

CURSO DE POSGRADO

Técnicas y Gestión de las Pruebas de Software

Gustavo Guimerans

DOCENTE

Herramientas de automatización

¿Herramientas?

Contexto: Herramientas de pruebas

Apoyo al equipo de pruebas en distintas etapas y actividades

Clasificadas según su función

- Ej. Gestión, automatización, seguimiento, generación de casos...

Considerar casos intrusivos

- Instrumentación del código
- Alteración del comportamiento
 - funcional
 - **performance**
- Pruebas finales sin ellas

Algunas están en la frontera desarrollo/pruebas

- Análisis dinámico de rendimiento, perfilado (*profiling*) de ejecución, modelado y análisis de ejecución de base de datos, cobertura

Contexto: Ej. de clasificación

- **Gestión**
 - ... de pruebas, ... de requisitos, ... de incidentes, ... de la configuración, ...
- **Pruebas estáticas**
 - Revisión, Análisis estático, Modelado de software, Medidas complejidad
- **Especificación o generación de pruebas**
 - Asisten en el diseño y especificación de casos de prueba; generación de datos.
- **Evaluación de pruebas**
 - Asisten en la evaluación del resultado de las pruebas
 - ayudando a determinar si el resultado observado es el esperado.
- **Framework de ejecución y registro de pruebas**
 - de automatización (robots de ejecución), Arnés de ejecución, Comparadores, Medida de cobertura
 - Permiten...
 - ejecutar pruebas en un ambiente controlado
 - observar el comportamiento del elemento bajo prueba
- **Análisis de rendimiento (performance) y administración/supervisión**
 - Medidas de rendimiento, supervisión, análisis dinámico y *profiling* de ejecución y uso de recursos
- **Especializadas**
 - diferentes plataformas, estándares industriales o productos, seguridad
- **Otras**
 - Cálculos de previsión y estimación, debugging

¿Automatización?

Contexto: Contexto

- Reducir costos y tiempos en las pruebas de regresión.
- Conjunto de pruebas de humo automatizadas.
- Probar sobre diferentes plataformas.
- Integración continua.

Contexto: Automatización

- Software que permite automatizar pruebas
 - controlar la ejecución de las pruebas
 - comparar el resultado obtenido contra el esperado
- Contribuye a mejorar la calidad
 - Testing más frecuente y antes
 - Mayor cobertura
- Reduce el tiempo destinado a testing
 - Corolario: reduce el tiempo de salida al mercado (*time to market*)

Contexto: Automatización

- Propietarias, integradas, freeware, código abierto
- Responsabilidades
 - Diseño
 - Especificación y preparación de los casos y datos de prueba
 - Automatización
 - “rentabilidad” casos de varias ejecuciones de “la misma prueba”
 - Costo tiempo de preparación vs. Ejecución de cada caso de prueba
 - *Releases* frecuentes inversión para ejecución de “pruebas de regresión”
 - Ejecución y análisis de resultados
- Múltiples
 - dispositivos (ej. Android, iOS)
 - navegadores
 - configuraciones

¿Pros / Contras?

Automatización: “Pros y Contras”

Pros

- repetitividad
- control más preciso
- mayor eficiencia
- integración con sistemas
- incrementa productividad
 - manual en otras áreas
- “reducción de trabajo”

Contras

- inversión en capital
- decremento en flexibilidad
- incremento dependencia
 - mantenimiento
 - reparación

Automatización: Vocabulario común

- Caso de prueba
 - Diseñado con un objetivo particular, consta de
 - Entradas de prueba
 - Condiciones de ejecución
 - Resultados esperados
 - Puntos de verificación
- Script de prueba
 - Instrucciones detalladas para
 - Configuración
 - Pasos a ejecutar
 - Verificación
 - Para un caso o un conjunto de casos de prueba

(Automatización: Ejemplo “EditUser”)

- **Caso de prueba**

- **Diseñado con un objetivo particular, consta de**
 - Entradas de prueba?
 - Condiciones de ejecución?
 - Resultados esperados?
 - Puntos de verificación?

- **Script de prueba**

- Instrucciones detalladas para
 - Configuración
 - Pasos a ejecutar
 - Verificación
- Para un caso o un conjunto de casos de prueba

Automatización: Vocabulario común

- Suite de prueba
 - Conjunto de scripts de prueba
- Grabación
 - Registrar las acciones que realiza el usuario utilizando herramientas
 - Se traduce en un script
- Parametrización
 - Definición de parámetros para los scripts
 - Parámetros contienen los datos de prueba

¿Manual o Automatizado?

- El testing manual es un proceso que
 - se adapta al cambio
 - puede hacer frente a la complejidad
 - puede detectar varios problemas a primera vista
 - permite evaluar la importancia de los problemas
 - introduce variantes útiles para descubrir errores
- La automatización, en cambio
 - se limita a la verificación definida explícitamente en el script

¿Manual o Automatizado?

- Las pruebas automatizadas
 - son repetibles y más rigurosas
 - Pasos claros para reproducir el incidente
 - tienen menor probabilidad de encontrar un incidente
 - permiten hacer testing más temprano
 - permiten hacer testing más frecuente
 - ayudan a lograr una mayor cobertura

Complementa el testing manual, no lo sustituye

Automatización: Valor de las pruebas

- Pruebas de regresión
 - Baja probabilidad de encontrar errores
 - Entonces, ¿para qué ejecutar las pruebas?
- Valor
 - Validar que no se han introducido defectos en la nueva versión
- Tranquilidad
 - Para salir en producción o liberar el producto
 - Para hacer ajustes y mejoras en la aplicación

¿Proyecto de automatización?

Proyecto de automatización

- Es un proyecto en sí mismo
 - hay que asignar
 - tiempo y recursos
 - personal especializado
- Requiere etapas
 - planificación
 - diseño
 - desarrollo
 - ejecución
- Requiere gestión de configuración
 - Control de versiones
 - scripts, datos de prueba, documentación, aplicación
 - Trazabilidad de las funcionalidades bajo prueba

Costo/beneficio de la automatización

- Aprender de la herramienta
- Planificar
- Diseñar
- Desarrollar
- Verificar y ajustar los scripts
- Mantener los scripts ante cambios en el producto
- Documentar
- Gestionar

- Es una inversión.
 - Se recupera luego de la ejecución de determinados ciclos de prueba.
 - Las pruebas automatizadas desplazan ciertas pruebas manuales.
 - aumento de la productividad en el testing
 - Hay riesgos... técnicos

Riesgos técnicos de la automatización

- Limitaciones en las herramientas de automatización
 - Funcionalidades
 - Extensiones
 - Plataformas objetivo
 - Productividad
- Dificultades con particularidades de la aplicación
- Mitigar riesgos
 - Haciendo una prueba piloto/concepto
 - Desarrollando la plataforma de automatización

Construcción de la automatización

- Pensar en la prueba primero
- Luego, en la automatización
- Herramientas de automatización
 - Generalmente “REC and PLAY”
 - atractivas
 - “fácil uso”
 - No automatizan, ayudan a automatizar
 - Las pruebas deben ser definidas y evaluadas usando
 - el conocimiento del dominio
 - el conocimiento de la aplicación
- Pueden asociarse varias herramientas
- Implementar para soportar cambios en la aplicación



Construcción de la automatización

- **Comparar comportamiento esperado con obtenido**
 - **Scripts sensibles**
 - Comparan más propiedades
 - Ejemplo: Mensaje de alerta con fuente, color, posición
 - **Script robustos**
 - Comparación más selectiva
 - Ejemplo: Mensaje de alerta sólo texto
 - **Diferentes fuentes de datos**
 - La propia aplicación
 - Otras
 - Bases de datos
 - Archivos datos (HTML, txt, xml)
 - Capturas de pantalla

Reporte de la automatización

- Información sobre la ejecución de las pruebas.
- ¿Qué información sería deseable presentar?
 - estado de la aplicación
 - estado del entorno
 - estado de la herramienta de automatizaciónEj. estado de archivos, estado de bd, *screenshots*,
...
- Nivel de detalle adecuado para la evaluación.

Gestión de la automatización

- Control de versiones
 - Git
- Manejo de distintos scripts
 - Pruebas
 - Generación de datos
 - Configuración del entorno

Habilidades para la automatización

- No coinciden exactamente con las del tester manual
 - Requiere capacidades de programación
- Perfil “técnico en TI”
 - Esto generalmente excluye a los usuarios finales
- Habilidades en automatización
 - Esfuerzo inicial para aprendizaje
 - Uso de herramienta de automatización
 - Uso de herramienta de grabación (si se tiene)
 - Manejo del lenguaje
 - Estándares de programación
 - Gestión de las pruebas
 - Especialización

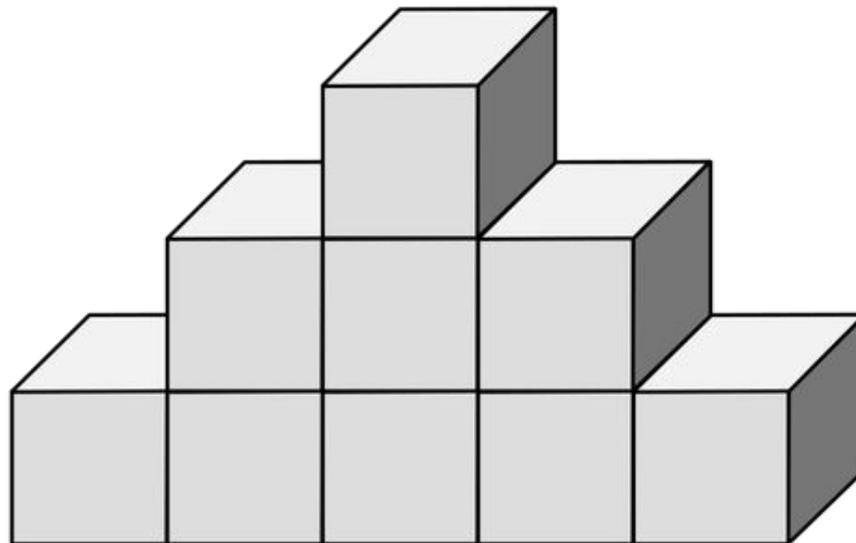
Consideraciones en la automatización

- Procedimientos ya consolidados y precisos.
- Si no hay procedimientos, hay primero que consolidarlos.
- Una herramienta inadecuada o inoportuna puede ser un freno al proceso de mejoras de las pruebas.
- Seleccionar e integrar estratégicamente las herramientas para la automatización
- Habilidades y curva de aprendizaje
- Su grado de inteligencia y flexibilidad es limitado
- No tienen la flexibilidad de una persona para...
 - entender lo que hay que hacer;
 - improvisar para resolver las dudas e
 - incompletitudes de las especificaciones de pruebas
- Seleccionar cuidadosamente las pruebas a automatizar

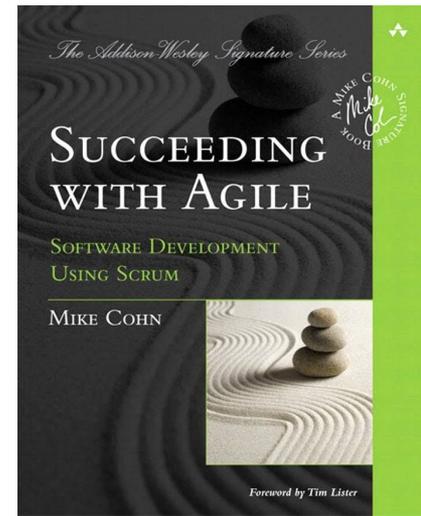
Resumen automatización

- La automatización es una inversión
- Se recupera luego de la ejecución de ciclos de prueba
- La automatización desplaza ciertas pruebas manuales
 - aumento de la productividad en el testing
- ¡Es un proyecto en sí mismo!

¿Pirámide?



Pirámide



Junio 2023

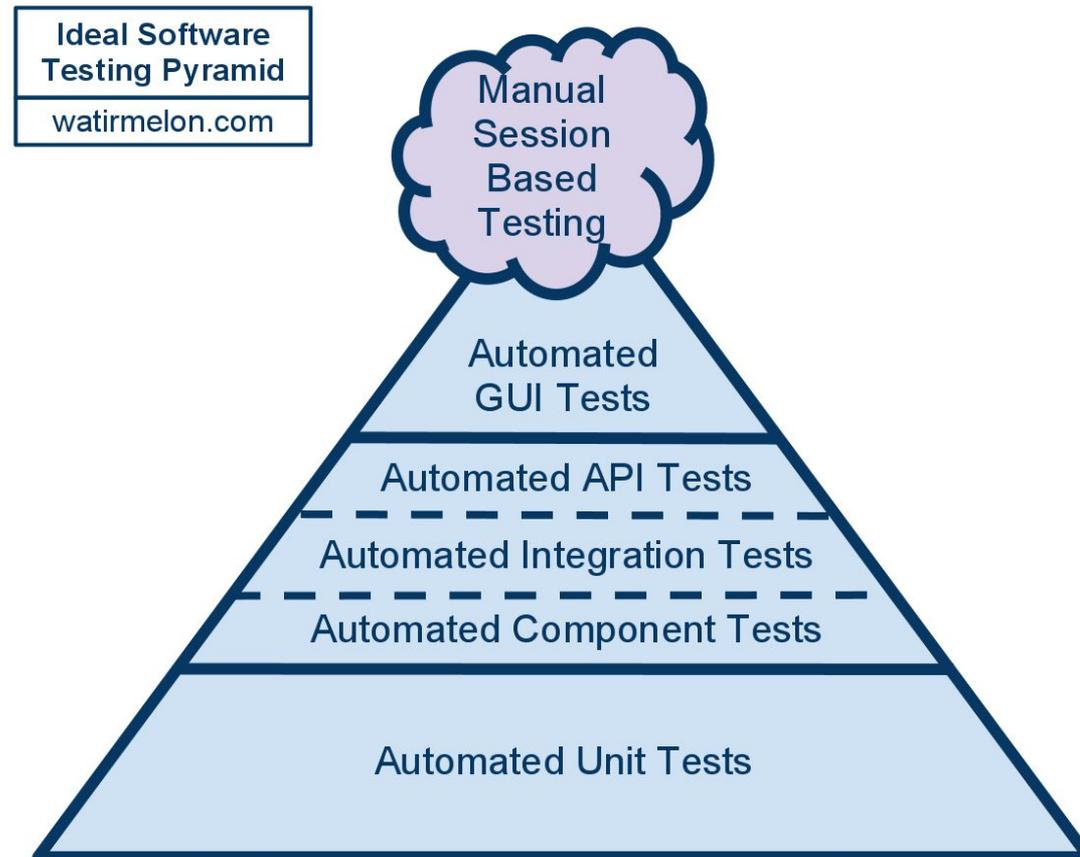
Pirámide

- Proporción de pruebas
 - Implementación
 - Mantenibilidad
 - Velocidad de ejecución
 - Facilidad para encontrar errores
- Como guía:
 - 50% pruebas unitarias
 - 40% pruebas de servicios
 - 10% pruebas de GUI



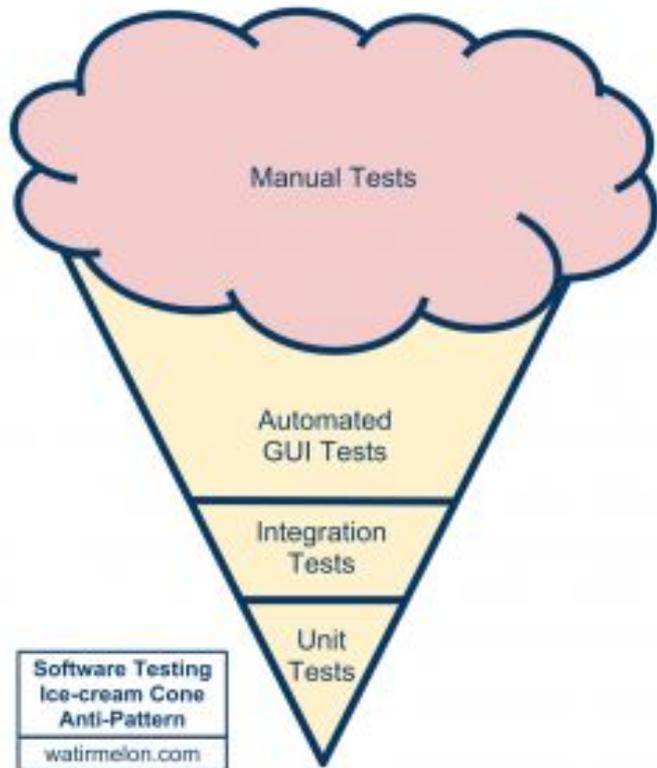
Pirámide extendida

Pruebas automatizadas, **no sustituyen el testing manual**

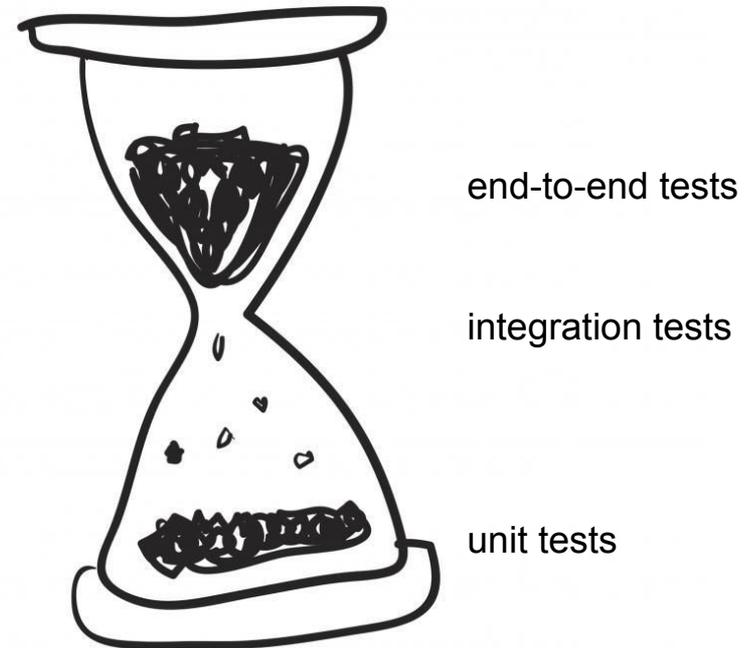


Antipatronos

Cono (*Ice-cream cone*)



Reloj de arena (*Hourglass*)



Capas / Niveles

- Pruebas Unitarias
 - **Ejecución temprana**
 - Sin necesidad de otros componentes
 - Escritas por el desarrollador
 - Mismo lenguaje que la aplicación

- Pruebas de Servicios
 - Pruebas de lógica de negocios
 - Integración
 - Otros sistemas
 - Componentes
 - REST/SOAP

Capas / Niveles

- Pruebas de GUI
 - **Verifican exactamente el comportamiento del usuario**
 - Costosas de mantener
 - Sistema completamente integrado
 - Desarrollo testeable

Integración continua (CI) Entrega continua (CD)

Integración continua

- Reduce costos de integración
 - El impacto sobre otros desarrollados es menor
 - Siempre se está a un paso de una versión integrada
- Detección temprana de defectos
 - Ejecución de pruebas en cada integración
 - Se detectan más temprano

“... no elimina los bugs, pero hace mucho más fácil encontrarlos y corregirlos.”

Martin Fowler

Integración continua

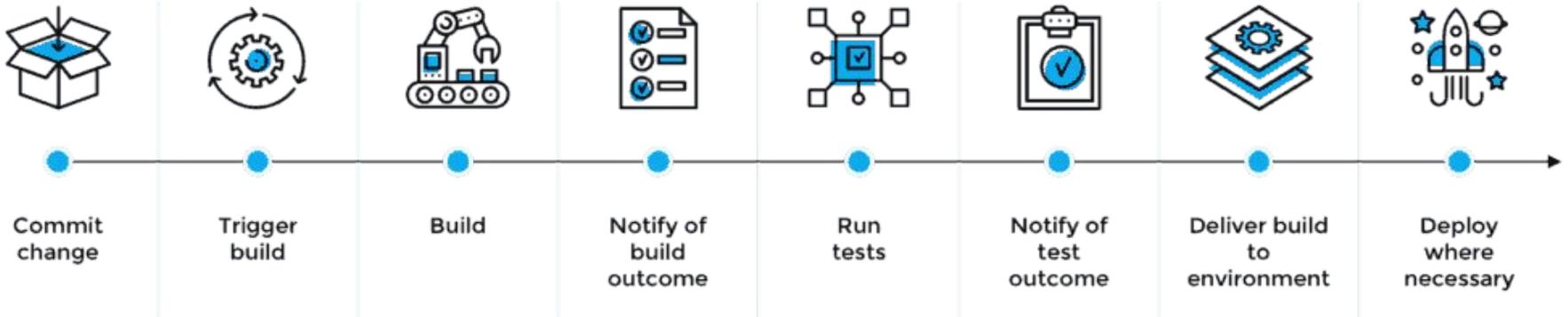
- Integraciones con compilación y ejecución de pruebas
 - *Build* automatizado con
 - compilación
 - ejecución de pruebas
 - unitarias, integración, sistema, performance, instalación, etc.
- Repositorio único compartido de código
 - Con lo necesario para construir la aplicación

Entrega continua

- Un paso más allá que CI
- Cada build deja listo para poner en producción.
- **No existe CI/CD sin testing automatizado**

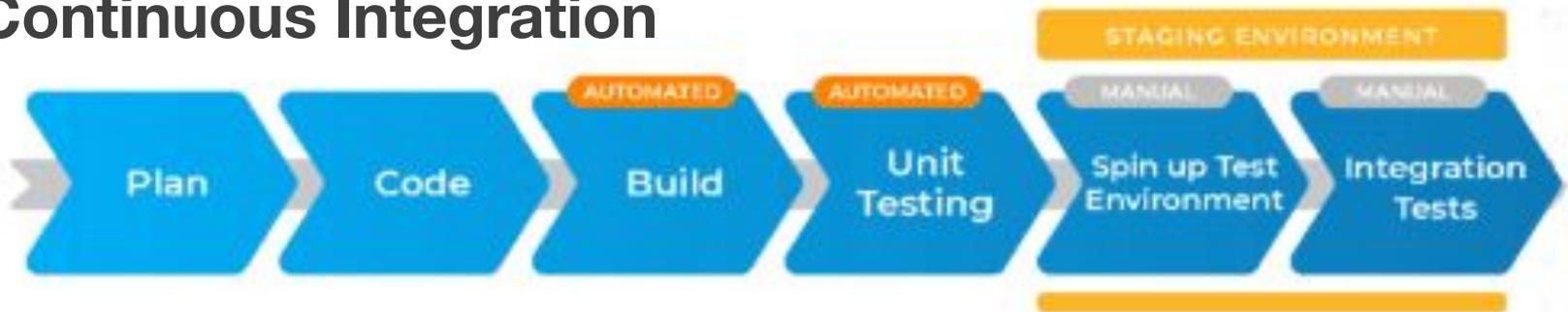
¿Cómo? *Pipeline* automatizado

CI/CD Pipeline

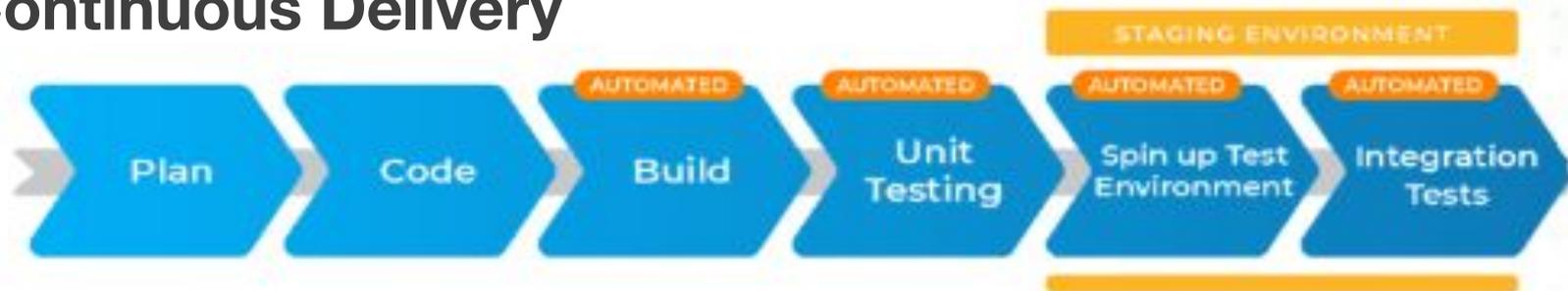


¿Cómo?

Continuous Integration



Continuous Delivery



¿Cómo?

- *Check out*
 - El desarrollador obtiene la última versión del código que compila
- Se codifica e implementan pruebas
- Se hace el *build* local
 - se ejecutan las pruebas locales
- Si pasan todas las pruebas localmente se hace *commit*
- Servidor de CI ejecuta *build* en repositorio central
 - se ejecutan las pruebas en repositorio central

Herramientas

- Build
 - Manejo de Versiones
 - Ej. GitHub, GitLab, Suversion ...
 - Integración Continua
 - Ej. Jenkins, GoCD, Bamboo ...
 - Construcción de software
 - Ej. Maven, Ant, Docker ...
- Testing
 - JUnit, SoapUI, Selenium, JMeter, Cucumber

