Taking DevOps to the Next Level with CMMI V2.0

Nearly all of today's business innovations are built on technology, which means an organization's ability to stay ahead of the competition depends on how quickly it can develop and deliver software and services. That's a key reason for the rise of DevOps, which merges development and operations practices to deliver applications and services at high speed. One report found that more than 78 percent of companies are already applying DevOps practices, and the number continues to rise.[1]

Since its origins with mobile app developers and internet companies, DevOps has rapidly spread through the world of commercial and enterprise software development. Now, the U.S. Department of Defense is planning to drive the use of DevOps practices among defense contractors as part of a sweeping revamp of the way it spends its multi-billion-dollar software budget. The DoD's move follows input from key advisory groups, including the Defense Science Board, which recommended replacing slow and outdated waterfall development methods with DevOps approaches such as a highly automated "software factory" that can quickly and predictably generate software that better meets requirements.

DevOps is a Culture, Not a Ruleset

DevOps is a culture and set of broad concepts and practices rather than a hard-and-fast set of rules or specific products. Besides the use of agile development techniques supported by automation, those concepts include greater collaboration during development, engaging stakeholders at all stages, continuously focusing on the entire system while developing each component, and – as the name suggests – moving operational tasks earlier in the development cycle.

What's sometimes not widely recognized is that applying DevOps successfully requires employing best practices for more than code development and deployment. DevOps success also demands best practices for all the other processes involved in delivering software and ensuring it meets expectations, such as integrating software with other products, managing interfaces, releases and updates, and monitoring performance and usage. That's critical not just to ensure a good user experience, but also because the information can be useful earlier in the product planning and development cycle. If customers aren't adopting your latest app, you need to know why – and you may need to reexamine your product roadmap to see whether

you need to change plans. Furthermore, few companies today focus solely on software development; most offer a blend of software and related services. So, they need an integrated approach that enables them to coordinate development and delivery of both.

That's why CMMI (Capability Maturity Model Integration) V2.0 is so important for ensuring the success of DevOps projects. A time-tested set of global best practices for continually improving performance, CMMI is built on many of the same principles that underlie the DevOps approach, including a broad systems approach, engaging stakeholders throughout the product lifecycle, and continuous improvement. It's also an integrated model that supports software, hardware and systems development – including the practices recommended by the DoD – as well as service delivery.

For example, the combination of CMMI Development and Services supports the DevOps "shift left" concept, which aims to improve software quality and catch and resolve problems sooner by including operational tasks earlier in the development life cycle than with traditional waterfall development. That's because CMMI integrates best practices not only for design and development, but also for operational aspects such as how to integrate software and services, deploy and update them, and measure whether they meet expectations.

CMMI also supports the critical task of automation, which is key to accelerating software production. DevOps teams use toolsets to automate cumbersome manual processes throughout the development cycle, from requirements to drive design, to generating test scripts, to creating user documentation. This automation represents a significant advance in software development: It frees developers from time-consuming routine chores so they can focus on the most important and challenging task of designing and building the software's functionality.

But reliance on automation makes it vital to select the right automation tools. Choosing a toolset simply because it's popular, or because it's suggested by a vendor, can quickly send a DevOps project off the rails. For example, a toolset may impose rigid working methods that don't align with your internal processes, so choosing the wrong tool may break your processes — or force you to customize the tool so extensively that you create a maintenance nightmare. You shouldn't have to change your business to fit the tool; your processes should be driven by your business needs, not by an external artifacts or systems. CMMI helps organizations avoid the pitfalls: It provides a straightforward, structured way to identify whether a toolset meets the organization's business needs and whether it will continue to do so in the future.

Even though DevOps aims to improve software quality, problems can and do slip through. Furthermore, the rapid, highly automated development process means there's less time to catch those problems before they reach users. Problem management and rapid issue resolution is as important as ever. If you release the wrong configuration, or your latest version introduces bugs, your users may lose service, your reputation takes a hit, and you need to scramble to fix the problem. CMMI helps organizations avoid those problems — by better controlling software configurations and releases, for example — and manage them when they do occur. To use a

highway analogy, CMMI V2.0 fosters a foundation of good driving habits but also includes protective measures, like sensors, airbags and guard rails, to ensure you get to your destination safely.

DevOps promises to help organizations innovate faster by accelerating the development of applications and services. But to make DevOps successful, organizations need more than the ability to produce code faster. By supporting all the processes involved in delivering and managing software and services, CMMI V2.0 helps organizations turn the promise of DevOps into reality.

---

[1] *State of the Cloud Report*, Rightscale; https://assets.rightscale.com/uploads/pdfs/RightScale-2017-State-of-the-Cloud-Report.pdf