



PREPÁRATE
PARA SER EL
MEJOR



+ **ENTREMIENTO
EXPERIENCIA**



BIENVENIDOS.



Azure DevOps

Ing. Erick Arostegui Cunza
Instructor

earostegui@galaxy.edu.pe

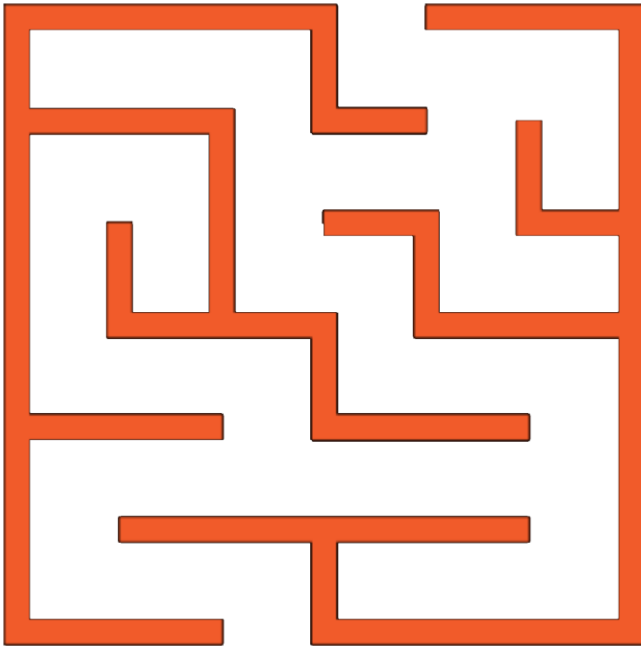


AGENDA

INTRODUCCIÓN A DEVOPS

- ▶ Problemáticas más comunes y desperdicios producidos
- ▶ Definiendo DevOps
- ▶ Caminando hacia DevOps (Cambio cultural, cambio Organizacional, definiendo objetivos)
- ▶ Automatización (Tecnologías y Plataformas en la Nube)
- ▶ Azure DevOps y ¿Por qué usarlo?

Despliegues complejos



Demasiada transferencia entre equipos.

Numerosos equipos involucrados

Intervención Manual

Tiempos muertos



COTS (Commercial Off The Shelf) vs Software a medida



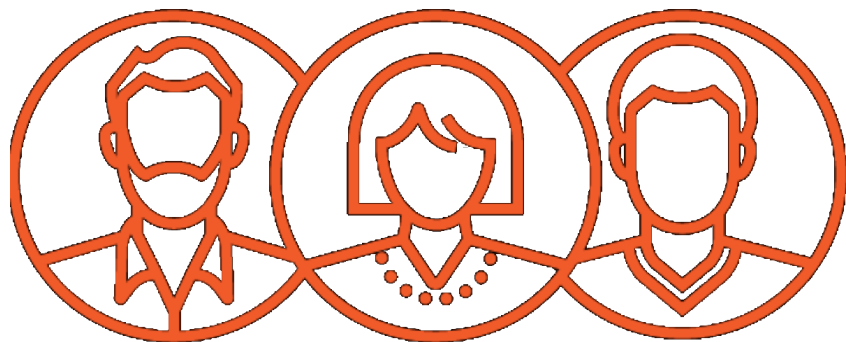
Comprar versus construir.

El software comercial se actualiza de 3-5 años.

El software a medida es utilizado en una área funcional del negocio



Mentalidad de proyectos



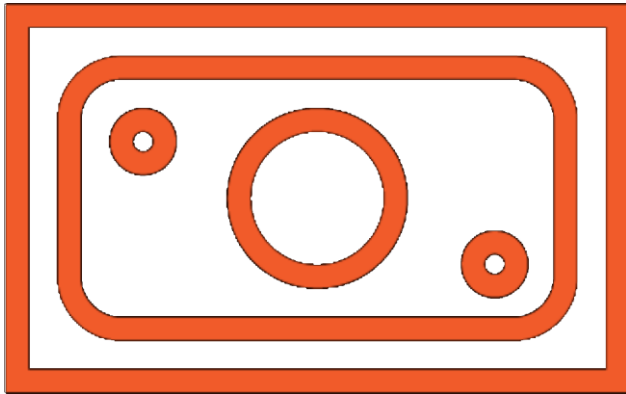
Financiación de proyectos.

Creación de equipos.

Después de la implementación, el equipo se dispersa y la aplicación pasa a manos de operaciones.

El único conocimiento relacionado es la documentación.

Optimización de costos sobre tiempos



Las áreas de TI le dan prioridad al ahorro de costos.

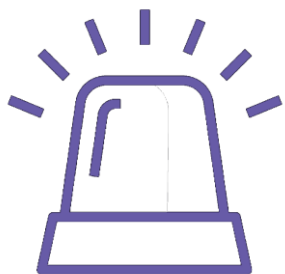
Enfoque en la maximización de eficiencia de TI.

Propuesta de valor, estabilidad, predictibilidad sobre capacidad al cambio.

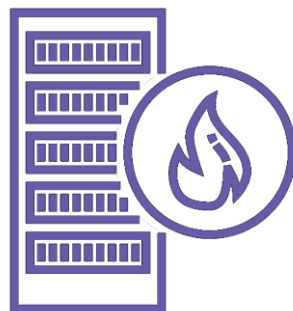


Problemáticas más comunes y desperdicios producidos

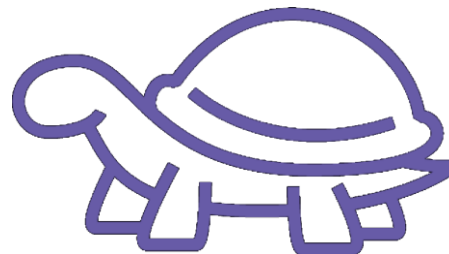
Dolor: Interrupciones graves



Pánico cuando
los sistemas se
caen



El mismo error
en diferentes
áreas



Recuperación
lenta es la ley



Perdida de
confianza en TI

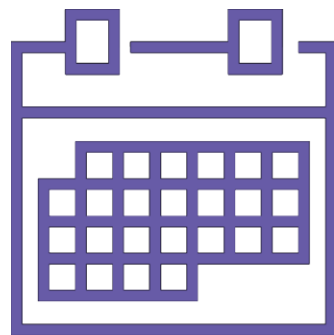


Problemáticas más comunes y desperdicios producidos

Dolor: Sin incremento de valor



Proyectos largos
con mucho
trabajo inicial



Despliegues
poco frecuentes
y de gran tamaño



No existe
experimentación

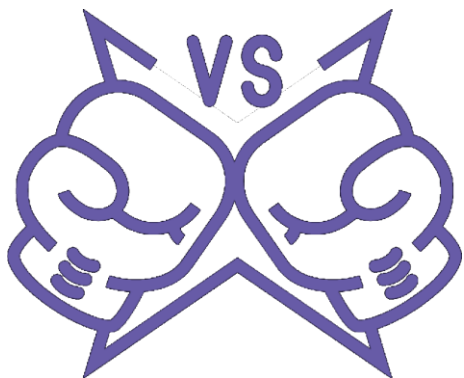


Las áreas de negocio (Line of
Business) buscan innovación
en otros lugares

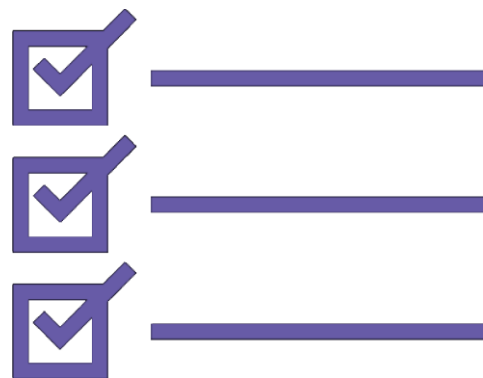


Problemáticas más comunes y desperdicios producidos

Dolor: Luchar internas



“Nosotros vs Ellos” – Poca transparencia



El área de TI tiene sus propias motivaciones



Trabajo no enfocado al cliente



Perdida de personas con talento debido a una mala cultura



¿Qué son los desperdicios?

Es todo aquello que nos demanda tiempo y no agrega valor para el cliente



Desperdicios de conocimiento

Producido por la
interrupción del **flujo**
de conocimiento

Producido por
interrumpir la
absorción de
conocimiento

Ejemplo: barreras
físicas e informes
demasiado complejos.



Desperdicios de espera

A menudo causado
por un procesamiento
desigual

Ocurre con personal
altamente
especializado.

Ejemplo: esperar
entornos o
aprobaciones antes de
implementar



Desperdicios de procesamiento excesivo

Se trata de hacer más trabajo del que se requiere o necesita

Iniciar el trabajo sin tener entendimiento del proceso y lo solicitado

Ejemplo: Software de ingeniería excesiva o especificaciones de diseño demasiado largas



Desperdicios de movimiento

Movimiento
innecesario de
personas o máquinas.

Puede solucionarse
reorganizando
físicamente o
actualizando la
tecnología

Ejemplo: doble
entrada de datos o
pruebas manuales
repetitivas



Desperdicios de transporte

Se trata de mover cosas sin agregar valor

“Siempre lo hemos hecho de esa manera”

Ejemplo: Mover documentos de aprobación de sellos de goma entre ejecutivos



Desperdicios de corrección

El producto no es utilizable y se requiere un nuevo trabajo

A menudo ocurre cuando la calidad no está integrada en el proceso

Ejemplo: tener "pruebas" de software como una fase distinta

Desperdicios de inventario

Las cosas se acumulan
y no se dan cuenta del
valor

Provoca un
procesamiento
irregular y agrupa
problemas ocultos.

Ejemplo: código
completado en espera
de ser desplegado



Desperdicios de sobre producción

Esto sucede cuando
produce más de lo
necesario

A veces se hace para
mantener ocupado al
personal o las
máquinas

Ejemplo: crear
entornos de prueba
para cada equipo



¿Qué quieres lograr?

Incrementar el valor
de entrega al cliente

Confianza en el
equipo



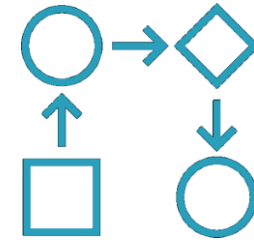
¿De qué se trata Lean?



Enfoque en dar valor al cliente



Eliminar desperdicios



Flujo de una sola pieza



Aprendizaje continuo y compartido



Eliminar la sobrecarga de personas



Teoría de las Restricciones



DevOps

C.A.L.M.S.



DevOps

C.A.L.M.S.

Cultura. Personas y procesos a la vanguardia.



DevOps

C.**A**.L.M.S.

Automatización. Repetibilidad al automatizar actividades.



DevOps

C.A.**L**.M.S.

Lean. Aplicando principios lean a I.T



DevOps

C.A.L.**M**.S.

Metricas. Mida ampliamente y use para mejorar



DevOps

C.A.L.M.S.

Compartir (Sharing). Aprendizaje continuo al compartir conocimiento.



¿DevOps Funciona?

2604 veces más rápido para recuperarse de incidentes

46 veces más frecuentes en despliegues de código

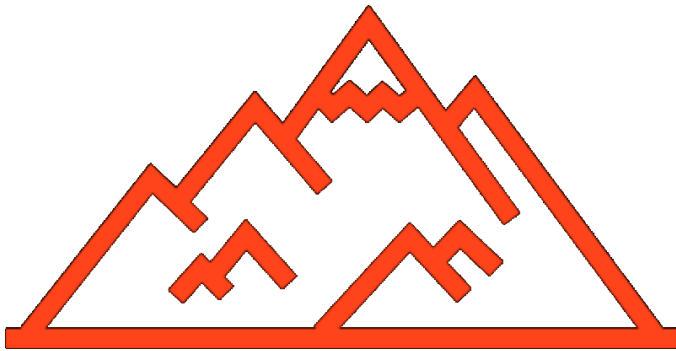
1.53 veces más probabilidades de cumplir o superar los objetivos comerciales

1.8 veces más probabilidades de recomendar su equipo a otros

Fuente: 2018 Accelerate State of DevOps Report



Cambio cultural - Restablecer los objetivos



¿Cuáles son tus objetivos compartidos?

¿Todos empatizan con el cliente?

¿Estás pasando por los movimientos (ágiles)?

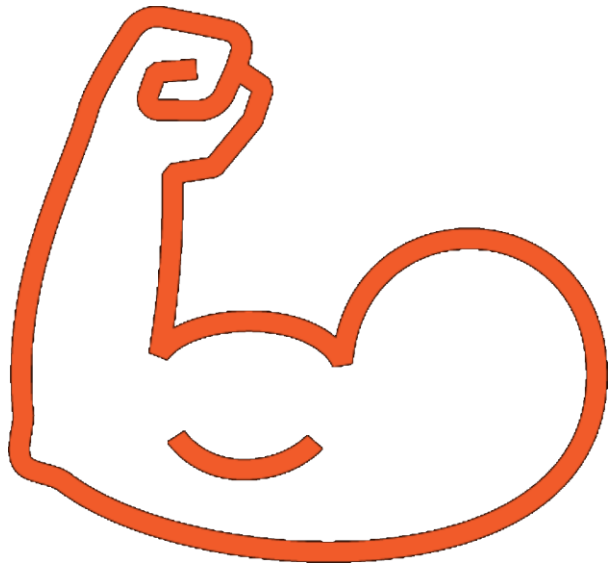
La desalineación aquí condenará tu esfuerzo



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio cultural - Elevar el empoderamiento



Confía en el equipo, déjalos tomar posesión

Los empleados hacen lo necesario para cumplir el servicio.

Cualquiera puede "tirar del cordón"

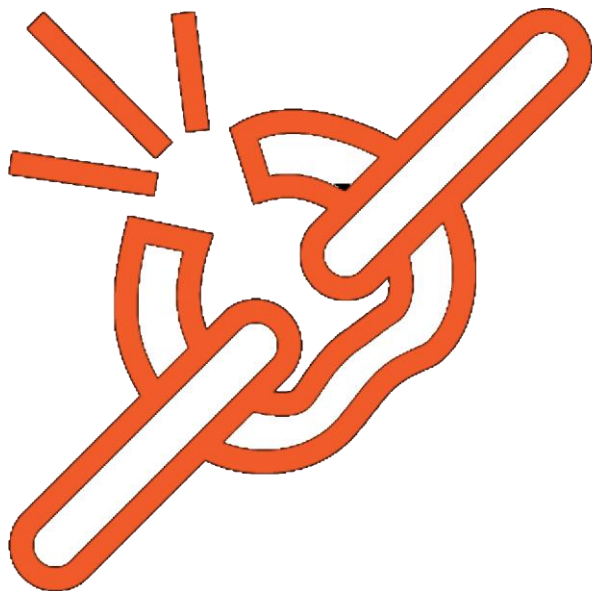
Se forman equipos durante la crisis para resolver problemas



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio cultural - Promover la responsabilidad



No se puede tener empoderamiento sin responsabilidad

Calidad integrada en el proceso

Compromiso compartido con la excelencia

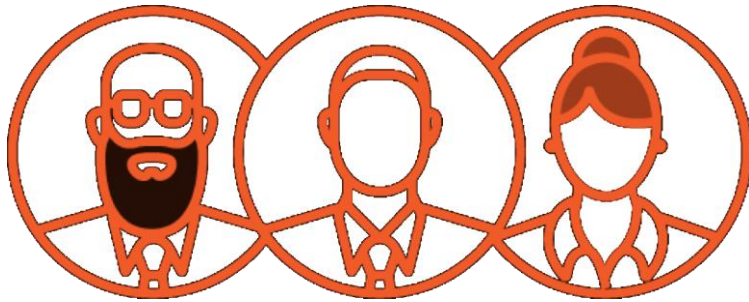
Recompense a quienes entregan y asuman la responsabilidad



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio cultural - Se requiere trabajo en equipo



Todas las disciplinas trabajan juntas, en crisis o en operaciones diarias.

Respetar las habilidades únicas

Encuentra excusas para unir al equipo

No recompense al "genio solitario" o al bombero



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio cultural - Fomentar el aprendizaje continuo



Fomentar el aprendizaje continuo

Ofrecer recursos y tiempo para aprender.

Fomentar la experimentación.

Realizar postmortems sin culpas

Facilita compartir y descubrir información



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio cultural - Comparta, viva y recompense los valores correctos



No solo digas tus valores, vívelos

¿Qué desencadena la promoción o bonificaciones?

Los gerentes deben liderar con el ejemplo



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio organizacional - Comprenda a sus clientes y flujos de valor

Desarrollar y valorar la empatía para los usuarios.

Retroceda y "vea" los sistemas en su lugar

Realice ejercicios de mapeo de flujo de valor para enfocar sus esfuerzos de mejora



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio organizacional – Enfocarse en los cuellos de botella

Es ineficaz optimizar en cualquier lugar, excepto en las restricciones (problemas)

Una restricción puede ser interna (por ejemplo, personas, procesos) o externa (por ejemplo, demanda limitada)

La teoría de las restricciones TOC (Theory of Constraints) lo ayuda a identificar y eliminar estos factores limitantes



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio organizacional – Acabar con entornos inconsistentes

Se aplica a diferentes entornos (por ejemplo, desarrollo, control de calidad), pero también dentro de cada entorno

¿Resultado de este cuello de botella?
Versiones de software más lentas y con problemas

Apuntar a esfuerzo bajo demanda



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio organizacional – Automatizar los procesos de producción

¿Qué tan rápido puede desplegar el código?
¿Se ha acumulado inventario?

La codificación o las pruebas más rápidas no lo ayudarán si el proceso de despliegue no está optimizado

La solución implica una integración continua y pipelines integrales y automatizados



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio organizacional – Construir con calidad por adelantado

Los equipos de control de calidad no son responsables de la calidad.

El compromiso compartido con la excelencia significa probar temprano y con frecuencia

Logrado a través de prácticas como TDD e integración continua



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Cambio organizacional – Romper las barreras de la comunicación

La mala comunicación da como resultado sistemas push, no basados en pull

Los circuitos de retroalimentación son críticos para el contexto y la propiedad compartida

Resuelto a través de una combinación de herramientas, prácticas e intenciones sinceras de colaboración.



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



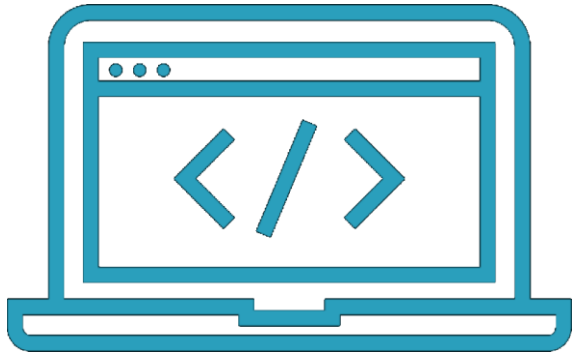
"DevOps es una reorganización, no un nuevo equipo para contratar".

Adrian Cockcroft



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).

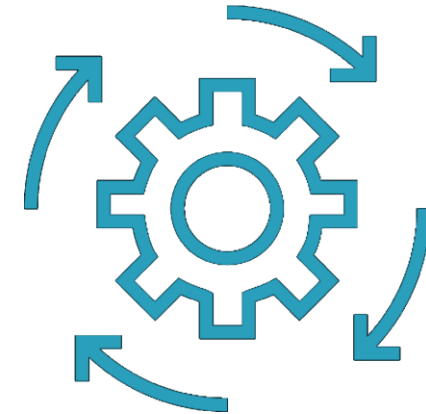
Definiendo objetivos – "DevOps nos hace mas seguros"



Desarrollos y despliegues mas rapido



Mejorar los lineamientos de seguridad



La automatización significa menos errores manuales y una resolución de vulnerabilidad más rápida



Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Intermedio

Definiendo objetivos – “Generación de equipos”



La ubicación conjunta de los equipos



Cultura compartida, valores e incentivos



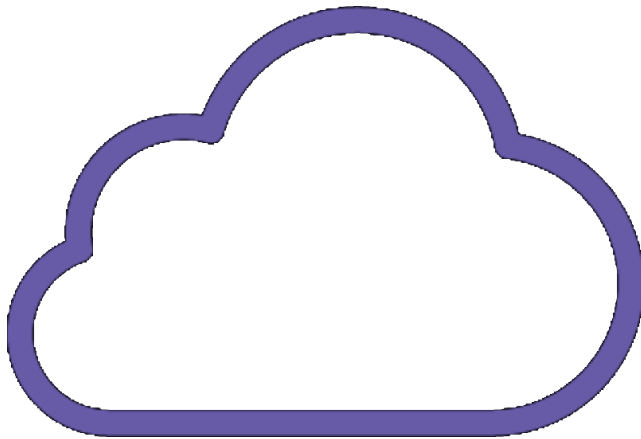
La naturaleza de las asociaciones subcontractadas tiene que cambiar



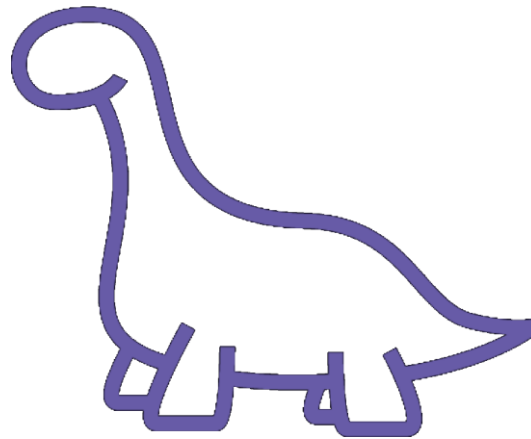
Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



Definiendo objetivos – “Tengo demasiados COTS y código heredado”



Los sistemas deben ser orientados hacia nube



TDD, integración continua, automatización de despliegues es posible con entornos heredados



Recuerde que el punto no es solo empujar el código más rápido, sino mejorar el flujo



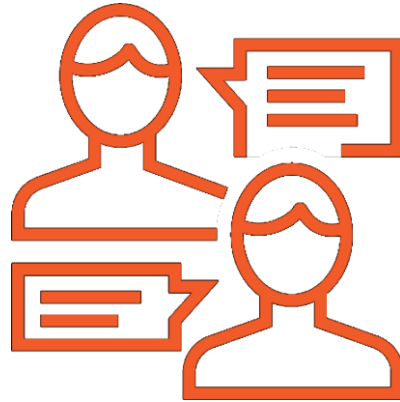
Caminando hacia DevOps (cambio cultural, cambio organizacional, definiendo objetivos).



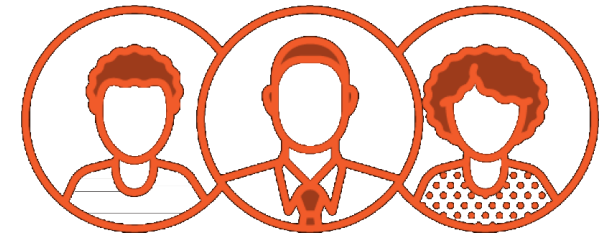
Definiendo objetivos – “Implementación”



Una inversión que vale la pena hacer



Muchas de estas son habilidades "blandas"

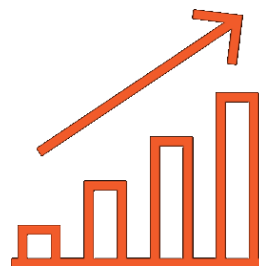
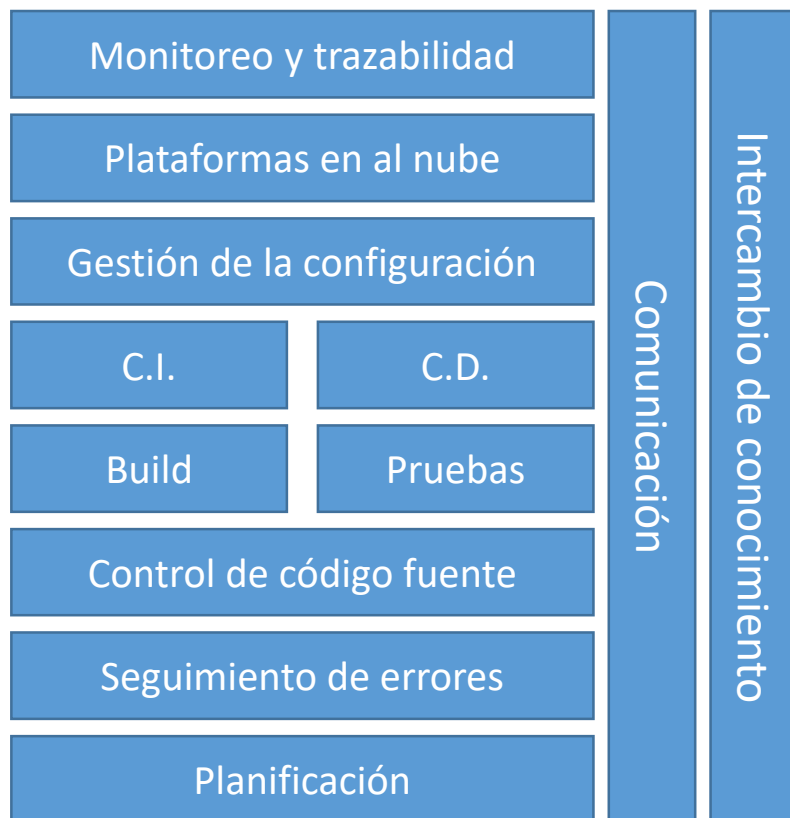


Haz que sea grupal y aplica inmediatamente



Automatización (tecnologías y plataformas en la nube).

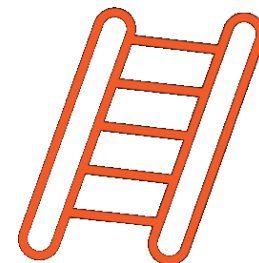
Herramientas DevOps



El objetivo es la mejora continua.



No puedes "comprar" DevOps

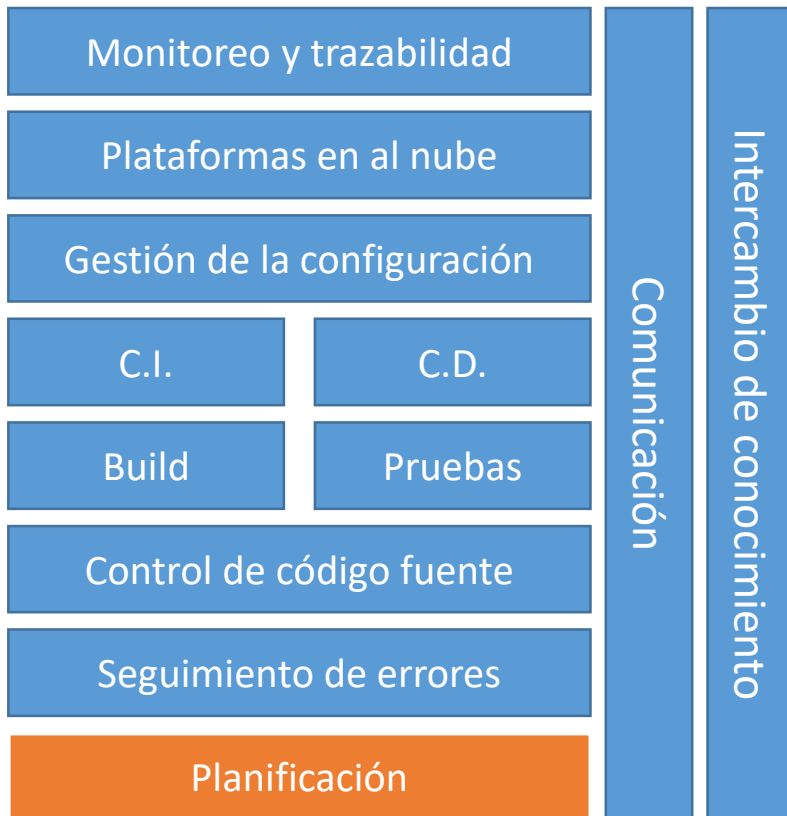


La adopción de herramientas ocurre en etapas



Herramientas DevOps

Por qué es importante para DevOps: propósito compartido, transparencia, empoderamiento



GitLab

TaskTop

CollabNet
VersionOne

Pivotal Tracker

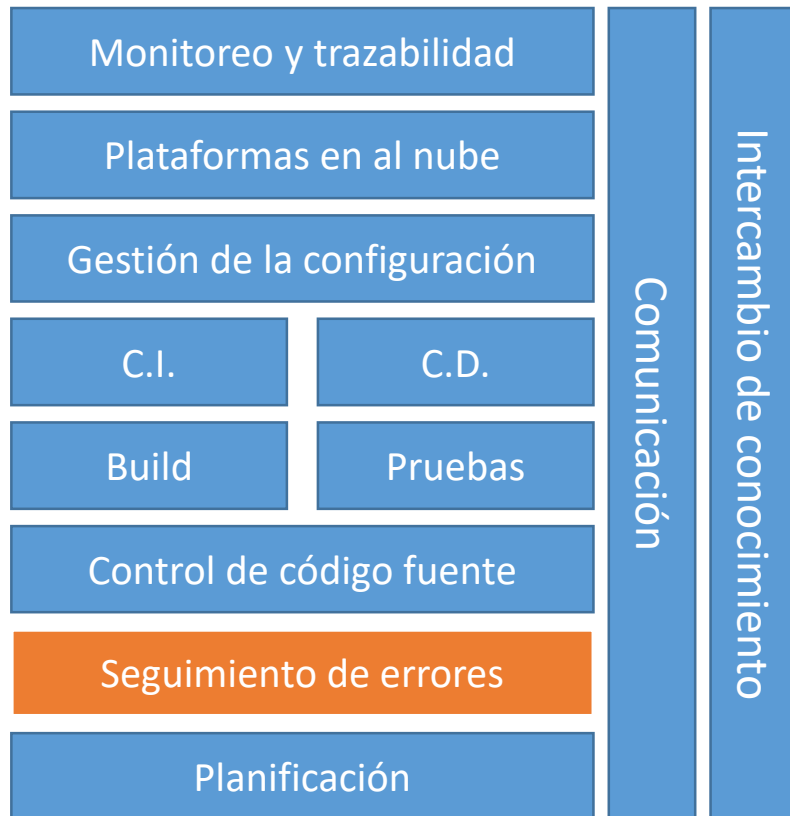
Trello

Azure Boards



Herramientas DevOps

Por qué es importante para DevOps: capacidad de respuesta del cliente, pérdida limitada de conocimiento, bucle de retroalimentación



Atlassian Jira

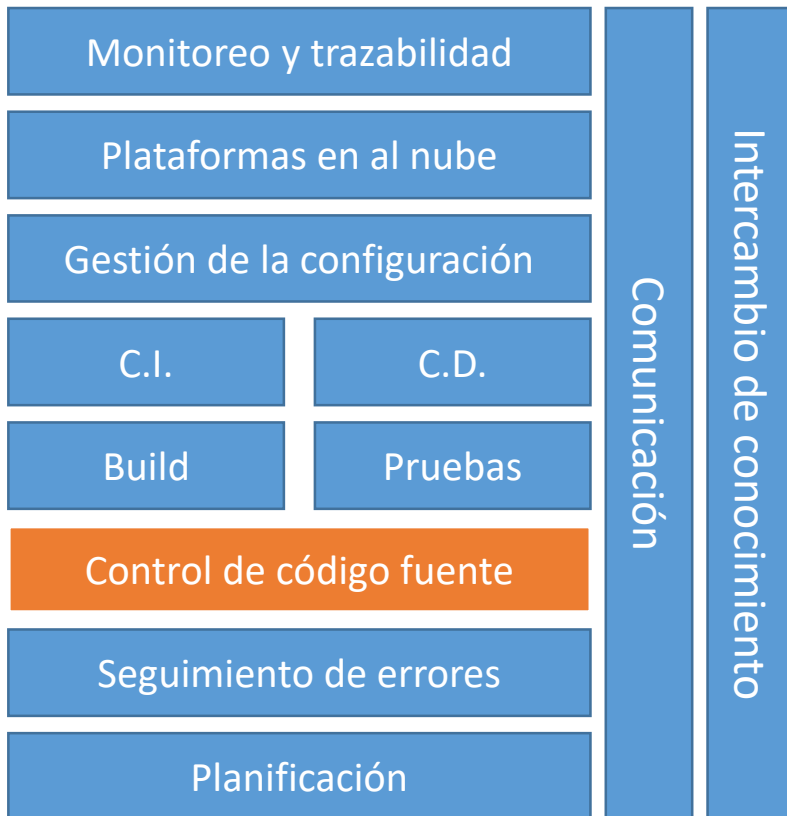
JetBrains
YouTrack

ZenDesk



Herramientas DevOps

Por qué es importante para DevOps: el control de activos limita el desperdicio de transporte y capacita a los equipos



git

GitHub

GitLab

Bitbucket

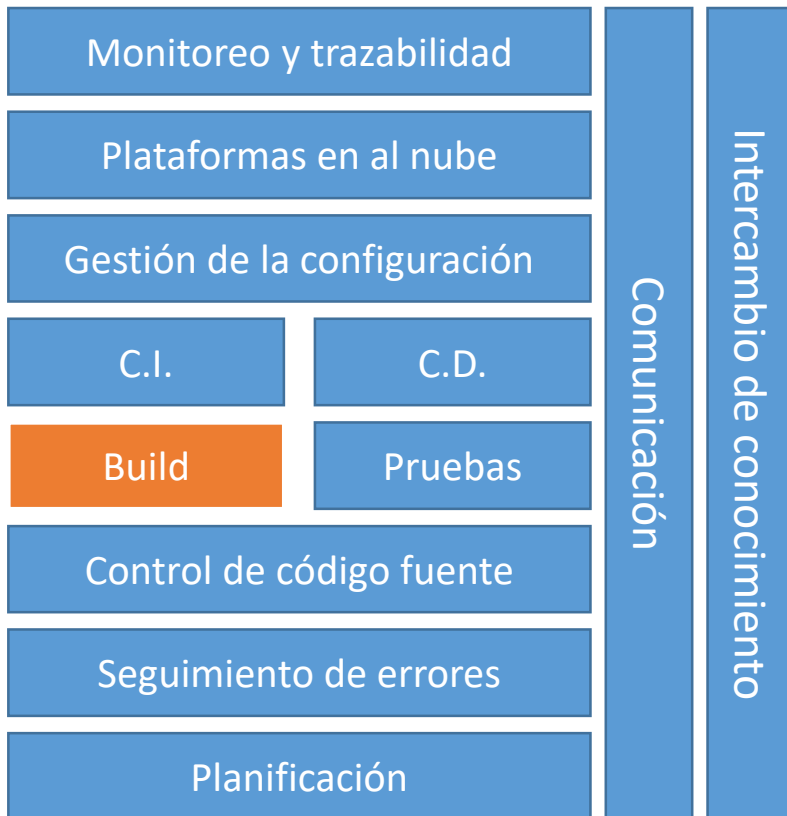
Subversion

TFS



Herramientas DevOps

Por qué es importante para DevOps: el empaquetado consistente, automatiza una actividad propensa a errores y genera señales de calidad tempranas



Maven/Grade

MSBuild

Rake

JFrog Artifactory

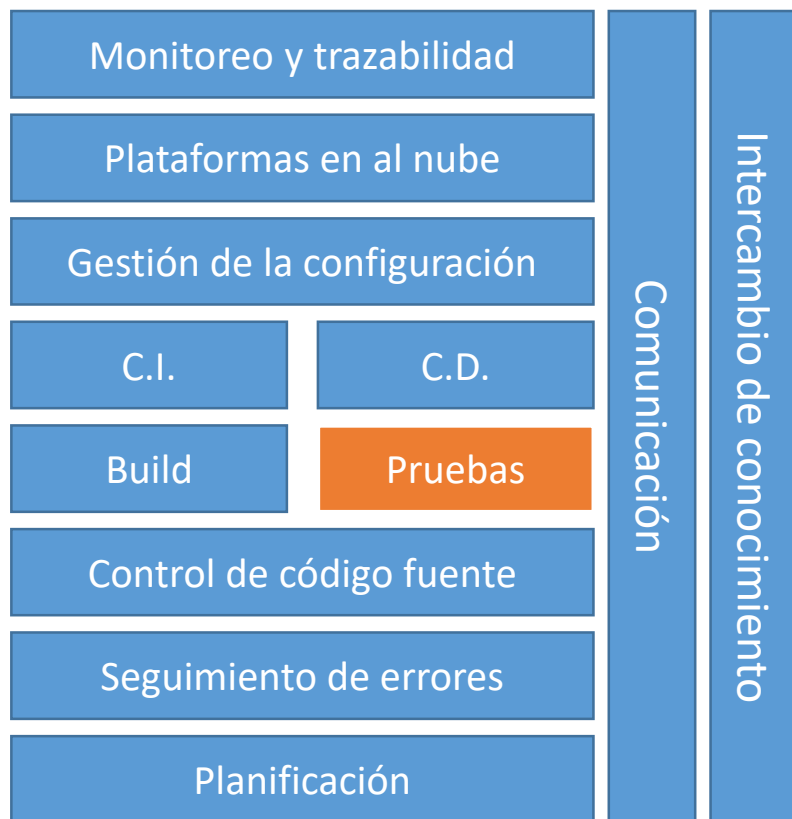
Sonatype

NuGet



Herramientas DevOps

Por qué es importante para DevOps: se centra en la calidad incorporada, crea confianza en los artefactos desplegados



JUnit

XUnit.net

Cucumber

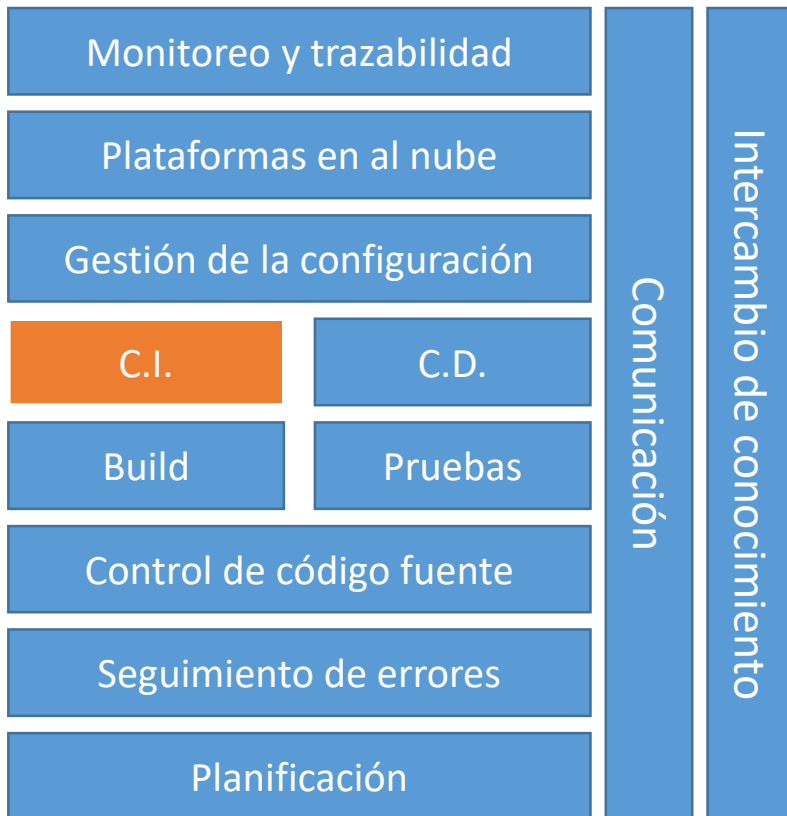
Selenium

Jasmine



Herramientas DevOps

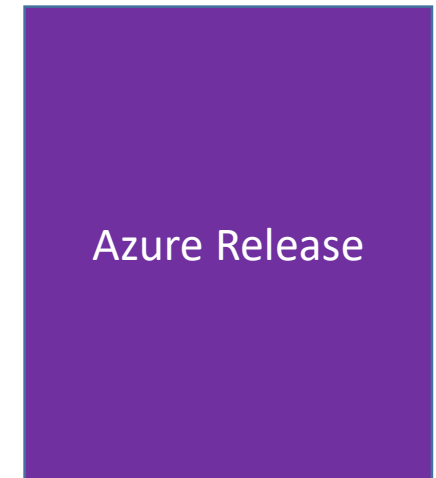
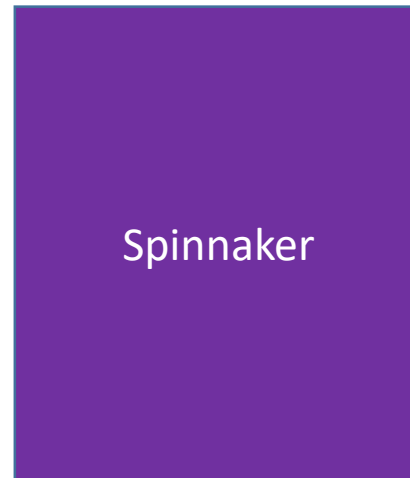
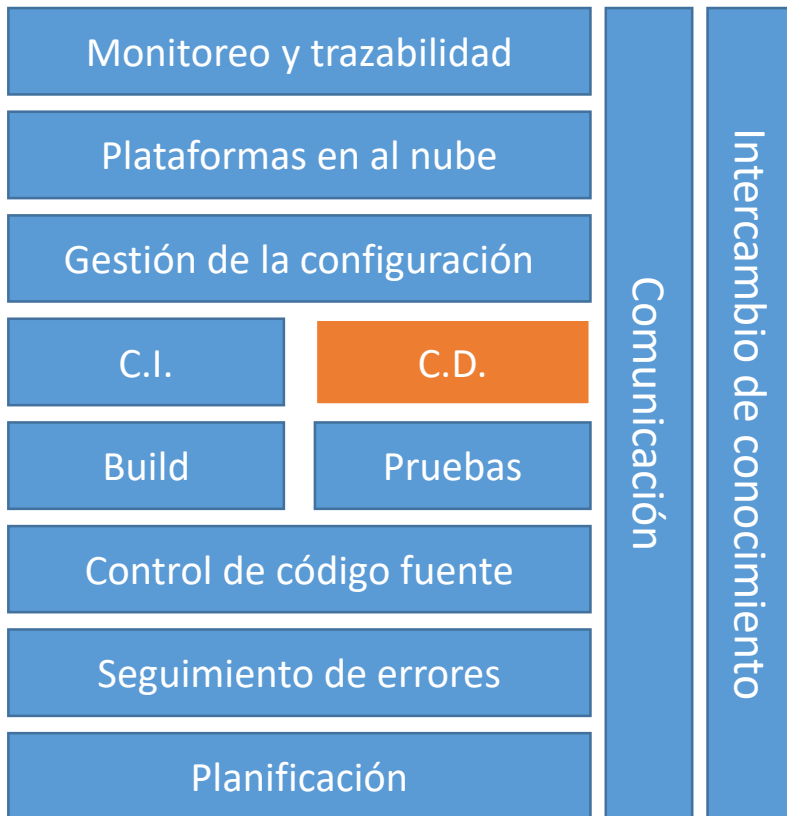
Por qué es importante para DevOps: retroalimentación rápida, reducir el desperdicio de defectos y el desperdicio de espera





Herramientas DevOps

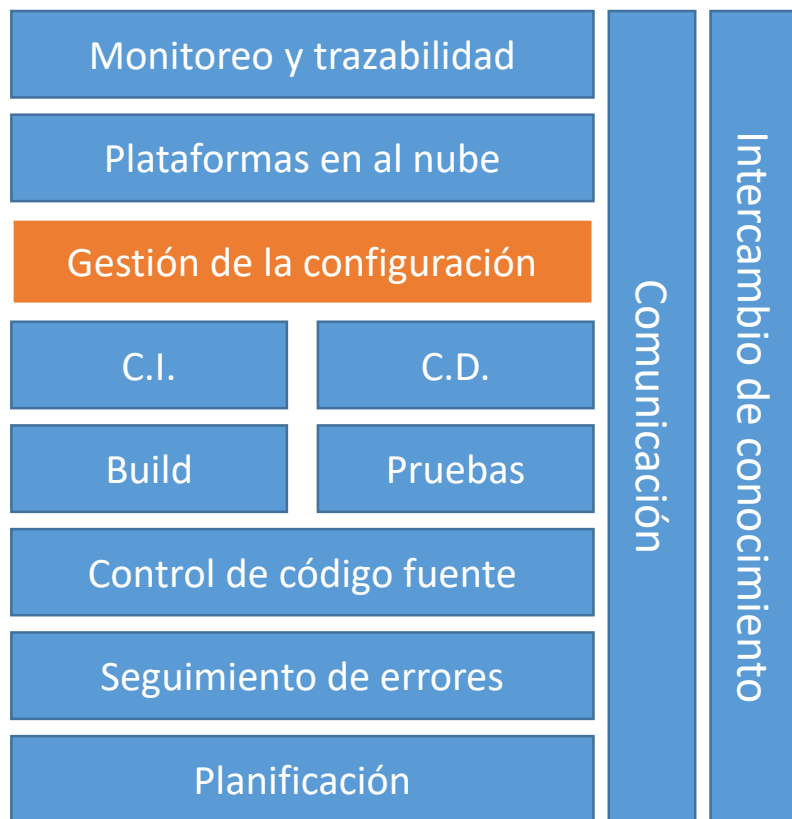
Por qué es importante para DevOps: limite su inventario de preproducción, automatice pipeline complejos, unifique al equipo para obtener valor para los clientes





Herramientas DevOps

Por qué es importante para DevOps: imponga la coherencia, trate la infraestructura como código



Terraform

BOSH

Chef

Ansible

Puppet

Google Cloud
Deployment
Manager



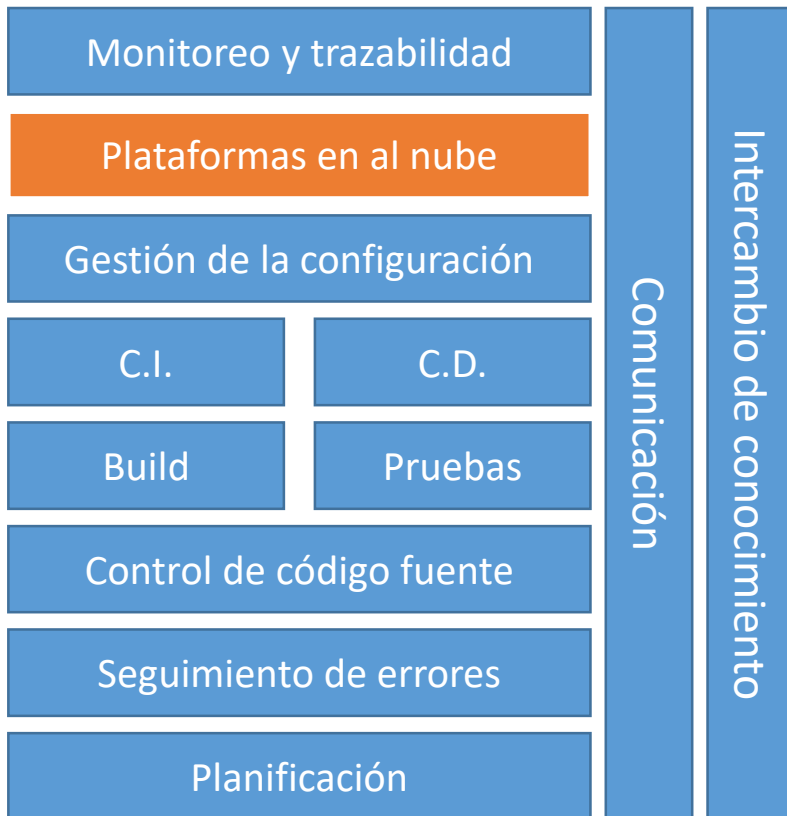
Automatización (tecnologías y plataformas en la nube).



Intermedio

Herramientas DevOps

Por qué es importante para DevOps: Automatización amigable



Amazon Web Services

Microsoft Azure

Google Cloud Platform

Pivotal Cloud Foundry

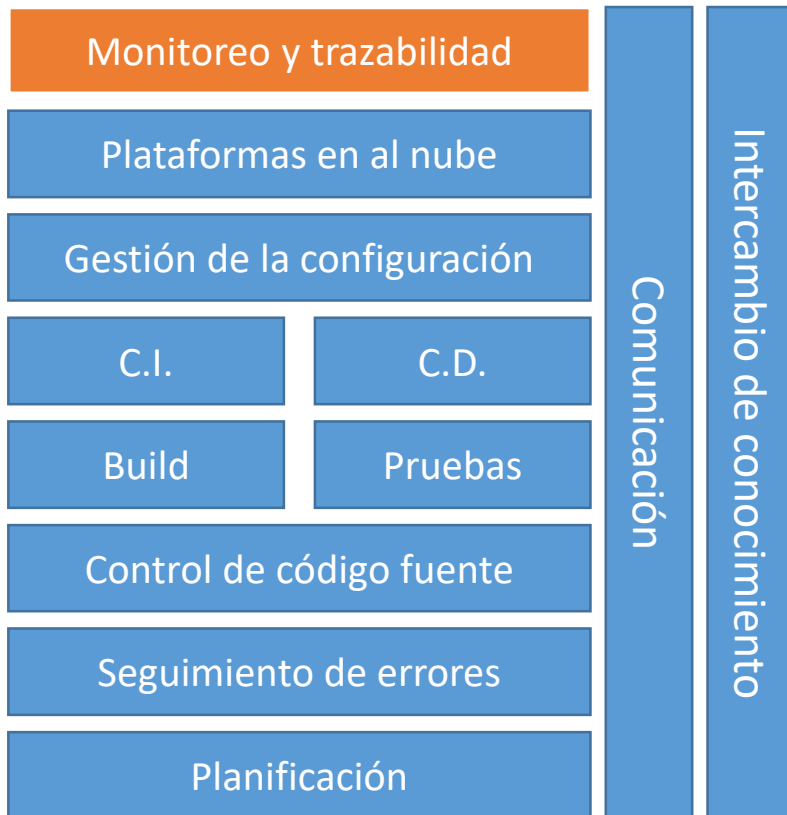
Heroku

Container Schedulers



Herramientas DevOps

Por qué es importante para DevOps: recuperación rápida, capacidad de respuesta, transparencia, participación humana limitada durante los incidentes



ELK Stack

Datadog

New Relic

Prometheus

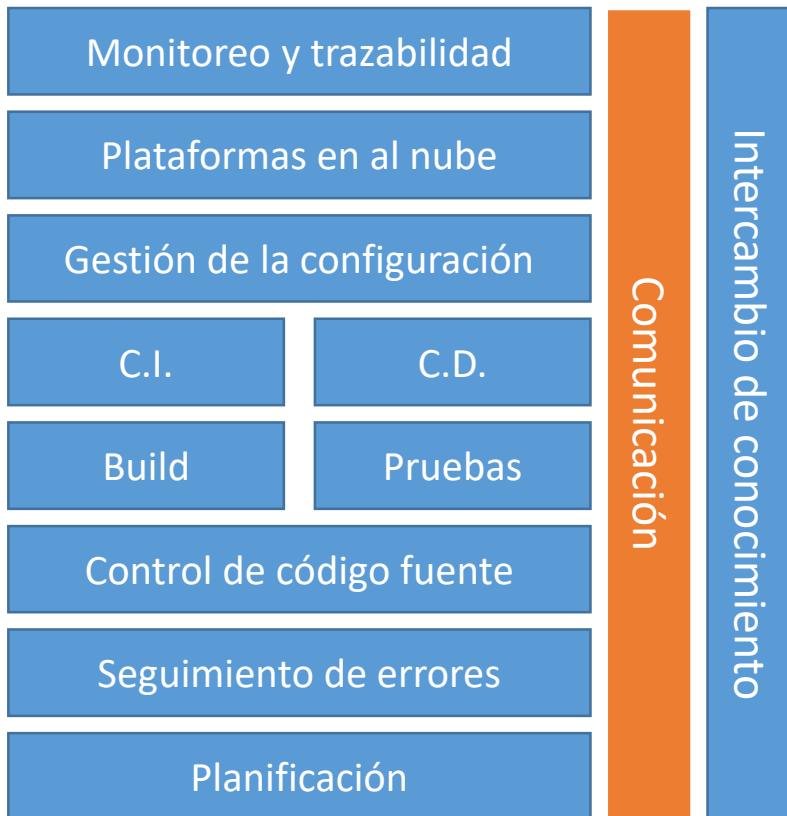
Zipkin

Azure Monitor



Herramientas DevOps

Por qué es importante para DevOps: conectar equipos, limitar el desperdicio de espera, mejorar la colaboración



Slack

Microsoft Teams

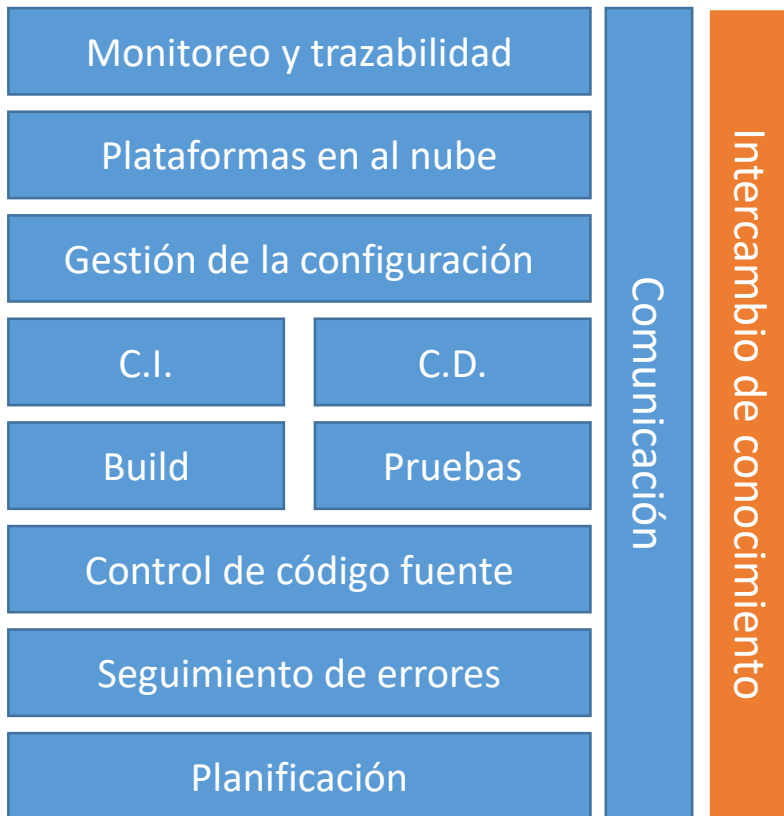
Google Hangouts

Zoom



Herramientas DevOps

Por qué es importante para DevOps: reduzca el desperdicio de conocimiento, aumente la productividad de los nuevos empleados, limite los errores repetidos



GitHub Pages

Confluence

Jekyll

Google Sites



¿Qué es Azure DevOps?

Es un conjunto de herramientas y servicios para ayudarlo a desarrollar y entregar software

Comparta código, realice un seguimiento del trabajo y envíe software.

Planifique de manera más inteligente, colabore mejor y envíe más rápido con un conjunto de servicios de desarrollo modernos.

<https://visualstudio.microsoft.com/tfs/>
<https://azure.microsoft.com/en-us/services/devops/>



Muchos cambios enormes en 2018

Cambió la interfaz de usuario en VSTS

Cambio de nombre de Visual Studio Team Services a Azure DevOps

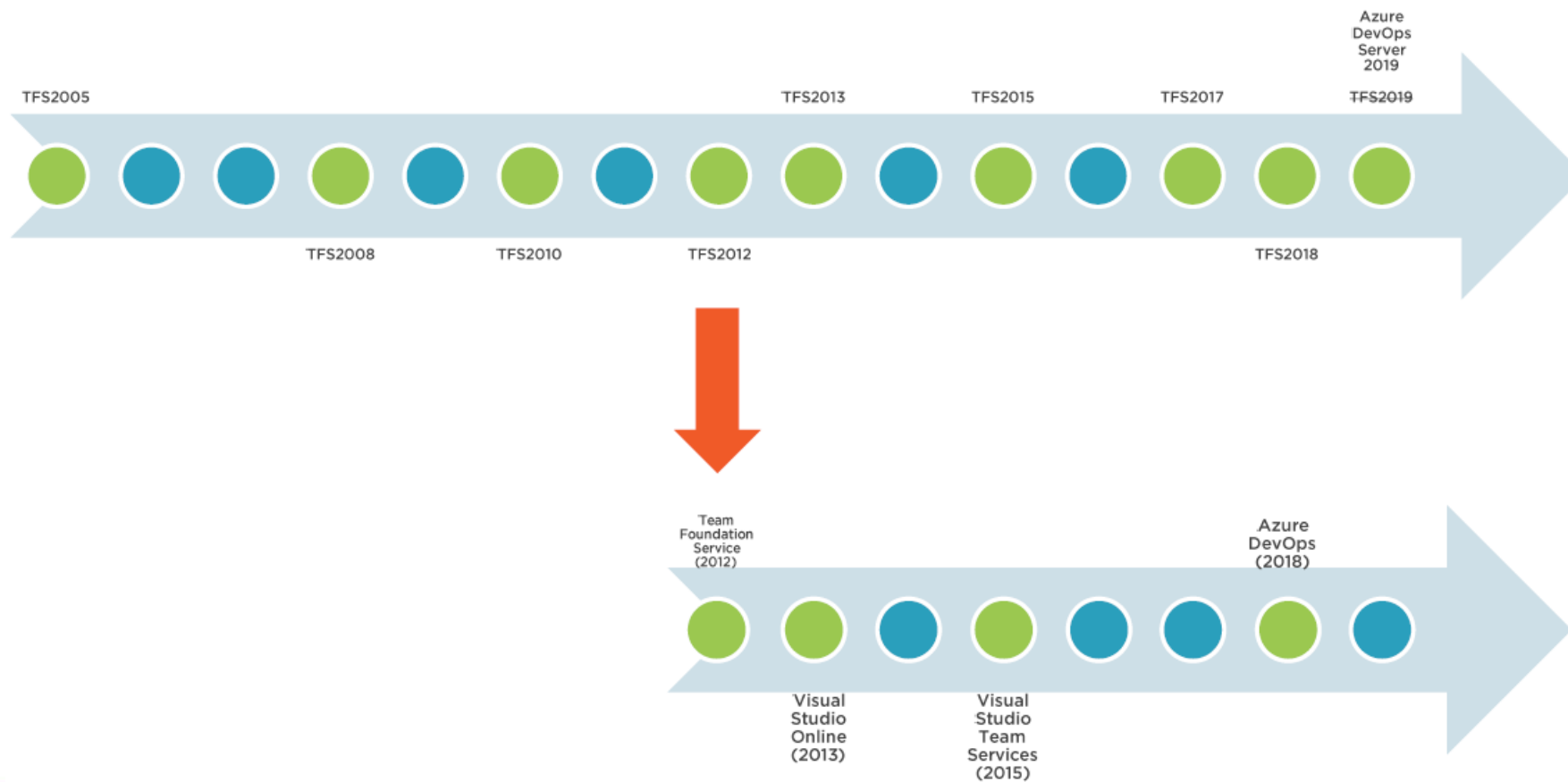
Renombrado de todas la piezas y componentes de VSTS

Cambio de nombre de Team Foundation Server a Azure DevOps



Azure DevOps y ¿Por qué usarlo?

Historia





¿Por qué usar Azure DevOps?

Fáciles de instalar, TFS o Azure DevOps Server

... pero hay gastos generales de licencias, costos de hardware, actualización de servidores, configuración de discos y performances, backups, servidores de build y release, acceso remoto.

No es imposible pero hay muchos detalles



¿Por qué usar Azure DevOps?

A comparación de Azure DevOps :

- Crea una cuenta
- Crea un proyecto
- Agrega personas a tu proyecto.
- **ES GRATIS!!**
- En menos de 5 minutos
- Máximo para 5 personas, pero para proyectos open source es ilimitados

<https://azure.microsoft.com/en-us/pricing/details/devops/azure-devops-services/>



GALAXY
TRAINING