

## FEATURED PAPER

# Conquering Complexity with Program Management (Part 2 of a Series)

By: Russ Martinelli and Jim Waddell

**Part 1 in the series are available at:**

[http://www.pmworldtoday.net/featured\\_papers/2007/feb.htm#2](http://www.pmworldtoday.net/featured_papers/2007/feb.htm#2)

## Introduction

“This is exactly what we need; we’re being consumed by complexity”. That was a comment we received from an audience member at a recent conference we were speaking at. This person’s comment was in response to our presentation showing how program management tackles complexity by breaking a complex development effort into multiple, smaller-scale projects that are managed in a concerted and synchronous manner within a program. He came to understand that the projects within his company have crossed the complexity threshold where conventional methods for managing a large-scale project as a single entity become ineffective.

This gentleman’s experience is not an isolated case. Our seemingly insatiable desire for ever-increasing wants drives our collective environment toward more challenging and complex ends. This is true in our careers, in our relationships, in our activities, and especially in the products and services we purchase and use. User demand for more customized solutions with highly integrated features is accelerating, leading to some significant business problems which we described in Part 1 of this article series<sup>1</sup>. One such problem, effectively managing complexity of product, service and infrastructure development is one of the most prevalent and challenging business problems to overcome.

Development complexity manifests itself in several ways: designs are more complex as features and integrated capabilities increase; the processes to manufacture the solutions have become more complex; the ability to integrate multiple technologies to meet end user desires is becoming increasingly challenging; and the development of the complex solutions requires a multi-discipline, multi-organization approach that many firms are not structured to implement or do not have the management skills required.

Of course, management of complex development efforts is not new. Complex products such as commercial and military aircraft, satellites and their launch vehicles, and technological weapons have been in development for decades in the aerospace and defense industries. What *is* new today is that development of highly complex products is not isolated to the aerospace and defense industries, but common in all industries where technology development is the cornerstone for gaining competitive advantage.

In this article we describe how companies that have been successful in managing complexity do two primary things: use a systems approach to define and develop the solutions, and adopt the program management model to manage and integrate the system elements into a holistic product, service or infrastructure capability.

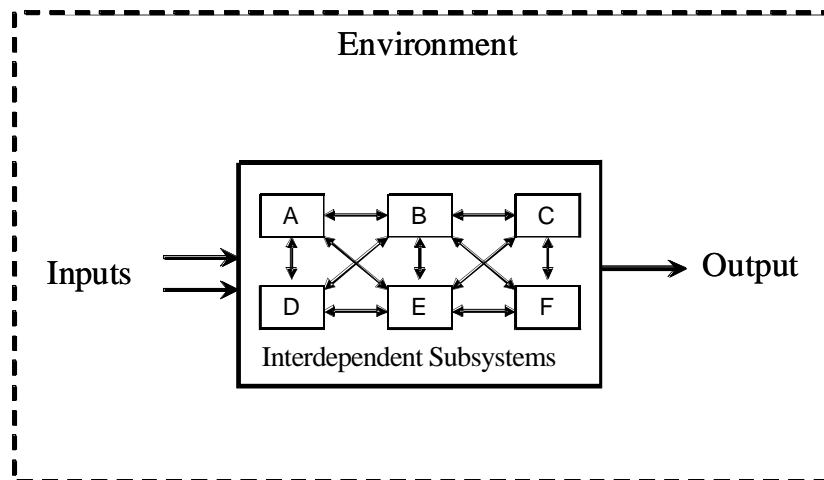
## The Genesis of Systems Thinking

The systems engineering approach to developing complex products was conceived during the cold war between the United States and the former Soviet Union. Following the Second World War, the U.S. continued a technology-focused strategy that replaced mass production of weapons. The cold war therefore became a contest in creating advanced technology weaponry<sup>2</sup>. However, development of these weapons – jet aircraft, nuclear submarines, cruise missiles, and spy satellites – required fundamental changes in organization and management practices. The rigid functional structure of the technical branches of the military and its project-oriented management techniques proved inadequate for the development of the complex weaponry. The military learned to think of weapons as systems that can be disaggregated into smaller subsystems and components, developed at the subsystem level, and then integrated into a final holistic system through a multi-discipline approach.

## Complexity and Systems

A system is defined as *a combination of parts that function as an integrated whole*<sup>3</sup>. Each part by itself is of little or no value. Value is only created in the interaction with the other parts of the system to perform the function of the whole. For example, the parts of a clock, such as cogs, dials, and hands, are of little value by themselves. They only provide synergistic value when they are assembled together and collectively perform the function of the clock (the system) – which is to record the passage of time<sup>4</sup>.

The simple system illustrated in Figure 1 can be visualized as a simple entity consisting of four primary elements: inputs, output, interdependent subsystems, and an environment within which the system operates. The nature of the system elements are unique to the specific function, architecture and environmental factors associated with the cohesive system. The subsystems are highly interactive and interdependent on each other and their role is to utilize the inputs provided by the environment to produce a desired output.

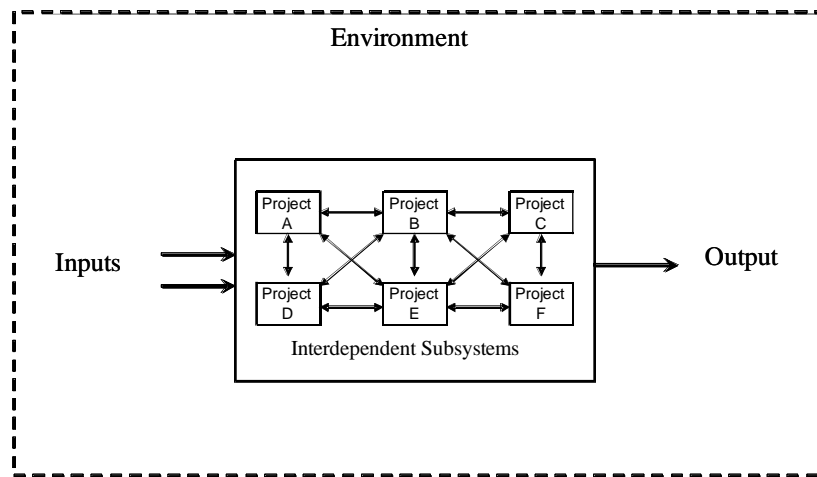


**Figure 1: Elements of a simple system**

A *complex* system is one that has many components and multiple technologies that are highly interactive with one another. They are interactive in such a way that it is difficult to completely

separate one part from another. The systems approach is effective in managing complexity because it involves breaking the complex system into smaller, more manageable subsystems, concurrently developing the subsystems, and then recomposing the system into a coherent solution within an existing architecture<sup>5</sup>.

From a program perspective, each of the subsystems is organized and managed as a project, led by a discipline-specific project manager. Therefore, the development program consists of multiple interdependent projects that must be managed collaboratively and synchronized over time by a program manager. Figure 2 show the correlation between systems engineering and the program management model.

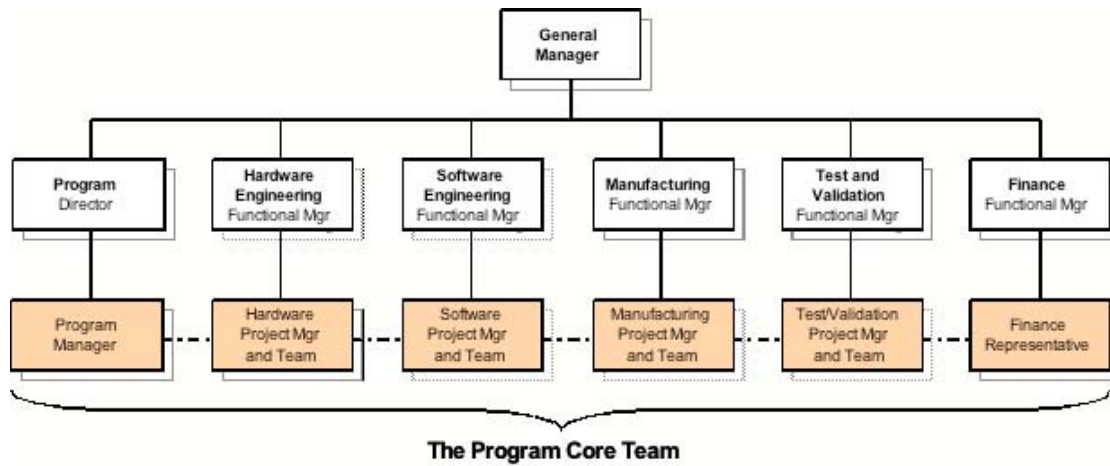


**Figure 2: Subsystems configured as projects within a program**

With the advent of the systems approach to breaking complex development efforts into smaller, more manageable elements, a new organizational structure and management approach had to be developed to comprehend the cross-discipline collaboration required to effectively integrate the system elements into a cohesive whole.

***Complexity, Systems, and Program Management***

While the U.S. Navy pioneered the systems engineering approach to develop complex products, the U.S. Air Force led the way in defining the organizational and management methods for coordinating and integrating the diverse talents and technologies across disciplines and time. Development efforts have been managed in a wide spectrum of organizational structures with varying degrees of success and efficiency in the past. These structures range from the purely functional organization at one extreme to the purely project-oriented structure at the other extreme. The optimal structure for the management of complex systems was found to be the matrix structure shown in Figure 3.



**Figure 3: The Matrix structure and Program Core Team**

Optimization came from the realization that the functional silos needed to be minimized in order to effectively manage the interdependencies between the functional specialists, and a new horizontally focused structure needed to replace the legacy vertical structure.

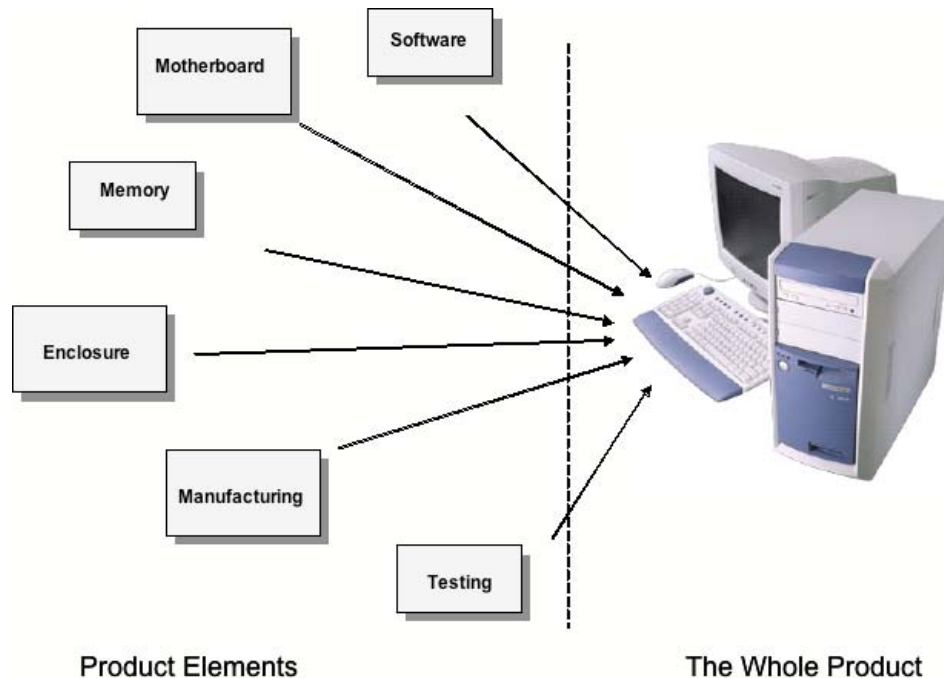
In the matrix organization, functional resources, expertise and training are maintained by the functional manager to support the long-term capability development of the firm. The functional project managers and specialists are ‘loaned’ to the program, and then return to the functional organization once the program is completed.

The management model that was conceived to manage complex technological development efforts is of course the subject of this article series – program management. In the 1950’s the new discipline of program management came into being as the managerial analogue to systems engineering. Under the program management model, ownership of the program belongs to the program manager, who is fully empowered by the senior management of the enterprise. The program core team, depicted by the shaded boxes in Figure 3, is the leadership and decision-making body of the system development effort that is responsible for ensuring that the business objectives and customer satisfaction are achieved. The program manager drives the cross-project, cross-discipline coordination and collaboration necessary to eliminate the barriers brought upon by functional silos. In the process, the program manager leads the simultaneous and synchronous development of the system components, and then drives the integration of the parts into a holistic system to be delivered to the customer.

### Program Management and the Whole Product

In our book, *Program Management for Improved Business Results*, we use the **whole product** concept to describe the systems approach for developing complex products, services and infrastructure capabilities. The whole product concept was first introduced in marketing literature; however, the concept is also an excellent way to describe the function and value of program management. We define the whole product as “*the integrated product solution that fulfills the customers’ expectation*”<sup>6</sup>.

For companies that use the program management discipline to develop products, services or infrastructure capabilities, the program manager is responsible for the delivery of the whole product to the marketplace or customer's environment. This means he or she must have a full understanding of what the customers expect in the total solution. The project managers on the program are responsible for the development and delivery of the individual elements of the whole product. Figure 4 graphically illustrates the whole product concept in the development of a personal computer which fits the definition of a complex system.



**Figure 4: The whole computer product**

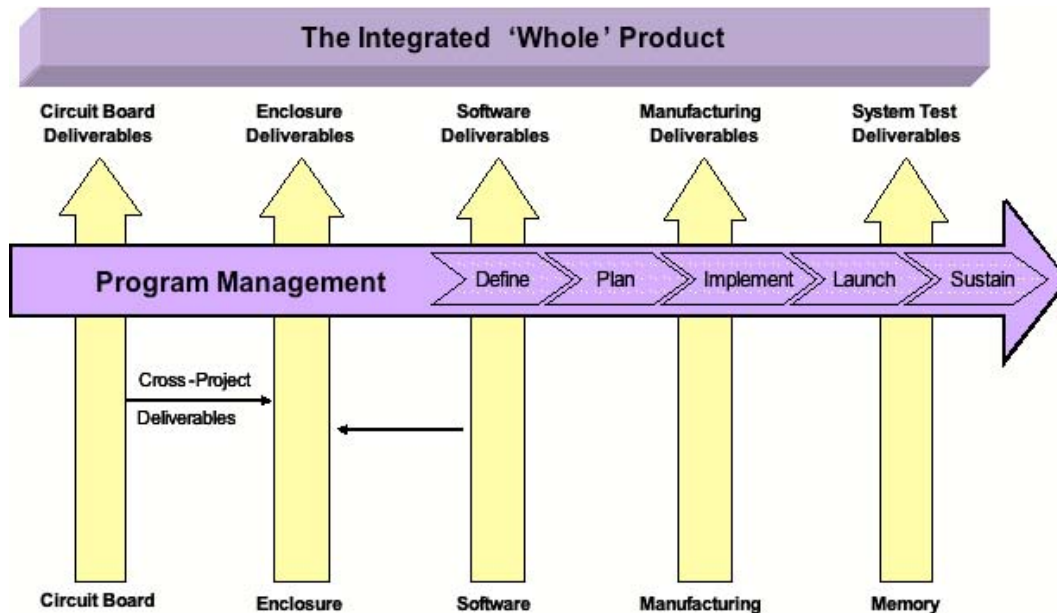
In this simplified example, the whole product consists of the integration of six elements: the motherboard development, the memory circuit board development, the software development, the enclosure development, product manufacturing, and product testing. Other functions normally involved in the development of a personal computer, such as architecture, product marketing, quality, customer support, quality assurance and finance are purposely not shown for the sake of simplicity.

The development program is organized in a similar manner. Each of the six elements of the personal computer shown in Figure 4 is organized as a project which is led by a project manager, and consists of a team of specialists in the domain they represent. The job of each project team is simple: plan, develop and deliver its respective element of the whole product to the other members of the program team.

The program manager in turn leads the development program, and is responsible for integrating the elements into a whole product and delivering it to the customers and stakeholders. In addition to

delivery of the product, the program manager is responsible for the achievement of the business objectives for which the product development effort was initiated.

When utilizing the program management model to develop and deliver the whole product, it is helpful to view management responsibilities within the program in two dimensions: vertically and horizontally. This concept, as illustrated in Figure 5, is a core characteristic of the program management model. Shown in the figure is a simple example of five elements that are involved in the personal computer example above – circuit board development, enclosure development, software development, manufacturing, and system test.



**Figure 5: Horizontal and vertical dimensions of a program**

Both the vertical and horizontal elements involved in the management of a program are clearly evident in the figure. First, let's look at the vertical or project management element. Each subsystem of the computer is structured as a project, and is led by a project manager. These project teams of specialists are responsible for the development and delivery of their respective piece of the computer under development, which represents a single element of the whole product.

The work of the program manager cuts *across* the project teams, therefore managing the horizontal dimension of the program. In order to create an integrated product, service, or infrastructure solution, the program manager is responsible for three primary things: 1) ensuring the deliverables from the project teams form an integrated whole product solution; 2) ensuring the highly complex network of project interdependencies is synchronized and coordinated throughout the program life cycle; and 3) ensuring the program business case remains viable. As this example illustrates, when a systems approach is employed for managing a complex development effort, program management is the most effective management model to employ.

## Conclusion

Project management has much of its roots in the construction industry. Program management on the other hand, was conceived in the U.S. defense and aerospace industries as the management analogue to the systems engineering approach to develop complex products. Even though program and project management have their roots in different industries, companies that have learned to utilize both disciplines to develop complex solutions under the program management model have found a competitive advantage over their business rivals who still struggle to conquer development complexity using conventional methods.

In our next article, “The Program Management Maturity Model: A Framework for Change”, we further characterize the program management model for managing products, services and infrastructure capabilities. We will show how the Program Management Maturity Model™ can be used as a framework for introducing and expanding the program management discipline within companies that wish to do so.

## References

1. Martinelli, Russ and Jim Waddell, “Program Management: It’s About the Business”, PM World Today, January 2007, Volume IV, Issue 1.
2. Sapolsky, Harvey M., *The Business of Systems Integration*, Oxford University Press, London, 2004
3. Titcomb, T.J., “Chaos and Complexity Theory”, Info-line, July 1998, Issue 9807
4. Stacey, Ralph D., Douglass Griffin and Patricia Shaw, *Complexity and Management*, New York, NY: Routledge, 2000
5. Sapolsky, Harvey M., *The Business of Systems Integration*, Oxford University Press, London, 2004
6. Milosevic, Dragan Z., R. Martinelli, J.M. Waddell, *Program Management for Improved Business Results*, Hoboken, NJ: John Wiley & Sons, 2007.

© Copyright 2007 by Russ Martinelli and Jim Waddell



*Russ Martinelli*  
Author



Russ Martinelli is the Manager of Program Management Methods within the Corporate Platform Office at Intel Corporation, where he focuses on the definition and implementation of program management practices across Intel. Additionally, Russ is the chairman of Intel's global Program Management Community of Practice, an adjunct professor at the University of Phoenix, and co-founder of the Program Management Academy. Russ has held a variety of positions at Intel and Lockheed Martin in the areas of systems engineering, general management, operations management, and project and program management.

Contact Russ at: <mailto:russ.martinelli@programmanagement-academy.com>



*Jim Waddell*  
Author



Jim Waddell is an independent consultant specializing in program management and mergers and acquisitions. He is the former Director of Program Management for Tektronix Inc. where he established and led the Tektronix's first worldwide Program Management Office (PMO). Additionally, Jim is an adjunct professor at the Oregon Graduate Institute, a founding member of the Program Management Forum in Portland, and the co-founder of the Program Management Academy. Jim has held a wide range of managerial and operational roles ranging across engineering, marketing, systems and manufacturing in the high tech and energy industries. Contact Jim at: [jim.waddell@programmanagement-academy.com](mailto:jim.waddell@programmanagement-academy.com)