DRAFT - ICT Project Guidance

Principles - Development

Version:

0.1

Author:

Sky Sigal, Solution Architect

## Description

This document describes the Guiding Principles developed specifically for development stakeholders.

## Synopsis

It is a requirement that Development Stakeholders adhere to guidance provided Organisation and Project Principles, defined elsewhere, common to all project stakeholders, as well as Development Principles, defined here.

## Contents

[Description 1](#_Toc145232973)

[Synopsis 1](#_Toc145232974)

[Contents 2](#_Toc145232975)

[Synopsis 3](#_Toc145232976)

[Introduction 3](#_Toc145232977)

[Heading Level 3 3](#_Toc145232978)

[Heading Level 4 3](#_Toc145232979)

[Appendices 4](#_Toc145232980)

[Appendix A - Document Information 4](#_Toc145232981)

[Images 4](#_Toc145232982)

[Tables 4](#_Toc145232983)

[References 4](#_Toc145232984)

[Review Distribution 4](#_Toc145232985)

[Audience 4](#_Toc145232986)

[Structure 4](#_Toc145232987)

[Diagrams 4](#_Toc145232988)

[Terms 5](#_Toc145232989)

## Introduction

Principles are Agreed Constraints.

## Purpose

Guiding Principles reduce risk and cost.

They contribute to developers having more agency to make decisions without the delay and cost of preparing presentations for govenrence to make decisions.

## Principles versus Patterns

Development Guiding Principles are distinct from Development Patterns. Guding principles guide decisions, whereas Patterns outline known processes to follow to deliver them.

## Principled Design

The Principles that the architect must be guided by and adhere to where possible and appropriate include:

* AoG obligations:
  + NZ Privacy Principles
  + NZ Digital Standards Principles
  + NZ Digital Service Principles
  + NZ Data and Information Management Principles
* This organisations obligations:
  + Architecture Principles
  + Integration Principles

These principles are outlined in an earlier section, as well as the Appendices.

## Principled Development

* **The Project Principles**: the developer must adhere to the following previously listed Principles:
  + Pipeline First
  + Secure First
  + Web Interface API First
  + Multiple Channels First
  + Universal User First
  + Smallest Visual Interface First
  + Succinct, Summarized, Actionable, Forgiving, Views First
  + Forgiving Behaviour First
  + YAGNI – as long as it is not used to defer required work.

The number of principles that a developer must adhere to are fewer:

* **Principled Design First:** Developers must understand that the system designer is constrained to be guided by the above principles and not make changes to designs without consulting the system designer.
* **Automated Test First:** Developers must not develop new functionality before developing the automated tests to demonstrate their correct functioning.

### Test Driven

Developers MUST follow an automated Test-Driven Development process.

The following is an example workflow:

### Loose Coupling, High Cohesion

Where possible, use SOLID development patterns – specifically Liskov's Substitution and Dependency Inversion design principles.

**Note:**In C#, this is implementable as:  
- Using a class' constructor to inject– typed as Interfaces -- any service dependencies needed.

### Real-time Notification over Delayed Batch notification

Where applicable, prefer real-time notification messages between systems of changes, over the use of delayed scheduled updates (i.e., no reliance on after hour “fat” ETL operations).

**Note:**Use Queued Command patterns to process batch or CPU costly operations as close to real-time while limiting the impact on more prosaic transactional operations.

Appendices

Appendix A - Document Information

### Images

[Figure 1: TODO Image 2](#_Toc144995112)

### Tables

[Table 1: TODO Table 3](#_Toc145048484)

[Table 2: TODO Table 2 3](#_Toc145048485)

### References

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

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### Audience

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

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### 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.

### Terms

Refer to the project’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.