Envisioning the Future of Software and DevOps
As with any form of IT methodology, once you sink your teeth into DevOps, certain expected and unexpected patterns begin to emerge, creating the opportunity to track performance, evolve practices and chart a path for future success.

Dramatically accelerating organizational velocity, specifically in the domain of applications delivery, presents tremendous opportunity to improve the user experience, as well as the potential to introduce bigger risks. Maturing DevOps dictates continuous analysis, from protecting against the proliferation of misguided practices or unbridled automation to the art of building models and metrics to measure progress. Of course, DevOps is all about continuous collaboration, experimentation and improvement.

In this e-book, DevOps experts from CA Technologies highlight some of the evolving best practices related to optimizing toolchain automation, further advancing cultural best practices and building those metrics that best detail how your application delivery efforts continue to mature.
When undertaking an effort as complex as re-architecting the enterprise to support agile and DevOps practices, the temptation to enact sweeping change is powerful, but must be tempered.

Such as with the mid-20th century trend of urban renewal that resulted in huge caverns of high-rise apartment buildings—those once considered modern marvels that eventually resulted in nightmarish living conditions—great care must be taken to ensure that today’s enterprise architecture isn’t constructed so quickly as to overlook potential outcomes.

Enlisting the same analogy in his blog, “Using DevOps to Avoid Architecting Application Slums,” CA Technologies DevOps expert Peter Waterhouse highlights the risks that practitioners face as they recast their architectures to support emerging best practices around applications delivery and support.

While developers may greatly contribute to the issue by rushing headlong into using tools and processes that make it difficult for architects to predict related requirements, organizations must focus closely on both ends of the spectrum to prevent either camp from going off in its own direction.

“Enterprise architects must recognize that any misalignment between their processes (especially the rigid and static ones) and the vision of an increasingly software-driven business will only further isolate the practice,” Waterhouse writes.

While emerging techniques such as containerization and cloud-based platform as a service (PaaS) offer incredible potential and a fitting complement for agile and DevOps initiatives, much like the cavernous housing projects of old, these low-cost and futuristic approaches should be applied with caution.

The model for conjoining enterprise architecture with agile and DevOps practices remains nascent; a careful approach is necessary to avoid potential pitfalls.
So, how do enterprise architects enable agile software delivery using DevOps without over-engineering or setting up too-rigid parameters in the process? Waterhouse suggests best practices, including:

**Avoid substandard or unsupported materials.**
With so many potential flavors of software and services at architects’ disposal, organizations must be careful to avoid anything too work-intensive or difficult to support.

**Never use new technologies to rehouse old problems.**
Creating a new home for an existing issue, or attempting to fix it with some new set of requirements, will likely just cause the problems to resurface down the road.

**Monitor and manage your subcontractors.**
Even when cloud or PaaS techniques offer a great opportunity, organizations must keep a close eye on retaining necessary control, such as via availability of open APIs.

As Waterhouse notes, aligning architecture to foster today’s emerging models may still be a work in progress, but by leveraging sensible methodologies, it shouldn’t be something that thwarts innovation nor sets the table for future chaos.
Avoid Introducing and Instituting Bad Practices

With the introduction of DevOps comes the opportunity to eliminate longstanding IT malpractices, and the requirement to avoid creating new ones.

In nearly every industry, professionals frequently bend rules to achieve results. The emergence of agile and DevOps techniques offers the chance to recast these practices in the software domain and avoid introducing additional issues.

In his blog, “DevOps and Deviance: How Bad IT Practices Become Accepted as Normal,” Waterhouse highlights the fact that, much like verticals including healthcare and aeronautics, IT practitioners have often been allowed to ignore certain warning signs when the task at hand appears to merit such exceptions.

Yet, many times this type of oversight, for instance within the domain of IT security where teams are dealing more alerts than they can feasibly investigate, such permissiveness has ultimately resulted in massive failures.

Enlisting DevOps practices, which promote a more comprehensive awareness of the entire applications lifecycle, presents the opportunity to remedy some of this behavior, Waterhouse writes. At the same time, increasing software factory velocity shouldn’t come at the expense of sticking to the rules.
How do teams ensure that longstanding areas of oversight and new workflows don’t become breeding grounds for “normalization of deviance?” To avoid such tendencies, Waterhouse recommends that IT pros review the same controls created in other industries, including:

- **Overlooking “stupid” rules.** Allowing for wholesale ignorance of inefficient requirements (e.g., testing) is bound to result in issues down the road.

- **The plausible deniability defense.** Refusing to understand difficult practices cannot be an accepted means for teams to ignore such requirements.

- **Don’t rock the boat.** New policies and tooling (e.g., containerization) affect perceived productivity, but merely refusing to get on board doesn’t work.

- **Big picture deferral.** Casting aside best practices in the name of driving toward organizational goals can’t be accepted as an excuse.

- **We don’t need badges.** Every team thinks it’s a special snowflake, but when everyone allows themselves exceptions, chaos is the likely outcome.

- **The silent treatment.** When employees have been taught that speaking up about problematic behavior is punished, best practices go out the window.

- **Follow the liar.** If management officials skew results or bend rules to suit their deliverables, others are sure to see this and replicate those practices.

Reviewing this list reveals how closely matters of DevOps and applications delivery success are linked with issues of sociology and management ethos. If you’re breaking all the rules to get the job done faster, at some point your team will face related consequences.
As DevOps best practices continue to evolve and mature, it’s important to reset goals to ensure that even these new methods don’t become stale or rigid.

The DevOps journey never ends, versus one that constantly changes in nature and desired outcomes over time; it’s all about fluidity.

Several years ago, most organizations’ main goal was merely to start using DevOps methodologies and tooling in specific pockets where it made the most sense. Today, an increasing number of practitioners are pushing hard to apply DevOps principles as widely as possible.

That’s an indicator of rapid change, amplified by experimentation and innovation—the core tenets underlying the DevOps movement at large—along with collaboration and automation. This means that organizations must frequently re-examine and reset goals, as well as related metrics.

In her forward-looking blog on this topic, “DevOps Resolutions for 2016,” CA Technologies DevOps expert Aruna Ravichandran takes stock of some of the most high-profile and pervasive measurements that today’s practitioners should employ to ensure their efforts are on track for success.
Among the most critical initiatives for organizations and management officials to emphasize in the current environment are practices and metrics including:

**Unify those KPIs.**
In the realm of key performance indicators (KPIs) for DevOps, measurements such as software release frequency, defect rates and mean time to repair (MTTR) are widely accepted. However, the culture of blame where teams are punished using these metrics doesn’t work; Ravichandran recommends “de-siloing” KPIs and placing more emphasis on collective responsibility.

**Cultural metrics.**
If DevOps is about culture first, then simply tracking applications delivery and defect metrics won’t tell the whole story. While those numbers indicate how people and processes are performing, creating measurements aimed specifically at tracing the ability of the organization to change and foster employee satisfaction can provide critical DevOps maturity data.

**Monitoring management.**
When creating said cultural metrics and holistic DevOps performance KPIs, Ravichandran notes that management officials must continue to look in the mirror and grade their own performance as part of the larger whole. Even if only assuring necessary DevOps engagement, such practices will result in a program with a stronger chance of overall success.

So, there you have it: some of the most important elements of DevOps assessment that every organization should incorporate, regardless of how deeply their efforts have advanced into the larger transformation.

By continually recalibrating indicators that are useful in measuring DevOps’ inherent nature of change, practitioners are far more likely to gain a firm understanding of where they actually stand or where they need to adjust.
According to the “2016 State of DevOps Report,”1 informed via direct input from today’s IT practitioners, organizations that have made the greatest effort to incorporate DevOps practices are realizing significant benefits and empowering digital transformation.

Nearly every section in the preceding chapters of this e-book references some notion of DevOps maturity or success. However, the best evidence of the extent to which enterprises are realizing those gains as real-world achievements is offered via targeted industry research.

As highlighted in her related blog, “State of DevOps Report: Reaping Results,” Ravichandran notes that the report—based on a survey of over 4,600 respondents—proves that high-performing organizations that have embraced agile and DevOps are massively outperforming those that haven’t.

“It reaffirms powerfully that so many of the promises made about DevOps over the last few years are neither hype, nor overstated,” Ravichandran posits. “The gap between the leaders and laggards is growing. Industry landscapes and business models are being redefined. This is critical transformational work that every organization must adopt to survive.”
Some specific, conclusive findings of the “2016 State of DevOps Report” include:

- High-performing IT organizations are deploying code 200 times more frequently than low performers, driving 2,555 times faster lead times, 24 times faster recovery times and producing 3 times lower change failure rates—all of which highlight significant acceleration of their overall velocity.

- Employees working at such organizations are more satisfied with their jobs. Based on Employee Net Promoter Score (eNPS), such workers were 2.2 times more likely to recommend their organization and 1.8 times more likely to advocate their team—which previous studies correlate to profitability.

- Related to ROI metrics, DevOps leaders spend 49 percent of their time on new work, versus break/fix or unexpected work driven by problems—giving them more time to build the next must-have application features, which impacts competitive standing and the bottom line.

In addition to these findings, the annual research project surfaces the effect that continuous delivery practices have in boosting applications delivery and quality measurements—notably, the growing use of test data management and related automation.

The numbers from the 2016 State of DevOps Report are tacit proof that DevOps is a valid, serious movement that’s having real-world impact.
Conclusion

Now that our trifecta of DevOps thought leadership e-books is complete, we hope you have a good feel for the current state of related best practices and a better understanding of where you stand today.

DevOps is all about being open to increased collaboration, experimentation, and ultimately, change. Clearly then, advancing your approach to DevOps over time should focus on continued learning—that’s the only way to practically accelerate applications delivery while ensuring optimal customer experience.

For more insights into the latest DevOps tips and best practices, check out the newly published book, *DevOps for Digital Leaders*.

Good luck on the next stage of your journey.