The Double Challenge in Engineering Complex System of Systems

Philip Boxer, Edwin Morris, Dennis Smith, Bill Anderson

Traditional software engineering practices were defined back when development was largely controlled by organizations that could define (relatively) stable requirements, build to those requirements, and deliver a system to the customer. But – to borrow Henry Ford's phrase – customers could have it in any color so long as it was black. More recently, attention has turned to coordinating activities of multiple organizations and systems within an enterprise to perform several tasks or deliver tailored responses. This change in focus from a specific delivered system to the several capabilities that could be provided is reflected in recent software engineering practices such as product lines and families of systems.

However, meeting the expectations for the emerging, complex systems of systems required to support integrated military strategies, homeland security responses, and nationwide health information networks requires coordination across multiple enterprises (e.g., police, fire, and military components in the case of homeland security) to respond to dynamic situations and demands.

As depicted in Figure 1, this presents a ***double challenge*** that involves:

1. A **Governance Challenge** of collaborating with an increasing number and diversity of enterprises. We must develop approaches that support cooperation across unrelated enterprises with no unifying controlling structure
2. An **Agility Challenge** of providing situation-appropriate response in changing situations. We must select technologies, processes, and structures that are sufficiently agile to support the desired response.

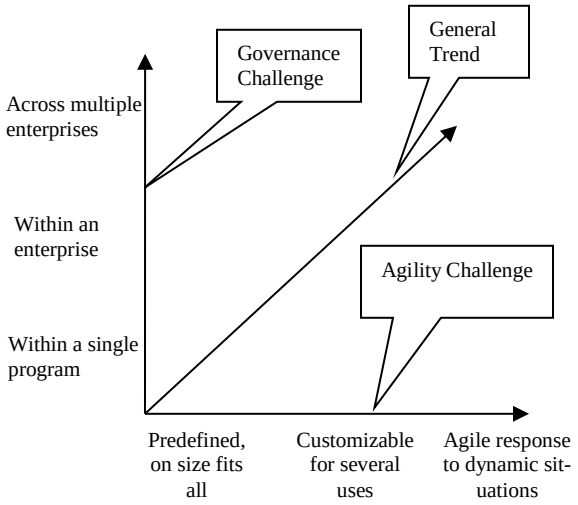

Figure 1: The Double Challenge

The Governance Challenge

The Governance challenge involves the changing nature of the collaboration needed to build computer systems. For much of the short history of computing, collaboration was only needed to coordinate activities involving point to point interfacing of systems within a well-defined program. As anyone experienced with building or maintaining relatively complex software can attest, even this required coordination of the activities of multiple personnel, perhaps with competing interests. However, there was normally one sovereign authority that could make and enforce decisions.

In more complex cases, collaboration was needed to coordinate the activities of multiple systems within a single enterprise. This situation was typical of an organization that attempted to relate and integrate the activities of several systems to provide consistent and related information across the enterprise. It became the driving force behind efforts to integrate data and processes within a single business organization in an Enterprise Resource Planning (ERP) system, or, to construct a common situational picture of the location of friendly and enemy assets by fusing the data contained in multiple military systems operated by the Department of Defense. We will refer to these approaches as integration and fusion approaches respectively.

Of course, enterprises of any significant size are not homogeneous and normally consist of multiple levels of relatively sovereign "sub-enterprises" (e.g., branches, divisions, directorates, teams). For example, the U.S. Air Force Space Command represents one level of enterprise. The Air Force is another. The U.S. Military represents yet another level of enterprise, and it must determine rules to allow coordination of activities across various commands. Further, NATO is another and it determines rules to coordinate activities across the militaries of member nations. In each of these cases, whether adopting an integration or a fusion approach, a hierarchy is assumed to exist that can (presumably) de-conflict the actions of participants in such a way that they can be brought together as an internally consistent whole.

But system integration activities become increasingly complex as they cross organizational boundaries. These activities are commonly managed by creating a special 'cross-cutting' management organization with authority that can span these boundaries. The success of these cross-cutting organizations depends on the degree to which they can establish centralized coordination and control of activities i.e., to establish sovereignty over the activities and systems to be integrated, as well as the integration activity itself.

We are familiar with the difficulties this presents to such cross-cutting organizations. The interests of the cross-cutting organizations often come into conflict with those of the organizations across which they are 'cutting'. The result is often an instantiation of the "Golden Rule": *He who has the gold rules*. In principle, these difficulties can be resolved by appeal to the higher authority of the enterprise as a whole. In practice this can prove to be infeasible because of the complexity of the issues involved.

When the cross-cutting approach is extended to multiple enterprises without an already-existing unifying hierarchy, however, a discontinuity occurs. This is illustrated along the vertical axis of Figure 1. Moving from bottom to top, collaborators become increasingly autonomous and their motivations, policies, procedures, and capabilities become increasingly diverse. Indeed, because of the fluidity of environment in which complex systems must execute, the owners of those systems cannot fully anticipate who these collaborators will be.

To meet the challenge posed by collaboration with an increasing number and diversity of enterprises requires the development of processes and strategies that support negotiation of relationships with these sometimes unanticipated partners. Therefore we must develop ways of negotiating *collaborative governance* across unrelated enterprises in rapid order.

The Agility Challenge

The second challenge, illustrated in Figure 1 along the horizontal axis, describes the way that an enterprise's[1] computing systems can respond to the needs of its customers. In simple cases, an enterprise can define the systems it needs to provide to its customers, put together (or otherwise acquire) these capabilities, get them into the right hands, and keep them there. This is the strategy used by enterprises in building tightly focused applications for general use (e.g., Microsoft Word). In more complex cases, a single predefined system capability is insufficiently flexible for the variety of customer demand to be addressed. In this case, enterprises try to parameterize their applications, processes, and organizational structures in such a way as to allow them to customize them to multiple customer situations. Technologies and systems with customizable interfaces and assembly strategies such as Product Lines or Families of Systems are employed to increase the range of customer needs that can be met by the organization. This more complex case is manifest in the exploding interest in service oriented architectures (SOA), which seek to maximize the extent to which services can be composed, but where the SOA environment retains control over the composition process.

Solutions to these first two cases are driven by what the supplier can provide, either in a predefined system or through a customizable set of applications designed to work within some prescribed set of behaviors. However, customers are expecting ever more flexible response to (often rapidly) changing situations. They need to determine when and how a technology is employed as well as the interconnections with other technologies in order to solve unique and changing problems. In these cases, the technology and systems brought to bear and the effects achieved with that technology are dependent on the specific context in which the customer wants to employ the capability and can no longer be under the control of the suppliers of the technology and systems. In effect, a customer is adding a new level of composition and synchronization of components, the individual services of which may well be being provided by SOA environments. Thus, this case represents a second discontinuity – here between situations controlled by suppliers and situations controlled according to the demand situation to which the user must respond.

Along the vertical (governance) axis, it is increasingly common and necessary to interact with enterprises external to ones own, both at the system and work process level. This tendency is evidenced in the engineering community by the widespread push for increasing interoperability and standards that support it. It is at the root of efforts to integrate the diverse systems of commercial enterprises into supply chains, to develop common operational pictures across domestic and allied military forces, and to provide emergence response capabilities crossing military, police, government, health care, and other networks.

Along the horizontal (agility) axis, there is increased recognition in commercial, government, and military sectors that advantage is best gained by developing system capabilities that can be rapidly aligned in new ways to support responses to changing demands. For the commercial

[1] We have used enterprise here to mean any sort of entity that must respond to the second challenge. That enterprise may entail one or more organizations.

organization, this means supplier relationships that can address changing needs – whether by supporting rapid changes to products and manufacturing approaches or providing new ways for service delivery as determined by the customer. Ideally, the alignment of organizational structures, processes, and systems capability is determined by the demand.

Thus, the general trend is clearly up and to the right in as depicted in Figure 1. However, the comfort zone for building systems capabilities is down and to the left as (generously) depicted in Figure 2.

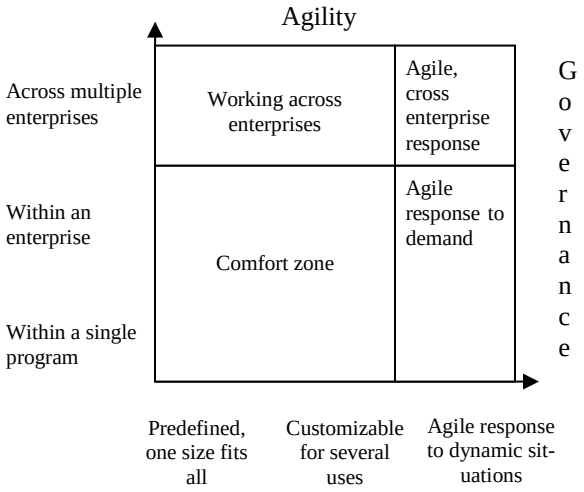

Figure 2: The Comfort and Other Zones

The apex of the up-and-right trend (at the top right) involves cross-enterprise and agile response to changing demand situations. This goal is perhaps best developed today in the "power to the edge" strategy of the U.S. Military, which has recognized that only at the "edge" can it recognize and respond appropriately to the demands being placed on it for cross-command (e.g., services, allied militaries, and non-traditional allied) responses.

We believe that the future of software engineering will be dominated by the double challenge of developing governance approaches that can work across enterprise boundaries and identifying ways of meeting the demand for increasing agility in the engineered capabilities that are provided.

We are concerned that focusing on developing engineering strategies to improve or "fix" individual symptoms of the double challenge, such as poor coordination of development efforts across organizations, and problems with configuration management may be helpful in some situations but ultimately will not solve the problem.

It is more likely that fixes for individual symptoms along one axis of the double challenge will actually complicate the problem along the alternate axis. Thus, for example, developing a new, virtual organization that imposes a hierarchy across enterprise boundaries may lead to reduced flexibility in response to demand situations. The key question to be answered then is how do enterprises develop the ability to work across enterprise boundaries while simultaneously providing the requisite agility to respond to the changing demands of the customer?

A starting point involves recognizing several distinct types of systems of systems based on the type of authority possible and the ability of that authority to control behavior. This approach allows us to begin to characterize requisite engineering practices. Of particular interest to us are those practices that support distributed collaboration (e.g., power-to-the-edge). A critical activity for the future is to consider the double challenge (i.e., governance and agility) in relation to achieving distributed collaboration.

In response to the double challenge, the SEI is developing the System-of-Systems Navigator[SM], an integrated set of priciples, tools, methods, techniques, and improvement cycle activities. The SEI is currently developing capabilities to recognize several distinct types of systems of systems based on the type of authority possible and the ability of that authority to control behavior. This approach allows us to begin to characterize requisite engineering practices. Of particular interest to us are those practices that support distributed[2] collaboration (e.g., power-to-the-edge). A critical activity for the future is to consider the double challenge (i.e., governance and agility) in relation to achieving distributed collaboration.

About the Authors

Philip Boxer has been a strategy consultant to organizations since the late 1970s, supporting leadership teams across many different industry sectors, both public and private. His focus is on the challenges organizations face from asymmetric forms of demand and the mitigation of risks associated with failing to develop requisite agility. He has developed a number of supporting methods and tools needed to support strategy formation and collaborative design processes, including visual PAN and its associated forms of analysis. He is a senior member of the technical staff of the Integration of Software-Intensive Systems (ISIS) team at the Software Engineering Institute.

Edwin Morris is a Senior Member of the Technical Staff at the Software Engineering Institute, assigned to the Integration of Software-Intensive Systems (ISIS) Initiative. He is currently investigating approaches to achieving technical interoperability between complex systems and programmatic interoperability between the organizations that build and maintain them. Previous activities involved improving processes and techniques for the evaluation and selection of COTS products, and the development of the COTS Usage Risk Evaluation (CURE) technology. Before coming to the SEI, Morris developed custom operating systems for embedded microprocessors along with support tools to predict and monitor the performance of real time systems.

Dennis Smith is the lead for the SEI Integration of Software Intensive Systems (ISIS) Initiative. This initiative focuses on addressing issues of interoperability and integration in large-scale systems and systems of systems. Earlier, he was the technical lead in the effort for migrating legacy systems to product lines. In this role he developed the method Options Analysis for Reengineering (OARS) to support reuse decision-making. Smith has also been the project leader for the CASE environments project. This project examined the underlying issues of CASE integration, process support for environments and the adoption of technology. Smith has published a wide variety of articles and technical reports, and has given talks and keynotes at a number of conferences and workshops. He has an MA and PhD from Princeton University and a BA from Columbia University.

[2] As apposed to directed collaboration that has a controlling authority.

William B. Anderson is a senior member of the SEI technical staff. Bill’s research interests include integration and interoperability of complex software systems, COTS and reuse management, cost estimation, and business case justification of complex systems. A former Vice President for a Fortune 500 company, Bill is broadly experienced with factory floor and business; processes, support systems, automation, and management. He has many years of experience in large system project management and has successfully led operational, financial, product line, and new product launch groups.

The views expressed in this article are the author's only and do not represent directly or imply any official position or view of the Software Engineering Institute or Carnegie Mellon University. This article is intended to stimulate further discussion about this topic.