

# DevOps y evaluadores

En una serie de artículos, Paul Gerrard, consultor y gurú de las pruebas, trata una variedad de temas de prueba. Aquí habla acerca de la adopción de DevOps desde el punto de vista de un evaluador y las pruebas. DevOps es parte de una estrategia general que las organizaciones usan para entregar software con frecuencia y alta calidad. El resultado más evidente de las implementaciones exitosas de DevOps es la reducción del tiempo que se requiere para que los cambios de software pasen de una idea a producción.

Paul Gerrard  
Gerrard Consulting

Patrocinado por



## ¿Qué significa DevOps para los evaluadores?

### Información general

En este artículo, quiero hablar acerca de la adopción de DevOps desde el punto de vista de un evaluador y las pruebas. El movimiento (a falta de una mejor etiqueta) de DevOps está avanzando con rapidez. Al igual que muchos otros movimientos que ha realizado el sector, la velocidad de adopción es más rápida que la definición del movimiento en sí. DevOps aún no se ha definido con claridad, y los matices culturales, la capacidad emergente de las tecnologías nuevas y la variedad de casos de estudio (principalmente exitosos) significan que aún se debaten ampliamente los problemas actuales.<sup>1</sup>

Según con quién se hable, DevOps puede ser una solución a un problema o una meta en sí. En algunas empresas, la meta es hacer la transición a lo digital, y DevOps forma parte de la estrategia general de entrega con frecuencia y alta calidad. Este es el contexto que daré por sentado en este documento. Sin embargo, en la comercialización de tecnologías y servicios relacionados con DevOps, esta meta puede ser un poco confusa. El reto del cambio cultural (o en forma más concreta, el cambio de comportamiento) necesario para el éxito suele subestimarse.

El otro aspecto que daré por sentado es que la idea general es nueva para los evaluadores que están involucrados en DevOps y se ven afectados por DevOps. Este artículo será tanto una introducción a DevOps para estos evaluadores como un análisis de su impacto en las prácticas de prueba. Si usted es un profesional con experiencia en DevOps, espero que el artículo también le resulte útil. Si no es un evaluador, por lo menos, podrá ver el punto de vista del evaluador.

### Para los no iniciados: ¿Qué es DevOps?

De manera simplista, DevOps es una etiqueta que se ha colocado en la idea equipos de desarrollo y operaciones de sistemas que trabajan de manera más cerca. En el llamado canal de entrega, desde la confirmación del código fuente hasta el funcionamiento en producción, los desarrolladores adaptan y automatizan algunas de las actividades de las operaciones. El equipo de operaciones tiene mayor visibilidad de las actividades de los desarrolladores y cierta influencia sobre estas. La motivación principal de esto es acelerar la instalación e implementación de software. Al acercar las operaciones y el desarrollo —de hecho, convertirlos en un equipo ágil—, se implementa lo que podrían llamarse “operaciones ágiles”.

El resultado más evidente de las implementaciones exitosas de DevOps es la reducción del tiempo que se requiere para que los cambios de software pasen de una idea al funcionamiento en producción. Cuando un desarrollador dice que un cambio de software está “listo”, la transición al uso en producción se lleva a cabo con la ayuda de una automatización general. Se usan herramientas y procesos automatizados en la configuración de sistemas, el proceso de compilación, las pruebas, la implementación en los entornos de prueba, ensayo y producción, el monitoreo posterior a la implementación, la evaluación y el funcionamiento.

### Entonces, ¿DevOps se trata solo de herramientas?

En un nivel, la meta de DevOps es eliminar los cuellos de botella en el canal de entrega a través de automatización. Pero la automatización de los procesos en etapas aún requiere gobernanza. La mayoría de los procesos automatizados no son autónomos en realidad, ya que no pueden realizar las tareas sin intervención humana en el mantenimiento o en el manejo de excepciones. Un proceso de DevOps completamente automatizado carece de sentido sin considerar el factor humano. Si bien las herramientas se encargan de gran parte del trabajo pesado, las personas que ejecutan el proceso son las que hacen que funcionen bien, o mal.

Entonces, ¿DevOps tiene que ver con personal de desarrollo y operaciones que trabaja en estrecha colaboración con la ayuda de herramientas?

No, tampoco es eso. Las transferencias entre procesos automatizados suelen involucrar otros procesos, en general, pruebas de un tipo u otro. Los desarrolladores y evaluadores son los encargados de crear pruebas automatizadas.

El objetivo de estas pruebas es proporcionar suficiente información para otros procesos —o, con la misma frecuencia, personas— a fin de realizar la transición entre las etapas del canal. Los evaluadores y desarrolladores que realizan pruebas se aseguran de que el proceso de DevOps funcione con éxito y de manera confiable.

“Me duele la cabeza. ¿Qué es DevOps en realidad?” Debo decir que es una disciplina emergente en desarrollo. Esta pregunta se plantea y analiza en detalle, en una excelente publicación que incluyo aquí.<sup>1</sup> Ese debate ocurrió unas semanas antes de que escribiera este artículo. Así, puede ver que aún no se ha acordado una definición de DevOps. Quizás, nunca se llegue a un acuerdo.

¿Qué significa eso para los evaluadores? Significa que todavía no existe “una sola forma correcta” y que su rol en un sistema de DevOps que está en desarrollo (y todo sistema está en desarrollo) aún no es fijo. Puede hacer dos aportes principales:

1. Debe prestar atención a las cosas que “duelen” y trabajar para hacer que sean menos dolorosas.
2. Debe identificar las oportunidades e intervenciones que aporten valor al proceso de DevOps.

Si existe un mantra que describe mejor el motor hacia DevOps, este es “si duele, hágalo más”. Puede ser un cliché, pero lo usaré como el contexto para implementar y mejorar las prácticas de prueba de DevOps.

### Si duele, hágalo más (seguido)

La dificultad o el dolor que experimentamos cuando hacemos un trabajo en particular nos afecta negativamente. Si no nos gusta hacer una tarea, tendemos a posponerla. Cuando finalmente nos enfrentemos a la tarea, esta será más dolorosa. Esto ocurre cuando tenemos que ir al dentista, limpiar el garaje, integrar software, realizar pruebas y así sucesivamente. Según nuestra experiencia, cuanto menos realicemos estas tareas, más traumática será la tarea cuando finalmente la hagamos. Martin Fowler sugiere tres razones por las que la ejecución frecuente o incluso continua de ciertas tareas reduce el dolor.<sup>2</sup>

La primera razón es que las tareas más grandes y complejas son difíciles de planificar, administrar y controlar. Al dividir las tareas grandes, es más fácil realizarlas, son menos riesgosas y, si algo sale mal, es más fácil desentrañar el motivo. La segunda razón es que muchas tareas (y las pruebas son el ejemplo perfecto) proporcionan comentarios. Si esos comentarios se reciben pronto y con frecuencia, los problemas pueden abordarse rápidamente y antes de que se pierda más tiempo. La tercera razón es que si practicamos una actividad con mayor frecuencia, la realizamos cada vez mejor. Aprendemos cómo realizarla de manera eficaz. Además, podemos ver oportunidades de automatizarla de alguna forma.

Desde el punto de vista del evaluador, este mantra nos obliga a tomarnos con mayor seriedad la idea de la automatización en el proceso de prueba. Si hay intervenciones manuales (en general, entre las etapas automatizadas del proceso de DevOps), se las verá como los puntos débiles: los cuellos de botella, las causas de demora y los aspectos del proceso que son potencialmente menos confiables y propensos a errores. Las pruebas manuales son dolorosas. Sí, le podrían encantar las pruebas exploratorias; podría temer que solo usted, como ser humano, pueda encontrar esos errores difíciles que nunca se encontrarán a través de la automatización, que usted, como evaluador, es la única persona de confianza que podrá evitar que ocurra un desastre.

Como evaluador, podrá ser doloroso para usted confiar en que los desarrolladores y la automatización realicen las pruebas correctamente. Si duele, debe hacerlo más seguido.

## Pruebas, automatización y confianza

Existe un gran debate en torno al significado de, por ejemplo, la verificación y las pruebas<sup>3</sup>, y en qué medida podemos confiar en evaluadores, verificaciones y automatización.<sup>4,5</sup>

No estoy diciendo que podemos confiar ciegamente en las verificaciones automatizadas. No hay duda de que necesitamos una mayor sofisticación que esa. Sin embargo, a los efectos de este artículo, podemos al menos separar las pruebas y la actividad de ejecución de pruebas en cuatro componentes.

1. Verificaciones que los desarrolladores pueden automatizar como parte de sus procesos de integración continua y verificación de componentes
2. Verificaciones que pueden automatizarse (en general, por parte de los evaluadores de sistemas) para ejecutar transacciones integrales, de enlace o a nivel de API
3. Pruebas mediante las que se pueden realizar verificaciones de compatibilidad entre navegadores, sistemas operativos, plataformas
4. Pruebas que solo puede realizar un ser humano

En este artículo, solo puedo brindar algunas sugerencias sobre cómo hacer estas distinciones; por supuesto, cada entorno es diferente. La pregunta que viene al caso de este artículo es: “¿Cómo hace el evaluador para ‘soltar’ las verificaciones manuales tardías?”. Ya he hablado sobre la eliminación de las verificaciones manuales tardías.<sup>6</sup> Para esto, se requiere un esfuerzo proactivo y confianza.

El enfoque principal de sus esfuerzos serán los siguientes puntos:

1. Siempre que sea posible, las verificaciones manuales que se puedan realizar en el nivel de un componente deberían estar a cargo de los desarrolladores. Como evaluador, podría sugerir estas pruebas en una sesión de emparejamiento o pizarra. Podrá ser necesario que usted las escriba y las incluya en el sistema de integración continua.
2. Las pruebas integrales o de la interfaz del usuario podrán requerir automatización. Estas pruebas deberán minimizarse, ya que, en general, su ejecución es lenta, son frágiles y requieren mantenimiento con frecuencia. Considere si es necesario ejecutarlas en cada verificación del código o si podrían reservarse solo para lanzamientos más grandes y menos frecuentes.
3. ¿Qué pruebas únicamente manuales podrían ejecutarse en componentes que aún no están integrados en un candidato para el lanzamiento? ¿Las pruebas manuales pueden realizarse en sesiones de emparejamiento con desarrolladores? ¿Existen alternativas para estas pruebas? ¿Los guiones gráficos o los prototipos del tipo BDD (desarrollo impulsado por comportamiento) podrían ser útiles? ¿Podrían realizarse verificaciones de IU en modelos o planos de pantalla?
4. ¿Cuáles son las verificaciones que deben ejecutarse una sola vez, en forma manual, a diferencia de las verificaciones que deben conservarse con fines de regresión y, por lo tanto, son candidatas para la automatización?

Arriba mencioné la idea de la confianza. Otra forma de verlo es especular sobre cómo se podría probar un sistema con precisión si no hubiera pruebas manuales tardías en absoluto. Imagine un entorno en el que todas las pruebas se ejecutaran a través de herramientas. ¿Sus preocupaciones tendrían que ver con el hecho de que no confía en que los desarrolladores hagan un buen trabajo de prueba? Si mueve las pruebas a las etapas iniciales de un ciclo de vida (tal como sugiero en mi artículo anterior), la duda debería ser menor. Como evaluador, si se desempeña más como un explorador que identifica los riesgos y los evalúa, selecciona pruebas y se asegura de que se las incorpore en el desarrollo y la automatización, sus inquietudes podrían minimizarse.

Sin duda, debe dejar de pensar que es el guardián de la calidad, la última defensa, la única persona a la que le importa. Debe pensar más como un visionario, un identificador y administrador de riesgos, un explorador, un facilitador y un capacitador/mentor.

## Práctica, monitoreo y mejoras

A pesar de todas las buenas intenciones de reducir o eliminar la dependencia de verificaciones manuales tardías, los errores seguirán apareciendo. Cuando se lanza software a producción, surgen los problemas. Una de las disciplinas principales de DevOps desde el punto de vista de operaciones es un nivel más profundo de monitoreo, es decir.

Monitoreo en cada capa, desde componentes y transacciones simples en las aplicaciones, hasta integración y mensajes, y, por supuesto, la infraestructura en sí. Una de las metas del monitoreo es generar alertas sobre errores antes de que los usuarios se vean afectados por estos. Esto es un tanto ambicioso, pero es la meta principal.

Cuando se encuentran problemas en producción, la tarea es usar los análisis obtenidos a partir del monitoreo no solo para buscar la causa y resolverla, sino también para perfeccionar el proceso de prueba, automatizado o manual, a fin de reducir las probabilidades de que ocurran problemas similares en el futuro. Aquí presentamos y analizamos el rol de las pruebas y los análisis en todo el proceso del canal.<sup>7</sup>

Las pruebas automatizadas en el proceso de DevOps podrían llamarse “monitoreo”. Junto con el monitoreo en producción, se podría decir que el monitoreo en todo el proceso de DevOps y en producción amplía el alcance de las pruebas. Por lo tanto, DevOps no disminuye el rol de los evaluadores.

---

## Conclusión

Hace poco, me hicieron la siguiente pregunta: “¿En qué caso no se debería implementar DevOps en una organización?”. Es una buena pregunta, pero creo que detrás de ella hay una preocupación con respecto a si DevOps llegó para quedarse y si los evaluadores deberían percatarse de esto. Mi respuesta es simple.

¿Por qué no querría que los desarrolladores y el personal de operaciones tengan una comunicación recíproca? ¿Por qué no querría que haya compilaciones e implementaciones más confiables en prueba y producción? ¿Por qué no querría la mejor tecnología para dar lugar a canales más precisos, eficaces e informativos? DevOps es algo bueno pero no siempre fácil de alcanzar. De más está decir que requiere un cambio cultural, que no siempre es fácil.

Para el evaluador, DevOps nos otorga una mayor influencia en las etapas iniciales de los proyectos, nos obliga a pensar más seriamente sobre la automatización en las pruebas, la entrega de información y la toma de decisiones. Los evaluadores deben adoptar DevOps debido a que les brinda oportunidades de ser proactivos y adquirir mayor autoridad y respeto en los equipos de proyectos.

## Acerca del autor

Paul Gerrard es consultor, profesor, autor, administrador de sitios web, desarrollador, evaluador, conferencista, instructor de remo y editor. Ha llevado a cabo tareas de consultoría en todos los aspectos de prueba y control de calidad de software, y se ha especializado en el control de pruebas. Ha dado discursos de apertura y ha presentado tutoriales en conferencias sobre pruebas en Europa, EE. UU., Australia, Sudáfrica, por algunos de los que ha ganado premios.

Paul, que se educó en las universidades de Oxford e Imperial College London, ganó en 2010 el Premio europeo a la excelencia en pruebas de Eurostar y en 2013, ganó el Premio a toda una vida de The European Software Testing Awards (TESTA).

En 2002, Paul escribió "Risk-Based E-Business Testing" (Pruebas basadas en el riesgo de negocios electrónicos) con Neil Thompson. En 2009, Paul escribió "The Tester's Pocketbook" (El cuaderno del evaluador). Paul escribió "The Business Story Pocketbook" (El cuaderno de las historias de negocios) con Susan Windsor en 2011 y escribió "Lean Python" (Python en pocas palabras) en 2014.

En 2014, Paul se desempeñó como director del programa para la Conferencia de EuroSTAR en Dublín.

Es director de Gerrard Consulting Limited, director de TestOpera Limited y el anfitrión del Test Management Forum.

Correo: paul@gerrardconsulting.com

Twitter: @paul\_gerrard

Web: gerrardconsulting.com

Para obtener más información, visite **Desarrollo y prueba** con CA Technologies.



Comuníquese con CA Technologies en [ca.com/ar](http://ca.com/ar).



CA Technologies (NASDAQ: CA) crea un software que impulsa la transformación en las empresas y les permite aprovechar las oportunidades de la economía de la aplicación. El software es el centro de cada empresa, en cada sector. Desde la planificación hasta el desarrollo, la administración y la seguridad, CA trabaja con empresas en todo el mundo para cambiar el estilo de vida, realizar transacciones y comunicarse, mediante entornos móviles, de nubes públicas y privadas, entornos distribuidos y centrales. Obtenga más información en [ca.com/ar](http://ca.com/ar).

### Referencias

1. "What is DevOps" (Qué es DevOps), The Agile Admin, <http://theagileadmin.com/what-is-devops/>
2. "Frequency Reduces Difficulty" (La frecuencia reduce la dificultad), Martin Fowler, <http://martinfowler.com/bliki/FrequencyReducesDifficulty.html>
3. "Testing and Checking Refined" (Perfeccionamiento de pruebas y verificaciones), James Bach, Michael Bolton, <http://www.satisfice.com/blog/archives/856>
4. "A New Model for Testing" (Un nuevo modelo para las pruebas), Paul Gerrard, <http://dev.sp.qa/download/newModel>
5. "The New Model and Testing v Checking" (El nuevo modelo y pruebas versus verificaciones), Paul Gerrard, <http://blog.gerrardconsulting.com/?q=node/659>
6. "How to Eliminate Manual Feature Checking" (Cómo eliminar la verificación manual de funciones), seminario web de Paul Gerrard, <http://blog.gerrardconsulting.com/?q=node/622>
7. "Thinking Big: Introducing Test Analytics" (Pensar en grande: presentación del análisis de pruebas), Paul Gerrard, <http://blog.gerrardconsulting.com/?q=node/630>