

A Business-Oriented Foundation for Service Orientation

Ulrich Homann
Microsoft Corporation

February 2006

Applies to:
Enterprise Architecture
Service-Oriented Architecture (SOA)

Summary: Automate business activities effectively and efficiently by using service-orientation with Web services.

Contents

[Introduction](#)

[Enter Business Capability Mapping](#)

[Business Capabilities - A More Stable Foundation](#)

[Capability Model Overview](#)

[Conclusion](#)

"It is not the strongest of the species that survive, or the most intelligent, but the ones most responsive to change."

Charles Darwin

Introduction

Service-orientation, with the advent of Web services, is the latest technology solution put forward to automate business activities. (For a good overview of SOA and related concepts in Microsoft's connected systems strategy, please refer to [Service Orientation and Its Role in Your Connected Systems Strategy](#).)

The concepts embodied by service-orientation evolved out of long-standing efforts to modularize and distribute complex computer systems that reflect and support the reality of the distributed business world.

According to the current definitions of service-orientation, services offer network accessible functionality through standard, published, and discoverable interfaces. These core characteristics promise at least protocol (syntactic) interoperability regardless of platform, provider, and location.

Copyright © 2006 Microsoft

These days, however, solving technical integration problems is not enough by itself to justify investing in service-orientation. To strengthen the argument, various claims are being made about service-orientation support for business agility because, the argument goes, in a service-oriented environment solutions are easier to build by composing services, re-configuring them, re-using them, and so forth. *Maybe*, but by the above definition of service-orientation, many businesses have, to some extent, implemented service-orientation and yet still struggle to provide the IT support that the business needs or desires.

To justify an investment in service-orientation, therefore, we need to address the following issues that come up in every project employing new architecture principles:

- How do we prevent service-orientation from following similar promising initiatives into the same architectural mistakes of the past?
- How do we ensure that the chosen implementation architecture relates to the business requirements?
- How do we maximize the life expectancy of the implementation in an ever-changing environment?

As we will see, the answers to these questions are interrelated. Service-orientation merely embodies a particular perspective of the system; it is not the starting point for the system, and a flawed basis will get you an imperfect implementation. This foundation is the important missing piece of the implementation puzzle—a way to derive said business requirements and relate them to service-oriented constructs in a formalized, repeatable, and innovative way.

There is generally no well-defined relationship between the methods used to describe or model business requirements and their eventual technical implementation. Today, most efforts to capture and represent business requirements use business process modeling techniques and technologies. While the formal modeling of business processes is a necessary part of any attempt to relate business requirements and strategy to IT strategy and implementation, it is not sufficient.

Business processes need to focus on the activities required to achieve the goals of the business regardless of traditional "brick-and-mortar" boundaries—something you do internally today might be outsourced tomorrow. Our designs must support these collaborative requirements with flexible business models, taking into account the boundaries, effects of networks, changing responsibilities, and so forth, while at the same time enabling smooth crossing of these boundaries regardless of systems, corporate, or other boundaries. Traditional business process modeling does not align business requirements with technology structures and investments. There is no conception of boundaries, deployment issues, encapsulation, and so forth.

In other words, a better model must, in addition to the representation of activities, support both the interdependence of the businesses and the autonomy of services that support it.

Enter Business Capability Mapping

To deliver against these objectives, we propose conceiving the business as a network of capabilities. A capability models *what* a business function does—its externally visible behavior (versus *how* it does it, its internal behavior)—and the expected level of performance. Pay Employees and Ship Product are examples of capabilities. These capabilities have "owners" and "customers" in the generic sense and they are

expected to perform at a certain level (such as units per period of time, or some quality measurement). This is "what" is done and the expected, or contracted, level it is done to. Inside the capability are the specifics of how that capability is implemented at a given point in time with respect to people, procedural steps, technology, and so forth. At this level of abstraction we focus on the external aspects, and how they are achieved is not important. The capability is essentially a "black box."

Capabilities are sufficiently descriptive to understand how a function fits in the business and what is required to interact with it, yet summarize enough to provide the stability required to provide a firm, longer-lasting base. The network aspect describes how capabilities interconnect to perform the work required by the business.

Already, with the emphasis on *external, observable, and measurable behavior*, and the *connections* with other capabilities, we can immediately see the parallels between capabilities and service-orientation. In their most abstract senses, business capabilities and Web services are both black boxes whose connections are important, related to but separate from their inner workings. Intuitively, this parallelism bodes well for marrying the two.

Shifting Boundaries

Before we jump into more details on business capabilities, it is important to have a thorough understanding of the problems of satisfying business requirements with IT. IT is both the cause and effect of architecture improvements and reoccurring projects. After all, the business expects a clear return on investment (ROI) and value strategy before supporting our efforts to embrace new technology.

The IT infrastructure for most businesses is a complex, interconnected system grown organically over time to meet various and continuously changing needs. Each new technology drives architectural changes that promise, given enough money, that they will be able to redesign the existing systems into an organized, flexible whole, composed of component parts, which aligns business needs, systems, and data flows. However, even if you could do a complete architectural redesign, with all data shared and no duplicate applications, over time the infrastructure would gradually devolve back to an organic structure similar to the original. Why and how does that happen? What are the issues involved here?

Connected Businesses

Pushed by globalization and competitive forces, the classic, linear supply chain has evolved into a complex value network of partners participating in a common business. Figures 1 and 2 demonstrate the evolution from a linear value chain to a complex value chain network. These networks are expanding to include additional services provided by an increasing number of partners: customers, the government, financial services organizations, and so forth. Investments in systems or applications need to take into account the requirements or opportunities enabled by this increasing interconnectedness.

Value Chain Evolution - Businesses as Networks of Connected Partners

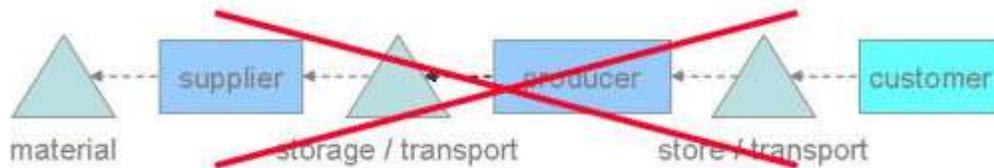


Figure 1. "Then"—*traditional* supply-chain

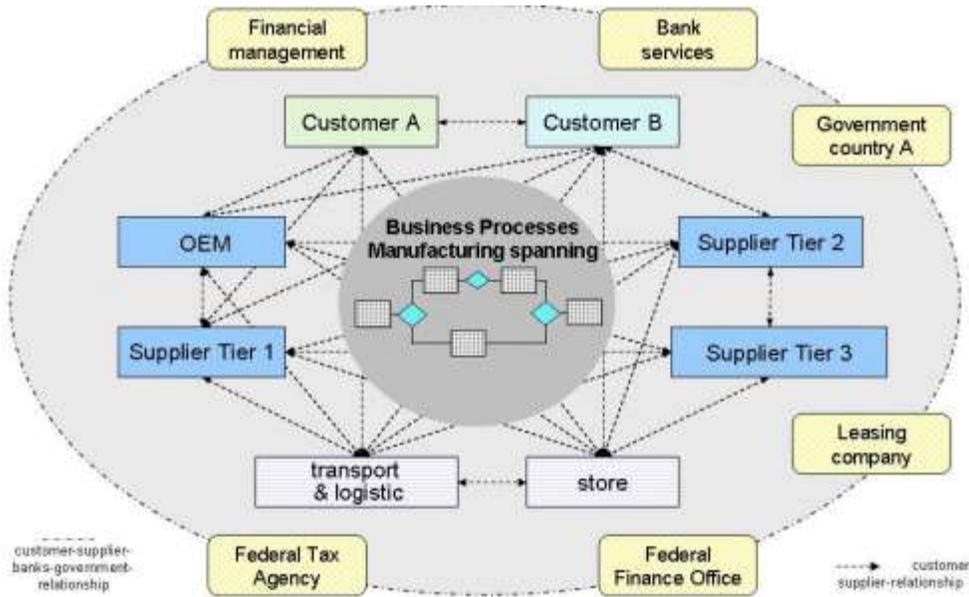


Figure 2. "Now"—*networks of cooperation*

As we think about companies, their customers, the government, and the evolution of how they work together to achieve their business aims, three main models emerge that complement each other to deliver against business goals:

- The traditional *value chain* partners focus on procuring materials, transforming them into intermediate or finished products, and distributing these finished products to customers.
- *Business process outsourcers (BPO)* augment the value chain by providing services for business requirements outside of material transformation or product distribution. Outsourcers for HR services like payroll have been available for a long time, but the offerings are becoming increasingly complex and focused on more business-critical activities. For instance, better communications technology has opened low-cost offshore labor markets for provisioning services such as software development and Help calls.
- In addition, *self-service* is becoming an important part of collaboration. Self-service ranges from customers or partners demanding more flexible interactions with suppliers, to suppliers offering incentives to customers and partners to interact in different ways. Even governments are adding to the trend of self-service by offering or requiring specific online activities such as completing forms or financial reports.

In other words, modern businesses cross traditional corporate boundaries in multiple dimensions, and it only makes sense that our modeling and solutions account for this. The corporate boundary is becoming analogous to the fiscal year for a company when it comes to planning. Both are important and relevant boundaries, but businesses have to plan and budget beyond these boundaries.

Systems Evolution - Connected Environment of Business Functions

If you examine the systems of any given company and look at what they do, you generally will find that the systems or applications support specific functionality or user communities. Finance systems for the finance community, customer relationship management (CRM) for the marketing, sales, and support communities, and so on. If they are connected at all, the connections are not very elegant. The result is that customers live with disconnected silos of functionality. The unpleasant fact for them is that in all likelihood it will stay that way, as new ways of doing business emerge faster than integration can occur.

Since we can never "catch up," much less get ahead of change, applications need to follow a different path in the future as innovation continues to drive businesses and customers in various ways. Instead of launching a monumental exercise of folding business capabilities into the existing solution, we should accept the fact that despite all efforts at an ultimate architecture, reality and experience speaks against such a possibility. By the nature of generally volatile business behavior influenced by rapidly changing technology, any architecture will be in flux to some degree. However, knowing that, we can look at our architecture and build it in a way to accommodate those changes as long as possible, turning architectural investments into enduring assets.

What Can We Do About It?

These observations about extending the life of architectural redesigns are fine, but the astute reader will notice that they really rely on getting the right answer to our second question: How does the implementation architecture relate to the actual business? How well we answer this question determines how well we can implement these ideas.

Existing business improvement techniques focusing on process improvement have made a lot of progress in meeting today's business challenges. However, focusing first on a view of "how" work is performed makes assumptions about "what" is performed. Solutions are then limited to improving the mechanics—not challenging the fundamental premise of the work. This is the convention most people use in trying to solve business problems today. To improve on this, we need to challenge these assumptions and ask different questions.

Business Capabilities - A More Stable Foundation

So, we find that the real question is "When you architect a solution for your customers, which elements of that architecture are really durable and allow you to deal with change?" Because that answer obviously provides the best ROI and the best insurance against architectural obsolescence.

What we have found is that the stable elements are much more about what businesses actually do (for example, create purchase orders, generate invoices, ship products, and so on). These business activities, which we refer to as business capabilities, are relatively stable, but how businesses implement them with people, procedures, and technology, and how those activities knit together into processes, is far less stable. So now, we need to find out what a business does, what its capabilities are.

Before we continue, let's introduce a few definitions for our discussion:

- A *business capability* is a particular ability or capacity that a business may possess or exchange to achieve a specific purpose or outcome. A capability describes what the business does (outcomes and service levels) that creates value for customers; for example, pay employee or ship product. A business capability abstracts and encapsulates the people, process/procedures, technology, and information into the essential building blocks needed to facilitate performance improvement and redesign analysis.

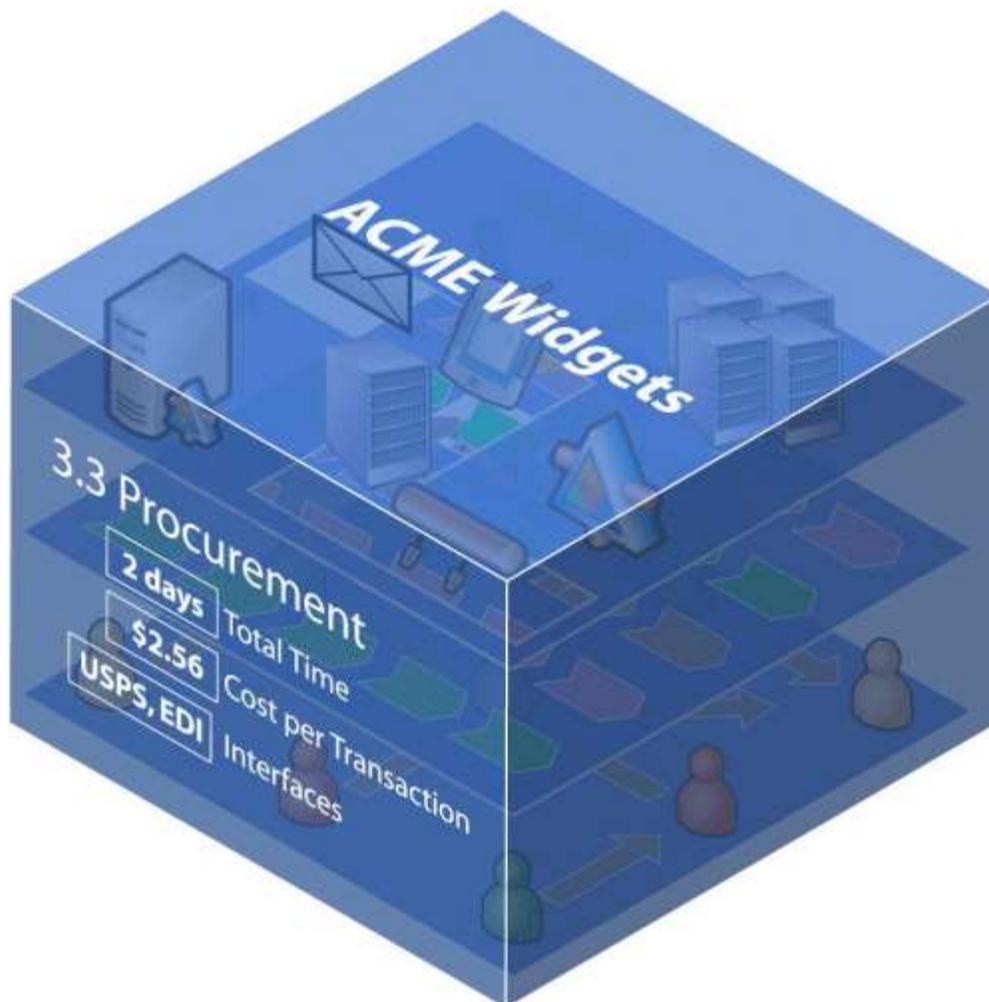


Figure 3. Capabilities are "black boxes," with inputs and outputs defined with explicit service level expectation

- *Capability connectors* represent the links that exist between business capabilities. Connections are not just simple messages; they contain rich semantic information. There are a variety of connectors in the capability model that focus on information exchange (input/output, supporting information), and control or establishment of policy (regulatory influences).

- *Business processes* describe how the business performs, or implements, the given capability, or how capabilities connect to deliver a desired outcome. Hammer and Champy, authors credited with creating the business process movement in the 1990's, carefully define business processes as end-to-end work elevated above procedures. Processes transcend organizational departments and functions.
- *Business capability mapping* is the definition and clear structural outline of the capabilities and their connections that drive the activities of a typical company. Business capabilities are the building blocks of the business architecture, so thinking of capabilities as an architectural blueprint is a good analogy, whereas the process is the implementation of that architecture at any given time. Once this more objective and stable view of the business is established, companies can understand significantly more about dependencies between capabilities and get a much better understanding of the business: within a line of business, across business units, and over time.

Capabilities - The Optimal Problem Solving Layer

We propose modeling the business as a structured—as opposed to a physically integrated—network of capabilities. This addresses the "rich interconnection" over which other aspects (such as the way technology is applied) can be layered from the start instead of added as an expensive and cumbersome afterthought. As you can see, this aligns closely with the black-box service-orientation model: Business capabilities are the structural elements (black boxes) that provide a stable foundation aligning service-orientation projects with their business drivers.

Business capability mapping and service-orientation provide a new set of complimentary tools that stretch the concepts of business beyond the physical corporate boundaries to include the entire value chain or ecosystem of business functions within the map. This abstraction of the business model allows management to examine *what* things work and *why* they work in a certain way before focusing on the details of *how* the work is done.

Capabilities—focused on the "what"—yield a substantially more stable and objective model of the area of focus. As discussed earlier, simple examples of business capabilities are "ship product" and "pay employees." Regardless of what the business implementation attributes ("how") of that capability are—whether it is in-sourced or outsourced, or manual or automated—the underlying capability of "pay employees" is the same.

As an example, take the checkout system at the grocery store. The cashier identifies a new customer, scans the products that the customer places on the conveyor belt, and eventually the customer pays the displayed total by some means. All the steps described are required capabilities that the grocery store has to provide in order to check out customers and receive the money for the products purchased. In the US—and increasingly in other areas of the world—grocery stores are adding self-service checkout systems. On first glance, it appears that self-service checkout as opposed to "managed" (executed by the cashier) checkout is sufficiently different to consider it as a new set of capabilities. As it turns out, though, the customer executes exactly the same steps outlined above—identification of a new customer/sale, scanning of the products, and eventual payment. However, there is one additional capability required that differentiates the self-service checkout: in order to avoid abuse by dishonest customers, self-service checkout requires a validity check. (This is implemented by placing the scanned product on a scale to

compare the weight of the scanned product with the weight of the product known by the systems supporting the scale.) So, while the set of capabilities differs only by this one—validating the purchased product—the processes for checking out customers varies quite significantly. The capabilities, the business architecture blocks, remain stable while the processes, or business implementations, change.

Capabilities Expose Interfaces

One of the most important aspects of capabilities is how they relate to other capabilities; thinking about capabilities in context of the ecosystem is really thinking about their connections. Thus, early detection of their connections to other capabilities, or essentially defining the interrelated ecosystem, is also a crucial step in understanding which boundaries can be traversed and which cannot, a crucial step for maximizing any re-architectural efforts. In fact, discovering these connections may be as valuable as defining the capabilities, because you manipulate and manage the connections, while the capabilities remain essentially unchanged black boxes. A connector could be as simple as the means that transitions the output of one capability to the input of another, an activity for instance that takes a customer request from a phone call (Take Order capability) and sends that to another department to place the order (Place Order capability). In addition, there may be a connector that controls another capability, such as a connection to the Place Order capability that requires notification of a new customer event so that pending orders may be combined.

At a business level, service level expectations (SLEs) strongly influence connections. So, the capability model is also based on the specific analysis of service levels and the notion that if changes in service levels can be achieved by changing who does the work (that is, evaluating different options of sourcing, among them outsourcing) the decision is made on an equal basis across all capabilities in the company. This allows businesses to exchange services without getting mired in the details of how the process is performed. For example, the company ADP can be used for the Pay Employees capability but there is no need to know all the details of how ADP processes the payroll—the only thing that matters is that defined service levels are met or exceeded.

Simply by knowing the differences in a capability's service levels, management can make decisions on the configuration of capabilities needed to improve business performance. Thus, interchangeability becomes a function of comparability. By knowing the rules for encapsulating a business capability (the rules for invoking and completing the services in a trusted fashion) you can complete the technical integration of the capability much more efficiently.

A Quick Illustration

You manage your relationship with your energy or telephone provider simply by knowing the capabilities they should provide and the service levels to which they should perform. You only care what they do, not how they accomplish it. There is a discrete number of capabilities they offer and a certain level of services that they contract to accomplish. To you, carriers are interchangeable based entirely on the service level they provide. If they don't perform you can take your business somewhere else where your service level expectations and actual delivery are more reliably met

You do not know the details of the process of delivering the dial tone to your cell phone, yet you can still very effectively manage the relationship and performance outcomes. The same is true for capabilities, they represent a level of abstraction where you can exchange services exchange and manage rules for performance.

Capability Model Overview

A business capability model is a nested hierarchy of business capabilities. It exposes all business capabilities across the relevant ecosystem. Because business processes span the entire value chain, a map of a business rarely covers the same information as the map of a physical company. For example, UPS and ADP are examples of companies that join with other companies to make up collective "businesses."

The business capability model is a taxonomic diagram that describes the network of capabilities used by the business.

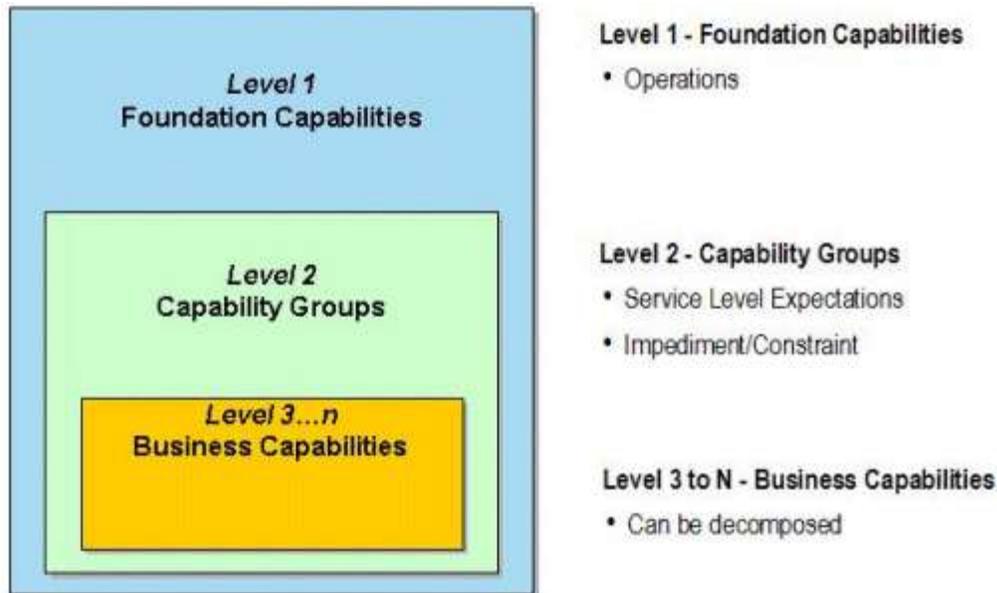


Figure 4. Business capability model taxonomy

Level 1 Foundation Capabilities

Foundation Capabilities address the entire ecosystem of the business. They are represented in two categories, Operations Capabilities (things inside of the physical business boundary of the company) and Environmental Capabilities (all of the other people and companies interact with the business that are outside of the physical boundary of the business).

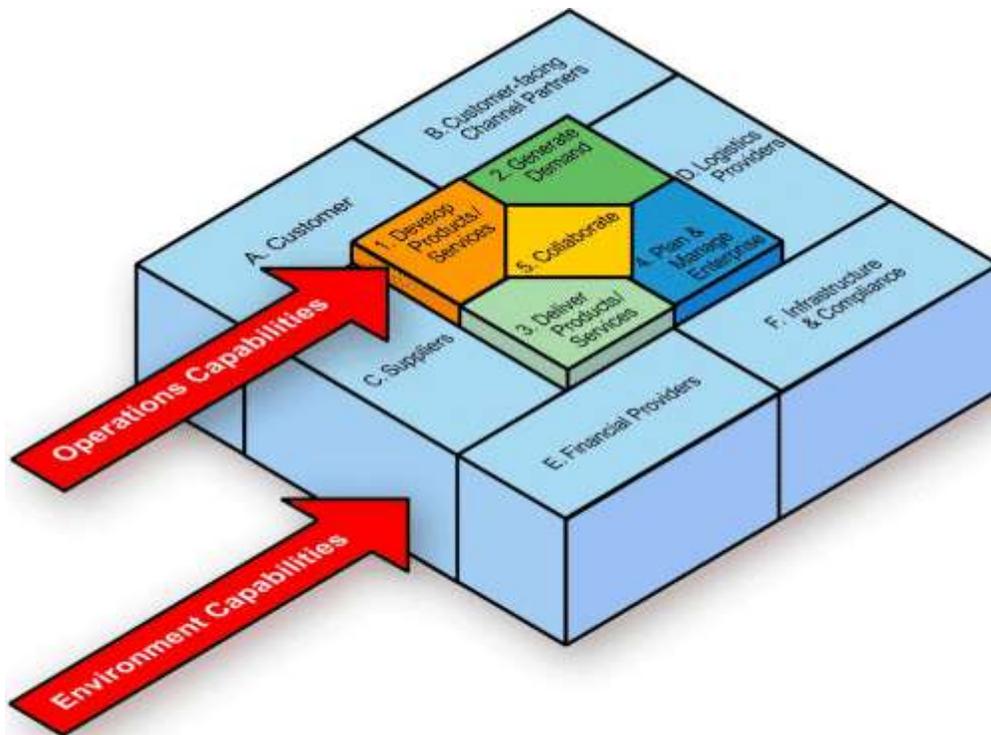


Figure 5. Level 1 Foundation Capability Model: Operations and Environmental Capabilities

Operational Capabilities

Capabilities—regardless of provider of any given capability—that are required to deliver the value identified as the goal of the business. Operational capabilities are the capabilities that a business owns or controls including how it:

- Develops products and services.
- Generates demand for those products or services.
- Produces or delivers the products and services.
- Collaborates and communicates with partners.
- Plans and manages the business.

These operations capabilities can take on industry and/or business specific names (for instance, "develop products/services" could also be "research and design") but the basic setup is consistent in almost every business

Environmental Capabilities

Environmental Capabilities address the capabilities outside the basic operations of the business that either influence value delivery (e.g., expectations of customers, compliance requirements by governments, or competitive capabilities by existing or emerging suppliers), or offer opportunities to leverage the ecosystem (including customers) for value delivery/differentiation. They are:

- Customers

- Customer Facing Channels
- Logistics Providers
- Infrastructure and Compliance
- Financial Providers
- Suppliers
- Governments and other regulatory bodies

Note that this model of a business includes the entire value chain so that it models your entire virtual business.

Level 2 Capability Groups

Capability Groups are the next level in the capability model. For example, within the core capability "1. Develop Products/Services" there is often a capability group called "1.1 Plan Products/Services." The Product Engineering group that plans products may further consist of a number of capabilities at level 3...n where specific capabilities and their attributes are described.

A capability group is often an important initial level for doing analysis because it is at the capability group level where service levels, impediments and constraints, and organizational ownership/accountability can first be abstracted and made actionable.

Level 3 Business Capabilities

Capability groups are decomposed into business capabilities. Business capabilities are the building blocks for the business capability mapping. Business capabilities can be decomposed into more granular business capabilities. There are detailed attributes that are captured at the business capability level. Within the analysis you may decompose some business capabilities to very detailed levels (Level 4 and beyond) and aggregate other capabilities at Level Three. It is not necessary to decompose all capabilities to the same level.

Conclusion

Originally, we asked:

- How do we prevent service-oriented architectures from following similar, hopeful initiatives into the same architectural mistakes of the past?
- How do we ensure that the chosen implementation architecture relates to the actual or desired state of business?
- How do we prolong the life expectancy of the implementation in an ever-changing environment?

The key point to take away is that service-orientation with Web services is only the *implementation* of a particular model. It is the quality and foundation of the model that determines the answers to these questions. Business capabilities give you a frame of reference to pose these questions and answer them for the various interconnected viewpoints in your business. It finds the stable elements of business to model your architecture around and provides a critical layer that closely aligns to service-orientation. service-orientation in turn supplies the compartmentalized, but connected, structure to implement capabilities so that IT meets the actual business requirements and provides a truly agile business.