xOps Ties ServiceNow to DevOps Processes

Question: What's the relationship between ServiceNow and DevOps?

Sean: At xOps we think about it in terms of building a process to most efficiently provide problem management, and how we can handle enterprise level application incidents. We want to know how the incident became a problem and what we need to automate in order to prevent the incident or problem.

We investigate what kinds of prior cases we have had that are the same and assign priorities and common attributes. When we're releasing, deploying and integrating applications, we document the change management process. These are requirements in ITIL process. ServiceNow is a very good tool for the ITIL process.

There are good tools for the deployment and budgeting, but in order to track interprocesses, I don't think there is a tool like ServiceNow.

Question: What are the steps in the xOps process?

Sean: When we are deploying an enterprise level application, we first need to plan to push it into production. Then we look at the production case and how it will impact people. We analyze the impact and change management plans. After that we map the integration points, and how to integrate the application. The developers often don’t know the integration points. All this has to happen at high velocity.

We also manage the community process in terms of the rest of the IT team. There is a TAP call for the TAP team. This sometimes includes additional teams, like the integration team. Within that structure, we are building consensus and agreeing with the process order, and are pushing these changes into production.

Once it is in production, something might happen. We need to contact the appropriate teams and coordinate the support teams. We make sure that all the contacts and all the teams are aligned. The data produced is also helpful for the change management. If there's a P1 incident, a priority one incident, a lot of people and revenue could be impacted, so we need to raise that incident, and it comes under the ITIL process.

In order to support priority incidents, we have a module called, not surprisingly, the Incident Module. We raise the incident and contact all the impacted people. We can change the priority, and assign a risk factor: high or low. Based on the priority, and based on the impact, we can convene a TAP call, and we can open a breach. Within the breach, we can contact tech admins, DD admins, application teams, support teams and the devo teams. With great collaboration, we can get all the help from different teams.

We typically also have a team called the Problem Management team. They know entire picture. In order to handle problems, we use the problem module that's in ServiceNow using DevOps style collaboration and we can automate the web flows within ServiceNow.

Question: Do a lot of DevOp practitioners actually think that that DevOps and ITIL are incompatible? That ITIL is a bunch of process mandates that constrain how you do things, and DevOps is collaborating and doing things more efficiently and more automatically.

Sean: Contrary to that point of view, here's actually wide synergy there.

The reality is that ITIL is a framework which provides a set of frameworks around processes. And DevOps is for the most part a culture of collaboration. And there's no reason you can't have a process framework integrate very well with a culture of collaboration. In fact, a process framework should meet your DevOps culture. One of the first steps is to determine what DevOps means for a given company, and then figure out how to use the pieces of process of framework which you like and apply them in the most successful way for the company.

ServiceNow is a good tool to implement the process. It's basically a work flow and the has ITIL work flow built in, so it's ready to go out of the box. We analyze all the different processes and figure out how they can fit into a culture of collaboration. I think the most interesting one of those is probably Change Management, and you can see that conflict of the old school side of approach where there are checks and balances. We turn that same process framework on its head to actually support DevOps.

Instead of using it as gates to prevent change, we use it as a process to enable change when we're doing things right. We often work with cultures that are doing 15, 20, 30 releases a day using the automated APR from ServiceNow, so they've got the whole deployment pipeline going. They are automatically creating change tickets. There is a record, they are doing it right, and they don't need to have 50 different approvals. The site will throw it over the wall to make it work, they still have the records, and the audit trail. It’s in the source record, which is ServiceNow.

There's a lot of great examples like that where the process framework, culture of collaboration and the tools can come together to really create a great synergy.

Question: Does xOps have some tools for ServiceNow?

Sean: It's basically a push button. We use the API, cut a change ticket and do the redeployment as per the client’s deployment pipeline. It's that simple. It doesn't need to be more complete than that, yet it is very practical.

Question: Tell me a little bit about the difference between doing ServiceNow remotely from Sri Lanka or on site. What are the advantages and disadvantages of each, or is it a complete advantage to do it offshore because of the price difference?

Sean: There are some huge benefits for any offshoring platform where you can easily provide 24/7 support and you can also provide it at a lower cost. I think that's something that global models will always provide you.

The challenges are around communication. We have great ServiceNow experience, understand the platform and can really ensure that clients get the value while providing the quality of service and working. We provide U.S.-based counterparts.

Question: Does the size of the organization matter?

Sean: When you're moving faster, you actually need to have the process framework in place, otherwise you're going to get lost.

When you're a 5 or 10 person shop you know what you're doing. You don't need to track everything. Once you get bigger than that, you're getting mid-size, and it actually becomes imperative that you're doing the processes to track what's going on. When you are doing 15 to 30 releases a day you do need to have a record of those, so when something breaks, you can quickly and easily go back.

Another innovative thing you can think about and look at is really capturing all your events, including deployments, auto scaling activities, etc, and capturing all of those is part of your data flow, so you've got that mass of data.

We also look at incident management. When you've got so many more moving pieces, when something breaks, it's so much more important that you've got a framework for responding appropriately to that incident. Because at small scale, you know who wrote the code, that the same guy set up the server.

We're all full stack developers. But in medium scale, you actually need to know what's been changing, what the process is, who needs to get involved so that you're engaging the right people and not all the people, and you're responding quickly to get your services restored.

Problem management becomes critically important. Because, to be good at collaboration and DevOps, you're also constantly doing some introspection and looking at what are the underlying causes of our problems. The best shops, the best DevOps shops, are the ones that are not just reacting when a problem occurs, but actually being proactive and making sure problems don't occur. That's what problem management is all about, and there's actually a whole systematic framework about how to do this and how to analyze it.

DevOps at scale makes it all the more important that you have a process framework to manage that speed and scale in an appropriate way. Now, that does not mean doing process for process' sake, and it does not mean putting barriers in the way to making things go. It means designing processes that are just enough to help you operate more efficiently.

# # #

www.xOps.it