**AS IS TO BE**

As Is Process Maps Will help to capture an accurate visual picture of your organization’s processes and visualize process breakdowns and areas of risk

To Be Process Maps Will Help to: Model the impact of any future process changes before you make them.

It's not difficult to understand the necessity for thoroughly analyzing your As Is and To Be processes. when you put together any kind of business activity model for the future, you first have to think about where you are today. Only then can you fully grasp where you need to go, and what you need to do in order to get there.

To undertake an As Is to Be analysis, one need to know what each one means separately, and how your As Is and To Be work together as part of an overall business activity model.

Defining the As Is Process

The As Is process defines the present state of a business process in any given organization or structure. The goal of analyzing the current state of a process is to find out which components could be improved. It is important to bear in mind that this As Is analysis will only show you what can be improved, but not necessarily how. Mapping the As Is process will only reveal how your business processes are working today.

Defining the To Be Process

Typically, a To Be process is defined as the future state of an organizational or business process. It is the ideal state of how you want your business processes to work and mapping the To Be processes will structurally clarify how you can get there.

Only then will you be able to see what changes are necessary in order for your business processes to reach their future To Be state.

When embarking on your As Is To Be analysis, it is important to remember that business process analysis does not exist in isolation. Instead of analyzing one process, it is crucial to document and map all other related processes as well. Only then you can get the full picture. For reviewing, documenting and mapping your organization’s As Is and To Be processes, you need to get your Quality Management team on board, who should not only create high-level process maps, but also seek the conversation with employees who are responsible for certain processes. This way, more detailed process maps can be created. Keeping your staff in the loop comes with the beneficial side-effect that working towards shared goals increases motivation in the workplace - meaning that things will start getting into motion.

**PARETO**

The Pareto principle, or the 80/20 when referring to Agile, basically means that 20 percent of the work effort on the project is necessary where 80 percent is futile, or a waste of time. however, takes the 80/20 rule a step further by utilizing the sprint system effectively. When utilizing Agile in any project, sprints are tasks completed by teams in a determined amount of time, usually anywhere from one to four weeks. At the end of the sprint, the task should not only be completed, it should be considered done and ready for the client or stakeholder.

Why is this principle so important in Agile project management? Essentially, not only does rule PARETO mean the iteration result of the sprint is complete, it also means not moving on to another task until the current one is done, finished, and acceptable.

**INVEST**

INVEST is a simple guide to write meaningful User stories. In general, User stories are supposed to have certain characteristic described by Bill wake as INVEST.

I – Independent – (stories should be as far as possible independent so each of them could be developed and delivered separately.

N – Negotiable – (User Stories should discussable further and there should be space of negotiation)

V – Valuable – (User Stories should result in adding value to the customer)

E – Estimable – (User Stories should be understandable enough so could be divided into the task and could get estimated)

S – Small – (User Stories should not be too big, usually should be done in 40 hours of work)

T – Testable – (User Stories, usually have acceptance criteria to test if they fulfill customer’s needs)

**UNION & UNION ALL**

UNION

The UNION command combines the result set of two or more SELECT statements (only distinct values)

The following SQL statement returns the cities (only distinct values) from both the "Customers" and the "Suppliers" table:

Example

SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;

## UNION ALL

The UNION ALL command combines the result set of two or more SELECT statements (allows duplicate values).

The following SQL statement returns the cities (duplicate values also) from both the "Customers" and the "Suppliers" table:

Example

SELECT City FROM Customers  
UNION ALL  
SELECT City FROM Suppliers  
ORDER BY City