## Glossary

**Actor** – a type of agent that interacts with a product

**Assumption** – a state of affairs taken for granted in product development

**Business Requirement** – a statement of a client or a development organization goal that a product must meet

**Class** – an abstraction of a set of objects with common operations and attributes

**Client** – someone outside a development organization with an interest in a product, such as customer or a user

**Constraint** – any factor that limits developers and particularly one that limits design solutions; in UML, a property that a model element must satisfy or a relationship between model elements that must be maintained

**Customer** - someone who pays for a product

**Database** – a computerized collection of records

**Data requirement** – a statement that certain data must be input to, output form, or stored in a product

**Functional- requirement** – a statement of how a software product must map program inputs to program outputs; also called a behavioral requirement

**High Fidelity (abbr. Hi-Fi)** – a term use to describe a high quality of an entity

**ID** – identification number which uniquely identify different products.

**Implementation** – the creation of executable artifacts for delivery to customers in a software product; the relation between an abstract

**Interface** – a boundary across which two entities communicate.

**Low Fidelity (abbr. Lo-Fi)** – a term use to describe a lower quality of an entity

**Microsoft.NET Framework** – is a software framework developed by Microsoft that runs primarily on Microsoft Windows

**NHibernate** – is an object-relational mapping solution for the Microsoft .NET platform: it provides a framework for mapping an object-oriented domain model to a traditional relational database

**Non-functional requirement** – a statement that a software product must have certain properties; also called a non-behavioral requirement

**Object-oriented programming (abbr. OOP)** – is a programming paradigm using "objects" – usually instances of a class – consisting of data fields and methods together with their interactions – to design applications and computer programs.

**Object-relational mapping (abbr. ORM)** – is a programming technique for converting data between incompatible type systems in object-oriented programming languages

**Product** – Refers to the twoCube system.

**Product vision** – a general description of a product’s purpose and form

**Product scope** – the work to be done in a project

**Respondents** – refers to the people who do the survey

**Responses** – refers to the answers the respondents enter when doing the survey.

**RESTful** – derived from REpresentational State Transfer (REST) which is a style of software architecture for distributed systems such as the World Wide Web; conforming to the REST constraints is referred to as RESTful.

**Service Oriented Architecture** – a set of principles and methodologies for designing and developing software in the form of interoperable services

**Software requirement** – a statement that a software product must have certain feature, function, capability, or property

**Software requirements specification (abbr. SRS)** – a complete description and documentation of the twoCube system. It includes a description of all the interactions the users will have with the software.

**Structured Query Language (abbr. SQL) -** special-purpose programming language designed for managing data

**Survey** – a sampling, or partial collection, of facts, figures, or opinions taken and used to approximate or indicate what a complete collection and analysis might reveal

**Stakeholder** – anyone affected by a product or involved in or influencing its development

**System –** Refers to the twoCube system

**Use case model** – a use case diagram together with use case descriptions for each use case in the diagram

**Use case diagram** – a UML notation representing a prodcut’s use case and actors involved in each use case

**User Interface (abbr. UI)** – refers to the visual component of the system by which the user and a computer system interact

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | |  | |