When Agile, DevOps and Lean Aren’t Enough: The Missing Imperative in Software Delivery
EXECUTIVE SUMMARY

Lean, Agile and DevOps principles have improved software delivery in many important ways. As evolving markets compel software organizations to increase quality and productivity, interest in these concepts has never been higher. But despite their considerable merits, individually and collectively, they can fall short in addressing the full range of issues confronted by modern software organizations. To close the gaps, these principles must be knitted together in a way that allows you to realize the full potential of your people, processes and technologies. The fastest and most effective way to do this is to create an integrated software lifecycle. From planning, development and testing to deployment and maintenance, integrating all the tools required to accomplish all the tasks related to software delivery will allow information to flow freely from practitioner to practitioner and across the entire team. This makes the extended team more effective and provides much-needed project visibility. Software organizations that meet this missing imperative can experience substantial gains in quality, speed, and traceability.

We have “done” Agile, now what?

DIFFERENT ROUTES, SAME DESTINATION

At the risk of oversimplifying, Lean principles are rooted in operations. Largely adapted from the auto industry, they focus on processes to improve quality, accelerate cycle times, eliminate waste and defects, empower teams, and promote continuous learning. Agile methods share these goals, but are more concerned with individuals and their interactions. Agile encourages a flexible and decentralized approach to software delivery, in which team members respond to changing environments through ongoing feedback and collaboration.

DevOps principles go further by placing an even higher value on communication and collaboration between the development (“Dev”) and operations (“Ops”) teams throughout the project lifecycle. They also stress the importance of integration and automation in any effort to deliver better software faster.

Clearly, there is a great deal of overlap among these three. They share a certain irony as well. In theory, each of them strives to streamline and simplify software delivery by providing a clear set of principles to improve performance. But in reality, things today are more complicated than ever.

TOOLS, TALENT, TIME AND OTHER COMPLEXITIES

To understand how software delivery has grown more complicated, just look at the variety of technological and economic pressures the team is under.

The recent past has brought an explosion in the number of technologies, tools and deployment paradigms the team must support. Users are no longer satisfied by ponderous client-server applications, released on an annual cadence. They demand nimble cloud-based or mobile apps, available everywhere and enhanced regularly.

Cycle times are getting shorter too; the good old days of semi-annual, quarterly, or even monthly release cycles may soon be a distant memory. For instance, Netflix deploys new software hundreds of times per day, not entire applications of course, but typically new versions of APIs. This model – faster delivery of more modular applications – is clearly where the industry is moving.

Then there are new complexities of a more existential nature. Ask yourself, for example: what is a bank? Is it a building down the street with tellers inside? Is it an app that lets you deposit checks and manage money? Is Apple now a bank? Is Google? Today, just about every industry faces some degree of software-driven disruption. What is a hotel? What is a taxi? What is a TV? What is a college? Not long ago, these were easy questions to answer. Today, not so much. Fact is, software has unprecedented power to accelerate an organization’s success or demise. Are you feeling the pressure?
THE MISSING LINKS TO YOUR TRANSFORMATIONS INITIATIVES

To counteract some of these complexities and meet the challenges they present, organizations have adopted one or more transformation initiatives, such as Lean Software, DevOps or Agile methods. But despite investing in training their people, rolling out new tools and adopting new methods, these organizations still find they are not reaping all the rewards promised by these initiatives.

What’s missing is a lifecycle view of software development and delivery, the notion that a project’s success depends on the entire extended team working in unison. Trouble is, the proliferation of software development tools for each discipline in the lifecycle has actually created an environment where the tools themselves are separating colleagues from one another and preventing an integrated lifecycle view of the project. Each tool does an excellent job of enabling the discipline and methodology it serves. Business analysts and product managers are made more effective with modern requirements elicitation and management tools. Testers have test and defect tracking tools. Agile planning tools help teams manage their work. But the artifacts that each of these disciplines create, and that these tools manage, are meant to be shared, not locked up in the individual tools. Unfortunately, because these tools are not integrated, the artifacts created within them stay within them. What is Lean, Agile or DevOps-like about that?

INTEGRATION IS THE ANSWER

Software Lifecycle Integration is an emerging discipline for the next generation of software organizations. It integrates the disparate application lifecycle management tools used for project and portfolio management; test and defect management; issue tracking; requirements management; Agile planning; help desk support and others, into a cohesive whole. Artifacts flow from practitioner to practitioner. Lifecycle data is collected to create comprehensive cross-discipline reports.

Finally, the artifacts created by each discipline are automatically liberated from their tool silos. And the people who created them are liberated from using email, bulk exports into spreadsheets and endless meetings to share information. It’s precisely what is needed to support transformation initiatives.

- Leaner Lean. When artifacts flow easily across tools, critical project information is no longer fragmented or locked up in closed silos. Rather, it moves through the system
automatically, giving practitioners the data they need when they need it. This helps eliminate the waste and repetitive tasks that drag down the delivery process. Much less time is spent on non-value-added work, which in turn helps decrease cycle times and project costs.

**Greater Agility.** When collaboration happens within the work, rather than in email or disconnected tools, team members can quickly provide feedback in context, based on real-time data. Reports and analytics emerge from on a holistic view of all artifacts over time, so practitioners across the organization can respond to any situation in the most appropriate way.

**More Connected DevOps.** In an integrated lifecycle, collaboration connections between development and operations become automatic. For example, handoffs between Development, QA and Operations are made efficiently in real time. Trouble tickets flow in real time into planning and development activities. And traceability and compliance become an automated by-product of the process, not an additional set of manual processes.

**AN INFRASTRUCTURE FOR REAL TRANSFORMATION**

Whether based on Lean, Agile or DevOps principles, any initiative to improve software delivery will involve a combination of people, processes and tools. By integrating your lifecycle, you can address all three dimensions in unison, and provide a platform that enables real transformation to happen.

In coming years, customer demand for better applications, more features, and faster cycle times will only increase. You can’t waste time on bottlenecks and impediments. You can’t waste money on unnecessary work. You can’t have errors falling through the cracks. Software Lifecycle Integration will allow your teams to collaborate much more effectively than they can today. It will maximize your investments in talent and tools, and vastly improve the analytics of your software delivery capability. It will bring a new level of visibility to the entire process, and automate compliance and traceability. In doing so, it can dramatically improve the quality and productivity of your software organization.

**ABOUT TASKTOP**

Tasktop connects the network of tools, teams, disciplines and processes required for planning, building and delivering software at scale.

The backbone of the most successful Agile and DevOps transformations, Tasktop is an easy-to-use, scalable and reliable tool integration infrastructure that helps enterprises connect, visualize and measure software delivery value streams.

With the ability to support hundreds of projects, tens of thousands of users and millions of artifacts, Tasktop automates and traces the flow of work from customer request to finished software product.

For more information about Tasktop, please visit [http://tasktop.com](http://tasktop.com)
To learn more about how Tasktop can help your business email us at: info@tasktop.com or call us at: +1 778-588-6896

READ OUR LATEST WHITE PAPERS

- The Hard Truth About Integrating Software Development Tools
- Understanding Software Development Productivity from the Ground Up
- When Agile, DevOps and Lean Aren’t Enough