International Software Benchmarking Standards Group

GLOSSARY OF TERMS

for

SOFTWARE PROJECT DEVELOPMENT AND ENHANCEMENT

Table of Contents

Introduction

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 TY PE

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.

NON-FUNCTIONAL SIZE

A size of the non-functional aspects of the product developed or enhanced by the project.

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.

	ISO 12207 Project Steps (Steps in ISO 12207 – Software Engineering Lifecycle Processes)	ISBSG Activity
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 E FFORT

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 FIUNCTION 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 [1]

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

---

[1] 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.*

International Software Benchmarking Standards Group

GLOSSARY OF TERMS

for

APPLICATION SOFTWARE MAINTENANCE AND SUPPORT

Glossary of Terms M&S V2.5.doc (15/07/2010)

Table of Contents

Introduction

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 data into the ISBSG Repositories

❖ standardising the way the collected data is analysed and reported

What follows is a consolidated list of ISBSG definitions and terms used in the Maintenance and Support repository. 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.

Note: where the UKSMA Defect Measurement Standard is quoted please refer to www.uksma.co.uk

PURPOSE OF DOCUMENT

This document provides definitions of terms used in ISBSG documents relating to Maintenance & Support. This includes Application Data collection forms, publications and data releases. Please note that the current Application Data collection forms do not collect all of the data for which the terms are defined here.

The Glossary is divided into two parts:
> Part A: Terms
> Part B: Metrics

ADAPTIVE MAINTENANCE

The modification of software application, performed after delivery, to keep a software application useable in a changed or changing environment. Note: adaptive maintenance provides enhancements necessary to accommodate changes in the environment in which a software application must operate. The changes are those that must be made to keep pace with the changing environment. For example, the operating system may be upgraded and some changes may be made to accommodate the new operating system. Note: for the purposes of the ISBSG M&S standard the adaptive maintenance to be included is that often called minor enhancements (see entry in Glossary) and are taken to require less than five staff days of effort.

APPLICATION

A software application is any tool that functions and is operated by means of a computer with the purpose of supporting or improving the software user's work. It is computer software that employs a computer to perform a task

APPLICATION DESCRIPTION

A general description of the application, to give some idea for research analysis.

APPLICATION EFFORT

The effort devoted to the application on various maintenance and support activities. This can be a total maintenance and total support, or a total to include all of these activities (See separate entries for Maintenance effort, Support effort, or Total M&S effort). A more complete breakdown may be

available. The unit of all effort is hours.

The activities within maintenance and support effort include the following classifications (and a separate entry can be found for each in this Glossary):

Perfective Maintenance
Preventative Maintenance
Corrective Maintenance
Adaptive Maintenance
Problem Investigation
Queries & Quick Service
User help and Advice
Management tasks

APPLICATION SIZE

An objective measure of the size of the software maintained.

Functional Size: This measure is derived by quantifying the Functional User Requirements (i.e., what functions the software must support). This excludes Quality and Technical Requirements. Units of functional size are Function Points (See list in separate entry for Sizing Methods qv.).

Code Size: The application size should also be measured as technical size, a measure of Logical Lines of Code. Units of technical size are thousands of lines of code. Code instructions are defined as: All program instructions created by project personnel and processed into machine code by some combination of pre-processors, compilers, and assemblers. It excludes comment and unmodified utility software. It includes Job control statements, format statements and data declarations (see Software Engineering Economics, Barry Boehm).

APPLICATION TYPE

This identifies the type of application being addressed by the maintenance and support activity, and how the application is meeting the business area requirements. Classification of an application as a type is according to its primary intended use.

3D modelling or automation
Catalogue or register of things or events
Customer billing
Customer relationship management
Device or interface driver
Document management
Electronic Data Interchange
Financial transaction process & accounting
Geographic or spatial information system
Graphics & publishing tools or system
Image, video or sound processing
Imbedded software for machine control
Job, case, incident, project management
Logistic or supply planning & control
Management or performance reporting
Mathematical modelling (finance or engineering)
Network Management
Online analysis and reporting

Operating system or software utility
Personal productivity (e.g. word processor, spreadsheet)
Process Control
Software development tool
Stock control & order processing
Trading
Workflow support & management

BENCHMARK PERIOD

The period in which maintenance and support measures submitted to the ISBSG have been collected for the application. Note that this is not the period over which the application has been maintained by the organisation. The recommendation is that a submission should contain data for a period of 12 months, though a shorter period can be accepted. Data for alternative periods will be scaled appropriately to 12 months before analysis.

BUSINESS AREA

The business area within the organisation that the application supports.

CASE TOOLS (COMPUTER AIDED SOFTWARE ENGINEERING)

The use of computer software to assist in completing tasks of maintaining application source code.

CHANGE REQUESTS

A count of the number of software changes requested of the application, that are undertaken by development as well as maintenance staff during the maintenance period. Counts of software changes are classified as: Changes Requested and Changes Accepted.

CONFIGURATION MANAGEMENT

Configuration Management is used for application version control.

CORRECTIVE MAINTENANCE

The reactive modification of a software application performed after delivery to correct discovered problems. Note: the modification repairs the software application to satisfy requirements.

COSMETIC DEFECT

A defect that may be or is related to the layout or presentation of data but which results in no corruption of data and no wrong values. For example labels, headings or Colours are missing or poorly chosen. Whilst this may be regarded as a software failure, the software is completely useable. There is no real impact on the customer-facing operations; customers are not aware of any degradation in services. However, users may feel annoyed and frustrated with the software and may suffer inefficiency (see also UKSMA Defect Measurement Standard).

Note: for the purposes of this ISBSG M&S standard cosmetic defects are combined with minor incidents to form a single count.

CRITICAL APPLICATION

Indication of whether the application is considered mission-critical to your business. This question identifies the importance of the application for the business.

CUSTOMER SATISFACTION

A customer satisfaction survey is a Questionnaire containing several questions on satisfaction with the application and maintenance of the application. This indicates if such a survey has occurred, and refers to the organisation's practice for this application.

DATA BASE SIZE

The size of database maintained by the application in Megabytes.

DATA QUALITY RATING

This field contains an ISBSG rating code of A, B, C or D applied to the 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 detected defect is a software fault delivered into the operational installation (see also UKSMA Defect Measurement Standard).

DOCUMENTATION AVAILABLE

A list of the documents available to the maintenance team and would include such items as functional specifications; technical specifications; logical design; physical design; operations manuals; and so on.

ELECTRONIC DATA INTERCHANGE

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

EMBEDDED

An embedded platform enables the delivery of functionality within larger mechanical or electrical systems, often with real-time computing constraints. Embedded systems encompass: portable devices such as digital watches, and MP3 players, to large, stationary installations such as traffic lights, factory controllers, and complex systems like hybrid vehicles, MRI and avionics.

END USER BASE

Data collected about the extent of usage of the application. The following classifications are used:

Distinct Installations

Number of distinct installations for which the maintenance and support effort is required. A 'distinct installation' is an individual installation of the complete software system.

User Locations

Number of user locations is a simple count of the number of geographically separate locations for which support is provided.

Distinct End Users

Number of distinct end users is those having access to the system, regardless of whether they use the system regularly or occasionally.

Concurrent Users

Number of concurrent end users is the number of users on-line at the same time (applies to single distinct installation). This data is held as a maximum and average number.

EXECUTION FREQUENCY

This is the operational regime of the software application. The expectation is that a high performance and execution frequency will have higher maintenance costs.

EXTREME DEFECT

A very severe defect that may render or has rendered the system inoperable - that is some business function is now no longer possible - manual procedures if they exist may have to be brought into operation. Examples could be: the software will not run or stops during operation; there is seriously wrong or corrupted data; the incident seriously impacts upon customer-facing operations and customers are aware of the incident (see also UKSMA Defect Measurement Standard).

FTE

FTE (Full Time Equivalent) is the number of hours worked by a full time member of maintenance

and support staff in a complete year.

HELP DESK CALLS

A count of the number of calls to the Help Desk function for this application made during the maintenance period. These may be direct calls, emails or issues raised via a query logging system.

INCIDENT

An incident occurs when the production is disturbed, (and if a helpdesk exists this is announced to the helpdesk). A defect of the software can lead to more than one incident. Sometimes a work-around is available to continue operation after an incident. There can be three categories:

Minor Incident

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

Major Incident

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

Extreme Incident

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

ISBSG ID

A primary key, for identifying submissions to the ISBSG repository, (These identification numbers have been 'randomised' to remove any chance of identifying a submitter).

INTERFACE AGREEMENT

Documented agreement on an interface between two or more systems. This physical document defines the conditions and consequences of the interface.

INTERVENTION

Intervention is a term used here as a generic name for the number of times Perfective, Preventive, Corrective, Adaptive problem investigations, Queries and Quick Services or User Help and Advise are asked for. For Corrective Maintenance it should be the number of defects in application software, in Preventive Maintenance it is the number of problems, for Queries and Quick Services the number of Queries and Quick Services asked for and so on.

MAINTENANCE

The activities and tasks required to keep a system operational after its implementation into Production. Maintenance tasks are categorized as follows:

- ❖ adaptive maintenance
- ❖ perfective maintenance
- ❖ preventative maintenance
- ❖ corrective maintenance

MAINTENANCE EFFORT

The number of hours devoted to maintenance work in the maintenance period.

MAJOR DEFECT

A severe defect that may seriously or has seriously degraded but not disabled some business function, amounting to failure. The business operation can continue at a lower rate of activity or only a portion of a business function is disabled. For example: customer details can still be entered but credit information cannot be confirmed; the customer facing operation is impacted, but this may not be evident to the customer (see also UKSMA Defect Measurement Standard).

MANAGEMENT TASKS

This is the time that cannot be directly attributed to the maintenance and support of an application (eg. Administration or Personnel management).

MINOR ENHANCEMENT

This can comprise any one or more of the following:

- ❖ Redesign and development of small portions of an existing software application
- ❖ Design and development of small interfacing software items, which require some redesign of the software application
- ❖ Modifications of the software application code, documentation or database structure

Note: for the purposes of the ISBSG data repository, minor enhancements in this ISBSG M&S standard are taken to require less than five (5) staff days of effort. They comprise Adaptive and/or Perfective Maintenance (qv).

MINOR DEFECT

A defect that may or has resulted in low-key disruption to business operations, causing for example, user inefficiency. The software suffers a failure, but it is still operable. For example: data values may be wrong or corrupted in a way that is tolerable for a limited period, or some minor aspect of functionality is unavailable. Business operations can continue with little degradation and any degradation is not evident to the businesses customers (see also UKSMA Defect Measurement Standard).

Note: for the purposes of this ISBSG M&S standard, cosmetic defects are combined with minor defects to form a single count.

MOBILE

Mobile platforms enable the delivery of functions on mobile devices such as smart phones.

NETWORK MANAGEMENT

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

ORGANISATION TYPE

A standard classification for the business or industry within which the organisation as a whole operates. The type of business in which the organisation is mainly involved (e.g. Banking, Insurance, Manufacturing, Financial services, Service industry and so on).

OTHER LANGUAGE

Other programming languages used after counting the primary and secondary programming languages.

OTHER TOOLS

Besides a test environment and a production environment, these are additional tools available to the M&S personnel for this application.

PERFECTIVE MAINTENANCE

The modification of a software application after delivery to improve performance or maintainability. Note: perfective maintenance provides enhancements (improvements) for users, improvement of program documentation, and re-coding to improve software performance, maintainability, or other software attributes.

PERSONNEL EXPERIENCE

Distribution of experience of M&S personnel on this application. The M&S team experience is a group of data items that indicate the level of knowledge of support personnel. The likely relationship to cost is that the higher the level of experience, the lower will be the cost. Percentages of the team are provided for knowledge in the following categories: Application, Technology, Business Area, and Language.

PLATFORM

The platform on which the bulk of the application resides e.g. Mainframe, Midrange, PC or Client/server.

PLATFORM ROLE

The role or function provided by a hardware platform to an application. In client/server or distributed software applications, one or more computers provide services to software operating on other computers. Platform role examples are: Server, Client, Core System, Warehouse, and Infrastructure.

PREVENTATIVE MAINTENANCE

The modification of a software application after delivery to detect and correct latent faults in the software application before they become effective faults. This might include redesign or expansion of the database in order to accommodate increases in data occurrences. It is recognised that such activities may well be bundled for the purposes of regression, integration and user acceptance testing.

PRIMARY PROGRAMMING LANGUAGE

The primary programming language is that used to program the bulk of the application.

PROBLEM INVESTIGATION

The support activity of problem investigation, which is carried out in order to determine if a reported incident is in fact a defect, or an error in user documentation or training or merely a user error.

PROCESS CONTROL

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.

PRODUCTION START DATE

The date of the first production usage of the application (even if it is the first release). If the application has been replaced, this is the date when the replacement went into production. This date has previously been referred to in ISBSG documents and products as the Launch Date.

QUERIES & QUICK SERVICES

These are one-time questions that are not part of the application and can be delivered by the support team on request of the user.

- ❖ A Query is a one-time extraction from the application data
- ❖ A Quick Service is the provision of a one-time service that involves some manipulation of the data in the database

RATING

See entry for Data Quality Rating.

REPAIR TIME

The average number of days elapsed between defect detection to repair, for the defects recorded for this application.

SECONDARY LANGUAGE

The secondary programming language is that used to program the bulk of the application after the primary programming language.

SEPARATE TEST ENVIRONMENT AVAILABLE

Indicator of whether a separate development environment is available to the maintenance team.

SERVICE LEVEL AGREEMENTS

An indication of whether there are any service level agreements affecting this application.

SIZING METHODS

The sizing methods currently recorded by ISBSG are as follows:

Function Points (including a standard & version number):

- ❖ IFPUG Function Points
- ❖ COSMIC Function Points
- ❖ NESMA Function Points
- ❖ FiSMA Function Points
- ❖ MKII Function Points

Feature Points.

Line of code measured in Logical KSLOC

SOFTWARE DEFECT

A software defect is a departure in a software application from its expected properties (see also UKSMA defect Measurement Standard).

SOFTWARE RELEASES

The number of software releases during the maintenance period.

SOFTWARE SIZE
In general, synonymous with APPLICATION SIZE, qv.

STANDARD OF DOCUMENTATION
Objective assessment of the standard of documentation:

- ❖ Low – does not readily provide information required
- ❖ Average – provides most information required
- ❖ High – readily provides information required without difficulty

SUPPORT
The activities undertaken to address users' problems/queries after a system has been implemented into Production. Support tasks are categorized as follows:

- ❖ user help
- ❖ queries & quick service
- ❖ problem investigation

SUPPORT EFFORT
The number of hours devoted to support work in the maintenance period.

SYSTEM AVAILABILITY
The required availability of the system (e.g. is it required for 7 hours/day 6 days of the week or 24hrs for 7 days of the week).

TEAM SIZE
The number of people (in whole numbers) regardless of their availability who are assigned to maintenance and support of this application.

TEST ENVIRONMENT AVAILABILITY
A percentage of the time that the support team can devote to bug fixing and enhancement activity. If the team has to support a 24 by 7 operation and are provided with a separate development environment in order to develop and bug fix, this is 100% availability. If on the other hand they are dealing with a system whose operational time is from 0800 until 2200 hours, and they must wait for operations to finish before they can undertake testing bug fixing and other activities, the availability is 40% (approximately).

TESTING TOOLS
The testing tools available and used on this application. This captures the extent of tool use in the testing of the application. The use of testing tools should reduce the number of errors in the operational environment and thus reduce the cost of M&S activities.

TOTAL M&S EFFORT
The number of hours devoted to maintenance and support work combined in the maintenance period.

USER HELP and ADVICE
General support that is not related to an incident and does not involve any data extraction or manipulation.

USERS FREQUENCY OF USE
The degree to which users might be self-supporting and addresses the capability in terms of their experience. The users are classified as one of the two types to reflect the bulk of the users of the application: Regular users and Occasional users.

Part B: Metrics

FUNCTIONAL RATE

Measures the maintenance and support rate of the application calculated as Total Effort (for maintenance and support, in hours) normalised to a period of one year, divided by Application Size (in Function Points). It is expressed as *hours per Functional Size Unit.*

KSLOC RATE

Measures the maintenance and support rate of the application calculated as Total Effort (for maintenance and support, in hours) normalised to a period of one year, divided by Application Size (in Logical KSLOC). It is expressed as *hours per KSLOC.*

MAINTENANCE RATE

Measures the maintenance rate of the application calculated as Maintenance Effort (in hours) normalised to a period of one year, divided by Application Size.

Where the application is sized in Function Points, it is expressed as *hours per Functional Size Unit.*

Where the application is sized in Logical KSLOC, it is expressed as *hours per KSLOC.*

SUPPORT RATE

Measures the support rate of the application calculated as Support Effort (in hours) normalised to a period of one year, divided by Application Size.

Where the application is sized in Function Points, it is expressed as *hours per Functional Size Unit.*

Where the application is sized in Logical KSLOC, it is expressed as *hours per KSLOC.*