To: Professor Krasso

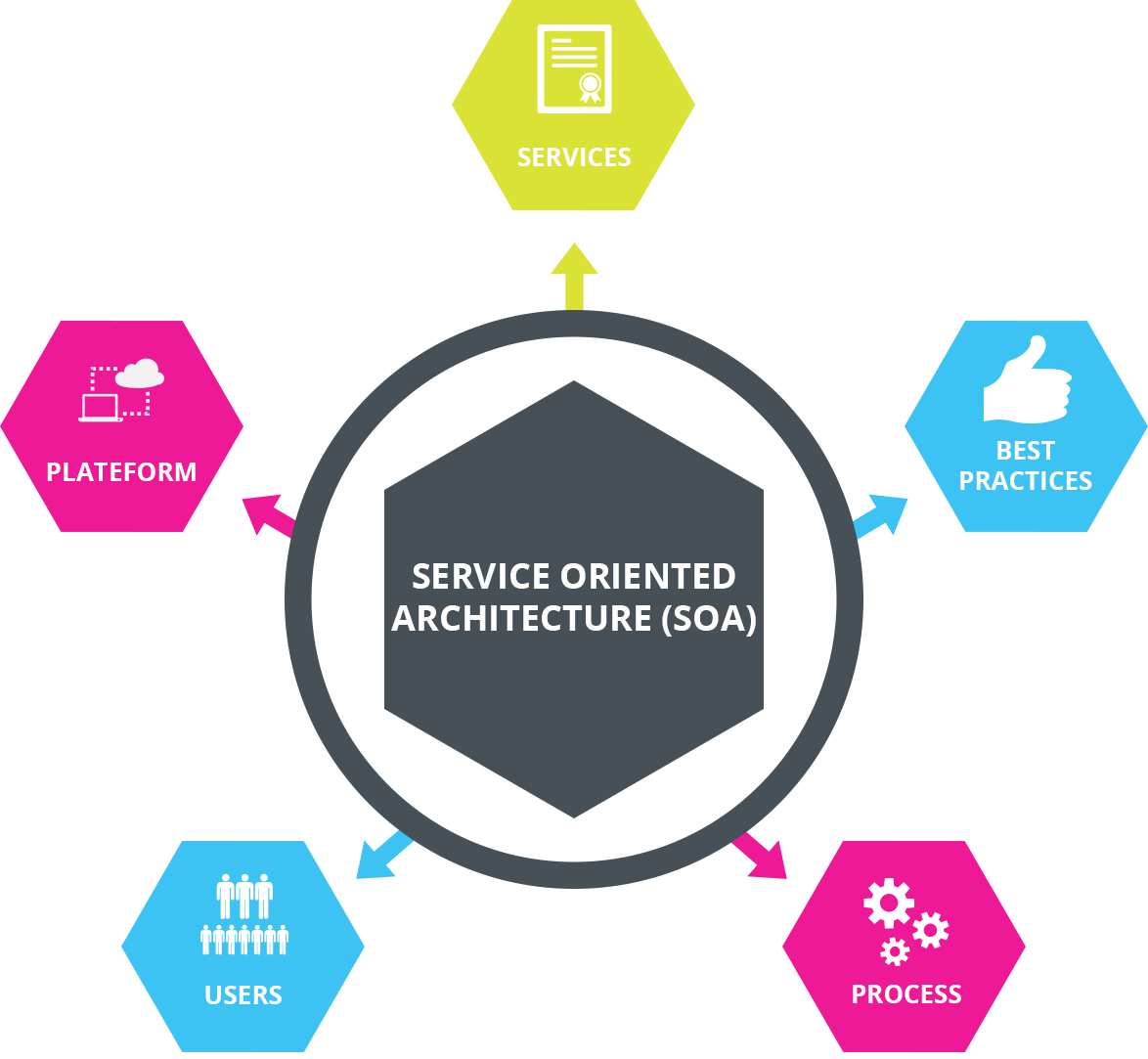
From: Faye Van Roekel

Date: December 1, 2019

Subject: From Legacy to SOA

Legacy applications are software programs that are obsolete. Typically, they have compatibility issues with browsers, operating systems and infrastructures which in turn can make them unstable. Many times they create complications for IT staff that try to support them.

Service-Oriented Architecture or SOA is a style of software design where services are provided to the other components by application components, through a communication protocol over a network (*Software Development Company, 2019*). The different areas or components of service-oriented architecture are:



* + - * Business value
* Strategic Goals
* Intrinsic Inter-operability
* Shared services
* Flexibility
* Evolutionary Refinement

Software Development Company. (2019). *Service Oriented Architecture*. Retrieved from https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec

The purpose of our transitioning from a legacy architecture to a service-oriented architecture (SOA) is to provide “loose coupling” of the systems from each other. That simply means that we will be making the network such that we can modify and update each system individually without worrying about it impacting the other systems. The SOA provides communications between the systems through an Enterprise Service Bus (ESB). As you can see from the diagram, all messages from the client to the systems and between the systems goes through the ESB. This means the systems do not need to speak the same language as each other in order to communicate.

We want to begin the transition with the internal systems we use in operating the database itself. We will start the transition with the GitHub system. That is the repository we use for our code. Transitioning it first will then in turn make transitioning the other systems easier. This will be used as a feasibility test of the migration process. If this process succeeds, we next want to transition to the Active Directory System, which is how the database for our network operates. Our plan is to complete these two transitions over a period of six months each in order to have adequate testing to see if it works.

After taking data and measuring the success of the first two transitions, and also assuming it makes sense for us to do so, we want to transition the systems that relate specifically to employees first. We will transition payroll, recruiting, and other human resource employee systems. We transition these employee systems before transitioning the other customer-based systems since they are our employees and because they do not involve our relationships with our customers. By doing this that makes a reduced risk to the company if something goes wrong. Finally, we will transition the other systems – budget, procurement and contract systems, and the service desk system.

References:

Dennis Smith, Grace Lewis. (2007). Migrating Legacy Assets to Service Oriented Architecture (SOA): A DOD Example (presentation). Retrieved from <https://apps.dtic.mil/dtic/tr/fulltext/u2/1014748.pdf>

Martin. (2015, Apr 20). How to Build A Service Oriented Architecture (SOA) (article). Retrieved from <https://www.cleverism.com/how-to-build-service-oriented-architecture-soa/>

Software Development Company. (2019). *Service Oriented Architecture*. Retrieved from <https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec>

Rouse, M. (2005, September 16). What is legacy application? - Definition from WhatIs.com. Retrieved December 1, 2019, from <https://searchitoperations.techtarget.com/definition/legacy-application>