**Microservices: A Business perspectives**

Define component software design and API economy

As has been shown by a number of organizations over the last few years, the ability for orgainisations to deliver new IT functions and features with speed to support their service operations wins in the marketplace. Netflix, eBay, Amazon, PayPal, Gilt, Buemix, to mention just a few, have proven this. These organisations have been successful due to their adoption of microservice architecture. Essentially, microservice architecture is a method of developing software applications as a suite of independently deployable, small, modular services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism to serve a business goal. By adopting a mictoservice architecture it gives orgaisations the ability to deploy independent

What

API economy

Operations need to own the services and drive their development. Define and manage the scope

In today's world of IT, the speed with which new features are released wins the marketplace. The speed of transforming an idea (or feedback from a customer) from the inception phase to getting it out to customers is very critical. Traditional ways of the SDLC in developing software hinders this speed. There is a lot of friction from product development. In this article, we will take a look at how building a microservice helps us remove the friction from product development and enable the speed and availability of new features.

Essentially, microservice architecture is a method of developing software applications as a suite of independently deployable, small, modular services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism to serve a business goal.

As Martin Fowler [points out](http://martinfowler.com/articles/microservices.html), Netflix, eBay, Amazon, the UK Government Digital Service, realestate.com.au, Forward, Twitter, PayPal, Gilt, Bluemix, Soundcloud, The Guardian, and many other large-scale websites and applications have all evolved from monolithic to microservices architecture.

First, software built as microservices can, by definition, be broken down into multiple component services.  Why?  So that each of these services can be deployed, tweaked, and then redeployed independently without compromising the integrity of an application.  As a result, you might only need to change one or more distinct services instead of having to redeploy entire applications.  But this approach does have its downsides, including expensive remote calls (instead of in-process calls), coarser-grained remote APIs, and increased complexity when redistributing responsibilities between components.

The rapid proliferation of APIs is forcing companies to rethink their approach on how they build or embed new functionalities into their products. APIs have opened up a much faster and a better way to build features and enhance the overall user experience. For example, if you want to embed maps into your application, you can easily embed Google Maps by using their public API without having to reinvent the wheel. This drastically cuts development costs.

Why business owners should understand and manage

How business can manage using system procedures

API management

Service design

If you are a providing IT products as part of your services to your customers (internal and external), you may be able to significantly bring down the cost of those offerings by taking the API approach. For example, you can build your own public APIs that allow other 3rd party apps to communicate with your products or enable your products to interface with APIs developed by other companies so as to be able to exchange data or embed features without having to build the functionalities from scratch.

Thus when creating a blueprint as part of a Service Design activity, web services and API’s should form a key part of this exercise.

With the evolving move towards component design in services and microservices, which has proven successful in delivering solution quickly, there is a need for business operations to take control. The move to microservices has been driven by the IT community and the design of such solutions are still within this remit. However, there is a need, particularly in an enterprise environment for the business to start driving the solution for these services.

Business owners of solutions normally look at a solution as forming part of a set of processes or as a set of system work instructions. If we look at it from a work instruction perspective. A work instruction is a piece of work carried out on one element within a system. For a set of business processes there could be a number of work instructions covering one or more systems.

Each work instruction should be linked to a microservice. So work instruction should be divided by microservice because any change to that microservice could have an impact on any of the related work instructions. Likewise any change to that microservice would require retesting all the work instructions within that microservice.

The advantage of dividing the work instructions into microservices is that the business starts thinking service

es rather than processes. What are the set of services that the enterprise application provides. When looking at improvements

If you look at any discussion on component design or microservices it does high Dev Ops, but in this sense the authors are nearly always talking about operations from an IT perspective not from a business perspective. If, as is the case in a lot of enterprise situations, that the application is supporting business owners provding a service to a customer or a wider internal business community. Changes

We should be able to map one Docker container to a set of Work Instructions. For any microservice the business should be able to see what effect any change to a microservice has. This should be limited, otherwise.

As you understand your microservices and API’s, this should give you the leverage to rquest adding an off the shelf API to a current microservice or adding this API as a new microservice in your landscape

In summary, it is important for the business woner of enterprise applications to understand and drive the development of component