), and after implementation (defects found within the first month of use of the software). Of the following only Defect Found and Repair & Rework Hours per project activity are currently collected:

##### **Defect Found**

The number of defects detected in the process in that particular Effort Breakdown or found within the first month of use of the software after implementation.

##### **Defect Originating**

The number of defects put into the process in that particular Effort Breakdown only.

##### **Defect Removed**

The number of defects removed from the process in that particular Effort Breakdown.

**Repair Hours**

The effort in hours taken to correct defects detected in that particular Effort Breakdown.

**Rework Hours**

The effort in hours taken in that particular Effort Breakdown after correction of defects, to return the project to point reached before defect detection.

***DEGREE OF CONFIDENCE***

An expression of the confidence the organisation has in the data provided, expressed in a range 1 - 4:

1 = not confident

2 = slightly confident

3 = confident

4 = very confident

***DEGREE OF CUSTOMISATION***

How much customisation was involved, if the project was based on a packaged software customisation.

***DEVELOPMENT METHODOLOGY***

Methodologies used during development. For ISBSG purposes a development methodology applies to the whole project development process. This is distinct from development techniques, which apply to activities within the development process.

***DEVELOPMENT PLATFORM***

Defines the primary software development platform, (as determined by the operating system used). Each project is classified as: PC, Mid Range, Main Frame or Multi platform.

***DEVELOPMENT TECHNIQUE***

Techniques used during development. For ISBSG purposes a development technique applies to individual activities within the development process. This is distinct from development methodologies, which apply to the whole project development process.

***DEVELOPMENT TYPE***

**New Development**

Full analysis of the application area is performed, followed by the complete development life cycle, (planning or feasibility, analysis, design, construction and implementation). Examples are:

1. A project that delivers new function to the business or client. The project addresses an area of business, (or provides a new utility), which has not been addressed before.
2. Total replacement of an existing system with inclusion of new functionality.

### **Enhancement**

Changes made to an existing application where new functionality has been added, or existing functionality has been changed or deleted. This would include adding a module to an existing application, irrespective of whether any of the existing functionality is changed or deleted.

### **Re-development**

The re-development of an existing application. The functional requirements of the application are known and will require minimum or no change. Re-development may involve a change to either the hardware or software platform. Automated tools may be used to generate the application.

This includes a project to re-structure or re-engineer an application to improve efficiency on the same hardware or software platform. For re-development, normally only technical analysis is required.

### **Other**

Any other type of software development not specified above.

## ***FUNCTIONAL SIZE***

A size of the software derived by quantifying the Functional User Requirements (i.e., what functions the software must support). This excludes Quality and Technical Requirements. This may be reported in different units depending on the Functional Size Measurement Method (e.g. UFP for IFPUG and NESMA, CFP for COSMIC etc.).

## ***FUNCTIONAL SIZE MEASUREMENT (FSM)***

The process of measuring functional size. Internationally recognised Functional Sizing Methods include: IFPUG, MARK II, NESMA, COSMIC etc.

## ***FUNCTIONAL SIZE UNIT (FSU)***

The unit of measure of size used by a functional size measurement method. (e.g. Function Points)

## ***FUNCTIONAL SIZING TECHNIQUE***

The technology used to support the functional sizing process. Certain technologies used in function point counting can impact on the count's potential accuracy.

## ***IMPLEMENTATION DATE***

The date when the work product is delivered, (i.e. the project software is placed in production, or the project deliverable is delivered). N.B. this is not necessarily the same as Project End Date.

If the project had multiple implementations, this is the date of the first or principal implementation.

### ***INTENDED MARKET***

This describes the relationship between the project's customer, end users and development team.

### ***LANGUAGE TYPE***

Defines the language type used for the project: e.g. 3GL, 4GL, Application Generator etc.

### ***LIFE CYCLE ACTIVITIES***

Used in the context of the time at which functional sizing is carried out, (see also PROJECT EFFORT BREAKDOWN)

#### **Early life cycle**

Up to the completion of the system requirements definition.

#### **Mid life cycle**

From requirements definition to completion of the technical design.

#### **Late life cycle**

From technical design specification until after implementation.

### ***MAXIMUM TEAM SIZE***

The maximum number of people during each component of the work breakdown who are simultaneously assigned to work full-time on the project for at least one elapsed month.

### ***METHODOLOGY USED***

Whether a development methodology was used by the development team to build the software.

### ***METHODOLOGY ACQUISITION***

Describes whether the development methodology (if used) was purchased or developed in-house, or a combination of these.

### ***NORMALISED WORK EFFORT***

For projects covering less than a full software development life-cycle, this value is an estimate of the full development life-cycle effort. For projects covering the full development life-cycle, and projects where development life-cycle coverage is not known, this value is the same as Summary Work Effort.

### ***ORGANISATION TYPE***

A standard classification for the business within which the organisation as a whole operates.

*Note: The organisation is that for which the project has been developed.*

### ***PACKAGED SOFTWARE CUSTOMISATION***

Where a decision is made to acquire an existing product to provide the major component of the required functionality. *Note: Count only the functionality required by the client.*

**PRIMARY PROGRAMMING LANGUAGE**

The primary language used for the software development: JAVA, C++, PL/1, Natural, Cobol etc.

**PRODUCTIVITY**

The ratio of work product to work effort. In ISBSG documents and products this is given by Project Delivery Rate (see entry in Metrics section).

**PROJECT**

A collection of work tasks with a time frame and a work product to be delivered. In ISBSG documents and products the work product delivered is software and its documentation.

**Project Start (Project Start Date)**

A client/management decision is made formally/informally to involve Information Technology personnel in the development. This point is commonly known as the commencement of the 'Survey', 'Feasibility Study' or 'Project Initiation' of the system development life cycle.

**Project End (Project End Date)**

The date when all work effort on the project ceased, and any remaining tasks (e.g. user support) was transferred to another organisation. In a full project life-cycle this would be the end date of implementation activities.

**PROJECT ACTIVITY SCOPE**

Synonymous with the Project Effort Breakdown.

**PROJECT EFFORT BREAKDOWN**

Project effort is sub-divided in two ways: by project activity and by role of the groups of people involved. For the breakdown by person role see WORK EFFORT BREAKDOWN. See also LIFE CYCLE ACTIVITIES. Project management is assumed included in the effort of each of the following activities. The following table shows the major components of each activity.

Project Activity	Possible Activity Components
Plan	Preliminary Investigations Overall Project Planning Feasibility Study Cost Benefit Study Project Initiation Report Terms of Reference
Specify	Systems Analysis Requirements Specification Review & Rework Requirements Spec. Architecture Design/Specification Review & Rework Architecture Spec
Design	Functional / External Design

	Create Physical / Internal Design(s) Review and Rework Design(s)
Build	Package Selection Construct Code & Program Software Review or Inspect & Rework Code Package customisation / interfaces Unit Test Integrate Software
Test	Plan System or Performance Testing System Testing Performance Testing Create & Run Automated Tests Acceptance Testing
Implement	Prepare Releases for Delivery Install Software Releases for Users Prepare User Documentation Prepare & Deliver User Training Provide User Support

The following table is provided as a guide for those organisations that use the ISO 12207 standard.

	<b>ISO 12207 Project Steps</b> (Steps in ISO 12207 – Software Engineering Lifecycle Processes)	<b>ISBSG Activity</b>
1	Requirements Elicitation	Specify
2	System Requirements Analysis	Specify
3	System Architecture Design	Specify
4	Software Requirements Analysis	Specify
5	Software Design	Design
6	Software Construct (Code & Unit Test)	Build
7	Software Integration	Build
8	Software Testing	Test
9	System Integration	Test
10	System Testing	Test
11	Software Installation	Implement
12	User Support	Implement

***PROJECT ELAPSED TIME***

The calendar period in months between the project start and end including any period of inactivity, (i.e. project end date minus project start date).

***PROJECT ID***

A primary key, for identifying projects, (These identification numbers have been ‘randomised’ to remove any chance of identifying a company).

***PROJECT INACTIVE TIME***

Total time, (rounded to whole months), during the project elapsed time, in which no project activity took place. This time, subtracted from Project Elapsed Time, derives the actual time spent working on the project.

***PROJECT LIFE CYCLE***

A collection of project activities which may be sequential, and whose name and number are determined by the control needs of the organisation or organisations involved in the project. A life cycle can be documented with a methodology. (See also PROJECT EFFORT BREAKDOWN).

***PROJECT WORK EFFORT***

All personnel effort that is directed towards the completion of a particular project including out-of-hours effort, whether paid or unpaid. It includes the effort of Client representatives in addition to that of Information Technology personnel.

A good test as to whether an activity constitutes PROJECT WORK EFFORT is to ask the question:

"Would the activity be undertaken if there was no project?"

It excludes NON-PROJECT ACTIVITIES, these being:

- Public Holidays
- Annual Leave
- Sick Leave
- Training (Non-Project).

It is measured in whole HOURS.

***RATING***

See entries for Data Quality Rating and Unadjusted Function Point Rating.

***REUSABLE / CUSTOM CRITERIA***

**Reusable Software**

This is any piece of software designed to be (potentially) re-usable as a component of other pieces of software. Examples: an object-class, a ‘service’ in a Service-Oriented Architecture, a common module of a program.



### **Custom (Application or infrastructure Software)**

Non-reusable software. A piece of software that is initially developed or acquired and implemented by a single project for a single purpose. It is not intended for re-use or any other purpose in the sponsoring organisation.

### **SOFTWARE SIZE**

In general, synonymous with FUNCTIONAL SIZE but the units of software size can be other than that derived by a functional size measurement method, e.g. Lines of Code.

### **SERVER ROLES**

The services provided by the host/server computer(s) to the software application or product.

### **STAGED DEVELOPMENT**

A decision was made during project planning to develop and implement the application as discrete functional units. This may apply to any development type, but must be pre-planned. When an application is developed in total, but implemented over a period of time at a number of locations it is not a Staged Development.

*Note: Where a stage of a Staged Development changes any functionality delivered in a previous stage the project should be defined and treated as an "Enhancement".*

#### **1st Stage**

The 1st Stage comprises a high level analysis of the overall application, (hence defining the scope of each of the stages and possibly some overall design), and full software development cycle of the 1st Stage.

#### **Subsequent Stage**

Subsequent stages of the software development will concentrate on detailed analysis and implementation of another logical part of the overall application.

### **SUMMARY WORK EFFORT**

See PROJECT WORK EFFORT.

### **TARGET PLATFORM**

Categorises the implementation platform, to describe the target environment - Determined primarily by the device the software is implemented into. A project may be classified as:

- Device Embedded (DE)
- PC
- Mid Range
- Main Frame
- Multi platform

## ***TASK***

A project related task or work task is the smallest element of the work breakdown structure, and may be grouped into project activities.

## ***TIME RECORDING METHODS***

The following methods of collecting WORK EFFORT are believed to be the most common:

### **Staff Hours (Recorded)**

The daily recording of all of the WORK EFFORT expended by each person on Project related tasks. As an example, where a person who works on a specific project from 8am until 5pm with a 1 hour lunch break will record 8 hours of WORK EFFORT.

### **Staff Hours (Derived)**

It is possible to derive the WORK EFFORT where it has not been collected on a daily basis as in Method-A. It may have only been recorded in weeks, months or years.

### **"Productive" Time Only (Recorded)**

The daily recording of only the "productive" effort, (including overtime), expended by a person on project related tasks. Using the same example as used in Staff Hours (recorded) above, when the "non-productive" tasks have been removed (coffee, liaise with other teams, administration, read magazine, etc.), only 5.5 hours may be recorded.

## ***TYPE OF SERVER***

A description of the server to the software application or product.

## ***UNADJUSTED FUNCTION POINT RATING***

This field contains an ISBSG rating code of A, B, C or D applied to the Functional Size (Unadjusted Function Point count) data by the ISBSG quality reviewers to denote the following:

- A** = The unadjusted function point count was assessed as being sound with nothing being identified that might affect its integrity
- B** = The unadjusted function point count appears sound, but integrity cannot be assured as a single figure was provided
- C** = Due to unadjusted function point or count breakdown data not being provided, it was not possible to provide the unadjusted function point data
- D** = Due to one factor or a combination of factors, little credibility should be given to the unadjusted function point data

## ***UNRECORDED EFFORT***

Where a breakdown of effort by activity is provided, and the sum of that breakdown does not equal the Summary Work Effort, the difference is the Unrecorded Effort. Where no breakdown is provided, this is the same value as the Summary Work Effort.

### **USER BASE**

Data collected about the extent of usage of the system produced by the project. The following classifications are used:

#### ***User Base – Business Units***

Number of business units (or project business stakeholders) serviced by the software application.

#### ***User Base – Concurrent Users***

Number of users using the system concurrently.

#### ***User Base - Locations***

Number of physical locations being serviced/supported by the installed software application.

### **VALUE ADJUSTMENT FACTOR (VAF)**

The adjustment to the functional size, that takes into account various technical and quality characteristics. The VAF is calculated based on an assessment of the 14 general system characteristics (GSCs) for an application, and when multiplied by functional size gives the adjusted size.

### **WEB DEVELOPMENT**

A derived indicator of whether the project data includes any comment that it is a web-development.

### **WORK EFFORT BREAKDOWN <sup>1</sup>**

Data collected about the people whose time is included in the project work effort. See also PROJECT EFFORT BREAKDOWN. Three levels are identified in the project data collection package. For example if Level 2 is specified, this means that the data submitted includes the development team and the development team support personnel effort. For the process of collecting and reporting Project Work Effort the following classifications are used:

#### ***Level 1 - Development Team***

Those responsible for the delivery of the application under development. The team or organisation, which specifies, designs and/or builds the software. It typically also performs testing and implementation activities. It comprises:

Project Team

Project Management

Project Administration

Any member of IT Operations specifically allocated to the project

---

<sup>1</sup> It should be noted that this Glossary of Terms reflects the data collection package introduced in 2002. The previous data collection package had four levels of work effort.

***Level 2 - Development Team Support / IT Operations***

Those who operate the IT systems that support the end-users and are responsible for providing specialist services to the Development Team, (but not allocated to that team). Support comprises:

Data Base Administration

Data Administration

Quality Assurance

Data Security

Standards Support

Audit & Control

Technical Support

Software Support

Hardware Support

Information Centre Support

***Level 3 – Customers / End Users***

Those responsible for defining the requirements of the applications and sponsoring/championing the development of the application. Also the software's end users. The relationship between the project customer and the software's end users can vary, as can their involvement in a software project. It comprises:

Application Clients

Application Users

User Liaison

User Training

## Part B: Metrics

### **DEFECT DENSITY**

Measures the quality of software in terms of defects delivered in unit size of software. It is defined as the number of Defects per 1000 Functional Size Units of delivered software, in the first month of use of the software. It is expressed as *Defects per 1000 Functional Size Units*.

### **MANPOWER DELIVERY RATE**

Measures the speed achieved by the project team in delivering a quantity of software over a period of time. It is defined as the Functional Size of the delivered software (measured in functional size units), over the Project Elapsed Time (measured in months) multiplied by the number of people in the project team. It is expressed as *Functional Size Units per person per elapsed month*.

The ISBSG previously called this metric speed of delivery (qv). In comparing projects, speed of delivery will vary according to team size. In order to compare like with like the ISBSG normalize speed of delivery with the team size to compare projects by manpower delivery rate.

### **PROJECT DELIVERY RATE, (PDR).**

Measures the rate at which a project delivers software functionality to the end user as a factor of the effort required to do so. In ISBSG documents and products it is defined as Project Work Effort, (measured in hours), over Functional Size of the delivered software, (measured in functional size units). It is expressed as *Hours per Functional Size Unit*.

Project delivery rate is used regardless of how the software is produced as it may:

- ❖ comprise all new software
- ❖ be a modification of existing software
- ❖ use packaged software in part or as the total solution

### **SPEED OF DELIVERY**

Measures the speed achieved in delivering a quantity of software over a period of time. It is defined as the Functional Size of the delivered software (measured in functional size units), over the Project Elapsed Time (measured in months). It is expressed as *Functional Size Units per elapsed month*.