ICT Project Guidance

Glossary of ICT Specific Terms:   
Development

Author:

Sky Sigal, Solution Architect

Version:

0.3

## Description

A Glossary of common ICT Terms for reference, to establish a common understanding, while reducing duplication of effort in downstream documents.

## Contents

[Description 1](#_Toc146637232)

[Contents 2](#_Toc146637233)

[Introduction 3](#_Toc146637234)

[Objective 3](#_Toc146637235)

[Terms & Acronyms 4](#_Toc146637236)

[System Development Industry Terms & Acronyms 4](#_Toc146637237)

[DRY 4](#_Toc146637238)

[Immutable 4](#_Toc146637239)

[Mutable 4](#_Toc146637240)

[OO 4](#_Toc146637241)

[Object Oriented 4](#_Toc146637242)

[SOC 4](#_Toc146637243)

[Separation of Concerns 4](#_Toc146637244)

[SOLID 4](#_Toc146637245)

[GRASP 4](#_Toc146637246)

[Appendices 6](#_Toc146637247)

[Appendix A - Document Information 6](#_Toc146637248)

[Images 6](#_Toc146637249)

[Tables 6](#_Toc146637250)

[References 6](#_Toc146637251)

[Review Distribution 6](#_Toc146637252)

[Audience 6](#_Toc146637253)

[Diagrams 6](#_Toc146637254)

## Objective

To develop a common understanding of terms used in artefacts used to deliver services with an ICT component.

# Terms & Acronyms

## System Development Industry Terms & Acronyms

#### DRY

: an acronym for a Don’t Repeat Yourself, a key development approach to decrease development effort while increasing analysability and maintainability qualities.

#### Immutable

: non-changing data. Examples include system categorisation lists which remain the same throughout the service’s lifespan. See Mutable.

#### Mutable

: authorised user entered data, that can subsequently be corrected or even logically deleted (data should not be *physically* deleted). Contrast to *Immutable*.

#### OO

* : see *Object Oriented*.

#### Object Oriented

* : a computer programming model that organizes software design around data, or objects, rather than functions and logic. See *SOLID* which provides principles on how to deliver OO, and *DDD*, which outlines how to organises OO code into deployable components and pakages that maximise *maintainability* and *modifiability* while minimising *complexity*.

#### SOC

: see *Separation of Concerns*.

#### Separation of Concerns

: a key development approach to diminish the cost of development, analysis, maintenance and may improve portability and reuse.

#### SOLID

* : a mnemonic acronym for five design principles intended to make object-oriented designs more understandable, flexible, and maintainable:
* Single Responsibility principle (see *Separation of Concerns)*.
* Open-Close Principle (open for extension, closed for modification).
* Liskov’s Substitution Principle (use interfaces rather than concrete classes).
* Interface segregation Principle (use smaller interfaces).
* Dependency Inversion Principle (see *Injection*).
* *Note it may be of interest to know that SOLID only captures the first 5 principles of Uncle Bob’s 10 principles.*

#### GRASP

* : an acronym for General Responsibility Assignment Software Principles. Another well known and valuable set of development patterns to guide development towards delivering long term value:
* controller,
* creator,
* indirection,
* information expert,
* low [coupling](https://en.wikipedia.org/wiki/Coupling_(computer_science)),
* high [cohesion](https://en.wikipedia.org/wiki/Cohesion_(computer_science)),
* [polymorphism](https://en.wikipedia.org/wiki/Polymorphism_(object-oriented_programming)),
* protected variations, and
* pure fabrication

Appendices

Appendix A - Document Information

### Images

### Tables

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Sandy Britain, Enterprise Architect |  |
| Amy Orr, Data Architect |  |
| Roger Govind, Security Architect |  |
| Archana Sahani, Business Analyst |  |
| Dijana Sneath, Business Analsyt |  |
| Vincent Weirdsma, Lead Developer |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (Archimate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.