IT Project Design

Service Client Design UX Basics

# Document

## **Synopsis**

A Service is costly to develop for no purpose if not made available via a Service Client’s service user interface. A Service client user interface with poor Recognisability, Learnability and Usability qualities is costly to operate, requiring secondary systems and/or organisation changes to compensate for the lack of access.

This document provides information to decrease the above risk by outlining expected user interface structuring of system information, structuring, operations and sequences -- irrespective of per-project customisation via style guidelines, graphics and resources.

## **Description**

Title: Service Client User Interface Risks & Mitigations

Author: Sky Sigal

Date: 31/03/2022

## **Distribution**

Distribution/Review List:

* {name}, {role}, {org}

## **Conventions**

### **Terms**

The Appendices include a Glossary of Terms to assist understand this document.

### **Diagrams**

Where applicable, diagrams are developed using ISO-\*, Archimate, UML or appropriate industry standards and conventions.

# Context

## Background

A Service’s capabilities that provides an interface with poor Recognisability, Learnability, Usability qualities is more costly to provide training, operate, use effectively and is used by less users.

This document outlines key Information Presentation Principles and the project delivery risks they mitigate.

### Principle Based Project Delivery

Project Delivery risk is diminished by increasing both delivery speed and acceptance while decreasing costs, including those of coordinating governance. This is achievable when deliverers are given observed agency to proceed independently towards common objectives, within pre-agreed constraints. Coordination of effort is maintained by working to plan for what is determinable upfront, and to a framework of agreed Guiding Principles for what is not.

Note: a common misconception around the use of guiding principles is that for effective governance reasons, projects should limit the overall number of Guiding Principles. This is only partly correct. It is correct that the project have a small set of project principles. But in turn each subgroup (deployers, stylers, interface deliverers, service developers, testers, etc.) *should also have their own* limited number of guiding principles as long as they are not in conflict with any other group’s guiding principles. Effectively each group is working to a set of principles working within the overall project’s larger framework of guiding principles.

### User Interface Styling Principles versus Information Presentation Principles

*User interface styling principles* and guidelines, graphics and text media are distinct per project.

In contrast, *information presentation principles* describe how best to structure interfaces, in order to operate, sequence and present information. These principles -- which are the generally the same, irrespective of the project or system -- are mature, proven, documented and readily available. The overwhelming majority of IT projects do not require attempting to seek alternative information presentation principles and guidelines.

To decrease delivery risk, projects should limit investigation and customisation discussions, time and cost to developing guidelines and principles for the superficial styling and effects.

A key project lifespan risk is the provision of a Service Client User Interface design which does not make available the information contained within the system.

Even when information is available within, an additional risk exists when information is not made available in a coherent, organised manner, such that it can be easily navigated. The impact of difficult to find, navigate, use information is a high cost of developing additional training material, time and expense to train, low uptake of the service by users who are not funded to train to use the system.

### Organic Design Approach

A lack of forethought and clarity as to a system’s Purpose, the Information required within the system to perform that purpose, and its Structuring leads to an adhoc foraging for stakeholder feedback on attempts to understand their needs, which results in an organic, unstructured and chaotic development of the user interface.

The impact on end users is diminished its understandability, discoverability, usability qualities. This in turn increases requires training material, training time. This in turn increase project costs, and diminishes natural uptake of the service.

There is a place for agile responsivity to user feedback on user interface design – but should be limited to what is not determinable by forethought.

# Information

## Domains

Systems manage two distinct yet related sets / domains of information:

* Business Domain
* Technology Domain

Irrespective of the technology, shape, size, or capabiliites of a service’s interface, it's key purpose is to make available to users the following key groups of information:

* Context
* State
* Resource

## Context

Context generally includes providing the end user access to understanding:

* + service (ie: app/service name, eg: “HR Service”),
  + tenant/organisation (ie: organisation name, eg: “IBM”)
  + Service group (ie: department, eg: “Sales”, “Refunds”, “Servicing”)
  + Service consumer (ie, the current Session User, eg: “Joe Bloe”)
  + Resource Discovery Parameters (ie, Search or Naviation)
  + Resource Context (ie, the current location on the web – the URL)
  + Resource Action Context (ie List Browse & Item(s) Select & Item Read, Add, Edit, Delete)

## State

State generally includes one or more of the following:

* State of the Service (eg: upcoming downtime, functionality availability, etc.)
* State of the Tenant, Department (eg: “Department will be closing for Christmas”).
* State of Operations (eg: “Changes Saved”, “Changes not Saved”, ”Save Failed”)
* Navigation State (eg: where are you in the sysem/web -- ie, the URL)

## Resource

The primary reason a user uses a service is to access Resources.

Resources can be divided into either Business Services Resources – the reason the service was automated – and common Technical Service Resources.

### Business Service Resources

Business Service Resources will be distinct service to service, but at a high level can be divided into the following:

* People
* Actions
* Events
* Relationships

### Technical Service Resources

Whereas Business Service Resource will change per project, a certain number of resources are common to all systems, whether they are immediately built or deferred:

* Sessions
* Invitations
* Identities
* [System] Users
* Names
* Settings
* System Duties
* Permissions
* Role Responsibilities
* Role Responsibility
* Roles
* Groups

### Item State

All Resources have State which requires making evident as it in turn affects system behaviour:

* Start & End Dates (eg: Not yet started, Current, OnHold, Expired)

Business Resources may have additional state:

* Financial (eg: Free, Trial, Current, Owing, OnHold)