To: Teresa Butterfield

From: Adam Donner

Date: 14 August 2019

Subject: From Legacy to SOA

As an organization, we have several business applications that management has expressed interest in integrating into one standard interface. There are many departments that require logging into several systems to perform daily activities. Ideally, this would all be done in a single application. A perfect example of this is HR having to log into three different systems, the employee system, the payroll system, and the recruiting system, to do daily activities. This is a cumbersome workflow and results in loss of productivity.

As a development team, we recommend using Service-Oriented Architecture (SOA) to modernize our application infrastructure. Service-Oriented Architecture is a software development model for distributed application components that incorporates discovery, access control, data mapping, and security features. The two primary functions or SOA is to create a broad architectural model that defines the goals of applications and the approaches that will help meet those goals. The other function is to define specific implementation specifications, usually linked to the formal Web Services Description Language and Simple Object Access Protocol specifications. Included in the development will also be an Enterprise Service Bus (ESB). An ESB is a type of software platform known as middleware, which works behind the scenes to add application-to-application communication. We will include the ESB because it allows legacy applications to be integrated into a standardized fashion.

Figure 1.1 is the proposed SOA implementation structure. Starting at the top, you will see End-User Application and Portals. These will be the graphical user interfaces in which our users interact with the system. The SOA infrastructure is proposed to be our Enterprise Service Bus, which will allow communication between the current applications to take place. This communication will take place without any end-user knowledge. All of the existing systems are listed in the blue circles. The ESB will allow these to communicate with each other. Finally, you will see the Legacy or New Service Code. In an ideal world, we will be able to adapt our existing systems into the SOA Infrastructure with little or no change to the current code. This will help alleviate the development time it will take to roll out the new system.

We plan on using the Service Migration and Reuse Technique (SMART) to analyze the viability of reusing legacy components. Figure 1.2 shows the process that we will be using. The first step is to Establish Migration Context. This step allows us to understand the business and technical context for migration, including programmatic constraints such as schedule and budget. The next step is to determine if migration of the legacy system is feasible. We may run into situations with our existing systems that will require significant programmatic change to the code or a total rewrite. At this point, we will reconvene about the specific system to discuss with management if a complete rewrite is acceptable or if continuing to use the legacy system is acceptable. If it is determined to be feasible, we will select a small number of services and will fully specify inputs and outputs. The next step will be to describe the existing capability of the system. This will include gathering data such as quality, existing problems, and user satisfaction. We will then describe the target SOA environment. This will identify the impact of specific technologies, standards, and guidelines for service implementation. We will then analyze the gap to define effort, risk, and cost to convert legacy components into services. Finally, we will develop the migration strategy. This will include the order in which to create the services, guidelines for the creation of the services, specific migration paths to follow, as well as needs for training and technology evaluation.

I feel that moving to a Service-Oriented Architecture is the correct path for our organization. Although this will require some development time and training, it will be mitigated by the increase in productivity by our users. This includes user training and access to systems after the SOA is deployed.

Sincerely,

Adam D. Donner

References

Carnegie Mellon. (2007). Migrating Legacy Assets to Service Oriented Architecture (SOA): A DoD Example. Retrieved August 14, 2019, from <https://apps.dtic.mil/dtic/tr/fulltext/u2/1014748.pdf>

What is an ESB? (2017, July 18). Retrieved August 14, 2019, from https://www.it.ucla.edu/news/what-esb

What is service-oriented architecture (SOA)? - Definition from WhatIs.com. (n.d.). Retrieved August 14, 2019, from https://searchmicroservices.techtarget.com/definition/service-oriented-architecture-SOA