Adam Donner

Web 430

Assignment 7.3

October 16, 2019

In DevOps, there are mechanisms that make it possible for new learnings and improvements that have been discovered locally to be captured and then shared globally throughout an entire organizations. This enables everyone doing work to benefit from the cumulative experience of the organization.

Many organizations use chart rooms and chat bots to automate and capture the organizational knowldege. The technique was pioneered at GitHub with ChatOps. The goal of this was to put automation tools directly int he middle of conversations happening in chat rooms. This helped to create transparency and documentation of the work that was being done. GitHub was able to reap benefits from this automation in the chat rooms. The benefits included everyone seeing everything that was happening. It also allowed engineers that were new to the organization the ability to see what daily work looked like. The found that people were more likely to ask for help when they saw others asking for help. The automation also allowed for a rapid organizational learning. Along with these benefits, it is also a very effective way to convert what is learned locally, globally Kim, Debois, Willis, & Allspaw, 2017). The benefits of using chat rooms include visibility to everyone, unlike email, events are documented in real time rather than something extra. When the library is effectively organized, archived and searchable, it will have a dramatic effect on the knowledge sharing (Wangler, n.d.)

One of the best ways to make the knowledge of an organization is to put that knowledge into a central source code repository and make it available globally. Source code is not the only thing that can be shared globally through the repository. Other things that can be shared are configuration standards for libraries, infrastructure and environments, deployment tools, testing tools and standards including security, deployment pipeline tools, monitoring and analysis tools, and tutorials and standards Kim, Debois, Willis, & Allspaw, 2017).

Dr. Steven Spear, author of The High Velocity Edge, said that “High Velocity organization multiply the power of their new knowledge by making it available, not only to those who discovered it, but also throughout the organization”. He mentioned that everything should be shared from solutions to processes. When these are shared, the entire organization benefits from the cumulative experience of everyone who has ever done that work. Google puts this into practice. Randy Shoup said in an interview that the most powerful mechanism for preventing future failures is using a single code repository for all of the Google properties. He pointed out how easy it for for knowledge to be shared across the organization. He pointed out that when knowledge is easy to propagate and discover, it will spread and lead way to an optimal way of doing things (Kim, 2015).

When an organization uses a shared library, it is important to enable rapid propagation of expertise and improvements. These libraries come self-documenting and show other engineers how to use them when the libraries have significant amounts of automated testing included. If there are test-driven development practices in place, the benefits are seen almost automatically. The test suites are turned into living, up-to-date specifications of the systems. Each library should have one owner or single team that is responsible for supporting it. There should also be only one version that is used in production ensuring that the best collection of knowledge is being used in the organization. Chat rooms or discussion boards can also be added to the library allowing for a more rapid propagate of the knowledge and speed of experience Kim, Debois, Willis, & Allspaw, 2017).

Once the project moves downstream to Operations, there are a set of non-functional requirements that enable the services to be easily depolyed and running in production. Some examples of these are sufficient production telemetry, services that are resilient, forward and backward compatibility between versions and the ability to trace the requests from uses through multiple services. When these non-functional requirements are coded, it is easier to leverage the collective knowledge and experience of the entire organization Kim, Debois, Willis, & Allspaw, 2017).

In Operations there are some work that cannot be fully automated or self-service. The goal then is to make the work as repeatable and deterministic as possible. This is achieved by making the work standardized, automating as much as possible and documenting the work. For any work in Operations that is recurring, what work is required, who is needed to do the work, what steps are needed to complete the work, and so forth needs to be known. Operations uses stories that are well defined allows for repeatable Operations work that in turn leads to better planning and more repeatable outcomes . Operations needs to have input into which components are used in production. This helps to eliminate bottlenecks, failures or unplanned work. When the components are removed, the Operations team is able to focus on an infrastructure that best needs the global needs of an organization Kim, Debois, Willis, & Allspaw, 2017).

Chat rooms and technologies such as a source code repositories, allow for the global spread of knowledge and experience. This sharing of knowledge and experience enables the entire organization to perform better and meet the overall goals of the organization.

References

Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2017). The DevOps handbook: how to create world-class agility, reliability, and security in technology organizations. Portland, OR: IT Revolution Press, LLC.

Kim, Gene. (2015, January 24). Uncovering The DevOps Improvement Principles at Google (Randy Shoup Interview). Retrieved October 22, 2019, from <https://itrevolution.com/uncovering-the-devops-improvement-principles-behind-google-randy-shoup-interview/>

Wangler, Russ. (n.d.) AgileJazz-Russ Wangler’s Blog. Retrieved October 22, 2019, from <http://agilejazz.blogspot.com/p/the-devops-handbook-summary-3-of-4.html>