



DIEZ PASOS HACIA LA
**AUTÉNTICA
AGILIDAD
MAINFRAME**

Un enfoque por fases para desarrollar
y operar entre varias plataformas con éxito



Lo que significa este enfoque para usted y su empresa

Las empresas deben transformar de manera rápida y decisiva sus prácticas de mainframe. Los procesos y métodos lentos e inflexibles del pasado se han convertido en impedimentos que no se pueden tolerar para tener éxito en los mercados digitales centrados en la innovación de hoy en día. Los líderes informáticos deben, por tanto, traer las ventajas constatadas de Agile, DevOps y las disciplinas relacionadas para hacer presión en las aplicaciones y datos de mainframe que hacen funcionar su negocio.

Pero, ¿cómo? ¿Y dónde empezar? La transformación del mainframe puede parecer abrumadora. Además, ningún líder informático quiere embarcarse en un proyecto “ambiciosamente imposible” que consuma recursos y genere riesgos sin una alta probabilidad de recompensas concretas, cuantificables y a corto plazo.

Este estudio aborda estas preocupaciones y explica en detalle un enfoque en fases, probado, para modernizar de manera medible las prácticas en mainframe. Al seguir este enfoque, los líderes informáticos pueden obtener “triumfos” de gran repercusión en cada paso del camino, a corto plazo, mientras siguen encauzados hacia el objetivo estratégico crucial que puede conseguirse completamente en menos de dos años.

Puesto que cada organización tiene sus propios procesos, herramientas y cultura existentes, esta estrategia de transformación del mainframe puede modificarse para que se ajuste a las necesidades específicas de cada organización.

Sin embargo, ninguna empresa puede permitirse posponer la transformación del mainframe o embarcarse en ella sin un plan claro. Para seguir siendo competitivos en una economía centrada en las aplicaciones, los mainframes deben ser tan adaptables como otras plataformas. Y el departamento de IT de las empresas debe ser capaz de gestionar DevOps de manera integrada en el mainframe, en entornos distribuidos y en la nube.

Este estudio traza un plan para hacer exactamente eso.



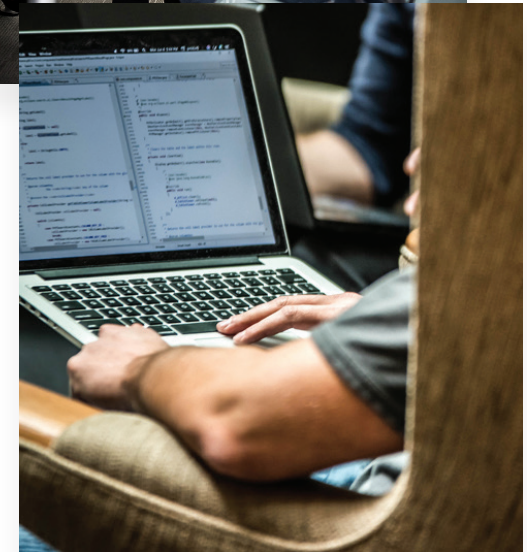
Paso 1:

DEFINIR EL ESTADO DESEADO

Antes de embarcarse en el proceso de transformación del mainframe, es sensato primeramente tener en claro los objetivos. Sus objetivos específicos dependerán de hasta dónde se hayan adoptado ya estrategias tales como DevOps y Continuous Delivery a lo largo de sus otras plataformas. No obstante, entre los objetivos de transformación del mainframe normalmente se incluyen:

- Agilidad que permite a su organización hacer cambios frecuentes y correctamente dimensionados en el código del mainframe, que satisfagan de manera precisa las necesidades actuales y más urgentes del negocio.
- Seguridad en que los cambios hechos al código del mainframe funcionarán en producción según lo esperado, sin afectar negativamente la disponibilidad, el rendimiento, la experiencia del cliente, la seguridad o la conformidad con las normas corporativas y/o legislativas.
- Eficiencia en el uso del tiempo y las habilidades del personal en relación con el desarrollo, las pruebas y la producción.
- Facilidad de uso que faculte incluso al personal sin experiencia en mainframe a participar y contribuir en las mejoras y actualizaciones de las aplicaciones.
- Integración del mainframe y no mainframe en los procesos DevOps corporativos de manera que permita mejorar y estandarizar el soporte y la entrega de servicios multiplataforma.

Tener en claro estos objetivos (y su prioridad relativa) le ayudará a decidir si modificar su plan de transformación y cómo hacerlo. Por ejemplo, si su punto débil más grande e inminente es la pérdida de programadores de COBOL veteranos porque se jubilan, probablemente querrá abordar la facilidad de uso en primer lugar. Si, por otro lado, su punto más débil es que la lentitud del plazo de entrega de las actualizaciones de las aplicaciones de mainframe le está impidiendo añadir nuevas características necesarias y urgentes en sus aplicaciones móviles, quizá quiera priorizar la agilidad.





Paso 2:

MODERNIZAR EL ENTORNO DE DESARROLLO MAINFRAME

La mayor parte del desarrollo del mainframe todavía se realiza en anticuados entornos ISPF “de pantalla verde” que requieren un conocimiento muy especializado y, de manera problemática, limitan la productividad del nuevo personal. Modernizar el mainframe comienza con la modernización del entorno de trabajo del desarrollador.

Un entorno de trabajo mainframe modernizado debería tener la apariencia y el comportamiento de los IDE al estilo Eclipse que, de hecho, se han convertido en el estándar para otras plataformas. Esta interfaz fácil de usar permitirá que el personal con distinto nivel de experiencia realice fácilmente el desarrollo y las pruebas tanto de aplicaciones mainframe como de aplicaciones no mainframe. El IDE modernizado de mainframe también soportará una variedad complementaria de herramientas de valor añadido a medida que continúe con la transformación de su mainframe a través de los pasos posteriores.

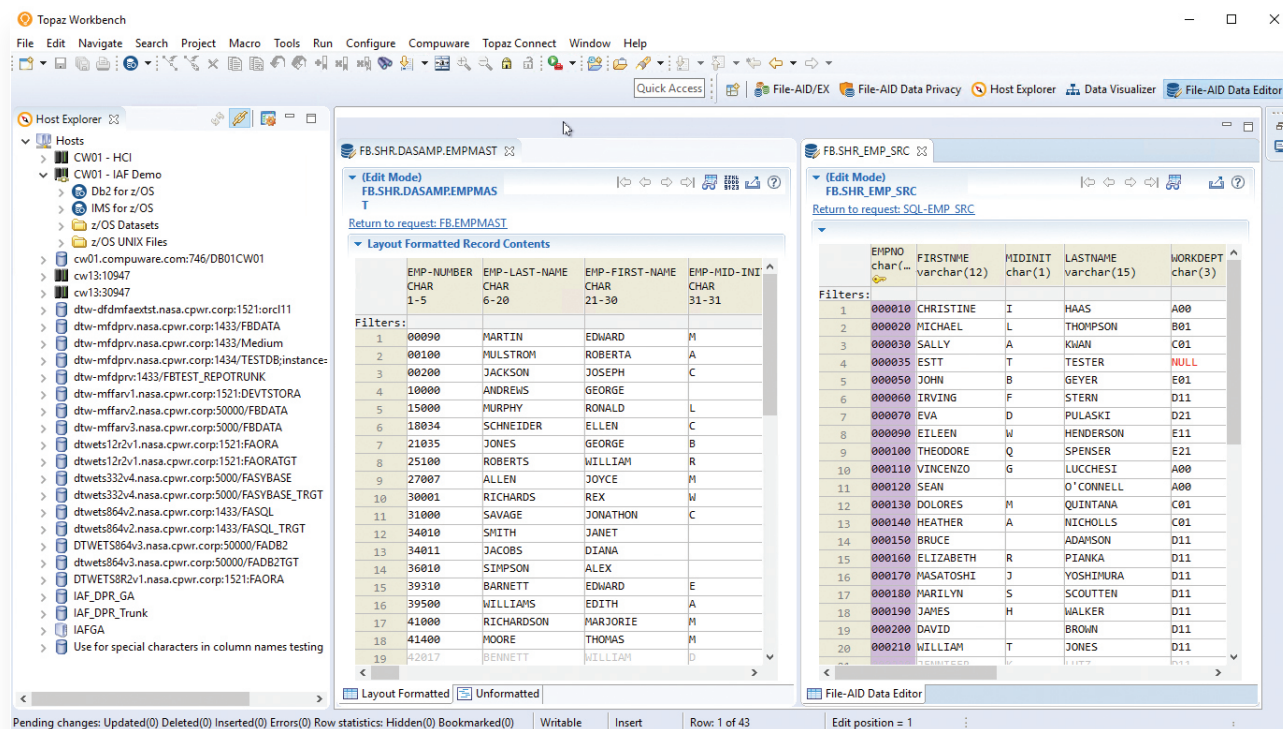
EMP-FIRST-NAME	EMP-MID-INIT
CHAR	CHAR
21-30	31-31
EDWARD	M
ROBERTA	A
JOSEPH	C
GEORGE	
RONALD	L
ELLEN	C
GEORGE	B
WILLIAM	R
JOYCE	M
REX	W
JONATHAN	C



HERRAMIENTAS

La tecnología facilitadora para el paso 2 es **Topaz Workbench de Compuware**, junto con las soluciones asociadas de Compuware, tales como **File-AID**, **Abend-AID** y **Xpediter**. Estos productos proporcionan una funcionalidad integral para la edición del código fuente, la depuración, el diagnóstico de fallos y la visualización/edición/comparación de datos. Topaz Workbench, en particular, permite a los desarrolladores escribir, compilar y ejecutar un código, todo desde un moderno entorno de desarrollo integrado basado en Eclipse. También aporta la facilidad de arrastrar y soltar para copiar ficheros entre particiones lógicas y en otras tareas comunes de los desarrolladores. Esto sitúa la experiencia de desarrollo en mainframe a la par de otras tecnologías de la empresa, tales como Java.

Obtenga más información sobre el **Topaz Workbench**.



Una sesión de File-AID en Topaz Workbench proporciona una interfaz gráfica para la edición de datos.

INDICADORES DE ÉXITO

- **Métricas empíricas de productividad** tales como funcionalidades entregadas, frecuencia de cambios de código, y la curva de aprendizaje reducida para los nuevos empleados
- **Comentarios positivos** de los equipos de desarrollo y pruebas
- **Habilidad para motivar a los desarrolladores no relacionados con el mainframe** a trabajar en actividades de desarrollo/pruebas relacionadas con mainframe



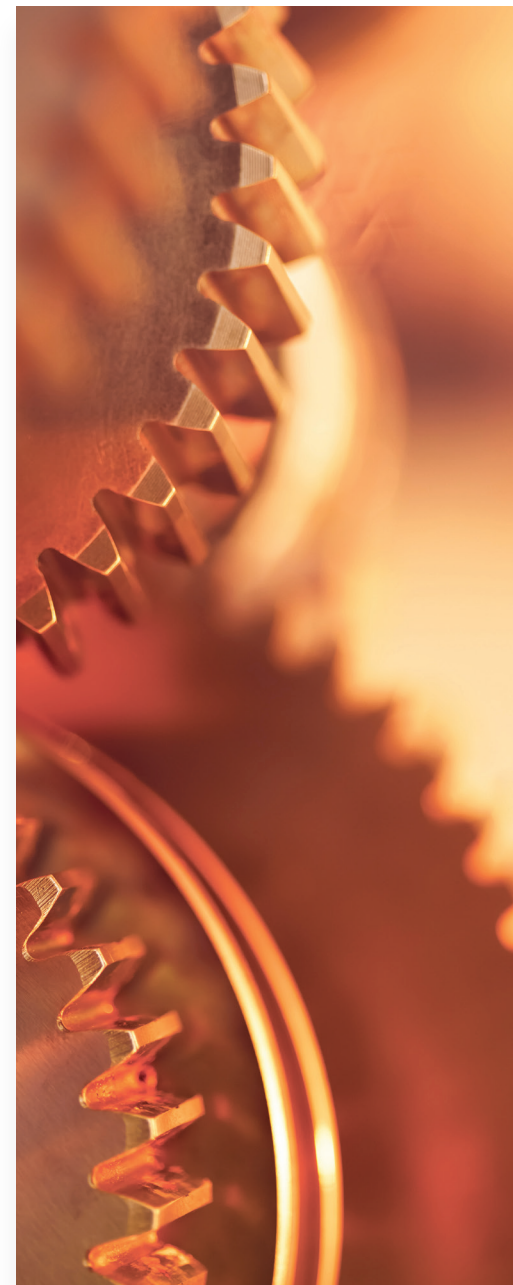
Paso 3:

ADOPTAR PRUEBAS UNITARIAS AUTOMATIZADAS

Las pruebas unitarias son clave para Agile. Hacer pruebas de manera frecuente en pequeños incrementos de código permite a los desarrolladores evaluar de manera rápida y continuada lo cerca que se alinea su trabajo actual a los objetivos de equipo inmediatos, de manera que se pueden hacer los ajustes necesarios rápidamente o seguir adelante con la siguiente tarea.

Por desgracia, los obstáculos técnicos históricamente han impedido que el tipo de pruebas unitarias que son comunes en Java sean de aplicación en el desarrollo en mainframe. Ahora que se han eliminado los obstáculos según se describe a continuación, las pruebas unitarias automáticas se pueden convertir en una realidad en el mainframe.

Por supuesto, las pruebas unitarias efectivas necesitan más que tecnología. Los desarrolladores de mainframe que no están acostumbrados a hacer pruebas unitarias deben aprender a cómo usar mejor la práctica para trabajar de manera iterativa en piezas de código mucho más pequeñas. Y los gestores de DevOps necesitan implementar controles que aseguren que las pruebas unitarias se llevan a cabo con éxito antes de promocionar el código. Pero las pruebas unitarias en todas las plataformas son un requisito fundamental para Agile.



HERRAMIENTAS

La tecnología facilitadora para el paso 3 es **Compuware Topaz for Total Test**. El producto descompone automáticamente el código de mainframe en unidades lógicas discretas y crea pruebas (incluyendo los resultados por defecto de las pruebas unitarias) para esas unidades lógicas. Esta creación de pruebas automatizada faculta a los desarrolladores, de todos los niveles de conocimiento, a validar de manera rápida, fácil y precisa y solucionar cualquier cambio que hagan en las aplicaciones mainframe.

Encuentre más información sobre **Topaz for Total Test**.

INDICADORES DE ÉXITO

- **Entregas más frecuentes de actualizaciones** de código necesario para el negocio
- **Menos errores encontrados más adelante** en el ciclo de vida DevOps
- **Colaboración estrecha y sincronizada en el desarrollo** conjunto de mainframe y no mainframe

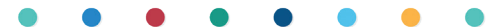
The screenshot displays the Topaz Workbench application. The main window shows a 'Test Project Demo_2018_08_1' with a 'Test Suite Summary' table. Below this, there are sections for 'Test Scenarios' and 'CWBCWCOBX_Scenario Test Scenario'. The bottom pane shows a list of test jobs with their status, owner, and submission time.

Name	Tests	Failures	Errors	Success rate	Time stamp	Job ID	System
CWBCWCOBX_Scenario	338	0	0	100%	2018-05-25 13:01:35	PFHSX00XJ0573591	CWCC LPAR

Test Case	Tests	Failures	Errors	Success rate
CWBCWCOBXCase1	338	0	0	100%

Job	Job ID	Owner	Description	Max RC	Class	Programmer	Submit Time	Position
PFHSX001	J0572739	PFHSX00	Cancelled		A	ISPW TRAINING	2018 MAY 24 09:39:17	1943
PFHSX00G	J0572412	ISPWXS	Ended with warnings	4	A	CWBCWCOBX - ISPW	2018 MAY 23 16:59:48	1682
PFHSX00G	J0572406	ISPWXS	Ended with warnings	4	A	CWBCWCOBX - ISPW D	2018 MAY 23 16:54:39	1686
PFHSX00G	J0572403	ISPWXS	Ended with errors	8	A	CWBCWCOBX - ISPW D	2018 MAY 23 16:53:56	1683
PFHSX00G	J0572400	ISPWXS	Ended with warnings	4	A	CWBCWCOBX - ISPW D	2018 MAY 23 16:45:03	1680
PFHSX00G	J0572397	ISPWXS	Ended with errors	8	A	CWBCWCOBX - ISPW D	2018 MAY 23 16:44:17	1676
PFHSX00G	J0572394	ISPWXS	Ended with warnings	4	A	CWBCWCOBX - ISPW D	2018 MAY 23 16:43:45	1673
PFHSX00G	J0572390	ISPWXS	Ended with errors	12	A	CWBCWCOBX - ISPW D	2018 MAY 23 16:42:45	1670

Configurar criterios de éxito de pruebas con Topaz for Total Test para validar que las partes del programa se ejecutan correctamente.



Paso 4:

GANAR VISIBILIDAD GRÁFICA E INTUITIVA DENTRO DE LA ESTRUCTURA EXISTENTE DE CÓDIGOS Y DATOS

Dado que las aplicaciones de mainframe han aumentado y se han expandido a lo largo de muchos años, se han vuelto bastante grandes y complejas. Además, normalmente no están muy bien documentadas. Esta combinación de complejidad y falta de documentación es un gran impedimento para los objetivos esenciales de la transformación, incluidas la agilidad, la seguridad y la eficiencia. De hecho, las peculiaridades no documentadas de las aplicaciones de mainframe y las estructuras de datos hacen, casi de manera universal, que TI corporativa dependa del conocimiento tribal/personal del personal de mainframe experimentado. Incluso peor, si un desarrollador de mainframe experimentado ya no está, el personal informático puede incluso temer hacer cualquier cambio.

Para superar esta dependencia, debe facilitar a cualquier nuevo miembro del equipo "leer" rápidamente la lógica de las aplicaciones existentes, las interdependencias entre los programas, las estructuras de datos y las relaciones entre datos. Los desarrolladores y otro personal técnico también necesitan poder comprender el comportamiento de la aplicación durante el tiempo de ejecución, incluyendo la secuencia real y la naturaleza de las llamadas del programa, así como el I/O de/a los ficheros y bases de datos, de manera que puedan trabajar incluso en los sistemas más complejos y desconocidos con claridad y seguridad.



HERRAMIENTAS

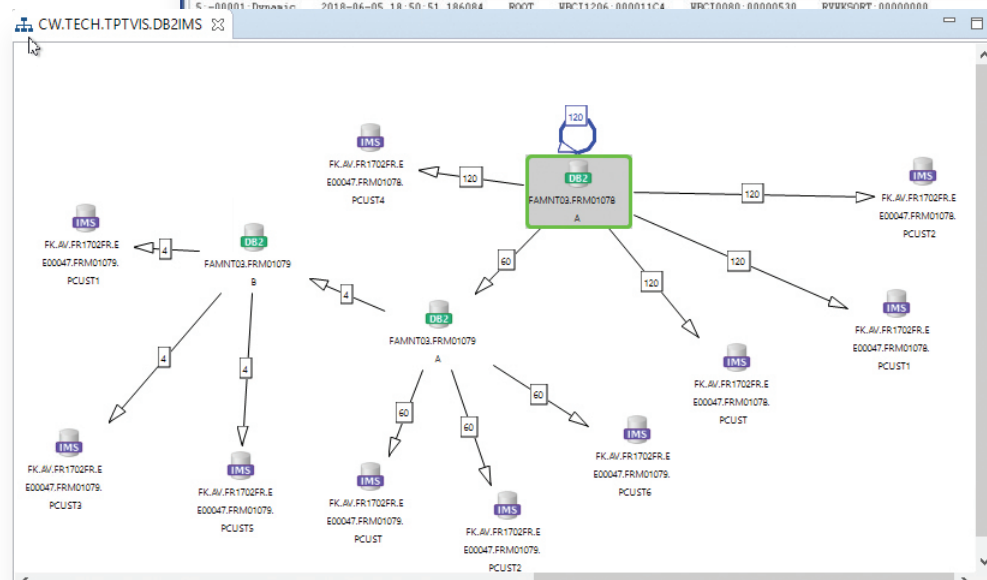
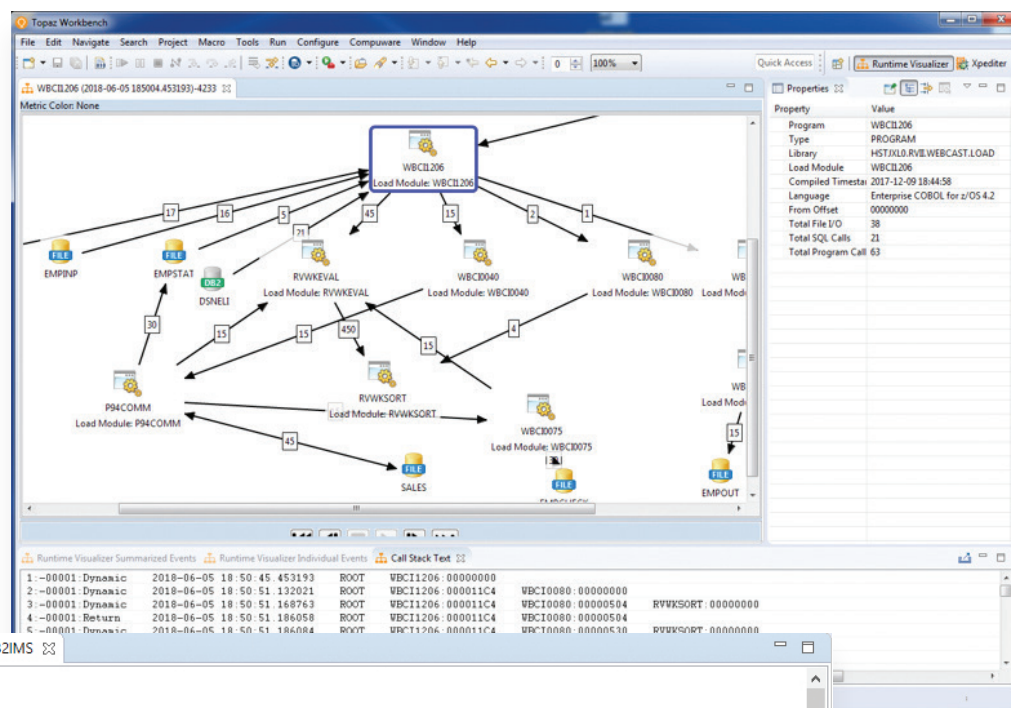
Las tecnologías facilitadoras para el paso 4 son **Compuware Topaz Program Analysis** y **Topaz for Enterprise Data**. Esas visualizaciones únicas y potentes revelan la lógica de los programas y las relaciones entre los datos a través de diagramas de gráficos intuitivos generados de manera dinámica. Estos diagramas muestran cómo están estructurados los programas COBOL y PL/I considerando variables y ficheros asociados, a la vez que permiten a los desarrolladores grabar, guardar y comparar visualizaciones del comportamiento de los programas durante el tiempo de ejecución, sin acceder a los ficheros de código fuente.

Encuentre más información en **Topaz for Program Analysis** y **Topaz for Enterprise Data**.

INDICADORES DE ÉXITO

- **Desarrolladores trabajando de manera independiente** con programas que desconocían
- **Desarrolladores con experiencia confirmando la exactitud y la utilidad** de las visualizaciones de los programas
- **Desarrolladores estimando mejor el esfuerzo necesario** para cumplir con los plazos del proyecto

La funcionalidad Runtime Visualizer de Topaz for Program Analysis muestra las llamadas realizadas por los programas y las actividades I/O que ayudan a los programadores a obtener conocimientos inestimables de las aplicaciones con falta de documentación.



La funcionalidad Extract Visualizer de Topaz for Enterprise Data reúne las evidencias críticas necesarias para crear datos de pruebas óptimos, incluyendo información del uso de la CPU, información de camino de acceso DB2 y sentencias SQL.



Paso 5:

HABILITAR LA DETECCIÓN TEMPRANA DE PROBLEMAS DE CALIDAD DE LA APLICACIÓN Y ESTABLECER INDICADORES CLAVE DE RENDIMIENTO DE LA CALIDAD

La detección temprana de los problemas de calidad reduce los costes y previene que los errores relativamente pequeños en el código se conviertan en un gran dolor de cabeza después en el proceso DevOps. Al hacer la transición desde las metodologías en cascada a Agile, es importante utilizar herramientas y técnicas modernas que permitan que se den procesos de calidad de manera continua a lo largo de los ciclos de desarrollo.

La detección temprana es particularmente importante en la transformación del mainframe por varias razones. Por un lado, las aplicaciones del mainframe a menudo se asocian con procesos de negocio esenciales que tienen una baja tolerancia al error. Por otro lado, a medida que los desarrolladores con menos experiencia en mainframe comienzan a contribuir en más proyectos relacionados con mainframe, la probabilidad de error humano aumenta.

Deben hacerse todos los esfuerzos posibles para salvaguardar de manera rigurosa la calidad de las aplicaciones a medida que el mainframe va siendo más ágil. La integración continua es especialmente importante en este contexto, dado que asegura que las validaciones de calidad se realizan continuamente, a medida que se modifica el código.



HERRAMIENTAS

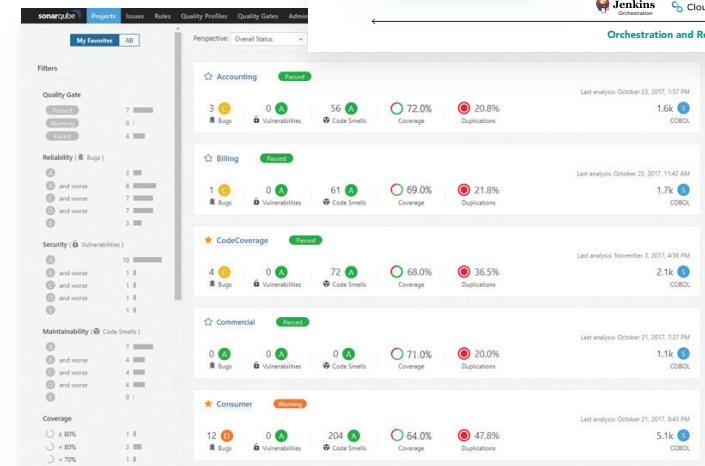
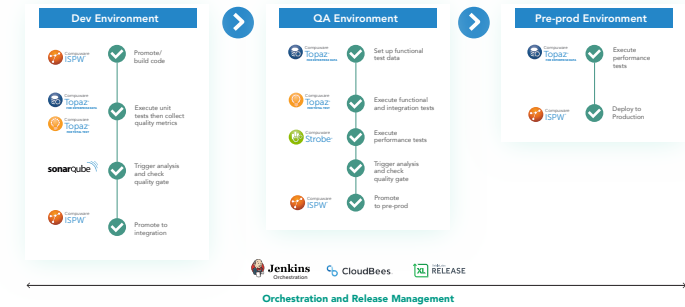
Las tecnologías adicionales para el paso 5 son **SonarLint de SonarSource** y **SonarQube**. SonarLint se integra a Topaz Workbench para proporcionar a los desarrolladores comentarios en tiempo real sobre la posibilidad de nuevos errores y problemas de calidad introducidos en su código. Con SonarLint, incluso los desarrolladores sin experiencia en mainframe pueden ser rápidamente alertados de los problemas de calidad de las aplicaciones, tales como posibles excepciones de memoria, exceso de condiciones anidadas (IF), o de ramificaciones (GOTO) fuera de alcance. SonarQube es un cuadro de mandos muy completo para el seguimiento de defectos, complejidad de los programas y deuda técnica. Proporciona a los gestores una clara visibilidad de las métricas individuales, del equipo, y métricas de calidad del proyecto, de manera que puedan detectar rápidamente las áreas que requieren una atención/formación adicional.

Jenkins, el servidor de automatización open source, es también especialmente importante en este paso porque proporciona funcionalidad de Integración Continua (IC). Jenkins puede utilizarse para llevar a cabo una serie de controles de calidad que pueden desencadenarse por cambios de código y otros eventos relevantes en el proceso de desarrollo.

Encuentre más información sobre [SonarQube](#), [SonarLint](#) y [Jenkins](#).

Continuous Delivery

INTEGRATE DATA INTO AUTOMATED DEVOPS TOOLCHAIN



El valioso cuadro de mandos de SonarQube identifica defectos, complejidades y deuda técnica a nivel individual, de equipo y de proyecto.

INDICADORES DE ÉXITO

- **Tasas más altas de detección de errores** antes de la compilación y de corrección de esos errores por los desarrolladores
- **Evolución positiva en métricas de calidad** y reducción de la deuda técnica
- **Número reducido de iteraciones** ,debido a errores, entre los equipos de desarrollo y pruebas

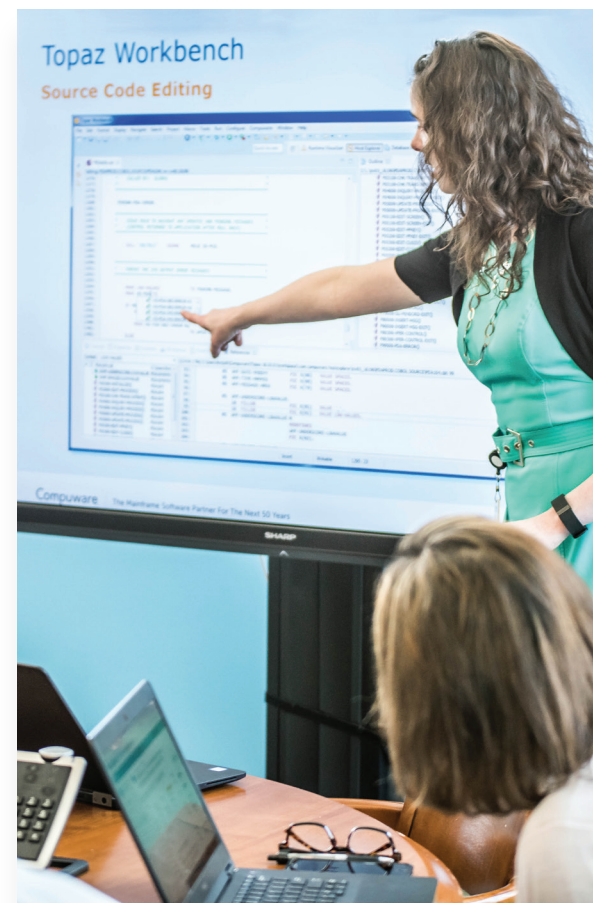
Paso 6:

FORMACIÓN INICIAL Y ADOPCIÓN DE PROCESOS ÁGILES

En este momento del viaje debería tener a punto el entorno de desarrollo adecuado, mientras que sus equipos de desarrollo deberían estar preparados para una formación real en las metodologías de desarrollo Agile. Una vez completado, podrá comenzar a cambiar sus procesos desde un modelo en cascada tradicional con amplios requisitos y largos plazos de proyecto a un modelo más incremental. El objetivo es que los desarrolladores de componentes mainframe, web y móvil colaboren en un solo equipo "Scrum". Los equipos se centran en "Epics" e historias de usuario que capturan unidades específicas de valor para el negocio, versus las tareas técnicas de un plan de proyecto. Al estimar el tamaño de estas historias y asignándoles su prioridad correspondiente, sus equipos pueden empezar a involucrarse en procesos Agile que les permitan moverse rápidamente hacia sus objetivos.

El paso desde los proyectos en cascada de gran escala hasta Agile representa un cambio significativo en la cultura de trabajo para la mayor parte de los equipos de mainframe. Por lo tanto, la formación en procesos Agile es obligada. Los puestos de liderazgo técnico y los propietarios de productos (Product Owners), en particular, necesitan una formación exhaustiva y coaching. No obstante, todos los miembros del equipo deberían tener, por lo menos, una introducción a los conceptos básicos de Agile, especialmente si se espera que lean pizarras "Scrum" (Scrum Board) o "Kanban."

Quizá quiera crear su equipo inicial Agile de mainframe seleccionando desarrolladores de mainframe específicos y emparejándolos con desarrolladores con experiencia en Agile de otras plataformas. Quizá también quiera considerar cómo gestionará el cumplimiento de los valores de Agile, tales como la transparencia, el intercambio de conocimientos y la generación de ideas en las evaluaciones periódicas de los desarrolladores.



HERRAMIENTAS

Las tecnologías facilitadoras para el paso 6 son **Atlassian JIRA** y **Confluence**. JIRA es una herramienta de gestión de tareas de Agile que es compatible con cualquier metodología Agile, incluidas Scrum y Kanban. Le permite planificar, hacer seguimiento y gestionar toda la actividad de desarrollo de Agile para que sus equipos estén correctamente encaminados y mejoren continuamente la adopción de Agile en términos de velocidad, eficiencia, calidad y, lo que es más importante, de alineación continua con las necesidades más urgentes de su negocio.

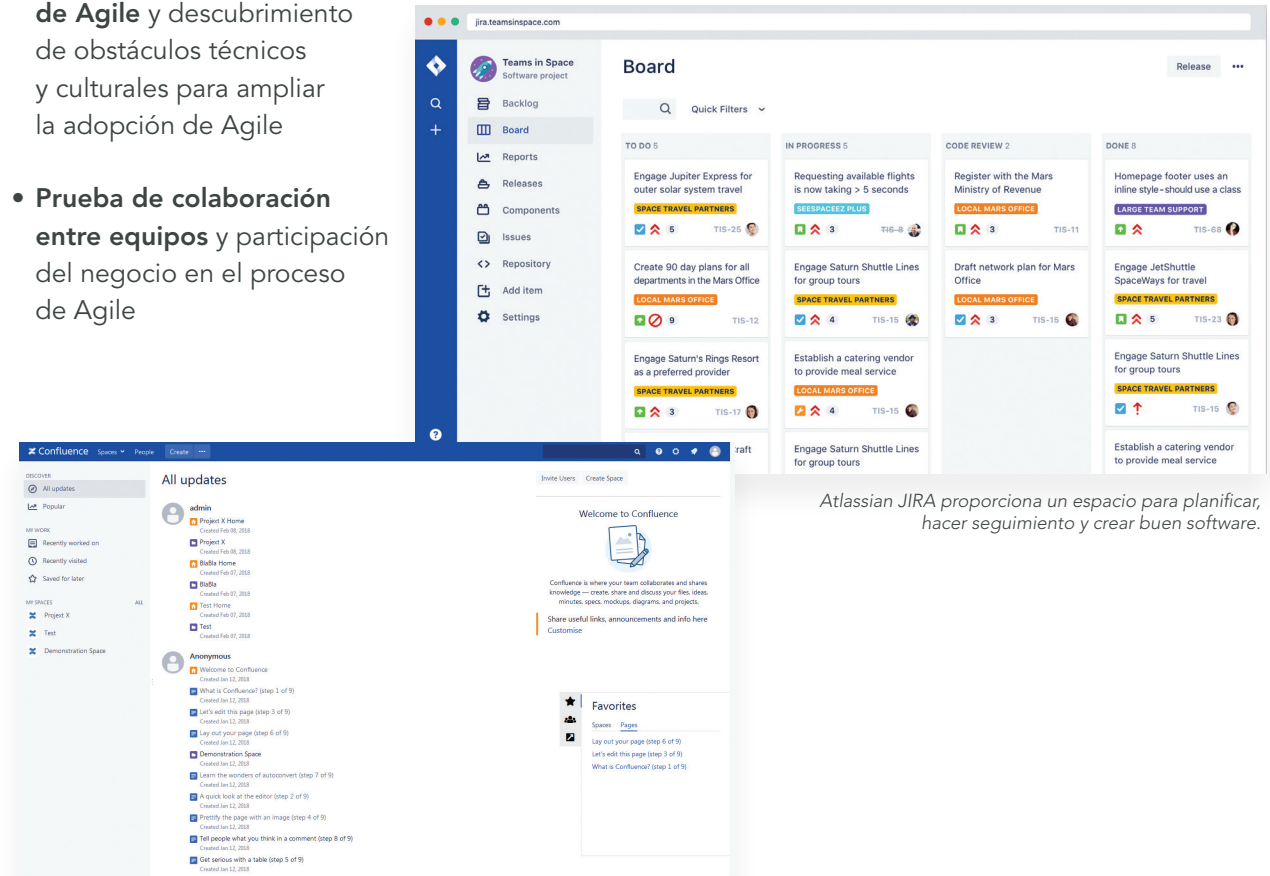
Atlassian Confluence complementa a JIRA al proporcionar un espacio de colaboración web bien organizado y centralizado donde sus equipos de Agile pueden compartir de manera fácil y flexible ideas y requisitos de productos, así como actualizaciones de estado y procesos, etc. Este tipo de colaboración facilita la cultura de innovación necesaria.

Hacer uso de herramientas de reconocido prestigio como las que proporciona Atlassian es normalmente más sensato que adoptar un enfoque monolítico que le haga realizar todas las actividades SDLC (Ciclo de Vida de Desarrollo de Software) en un conjunto de soluciones de un único proveedor, dado que le permite evitar el monopolio del proveedor a la vez que se beneficia de las nuevas innovaciones en la gestión de Agile.

Encuentre más información en **Atlassian Jira** y **Confluence**.

INDICADORES DE ÉXITO

- Enfocarse en un porcentaje de **personal de desarrollo y pruebas que complete la formación de Agile** con el objetivo del 100 % formados
- **Primera entrega de artefactos desde los equipos iniciales de Agile** y descubrimiento de obstáculos técnicos y culturales para ampliar la adopción de Agile
- **Prueba de colaboración entre equipos** y participación del negocio en el proceso de Agile



Atlassian JIRA proporciona un espacio para planificar, hacer seguimiento y crear buen software.

Crear, compartir y colaborar en proyectos en un solo espacio con Confluence para hacer que los proyectos avancen más deprisa.

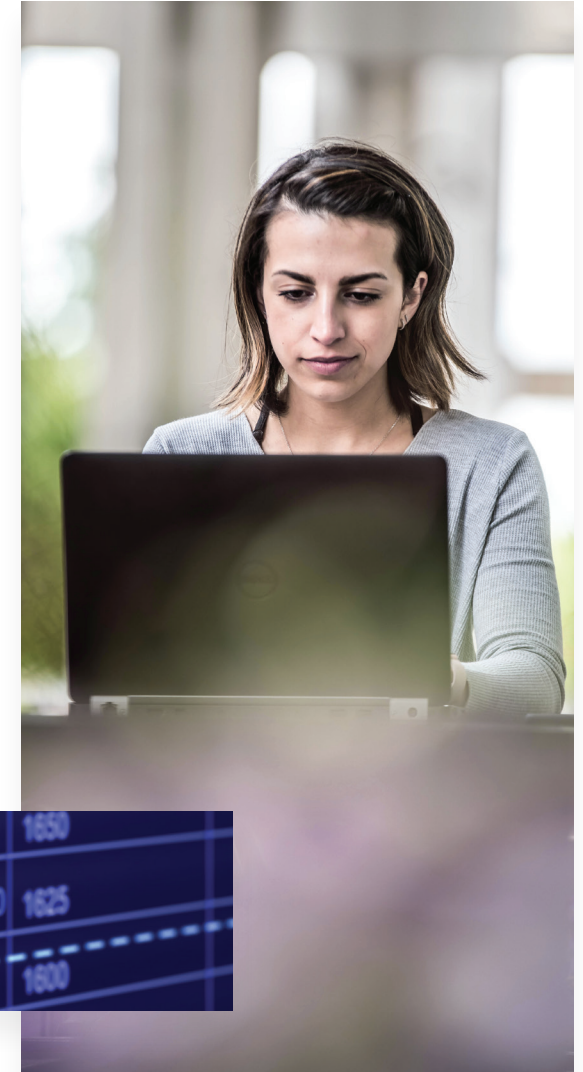


Paso 7:

UTILIZAR DATOS OPERATIVOS A LO LARGO DEL CICLO DE VIDA DE DESARROLLO, PRUEBA Y PRODUCCIÓN

Para asegurarse de que sus aplicaciones tengan un óptimo rendimiento en el entorno de producción, no basta solamente con escribir buen código. También debe comprender exactamente cómo se comportan sus aplicaciones, dado que consumen capacidad de procesamiento, acceden a sus bases de datos e interactúan con otras aplicaciones.

Una buena manera de entenderlo mejor es usar los datos operacionales de manera continua a lo largo del ciclo de vida de DevOps. Esto les proporciona a los equipos de Desarrollo y Operaciones un entendimiento común de las métricas/características operativas de una aplicación en su ciclo de vida, y les ayuda a medir el progreso hacia los objetivos del equipo de una manera más completa y precisa. El uso temprano de datos operativos también puede reducir drásticamente sus costes relacionados con MIPS/MSU, lo que le permite descubrir y mitigar un consumo de CPU evitable causado por un código ineficiente.



HERRAMIENTAS

Las tecnologías facilitadoras para el paso 7 son **Compuware Abend-AID** y **Strobe**. Abend-AID proporciona un análisis a nivel de fuente de los fallos de las aplicaciones, lo que ahorra tiempo a los desarrolladores y elimina el trabajo de cruzar manualmente referencias de numerosas páginas de “dumps” de memoria, listados de fuentes y código de aplicación. Con Abend-AID, el personal de desarrollo y pruebas puede señalar rápidamente las sentencias erróneas de código, identificar los problemas de datos y aislar muchos otros tipos de problemas de las aplicaciones.

Strobe detecta las ineficiencias de las aplicaciones, tales como las sentencias SQL incorrectas, los servicios de sistema DB2 ineficientes que causan un consumo de CPU excesivo, la lenta recuperación de datos y otros problemas que incrementan costes a la vez que socavan la experiencia del usuario final. Al identificar estos problemas de manera automática, Strobe le permite incluso al personal de desarrollo/pruebas más inexperto contribuir a la entrega más rápida de aplicaciones con mejor rendimiento.

Encuentre más información en **Abend-AID** y **Strobe**.

INDICADORES DE ÉXITO

- **Detección temprana de consumo de CPU evitable**
- **Reducción en Abends en producción**
- **Reducción en el coste medio por error y el tiempo medio de resolución**

Root Cause Analysis Information

Fault Details

1. Program **CWAACOB1** contains the statement in error.
2. The content of **H-EMP-RATE** does not match its field definition. It contains the value **X'025C5B'**. This was likely the cause of the S0C7 abend.
3. **H-EMP-RATE** is defined within the **01 FILES** structure in the LINKAGE SECTION of called program **CWAACOB1**. Its value was passed by calling program **CWAACOB1**.
4. Field in error corresponds to field **03 SA-EMP-RATE** in calling program **CWAACOB1**.

Level/Field Name	Picture/Type	Value
02 H-EMP-RATE	# 9(3)V99 COMP-3	X'025C5B'
01 EMPLOYEE-SALARY-AREA		
02 SA-HOURLY-EMPLOYEE-DATA		
03 SA-EMP-RATE	# 9(3)V99 COMP-3	X'025C5B'

defined in LINKAGE SECTION in program **CWAACOB1**.

defined in WORKING-STORAGE SECTION in program **CWAACOB1**.

5. There are open **Files**, however: no association could be made to this field **H-EMP-RATE**.

Error Statement Information

Recommended Actions

Verify that field **H-EMP-RATE** is defined and used correctly/appropriately in program **CWAACOB1** and has not been corrupted in **CWAACOB1**.

- Select **H-EMP-RATE** to see a Field Usage Summary.
- Select **X'025C5B'** to see other fields in the same program with the same value.
- Select **X'025C5B'** to see literals in the same program with the same value.

Abend-AID aporta pasos claros a seguir para reducir el tiempo de resolución de problemas.

Measurement Time

User Application CPU

Source statement

Module/Section Attributes

Compiler Options That May Impact Performance

CPU 1: INSPBAD1 - 98.11% (8,176.88 sec) of CPU

CPU 2: INSPBAD1 - 1.34% (84.42 sec) of CPU

CPU 3: INSPBAD1 - 0.89% (5.52 sec) of CPU

CPU 4: INSPBAD1 - 0.88% (5.09 sec) of CPU

CPU 5: INSPBAD1 - 0.88% (5.09 sec) of CPU

Strobe Advisor muestra a los desarrolladores los principales componentes de las aplicaciones que afectan el rendimiento.



Paso 8:

HABILITAR Y AGILIZAR LAS FUNCIONES CLAVE EN LA GESTIÓN DE CÓDIGO FUENTE

Los entornos SCM tradicionales están inherentemente diseñados para el desarrollo en cascada y, por eso, son incapaces de proporcionar capacidades de Agile esenciales como, por ejemplo, el trabajo de desarrollo paralelo en diferentes historias de usuario.

El cambio desde SCM basado en modelo de cascada a un SCM compatible con Agile es un momento crucial en la transformación del mainframe y debe plantearse cuidadosamente para evitar la interrupción del trabajo que esté actualmente en progreso. Es, sin embargo, un cambio completamente esencial para cualquier organización informática que espera aumentar la velocidad y la frecuencia con la que se pueden preparar nuevas entregas de código mainframe, y reducir el tiempo perdido por los desarrolladores a la espera de que haya nuevo código disponible.

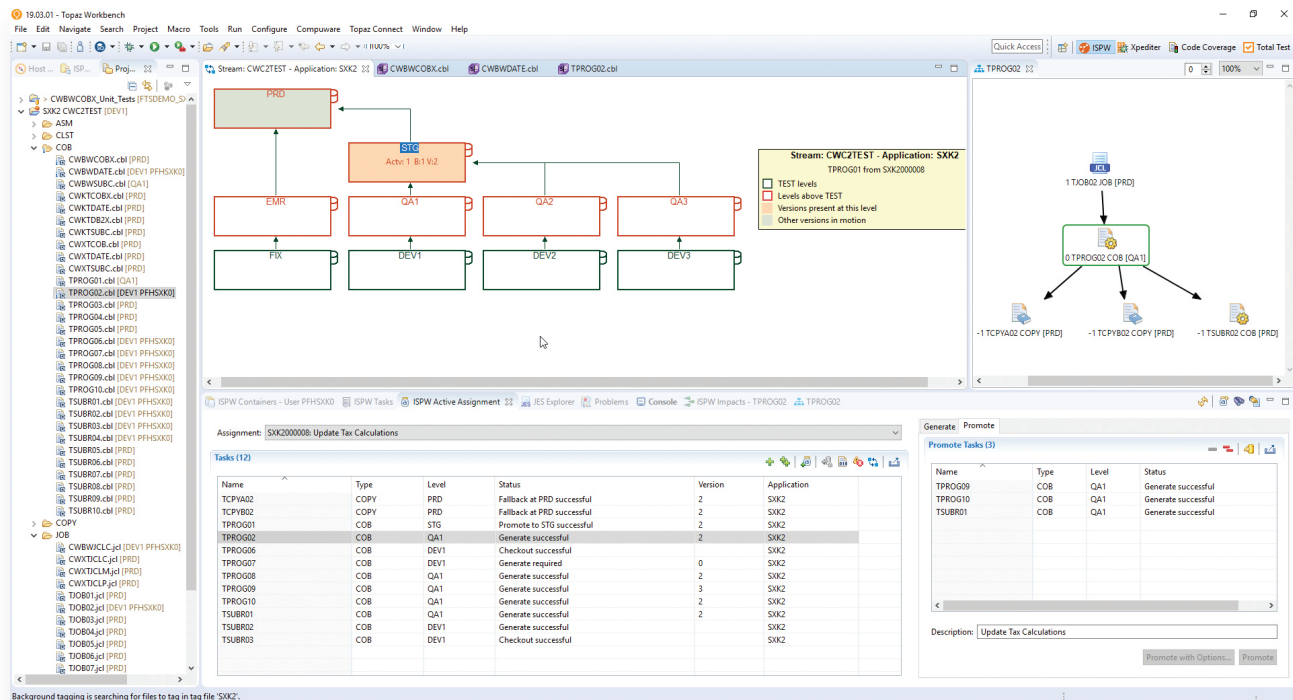




La tecnología facilitadora del paso 8 es **SCM de Compuware - ISPW**. ISPW es un moderno SCM, Agile de principio a fin, y una solución de automatización de despliegues que permite a los desarrolladores de mainframe de todos los niveles de conocimiento cumplir con los requisitos del negocio, optimizar la calidad del código y mejorar la productividad del desarrollador a través de DevOps de mainframe. Proporciona una gestión de cambios automatizada que elimina los pasos manuales y faculta a sus equipos para moverse rápidamente a través de los ciclos de desarrollo, pruebas y control de calidad. ISPW SCM también está completamente integrado en el entorno Topaz Workbench, de manera que los desarrolladores tienen acceso completo a todas sus herramientas desde una sola interfaz intuitiva.

Encuentre más información en [ISPW SCM](#).

Compuware también ofrece [servicios de migración SCM](#) para agilizar y hacer menos riesgosa esta conversión crítica.



ISPW SCM proporciona visualización gráfica de dónde está el código en el ciclo de vida de desarrollo y cómo se relaciona con otros programas, jobs y copybooks.

INDICADORES DE ÉXITO

- Desarrolladores que ya no trabajan con “librerías personales” y diferentes equipos Agile trabajando en diversas historias paralelamente
- Los desarrolladores con diferentes niveles de habilidades de mainframe son capaces de entender rápidamente el alcance de los cambios antes de empezar a codificar
- Reducción de los retrasos en las aprobaciones del código



Paso 9:

IMPLANTACIÓN AUTOMATIZADA E INTELIGENTE

El desarrollo de Agile por sí solo es insuficiente para la completa agilidad de un negocio digital. Para seguir el paso de los mercados de hoy que se mueven rápidamente, su negocio también debe ser capaz de incorporar de manera rápida y fiable un nuevo código en producción. Esto significa automatizar y coordinar el despliegue de todos los artefactos relacionados con el desarrollo en todos los entornos objetivo de una manera muy sincronizada. También necesitará detectar todos los problemas en el despliegue tan pronto como ocurran para que se puedan tomar acciones correctivas inmediatas.

Y si esa acción correctiva no es evidente de manera inmediata o no produce de inmediato el efecto correctivo esperado, tiene que poder volver a restaurar la versión de la aplicación que estaba en funcionamiento previamente de manera rápida y automática. Esta restauración automatizada es, de hecho, un facilitador clave en el rápido despliegue, dado que es el medio principal de mitigar el riesgo de negocio asociado con la promoción del código.



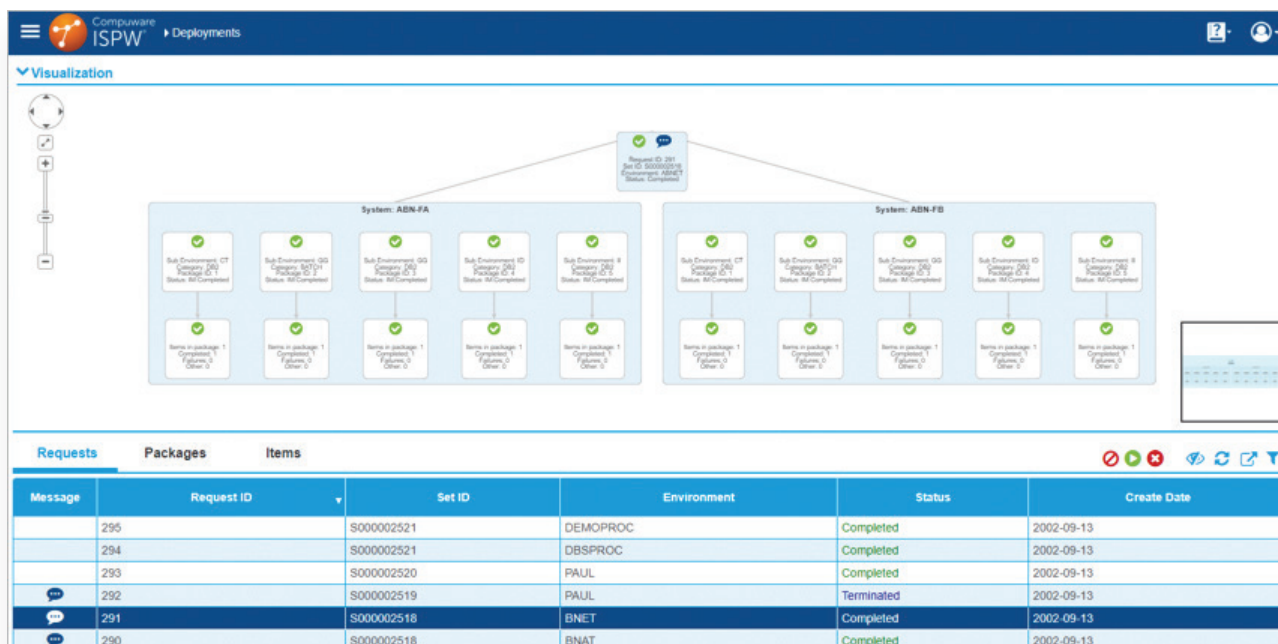
HERRAMIENTAS

La tecnología facilitadora del paso 9 es **ISPW Deploy**. ISPW Deploy complementa las capacidades esenciales de ISPW SCM con capacidades de despliegue de mainframe avanzadas para que rápidamente pueda mover código a través del proceso de despliegue, incluidas la fase de pruebas y las aprobaciones, a la vez que proporciona restauraciones parciales o totales muy simplificadas. Como tal, ofrece una significativa solución superior a los tradicionales "scripts" construidos en casa que no aportan las capacidades esenciales de restauración, visibilidad de progresos y auditabilidad de los procesos de despliegue.

ISPW Deploy también aporta visualización que permite a los gestores DevOps detectar rápidamente los problemas de despliegue, tanto para solventar de manera inmediata los problemas de instalación como para abordar cuellos de botella persistentes en la promoción de código.

ISPW Deploy cuenta con una interfaz móvil que permite a los gestores DevOps de mainframe responder inmediatamente cuando reciben una alerta de que hay cambios de código listos para su aprobación. Esta gestión móvil a cualquier hora/en cualquier lugar elimina una causa común de los retrasos en la promoción de código de mainframe.

Encuentre más información en **ISPW Deploy**.

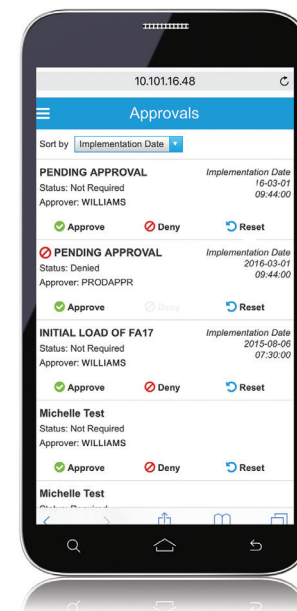


ISPW Deploy proporciona una vista del estado de todas las solicitudes de despliegue y de los entornos en los que se han instalado. Al hacer clic en el botón derecho en el campo de estado se muestra un gráfico jerárquico que guía a los operadores a cualquier punto de fallo, de manera que se pueden tomar acciones correctivas.

INDICADORES DE ÉXITO

- Lanzamientos más rápidos de aplicaciones a producción
- Reducción de fallos de promoción de código
- Primera restauración automatizada con éxito de una puesta en marcha fallida

Una interfaz móvil en ISPW Deploy permite a los gestores de DevOps responder inmediatamente a las alertas cuando hay cambios de código listos para su aprobación.





Paso 10:

ENTREGA CONTINUA MULTIPLATAFORMA

Las aplicaciones y datos de mainframe sirven, cada vez más, como recurso “back-end” para aplicaciones multiplataforma de trato directo con los empleados y clientes que incluyen componentes móviles, web y/o en la nube. Los equipos DevOps deben, por lo tanto, ser capaces de sincronizar completamente en todas las plataformas la entrega del código nuevo aprobado. Estos controles de despliegue deberían también contar con restauraciones entre plataformas e informes de progreso.

Este es el estado objetivo de DevOps de la empresa cuando se complete el paso 10: Un entorno unificado donde el mainframe sea solo una plataforma más (uno que, no obstante, sea especialmente escalable, fiable, de gran rendimiento, rentable y seguro) que se pueda modificar de manera rápida y adecuada, según sea necesario, para cumplir con las necesidades de su negocio por medio de cualquier recurso que esté disponible para tal fin.

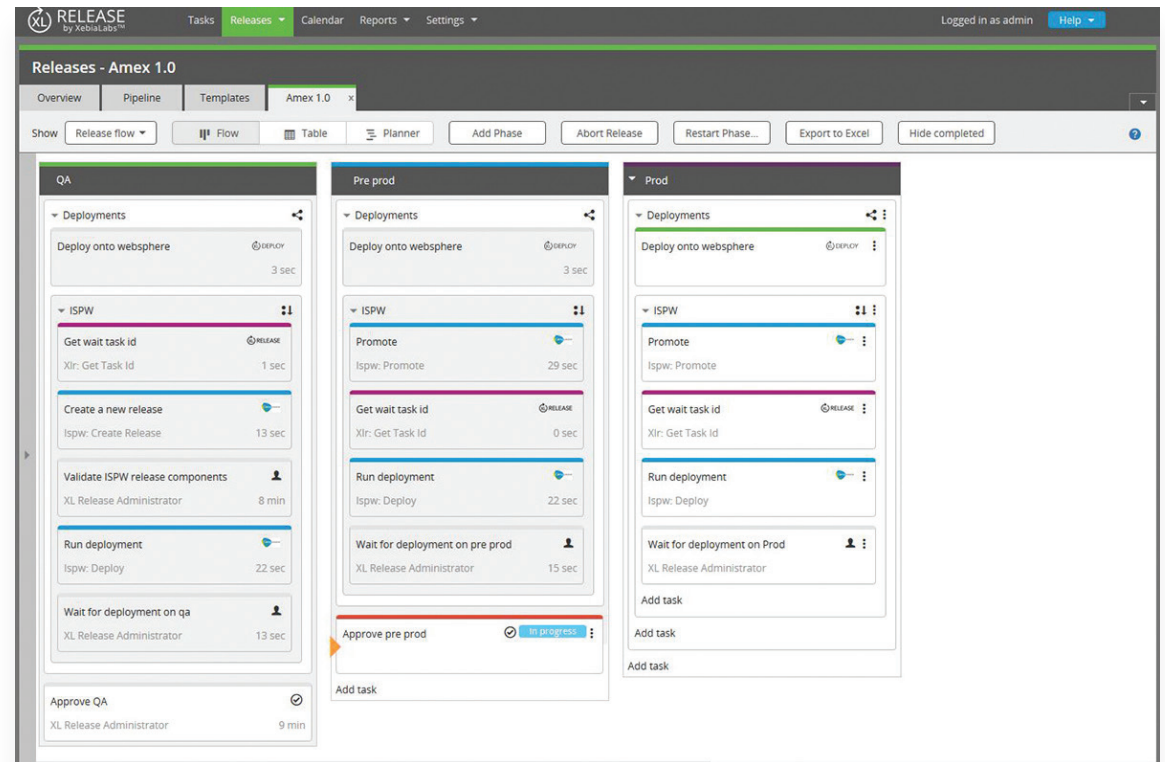


HERRAMIENTAS

Las tecnologías facilitadoras del paso 10 son las **ISPW REST API** y la integración con **XebiaLabs XL Release**. ISPW se integra con herramientas distribuidas para proporcionar un punto de control único para todos los cambios en z/OS, Windows, Unix y las demás plataformas. Las REST API son especialmente importantes para asegurar una completa flexibilidad en la mezcla y combinación de las herramientas de reconocido prestigio, así como para evitar el monopolio del proveedor.

XL Release es una solución altamente avanzada para la gestión del flujo del proceso de despliegue que facilita la planificación, automatización y el análisis de los despliegues de software multiplataforma para que mejore la promoción de código, mantenga completa visibilidad del progreso de despliegue, señale los problemas de instalación y aborde cualquier cuello de botella crónico en sus procesos DevOps.

Encuentre más información sobre la integración **XebiaLabs XL Release**.



ISPW y XebiaLabs XL Release se integran para permitir a las empresas automatizar, estandarizar y monitorizar los despliegues de código a través de múltiples plataformas en múltiples entornos de destino.

INDICADORES DE ÉXITO

- Capacidad para **trabajar con un código relacionado en múltiples plataformas en paralelo**
- **Mejora de comunicación y colaboración** entre desarrolladores previamente separados por diferencias de conocimientos y plataformas
- **Primer despliegue multiplataforma automatizado exitoso**



Paso 1:

DEFINIR EL ESTADO DESEADO

Paso 2:

**MODERNIZAR EL ENTORNO
DE DESARROLLO MAINFRAME**

Paso 3:

**ADOPTAR PRUEBAS UNITARIAS
AUTOMATIZADAS**

Paso 4:

**GANAR VISIBILIDAD GRÁFICA
E INTUITIVA DENTRO DE LA
ESTRUCTURA EXISTENTE DE
CÓDIGOS Y DATOS**

Paso 5:

**HABILITAR LA DETECCIÓN
TEMPRANA DE PROBLEMAS
DE CALIDAD DE LA APLICACIÓN Y
ESTABLECER INDICADORES CLAVE
DE RENDIMIENTO DE LA CALIDAD**

Paso 6:

**FORMACIÓN INICIAL Y
ADOPCIÓN DE PROCESOS ÁGILES**

Paso 7:

**UTILIZAR DATOS OPERATIVOS A
LO LARGO DEL CICLO DE VIDA DE
DESARROLLO, PRUEBA Y PRODUCCIÓN**

Paso 8:

**HABILITAR Y AGILIZAR LAS FUNCIONES
CLAVE EN LA GESTIÓN DE CÓDIGO
FUENTE**

Paso 9:

**IMPLANTACIÓN AUTOMATIZADA
E INTELIGENTE**

Paso 10:

ENTREGA CONTINUA MULTIPLATAFORMA

Posdata: Integración con la oficina de servicio

La evolución de su mainframe no tiene que parar una vez que logre el estado deseado de agilidad e integración entre plataformas en sus flujos de trabajo DevOps. De hecho, probablemente desee basarse en ese logro para mejorar más la agilidad y la eficiencia digital de su empresa con el paso del tiempo.

Una manera especialmente convincente de hacerlo es dándole a su equipo de gestión de servicios informáticos (ITSM) un entorno unificado tanto para las aplicaciones mainframe como no mainframe. Este modelo ITSM unificado será cada vez más útil a medida que la proposición de valor digital de su empresa se base en aplicaciones multiplataforma, desde sistemas de registro de mainframe internos a aplicaciones móvil y web de trato directo con el cliente.

Topaz Connect proporciona este tipo de integración ITSM entre plataformas unificando soluciones externas ITSM como ServiceNow, BMC Software, Tivoli y CA. A través de esta integración, el personal ITSM puede hacer un seguimiento de los procesos para las aplicaciones mainframe de la misma manera que lo hace con otras plataformas de hardware y software. Cuando se inicia una petición de cambio de flujo de trabajo en su sistema ITSM que requiere un cambio de código, las especificaciones del cambio se comunican automáticamente a ISPW. Y a medida que esas modificaciones se entregan, su entorno ITSM puede hacer un seguimiento del progreso del flujo de trabajo hasta la misma puesta en marcha.

El imperativo de transformación

Los líderes de TI tienen la gran presión de cumplir en varios frentes. Deben proporcionar buenas analíticas a la vez que salvaguardan la seguridad cibernética empresarial. Tienen que cumplir con las expectativas crecientes de movilidad de los clientes y los empleados. Y deben hacer todo esto con desafiantes restricciones de presupuestos para inversiones y mantenimiento.

La transformación del mainframe, sin embargo, es crucial en el éxito de todos estos esfuerzos e incluso más. Si sus sistemas centrales de registro no son ágiles (y si no está completamente preparado para alargar la vida útil de esos sistemas de registro durante la próxima década, incluso si va perdiendo su cohorte de veteranos mainframe actuales porque se jubilan), entonces el resto de sus esfuerzos solo pueden proporcionar beneficios limitados. Al final, el rendimiento de su negocio se verá limitado por las restricciones de su entorno mainframe.

Las buenas noticias son que ahora las mejores prácticas, las herramientas modernas y los socios comprometidos están disponibles para ayudarle en sus esfuerzos. Todo lo que necesita es decidirse y tener un plan. Entonces puede empezar.

El socio de software para mainframe para los próximos 50 años

Compuware, una compañía de BMC, capacita a las empresas más grandes del mundo en su transformación con Autónoma. Para ello ofrece un software innovador que permite a los profesionales de TI con habilidades genera gestionar aplicaciones de mainframe con facilidad y agilidad.

[Obtenga más información en compuware.com.](https://www.compuware.com)