

CALIDAD DEL SOFTWARE Y TESTING



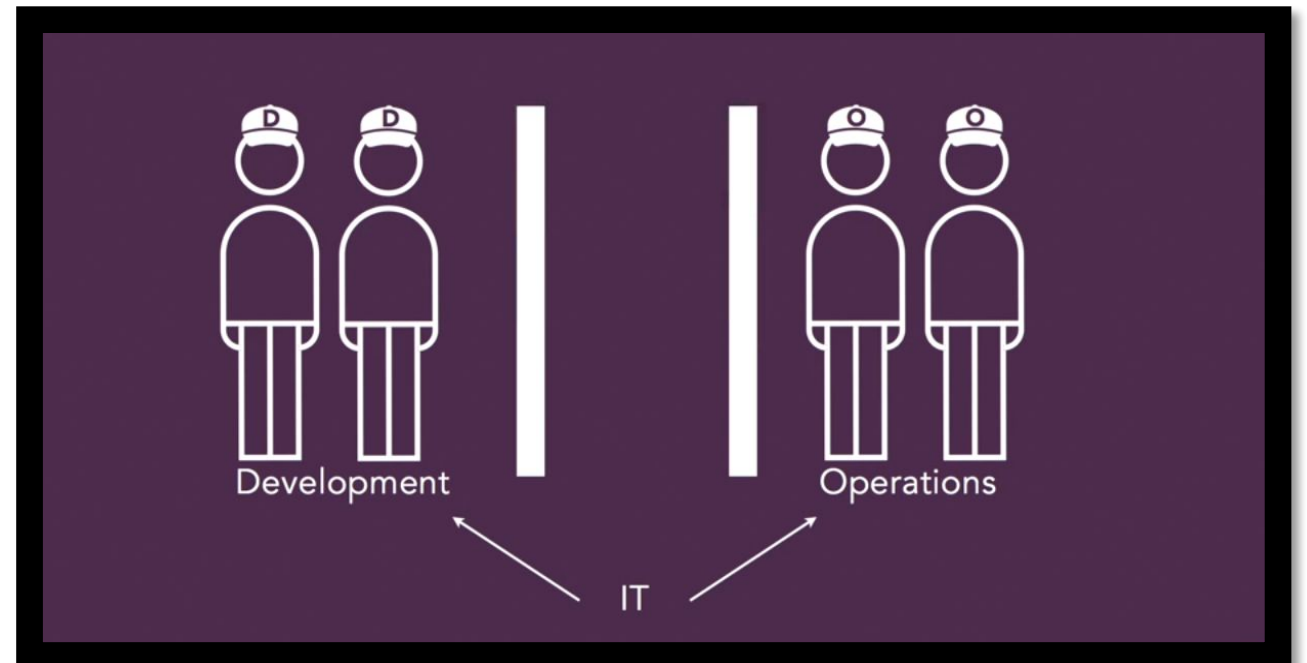
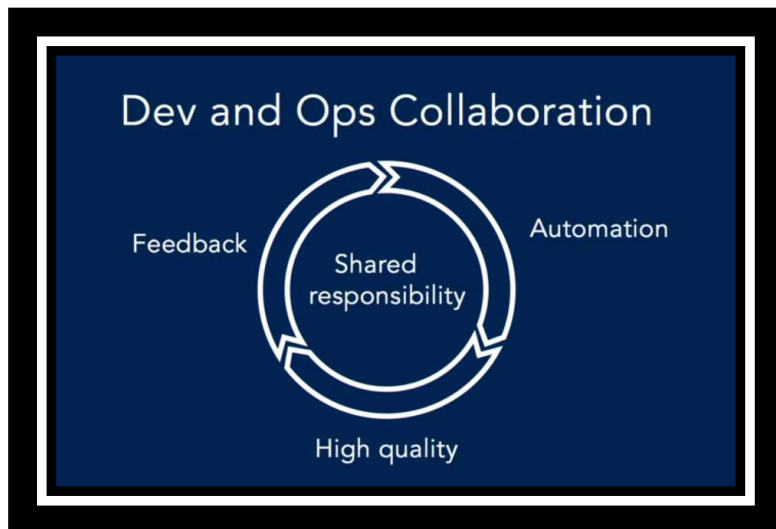


DevOps, Herramientas y Métricas:

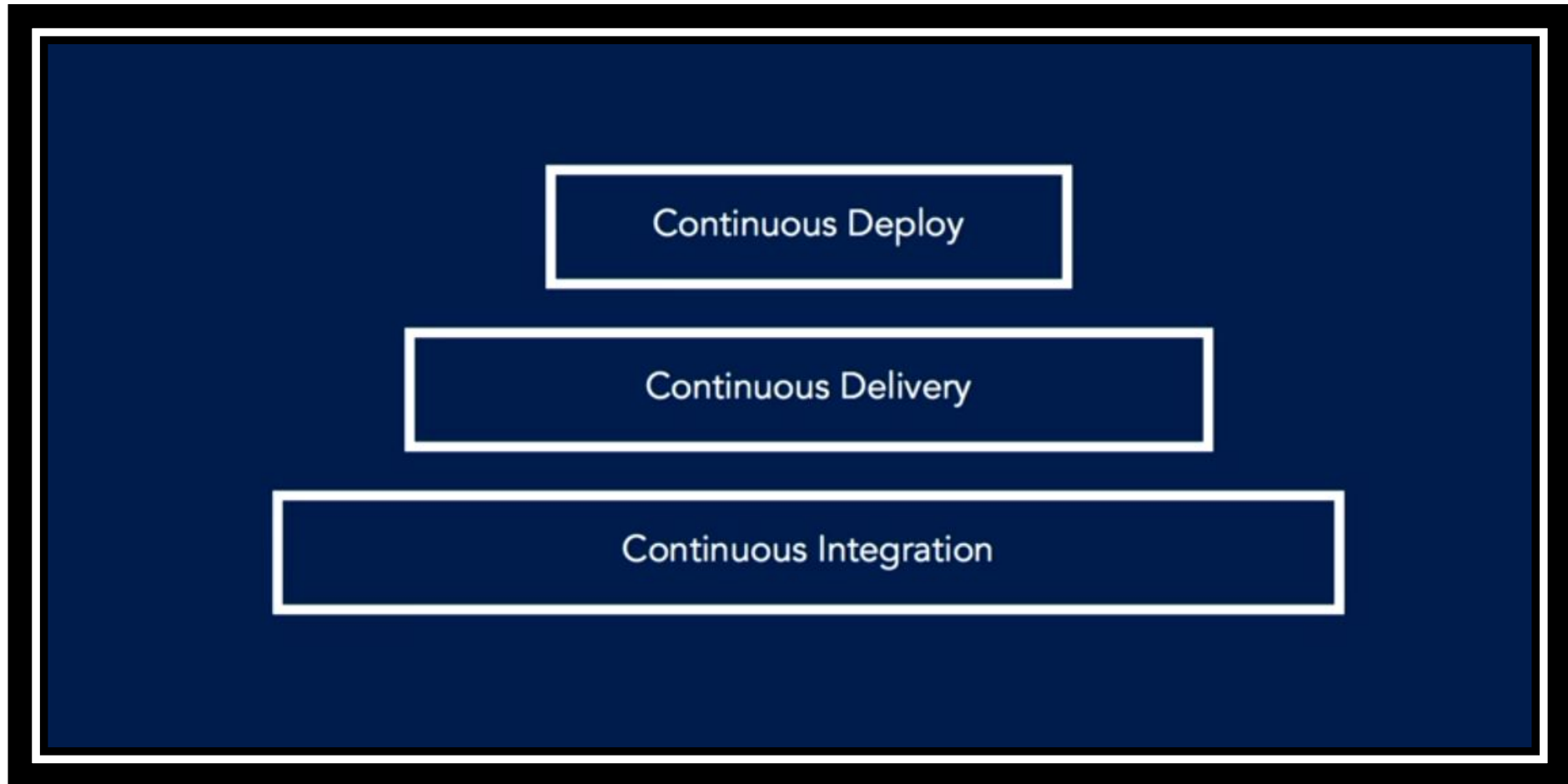
"A fool with a tool is still a fool."

DEVOPS:

- **DevOps** = *Development + Operations + other teams.*
 - *Dev* = Código. Desde *Front-End* hasta *QA*.
 - *Ops* = Sistemas. Desde *Linux Administrators* hasta *Network Admins* y *DBAs*.
- **DevOps** != Problema Tecnológico
- **DevOps** = Problema cultural
- **KEY** = Colaboración



DEVOPS: CI/CD



DEVOPS: CI/CD

▣ **Continuous Integration:** Es la práctica de, frecuentemente, construir y probar a nivel unitario la aplicación completa. Idealmente en cada entrada nueva de código.

▣ **Continuous Delivery:** Es la práctica adicional de desplegar cada cambio a un entorno similar a producción y ejecutar pruebas automáticas de integración y aceptación.

▣ **Continuous Deployment:** Extiende el concepto donde cada cambio pasa directamente por suficientes baterías de pruebas automáticas y es desplegado automáticamente en producción.

DevOps core concept: CI/CD

DEVOPS: CONTINUOUS INTEGRATION

□ En cada *commit*:

- Se construye el código.
- Se ejecutan todas las pruebas unitarias.
- Se hacen otros pasos de validación.
- Se empaqueta y se guarda la información del resultado.

□ Si las pruebas fallan:

- La construcción falla para todo el equipo.
- Nuevos arreglos relanzarían todo el proceso.

□ Esta disciplina promueve:

- Bucles de *feedback* rápido.
- Escritura de pruebas automáticas.
- Reduce el re-trabajo.

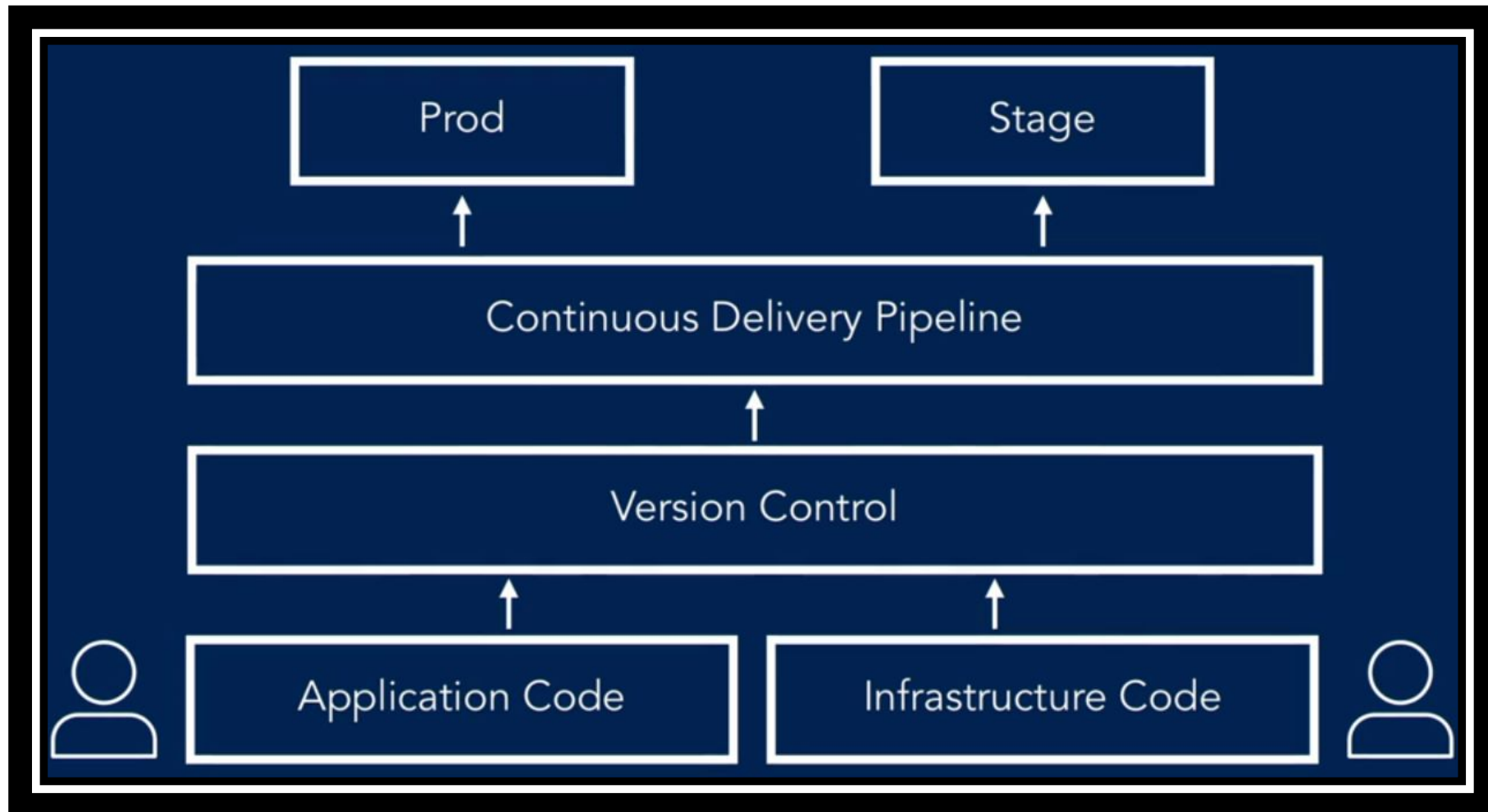
"The goal of continuous integration is that software is in a working state all the time."

—Jez Humble

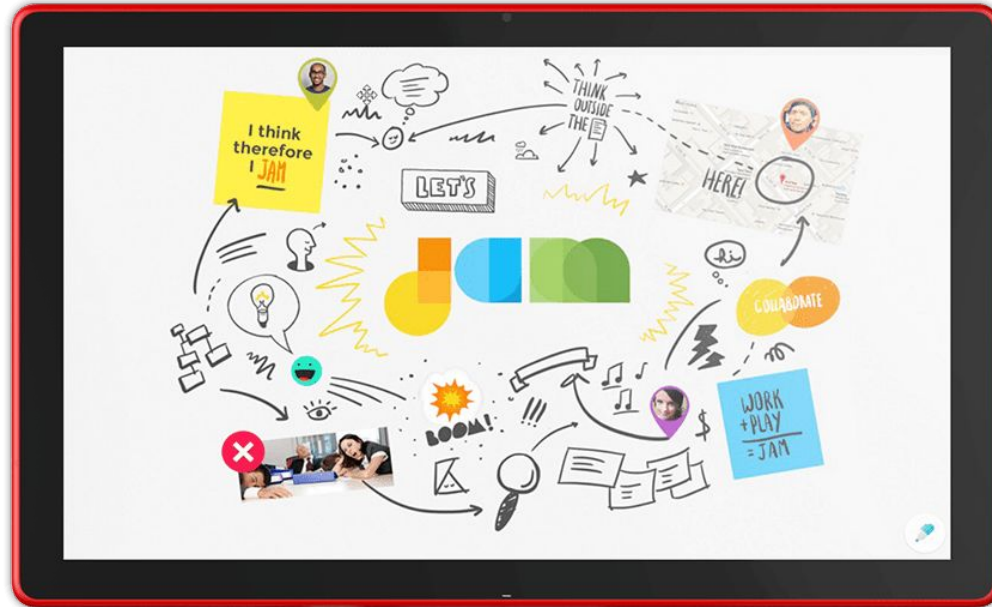
DEVOPS: CONTINUOUS DELIVERY

- Es una metodología *DevOps*.
- Práctica de desplegar cada cambio a un entorno similar a producción.
- Ejecutar pruebas automáticas de integración y aceptación.
- Usar entornos de prueba:
 - *Docker containers*.
 - *Virtual Machines*.
- El entorno de prueba permite:
 - Validar el proceso de despliegue.
 - Ver el estado de la funcionalidad y el rendimiento antes de un alcanzar un entorno real de Producción.
- Los artefactos contruidos deben ser los mismos para todos los entornos disponibles y no ser modificados. Esto consigue una mejora en:
 - Confianza entre Dev and Ops, and QA.
 - Auditoría.

DEVOPS: CI/CD

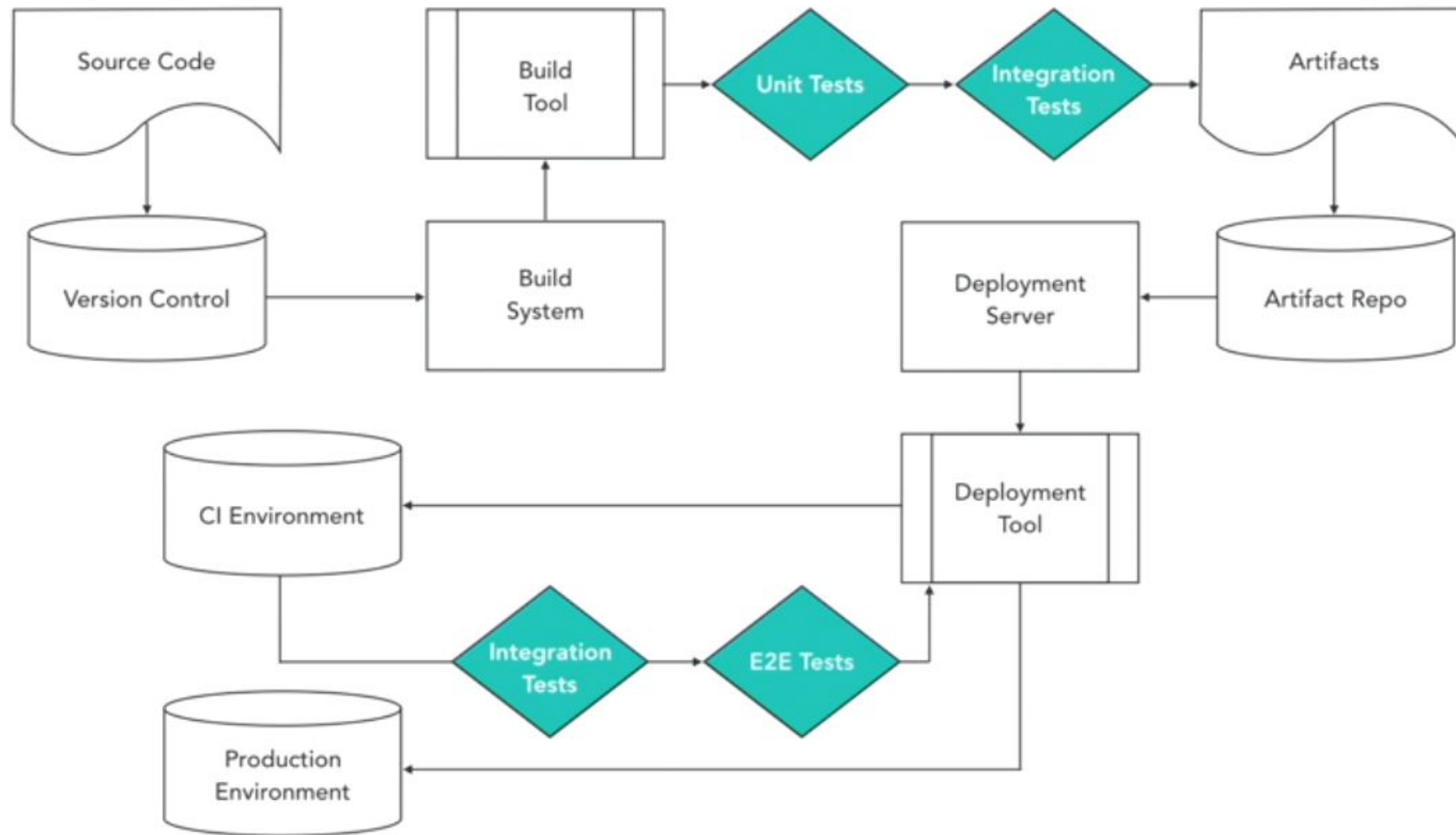


DEVOPS: CI/CD



CI/CD

DEVOPS: CI/CD



DEVOPS: Continuous Testing

Continuous testing es el proceso que permite obtener valor analizando los riesgos asociados al negocio y la entrega de la aplicación siguiendo una aproximación CI/CD.

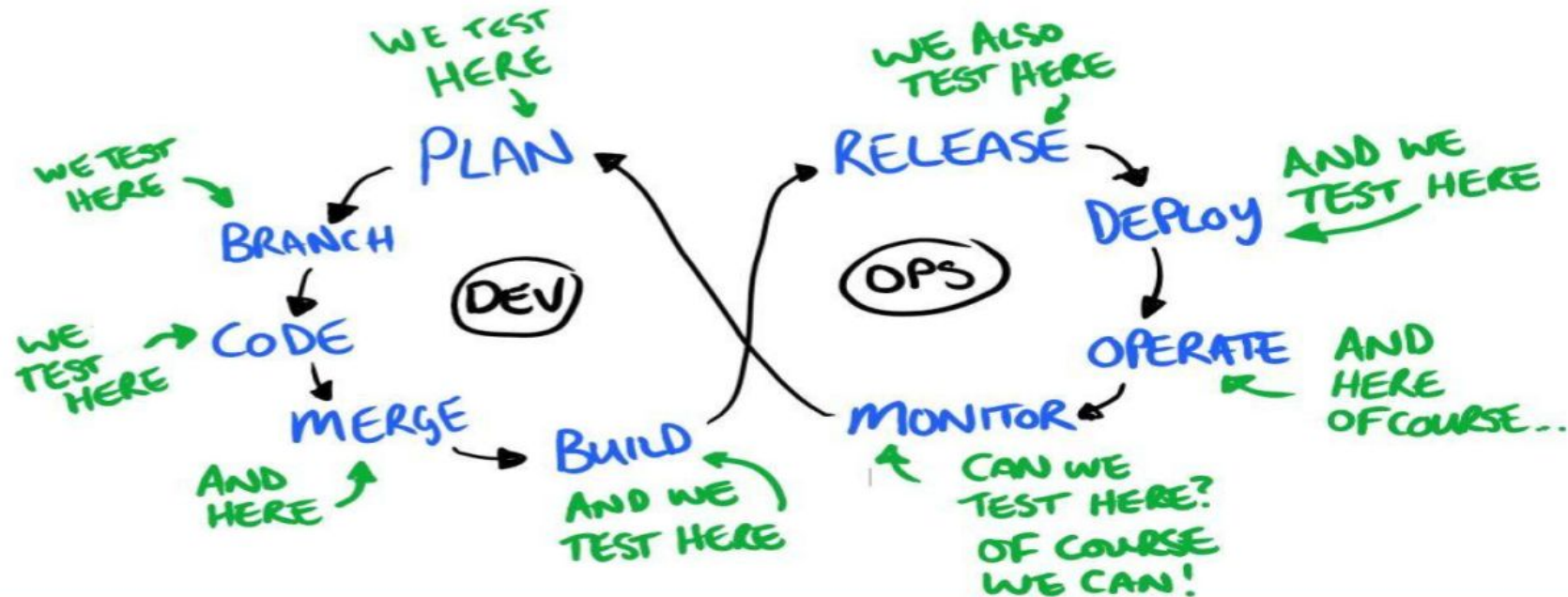
No importa si despliegas una vez al mes o cada minuto, Continuous Testing puede responder a la pregunta: "¿Estamos contentos con el valor que aporta el incremento a nuestro negocio, stakeholders o usuarios?".

"The key to building quality into our software is making sure we can get fast feedback on the impact of changes. [...] In order to build quality in to software, we need to adopt a different approach. Our goal is to run many different types of tests, both manual and automated, continually throughout the delivery process."

"Testing is a cross functional activity that involves the whole team, and should be done continuously from the beginning of the project."

DEVOPS: Continuous Testing

CONTINUOUS TESTING



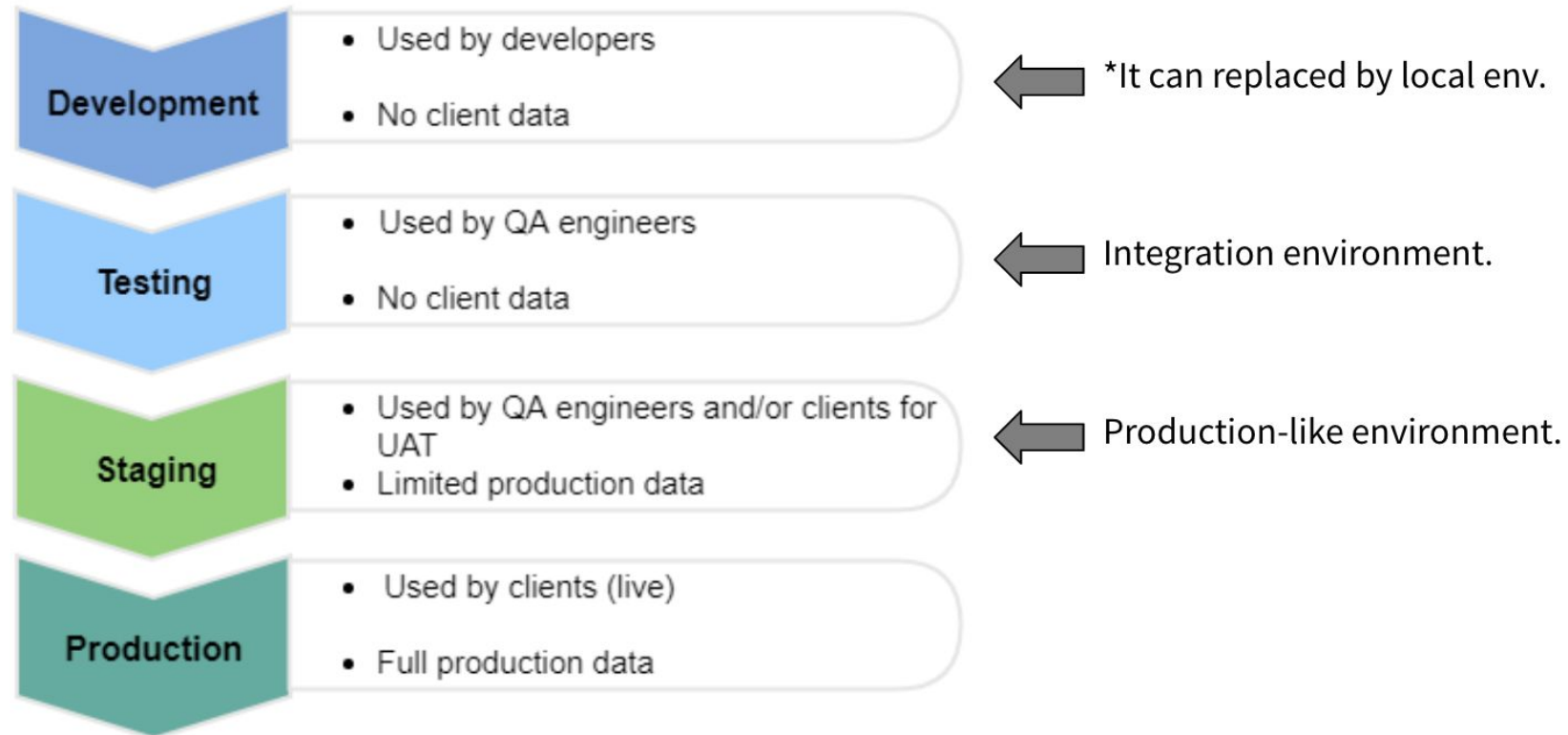
DEVOPS: Continuous Testing

Beneficios:

- Primeramente, obtenemos *feedback* rápido en cualquier fase, los test automáticos incluidos en nuestra *pipeline* tienen que fallar lo más pronto posible para detectar los problemas tempranamente.
- Otra cuestión, siempre estamos aprendiendo de nuestra aplicación y la calidad que esperamos que tenga.
- Como consecuencias, acelera el desarrollo y reduce los tiempos de bloqueo, limita el trabajo en progreso beneficiándonos de entregas más temprano y mejor *time to market*.
- *Continuous Testing* incrementa la confianza de que vamos por buen camino e incrementa la moral del equipo.

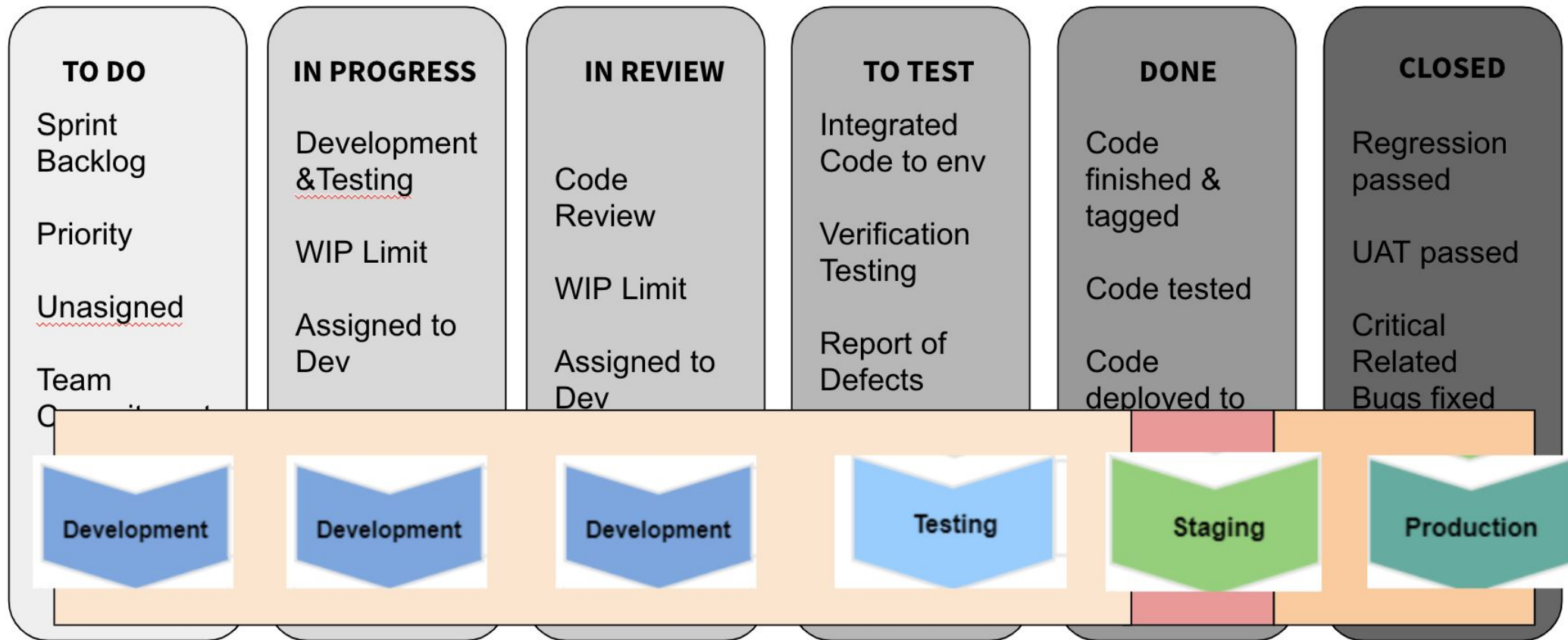
DEVOPS: Environments

Environment Configuration



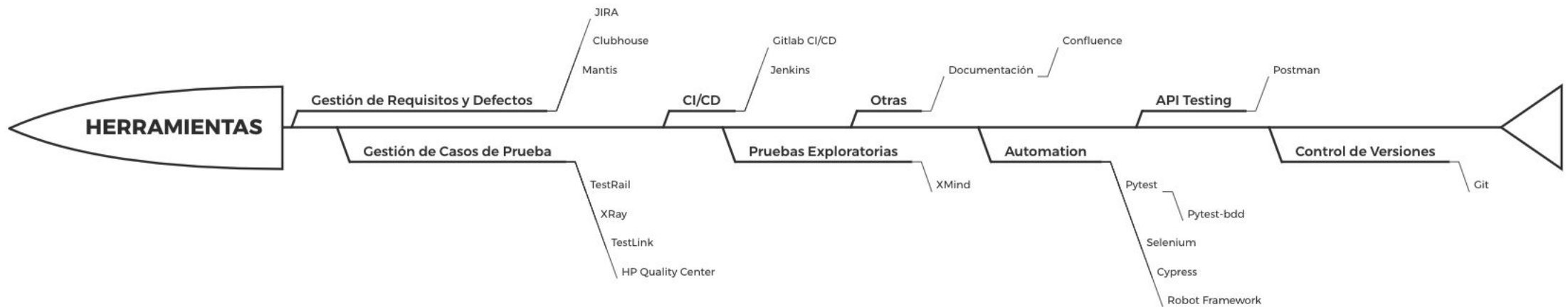
DEVOPS: Environments

Environments & Board & SDLC



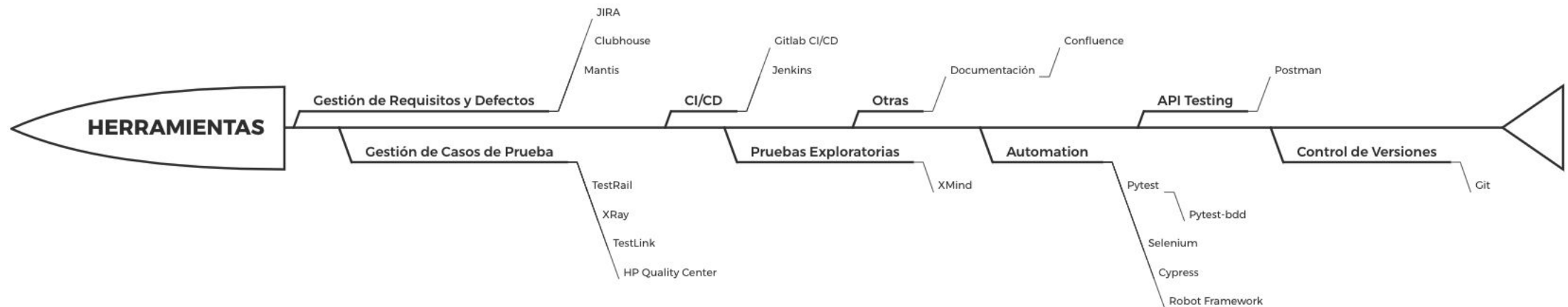
DEVOPS: Herramientas

Actualmente el número de herramientas para pruebas de software disponibles, tanto en el mercado como de manera gratuita y herramientas de código abierto, es muy amplio.



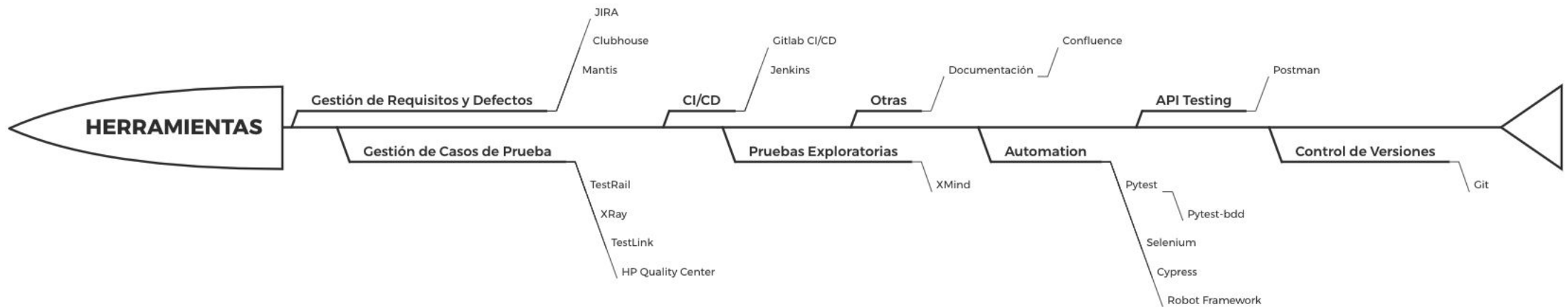
DEVOPS: Herramientas

Actualmente el número de herramientas para pruebas de software disponibles, tanto en el mercado como de manera gratuita y herramientas de código abierto, es muy amplio.



DEVOPS: Herramientas

Actualmente el número de herramientas para pruebas de software disponibles, tanto en el mercado como de manera gratuita y herramientas de código abierto, es muy amplio.



DEVOPS: Herramientas

Al medir nuestras acciones, podemos saber cómo estamos trabajando en nuestras tareas, cuánto tiempo están tomando y cuánto están costando. Esta información puede ser utilizada para mejorar nuestros procesos y calidad.

Además, podemos aplicar estas métricas y técnicas a nuestras actividades para obtener información importante, que podemos utilizar para tomar mejores decisiones; decidir bien nos lleva a tomar acciones.

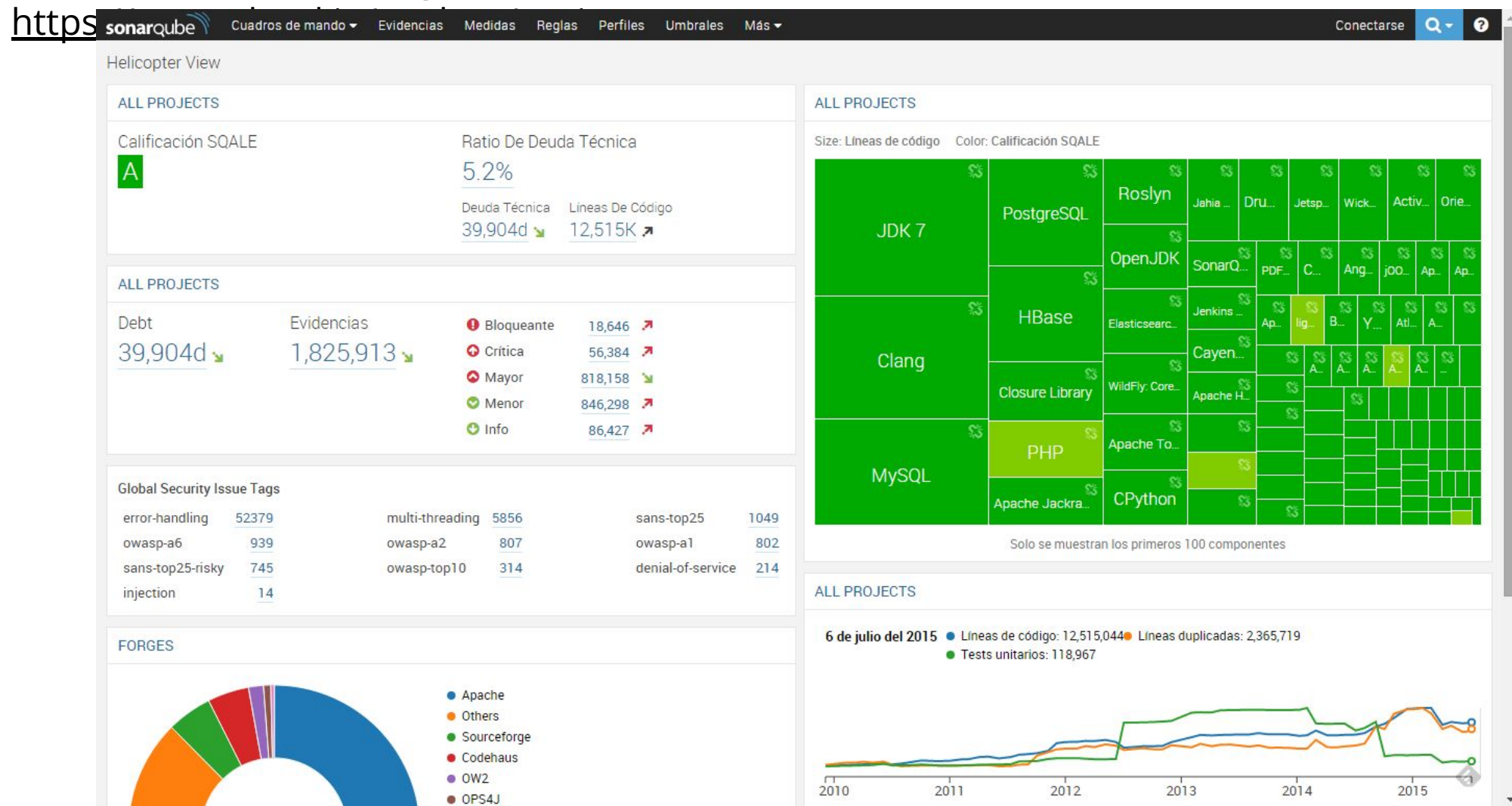
Mejorar el nivel de calidad depende de las métricas y es por eso que debemos elegir las que ayuden a determinar la productividad y la eficiencia.

“Si algo no se puede medir, no se puede controlar”



DEVOPS: Métricas

Calidad estática del código:

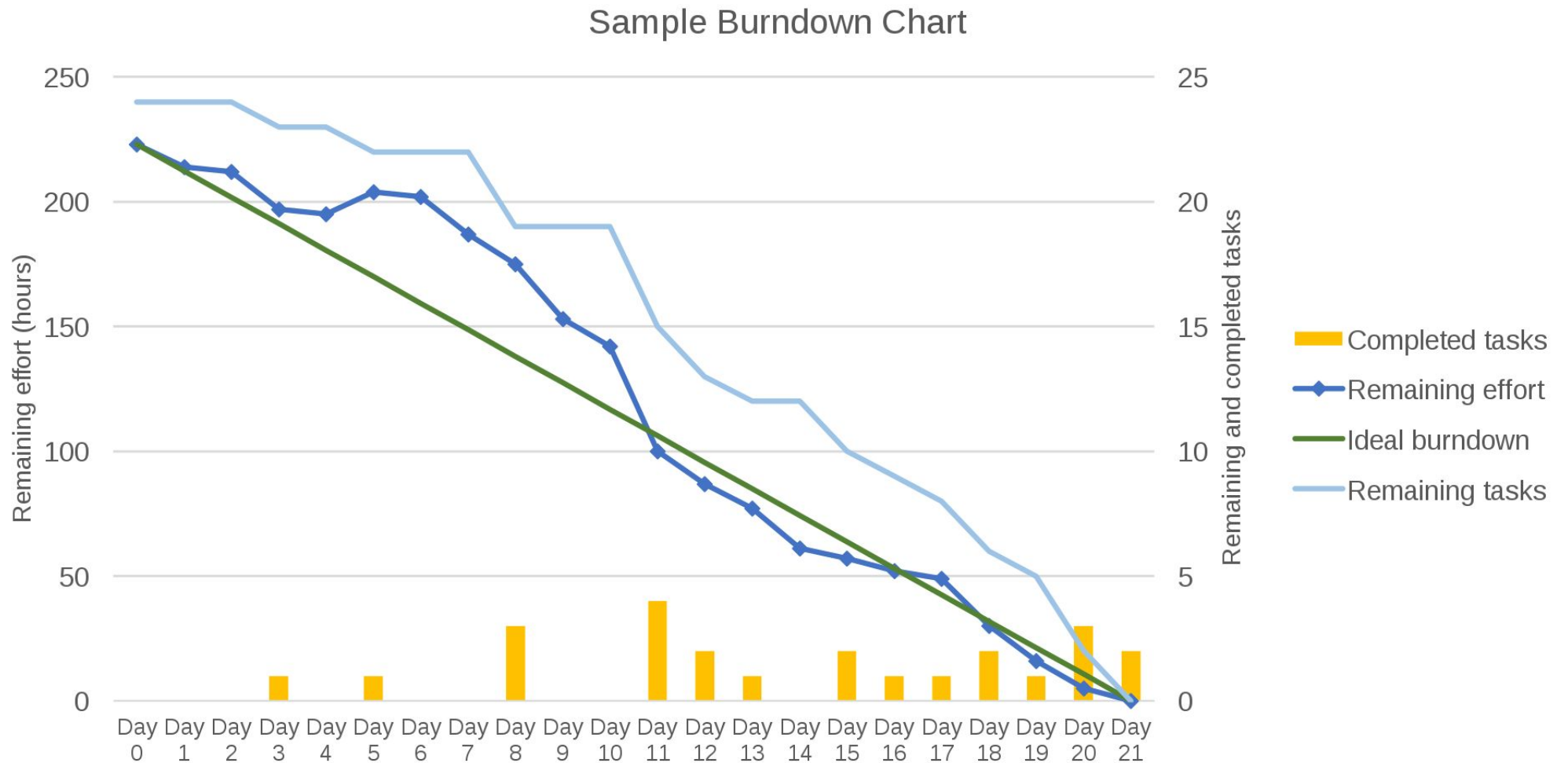


DEVOPS: Métricas

Un diagrama burn down o diagrama de quemado es una representación gráfica del trabajo por hacer en un proyecto en el tiempo. Usualmente el trabajo remanente se muestra en el eje vertical y el tiempo en el eje horizontal. Es decir, el diagrama representa una serie temporal del trabajo pendiente.

- El burndown chart, es muy fácil de crear, mantener y comprender. Esto lo convierte en una herramienta sencilla pero muy importante para la gestión ágil de proyectos, que permite al propietario del producto, al Scrum Master y al equipo de desarrollo hacer un seguimiento para saber si se pueden cumplir los requisitos y plazos del proyecto.
- Por lo tanto, el burndown chart también sirve como un sistema de gestión de riesgos y de alerta temprana. Los problemas, cuellos de botella y desviaciones de los puntos de vista estimados y de la implementación real se identifican rápidamente.
- El burndown chart puede ser una gran herramienta para visualizar el progreso del proyecto y hacerlo tangible. La capacidad de mostrar el progreso en cualquier momento muestra al cliente que el equipo está trabajando de forma profesional y fiable.

DEVOPS: Métricas



DEVOPS: Métricas

QA Sign-off summary of <PROJECT NAME>

Report Date	<dd MMM yyyy>
Author	<Name>

Summary Status:

Status	Commentary
GREEN	The release has been signed-off and ready for next phase.
AMBER	The release is signed-off and ready for the next phase, BUT subject to acceptance of the risk items included below.
RED	The release is NOT signed-off for the reasons noted below.

Release Candidate Stability

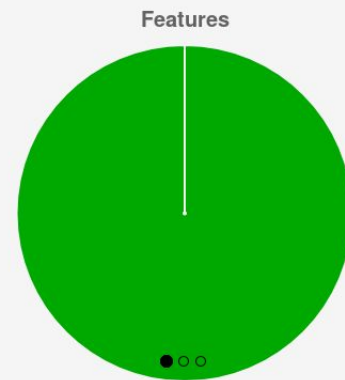
Green	This release is free of Components of the product where the issues (only priority >= major) are still rising (Futured > 0).	1. Issues:
Green	The release is free of patches required to get new builds working (environmental issues related to product components)	1. ENV tickets related to product components:
Green	A complete cycle of regression testing has been run without any build changes/code updates/fixes (regression testing ran on one version only, if not comment reason).	
Green	New Test Plans have been reviewed by Dev Team.	% Reviewed:
Green	The Product Team confirms that all User Stories have been closed and there are no showstoppers for this release.	

DEVOPS: Métricas

Project	Number	Date
website_tests	122	20 Feb 2018, 16:57

Features Statistics

The following graphs show passing and failing statistics for features



	Test Cases (Manual)		Test Cases (Automation)		Total	
	#	%	#	%	#	%
Total	6243	100,00	856	100,00	7099	100,00
Pass	6155	98,59	856	100,00	7011	98,76
Fail	88	1,41	0	0,00	88	1,24
Not Run	0	0,00	0	0,00	0	0,00

Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Login	9	0	0	0	0	9	1	0	1	11.588	Passed
Register	23	0	0	0	0	23	1	0	1	15.681	Passed
	32	0	0	0	0	32	2	0	2	27.269	2
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%

DEVOPS: Métricas

Issues discovered in this release (by Priority):

Status	Blocker	Critical	Major	Minor	Total
Total (Issues found in this release while testing)					
Futured (Issues found in this release but to be fixed in the future)					
Fixed (Issues found in this release which are now fixed)					
Closed (Issues found in this release, fixed and retested)					

Severity (Bugs)	
Status	# of open issues
Blocker	
Critical	
Major	
Minor	
Trivial	
Total	

Priority (Bugs)	
Status	# of open issues
Blocker	
Critical	
Major	
Minor	
Trivial	
Total	

Bugs delivered as fixed in this release (by Priority):

Status	Blocker	Critical	Major	Minor	Total
Delivered					
Closed					
Reopened					
Untested					

¿Preguntas?



fuentes

- <https://jira.atlassian.com/>
- <https://www.gurock.com/testrail/>
- <https://www.getxray.app/>
- <https://www.xmind.net/>
- <https://www.postman.com/>
- <https://www.sonarqube.org/>
- <https://www.selenium.dev/>
- <https://pytest.org/>
- <https://docs.gitlab.com/ee/ci/>
- <https://git-scm.com/>

¡GRACIAS!

