



Creative Shorts: Business agility through Agile lifecycle practices

12 strategies for delivering end-to-end agility

A Creative Intellect Consulting Shorts Report Series (ALM)

In this report from our Creative Shorts series on Application Lifecycle Management, we propose a strategy for employing Agile lifecycle practices to ensure end-to-end agility for IT application and service delivery.

Bola Rotibi, Research Director, Creative Intellect Consulting

Ian Murphy, Principal Analyst, Creative Intellect Consulting

December 2011

Shorts Overview

Agile has been seen as a methodology for software development for some time now. However, this ignores the ability of Agile to provide a lifecycle process that goes beyond core software development to address the application and service lifecycle: from business inception and management through to development, release, operations and to final retirement. An Agile-based approach is just as effective inside operations and, for those looking to move to the Cloud, as a method for building, deploying and managing Cloud-based services. But the disconnect and mismatch in Agile practices for all interconnecting and dependent services and processes is creating Agile bottlenecks that are threatening to prevent the full realization of agility across a business. Ultimately, what is at stake is end-to-end business agility.

A history of Agile success

Time to market is a concept driving a lot of decision-making and software development inside IT departments. A key driver for this is IT trying to deliver solutions as close to the speed that business units are demanding them. Too often however, software owners and clients fail to get the features that they want and expect in this rushed time-frame.

One of the failures of software delivery is that it is not inclusive. Users are often only involved in initial requirements acquisition, whilst the testers and Quality Assurance team, along with operations, only get involved much later in the process. With time-to-market and delivery pressures, testing and quality assurance processes that are executed after code is developed often get squeezed. This increases the risk for quality vulnerabilities to get overlooked until after the application or service is deployed into production.

Without a proper framework, process and workflow, there cannot be a stable and adaptive lifecycle to ensure what is needed and wanted, can be delivered safely and at the right quality.

The advantage that Agile brings to the table is the delivery of short controlled projects using multi-disciplined teams with an emphasis on software owner or client involvement and collaboration throughout. This means that all stakeholders are actively engaged whether it be in software inception, development, delivery or the operational management of a solution. It also helps to smooth over traditional organizational boundaries between teams, ensuring that time and money are focused on providing a solution that benefits a broad group of stakeholders.

Unquestionably, Agile development practices have delivered successes with regards to improved client satisfaction and the timely release of better quality software code. Agile success has driven more and more organizations from multiple industries to look to Agile to transform their processes and engagement practices.

The demand for an Agile Lifecycle approach

While Agile has a proven track record with software development teams, the positive impact of Agile outside development has been more limited. IT organizations can do much to extend the principles and benefits of the Agile Manifesto to upstream and downstream business processes. An Agile lifecycle approach can look to support:

- **Consistent agility across the application lifecycle:** Although Agile development practices can improve the speed and quality of code, it does not automatically translate to faster delivery of the developed solution or time to market. Getting to market quicker requires agility throughout the delivery process. This means extending agile practices beyond the software development process to include and address all interacting and dependent processes such as operations, legal, support, manufacturing and business analysis where appropriate. Every step in the lifecycle process can be a bottleneck. Unless agile practices are introduced from end-to-end, Agile will just shift the bottleneck to another point in the lifecycle.

- **Smoother handover between development and operations:** The first time the operations team often sees an application is at the handover between development and operations. This means that issues surrounding deployment scripts, infrastructure resource requirements and integration with other systems are ignored until the point of handover. Agility in development not only breaks down at the development and operations interface but also during the feedback process. By having operations as part of the project team, such issues can be considered earlier in the development process and resolved ahead of deployment into production systems. It also allows for tighter alignment between help desk support and the development process to improve problem resolution.
- **Establishing solid controls for Cloud service development:** One of the challenges of a Cloud platform is that there is no scope for errors in services being delivered. Taking all the elements of Agile identified above, it can be seen that controlled delivery, integrated continuous testing, operational issues and change management processes can be applied to Cloud services to ensure a stable delivery platform for customers.
- **Applicability and alignment with other methodologies where appropriate:** One of the focuses for Agile is an ability to support change. Therefore Agile offers a value proposition that can be useful when used within more complex process methodologies and cross-organizational execution strategies. After all, a formal methodology such as Waterfall might be the right approach for delivery but not the way one needs to work.

Changing the software delivery and deployment processes inside an organization is challenging, yet change is needed to meet the current demands of business teams. Agile practices focus on breaking down tasks into self-contained, manageable chunks – removing the “leave it to tomorrow” approach to deliverables. At the same time, Agile methodologies require all the stakeholders to play an equal and important role rather than be bit players on the application and service lifecycle stage.

Agile Lifecycle in a nutshell

The Agile lifecycle looks to address agility across the entire application or service lifecycle. More specifically, it:

- Embodies the core principles of delivering and deploying **working** software code that is aligned to both functional and non-functional requirements within the constraints, and capabilities of the operational environment.
- Calls for better engagement between all the lifecycle stakeholders and the software customer and owner. This in turn will lead to better quality outcomes that are expected along with an ability to respond to change quickly.

Many believe Agile practices shy away from process intensity. Not so. Agile practices in general are geared towards delivery and stakeholder engagement and interaction. However, core to Agile principles is a reliance on process discipline with a greater emphasis on interactivity through collaboration, communications and focused time-boxed delivery.

Agile practices applied to one part of the application and service lifecycle are pointless if not addressed from the whole lifecycle approach. This means understanding and addressing the impact of agile practices to all interconnecting and dependent processes such as legal, operations, support, etc. to avoid bottlenecks elsewhere in the lifecycle and improve the overall time to market. The Agile lifecycle requires an alignment of process practices employed across stakeholder teams and roles to ensure consistency in goals, outcomes and agility.

Ultimately, the Agile lifecycle needs to be:

- An end-to-end approach that draws from multiple related Agile methodologies to address the different levels of organizational and business requirements, team dynamics and software process maturity.
- An approach with core principles that must extend beyond the development phase to incorporate the interactions of software business management and inception, application and service development and delivery, operations and ongoing maintenance.

Crucial to the success of the Agile lifecycle is the ability to support and drive agility through the handover points and the interfaces between the core phases of the lifecycle. Out in the industry today there is a focus on bridging the development and operations divide through automation and alignment of common practices and processes. This is especially the case at the core interaction points of change and configuration management, testing and Quality Assurance, release management, problem resolution and help desk support. It is a strategy that works towards meeting the objectives of the Agile lifecycle at the handover points.

Agile Lifecycle foundations

The Agile lifecycle relies on a number of common core agile characteristics:

- **Quality delivery:** It is important to address the consistency of quality across the lifecycle in the face of time-to-market challenges. Significantly it will be an ability to test early and often that will be crucial to quality delivery. It will require the support for test driven development and deployment strategies, production like test platforms and tooling. Paramount will be considerable levels of automation throughout the lifecycle process to foster the environment of trust required to maintain agility across the lifecycle.
- **Process alignment and orchestration:** Agile execution requires process management and orchestration to ensure clear adherence to Agile goals across all interconnected lifecycle practices. Bringing all relevant stakeholder roles and teams into the Agile activity stream raises the profile and need for process orchestration. Executing process management effectively across the application or service lifecycle is being able to address the multiple levels of process and workflows that impact end to end delivery, management and ongoing maintenance.
- **Adaptability:** Few organizations enjoy the luxury of operating in completely greenfield scenarios, uncluttered from legacy constraints and regulations. Almost all organizations have to deal with established processes. The Agile lifecycle must be based on realism that reflects the dynamics of teams and the organizational environment and culture. It must also support the ability to change and evolve as implementation and execution lessons are learnt. Catering for a level of process reassessment and evaluation will help to deliver continuous improvement and alignment.
- **Applicability:** The Agile lifecycle does not preclude the use of other process methodologies that may be judged to be more appropriate for specific execution practices. Most notable is the supporting role that Agile practices can play within formal methodologies such as Waterfall and the value that an Agile lifecycle approach can bring to the overall delivery schedule and quality levels of more complex projects and systems. The Agile lifecycle is about adaptability and applicability. The most successful Agile organizations understand how to frame the scope of Agile applicability and determine where the best fit for Agile is.
- **Automation:** A level of automation is required in the face of complex deployment environments and intricate interaction and update processes (as in the case of server and systems patching). Without a level of automation, regular Agile updates into production will not be sustainable.

- **Governance and leadership:** Governance is vital for delivering the necessary levels of control, transparency, traceability and management required to maintain Agile goals. Discipline is a core value proposition of governance. It is required to ensure that Agile practices continuously deliver quality working solutions that are in-line with client expectations, changing needs and operational infrastructure. Collaborative engagement amongst Agile stakeholder teams will help drive down unnecessary bureaucracy even in the face of well-established processes and practices. Leadership is required to guide, support and ensure results that will encourage steadfast commitment to Agile processes.
- **Realistic and pragmatic delivery schedules:** A core principle of Agile is timely delivery. This can only be achieved by understanding the real capabilities of individuals and teams, which can then be reflected in realistic Agile iterations and delivery schedules.
- **Metrics and measurements:** These will help to provide the benchmarks for improvement along with the historical trends that will provide insight, predictability and support risk analysis to avoid future pitfalls. Burn-down charts are not sufficient – metrics and measurements must be put in place across the entire lifecycle.
- **Succinct requirements:** Organizations should focus on use cases and scenarios in order to determine the most appropriate and relevant functional and non-functional requirements for design, development and deployment. Focusing on the use case will allow better business process and business outcome alignment.
- **Stakeholder participation, team ownership and accountability:** Organizations must focus on partnership and teamwork across all stakeholder parties. Agile success relies on the characteristics of individuals and teams, especially in their ability to self-organize, collaborate, communicate and engage. The team makeup and culture is therefore critical for successfully meeting delivery targets and resolving issues quickly, all of which help to drive motivation and a virtual circle of well-being and active engagement.
- **Controlled time boxed iterations and stakeholder participation to improve transparency:** With Agile there is a fixed length short iteration with the allocation of tasks and resources to meet the deadlines. Any slippage should result in a task being dropped and then considered for the next iteration, which will allow change management and delivery schedules to be kept. By using short iterations with members drawn from all the stakeholders, transparency becomes the norm not the exception, and failure is no longer the predicted or expected outcome.

All of the above in unison leads to promoting the level of trust that must exist to drive agility throughout the lifecycle of end-to-end engagement and interaction. Without strong transparent support for all the criteria highlighted, Agility across the lifecycle will be curtailed.

Guidance strategies

It is important to ensure that the existing application and service lifecycle processes are compatible with the goals of Agile and that there is alignment. Breaking down the barriers and silos that exist between teams and stakeholders in order to improve the governance, delivery and quality of software solutions that better meet the goals and requirements of the business in a timely manner are the core principles of application lifecycle management. In this regard, they mirror many of the core tenets of Agile.

The Agile lifecycle is an approach that does not require a "big bang" implementation. It can start with a single project where the team structures, iteration length and management processes can evolve without destabilizing existing projects. But it does require a commitment from all parties to the improvement of application and service lifecycle management processes.

People and processes:

An Agile Lifecycle requires you to assess where and when Agile is appropriate, especially as the complexity of teams, systems and solutions increase. As part of deploying Agile, it is important to establish what the goals are – whether reducing the time to market of applications, allowing IT to provide business with a competitive edge, or promoting innovation in software design and development. The short iterations of Agile enable organizations to try something different in a controlled and manageable manner should it not deliver expected results.

More specifically you need to address the following:

1. **Process realignment and reassessment:** Existing processes have been designed with current tooling in mind. Review how the processes are designed and ensure that they exhibit the goals that Agile brings. The speed of Agile is often an opportunity to review heavyweight processes and make them more dynamic. This has to be balanced against good governance, but the goal here is controlled speed not reckless abandonment.
2. **Recognize the value of process orchestration and management:** Process facilitation, orchestration and management support are crucial in managing the interactions between teams and individual stakeholders that must happen to ensure the smooth progress of the application and service lifecycle. It is important to facilitate an Agile lifecycle that makes sense for the organizational environment and the business domain. For more insight see our Creative Shorts report “*The business value of ALM governance through process orchestration and management*”.
3. **Good team dynamics:** One of the key elements of an Agile approach is the development of self-starting and self-managing teams drawn from all the stakeholders of an application and service lifecycle. One can reduce unnecessary bureaucracy by ensuring that all team members are actively engaged on the project from the outset.
4. **Review established approaches to application development and deployment:** While many processes seek to bring the help desk into the development process, Agile treats all bug fixes, enhancements and new features as equal value, which ensures that the help desk is not out of sync with software development. Create a single process for development by removing the distinction between bugs, enhancements and new features. Build an environment where changes in corporate processes (such as a change in legislation) can be easily addressed within the Agile lifecycle of future software development projects.
5. **Educate Agile principles and processes:** This is important particularly at key handover interfaces of the application and service lifecycle. Ensure everyone understands what type of Agile your business is following. Doing so will help drive better engagement and trust throughout the organization and foster a more sustainable Agile lifecycle.
6. **Promote and embrace the value of discipline and governance:** Lack of discipline will lead to Agile “fatigue” and a feeling that Agile is an excuse for others not to do their jobs properly. Agile requires disciplined execution and cannot be an excuse to circumnavigate failing or overly complex processes. Foster a disciplined approach across the enterprise to software development and delivery, so that what is built is used rather than shelved as outdated and unusable before deployment.
7. **Look to introduce “Agile for Operations”:** Agile development practices are well established and garnering wide spread support and adoption. However, the processes, tooling and education support for maintaining agility at the interfaces particularly between development and operations are only now taking shape. What is still required is greater direction for “Agile Operations” practices to enable the operations team to maintain and continue the agility drive. Our Creative Commentary report “*The business value of a Developer Operations bridge: Realigning and strengthening the development and*

operations bond through connected ALM and ITSM” provides further guidance for those looking to understand the agility connections between development and operations. Enshrine testing into the development process where the tests are not just functional but operational.

Tooling strategy:

Increasingly, integrated development environments and project management software tools are providing support for Agile approaches. More specifically the following should be considered:

1. **Engage built-in workable Agile support:** Make sure your tools have built-in support for Agile practices. This will enable it to be part of the working routine for key participating roles such as developers, testers and QA staff, project managers, operations and help desk staff, rather than something bolted on to the side of existing business processes.
2. **Elevated Team support:** Team dynamics and size are very important when looking to Agile practices. Ensuring all teams and stakeholders can easily collaborate, communicate and participate is critical no matter the methods chosen to enable this. That said, you must be able to connect geographically located teams and individuals, and enable them to participate in activity and communication streams. Look for automation and tools that support and address social interactions and networks.
3. **Integrated testing, development and operational capabilities:** Making testing continuous means more than just advanced unit tests. Examine existing test tools and see how they can be made to provide continuous support from the outset rather than as a sequential step in a longer process. Operational data such as data center design, infrastructure constraints, help desk reporting and change management processes can be brought into the Agile iteration to make it more effective in delivering ready-to-run solutions.

Business engagement:

Where and how you implement Agile practices should be guided by the core objectives and priorities of your business and senior management team. Ultimately the perception and reality of agility must be towards generating business value. This can be supported by:

1. **Metrics and validation proof points are important for supporting the business value case:** Create transparency in the processes so that the business can see where it is spending money and change its priorities if that spend is not delivering. Provide a basis on which financial management of IT assets can be based, rather than have software development sitting outside of asset management and control.
2. **Dispel any myths and negative perceptions:** This will be crucial to paving the way for Agile acceptance across the management layers and by all relevant stakeholders. Removing any passion from the debate will be vital for helping to clearly present the business value and benefits and for building the trust with the business.

About Creative Intellect Consulting

Creative Intellect Consulting is an analyst research, advisory and consulting firm founded by Bola Rotibi, an experienced and renowned expert analyst in the field of software development, delivery and lifecycle management processes, technologies and tools.

The company's key areas of analysis are software development, delivery and lifecycle management across the Software and IT spectrum along with their impact on, and alignment with business.

At Creative Intellect Consulting we combine our independent technology and supplier expertise with our knowledge of enterprise needs to help vendor clients:

- optimize their market propositions
- position themselves against the competition
- validate their solution definitions and roadmaps

For end-user organizations, our research reports and consulting services help to make technology choices and implement strategies that are right for them whilst maximizing the value from existing software investments.

We pride ourselves on our accessibility, flexible approach and high levels of knowledge and experience.

Read more about our services and reports at www.creativeintellectuk.com