

The New World: What do Agile and DevOps mean for ITSM and ITIL®

By Charles T. Betz and Christoph Goldenstern



You'd have to be living under a rock to have missed the impact of Agile and DevOps on all things IT lately. From startups to the largest enterprises on the planet, Agile and related techniques are transforming how IT is planned, built, delivered, and operated.

What does this transformation mean for IT service management professionals and their preferred framework—ITIL? Much. DevOps has changed the conversation in unexpected ways. For example, it was long assumed that change was the enemy of stability, and so organizations opted for infrequent, “well-planned” releases—which never seemed to work that well.

Then along came DevOps. “10 Deploys a Day at Flickr” was the first rallying cry during 2009. Surely, its systems must be crashing constantly? No, they weren't. When Continuous Delivery is well understood and performed correctly, systems stability improves. Only for Silicon Valley startups, right? During September 2016, Barclays Bank stated that the more frequently its 800 Agile application teams deploy,

the more stable its services. At all scales, it's clear that smaller, more incremental changes to complex systems are lower risk and promote stability. In addition, the fast feedback of those small, incremental changes enables a new culture of learning based on testing hypotheses by bringing (in Lean Startup terms) Minimum Viable Products quickly to the customer.

What's occurring and what might this mean for the established enterprise versus a start-up organization? One way to understand the impact of Agile and DevOps is through a scaling or "emergence" model. The trouble with frameworks, such as ITIL and COBIT, is that they are presented at an enterprise scale. The framework may state that it should be adapted to the needs of the particular enterprise; but exactly how to do this is often left to consultants. What works for a large enterprise may not make sense for a start-up. Verne Harnish in the book, *Scaling Up*, observes that there are natural clusters of firms at certain sizes:

- 1–3 employees
- 8–12 employees
- 40–70 employees
- 350–500 employees
- 2,500–3,500 employees

The scaling process can help us understand current debates in the industry, such as "DevOps versus ITIL." Think about IT processes in these terms. Would you recommend a full-blown change management process for a 10-person firm? Could you run a 3,000-person company without one? At what point would you introduce one, and why? What other processes would you introduce and when?

Agile works well in smaller contexts. It is team-oriented, and companies of all sizes increasingly are realizing that the collaborative team is where value is produced. Well-established research has shown that collaborative cultures outperform all other cultures (including competitive cultures). A 10-person company is a team, but a 50-person company must think of itself as a "team of teams." The question is how do we provide "the glue" for all those teams so we don't lose alignment. The more "loosely coupled" we are (in Spotify's engineering culture terms) the more we need to be "closely aligned" with common approaches that facilitate collaboration and problem solving.



This may seem obvious, but as companies scale up, the pattern has been to specialize according to functions:

- Marketing
- Research and development
- Sales
- Operations and service
- Back office (Finance, HR, IT)

In addition, there are sub-specialties within each function (e.g., IT specializes further into applications and infrastructure teams; infrastructure teams specialize into server, storage, networking, 24 x 7 NOC, and so forth.)

IT organizes itself as an “order taker,” both in its relationship to the business and internally. Application teams submit “tickets” to the infrastructure team for needed resources, for example. This model can produce IT systems and services that are reasonably stable, but they are often slow to deliver and slow to change. Functional silos versus end-to-end-process thinking is the norm, which is a bit ironic because that’s not what frameworks like ITIL advocate.



Today digital transformation is challenging and disrupting silos. As market-facing products contain increasing amounts of information technology, “back office” IT converges with research and development and general operations and service. Now that IT is critical to a company’s survival, it is required to be more responsive to market needs. Stability is still required, but stable systems that don’t satisfy fast-changing market needs are worthless.

Functional silos require handoffs. Handoffs cause delay and slow responsiveness. Functional silos tend to develop an “us-versus-them” attitude towards the teams they are servicing, and from which they are requesting services. That is why Agile methods promote multi-skilled teams: as Marty Cagan says in his influential book, *Inspired: How to Create Products Customers Love*, the team minimally needs to be able to drive a product towards three necessary qualities:

- Is it valuable?
- Is it usable?
- Is it feasible?

A team that can drive outcomes in alignment with these three dimensions can be called a “full-stack”. Scrum and other Agile methods repeatedly emphasize that the team must be able to operate in general, on its own, with minimal external dependencies and blockages.

Another current practice is “you build it, you run it.” This is a good practice and a big change from the old days of “throw it over the wall and run,” when developers took little responsibility for writing software that could actually be run in production. Essentially, the emphasis moves from a vertical IT “factory model” to a more “horizontal management” approach. This is where the team has end-to-end responsibility, including some of the more traditional ITIL disciplines of Incident and Problem Management.

As Amazon CTO Werner Vogels famously said, “Giving developers operational responsibilities has greatly enhanced the quality of the services, both from a customer and a technology point of view.” Now, developers increasingly “wear the pager,” and are incentivized to write software that is stable, scalable, and operates well, in addition to meeting the user’s expectations for functionality.

These team-based approaches have been shown to work remarkably well, which is why organizations, large and small, around the world are hurrying to adopt Agile and DevOps. However, at the “team of teams,” large organizational levels, communication and collaboration must cross teams. We can try to

minimize the need for such communication, but at some point, how do you know two changes won't collide? Cross-team processes to coordinate and synchronize activity, need to quickly focus on the critical pieces of information that are vital to operations and that provide a minimal, but essential quality check (e.g., the incident or problem statement).

A common approach for issue resolution across the teams, removes some of the barriers between incident, problem and change management. When everyone "speaks the same problem solving and execution language" it minimizes the "dead time" of ineffective or repetitive activity and improves the way data is used and shared.

Change management

Because ITIL has long advocated a rigorous change process, it has become an obstacle for many Agile and DevOps advocates. Yet slowing the throughput of changes (which ITIL Change Management tends to do) is not correlated with systems stability.

Now, in fairness to ITIL, continuous updates to an application or service whose platform is stable in general, are seen as "standard" changes not requiring discussion or approval. There is nothing in ITIL preventing this. The reality in too many organizations, however, is to "make the developers wait" by using a one- or two-week change-control cadence.

When operations engineers are responsible for making the required change to production, a change delay may stem from too much work in process not from any lack of cross-team synchronization (such as the use of a bi-weekly Change Approval Board meeting for assessing risk). However, as more teams operate on a "you-build-it, you-run-it" basis, having operations implement production changes is seen as non-value-add. Even the frequently-cited "segregation-of-duties" concern has faded. (See the DevOps Audit Defense Toolkit, co-written by DevOps evangelist Gene Kim and IT auditor James DeLuccia.)



Beyond change management

Beyond Change Management, how have Agile and DevOps teams experienced ITIL? Teams that manage operations, including the help desk function and 24 x 7 centers (which are two different services), tend to adopt ITIL training and terminology and have service teams operating as functional silos.

These silos are defended with comments like, "we don't have enough people to give every development team their own operations personnel or infrastructure engineers!" But this misses the point of modern cloud-based DevOps practices and overlooks important aspects of IT service management. ITIL advocates the establishment of Service Catalogs, which are often used to "front-end" infrastructure services. Historically, a Service Request Management process supports these services, often with manual work (e.g., an engineer analyzing a request for some new servers).

Cloud and micro services approaches are changing the face of Service Request Management with a consistent, catalog-based front-end and fully automated service. What is the Amazon or Azure Cloud portal but a service catalog with a high-degree of automation? Self-service and automation empower functional teams and free the infrastructure teams from most on-demand consulting and engineering services so they can focus on building and sustaining a shared, self-service infrastructure.

Moving to enterprise scale

What happens when an Agile mindset is brought to true enterprise scale? Beyond the need for “team of teams” coordination, there are problems with risk management, governance and more. Business continuity, problem management and major incident response become critical concerns. It's Kepner-Tregoe's view that major incident management, in particular, requires specialized skills that help ensure the enterprise against catastrophic damage and loss. This “stop-gap ability” to stop the bleeding when major outages occur requires specialists with a combination of both outstanding problem solving as well as facilitation and communication skills, due to the naturally high-pressure environment and the plethora of stakeholders to satisfy.



Furthermore, organizations can't afford—in this fast-moving environment—to continue to solve the same old issues. Introducing Agile and DevOps principles into an organization with an insurmountable backlog of open problems (and, therefore, rising incident volumes) is a risky endeavor. For Agile and DevOps to succeed, organizations need to start taking Problem Management seriously and dedicate resources to finding the root cause of issues. Feeding Problem Management back into the team backlog, on the same footing as new user “stories,” is an emerging best practice.

On the flip side, one risk of scaling up is when the organization implements so many processes that the all-important team experience is disrupted. Multiple processes more driven by the need for administration/documentation versus the value of their outputs can block team delivery and their cohesion and ability to deliver customer-value deteriorates. Kind of performance degradation is also an enterprise risk; possibly the biggest one of scaling up.

In conclusion

There is much that ITSM practices have to offer the new Agile/DevOps world. They provide an alignment around language and proven practices. Service catalogs, Change, Incident, and Problem Management all are relevant. Organizations should guard, however, against using ITSM as a rationale to emphasize structure and process over service outcomes, losing some of the original intent of frameworks like ITIL. A service-centric approach to user outcomes has long been a part of the ITSM philosophy, and service managers who keep that focus and have the ability to apply “quality thinking at speed” will continue to do well. At the end of the day, it's all about that customer experience, and their daily moment of truth when encountering your digital systems both in terms of quality and stability.

About Kepner-Tregoe

Founded in 1958, and based on ground-breaking research regarding how people think, solve problems, and make decisions, Kepner-Tregoe provides a unique combination of training and consulting services to improve quality and effectiveness while reducing overall costs. The KT methodology is used at every level of client organizations: to implement strategy, achieve continuous improvement, increase customer satisfaction, and drive effective issue resolution throughout the organization.

