



SOA DEVELOPMENT ORGANIZATION

SOA Blueprint

A structured blog by Yogish Pai

Introduction

Best practices recommend develop the organization model by defining the target state, developing the roadmap, and identifying components that need to be built and how. Governance and organization end up being a natural outcome of this exercise.

An alternate approach is to build the organization model based on the tasks and not the components that need to be built. This section briefly describes some of the organization models that enable faster adoption of SOA.

Traditional Development Approach

Most IT projects fit into one of these categories:

- **e-business solutions:** portal applications for both internal and external users
- **Packaged applications:** best-of-breed point solutions
- **Integration:** integration of applications, portals and data across the enterprise or LOB
- **Infrastructure:** data center, networks, servers, and software platforms.

The business sponsors and IT leadership team prioritize projects and monitor progress periodically.

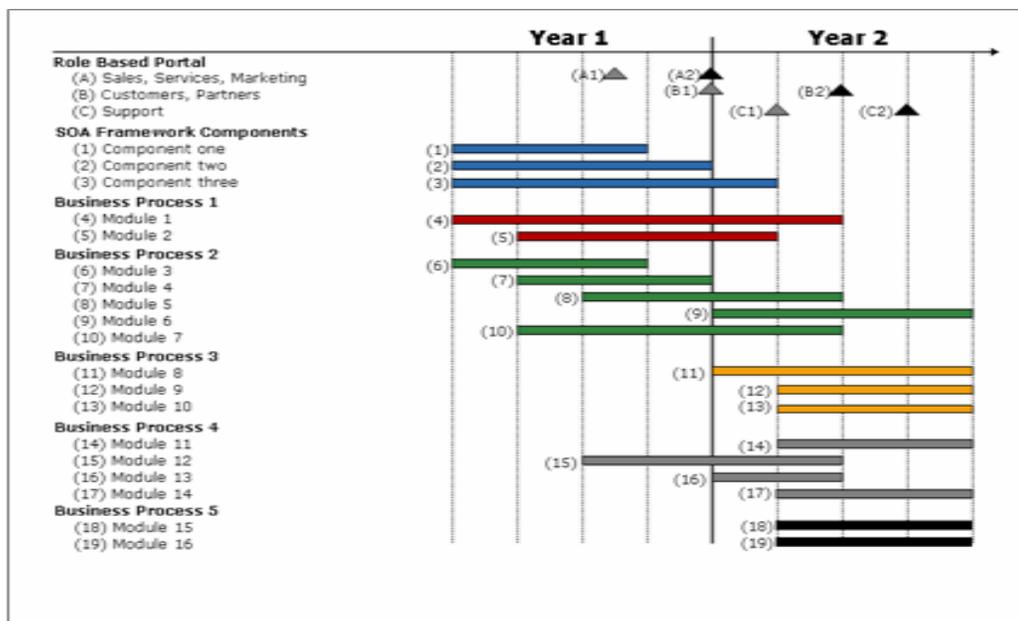


Figure 1: Typical IT Roadmap

The traditional development lifecycle applies consistently across all the initiatives as shown below, but resource allocation varies based on the initiative category. The following diagram illustrates a typical development lifecycle.



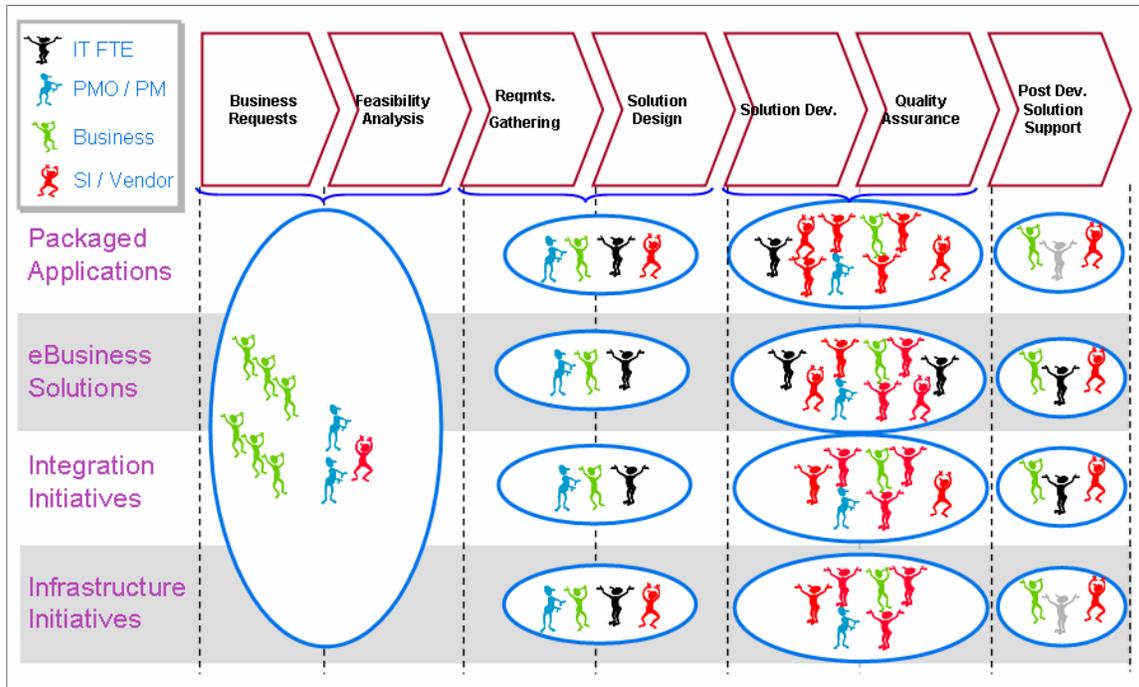


Figure 2: Typical IT Development Lifecycle

Typically a team consisting of business operations, PMO, project manager, and the IT staff of business analysts and architects review the high-level business requirements of each initiative. Based on their analysis, the IT staff provides the joint leadership team with a recommended approach and estimates for funding approval. Even though the lifecycle is the same, the approach is different for each type of initiative.

- **Packaged applications:** the typical approach is to outsource development to a system integrator or the product vendor and later off-shore support of these packaged applications to reduce cost. This is considered a best practice because packaged applications are proprietary and developing these skill sets in-house as that wouldn't provide a strategic value to the company. While enterprises could standardize on one packaged application vendor and build an in-house team, this is not a practical solution for large corporations. The corporation would be too much exposed to the business issues of its vendor. Besides, most LOBs want packaged solutions specific to their own needs.
- **e-business solutions:** most enterprises look at this as a strategic investment, because this is one of the fastest-growing channels for customers, partners, suppliers, and employees to interact and collaborate with each other. The preferred approach is to dedicate the company's best architects and developers to creating and supporting these solutions, and to build these skills sets within the organization. Developing in-house expertise makes sense, especially for consumer-focused organizations, because it facilitates making rapid changes to the customer-facing solution to help the organization keep its competitive edge. For some lower-profile business functions, support could potentially be outsourced or off-shored.
- **Integration:** integrating e-business solutions with packaged applications and later with the platform for BPM is complex. Execution depends on the quality of interaction between the various teams, so outsourcing is not the most appropriate approach. Instead, the industry best practice is

to embed all the customization into the middleware. The LOB should stick to providing the business processes and rules and resolving open issues related to business definitions. Enterprise architecture teams should make the integration decisions.

- Infrastructure:** the best approach to infrastructure varies depending on the size of the organization. Organizations should keep this in-house until they grow to over \$1 billion in revenue. Larger enterprises may out-source infrastructure to a third party, which then makes it easier to estimate the real cost of infrastructure. Keeping infrastructure in-house could overload IT operations staff, resulting in longer turn-around time and reduced efficiency.

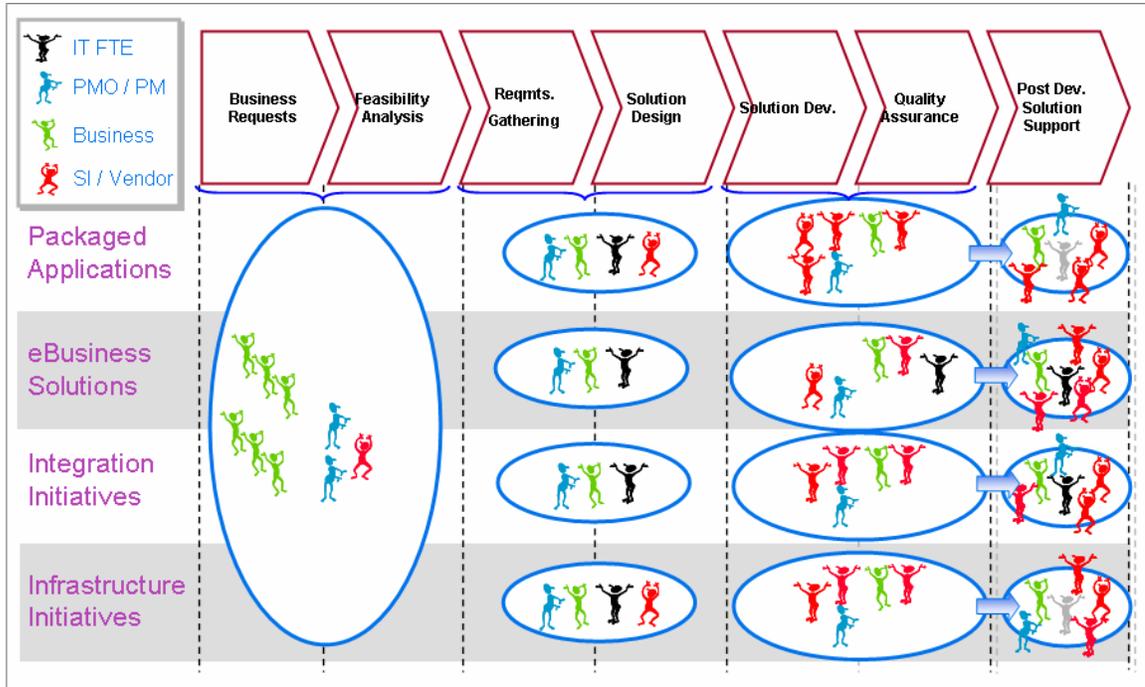


Figure 3: Result of the current Lifecycle approach

One of the major drawbacks of the traditional approach is that as IT delivers new capabilities, resources get diverted to support these capabilities, increasing the cost of the capability. IT may need to bring in additional resources to develop new services or provide support for existing services. IT organizations cannot halt this trend but can slow down the cost of new development by outsourcing and later off-shoring the support function. However, this approach can only take the IT organization so far. Enterprises need to develop a different approach for staffing their development organizations.

Recommended Approach

The recommended approach is to create teams based on technical capabilities, as follows:

- Composition team:** wires the services together based on business processes and requirements. The wiring of services not only identifies the capabilities required by each service but also the deployment, configuration, and management model. This team should consist of architects and technical leads.
- UI team:** develops the business interaction layer which consists of the wire frames, user interaction flows, navigations, and type validations. Development of this layer could be outsourced, provided there an excellent process in place to capture and document the requirements.

- **Services team:** designs and develops the business logic. The recommendation would be to staff the key positions in-house, especially the architects and key developers. The development itself could be outsourced as long as it is based on the architecture guidelines and managed and reviewed by internal staff.
- **Data team:** develops the shared or specific data services for the other teams. Information and data architects are the key members of the team and should be company employees. They define the enterprise data model, data quality, and common objects such as customers, orders, and products. The development itself could be done anywhere but has to be based on the architecture guidelines and managed and reviewed by the information and data architects. While the other teams receive services from this team, they do not need to understand the enterprise data model.

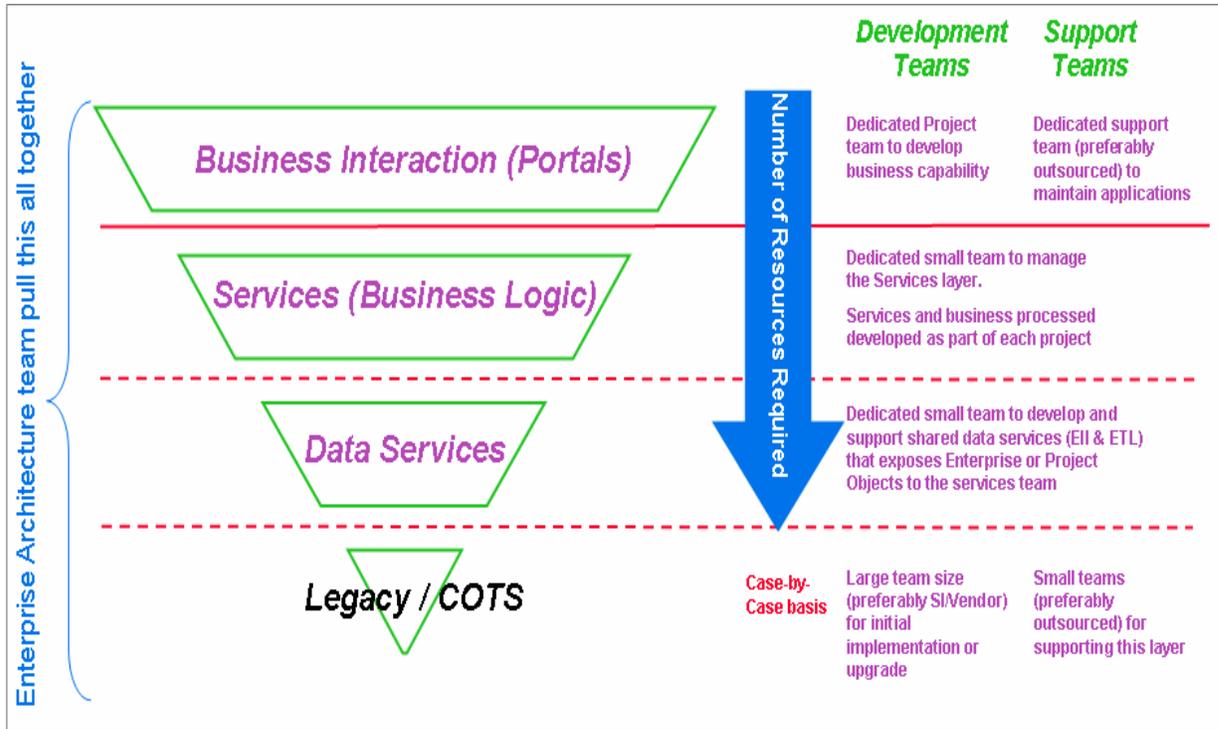


Figure 4: Organization based on capabilities

Summary

This recommended approach organizes development teams based on their capabilities. It results in lower costs because it eliminates the need to staff every project team with similar skill sets.

Copyright

Copyright © 2006.

The author grants a non-exclusive licence to everyone to publish this document in full or part by acknowledging the source. Any other usage is prohibited without the express permission of the authors.