

# Glossary

## Table of contents

1	WysDoM Glossary.....	2
---	----------------------	---

## 1. WysDoM Glossary

Term	Definitions	Notes
Abstract Data Type	1. A data type for which only the properties of the data and the operations to be performed on the data are specified, without concern for how the data will be represented or how the operations will be implemented. From [IEEE90].	
Attribute	1. A characteristic of an item. From [IEEE90].	
Computer-Aided Software Engineering (CASE)	1. The use of computers to aid in the software engineering process. May include the application of software tools to software design, requirements tracing, code production, testing, document generation, and other software engineering activities. From [IEEE90]	
DateBench(tm)	1. A specification and set of Java applets to test Year 2000 compliance. See Also: <a href="#">WayDate</a>	
Exception	1. An event that causes suspension of normal program execution. From [IEEE90].	
Extensible Markup Language (XML)	1. A subset of the Standard General Markup Language, created to be used for documents on the World Wide Web. The language uses tags to define the structure of the language. XML is standardized by World Wide Web Consortium.	

Forrest	1. An open-source Web-based publishing framework. See <a href="http://forrest.apache.org/">http://forrest.apache.org/</a> .	
Function Point Analysis (FPA)	1. A method of measuring and estimating the effort involved in constructing or maintaining software system. See International Function Point Users Group.	
Functional Programming Language	1. A programming language in which the primary mode of computation is the definition and application of functions.	
Good Enough Software	1. The concept that the level of defects in software is a product characteristic to be traded off with other product characteristics like functionality, availability, and price. Under this concept, the market will accept higher levels of defects if other product characteristics are acceptable. The extension of this concept is that elimination of defects can be and should be sacrificed to improve other product characteristics.	
Invariant	1. An assertion that should always be true for a specified segment or at a specified point of a computer program. From [IEEE90]	
MetaLanguage (ML)	1. A functional programming. See [ULLM98] See Also: <a href="#">Functional Programming Language</a>	
Package	1. A general purpose mechanism for organizing elements into groups. Packages may be nested within other packages. A system may be thought of as a single high-level package, with	

	everything else in the system contained in it.	
Portable Document Format (PDF)	1. A format for storing and displaying documents in a way that retains the layout identical to the way the document would be printed. Developed by Adobe, Inc., PDF is often used as a way of rendering a document from the World Wide Web so that it can be printed.	
Requirements-Design-Implement (RDI)	1. An iteration in the software lifecycle resulting in some or all of a software system.	
Record	1. A set of related data items treated as a unit. From [IEEE90]	
RickBench(tm)	1. A Microsoft OneNote notebook for tracking project risks. See Template page.	
Tuple	1. A list of two or more values or expressions.	
Unified Modeling Language (UML)	1. a language for specifying, visualizing, and constructing the artifacts of software systems, as well as for business modeling. It combines the object-oriented analysis and design conventions from the Booch, Jacobson, and Rumbaugh methods.	
Use Case	1. A typical interaction between a user (or agent) and a computer system. It captures some function visible to the user. The use case achieves a goal for the user.	
WayDate	1. A specification and Java class for date calculations.	

Glossary

	See Also: <a href="#">DateBench(tm)</a>	
Waysys Development Method(tm) (WysDoM)	1. An approach to software requirements combining several modern techniques with formal methods.	