Q&A from May 2015 Webinar
Secrets of a Scrum Master

Donna Knapp – ITSM Academy

Q: Will you be talking about how DevOps fits into all this?

A: Organizations that are adopting DevOps practices such as continuous integration, continuous delivery and continuous deployment are finding that there is a dramatic effect on service and support. Two of the aims of DevOps are to build in quality at the source and ensure that software is always in a releasable state (i.e., the software isn't failing any existing automated tests) throughout the development lifecycle. In the context of IT service management (ITSM) this means that incidents and known errors are being captured and addressed earlier in the development lifecycle. As DevOps realizes its promise, organizations will see fewer incidents making their way into the production environment, more incidents being detected and, optimally, resolved automatically via monitoring and event management practices, and therefore fewer incidents being reported to the service desk. The good new here is that this frees service desk analysts to work on more proactive activities such as enabling people to more effectively *use* technology rather than simply *fixing* technology.

Q: Can you point me to the data that shows the DevOps approach results in fewer incidents? Is there an "average" % of reduction?

A: There's not an average percent of reduction because it depends on the organization and the maturity of their DevOps practices but studies such as the State of DevOps Report and a recent study conducted by CA show some of the dramatic results that organizations have seen thus far.

Q: So it's more of a hybrid environment for IT?

A: I'm not sure I understand what is meant by 'hybrid' but if what you mean is bringing together multiple cultures then I guess I would say that is sort of the case. Scrum is a methodology (i.e., a way of working) and so, as with any methodology, the aim is to mature the culture of your organization. Scrum is no different than a process such as the ITIL incident or problem management process. It's not something that has to affect your organizational structure. It's also not something that's 'all or nothing.' You can use Scrum or Kanban in parts of your organization and not others. For example, some organizations, are already using Scrum and Kanban in their Development organizations. The goal of this presentation is to put forth the idea that these practices will benefit the service desk and support organizations as well, especially when it comes to the handling of planned projects.

Q: Are there any situations where we will not be able to develop a releasable product within a 4 week timeframe?

A: The outcome of a sprint is a potentially releasable product. In other words, just because you could release doesn't mean you will. Having said that, part of refining or 'grooming' the product backlog is to create small tasks (often referred to as 'user stories') that you can, in fact, complete during a sprint. If you're not able to complete a potentially releasable product during a sprint, you may need to break your user stories into smaller chunks. Some of the goals here are to get fast feedback by quickly demonstrating, even if only simple results, and focusing on a limited number of tasks, rather than getting lost in too big or complex of a task and not really getting anything done. It's important to keep in mind that a potentially releasable product may be as simple as a single line of code. Or as simple as a new standard image for a workstation. It doesn't have to be that all workstations have been updated to the new image.

Q: Is it compulsory to have releasable product after completing sprint?

A; A potentially releasable product, yes. At the start of each sprint you will establish a sprint goal based on the backlog items that were pulled out during sprint planning. A sprint goal summarizes the desired outcome of an iteration (i.e., what increment of a product you aim to produce during the sprint). As stated above, keep in mind that a potentially releasable product may be as simple as a single line of code. Or as simple as a new standard image for a workstation. It doesn't have to be that all workstations have been updated to the image.

Q: What is UFFA and what is KCS?

A UFFA is a Knowledge-Centered Support (KCS) concept: **U**se It, **F**lag it, **F**ix it, **A**dd it. KCS is a knowledge management methodology that is often used by service desk and support organizations. Its principles are inherently agile and iterative and so it can be adapted to any type of organization. Information about KCS is available via organizations such as the Consortium for Service Innovation and through HDI. KCS is a perfect complement to the ITIL framework; particularly in the context of knowledge management.

Q: In the context of process design and improvement, wouldn't the sprint be just the same as running a Process improvement Team (PIT)?

A: In the context of Scrum, a PIT would be the "team" (i.e., the stakeholders in the process improvement initiative). That team could then use sprints to design or improve a process. In the case of designing a process, one sprint could be devoted to defining the process. Another sprint could be devoted to defining the policies and procedures associated with an activity or set of activities. Yet another sprint could then look at how to automate those activities. The PIT is the team itself and you use Scrum practices to iteratively design or improve your process.

Q: Is a Sprint similar to a Kaizen?

A: A sprint is conceptually similar to a Kaizen event but there are some differences. A sprint is a time-boxed activity (typically 2 to 4 weeks) within a methodology called Scrum. The aim of a sprint is to produce an increment of work. That could involve development an increment of a new product (that didn't exist before) or improving an existing product. Kaizen is a Japanese term that stands for 'change for the better.' Kaizen is the practice of continuous improvement. There is something called a 'Kaizen event' where cross-functional teams get together to undertake an improvement project. In other words, kaizen events are *typically* aimed at improving processes or products that already exist, rather than developing something new.

Q: How does the Kanban show which tasks are dependent on the completion of other tasks?

A: Part of release planning (in the context of the product backlog) and sprint planning (in the context of the spring backlog) involves identifying those dependencies so the when you look pull work onto a Kanban the dependencies are understood. In terms of how to make the dependencies visible on a Kanban, it depends on your organization. Some organizations, for example, use color. The green tasks must be completed before you can move on to the yellow tasks. Other teams handle this by putting items into particular lanes on their Kanban board. The items in one lane must be completed before you can begin pulling work from the next lane. A third way could be to simply tag items with a number (1,2,3) when a specific order is required.

Q: Donna, you suggest Agile practices should be applied to planned work such as introducing new products or handling projects that flow out of incidents such as changes required to resolve an incident, but how do you apply Agile to typical application support tickets (i.e., L1, L2, and L3)?

A; My suggestion is that you wouldn't. If you have processes in place such as incident management and problem management for handling 'unplanned' support tickets, let those process do their job as the priority there is to restore service as quickly as possible. You may use Agile practices to incrementally improve those processes and make them more lean and effective but I'm not suggesting you replace your ITSM processes with Scrum. I know some organizations use Kanban boards to manage outstanding tickets but I think it's likely that will only work if you have a relatively small volume of tickets. There are a lot of get ITSM tools you can use to manage this type of ticket and something like a Kanban board may actually slow you down. In other words, my suggestion is, don't reinvent that wheel. What I am suggesting in this presentation is that you introduce practices such as Scrum and Kanban for project-related (i.e., planned) work vs. unplanned work such as handling customer incidents (i.e., work disruptions). Where this could come into play in the context of L1 and L2 could be if you have a request that comes in and you need multiple teams to fill that request (i.e., handle that project). In such a case, the project could be handled via sprint(s) (Scrum) or made visible via a Kanban board (Lean).

Q: Question: We have strong Agile/Scrum environment but we have no ITSM. We will start to introduce ITSM soon (reverse order of the topic). What advice or resources would you recommend?

A: You're lucky. You have the benefit of 20/20 hindsight. I would first of all make sure that you understand what's going on in your Dev organization. Are they using Agile practices? Are they adopting DevOps practices? Learn as much as you can about Agile and DevOps and let that knowledge influence how you go about implementing your ITSM processes. Here's an example. In a lot of organizations the change management process is a very rigorous (sometimes overly rigorous) process. In these organizations, a high percentage of changes have to go to the Change Advisory Board (CAB) or through multiple layers of approval before they can be implemented. This is a very common misinterpretation of the ITIL guidance and it often slows down the deployment of changes. In organizations that have adopted Agile and DevOps practices, they are typically making smaller, more frequent changes. These smaller changes carry with them less risk and so can often be handled as what ITIL calls standard changes. In ITIL, standard changes are pre-approved and so won't be viewed as a constraint. On the other hand, if developers have to wait till the next CAB meeting (typically once per week) to get a change approved they will be very much view ITSM as an obstacle and they will find every creative way possible to circumvent those processes. Agile and DevOps aren't just about making changes more quickly. Organizations that use Agile and DevOps practices are also introducing more automated testing practices to reduce errors and automated monitoring/event management practices to more quickly detect errors and perhaps even automatically correct errors. Your challenge is to ensure you put in place 'just enough' process for each ITSM process to meet its objectives and that you continuous adapt those processes to the needs and goals of your company. Our DevOps Foundation class has a module that suggests ways you can adapt your ITSM processes in a DevOps culture. You may also find our Certified Process Design Engineer (CPDE) and Certified Agile Service Manager (CASM) classes useful. Good luck on your journey!

Q: Donna, how would Agile affect tools that are currently in place?

A; Organizations are beginning to look for shared tools, so that Dev and Ops folks can use the same tools to collaborate or so that they can easily pass information (e.g., information about known errors) between Dev and Ops. In the context of DevOps you often also hear references to a 'tool chain' concept with involves integrating tools so that work can flow smoothly or even better, automatically across the IT value stream.

Q: Can Agile be adopted in a fully-implemented ITIL environment?

A: Absolutely! Your ITIL processes such as change, release and service asset and configuration management may need to be adapted a bit to accommodate the pace of Agile and DevOps but they are absolutely essentially. You also may find that processes such as problem and knowledge management become elevated in importance as they are used throughout the IT value stream, not just in production. Service portfolio and catalog management also become increasingly important as more self services are introduced and as there is a greater need to set expectations about the pace of change. As indicated above, a good first step in such an environment would be to help people understand the goals of Agile and DevOps practices and the use Agile practices themselves (e.g., Scrum) to refine as needed and continuously improve your processes. Just because your ITIL processes are 'fully implemented' doesn't mean they are done. Processes need to be continuously improved and aligned with the needs of your business.

Q: Is there any relevance between Agile and DevOps?

A; Yes in fact, DevOps has emerged as a movement in response to the fact that in a lot of IT organizations, Dev was using Agile practices but Ops wasn't and so there ended up being a chronic conflict. The Dev teams who were using Agile wanted to make changes more quickly and viewed the Ops teams who were often following overly rigorous processes as an obstacle. DevOps emerged when a gentleman named Patrick Dubois starting trying to understand how the Dev, Test, and Ops organizations could more effectively work together throughout the Development lifecycle in an effort to more quickly deliver better software for the benefit of the business. There's a direct correlation between Agile practices and DevOps.