

Héctor Agudo

Miembro del Centro Experto
en Integración Continua de
atSistemas

Integración Continua para plataformas COBOL

El objetivo final de la implantación de un sistema de Integración Continua (IC) es evitar algunos riesgos críticos que suelen surgir en todo proyecto, sobre todo en sus etapas finales: fallos en la integración del código de distintos miembros del proyecto, código no testado, código innecesario que no se utiliza, fallos en el despliegue e implantación, fallos de regresión, desinformación del proceso de construcción, descoordinación de versiones de diferentes tecnologías o plataformas que deberían trabajar de forma integrada...

Ese concepto, que está ya completamente aceptado y asentado entre, por ejemplo, los equipos de desarrollo Java, parece más alejado de la forma de trabajar de quienes manejan otras tecnologías, como COBOL. Sin embargo, aunque no sea lo más habitual, también es posible implantar entornos de IC para el desarrollo COBOL, o integrar los desarrollos COBOL en una plataforma IC ya existente en la organización.

Así, las plataformas pensadas para lenguajes COBOL también pueden contar con un proceso de IC cuyo motor ejecute pasos como los siguientes:

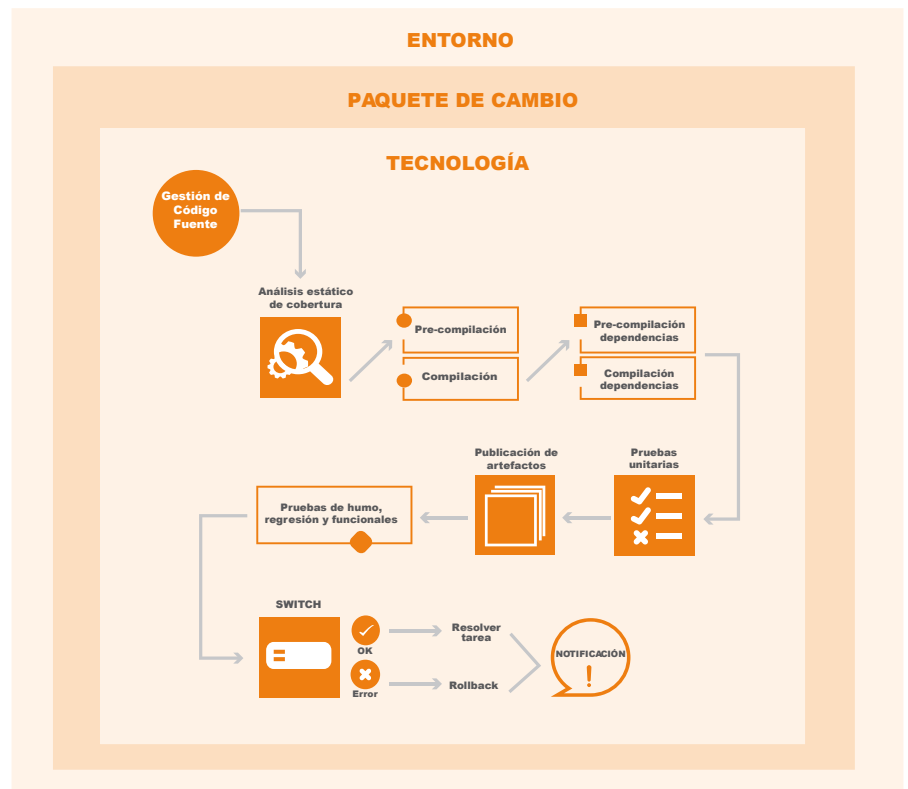
- Análisis de calidad estática de fuentes
- Realización de pruebas unitarias de los desarrollos
- Entrega de forma continua de los desarrollos para detectar problemas de integración de forma temprana
- Hacer implantaciones multiplataforma de forma conjunta
- Ejecución de pruebas funcionales y de regresión a los programas desarrollados
- Informar del resultado de la aplicación de los anteriores pasos

Todos estos pasos se aplicarían de forma automática en entornos de desarrollo, no en entornos productivos. El motivo de no aplicarlo sobre entornos productivos se debe a que muchas veces la puesta en producción de las aplicaciones debe realizarse dentro de una ventana de tiempo acotada. También se dan periodos de congelación de cambios según el negocio de la propia organización para no sufrir determinados riesgos en el negocio. Por esto no se recomienda aplicar IC sobre la LPAR de Producción sino sobre la LPAR dedicada a desarrollo.

Integración Continua para plataformas COBOL

Otra ventaja de aplicar IC es la de poder hacer despliegues de paquetes de cambio de un producto completo, entendiendo como producto completo los desarrollos hechos en diferentes tecnologías y plataformas que trabajan de forma conjunta. Para ser más conciso, un producto podría englobar, por ejemplo, la parte de frontal desarrollada en Java más la parte COBOL/CICS de transacciones CICS (con sus correspondientes mapas de pantalla) lanzadas por colas MQ o IMS o sentencias DB2 de Mainframe o JCL para JOBS e incluso desarrollos críticos para Unix en C. De esta forma podemos desplegar al unísono un producto que implica a diferentes tecnologías.

Un ejemplo sencillo del conjunto de pasos a dar por un proceso de IC podría ser el siguiente:



Estos pasos son muy genéricos y varían en función de la holística tecnológica de cada organización. Además, sería necesario entrar en detalle de casos concretos como el método de obtención de fuentes por parte del motor de IC, la reordenación de paquetes de cambio para optimizar el trabajo del motor de IC, la modificación de elementos de la librería de COPYS, el árbol de organización de descarga de fuentes, la conversión de esqueletos de JCLs para ser lanzados por el motor de IC, la captura de resultados del JES Monitor, la precompilación de los paquetes de cambio en entornos distribuidos para el ahorro de MIPS, el lanzamiento de pruebas automatizadas, la integración del motor de IC con el gestor de tareas y el control de flujo de estados de estas, ...

Integración Continua para plataformas COBOL

Opciones disponibles para la Integración Continua en COBOL

Durante la implantación, se debe tratar de usar los mecanismos que la organización ya tenga elaborados como , por ejemplo, la reutilización de esqueletos para la ejecución de JCL, y las herramientas que mejor se adapten a la organización evitando, así, reinventar nada que ya esté operativo y funcionando. Algunas de estas herramientas para mainframe que se adaptan bien a la implantación de IC son el IDE “IBM Rational Developer for System Z” (con su módulo RDz Unit Test, un emulador de Mainframe para hacer pruebas unitarias en una réplica de un Z/OS independiente para desarrolladores) o el producto “IBM Rational Team Concert for Z”(RTC for Z), un gestor de versiones, motor de apoyo a la construcción y gestión tareas y planificación.

Hay más fabricantes, como Microfocus o Team Vision, que también cuentan con herramientas y soluciones que pueden emplearse para aplicar este proceso. También hay productos, como Open COBOL, que podrían emplearse para precompilar programas COBOL en un entorno distribuido con el consecuente ahorro de MIPS que ello conlleva en plataformas Mainframe, así como la eliminación de los tiempos de espera que sufren los desarrolladores puesto que los trabajos enviados por los desarrolladores suelen tener la prioridad más baja (refiriéndose a ser los últimos en entrar en ejecución).

En sistemas muy ajustados, y debido a la bajada automática de prioridad de sus tareas, cuando la LPAR de producción requiere mayor consumo de MIPS, los trabajos del equipo de desarrollo son encolados generando cierta frustración en los miembros del equipo. Esta frustración aumenta cuando el elemento enviado contiene errores que podrían haberse detectado rápidamente, y sin apenas esperas, gracias a un pre-compilado en un sistema externo controlado.

También existen productos de COBOL para plataformas System Power (System i, AS/400, System p, AIX) como el IDE “IBM Rational Developer for i, RPG and COBOL Tools” o el IDE “IBM Rational Developer for AIX and Linux AIX COBOL Edition”, con lo que también se cubre el ámbito para estos sistemas medios que corren programas en COBOL. Estas plataformas también tienen disponible el “IBM RTC” comentado en líneas anteriores.

Hay otras herramientas que sirven de apoyo a la ejecución de pruebas funcionales o de regresión, a la preparación de la carga de datos para la ejecución de pruebas, análisis de código estático,...

El motor de IC más extendido es la herramienta “Jenkins”, dadas sus grandes posibilidades de flexibilidad de adaptación y customización; pero se pueden emplear otros motores IC y herramientas sugeridas por cada organización según se estime oportuno. Fabricantes como IBM o Atlassian cuentan con sus motores de IC de calidad y prestaciones enteramente comparables al conocido Jenkins.



En atSistemas somos más de 550 profesionales dedicados desde 1994 a la consultoría, servicios de IT y desarrollo de software. Nuestros servicios se caracterizan por la flexibilidad y la agilidad, lo que nos permite ayudar a grandes empresas de todos los sectores, aportando conocimiento y experiencia sobre el más amplio abanico de tecnologías.

Nuestra cartera de clientes incluye más de 200 de las principales empresas del país, con representación de todos los sectores de actividad, a los que prestamos servicio desde nuestras oficinas de Madrid, Barcelona, Cádiz, Zaragoza y A Coruña.

Nuestro portfolio de servicios abarca desde el desarrollo de software a medida hasta la integración de grandes soluciones de software empresarial, en áreas que van desde la más compleja arquitectura de sistemas hasta las soluciones más novedosas de comercio electrónico o aplicaciones móviles.

Valle de Alcudia, 3
28232 Las Rozas, Madrid

Passeig de Gràcia 55, 8º - 4ª
08007 Barcelona

 902 888 902



atsistemas.com



info@atsistemas.com