Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 8 (2012) 303 – 308

Procedia Computer Science

New Challenges in Systems Engineering and Architecting
Conference on Systems Engineering Research (CSER)
2012 – St. Louis, MO
Cihan H. Dagli, Editor in Chief
Organized by Missouri University of Science and Technology

Patterns of Success in Systems Engineering of IT-Intensive Government Systems

George Rebovich, Jr., Joseph K. DeRosa

The MITRE Corporation, M/S M370, 202 Burlington Road, Bedford, MA 01730, USA

Abstract

The objective of this effort was to discover patterns of success in the systems engineering of information-intensive systems in a government acquisition environment using the method of positive deviance. Two large-scale success patterns were observed, each with several recurring sub-patterns. "Balancing the Supply Web" addresses "social" interdependencies among enterprise stakeholders who have different equities in the capability being developed. "Harnessing Technical Complexity" addresses the technical interdependencies among system components that together deliver an operational capability for the enterprise. The large-scale patterns are two sides of the same coin. The programs studied achieved success because of the way they each navigated through these dual interdependencies.

Keywords: positive deviance, patterns, systems engineering, acquisition.

1. Introduction

Despite numerous attempts at reform of systems engineering and acquisition processes, government acquisition of IT-intensive systems has remained resistant to improvement. Both more and less oversight has produced equally unsatisfying results. Better requirements gathering, evolutionary development, etc., all aimed at solving specific problems, seem not to produce lasting results.

When faced with the question of how to improve the practice of systems engineering and acquisition in a government department or agency, a common solution approach is to put a spotlight on what has gone wrong in recent, high-profile programs and to posit specific fixes. Often, the fixes are variations of approaches that have worked well in environments different from that of the department or agency, such as commercial businesses or government "skunk works." The difficulty in transferring approaches that work in one environment to another is in inferring exactly which patterns of an approach must be copied intact, which should be modified, and which may be ignored, to duplicate success in the new environment. In complex social systems many results – both positive and negative – derive from interactions among

1877-0509 © 2012 Published by Elsevier B.V.
doi:10.1016/j.procs.2012.01.064

multiple patterns that are not always fully understood. Attribution of credit in complex systems is fraught with difficulties and this is why solution approaches transplanted from other environments often fail.

2. Positive Deviance

The positive deviance approach was pioneered in 1990 by the Sternins in fighting malnutrition of Vietnamese children [1]. Later it was applied to management problems by Seidman and McCauley [2]. More recently, it was popularized by Gawande in medical settings [3]. The team developing United States Department of Defense guidance for the engineering of systems of systems (SoS) used some of the key principles of positive deviance in its work [4, 5].

Positive deviance is an approach to improvement based on the idea that every community performing an activity has individuals or teams whose attitudes, practices, strategies or behaviors enable them to function more effectively than others with the same resources and environmental conditions. It is a search for what works. Because positive deviants are embedded in the same environment as the rest of the community, attribution of credit problems are less severe than when transferring solution approaches across environments. Because many communities are reluctant to change fundamental beliefs based on outsider say-so, positive deviant ideas are more likely to be accepted by their community. We frame the positive deviance approach to improvement in the evolutionary processes of variation, selection and amplification [6]. It is fundamentally different from improvement processes like Total Quality Management or Six Sigma. Positive deviance takes advantage of natural variations in processes rather than trying to eliminate them.

3. Patterns

Architect Christopher Alexander introduced patterns as a way to capture design solutions for recurring problems in constructing buildings and cities [7]. Patterns have been applied to software design [8] and system architecture [9]. They were first proposed at an INCOSE conference in 1998 [10] and subsequently used by Haskins to deal with social aspects of systems engineering [11]. Simpson [12] and Cloutier [13] have developed systems engineering patterns. We use patterns to capture and communicate positive deviance observed in systems engineering and acquisition of IT-intensive government systems.

Patterns provide tested methods that incorporate past experiences. Because a pattern provides only the essence of a solution, there is latitude for innovation to tailor it to the situation at hand. It becomes a mechanism to communicate among peers and can be improved via adaptation and learning [14].

The construction industry relies heavily on patterns, yet no two developments or buildings are ever exactly the same. Object-oriented programming relies heavily on architectural level patterns and for such functions as delegation and aggregation. The World Wide Web is an example of a large IT-intensive enterprise using patterns to build system components. These have all evolved over time through the efforts of many individuals and have endured because of their relative success.

4. Identifying Success Patterns

The authors identified 30 IT-intensive government programs all developing capabilities to be embedded in a larger information enterprise, which had demonstrated some notable success. Twelve programs were selected for detailed analysis. The complexity of the programs was characterized across eight dimensions, using a systems engineering profile tool [15]. Each program had moderate or high complexity in several dimensions. The authors interviewed the front-line SE practitioners who cope with

the exigencies of the government acquisition system and are in a position to influence or observe positive deviance in acquisition performance.

Each of the programs achieved success because of the way they navigated through the two large-scale patterns via a number of fine-scale sub-patterns. In most cases they came together and worked collectively to help construct a successful acquisition program, rather than being isolated "silver bullets."

5. Balancing the Supply Web

This pattern sufficiently balances the web of potentially conflicting stakeholder demands as a strategy for supplying a new capability in a complex development and acquisition environment. It is akin to how a balanced ecosystem thrives while individual species have their ups and downs.

The reality of the government acquisition process in an IT-rich environment is that the program offices are best viewed as being immersed in a supply web rather than being the middle men in a supply chain. In a traditional supply chain, goods and services flow in one direction and compensation flows in the reverse direction. Government acquisition is more complicated. The money flows from tax dollars through the Congress and various departments or agencies of government to a designated program office and fans out to contractors and other agencies involved in the acquisition. Goods and services flow from contractors to end-users. Certain government agencies establish regulations and rules, and expect assurance of compliance to flow back to them, including independent test organizations and various "watchdog" agencies. It is this supply web that is the real and difficult context of government acquisition.

Increasingly, the government develops operational capabilities by combining new components with existing infrastructure and systems. Rarely does a new capability result from a system composed of a single piece of equipment or software or from investment by a single authority. Program offices are delivering a capability in context of the enterprise. Acquisition becomes more complex because of the web of dependencies among the various components and stakeholders.

With enterprises moving from a platform-centric to net-centric model, the measures of how well they achieve their objectives are more diffused and implicitly defined. Control gives way to influence and accommodation as the system or component being developed must fit into the context of the larger enterprise rather than exist as a standalone system. The one-dimensional supply chain view of acquisition often causes program offices to underestimate resources required to balance the supply web.

Usually the program office does not have enough resources to completely satisfy all stakeholders. A balanced strategy is needed that copes with the interdependencies of the stakeholders and their sometimes conflicting constraints. Since the influence and equities of the stakeholders vary from situation to situation, interactions with some stakeholders must be strengthened and others reduced. The program office must manage not only stakeholder needs but also their expectations and, ultimately, their acceptance that the enterprise capability is sufficient and satisficing.

Balancing the Supply Web is essentially a "stakeholder balancing act." Several principle forces at work were observed.

Conflicting definitions of success: While all stakeholders might agree on a common overarching program goal, each stakeholder frequently defines success differently. Different parts of the government may be responsible for setting requirements, providing logistics support, defining key performance parameters, and developing the system. Accepting all stakeholder views of success as immutable and

attempting to satisfy them can lead to confusion, frustration, and dissipation of resources and effort, ultimately resulting in failure to deliver.

Competing stakeholder resources and influence: Stakeholders have resources (e.g., funding or personnel) and influence (e.g., in the form of end-user legitimacy or policy compliance imperatives) that they use to define the value proposition of the item being developed. They use these resources to move other stakeholders towards their own definition of success. Ultimately, program success is determined by the results of all this enterprise-wide cooperation and competition.

The positive deviant program offices were aware of the interdependencies of their stakeholders. They viewed acquisition as a supply web, where contractors, competitors, budget authorities, regulators, etc. all interacted with the program office as well as with each other. As in system dynamics [16] or food webs [17], the interdependencies had to be identified and understood by the program office.

Successful program offices also apply their limited resources judiciously across the supply web, expending more or less of their own resources and stimulating interactions between stakeholders as needed. Because they understand the interactions, they are able to go beyond balancing stakeholder needs to actively influence stakeholder perceptions of the enterprise context and acceptance of its consequences.

Depending on project circumstances and the environment, getting stakeholders to understand and accept a context beyond their own perspective is accomplished in different ways. Thus, the new stakeholder balancing act requires delivery of capability and context.

6. Balancing the Supply Web Sub-Patterns

A number of recurring Balancing the Supply Web sub-patterns were observed in the interviewed programs [6].

- ***Up Close and Personal*** establishes strong and intimate ties with end-users to ensure a high-priority, pressing need is met.
- ***Close, but not too Close*** concentrates on getting a large number of end users to accept a standard set of capabilities and compensating them with rapid deliveries of their most valued capability.
- ***Divide and Conquer*** deals decisively with all stakeholders by dividing them into groups and satisfying each group's interests separately.
- ***Circle of Trust*** fosters positive social interactions among stakeholders to improve the willingness of opposing factions to compromise.
- ***Role and Responsibility Sub-nets*** clearly defines sub-nets within the stakeholder community for each decision or product to be supplied.
- ***Seek Secondary Sources*** seeks small flows of resources from secondary sources that have large impact on robustness of program and capability delivered.
- ***Network Beats the Node*** deliberately takes advantage of relationships in the network of stakeholders to create a resource greater than the sum of the parts.
- ***Top Cover*** uses informed acquisition authorities to shape the stakeholder environment.

The first three represent alternative strategies for matching stakeholder needs to available program office resources. The next two recognize the importance of positive social interactions and clear roles and responsibilities in managing a supply web. Seek Secondary Sources recognizes that in complex systems, small changes can have large effects. Network Beats the Node emphasizes that the power of the network is as important in the social context as the technical. Top Cover relates back to Close, but not too Close,

and recognizes the age-old maxim that the right kind of senior management involvement helps ensure project success.

7. Harnessing Technical Complexity

This pattern harnesses the technical interactions between systems in a networked enterprise. Layered architectures have been pervasive since the early days of information systems. The N-tier architecture is fairly well-established and its operation and benefits are well-understood. However, not all programs in the enterprise agree on the definition of layers or on the couplings between layers.

When a program office develops a component to enable a new capability, it does so in conjunction with other components and existing infrastructure. The infrastructure most likely is not optimized for the component being developed, but rather is designed for broadest possible use. This is analogous to using the federal highway infrastructure to travel between two points even though it is not the shortest distance between origin and destination.

A strategic technical plan (STP) codifies a small number of agreements on such things as the organizations of the layers and the coupling between them. These agreements are not conceived by experts as the proper way to do things, but rather are observed by practitioners as the common way things are done across the enterprise. Typically, a small number of common elements are responsible for a large percentage of the workflow in an enterprise. These common elements represent points of strategic convergence and bring order to the enterprise while allowing variety between convergence points.

By adhering to an enterprise-wide STP, each program is assured of at least some interoperability with others in the enterprise. By carefully choosing a small number of standards that represents a large percentage of the enterprise workflow, the enterprise comes alive with possibilities not previously available. Some new capabilities are added without ever having been planned.

The successful programs took advantage of Internet and WWW standards. They adopted the standard ISO model for layers, IP for enterprise routing and communication, XML for data representations, etc. However, often databases, message sets or service interfaces are not common across the enterprise. Details are in [17]. A small amount of standardization can result in a large amount of interoperability.

8. Harnessing Technical Complexity Sub-Patterns

A number of recurring Technical Complexity sub-patterns were observed in the interviewed programs [6].

- ***Seeing is Believing*** builds a capability reference implementation for the enterprise that shows what can be done, how it works and what it should do when done.
- ***Riding on the Infrastructure*** builds new capabilities are built on top of the existing infrastructure.
- ***Loose Couplers*** establishes isolation between layers and integration across the enterprise.
- ***Social and Technical Alignment*** aligns people, processes and technologies to match development and acquisition to the enterprise structure.
- ***Plan to Re-plan*** stimulates desired behavior through feedback and incentives, and then learn from results what behavior is desired next.
- ***Technology Surfing*** uses an ongoing process of identifying new and emerging technologies, experimenting with them and integrating what works into the evolving enterprise – “catch the next technology wave” rather than “create or wait for the big one.”

- ***Architect.org*** in which the government program office team assumes full responsibility for architecting and overseeing development of the system capability.

The capability reference model in the first pattern reduces complexity via a proof-of-concept prototype as a baseline from which users, contractors and other system developers can work. Building new capabilities from infrastructure and loosely coupled components as in the next two patterns imitates the WWW architecture in harnessing complexity. The Social and Technical Alignment pattern leverages all these patterns by structuring the acquisition to match the technical architecture. The next two emphasize an evolutionary approach that specifically takes advantage of experimentation and learning for re-planning and technology insertion. Architect.org, uses a government architect to manage the system development from inception to operation, adhering to the enterprise context.

9. Future Work

These preliminary results are promising. Interviewing additional programs would build confidence in them, as would interviewing unsuccessful ones to determine whether their patterns were different.

References

[1] Pascale, R., J. Sternin and M. Sternin. The Power of Positive Deviance: How Unlikely Innovators Solve the World's Toughest Problems. Harvard Business Press, 2010.

[2] Seidman, W. and M. McCauley. Harvesting the Experts' "Secret Sauce" To Close the Performance Gap. Performance Improvement Journal, vol. 42, no. 1, pp. 32-39. 2003.

[3] Gawande, A. Better: A Surgeons Notes on Performance, New York: Henry Holt, 2007.

[4] OUSD AT&L. Systems Engineering Guide for Systems of Systems. Washington, D.C.: Pentagon, 2008.

[5] Dahmann, J., G. Rebovich and J. Lane. Systems Engineering for Capabilities. CrossTalk: The Journal of Defense Software Engineering, vol. 21, no. 11, pp. 4-9. 2008.

[6] Rebovich, G. and J. DeRosa. Patterns of Success in Systems Engineering: Acquisition of IT-Intensive Government Systems. MITRE Product 110369, The MITRE Corporation, Bedford, MA 2011.

[7] Alexander, C. The Timeless Way of Building, Oxford Press, 1979.

[8] Gamma, E., R. Helm, R. Johnson, and J. M. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. New York: Addison Wesley, 1994.

[9] Fowler, M. Patterns of Enterprise Application Architecture New York: Addison Wesley, 2002.

[10] Barter, R. A Systems Engineering Pattern Language. Proceedings of the 8th Annual International Symposium of the International Council on Systems Engineering Vancouver, BC: July, 1998.

[11] Haskins, C. Using Patterns to Transition Systems Engineering from a Technological to Social Context. Systems Engineering, Volume 11 Issue 2, Summer 2008, pp. 147 – 155.

[12] Simpson, J. and M. Simpson. Foundational Systems Engineering Patterns for a SE Pattern Language. Proc. 16th Annual INCOSE Symposium, Orlando, FL July, 2006

[13] Cloutier, R. and D. Verma. Applying Pattern Concepts to Enterprise Architecture. Journal of Enterprise Architecture. May 2006, pp. 34–50.

[14] Holland, J. Hidden Order: How Adaptation Builds Complexity. Cambridge, MA: Helix, 1996.

[15] Stevens, R. Engineering Mega-Systems: The Challenge of Systems Engineering in the Information Age. Boca Raton, FL: Auerbach/Taylor & Francis, pp. 118-120, 2011.

[16] Sterman, J. Exploring the Next Great Frontier: System Dynamics at Fifty. System Dynamics Review, vol. 23, pp. 89-93, 2007.

[17] Dunne, J.A., R.J. Williams, and N.D. Martinez. Network Structure and Biodiversity Loss in Food Webs: Robustness Increases with Connectance. Santa Fe Institute Working Paper 02-03-013, 2002.