Exploring Metrics in the DevOps Entry Point

Part three of a series that will identify key personnel for a transition to DevOps.  
Al DuPree  
Wed, 04/03/2019 - 10:10

Photo Credit: OnBlast/iStock

The first and second parts of this series covered how we’ve arrived where we are when it comes to system deployment in the enterprise. We talked about the need to see systems as a single image (hardware, software, processes, people) that owns the product. We also covered the need to evaluate how proficient we were as an organization through the adoption of a capabilities maturity model. This final part will identify personnel who are key to this transition and the key metrics that any DevOps organization should consider adopting.
Day 0: Adopt Agile and Prepare Your Evangelists

Having either renamed or established a DevOps center of gravity by renaming and reorganizing your PMO (or starting from scratch), it’s important to state the obvious: DevOps is a direct outgrowth of the Agile development methodology. As such, it is dependent on many of the practices and approaches found in Agile. Therefore, if your organization hasn’t adopted Agile as a standard organizational approach to software development, now would be the time. Agile will profoundly affect how you view, conduct and evaluate your IT projects. Starting from your PMO and working outward provides the ability to lay the foundations and manage the impact of that change as efficiently as possible.

That center of gravity delivers additional significant advantages for management, the most practical of which is that it makes it possible to nurture and develop a new role in the organization: the role of evangelist.

Back in the mid 90s, I had occasion to make a pitch to a venture capital organization funded by Palm Computing (makers of some of the early PDAs). Palm Computing established the role of evangelist in their company (the managers of evangelists were called, by the way, evangelical shepherds), which, as I understood it, they borrowed from Apple. Palm evangelists were sources of three things: unparalleled subject matter expertise, a passion for change and boundless energy. Which is where you start if your evangelists are going to instigate the cultural change on the scale that’s necessary.
It’s fair to say that PMOs (like most common IT functions) run the gamut, from organizations that routinely employ a rubber stamp to places where true scrutiny of project effort takes place. Regardless of where on the spectrum your PMO may fall, transitioning from what was a centralized command-and-control environment, with long project timelines, large deliverables and a limited set of stakeholders present, to (at least tactically) a more decentralized self-managed and self-organizing approach, centered on the concept of the sprint, with much smaller deliverables and many more stakeholders at the table is going to be a significant, if not seismic, shift. The evangelist role should, therefore, along with the responsibility of "spreading the word," assist with, guide and facilitate this transition, serving as the change agent that provides clear and unvarnished feedback throughout the transition.

**Speaking of Evangelists**

There is a specific set of skills necessary to fulfill the evangelist role. Business process analysts and engineers are uniquely suited for such roles. If you have a business process area, fold it into the newly named (or formed) DevOps office and charge them with proselytizing the newly adopted approach and serving as the judge and jury for how well and completely it’s being adopted. It’s relatively natural for such individuals to provide guidance to organizations when adopting new processes. While DevOps isn’t itself a “process,” the implications of adopting it are sufficiently like process changes that such skills will come to bear. Agile is very much a process and its implementation will substantially benefit from the presence of such professionals.

And the final tool that you need to provide your organization with is the means to consistently assess, measure and communicate progress.

**About Those Metrics**

DevOps metrics are available all over the internet. The following table includes some of the most common metrics associated with DevOps as well as several novel measures being proposed here.
Conventional Speed Metrics

Conventional Quality Metrics

Newly Proposed Metrics

Frequency of Deployment

Customer Tickets

Participation Rates

Time to Deployment

Defect Escape Rate

Representation Errors

Lead Time

Error Rates

Role Deficits

Deployment Size

Mean Time to Detection/Recovery

What we propose here is that there are three factors that determine when where and how you choose to make use of these metrics:

1. Where you fall in the spectrum of maturity for the CMM you’ve adopted.
2. What the goals and aspirations of the business that you serve are.
3. Your assessment of the volatility and competitive environment of the technology platforms that you presently depend on.

I’m going to reference the CMMI here as a proxy. The overall “phases” of maturity (again, loosely immature, maturing and mature) can be applied with any model you adopt.

**CMMI Levels I and Early II: The Immature Stages**
Levels I and II in CMMI are referred to as initial and defined (please remember these are only proxies, you can use whatever maturity model you prefer) and should be considered immature organizations. These are organizations that are working to gain and improve control of processes which are presently less than consistent in what they’re able to produce. As such, initial metrics in this phase of maturity should be basic building blocks for future success and provide insight that allows the organization to proceed.

Metric candidates for consideration: A critical component of the Agile approach is smaller deliverables and faster development cycles. The place to start in the immature organization is with measures of "frequency of deployment" and "time to deployment." These metrics at this point in the maturity cycle should be used more as benchmarks and tools for calibration in later phases. These are standard DevOps/Agile metrics (with links above for more detailed definitions).

At this point in organizational maturity, there should be a tremendous emphasis on the cultural changes that will fuel momentum in later stages. So it’s important to emphasize measures that focus on process participation. These are not “standard” DevOps/Agile metrics and require definition here.

Participation rate should be used to measure the number of development meetings in which all stakeholder groups are represented out of the total number of meetings. The Agile retrospectives should use this metric as a means of calibrating who needs to be in place at launch and who needs to be encouraged to return, and most especially who was included that didn’t need to be there. Representation errors should be used to track the number of instances in which teams fail to identify all requisite stakeholders or when a stakeholder is included unnecessarily. Finally, role deficits is a measure that is intended to capture instances in which teams may or may not have the requisite skills necessary to operate at the prescribed level. Role deficits can be derived by comparing the performance of teams over time, taking into account their composition and pinpointing groups or individuals who may account for differences in performance.

CMMI Levels Late II, III, and Early IV: The Maturing Stages
In the later stages of defined, throughout managed and the early stages of quantitatively managed (CMMI levels II, III and IV) allow for the introduction of an entirely new set of metrics and also support the use of previous metrics in new and more effective ways.

Metric candidates for consideration: From the early stages of maturing as an organization forward, frequency of deployment and time to deployment can be used for goal setting. These are no longer “benchmarks” to “see where we are” or what’s possible within our limitations rather they are used to accelerate the pace of activity to see how rapidly the organization can produce before quality starts to suffer. Rather than establishing the limits, these measures are now used to test and extend those limits. Initiating some measure of deployment size will provide additional input for refining and increasing the above. In line with notions of limit testing, measures of quality are introduced to determine the overall bounds of capacity. Included in measures of quality would be tracking errors. Note that errors include more than simply software bugs and failures but encompass everything from query timeouts to the myriad of production issues that we all know and love. Defect escape rates, that is the number of bugs found in production versus development/test also adds a level of precision and provides for the ability to pinpoint issues in development and QA.

In the later stages of a maturing organization levels of precision should increase substantially. Maintaining the metrics cited above and adding "mean time to detection and recovery" will contribute to that precision. Wrapping many of these metrics up (and adding additional features) and packaging them as service level agreements with the client organizations codifies relationships and advances the notion of accountability.

Maturing organizations must also consider the goals and aspirations of the enterprise as a whole as well as the constraints imposed by technology choices already made and those anticipated. Forward leaning enterprise, or those which find themselves in highly competitive environments will place demands on all the metrics cited above. Technology platforms which are subject to rapid or numerous changes because they operate in highly competitive markets or those platforms in which vendors have substantial market power to dictate to their customers must also be taken into account. This accounting is best expressed in the language of risk. It is well beyond the scope of this post to provide insight into how best to express and manage risk. But it is important to emphasize the following: an enterprise operating with competitive pressures which demand rapid deployment or
which includes excessive changes in product requirements or includes technology platforms subject to their own volatility must explicitly take into account and adapt to increased risks at every stage of the process. And the willingness to accept higher or lower levels of risk must always be explicitly negotiated by all stakeholders.

CMMI Levels Late IV and V: The Mature Stages

For those organizations at the top of the maturity hierarchy service level agreements, which encompass many of the metrics cited above are a necessity. Further I think it’s important to emphasize something that is too often missing in any discussion of metrics: qualitative measures.

William Delone and Ephraim McClean in 1992 authored a research paper titled IS Success Model, which has been cited in thousands of articles since. In it they identify the “six most critical dimensions of success” for information systems, one of which (customer satisfaction), is rarely (if ever) represented in methodologies that purport to deliver information systems. I would contend that DevOps/Agile approaches within highly mature organizations should undertake explicit qualitative surveys of user satisfaction with the products they produce and the methods used to produce them. They should, in turn, develop/derive quantitative measures from their findings. The voice of the customer when reduced to a series of numbers initially, will often fail to capture options and opportunities for innovation and improvement.

Last Thoughts

We’ve labored for years with the consequences of choices that were made starting in the middle of the last century. A change in approach is long overdue. Such a change should carefully reimagine the organization and how it applies its resources. Such a change should also separate the evaluation of a process, from the evaluation of the product such a process produces. And finally, such a change should, clear-eyed and cold sober, start out by determining what we can produce, instead of tasking us with producing what we can’t.
DevOps and its underlying support, Agile, are effective breaks with the past. Pairing DevOps and Agile with an appropriately selected capabilities maturity model is a solid step toward the future.

This was adapted from an original blog post by Al DuPree, who is an information systems professional with international experience in strategic planning, software application development and delivery, data center operations and information technology related service delivery. His past experiences include chief of innovation and solutions division at the National Institute of Standards and Technology and CIO at the Congressional Budget Office. You can hear more from DuPree in an upcoming episode of our new podcast, The Agile Advocate.