

# 4 Métricas de DevOps

María González Otero

Lucía García Sopena

Sergio Murillo Álvarez



# ¿CÓMO CONSEGUIR UN SOFTWARE DE CALIDAD?



BUSCAR UN OBJETIVO GLOBAL



PREMIAR A DESARROLLADORES  
POR CALIDAD Y NO CANTIDAD DE  
CÓDIGO

# MÉTRICAS SEGÚN SU FUNCIÓN

## VELOCIDAD

- Frecuencia de implementación
- Tiempo de entrega de cambios



## ESTABILIDAD

- Tasa de fallos en los cambios
- Tiempo para restaurar el servicio



# ¿PARA QUÉN ESTAN DIRIGIDAS?



Quieres medir el  
rendimiento de entrega de  
software

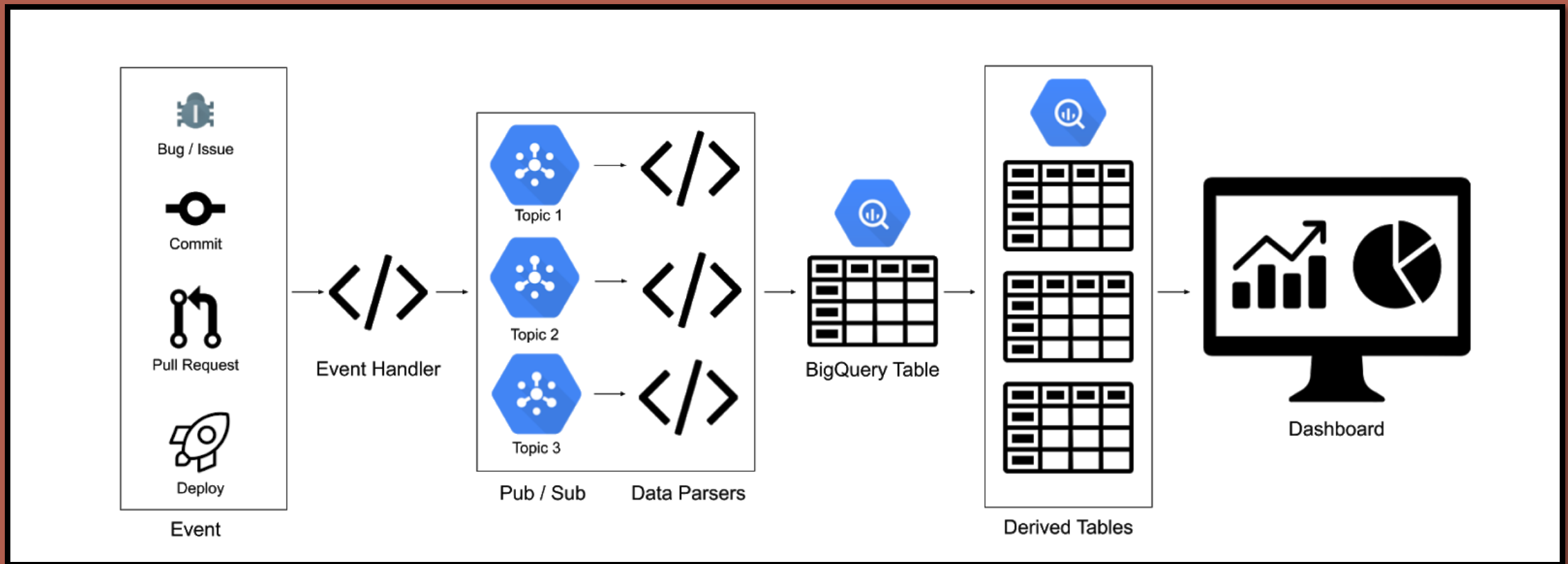


Tienes un proyecto en  
GitHub o GitLab.



Proyecto con despliegues

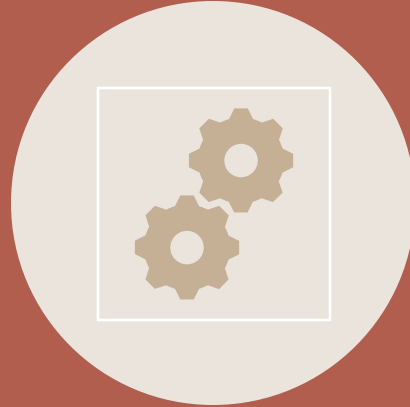
# ¿CÓMO FUNCIONA?



# ¿CÓMO USARLO EN NUESTRO PROYECTO?



HACER UN FORK AL PROYECTO DE  
GITHUB



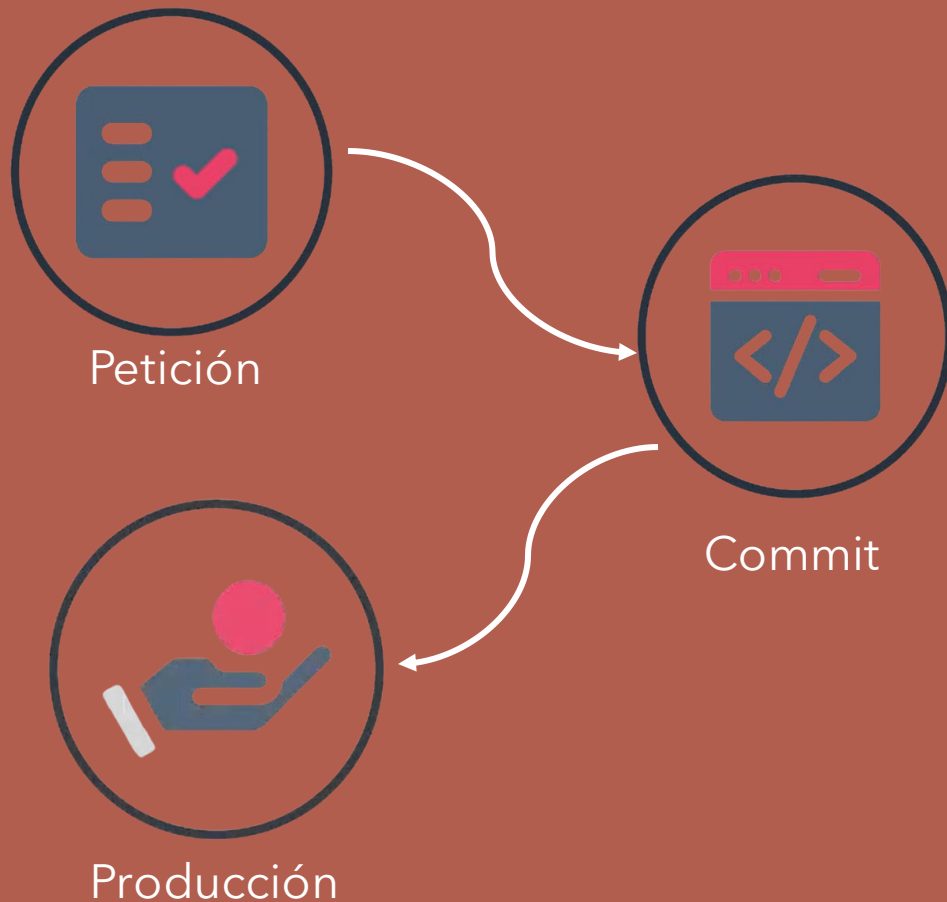
LANZAR LOS SCRIPTS DE  
AUTOMATIZACION



CONFIGURAR EL ENTORNO DE  
DESARROLLO PARA ENVIAR  
EVENTOS AL WEBHOOK

# 1. Tiempo de espera

Mide el tiempo que un cambio tarda en llegar a producción. Es un buen indicador de la eficiencia y capacidades de un equipo.



Podemos clasificarlo en función del tiempo:

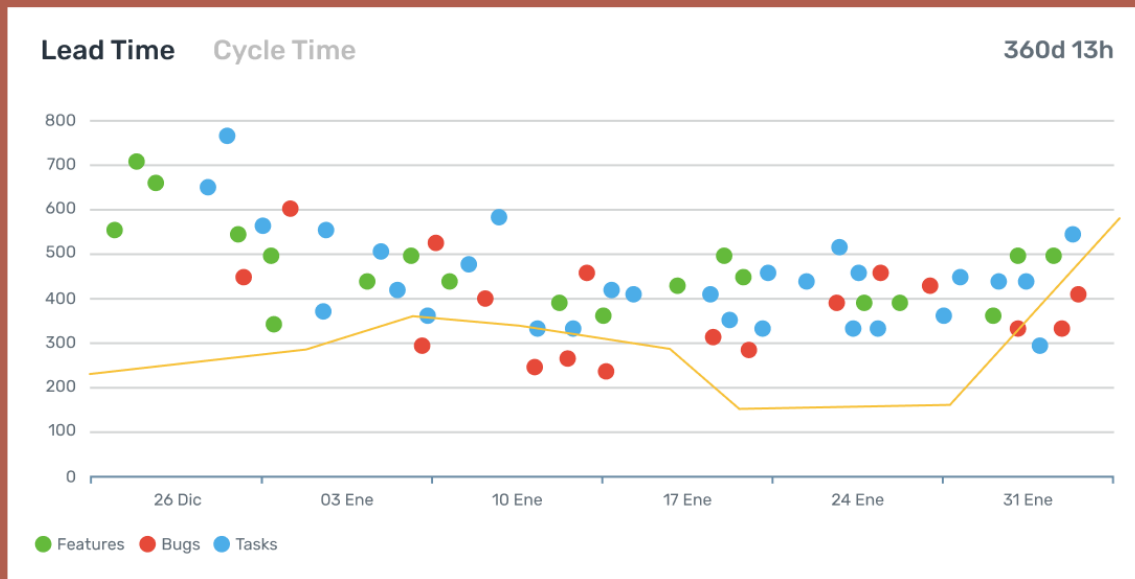
- Élite.
- Alto.
- Medio.
- Bajo.

# 1. Tiempo de espera

El objetivo debe ser reducir esta métrica a medida que pase el tiempo.

Herramientas

Buenas prácticas



Automatización de pruebas:  
Mediante integración continua y entrega  
continua

Cambios pequeños > Cambios grandes

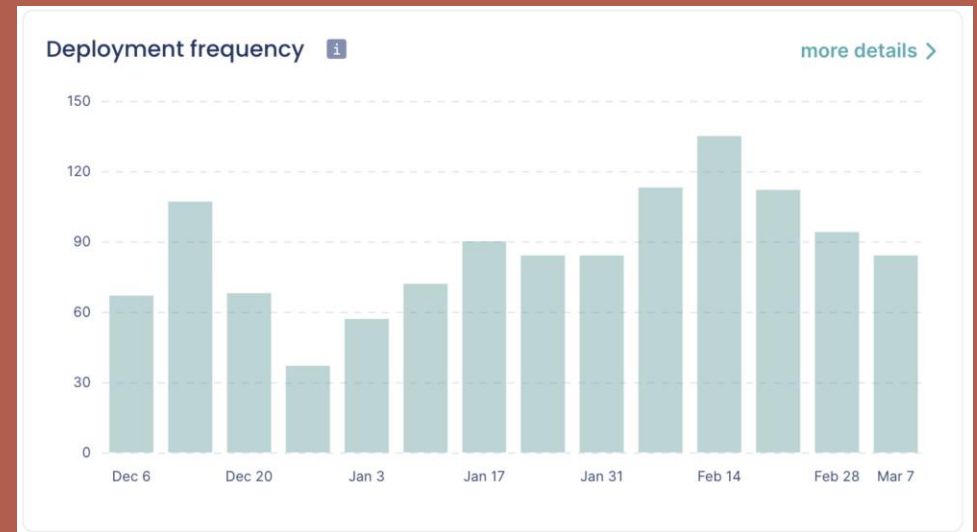


# 2. Frecuencia de despliegue

Indica la frecuencia con que una organización pasa a producción de forma exitosa. El objetivo que buscamos es hacer implementaciones más pequeñas que sean constantes.

La medida puede realizarse mediante:

- Aplicaciones que contabilicen los despliegues.
- Scripts manuales.



El objetivo sería hacer despliegues bajo demanda. Indicaría la cohesión del equipo, y su eficiencia y efectividad.

# 3. Tiempo medio de restablecimiento

Nos ayuda a cuantificar el tiempo de un fallo. El tiempo que se necesita para volver a servicio desde que se produce un fallo.



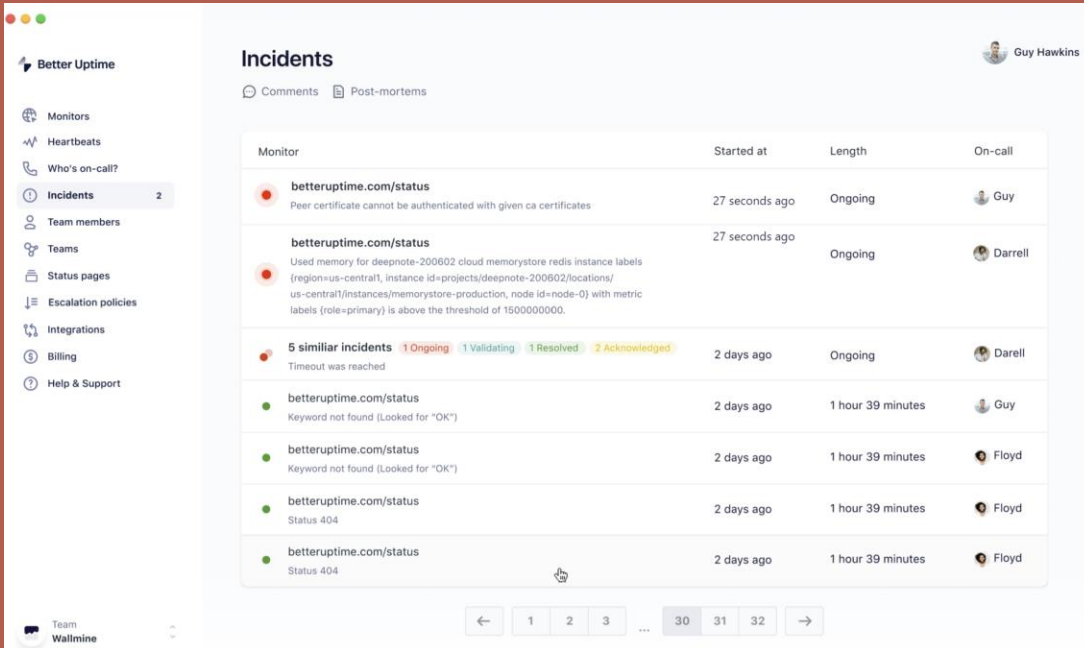
Se espera que en equipos de alto rendimiento recuperen servicios en menos de una hora.

# 3. Tiempo medio de restablecimiento

El objetivo es aumentar la velocidad de implementación. Para minimizarlo podemos usar herramientas y/o buenas prácticas.

Herramientas

Prácticas



The screenshot shows the 'Incidents' page in the Better Uptime dashboard. The left sidebar contains navigation options: Monitors, Heartbeats, Who's on-call?, Incidents (2), Team members, Teams, Status pages, Escalation policies, Integrations, Billing, and Help & Support. The main content area displays a table of incidents with columns for Monitor, Started at, Length, and On-call. The incidents listed are:

Monitor	Started at	Length	On-call
<b>betteruptime.com/status</b> Peer certificate cannot be authenticated with given ca certificates	27 seconds ago	Ongoing	Guy
<b>betteruptime.com/status</b> Used memory for deepnote-200602 cloud memorystore redis instance labels (region=us-central1, instance id=projects/deepnote-200602/locations/us-central1/instances/memorystore-production, node id=node-0) with metric labels (role=primary) is above the threshold of 1500000000.	27 seconds ago	Ongoing	Darrell
<b>5 similar incidents</b> 1 Ongoing 1 Validating 1 Resolved 2 Acknowledged Timeout was reached	2 days ago	Ongoing	Darell
<b>betteruptime.com/status</b> Keyword not found (Looked for "OK")	2 days ago	1 hour 39 minutes	Guy
<b>betteruptime.com/status</b> Keyword not found (Looked for "OK")	2 days ago	1 hour 39 minutes	Floyd
<b>betteruptime.com/status</b> Status 404	2 days ago	1 hour 39 minutes	Floyd
<b>betteruptime.com/status</b> Status 404	2 days ago	1 hour 39 minutes	Floyd

At the bottom of the incident list, there is a pagination control showing page 1 of 32, with a current page indicator on page 1.

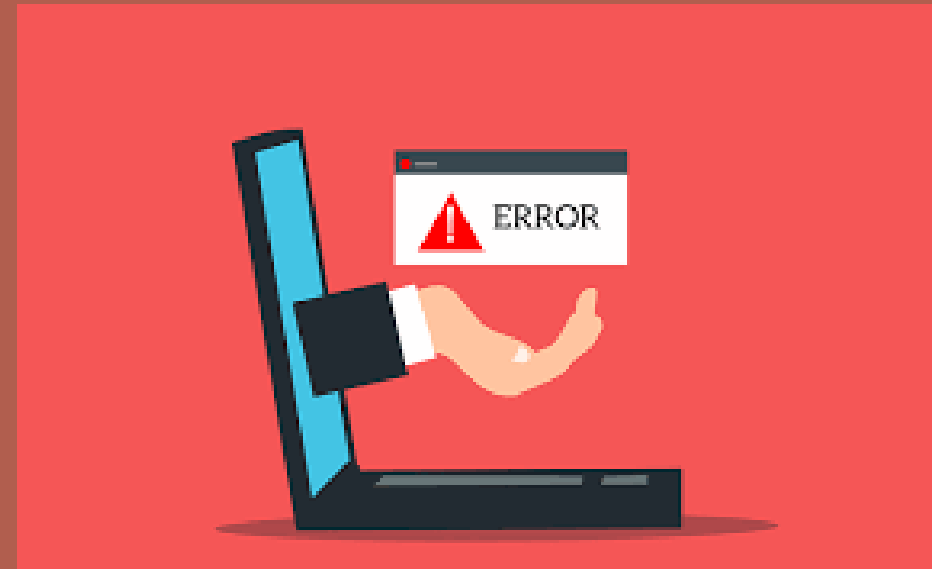
Simplificar los procesos que se realicen. Al ser más sencillos, tendremos errores que sean más fáciles de identificar.

# 4. Tasa de errores por modificaciones

- Es la relación entre modificaciones que han fallado y modificaciones exitosas.

¿Cómo se calcula ?

$$\frac{n^{\circ} \text{ veces despliegue da error}}{n^{\circ} \text{ total despliegues}}$$



## 4. Tasa de errores por modificaciones



**¿Por qué es importante?**



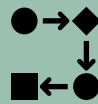
Todo el tiempo dedicado a resolver un problema es tiempo que no se está usando en liberar nuevas funcionalidades.

## 4. Tasa de errores por modificaciones

### ¿Cómo se mejora ?



Asegurar que todo el código está cubierto por test automatizados.



Mejorando los procesos de CI para incorporar y ejecutar los test implementados.



Realizando revisiones de código exhaustivas y completas para ayudar a evitar que se produzcan errores en producción.

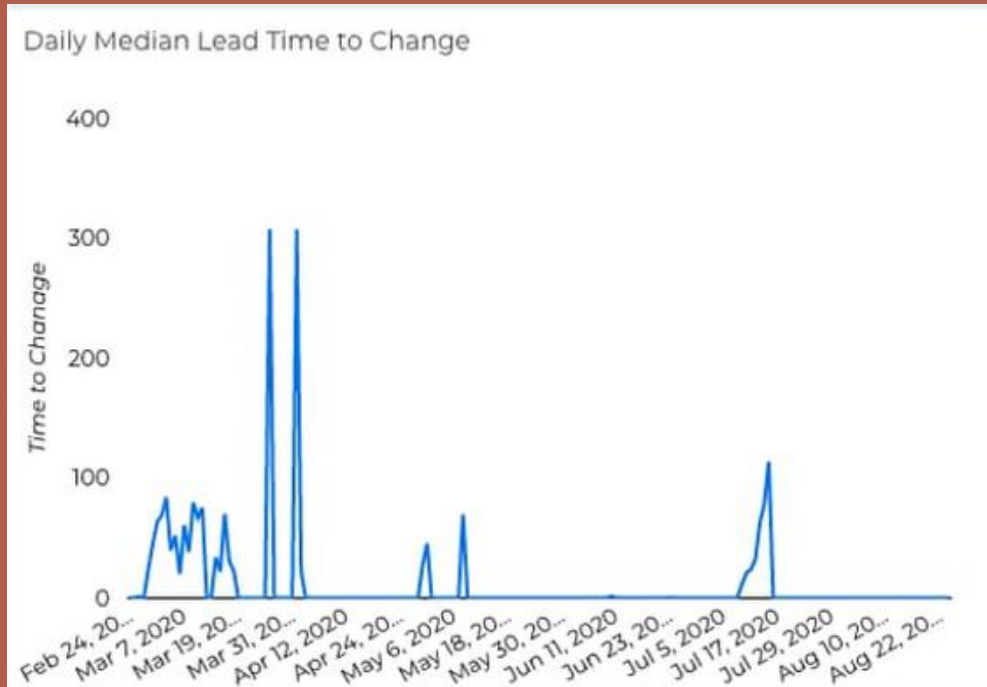


Análisis de dependencias y sistemas sensibles a ser afectados por un cambio.

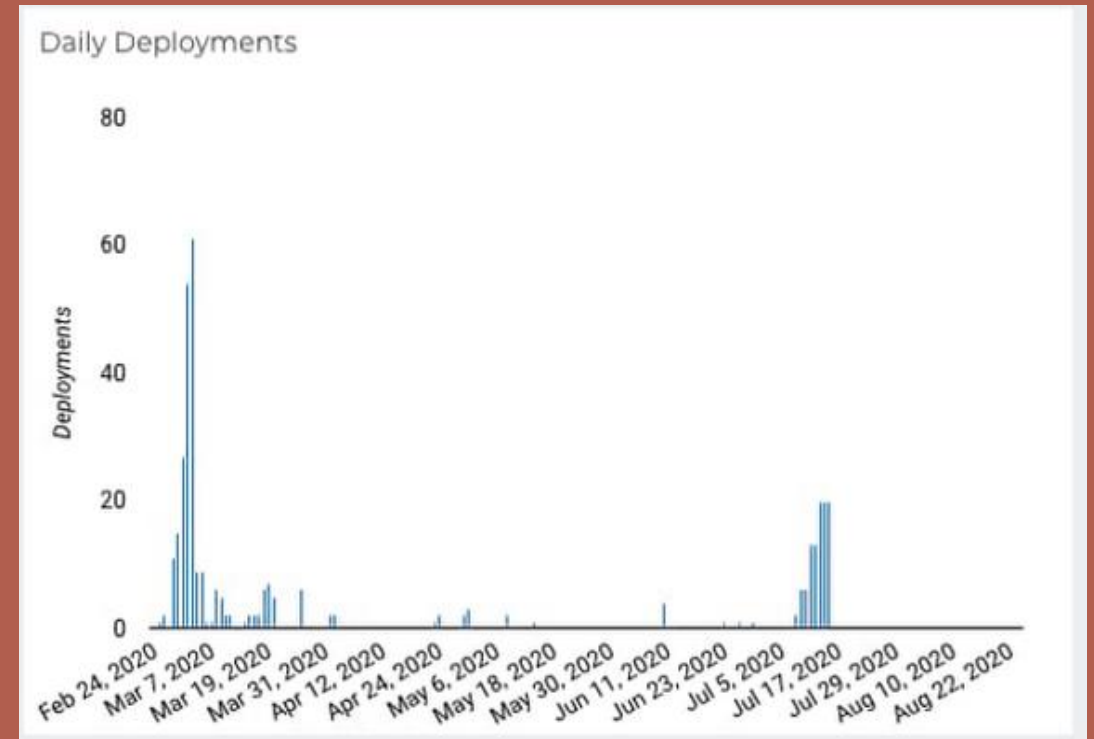
# Clasificación de las organizaciones según las métricas

	<b>Elite</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>
<b>Tiempo de espera</b>	Menos de un día	Entre 1 día y 1 semana	Entre 1 semana y 1 un mes	Entre 1 y 6 meses
<b>Frecuencia de despliegue</b>	Múltiples despliegues por día	Entre 1 vez al día y 1 a la semana	Entre 1 vez a la semana y 1 al mes	Entre 1 vez al mes y 1 cada 6 meses
<b>Tiempo de restablecimiento</b>	Menos de 1 hora	Menos de un día	Menos de un día	Entre una semana y un mes
<b>Tasa de errores por modificación</b>	0-15%	0-15%	0-15%	46-60%

# Resultados



El tiempo medio para que un commit sea desplegado en producción

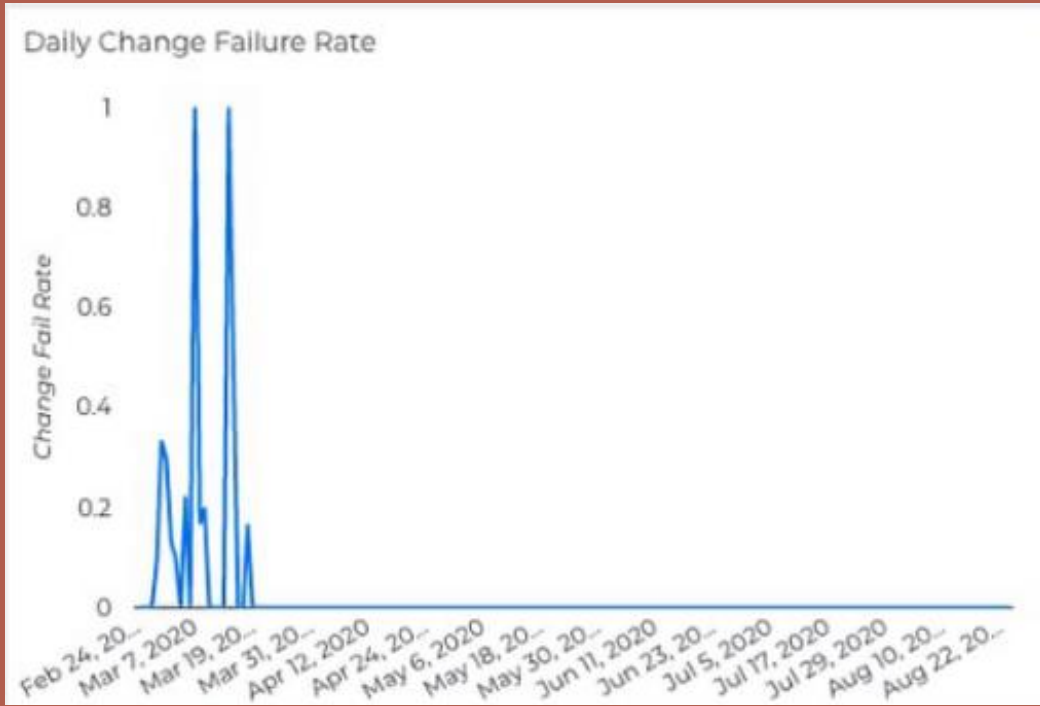


nº despliegues por día

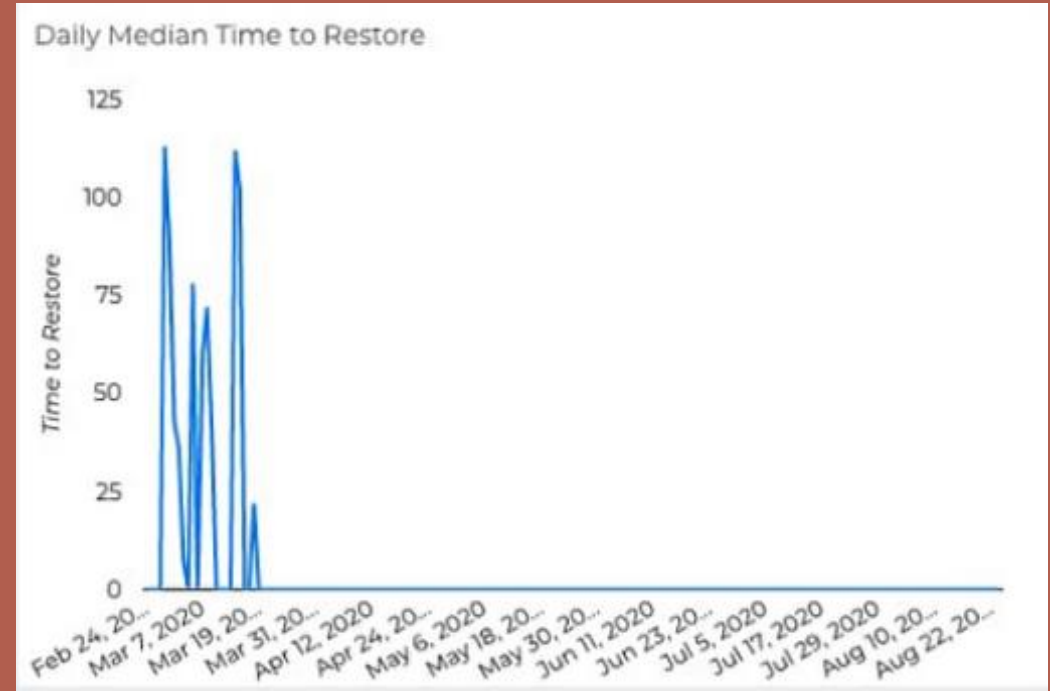




# Resultados



Nº de fallos por nº de despliegues



Tiempo de recuperación tras un fallo



# Conclusión

- Son fundamentales para mejorar las entregas de software
- Permiten mejorar tiempos de nuestro proyecto
- Costoso generar los datos
- Ausencia de definición clara para la medición