



## 1- Introducción

Jeffery Smith ha estado en la industria de la tecnología durante más de 15 años y ha gestionado los cambios de operaciones de desarrollo en la empresa de tecnología publicitaria *Centro* y en la plataforma de pedidos en línea *Grub Hub*. Jeff es el autor del libro *Operations Anti-Patterns DevOps Solution*, que vamos a tratar en este trabajo.

## 2- ¿Qué es DevOps?

El término “DevOps” surge de la unión de las palabras “*development*” (desarrollo) y “*operations*” (operaciones), pero representa un conjunto de ideas y prácticas que van más allá de la combinación de ambos conceptos. DevOps describe los enfoques para agilizar los procesos con los que una idea pasa del desarrollo a la implementación en un entorno de producción en que puede generar valor para el usuario.

Estos enfoques requieren que los equipos de desarrollo y operaciones se comuniquen con frecuencia que los desarrolladores trabajen en estrecha colaboración con los equipos de operaciones de TI para agilizar el diseño, las pruebas y el lanzamiento de los sistemas de software.

Por supuesto, esto implica cambios más frecuentes en el código y el uso más dinámico de la infraestructura. Las estrategias tradicionales de gestión no pueden seguir el ritmo de la demanda, así que necesitará implementar algunas modificaciones para obtener la ventaja competitiva.

## 3- ¿Qué son los Anti-patrones?

Un Anti-patrón es, entonces, la antítesis de una solución a un problema recurrente. Es decir, son actividades, procedimientos o rutinas, usos o hábitos, que personas dentro de una organización o entorno adquieren o ejecutan, con la idea de que son la forma ‘correcta’ de hacer las cosas. Hay muchos anti-patrones, pero nosotros en este trabajo nos centraremos en los 4 que menciona Jeffery en el *podcast*, que son:

- ❖ Acumulación de información
- ❖ Cultura por decreto
- ❖ *Alert Fatigue*
- ❖ Caja de herramientas vacía



## 4- Acumulación de información

Este Anti-patrón se manifiesta en las empresas cuando tienen una persona que tiene la gran parte del conocimiento sobre un tema en concreto, de forma que, si se quiere realizar una tarea relacionada con ese tema, siempre hay que recurrir a ésta. Para explicar este concepto, se hace referencia al personaje Brent del proyecto Phoenix, el cual es un acaparador de toda la información al que hay que acudir cada vez que algo sale mal.

El mayor problema sobre este tema radica en que la mayoría de las veces sucede de forma accidental, es decir, no te das cuenta de que estás acaparando toda la información, pero tampoco haces nada por transmitir este conocimiento a los demás.

Para solucionar este Anti-patrón se debe promulgar en la empresa el desarrollo de documentación como si fuera una tarea más a realizar. El inconveniente es que los miembros de la empresa no quieren “perder” tiempo en esto, pues entre desarrollar una nueva tarea de programación y realizar la documentación oportuna, se le da más valor a un nuevo desarrollo. Además, se suele considerar como una tarea fácil que cualquiera puede desarrollar, pues solo tiene que escribir, pero la realidad es que desarrollar una buena documentación es una tarea difícil, pues se necesitan ciertos conocimientos técnicos.

Debido a que realizar documentación escrita no es una tarea que agrade mucho a los desarrolladores, existen otras formas de transmitir esta información. Muestra de ello puede ser una presentación realizada por el desarrollador sobre un proyecto o, incluso, una charla de 5 minutos a la hora del almuerzo en la que se comenta como se llevaron a cabo las tareas propuestas. Mediante estas otras técnicas, la información se transmite de una forma más interactiva. Es por ello, que en las empresas se debería optar más por esta opción que por la tradicional documentación escrita.

## 5- Cultura por defecto

Las empresas en muchas ocasiones se suelen definir a si mismas con unas características y unas cualidades que creen que cumplen y que utilizan para promocionarse. El problema recae en que, si preguntas a los trabajadores de la empresa, en la mayoría de las ocasiones, te van a decir que no consideran que se cumplan esas supuestas virtudes.

Especialmente a medida que se asciende en la organización y se comienza a tratar con el liderazgo superior, si este piensa que tienen esta gran y fantástica cultura, no están abordando ninguno de los problemas creados por esa realidad errónea.

Por otro lado, DevOps debe ser considerado parte de la cultura de las empresas, pues básicamente se trata de colaborar y cooperar entre los distintos departamentos de la empresa. De hecho, tal y como se comenta en el podcast sería buena idea que un ingeniero



de TI entreviste a una persona de operaciones para saber su perspectiva y poder concluir si tiene las habilidades que se espera que tenga un ingeniero de operaciones.

Podemos concluir que uno de los aspectos más importantes a tratar dentro de una empresa debe ser la comunicación, pues nos permite conocer cuáles son los puntos débiles y mejorar en estos aspectos.

### 6- Alarm Fatigue

Este concepto se podría traducir como el desgaste que producen las alarmas en el equipo que tiene que estar al tanto de las mismas. En palabras del propio entrevistado, son mucho más numerosas las veces que una alarma va a resultar inútil que aquéllas en las que va a resolver un problema realmente grave.

Sin embargo, según Smith, hay mucha irracionalidad respecto a este tema debido al sesgo -producido en gran parte por la gente de *business*- de querer estar prevenido contra todo aquello que pueda ir mal. A pesar de ello es clave tener en cuenta el coste implícito de establecer alertas ineficientes o innecesarias. Según su filosofía, “es preferible llegar 5 minutos tarde a una alarma real que responder 5 veces a una que es falsa<sup>1</sup>”.

Además, hay que evitar establecer alertas vagas o imprecisas. Lo ideal es que cada una tenga una descripción concreta de lo que está sucediendo, las consecuencias que puede acarrear y los puntos que sería necesario revisar. Incluso estaría bien añadir información sobre ocasiones pasadas en las que esa misma alarma ha servido de ayuda para que el ingeniero que vaya a hacerse cargo sepa exactamente qué hacer.

Finalmente, y como método para llevar a cabo el proceso de establecer sistemas de alarmas, es recomendable primero establecer todos aquellos procesos que pueden ir mal para, después, establecer prioridades individualizadas basándonos en 3 factores: La facilidad con la que el sistema puede fallar, la gravedad que tendría en tal caso y la probabilidad que hay de solucionarlo antes de que las consecuencias lleguen al cliente final.

Así, podremos establecer alertas que sean fiables y que cubran las situaciones en las que la atención urgente estará más justificada, lo que supondrá mantener la calidad del servicio sin perjudicar la experiencia del equipo de trabajo.

---

<sup>1</sup> N. del T.: Literalmente *bogus* (falso, espúreo).



## 7- Caja de herramientas vacía

Definida por Robert Bulmen como “la tendencia de las organizaciones de no invertir lo suficiente en herramientas”. Según indica el entrevistado, de hecho, está extendida la visión de que, si no existe una herramienta *open source*, seguramente no merezca la pena invertir en ello. Sin embargo, esto a veces produce un círculo vicioso en el que no se invierte en los procesos y se pierden recursos de desarrollo por ello, aunque sean costes difíciles de contabilizar.

Para manejar este problema, hay que tener en cuenta factores como el tiempo que requiere una tarea y los tiempos de cola que llevan asociadas, que pueden ser realmente costosos y tener un impacto grande en la eficiencia y productividad.

Además, la automatización es una gran herramienta para garantizar la precisión de las actividades. No sólo por la consistencia implícita de los procesos que hemos definido sino también por el contraste con la facilidad con la que el error humano aparece en tareas repetitivas y complejas. Este factor es muy importante en *DevOps*, ya que permite que las tareas ejecutadas sean idénticas en diferentes entornos.

Por último, hay que tener en cuenta, una vez más, el riesgo que existe a sobre-implementar herramientas. Por ello, conviene considerar cada cuánto se desarrolla una tarea, el riesgo real de que se comentan errores, el tiempo que implica y la posibilidad que pueda haber de que el proceso que se podría automatizar cambie, ya que esto podría ser un motivo para descartarlo.

## 8- Conclusión

La conclusión que aporta Jeffrey Smith no podría ser más clara: el auténtico punto sobre el que poner el foco es el equipo que va a trabajar en un proyecto. Es muy fácil perderse en el estudio de herramientas y “*fancy whizzbang stuff*”, pero los mayores dividendos nos los van a dar que la gente esté cómoda y los procesos funcionen de manera fluida.