

El DevOps una filosofía o un conjunto de prácticas a seguir, el termino surge de la combinación de los términos en ingles de Desarrollo y Operaciones, pero no solo se centra en esas dos cosas también trata con temas de seguridad, análisis de los datos.

DevOps en si es también una cultura, que es otra manera de decir que es una manera de trabajar o una filosofía para abordar un problema en ese caso pensado para el mundo del desarrollo. Esta cultura sigue los principios del open source, es decir, compartir la información libremente, esto incluye la toma de decisiones o evitar el miedo a cometer errores, y suele adoptar ciertos enfoques de una metodología ágil y transparente.

El objetivo principal de DevOps es acelerar el desarrollo de las aplicaciones y que tengan una entrega continua y de una buena calidad. DevOps no es en sí una manera de trabajar, es una forma de complementar a las metodologías ágiles como puede ser SCRUM, Kanban o Agile.

DevOps se divide en diferentes etapas y cada empresa que decide adaptar esta filosofía de trabajo modifica un poco el nombre de las mismas o las divide o junta, pero la mayoría se basa en las siguientes. Lo primero es la planificación, en este punto antes de programar una vez el problema está claro cuál es, se hace una fase en la que se planea como se va desarrollar, las herramientas e IDEs que se quieren utilizar, así como se va abordar el problema en general y cuál va a ser la arquitectura general de la solución. Esta fase cambia un poco cuando se hace una segunda interacción del producto, ya que en esta fase se revisarán los problemas que se han podido captar en otras fases y cómo van a ser solucionados. La segunda fase es la codificación en la cual ya se diseña el software, se desarrolla la aplicación en términos de codificarla y las modificaciones al código que sean necesarias. La tercera fase es la de construcción o compilación, en esta fase es la que se gestionan las versiones y se crean paquetes de código para luego ser usados más adelante, también se pueden usar gestores externos de paquetes como puede ser maven o npn. La cuarta fase es la fase de pruebas, donde para garantizar que el código es de buena calidad se realizan tanto pruebas manuales como escritas. La siguiente es la de lanzamiento o puesta en marcha, en esta fase se usan herramientas automáticas que ayudan a la gestión de las versiones y las compilaciones del software. A partir de aquí las fases van variando mucho más que las anteriores, en algunos casos aquí hay una fase que se encarga del despliegue de la aplicación con distintas herramientas automáticas, pero hay en muchas ocasiones que esta fase o bien se junta con la anterior o simplemente se toma como parte del trabajo sin tenerse en cuenta. Luego, la fase con el nombre de funcionamiento también se cambia en algunos casos y se junta con la siguiente, en esta fase se estudian métricas como registros para observar como es el desempeño de la aplicación con los usuarios de la misma, se suelen usar datos recopilados automáticamente de los usuarios que usan la aplicación para luego hacer un estudio. Y por último esta la fase de monitoreo, que en este caso se puede juntar mucho con la anterior, solo que aquí no se analizan datos obtenidos automáticamente, sino que se basan en problemas concretos que tienen los usuarios al utilizar la aplicación, por este motivo se todo esto es automático se puede hacer todo en la misma fase solo que hay empresas que prefieren separarlo.

Antipatrón cultura por defecto

El antipatrón "cultura por defecto" hace referencia a la situación en la que una organización mantiene su cultura o dicho de una manera más coloquial "su manera de trabajar" sin reflexionar o discutir sobre cómo mejorarla. Los miembros simplemente aceptan la cultura actual sin participar activamente en su evolución. Este antipatrón puede surgir por la falta de liderazgo, conciencia sobre la cultura organizacional o recursos para mejorarla. A largo plazo, puede llevar a una cultura estancada que impide la innovación y el crecimiento.

Para evitar este antipatrón, los líderes de la organización deben fomentar una cultura y colaborativa. Esto implica promover la mejora continua, crear espacios para la discusión y retroalimentación, y dedicar recursos para la formación y el desarrollo de habilidades de liderazgo. También es importante que los miembros se sientan empoderados para participar en la evolución de la cultura y que se fomente la colaboración y el apoyo mutuo en la organización.

Antipatrón caja de herramientas vacía

El antipatrón conocido como "caja de herramienta vacía" se refiere a la situación en la que un equipo de desarrollo tiene acceso a una amplia gama de herramientas y tecnologías para trabajar en un proyecto, pero sin ninguna orientación clara sobre cómo utilizarlas de manera efectiva. En otras palabras, los desarrolladores tienen una gran cantidad de opciones de herramientas, pero no saben cuál usar para cada tarea específica.

Este antipatrón puede surgir debido a la falta de liderazgo en el equipo de desarrollo, a la falta de experiencia del líder del proyecto o a la presión para adoptar nuevas tecnologías sin una evaluación adecuada. Los efectos del antipatrón "caja de herramienta vacía" pueden ser graves para un proyecto, ya que los desarrolladores pueden perder mucho tiempo tratando de averiguar qué herramientas utilizar y cómo utilizarlas. Además, esto puede resultar en un código inconsistente, ya que cada desarrollador puede utilizar herramientas diferentes para realizar tareas similares.

Para evitar este antipatrón, es importante que los líderes del proyecto establezcan una guía clara y concisa sobre las herramientas y tecnologías que se deben utilizar para el proyecto. Además, se deben establecer prácticas recomendadas y normas para el uso de herramientas y tecnologías en el proyecto. De esta manera, los desarrolladores tendrán una comprensión clara de lo que se espera de ellos y podrán utilizar las herramientas y tecnologías de manera efectiva para lograr los objetivos del proyecto.

Alerta fatiga

Se describe la fatiga como un estado en el que los miembros del equipo de operaciones sufren de cansancio mental y físico debido a una sobrecarga de trabajo constante y excesiva. La fatiga puede ser causada por diversos factores, como la falta de personal, procesos ineficientes o herramientas inadecuadas, y puede tener efectos perjudiciales tanto para la salud y el bienestar del equipo como para el rendimiento general del equipo.

La fatiga puede manifestarse en una variedad de síntomas, como dificultades para concentrarse, disminución en la calidad del trabajo, aumento de errores y distracciones. Para evitar la fatiga en el equipo de operaciones, es importante implementar medidas para reducir la carga de trabajo, como la automatización de procesos, la optimización de herramientas y la priorización de tareas críticas. Además, es fundamental crear una cultura de trabajo equilibrado y apoyo mutuo, donde se respeten los límites personales y se promueva la comunicación abierta y la colaboración en equipo.