

# Continuous Delivery

Entregas continuas y sistemas eventualmente exitosos

Por: Juan Timoteo Ponce Ortiz

# Anécdota

# Servicio X



DIE X AUGUSTI ILE ET III SEPTEMBRIS MDCCXC

# Empresa X





Inconsistencia entre que el usuario quiere y lo que el programa hace

Lentas actualizaciones del servicio

Necesidad constante de modificar el servicio

Costos elevados de soporte/mantenimiento

Costos elevados de desarrollo/Ingresos reducidos

El servicio X es inestable e inflexible, lo cual genera dificultades para que los clientes sigan usando el servicio y las capacidades del mismo se extiendan para nuevos clientes.

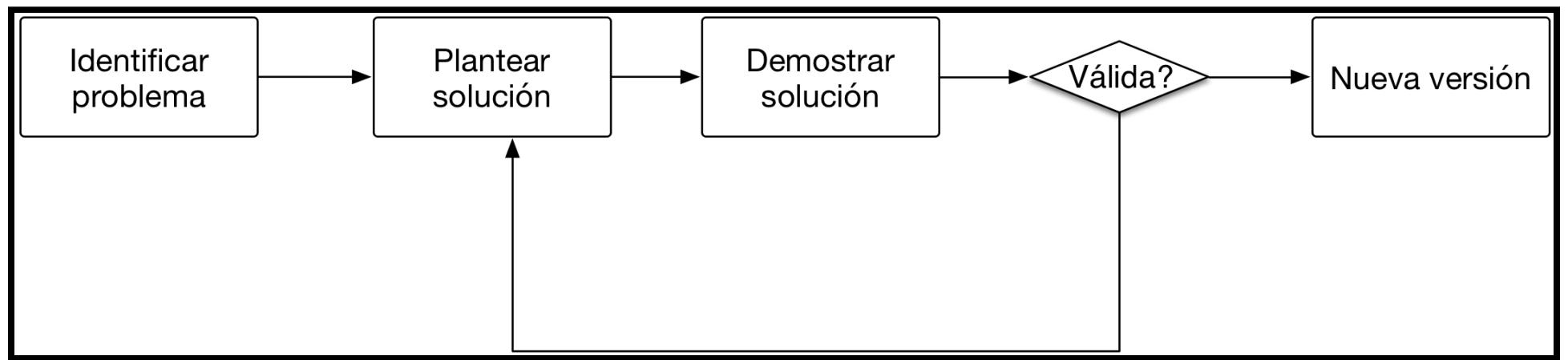
# Solución propuesta





---

Pieter Hintjens



Implementar el método científico como proceso de desarrollo para estabilizar y flexibilizar el servicio X a través de mejoras incrementales en el software.

# Ingredientes

Proceso de desarrollo iterativo (scrum, crystal, xp)

Sistema de control de versiones (git, hg, fossil)

Pruebas automatizadas

Instalaciones automatizadas

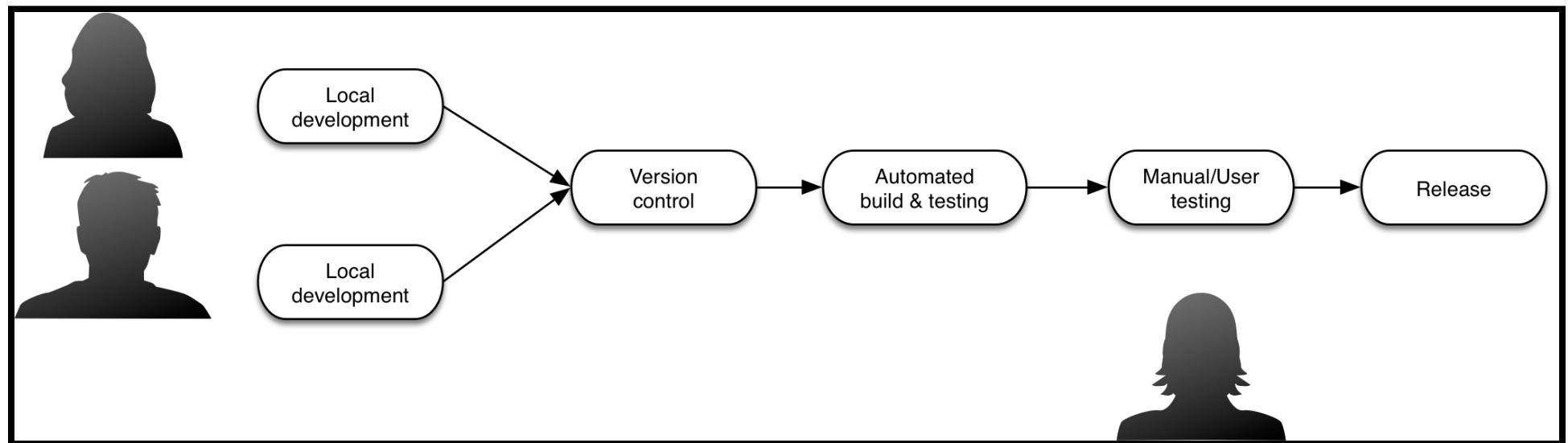
TLDR: Aplicar Continuous Delivery

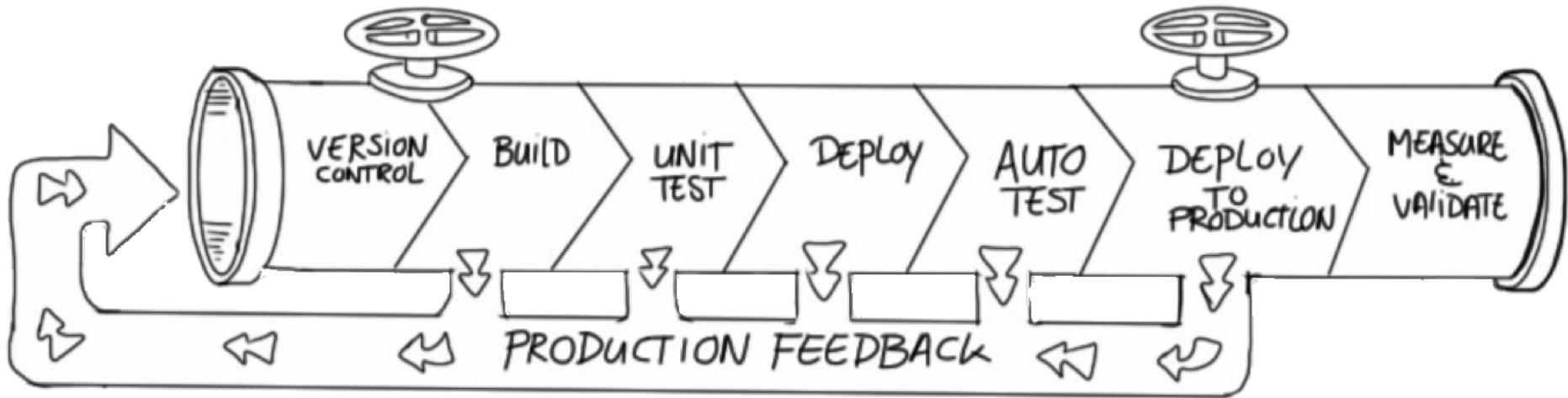
# Continuous Deployment

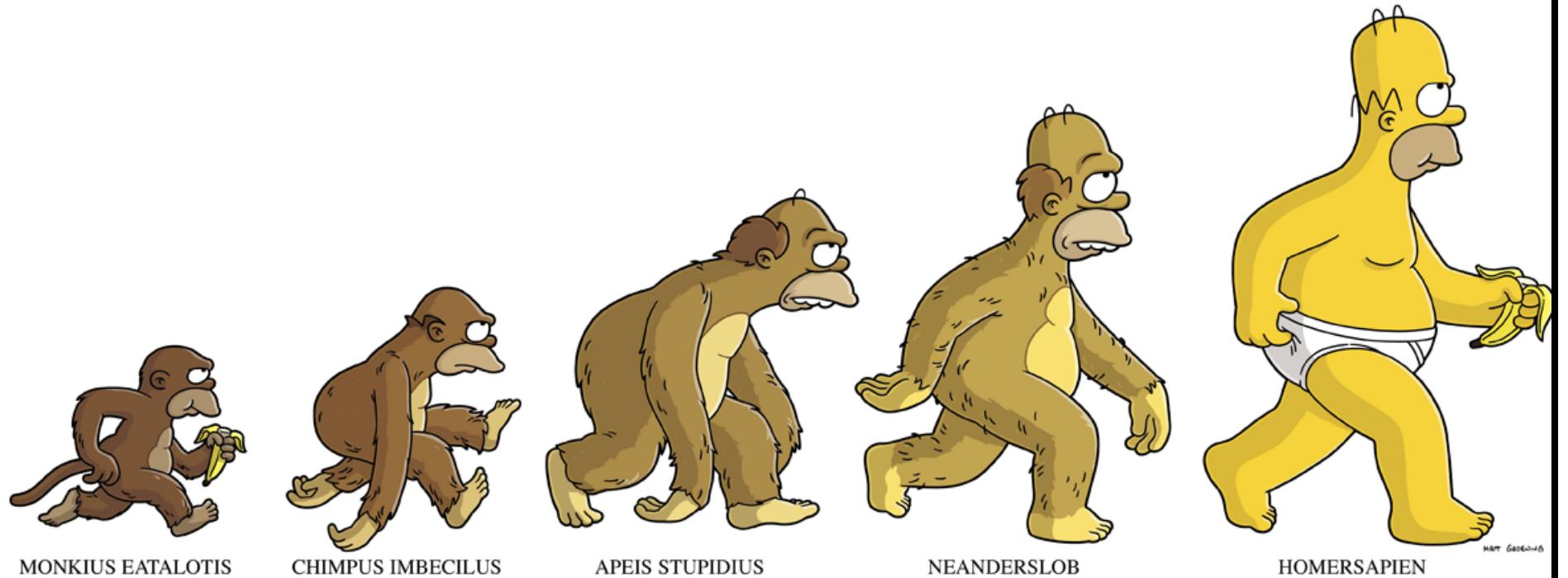
Cuánto tiempo les toma hacer un cambio y subirlo a producción?

Práctica que permite producir software en cortos períodos de tiempo y de manera frecuente, en un estado listo para producción

# Proceso







## HOMERSAPIEN

# Beneficios

- Feedback rápido
- Escenario de producción
- Reactividad
- Estabilidad eventual

# Riesgos

- Consistencia dependiente de pruebas
- Dependencias externas/software
- Lentitud inicial

# Herramientas

	docker-clean	21 hr - #311	1 mo 4 days - #276	2.1 sec			
	install build-server-8.0 to sf80.screenfoodnet.com	1 yr 2 mo - #5	N/A	1 min 36 sec			
	install build-server-8.1 to deploy.screenfood	11 mo - #62	N/A	1 min 39 sec			
	install build-server-8.1 to sf81.screenfoodnet.com	10 mo - #3	N/A	1 min 40 sec			
	install build-server-9.0 to deploy.screenfood	4 mo 29 days - #70	N/A	1 min 31 sec			
	install build-server-9.0 to sf90.screenfoodnet.com	4 mo 12 days - #21	N/A	1 min 35 sec			
	install build-server-9.1 to deploy.screenfood.com	8 days 2 hr - #54	N/A	1 min 39 sec			
	install build-server-9.1 to testing.screenfoodnet.com	17 hr - #307	N/A	1 min 45 sec			
	install build-timo to deploy.screenfood	13 days - #283	N/A	1 min 39 sec			
	install cms-new-ui-react to deploy	9 mo 18 days - #52	N/A	3 min 36 sec			
	install sfn-cms-ui to deploy	27 days - #33	N/A	55 sec			
	install_build_sfn-cms-ui_timo_to_deploy2	5 mo 11 days - #37	N/A	44 sec			
	install_sfn-cms-ui_to_testing	5 hr 0 min - #458	N/A	56 sec			
	publish_sfn-cms-core_9.0	4 mo 12 days - #12	4 mo 25 days - #2	6 min 30 sec			
	publish_sfn-cms-core_9.1	1 day 1 hr - #201	29 days - #168	3 min 5 sec			
	sfn-cms-adapters	2 days 14 hr - #71	2 mo 7 days - #63	5 min 44 sec			

# Herramientas

- Tests: xUnit, xSpec
- Automation: Bash, Rake, Ant, Buildr
- Containers: Docker, Vagrant
- Deployment: Capistrano, Fabric, Ansible
- CI/CD hosts: Jenkins, GoCD, Travis, Bitbucket pipelines

# Topologías/Patrones

# Clay

Negocios en crecimiento, startups, tiempo al mercado es crítico

---

Plataformas dinámicas: Ruby, NodeJs, Python

---

Ambientes dinámicos: AWS, Docker, Azure

---

Despliegues frecuentes, diarios, semanales

# Wood

Negocios semi-establecidos, ME, estabilidad e integración con otros servicios

---

Plataformas dinámicas y estáticas: C#/Java, Python

---

Ambientes dinámicos y estáticos: AWS, Docker, Azure, Servidores locales

---

Despliegues aprobados por alguien, semanales, mensuales

# Iron

Negocios establecidos, estabilidad , integración y consistencia

---

Plataformas mayormente estáticas: C#/Java, Cobol, Scala

---

Ambientes mayormente estáticos: Servidores locales, granja de servidores

---

Despliegues poco frecuentes pero estables, mensuales, trimestrales, anuales

# Resultados

Costos de desarrollo reducidos

Reducción de tiempos de soporte

Rápida respuesta al cambio

Cambios/mejoras obligados a la arquitectura

# Recomendaciones

El desarrollo de software es ampliamente experimental en muchos aspectos

Ambientes experimentales requieren que los cambios sean fáciles de realizar y rápidos para descartar

Terminado quiere decir, desarrollado, verificado y desplegado en producción

# Gracias!!

Alguna pregunta?