DITY™ Weekly Newsletter

The workable, practical guide to Do IT Yourself Vol. 6.35 • September 29, 2010

Creating Organizational Agility

By Charles Araujo
Vice President & Managing Consultant, Castlepointe. View Charles website at www.CharlesAraujo.com.

Business is moving faster and faster every day - and demanding that IT move with it. As a result, IT organizations are moving toward new development methodologies, approaches and tools such as SOA and Agile to reduce time to delivery of critical business applications. But as these same organizations adopt IT Service Management principles, an apparent conflict arises pitting SOA and Agile against Change and Release Management. Do Service Management principles stand in the way of rapid development and business responsiveness? They do not. In fact, an effective application of service management principles executed in concert with these approaches will offer your business customers everything they want: responsiveness, adaptability and reliability.

Whatever you think of Agile and SOA specifically, the move toward rapid development approaches and service-based architectures are no fad, they are the future. The rate of change for every business has increased steadily over the last two decades and shows no sign of letting up. To remain competitive, your organization must constantly adapt to changes in the market and changing customer expectations. None of that can happen without IT, and IT must be able to adapt and change with the business. Agile and SOA are just two of the most recent ways in which IT has attempted to respond to this need.

So whether Agile and SOA are "it" or get replaced by something else down the road, the general principles of service-based architectures and rapid development approaches are here to stay. That means that IT operational teams must learn to live in this kind of environment.

We're on the Same Team

Every organization must come to grips with two things; first, that Agile, SOA and Service Management are all different. Agile is a development methodology. SOA is an architectural approach. Service Management is an operational management framework. They are not the same thing, yet they are not mutually exclusive.

The second thing is to recognize that all three of these, while serving different functions, are all really after the same goal. Each is an approach that helps IT serve its customers more effectively, operate more efficiently and deliver value to customers more rapidly. We are all on the same team here.

Some people - particularly those on the service management side of the house - may react to this by asking, "But isn't Change Management all about controlling changes - and doesn't that mean slowing things down, not speeding them up?" The answer is an emphatic "no!" While **one** of the core objectives of Change and Release management is to minimize and mitigate risk, it is not the primary objective. The primary objective of Change and Release Management is to enable changes to the environment as rapidly as possible, but in a way that does not cause impact to the services being provided. Simply stated, a slow, lumbering, bureaucratic change and release process - even if it eliminates ALL incidents caused by change - is one that fails the business. So, in fact, the goals of Agile, SOA and Service Management are very much the same - rapidly delivering services (and changes to services) that drive value to the organization.

Bringing Service Design to the Center of Service Management

One of the fundamental reasons for this misperception is that many people look at Service Management principles from a purely operational perspective. While I believe that Service Management largely exists as an operational framework, an effective operational management approach must begin with a solid design of the service in the first place. This is the underlying intent of the ITIL v3 Service Design book and where we see Service Management, Agile and SOA really come together.

Designing a service effectively - or designing a change to the service effectively - is what will ultimately lead to a stable and effective operation. Without an effective design, the best we can hope for is to use operational principles such as Incident Management to minimize the damage - but it will be a losing battle. Likewise, if the service fails to meet the true needs of the business - or fails to adapt quickly enough to meet their changing needs - having a stable operation is meaningless.

To create sustained value for your customers, you must move past the view of Service Management being solely about operations and embrace a more holistic view that begins with Service Design. Moreover, that is where Agile and SOA become incredibly relevant. They (and other similar approaches) are the tools that IT organizations must use to ensure that a service will meet the needs of the business, to enable that service to be adapted quickly to business changes and to enable a sustainable and reliable service environment. It therefore becomes essential that these three disciplines are managed and executed as the means to the same end.

Creating an Agile & SOA Approach to Service Management

Integrating these three distinct disciplines is not always easy. It is convenient to put each of them in their respective boxes and to pretend that integration is not required. However, failing to take an integrated approach to the execution of these disciplines will leave your organization in a constant state of conflict. It does not need to be that way.

Integrating these three disciplines requires that all parties come together and realize that all of IT has a stake in creating an integrated approach centered on the rapid delivery of value to the business. From that context, all functional teams must realize that there will be "give and take" on all sides. With that foundation, a truly integrated approach can be achieved. To do so requires five core elements of integration: Perception, Discipline, Inclusivity, Rebalancing and Iteration.

Perception

To bring Agile, SOA and Service Management together, the first thing that must change is the perception that Change Management is solely about controlling risk, that Agile is the wild, wild west and SOA isn't relative to operations.

Perceptions - or more to the point, misperceptions - are the root of most of the divide between teams. Nevertheless, these three disciplines are really on the same page. All parties must commit to set misperceptions aside and come to the table seeking constructive solutions to bring these disciplines together. While all perceptions must change, however, the perception that Change Management is all about risk may be the biggest one to tackle. Change Management is really about successfully introducing as much change as is required to meet business needs, as fast as possible, while maintaining the integrity of the impacted services. The first step needs to be the breaking down of this mental barrier that sees rapid development and deployment approaches as the arch enemy of Service Management and vice versa.

Discipline

Misperception aside, some teams do look at a rapid development approach as an excuse to break through what they see as needless rules. Rapid development does not mean a free-for-all approach to development. In fact, you could argue that proper use of Agile and SOA approaches actually introduce greater degrees of discipline and planning. The reality is that discipline and planning are required to an even higher degree when you are in a rapid development state.

At times, however, development teams view this discipline and planning as somehow separate from operational needs. Agile and SOA teams need to understand that they must work ***together*** with Service Management teams to achieve three goals: to deliver value/meet specific business needs, to enable rapid adaptation and to maintain operational integrity. Development, Architecture and Operations teams **MUST take JOINT responsibility for achieving ALL three of these goals**. There is no option to divide and conquer here. You cannot split the baby into thirds with Development taking "value", Architecture taking "adaptability" and Operations taking "integrity." It just will not work. It must be viewed and executed as an integrated and shared discipline.

Inclusivity

If you have taken the previous point to heart and created an integrated and shared discipline, then the next step is relatively easy - creating inclusivity. Executed from the perspective of functional silos, it is easy to become insular during the development and adoption of any of the three disciplines. Development adopts an Agile approach; Architecture moves toward the utilization of SOA concepts and Operations " adopts" ITIL. It is easy to do it this way because the senior functional heads do not need to sell it outside of their domains. You do not have to build bridges. No enterprise wide communication is required. You are just focusing on doing your job a little bit better, right?

Except you are not. You need to create a shared, integrated discipline that will enable you to meet all three goals of delivering business value, enabling rapid adaptation and maintaining operational integrity. That cannot be achieved within a functional silo. You must move from a paradigm of exclusivity to one of inclusivity. You must look for (and create) opportunities to build integrated teams focused around solving specific business problems. That means including people from Operations on an Agile team. It means having development managers help lead service management efforts. You have to make sure that the level of participation is relevant and valuable, but you must ensure that your efforts are explicitly inclusive.

Inclusivity also means creating shared understanding. Development teams need to understand that Operations is fundamentally "on the hook" if something breaks in the environment and need to be sensitive to these needs. Operations teams must become aware that rapid development approaches bring increased susceptibility to environmental changes and require MORE communication and coordination - not less. Creating this level of shared commitment, understanding and empathy is a critical element of integration.

Rebalancing

Part of the reason that Service Management seems to clash with Agile and SOA principles is, frankly, because Change Management is so often broken. In many enterprise organizations, Change Management is a bloated bureaucracy. It ends up much more focused on documentation and an overly conservative approach to risk elimination than on creating agility and enabling rapid change.

Service Management teams need to take a critical look at these processes and rebalance them. They must include SOA and Agile team members in a redesign effort to identify ways to streamline the process and focus on controlling only what needs to be controlled. This will inevitably mean removing many current control elements. However, Service Management teams must come to the realization that, in most cases, these controls are mere placebos. They create the appearance of due diligence and risk management, but in reality, provide little real protection.

Perhaps the greatest need, however, during this rebalancing effort is the need to introduce a higher level of mutual trust. Particularly in rapid development environments, you will never be able to create a process framework that can account for every eventuality. Moreover, every attempt you make to encompass additional scenarios only adds to the bloat and complexity of the change process. Instead, you must focus on the critical control points and then trust your teams to adhere to the spirit of those controls. This trust must manifest itself as respect for the resulting process and, of course, must work both ways.

Finally, it's important to recognize that "rebalancing" does not mean simply cutting away. It means focusing on those things that must be controlled to balance business value, adaptability and integrity. In some cases, that can actually mean more control, not less. Agile, in particular, brings with it a need for careful (if dynamic) planning and stakeholder approval. A streamlined, but well-structured Change and Release Management process can provide an effective means to enable a rapid development approach.

Iteration

Stop the waterfall. One of the biggest problems with most by-the-book ITIL Change Management processes and policies is that they are linear. They expect a change to go through a structured, step-by-step development process that begins with a Request for Change, proceeds through planning, change approval, development, and finally to a production release.

This works great if you are using a waterfall development methodology, but breaks down quickly in a rapid development and service-oriented model. The reality in today's environments is that change and development rarely occurs that way. Even if your organization has not explicitly adopted an Agile or SOA-type approach to development and architecture, the days of long, "big bang" planning, development and deployment cycles are largely behind us. Business is simply changing too fast to allow for this luxury. So the Change Management process must operate in the reality that faces IT organizations today.

To do this, organizations need to employ a concept of graduated change management. Graduated change management is an approach in which the overall project is approved at a conceptual level and relevant stakeholders are identified. Stakeholder identification is critical, because it is to them that you will entrust the management of the change.

Then, working within the context of a specific development methodology (Agile variant), use the change control approaches embedded within it to manage detail-level change. This includes appropriately delegating change authority and managing the execution of changes within the broader, approved Change. Essentially, this amounts to extending change management into the development process itself. It goes back to the concepts of creating mutual trust and requires that your change management process be flexible enough to allow for it.

Changes delivered to production still need to be monitored and controlled, but the initial approval can act as a kind of specialized Standard Change. It delegates change authority to a project team, establishes the "rules of engagement" dictating the limits of that authority and places specific requirements on stakeholders and the project team in the execution of changes. This approach enables the project team to react fluidly and matches control requirements to the specific circumstances of the project to enable the rapid delivery of changes.

Summary

Changing the way you look at Agile, SOA and Service Management isn't easy. Creating an integrated and shared discipline that encompasses and spans functional areas in pursuit of common goals takes a lot of work. However, by doing this hard work, your IT organization can bring together the power of these individual disciplines and use them to speed service delivery and deliver exponential value to your customer.

References:

- Chou, David. "SOA Change Management Strategies". MicrosftSoCalArchitectBlog. April 22, 2008. September 13, 2010 http://blogs.msdn.com/b/socalarchitect/archive/2008/04/22/soa-change-management-strategies.aspx
- Morgenthal, JP. "Using ITIL v3 as a Foundation for SOA Governance". InfoQ. February 5, 2010. September 13, 2010 http://www.infoq.com/articles/itil-v3-soa-governance
- Swabey, Pete. "Agile and ITIL are Complementary Partners. InformationAge. February 12, 2010. September 13, 2010 http://www.information-age.com/channels/management-and-skills/perspectives-and-trends/1149583/agile-and-itil-are-complementary-partners.thtml
- Woodill, Chris. "Implementing an Agile Change Control Process". Chris Woodill: Diary of an IT Innovator. July 4, 2007. September 13, 2010 http://chriswoodill.blogspot.com/2007/07/implementing-agile-change-control.html