

WHITE PAPER

Actualización de versiones de MATLAB y Simulink para organizaciones de gran tamaño

Judy Wohletz, Vinod Reddy y Jim Ross - MathWorks

Índice de contenido

1	Introducción	4
2	Visión general	5
2.1	Evaluación	5
2.2	Planificación	6
2.3	Migración, pruebas y lanzamiento	7
2.3.1	Migración	7
2.3.2	Pruebas	10
2.3.3	Lanzamiento	10
2.4	Soporte.....	11
3	Flujo de trabajo detallado de la actualización	11
3.1	Evaluación	11
3.1.1	Consideraciones al inicio del proyecto	12
3.1.2	Selección de la versión de actualización de MATLAB y Simulink	12
3.1.3	Preparación del entorno técnico	12
3.1.4	Pruebas iniciales.....	13
3.1.5	Pruebas de regresión.....	14
3.1.6	Decisión de actualizar.....	15
3.2	Planificación	15
3.2.1	Gestión del proyecto de actualización.....	15
3.2.2	Creación de un caso de negocio	17
3.2.3	Definición de objetivos por adelantado.....	17
3.3	Migración	18
3.3.1	Configuración del entorno.....	18
3.3.2	Identificación de modelos que se desean migrar	18
3.3.3	Migración inicial.....	18
3.3.4	Migración de herramientas personalizadas	19
3.3.5	Migración automática.....	22
3.4	Pruebas	24
3.4.1	Pruebas de regresión	24
3.4.2	Pruebas de equivalencia.....	25
3.4.3	Pruebas de integración continua	26
3.4.4	Pruebas beta	26
3.4.5	Pruebas de actualizaciones	26

3.5 Lanzamiento	26
3.5.1 Formación	26
3.5.2 Lanzamiento	27
3.6 Soporte.....	27
3.6.1 Actividades posteriores a la actualización	27
4 Sostenimiento: Una filosofía de actualización continua.....	28
4.1 Pruebas de versiones preliminares.....	28
4.2 Pruebas de modelos de clientes.....	29
4.3 Seminarios, webinars y conferencias	29
5 Soporte de MathWorks	29
6 Anexo	30

1 Introducción

Históricamente, los equipos de desarrollo actualizaban a nuevas versiones de las herramientas antes de comenzar un nuevo proyecto y ya no podían cambiar de versión durante todo el ciclo de desarrollo del producto, que se prolongaba varios años. Hoy en día, las empresas están adoptando nuevas tecnologías a un ritmo sin precedentes, lo que hace difícil actualizar o incluso investigar nuevas tecnologías mientras un producto está en desarrollo. Una estrategia de actualización proactiva ofrece a las organizaciones la oportunidad de encontrar un equilibrio entre el riesgo de cambiar de versión y el de no poder incorporar nuevos métodos y tecnologías.

Actualizar a una nueva versión de MATLAB y Simulink permitirá que su organización disponga de la tecnología más reciente del diseño basado en modelos, lo que conlleva mayor productividad y mejores resultados. El beneficio neto de la actualización puede medirse como rentabilidad sobre la inversión. El cálculo de la rentabilidad incluye aumentos de la productividad, y reducciones del coste y el riesgo. La inversión se mide como el coste de evaluar e implementar una actualización más el coste de los cambios en procesos y herramientas necesarios para que la organización pueda hacer un uso efectivo de las nuevas prestaciones de la actualización.

El éxito de toda actualización depende de la estrategia utilizada para implementarla. En este white paper, MathWorks ofrece recomendaciones para aplicar un enfoque sistemático a las actualizaciones que afecten a toda la empresa, para aprovechar los avances de los productos de MATLAB y Simulink lo más posible, y disminuir los costes, riesgos y perturbaciones que implican los beneficios que se obtienen. Estas recomendaciones se basan en la experiencia práctica adquirida orientando a clientes corporativos en actualizaciones a nuevas versiones de MATLAB y Simulink.

El proceso de actualización consta de seis fases principales:

1. Evaluación
2. Planificación
3. Migración
4. Pruebas
5. Lanzamiento
6. Soporte

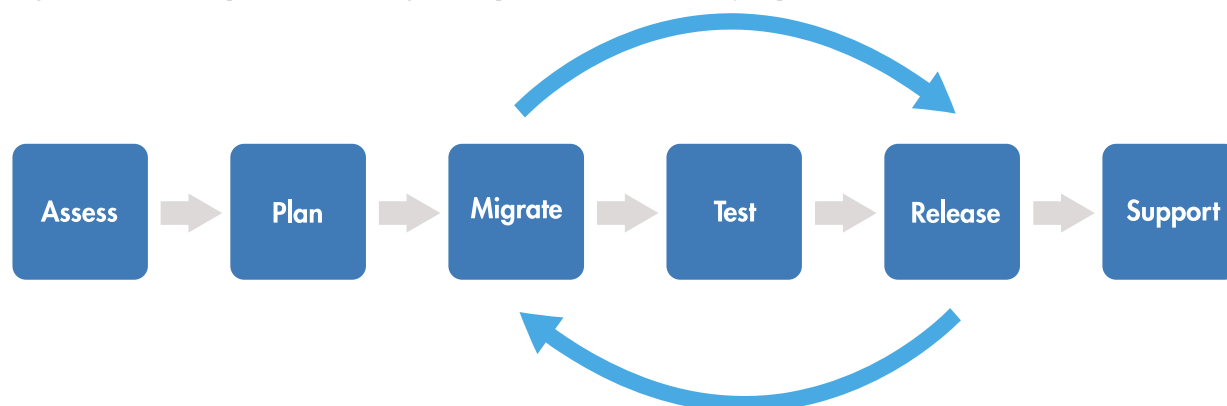
El aspecto más importante de cada fase del proceso es la comunicación. Es fundamental que todos los grupos de interés estén perfectamente informados durante la actualización de una versión. Notificar anticipadamente de la evaluación, el resultado y los principales motivos que justifican la decisión de actualizar puede ayudar a las partes interesadas a prepararse adecuadamente. Una vez que se toma la decisión de actualizar, se debe comunicar rápidamente el plan organizativo para permitir la planificación local. Es importante comunicar los plazos, los éxitos y las dificultades a lo largo del camino, con el fin de garantizar el éxito de los equipos de trabajo cuando llegue el momento de actualizar.

Una última consideración para que el proceso de actualización sea eficiente y sólido es el uso de la integración continua. Utilizar un entorno de integración continua en los flujos de trabajo de desarrollo mejora la repetibilidad y reduce el esfuerzo que conlleva realizar pruebas. Además, emplear un entorno de integración continua compensa durante el trabajo diario, y reduce notablemente el esfuerzo necesario para evaluar las actualizaciones. Si la integración continua ya forma parte de los flujos de trabajo de desarrollo, bastarán cambios menores para disponer de flexibilidad para probar versiones de MATLAB específicas. Si la integración continua aún no forma parte de los flujos de trabajo de desarrollo actuales, sería buena idea incorporarla.

Estas actividades se combinan para minimizar el riesgo y maximizar la probabilidad del éxito de la actualización.

2 Visión general

Para optimizar la rentabilidad sobre la inversión, antes de actualizar es imprescindible comprender el proceso de actualización, las posibles rutas de actualización y, sobre todo, los beneficios potenciales y los costes relacionados. Las tareas o actividades necesarias para actualizar a una nueva versión pueden agruparse en fases lógicas: evaluación, planificación, migración, pruebas, lanzamiento y soporte.



Flujo de trabajo de las fases de un proyecto de actualización típico.

Cada fase del proceso tiene desencadenantes, información de partida, actividades, resultados y criterios de salida. El proceso en sí cuenta con el soporte de una estrategia de sostenimiento. Este flujo de trabajo debe estar respaldado en todo momento por un plan de comunicación específico que mantenga informadas a todas las partes interesadas sobre el propósito, el estado, las decisiones y los plazos, así como las cuestiones importantes que se identifiquen durante la actualización.

2.1 Evaluación

En la primera fase del proceso de actualización, el objetivo es comprender el efecto general de la actualización, y evaluar si los beneficios superan el coste, el riesgo y el esfuerzo. El propósito es determinar si existe algún problema grave en las primeras etapas de desarrollo y así evitar dedicar un esfuerzo adicional a evaluar esa versión concreta. Se necesita suficiente rentabilidad sobre la inversión antes de proceder a la siguiente fase del proceso de actualización.

Evaluación	
Objetivo: Comprender el impacto de una actualización	
Desencadenantes	<ol style="list-style-type: none"> 1. Obsolescencia del sistema operativo 2. Nuevas funcionalidades de MATLAB y Simulink, como prestaciones y productos completamente nuevos 3. Nueva plataforma de desarrollo de hardware 4. Colaboración entre OEM, proveedores y partners 5. Nuevo proyecto o programa
Información de partida	<ol style="list-style-type: none"> 1. Estado actual del proyecto que se va a actualizar 2. Cantidad de versiones de MATLAB y Simulink entre la versión actual y la actualizada 3. Versión actual de MATLAB y Simulink, software y hardware de terceros, software y hardware de aplicación o desarrollo, y plataforma de desarrollo, tales como equipos y sistemas operativos 4. Cronología de la actualización

Actividades	<ol style="list-style-type: none"> 1. Seleccionar la versión de destino inicial 2. Crear una lista de requisitos y problemas que la nueva versión podrá solucionar 3. Identificar un equipo de trabajo para migrar una muestra pequeña y representativa de modelos, códigos y scripts con Upgrade Advisor 4. Migrar una muestra pequeña y representativa de modelos, códigos y scripts para comprender mejor los efectos de la migración 5. Ejecutar pruebas de regresión en las tareas más frecuentes para recopilar métricas sobre cuánto tardarán esas tareas en la nueva versión 6. Evaluar si este equipo de trabajo tiene la capacidad de ejecutar correctamente el proyecto de actualización o si necesita formación técnica adicional 7. Evaluar si el equipo de trabajo cuenta con los recursos necesarios para gestionar el proyecto de actualización 8. Preparar un informe de evaluación de la actualización que incluya: <ul style="list-style-type: none"> • Procedimientos de actualización • Resultados con beneficios y métricas, tales como velocidad de simulación, tamaño del código, y más • Desafíos • Mensajes de error • Estimación de la rentabilidad sobre la inversión en función de la información actual sobre beneficios potenciales, costes estimados y riesgos
Resultados	<ol style="list-style-type: none"> 1. Decisión de proceder o no proceder 2. Versión de destino inicial 3. Informe de evaluación de la actualización

2.2 Planificación

El propósito de la fase de planificación es definir el alcance general y planificar el proyecto de actualización en su conjunto.

Planificación	
Objetivo: Definir el alcance general del proyecto de actualización	
Desencadenantes	Decisión de proceder
Información de partida	Informe de evaluación de la actualización
Actividades	<ol style="list-style-type: none"> 1. Crear un caso de negocio 2. Establecer el alcance identificando qué modelos, proyectos, organizaciones y grupos de interés clave se ven afectados
Resultados	<ol style="list-style-type: none"> 1. Caso de negocio para la actualización 2. Plan de actualización que contenga plazos de actualización; lista de proyectos, modelos, organizaciones y grupos de interés afectados; estimaciones del esfuerzo, formación y recursos necesarios, costes y riesgos, dependencias, y versiones de destino de hardware y software

2.3 Migración, pruebas y lanzamiento

Las fases de migración, pruebas y lanzamiento están estrechamente relacionadas y se realizan de manera iterativa para actualizar los modelos a la nueva versión.

2.3.1 Migración

En la fase de migración, se actualizan los modelos de Simulink, el código de MATLAB, las apps de MATLAB y las herramientas personalizadas de manera iterativa para que funcionen con la nueva versión. En este contexto, una herramienta personalizada es toda aplicación o configuración que se ha creado en un entorno con MATLAB y Simulink. He aquí algunos ejemplos de herramientas personalizadas:

- Librerías de bloques de Simulink personalizados
- Bloques de función S
- Scripts de inicio de MATLAB
- Clases de almacenamiento personalizadas en Embedded Coder Dictionary
- Archivos de destino del sistema personalizados
- Archivos de configuración de MATLAB Report Generator o Simulink Report Generator
- Comprobaciones personalizadas de Model Advisor
- Pautas de estilo para modelado
- Parámetros de configuración predeterminados
- Scripts de MATLAB para personalizar el entorno de MATLAB y Simulink

La fase de migración se ejecuta en varias iteraciones de tres subfases: migración inicial, migración de herramientas personalizadas y migración automática. Por lo general, estas iteraciones requieren una combinación de enfoques manuales y automatizados para actualizar modelos, scripts, interfaces de usuario, plantillas, interfaces a herramientas de terceros y otras aplicaciones.

Las actividades de esta fase culminan en la migración de modelos a una nueva versión. Sin embargo, deberá probar los modelos durante la fase de pruebas, que se describe más adelante.

2.3.1.1 Migración inicial

En la subfase de migración inicial, se debe migrar un pequeño conjunto de modelos y proyectos de Simulink que sean representativos de los modelos desarrollados en la organización, con el objetivo de desarrollar procedimientos y herramientas, y comprender mejor las nuevas funcionalidades. Para ello, se utiliza [Upgrade Advisor](#), una herramienta diseñada para ayudar a actualizar y mejorar los modelos con la versión actual. Esta subfase sirve como base de la siguiente subfase, y contribuye a reducir el riesgo y el esfuerzo.

Migración inicial

Objetivo: Migración de un subconjunto de modelos

Desencadenantes	Aprobación del caso de negocio
Información de partida	<ol style="list-style-type: none">1. Informe de evaluación de la actualización2. Plan de actualización3. Modelos y otros artefactos relacionados
Actividades	<ol style="list-style-type: none">1. Ejecutar Upgrade Advisor en los modelos2. Realizar correctamente la actualización del diagrama, la simulación y la generación de código para los modelos

	<ol style="list-style-type: none"> 3. Documentar los mensajes de error o advertencia con Diagnostic Viewer en Simulink 4. Solucionar los posibles errores 5. Comparar los mensajes de advertencia para determinar si son exclusivos de la nueva versión o también aparecen en la versión actual 6. Determinar la gravedad de los mensajes de advertencia 7. Decidir si solucionar los mensajes de advertencia 8. Documentar cualquier otro problema en una lista de problemas de la actualización
Resultados	<ol style="list-style-type: none"> 1. Informe actualizado de evaluación de la actualización 2. Nueva lista de problemas de la actualización 3. Plan actualizado de la tarea de actualización 4. Subconjunto de modelos migrados a la nueva versión

2.3.1.2 Migración de herramientas personalizadas

Además de migrar las herramientas personalizadas, también es importante eliminar las funcionalidades anteriores de herramientas personalizadas y adoptar las funcionalidades integradas durante la subfase de migración de herramientas personalizadas. El objetivo a largo plazo es eliminar las herramientas personalizadas y adoptar funcionalidades integradas. Eso permitirá que las futuras actualizaciones sean más sencillas y automáticas. Asimismo, reducirá considerablemente el esfuerzo y el coste de futuras actualizaciones.

Migración de herramientas personalizadas	
Objetivo: Actualizar modelos y herramientas personalizadas y desarrollar herramientas de automatización	
Desencadenantes	<ol style="list-style-type: none"> 1. Plan de actualización aprobado 2. Subconjunto de modelos migrados
Información de partida	<ol style="list-style-type: none"> 1. Informe de evaluación de la actualización 2. Lista de problemas de la actualización 3. Modelos y otros artefactos relacionados 4. Herramientas personalizadas
Actividades	<ol style="list-style-type: none"> 1. Actualizar las herramientas personalizadas a la nueva versión 2. Reemplazar las herramientas personalizadas con funcionalidad integrada de MATLAB y Simulink siempre que sea posible 3. Validar las herramientas personalizadas en la nueva versión 4. Desarrollar herramientas de automatización de los pasos manuales del proceso de actualización para reducir el volumen de trabajo y garantizar la uniformidad
Resultados	<ol style="list-style-type: none"> 1. Informe actualizado de evaluación de la actualización 2. Lista actualizada de problemas de la actualización 3. Plan actualizado de la tarea de actualización 4. Herramientas personalizadas actualizadas 5. Herramientas de automatización

2.3.1.3 Migración automática

En la subfase de migración automática, se debe actualizar el resto de los modelos empleando el procedimiento y las herramientas de automatización que se han desarrollado en la subfase anterior. Dos objetivos principales de esta fase son solucionar los problemas de migración y mejorar las herramientas de automatización, para que los demás equipos de trabajo puedan utilizarlas.

Migración automática	
Objetivo: Solucionar problemas conocidos en modelos, herramientas personalizadas y de automatización. Actualizar modelos, herramientas personalizadas y de automatización	
Desencadenantes	<ol style="list-style-type: none"> 1. Actualización de modelos y herramientas personalizadas 2. Desarrollo de herramientas de automatización
Información de partida	<ol style="list-style-type: none"> 1. Herramientas de automatización 2. Herramientas personalizadas actualizadas 3. El resto de los modelos y artefactos relacionados 4. Lista de problemas de la actualización
Actividades	<ol style="list-style-type: none"> 1. Ejecutar las herramientas de automatización junto con Upgrade Advisor en el resto de los modelos. Las herramientas de automatización deben invocar a Upgrade Advisor. 2. Realizar correctamente la actualización del diagrama, la simulación y la generación de código para los modelos 3. Documentar los mensajes de error o advertencia con Diagnostic Viewer en Simulink 4. Solucionar los posibles errores 5. Solucionar las advertencias que dificulten la migración 6. Documentar los problemas en la lista de problemas de la actualización 7. Actualizar la documentación de ayuda de las herramientas personalizadas y de automatización 8. Redactar notas de la versión para las herramientas personalizadas y de automatización 9. Solucionar los problemas mejorando las herramientas personalizadas y de automatización 10. Crear materiales de formación o actualizarlos, e incluir las diferencias relevantes entre la versión antigua y la nueva, las herramientas personalizadas y de automatización
Resultados	<ol style="list-style-type: none"> 1. Mejoras en las herramientas de automatización 2. Informe actualizado de evaluación de la actualización 3. Lista actualizada de problemas de la actualización 4. Plan actualizado de la tarea de actualización 5. Modelos actualizados a la nueva versión 6. Notas de la versión para las herramientas personalizadas y de automatización 7. Documentación de ayuda actualizada de las herramientas personalizadas y de automatización 8. Materiales de formación nuevos o actualizados

2.3.2 Pruebas

El objetivo de la fase de pruebas es garantizar que los resultados producidos por el modelo y el código utilizando la nueva versión sean funcional y numéricamente equivalentes a los producidos con la versión anterior, dentro de límites aceptables.

Pruebas	
Objetivo: Validar los modelos actualizados	
Desencadenantes	Actualización de modelos
Información de partida	<ol style="list-style-type: none">1. Herramientas personalizadas actualizadas2. Modelos y otros artefactos relacionados3. Datos de prueba de entrada y resultados esperados
Actividades	<ol style="list-style-type: none">1. Ejecutar pruebas de regresión en las tareas más frecuentes y comparar los resultados con los de la versión anterior2. Realizar simulaciones de lazo abierto, model-in-the-loop (MIL), software-in-the-loop (SIL) y processor-in-the-loop (PIL) en el equipo de escritorio3. Realizar simulación de hardware-in-the-loop (HIL), prototipado rápido y prototipado rápido en la plataforma de destino en laboratorio4. Realizar las pruebas de sistema o en hardware que se suelen efectuar5. Revisar los resultados y aceptar el modelo validado6. Definir los datos de prueba adicionales y los resultados esperados según sea necesario
Resultados	<ol style="list-style-type: none">1. Modelos validados en la nueva versión2. Datos de prueba adicionales y resultados esperados, si es necesario3. Informes de pruebas y artefactos relacionados

2.3.3 Lanzamiento

En la fase de lanzamiento, se publican los modelos actualizados, las herramientas personalizadas y las nuevas versiones de MATLAB y Simulink.

Lanzamiento	
Objetivo: Lanzar la nueva versión de MATLAB y las herramientas personalizadas	
Desencadenantes	Fase de pruebas completa
Información de partida	<ol style="list-style-type: none">1. Modelos validados2. Datos de prueba de entrada y resultados esperados3. Documentación de ayuda de las herramientas personalizadas4. Notas de la versión de las herramientas personalizadas5. Plan de actualización
Actividades	<ol style="list-style-type: none">1. Actualizar el plan de actualización2. Programar e impartir clases de formación3. Crear un repositorio para todos los materiales4. Notificar a usuarios

Resultados	<ol style="list-style-type: none"> 1. Plan actualizado de la tarea de actualización 2. Documentos, herramientas y formación de la versión lanzada
-------------------	---

2.4 Soporte

En la fase de soporte, la organización presta soporte continuo a usuarios de modelos, herramientas personalizadas y productos de MATLAB y Simulink a medida que surgen problemas.

Soporte	
Objetivo: Solucionar los problemas detectados	
Desencadenantes	Problema detectado
Información de partida	<ol style="list-style-type: none"> 1. Herramientas personalizadas 2. Modelo con problemas 3. Documentación de ayuda 4. Material de formación
Actividades	<ol style="list-style-type: none"> 1. Reproducir el problema 2. Identificar la causa raíz 3. Corregir el problema (del modelo, las herramientas personalizadas o lo que corresponda) 4. Notificar a los usuarios de las nuevas versiones 5. Actualizar el repositorio 6. Actualizar la lista de seguimiento de problemas 7. Actualizar las notas de la versión de las herramientas personalizadas si es necesario 8. Realizar el lanzamiento de la versión actualizada con la corrección si es necesario
Resultados	<ol style="list-style-type: none"> 1. Notas de la versión actualizadas 2. Versión corregida del modelo o las herramientas personalizadas 3. Lista de seguimiento de problemas actualizada

3 Flujo de trabajo detallado de la actualización

En este capítulo, se analizan más detalladamente las seis fases del proceso definidas en el capítulo 2.

3.1 Evaluación

En la fase de actualización, el objetivo es comprender el efecto general de la actualización y determinar si sus beneficios superan el coste, el riesgo y el esfuerzo. El propósito es determinar si existe algún problema grave en las primeras etapas de desarrollo y así evitar dedicar un esfuerzo adicional a evaluar esa versión concreta. Si la rentabilidad sobre la inversión de la actualización no es suficiente, es posible que la organización decida retrasar la actualización hasta la próxima versión. Cada vez que se retrasa una actualización, aumenta la dificultad para actualizar a futuras versiones. Son muchos los eventos que pueden desencadenar un proceso de actualización; por ejemplo, nuevos proyectos, colaboración con nuevas empresas, nuevo hardware o nuevos sistemas operativos.

3.1.1 Consideraciones al inicio del proyecto

El primer paso del proceso de actualización es evaluar la situación actual. La evaluación incluye todos o la mayoría de los siguientes pasos:

- Crear un inventario del software y el hardware utilizados actualmente en la empresa
- Documentar las versiones de MATLAB y Simulink utilizadas actualmente en la empresa y en las empresas con las que colabora
- Documentar las herramientas de software y hardware de terceros utilizadas actualmente con productos de MATLAB y Simulink
- Identificar con qué versión de los productos de MATLAB y Simulink los proveedores externos planean trabajar
- Identificar con qué equipos y sistemas operativos la empresa trabaja actualmente y planea trabajar en el futuro
- Colaborar con las partes interesadas clave que se vayan a ver afectadas por esta actualización y acordar un plazo de ejecución estimado
- Revisar problemas de actualizaciones anteriores

3.1.2 Selección de la versión de actualización de MATLAB y Simulink

Antes de decidir a qué versión de MATLAB y Simulink se debe actualizar, filtre y busque las nuevas funcionalidades de versiones específicas en las *notas de la versión*, los problemas conocidos en los *informes de fallos*, los *requisitos del sistema* y los *compiladores soportados*. Consulte también la *disponibilidad de plataformas por productos* y *cómo elegir un equipo para ejecutar los productos de MATLAB y Simulink*, si es relevante para la situación de su organización. Otra cosa que debe tener en cuenta al seleccionar la versión de MATLAB y Simulink es la actualización. Puede acceder a las actualizaciones haciendo clic en el ícono de campana en la esquina superior derecha de la ventana de MATLAB. MathWorks recomienda aplicar la última actualización en cuanto esté disponible y sea conveniente para su organización o programas. Si utiliza software de terceros, comuníquese con los proveedores para verificar con qué versiones y actualizaciones de MATLAB y Simulink trabajan actualmente y planean trabajar en el futuro. También es importante conocer qué plazos siguen tras el lanzamiento de nuevas versiones y actualizaciones de MATLAB y Simulink, especialmente si pretende actualizar a una versión de MATLAB y Simulink que todavía no se ha publicado.

Tenga en cuenta que quizá desee cambiar la versión de actualización de MATLAB y Simulink durante el proceso de actualización. Es posible que se encuentre con problemas inesperados, o que MathWorks lance nuevas funcionalidades que podrían beneficiar a su organización. En lugar de fijar una versión específica tan pronto en el proceso, **se recomienda ser flexible y seleccionar la versión que genere la mayor rentabilidad sobre la inversión de acuerdo con sus circunstancias concretas**. Si la frecuencia con la que actualiza es superior a un año, se recomienda aprovechar las nuevas funcionalidades cambiando la versión de actualización, en lugar de esperar años para utilizarlas.

3.1.3 Preparación del entorno técnico

El siguiente paso es asegurarse de que tiene acceso a los equipos, los sistemas operativos, la versión de MATLAB y Simulink, el software y hardware de desarrollo, y el software y hardware de terceros que su empresa utiliza para los proyectos afectados por la actualización. Disponga un entorno aislado para las tareas de evaluación y migración, ya que es difícil abrir en la versión original un modelo que se guardó en la versión nueva. Por último, con las nuevas versiones, debe tener en cuenta configuraciones adicionales para garantizar resultados adecuados.

Un entorno aislado implica una o varias ramas nuevas dedicadas en un repositorio que permitan:

- La colaboración entre múltiples usuarios o equipos de trabajo
- El uso compartido de actualizaciones de herramientas o librerías
- El uso sencillo de flujos de trabajo de integración continua existentes
- La fácil reutilización de flujos de trabajo de integración continua nuevos o mejorados que se hayan desarrollado como parte de la actualización

Es importante poder escalar el entorno aislado para utilizarlo en las pruebas iniciales dentro de la fase de evaluación con la posibilidad de aplicar el mismo enfoque a mayor escala posteriormente en las fases de migración, pruebas y lanzamiento. En la fase de evaluación, las pruebas deberían ser mínimas y el énfasis en detectar problemas que no se puedan resolver con facilidad, comprender las advertencias o identificar problemas de rendimiento. Se recomienda realizar pruebas más completas en las fases de migración y pruebas.

3.1.4 Pruebas iniciales

Una vez que se haya configurado el entorno técnico y se hayan establecido los criterios de actualización, seleccione el modelo que desea actualizar. Es buena idea seleccionar un modelo de gran tamaño que contenga patrones de modelado frecuentes, cumpla con el estilo de modelado y represente la mayoría de los modelos que se utilizan en la organización.

Una vez que haya seleccionado un modelo, siga los pasos a continuación para completar una prueba inicial. Aunque este proceso suele requerir varios pasos manuales, es fundamental para identificar problemas con la actualización de MATLAB y Simulink. Pruebe modelos, y librerías y herramientas personalizadas tal cual estén, haciendo solo las modificaciones imprescindibles para eliminar los mensajes de error que puedan impedir seguir realizando pruebas. No olvide que el propósito de la fase de evaluación es obtener información sobre la viabilidad de la actualización lo antes posible y, al mismo tiempo, identificar problemas que deban solucionarse para poder realizar una migración correcta a mayor escala.

Las actividades recomendadas para la fase de evaluación son:

1. Prueba de la versión actual
 - a. Abra la versión actual de MATLAB y Simulink
 - b. Agregue librerías o herramientas personalizadas a la ruta de MATLAB
 - c. Abra el modelo o los proyectos de Simulink en la versión actual de MATLAB y Simulink. Acepte los parámetros de configuración recomendados del modelo de Simulink.
 - d. Actualice el diagrama
 - e. Simule el modelo
 - f. Solucione todos los mensajes de error y, si corresponde, determine la acción adecuada en caso de advertencias
2. Prueba de la versión de destino
 - a. Abra la versión de destino de MATLAB y Simulink
 - b. Agregue librerías o herramientas personalizadas a la ruta de MATLAB
 - c. Abra el modelo o proyecto de Simulink en la versión de destino de MATLAB y Simulink
 - d. Ejecute Upgrade Advisor en el modelo
 - e. Registre los mensajes de advertencia y error
 - i. Corrija los mensajes de error
 - ii. Corrija los mensajes de advertencia solo si dificultan la realización de pruebas

iii. Documente y realice el seguimiento de todos los problemas que surjan (es probable que también surjan con otros modelos)

f. Solucione los problemas antes de lanzar la versión actualizada de MATLAB y Simulink

Si surge un problema que no puede solucionar mientras actualiza el modelo seleccionado a la versión de destino de MATLAB y Simulink, pruebe el modelo en una versión intermedia. Si está actualizando desde una versión antigua, es posible que deba actualizar a una versión intermedia de MATLAB y Simulink. Es posible que este paso extra sea necesario para asegurarse de que se identifican las advertencias y los mensajes de error sobre incompatibilidades. Si el tiempo que transcurre entre las actualizaciones es mayor que el plazo de eliminación gradual de los mensajes de error y advertencia, es posible que se pasen por alto mensajes importantes que ayudan a depurar la causa raíz de un problema. Este enfoque puede ayudar a depurar o aislar un problema, pero por lo general, la versión intermedia solo es una transición a la nueva versión de destino. Se recomienda seleccionar una versión de MATLAB y Simulink intermedia entre las versiones actual y de destino.

3.1.5 Pruebas de regresión

Ejecute pruebas de regresión en las tareas más frecuentes para poder estimar cuánto tardarán en la nueva versión de MATLAB y Simulink. Siempre que sea posible, ejecute las mismas tareas en la versión anterior y compare los resultados de las pruebas. Se recomienda ejecutar estas pruebas en modelos de gran tamaño que se estén utilizando actualmente. Esta actividad servirá para descubrir posibles problemas antes de lanzar la nueva versión.

En la fase de evaluación, las pruebas de regresión son pruebas rápidas para comprobar si la versión específica de MATLAB y Simulink es una candidata viable para la actualización. Lo ideal es realizar estas pruebas de regresión rápidamente y con todas las versiones preliminares, independientemente de la versión a la que se desee actualizar. En la siguiente tabla se proporciona una lista de elementos clave que se deben comprobar para evaluar una versión específica de MATLAB y Simulink en la fase de evaluación. Es una lista parcial y representativa de las características importantes. En la fase de evaluación, el aspecto más importante que se debe evaluar es el rendimiento y si existen problemas graves que no se puedan solucionar. En la fase de pruebas se proporciona más información sobre las pruebas de regresión.

Resultado	Ideal	Notas
Simulación	Simulación sin errores	El modelo se simula con errores, pero los errores se pueden solucionar.
Generación de código	Generación de código sin errores	El modelo genera código con errores, pero los errores se pueden solucionar.
Duración de la simulación	Menor que en la versión anterior	Un ligero aumento de la duración de la simulación puede ser aceptable, pero la duración más breve debe mantenerse como estándar.
Duración de la generación de código	Menor que en la versión anterior	Un ligero aumento de la duración de la generación de código puede ser aceptable, pero la duración más breve debe mantenerse como estándar.

3.1.6 Decisión de actualizar

Una vez que se han completado las pruebas de regresión y evaluado los resultados con los criterios de actualización, se puede decidir pasar a la siguiente fase del proceso o posponer la actualización.

3.2 Planificación

El propósito de la fase de planificación es definir el alcance general y planificar el proyecto de actualización. Debe abarcar el resto de las fases del proceso: migración, pruebas, lanzamiento y soporte.

3.2.1 Gestión del proyecto de actualización

Desarrolle un plan de actualización general que incluya la fecha objetivo y la versión de actualización de MATLAB y Simulink. El plan debe definir con claridad las fases y actividades necesarias para cumplir con los criterios de salida de cada fase. También debe definir las funciones y expectativas de cada integrante del equipo, así como los diferentes equipos de trabajo involucrados en el proceso de actualización. Típicamente, incluye: principal de actualización de software, equipo de ingeniería, grupo responsable del proyecto, equipo de TI, gerente del programa, equipo de herramientas de software, gerente de proyecto, proveedores externos y personal de consultoría de MathWorks. Consulte la siguiente tabla para obtener más detalles.

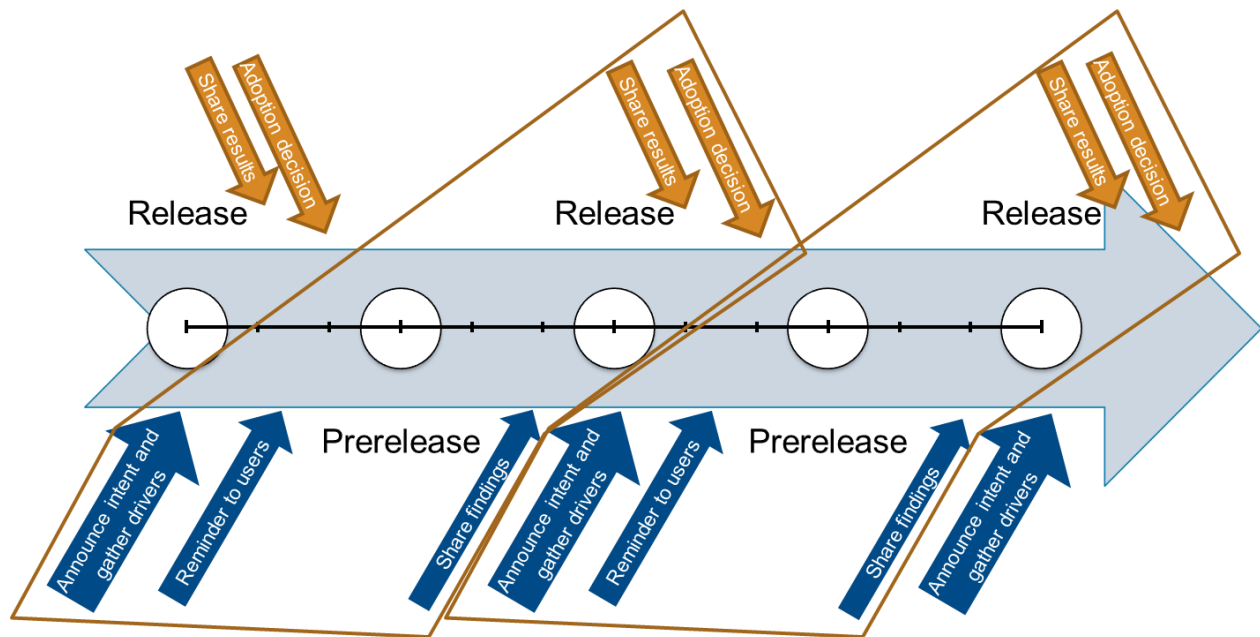
Fase	Principal de actualización	Equipo de ingeniería	Grupo responsable del proyecto	Equipo de TI	Gerente del programa	Equipo de herramientas
Evaluación	Liderar	Informar	Informar	Respaldar	Informar	Contribuir
Planificación	Liderar	Contribuir	Contribuir	Contribuir	Contribuir	Contribuir
Migración	Liderar	Contribuir	Informar	Respaldar	Informar	Contribuir
Pruebas	Liderar	Contribuir	Informar	Informar	Informar	Contribuir
Lanzamiento	Contribuir	Liderar	Informar	Informar	Informar	Liderar
Soporte	Liderar	Contribuir	Informar	Informar	Informar	Contribuir

La comunicación es fundamental durante todo el proceso de actualización: los planes, los estados, las decisiones y los siguientes pasos se deben compartir periódicamente con todas las partes interesadas. Es importante comunicar una posible actualización lo antes posible. Notifique a la comunidad de usuarios las nuevas funcionalidades que justificarían una actualización, e incluya un plazo para las fases de evaluación y migración. Comparta los resultados de cada fase, incluyendo lo positivo y lo negativo, identificados por los equipos de herramientas o los usuarios. Mantenga a todos informados sobre los estados, las decisiones y los cambios en los plazos de entrega. Tenga presente que la comunicación debe ser una tarea continua durante una actualización. Por último, para lograr un mayor impacto, tenga en cuenta las necesidades de cada parte interesada y adapte el estilo de comunicación como corresponda.

Programe reuniones sobre hitos y estados para poner al día a las partes interesadas periódicamente y proporcionar informes de estado sobre la actualización. En estas reuniones, revise el estado con cada integrante del equipo, y comparta todos los problemas que surjan en el proceso de actualización. Periódicamente, realice revisiones técnicas de las actividades completadas en cada fase. La tabla y la figura siguientes son ejemplos de la cronología de comunicaciones que muestra el cronograma de lanzamiento de la versión preliminar y final de MATLAB y Simulink.

Cronología de comunicaciones

Fecha	Destinatarios	Temas
3 meses antes de lanzar la versión preliminar de MATLAB y Simulink	Comunidad usuaria y gerencia	Recordatorio de pruebas de la versión preliminar Propósito de la actualización, incluyendo: <ul style="list-style-type: none"> • Lista de motivos conocidos para la actualización • Lista de obstáculos conocidos para la actualización Solicitar voluntarios para probar la versión preliminar
1 mes antes de lanzar la versión preliminar de MATLAB y Simulink	Comunidad usuaria	Recordatorio de pruebas de la versión preliminar Cronograma de pruebas de la versión preliminar Cambios en los motivos u obstáculos conocidos
Lanzamiento de la versión preliminar de MATLAB y Simulink	Participantes activos en las pruebas de la versión preliminar	Compartir el progreso y el cronograma de pruebas de la versión preliminar
1 mes después del lanzamiento de la versión preliminar de MATLAB y Simulink	Comunidad usuaria	Compartir los resultados positivos y negativos de las pruebas de la versión preliminar Compartir el resto de los planes de pruebas de la versión preliminar, si los hay
1 mes antes del lanzamiento de la versión final de MATLAB y Simulink	Comunidad usuaria y gerencia	Compartir los resultados de las pruebas de la versión preliminar Compartir el plan de evaluación de la actualización
Lanzamiento de la versión final de MATLAB y Simulink	Participantes activos en la evaluación de la actualización	Compartir el progreso y el cronograma de la evaluación de la actualización
1 mes después del lanzamiento de la versión final de MATLAB y Simulink	Comunidad usuaria	Compartir los resultados positivos y negativos de la evaluación de la actualización Compartir el resto de los planes de evaluación de la actualización, si los hay
2 meses después del lanzamiento de la versión final de MATLAB y Simulink	Comunidad usuaria y gerencia	Compartir la decisión de actualizar cuando se tome, incluyendo planes y cronograma



Cronología de comunicaciones sobre lanzamientos de MathWorks

3.2.2 Creación de un caso de negocio

Crear un caso de negocio antes de actualizar a una nueva versión permite aclarar los beneficios y los costes que conlleva. Los beneficios pueden incluir mejoras en los flujos de trabajo posibilitadas por las nuevas funcionalidades de MATLAB y Simulink. Asimismo, las actualizaciones de sistemas operativos o software de terceros pueden influir en la necesidad de actualizar, cuando la versión actual de MATLAB y Simulink no es compatible con el nuevo sistema operativo o software de terceros.

3.2.3 Definición de objetivos por adelantado

Es importante determinar el alcance del proceso de actualización. Tratar de hacer demasiadas cosas al mismo tiempo aumenta el riesgo de retrasos o incluso de que el proyecto de actualización fracase.

Siempre que sea posible, actualice los modelos tal cual estén, sin incorporar funcionalidades nuevas. Incorporar funcionalidades nuevas mientras se introducen nuevas herramientas personalizadas y una versión nueva de MATLAB y Simulink complica el proceso de actualización y dificulta la validación de modelos. Espere hasta que el modelo se haya actualizado y validado por completo en la versión nueva de MATLAB y Simulink antes de incorporar nuevas funcionalidades.

Durante el proceso de actualización, concentre las pruebas de actualización en los flujos de trabajo típicos de su organización; por ejemplo, la actualización, la simulación y la generación de código a partir de modelos. Después de actualizar los modelos y las herramientas personalizadas, el equipo de ingeniería responsable de su desarrollo deberá validar los modelos en la nueva versión.

Es buena idea incluir como objetivo el reemplazo de las funcionalidades de herramientas personalizadas por las funcionalidades integradas de Simulink cuando sea posible. Por ejemplo, puede planificar reemplazar bloques de librerías personalizadas por nuevos bloques de Simulink que ofrezcan la misma funcionalidad. Establezca como objetivo eliminar las pautas de estilo de modelado que no se apliquen a la nueva versión y agregar nuevas pautas para nuevas funcionalidades de Simulink que tenga previsto utilizar. Considere como objetivo evaluar la documentación de las herramientas personalizadas y el material de formación.

3.3 Migración

En la fase de migración, actualice los modelos y las herramientas personalizadas a la nueva versión, de forma iterativa. En cada iteración, por lo general, se necesitan pasos automáticos y manuales para actualizar modelos, scripts, interfaces de usuario, plantillas, interfaces para herramientas de terceros y otras aplicaciones. El enfoque iterativo ofrece una base sólida en cada fase del proceso, genera confianza en la versión nueva, y reduce el riesgo y el esfuerzo para toda la organización y los usuarios finales.

En la fase de migración, los modelos se actualizan a la nueva versión, pero no se prueban. Los modelos se prueban durante la fase de prueba.

3.3.1 Configuración del entorno

Antes de migrar los modelos a la nueva versión, deberá configurar un entorno de modelado. En función del entorno, este paso puede incluir acceso a herramientas personalizadas, configuración de la ruta de MATLAB, configuración del compilador e inicio de un proyecto. Es importante realizar la cantidad mínima de cambios para configurar el entorno en la nueva versión de MATLAB y Simulink. Posponga los cambios en herramientas personalizadas y scripts de MATLAB hasta que comience la fase de migración de herramientas personalizadas. Escale el entorno aislado utilizado en la fase de evaluación para emplearlo en varios proyectos.

Suele ser beneficioso crear parámetros de configuración estándar al comienzo de la fase de migración. Puede servir para disminuir la duplicación de esfuerzos y reducir problemas de integración posteriores. La configuración recomendada para las pruebas iniciales es un buen punto de partida, pero revise la configuración nueva o modificada de cada destino de código para asegurarse de que sea adecuada. Aplique esta configuración revisada a los modelos, al comienzo del proceso de migración.

3.3.2 Identificación de modelos que se desean migrar

En la fase de evaluación, se seleccionó, actualizó y probó un modelo en la nueva versión para evaluar el proceso e identificar posibles problemas. En la fase de migración, las pruebas se extienden a más modelos. Si no puede probar todos los modelos disponibles, se recomienda seleccionar modelos de diferentes tamaños y estilos de modelado de varios equipos de trabajo. Para esta fase del proceso, seleccione casos límite y patrones de modelado poco frecuentes. Si conoce personas o grupos concretos que crean modelos fuera de la norma, inclúyalos en esta fase de pruebas. Probar casos límite ayuda a identificar problemas no previstos y solucionarlos mientras el proceso de actualización está en curso, en lugar de después de lanzar la nueva versión de MATLAB y Simulink. De este modo, se dispone de más tiempo para solucionar problemas sin afectar los plazos de producción. Los siguientes casos avanzados típicos son de modelos con:

- Librerías anidadas
- Bloques de referencia de modelos
- Subsistemas o variables configurables
- Bloques de referencia de modelos y librerías anidadas

3.3.3 Migración inicial

El propósito de la fase de migración inicial es probar un flujo de trabajo típico; por ejemplo, actualización de un diagrama, ejecución de una simulación y generación de código. Utilice la siguiente lista de actividades como guía para completar esta fase.

En la versión anterior:

- Abra los proyectos
- Realice correctamente la actualización del diagrama, la simulación y la generación de código para los modelos
- Documente todos los mensajes de advertencia utilizando Diagnostic Viewer en Simulink

En la nueva versión:

- Configure la ruta y el entorno en MATLAB utilizando proyectos
- Ejecute Upgrade Advisor en los modelos seleccionados:
 - Revise el informe
 - Solo aplique las correcciones necesarias. Las que no dificulten la migración de los modelos seleccionados pueden aplicarse posteriormente durante el proceso.
 - Realice correctamente la actualización del diagrama, la simulación y la generación de código para los modelos
 - Documente los mensajes de error y advertencia utilizando Diagnostic Viewer
 - Solucione los errores
 - Compare los mensajes de advertencia para determinar si son exclusivos de la nueva versión o también aparecían en la versión anterior
 - Determine la gravedad de los mensajes de advertencia
 - Evalúe si debe solucionar los mensajes de advertencia realizando actualizaciones en el modelo o adoptando nuevas funcionalidades
 - Documente cualquier otro problema en la lista de problemas de actualización; por ejemplo, documente las herramientas personalizadas que hayan presentado problemas no detectados por Upgrade Advisor

Durante la fase de migración inicial, solo realice la cantidad mínima de cambios imprescindibles en archivos, diccionarios de datos, librerías de bloques personalizados y herramientas personalizadas de MATLAB, para asegurarse de que el modelo pueda actualizarse. Todos los archivos de los que dependa el modelo se deben agregar a la ruta de MATLAB. Los cambios en nuevas funcionalidades, optimizaciones, conjuntos de configuración y API de función S se deben realizar posteriormente, en la subfase de migración de herramientas personalizadas.

Para el diccionario de datos, cargue los datos asociados a los modelos o el diccionario de datos, y solucione cualquier posible problema cargando los datos o actualizando correctamente el modelo después de cargarlos.

En el caso de las librerías de bloques personalizados, Upgrade Advisor actualiza automáticamente los bloques de librería personalizados de Simulink presentes en el modelo. Upgrade Advisor actualizará los bloques personalizados creados a partir de bloques integrados de Simulink si están presentes en una librería de bloques personalizados y los enlaces de la librería se encuentran activos. También actualizará los bloques de función S personalizados presentes en el modelo o en una librería personalizada, y generará resultados para cada comprobación de Upgrade Advisor. Después de revisar los resultados, Upgrade Advisor puede implementar los cambios en los bloques automáticamente. Upgrade Advisor no actualizará las máscaras; eso se debe hacer manualmente.

En situaciones excepcionales, cuando se actualiza una versión mucho más antigua que la de destino, es posible que Upgrade Advisor no consiga actualizar los modelos y no pueda identificar la causa raíz del problema. Muchas veces, estas situaciones se deben a personalizaciones realizadas en MATLAB y Simulink o a un patrón de modelado límite que no se pretendía o esperaba para Simulink. En estos casos, se recomienda actualizar a una versión intermedia de MATLAB y Simulink en primer lugar, para aislar y depurar los problemas, y actualizar a la versión de destino una vez que se hayan solucionado. Consulte la sección de pruebas iniciales de este documento para obtener más información sobre cómo depurar problemas en una versión intermedia.

3.3.4 Migración de herramientas personalizadas

Además de migrar las herramientas personalizadas, también es importante aprovechar la ocasión para eliminar las funcionalidades de herramientas personalizadas y adoptar las funcionalidades integradas.

3.3.4.1 Actualización de herramientas personalizadas

Se recomienda actualizar las clases de almacenamiento personalizadas, los archivos de destino del sistema personalizados, los archivos de configuración de MATLAB Report y Simulink Report Generator, las comprobaciones de Model Advisor personalizadas, las pautas de estilo de modelado y los scripts personalizados de MATLAB cuando actualice a una nueva versión de MATLAB y Simulink. Si cuenta con modelos de plantillas o de demostración, considere modificarlos adoptando las funcionalidades nuevas de Simulink que le beneficiarían. Además, este es un buen momento para migrar los parámetros de configuración a los ajustes recomendados por MathWorks. También puede ser una buena ocasión para acordar parámetros de configuración comunes para toda la organización.

Las pautas de estilo de modelado se deberán actualizar a la nueva versión. Considere eliminar las que ya no se apliquen y agregar pautas actualizadas para nuevas funcionalidades de Simulink que tenga previsto utilizar.

Mientras se desarrollan y migran las herramientas personalizadas, se recomienda crear un documento de diseño en el que se detallen los requisitos y la funcionalidad prevista para esas herramientas. Para las herramientas personalizadas, siga un proceso de desarrollo similar al que emplee para el software de producción. Desarrolle planes y casos de prueba, y utilícelos para probar las herramientas personalizadas cuando actualice a una nueva versión de MATLAB y Simulink. Cuando sea posible, automatice los planes de prueba y ejecute las pruebas automatizadas en cada versión preliminar y en la versión de destino de MATLAB y Simulink. Una vez que haya actualizado a una nueva versión de MATLAB y Simulink, actualice los documentos de diseño, y los planes y casos de prueba como sea necesario.

Revise las siguientes secciones e identifique los archivos que deba actualizar. Revise las nuevas funcionalidades de Simulink, las optimizaciones y las API documentadas para aprovechar la ocasión de eliminar las funcionalidades de herramientas personalizadas y adoptar las funcionalidades integradas de Simulink.

Scripts y archivos de MATLAB

Actualice los scripts y archivos de MATLAB que sean necesarios para realizar correctamente la actualización, simulación y generación de código desde el modelo sin errores. He aquí algunos ejemplos:

Scripts de MATLAB que interactúan con MATLAB	Scripts de MATLAB que interactúan con Simulink
<ul style="list-style-type: none">• Proyecto de MATLAB• Scripts de configuración del entorno de MATLAB• Scripts de configuración del compilador• Scripts de análisis de datos• Interfaces gráficas creadas por usuarios	<ul style="list-style-type: none">• Scripts que cargan parámetros• Scripts que cargan conjuntos de configuraciones• Scripts que establecen opciones de configuración• Scripts para menús personalizados• Scripts de interfaz de usuario• Scripts de callbacks de bloques• Scripts de máscaras

Si ha desarrollado interfaces gráficas de usuario con GUIDE, conviértalas a App Designer. MathWorks recomienda utilizar proyectos en lugar de callbacks de modelos. Si aún necesita utilizar callbacks de modelos (por ejemplo, scripts de precarga y poscarga), asegúrese de activar el seguimiento de callbacks como ayuda para depurar los problemas que surjan.

Personalización y configuración de Simulink

Actualice los archivos de Simulink que sean necesarios para realizar correctamente la actualización, simulación y generación de código desde el modelo sin errores. He aquí algunos ejemplos:

Simulación	Generación de código	Verificación y validación
<ul style="list-style-type: none"> • Datos de la simulación • Personalización del registro • Personalización de la visualización • Archivos de configuración de la generación de informes • Herramientas de comparación de datos 	<ul style="list-style-type: none"> • Archivos de destino del sistema (STF) • Archivos de creación de plantillas (TMF) • Archivos de compilador de lenguaje de destino (TLC) • Archivos de plantillas de generación de código (CGT) • Archivos de librería de sustitución de código (CRL) • Entornos SIL/PIL 	<ul style="list-style-type: none"> • Personalización de pruebas unitarias • Datos de prueba • Informes • Herramientas de creación de señales y pruebas • Comprobaciones personalizadas de Model Advisor • Personalización de Simulink Design Verifier • Configuración de herramientas de análisis estático de Polyspace • Entornos SIL/PIL

Parámetros de configuración

Una actualización siempre implica cambios en los parámetros de configuración: ajustes nuevos, modificados y obsoletos. Estos cambios se deben identificar, comprender, documentar y aplicar. En las versiones más recientes, los parámetros de configuración de un modelo se pueden exportar desde Model Explorer. Para información sobre cómo guardar un conjunto de parámetros de configuración, consulte [Save a Configuration Set](#) en la documentación de Simulink. En versiones más antiguas que carecen de esta funcionalidad, se puede crear un script personalizado para exportar estos parámetros.

Los parámetros de configuración se deben actualizar al principio de la fase de migración, para garantizar que se prueben correctamente, que los modelos sean compatibles para la integración y beneficie a todos los usuarios. Se recomienda establecer un proceso para adoptar los cambios en los parámetros de configuración.

Diccionario de datos

Cargue el diccionario de datos y solucione problemas cargando o actualizando el modelo correctamente. Actualice los scripts de MATLAB y los diccionarios de datos basados en ASCII a Simulink Data Dictionary y Embedded Coder Dictionary. Consulte [Simulink Data Dictionary](#) y [Embedded Coder Dictionary](#) para obtener más información.

Librerías de bloques personalizados

Cuando actualice las librerías de bloques personalizados a la nueva versión, aproveche la oportunidad para migrar los bloques personalizados a funcionalidades o bloques integrados de Simulink. Puede utilizar las [notas de la versión](#) para filtrar y buscar nuevos bloques y funcionalidades. También es una oportunidad para mejorar y optimizar los bloques personalizados si existen funcionalidades nuevas o mejoradas.

Es posible que los nuevos bloques integrados con la misma funcionalidad tengan valores de parámetros de diálogo diferentes a los de los bloques personalizados actuales. En tal caso, puede copiar los parámetros de diálogo de los bloques desde los campos de parámetros antiguos a los campos de parámetros nuevos. Este paso se puede realizar manual o automáticamente con un script de MATLAB. Si va a eliminar gradualmente un bloque que ya no se desea utilizar, deberá crear una librería para elementos existentes en la ruta de MATLAB y marcarla claramente para que los usuarios no accedan a ella. Puede eliminar esta librería después de actualizar los modelos en la versión nueva de MATLAB y Simulink, y ejecutar los scripts de actualización de MATLAB. Ejecute Upgrade Advisor en las librerías. Si tiene previsto ofrecer soporte a más plataformas y ha realizado cambios en las funciones S, deberá volver a compilarlas.

Configuración personalizada de Model Advisor

Las comprobaciones y configuraciones de Model Advisor suelen cambiar con cada nueva versión. Una actualización ofrece la oportunidad de revisar la configuración actual de Model Advisor y considerar nuevas comprobaciones. Todas las comprobaciones personalizadas deben probarse, y actualizarse si es necesario. Se debe desarrollar por adelantado un conjunto de comprobaciones específicas de la versión que satisfagan las necesidades y pautas de la organización, para ahorrar tiempo a varios equipos de trabajo.

3.3.4.2 Pruebas de herramientas personalizadas

Utilice planes y casos de prueba para probarlas y verificar que las herramientas personalizadas de la nueva versión de MATLAB y Simulink funcionan según lo previsto. Después, pruebe el proceso de actualización de los modelos. Es buena idea probar las herramientas personalizadas y el proceso de actualización en varios equipos informáticos con todos los sistemas operativos compatibles. Si no ha desarrollado pruebas de regresión automatizada para las herramientas personalizadas, créelas ahora para que se puedan reutilizar cada vez que actualice esas herramientas personalizadas.

3.3.4.3 Pruebas de herramientas de terceros

Se recomienda probar todas las herramientas de terceros con la nueva versión de MATLAB y Simulink antes de lanzarlas. En algunos casos, es posible que los proveedores no lancen inmediatamente una nueva versión que sea compatible con una versión nueva de MATLAB y Simulink. Verifique los plazos con sus proveedores y planifique consecuentemente. He aquí algunos ejemplos:

Productos de terceros que se deben tener en cuenta al actualizar

- Herramientas de administración de requisitos (por ejemplo, IBM Rational DOORS)
- Herramientas de soporte para calibración
- Herramientas de administración de configuración integradas en los productos de MATLAB y Simulink
- Herramientas de administración de cambios integradas en los productos de MATLAB y Simulink
- Herramientas de creación/ingeniería de sistemas
- Herramientas de cosimulación
- IDE
- Compiladores
- Hardware

3.3.5 Migración automática

Después de realizar la migración inicial y de herramientas personalizadas, el siguiente paso es automatizar el proceso de actualización lo más posible.

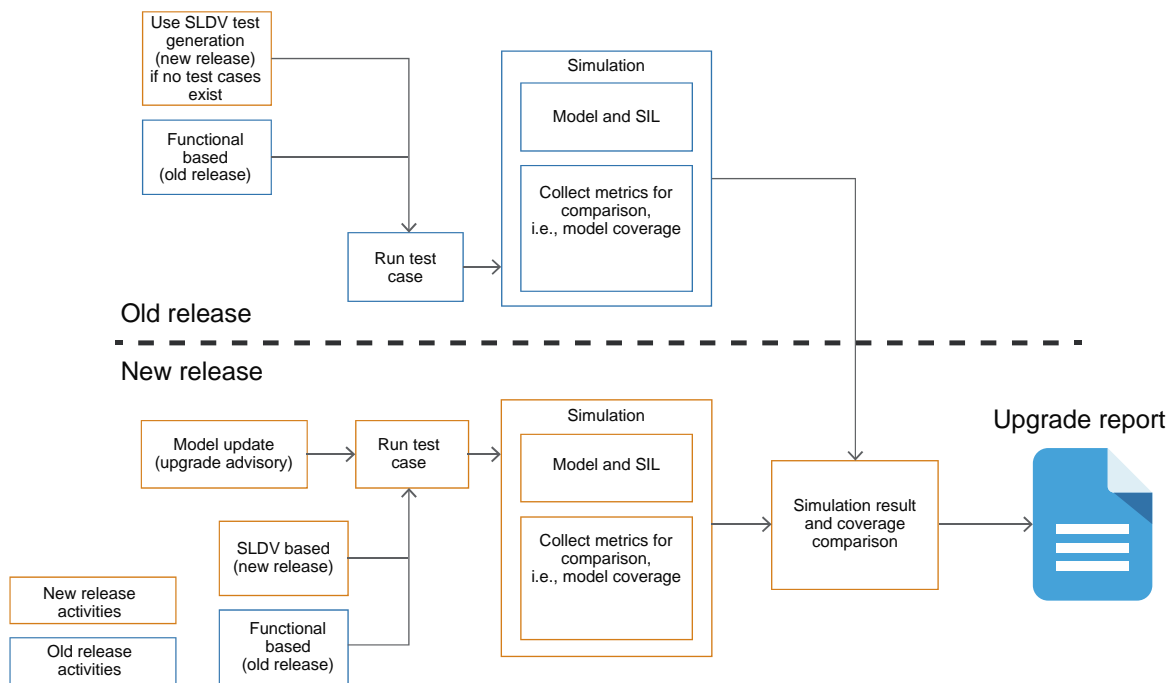
3.3.5.1 Herramientas de automatización de la actualización

Se recomienda automatizar el proceso de actualización de modelos y herramientas personalizadas. Un enfoque que resulta práctico es crear un script principal de actualización de MATLAB que llame a todos los scripts de actualización, de modo que solo haya que ejecutar un único script de MATLAB para actualizar un modelo. Este proceso se debe automatizar lo más posible. El script de MATLAB debe llamar a diversas funciones para automatizar lo siguiente:

- Upgrade Advisor
- Contenido de librerías personalizadas

- Parámetros de configuración
- Objetos del espacio de trabajo
- Pruebas de regresión después de la actualización
- Proceso de actualización de herramientas personalizadas

Estas herramientas de automatización generan un informe con detalles sobre errores y advertencias de cada fase. A continuación, se muestra un ejemplo de flujo de trabajo utilizando Simulink Design Verifier (SLDV) y simulaciones de SIL para lograrlo. Este flujo de trabajo se puede implementar simplemente con scripts, o con una interfaz gráfica que automatice el flujo de trabajo y guíe a los usuarios durante el proceso de actualización. La siguiente imagen muestra un ejemplo del proceso de actualización automática de modelos de Simulink de una versión a otra.



Ejemplo de proceso de actualización

3.3.5.2 Pruebas de herramientas de automatización

Ejecute el script de actualización con las herramientas personalizadas en todos los modelos antes de lanzar la nueva versión de MATLAB y Simulink internamente en la organización. Si no puede actualizar todos los modelos, seleccione modelos con estilos de modelado y tamaños diferentes de diversos grupos. Una vez que haya actualizado los modelos, debe actualizar y simular los diagramas, y generar código. Revise los mensajes de advertencia y documente los errores que aparezcan.

Una práctica recomendada es automatizar este procedimiento de prueba, con el fin de ejecutar pruebas por lotes en un grupo de modelos y documentar los problemas automáticamente. Es posible que tenga que buscar soluciones temporales o desarrollar scripts de MATLAB para resolver algunos de esos problemas antes de lanzar la nueva versión de MATLAB y Simulink. Esta ejecución por lotes también se puede implementar en un servidor de integración continua. Empleando un servidor de integración continua, se pueden agrupar los recursos informáticos, y las actividades se pueden realizar automáticamente a la hora de comenzar un proceso de actualización.

3.3.5.3 Migración del resto de los modelos

Hasta ahora, ha actualizado las herramientas personalizadas y las librerías de bloques personalizados, y ha desarrollado herramientas de automatización de la actualización. El siguiente paso es actualizar el resto de los modelos, ya sea en un equipo local, o en un servidor de integración continua para completar el proceso de actualización automáticamente.

Mientras se actualizan los modelos, documente los problemas que surjan en modelos, herramientas de automatización, librerías de bloques personalizados o herramientas personalizadas. Solucione los problemas siguiendo los pasos recomendados en la fase de migración, y luego vuelva a ejecutar las herramientas de automatización. Repita estos pasos de manera iterativa hasta solucionar todos los problemas conocidos. Este bucle iterativo también se puede realizar en un servidor de integración continua que ejecuta las actividades de actualización.

3.4 Pruebas

El objetivo de la fase de pruebas es garantizar que el modelo y el código de la nueva versión de MATLAB y Simulink sean funcional y numéricamente equivalentes al modelo y el código de la versión anterior, dentro de límites aceptables.

3.4.1 Pruebas de regresión

Ejecute pruebas de regresión en las tareas más frecuentes para estimar cuánto tardarán esas tareas en la nueva versión de MATLAB y Simulink. Cuando sea posible, ejecute las mismas tareas en la versión anterior y compare los resultados de las pruebas. Se recomienda ejecutar estas pruebas en modelos de gran tamaño que se estén utilizando actualmente.

Esta actividad servirá para descubrir posibles problemas antes de lanzar la nueva versión. Es más fácil retrasar la actualización de MATLAB y Simulink que retraer una versión nueva o de herramientas personalizadas ya lanzadas. Las pruebas son fundamentales para minimizar el efecto de la actualización en los programas de producción, ya que permiten identificar problemas y buscar soluciones o desarrollar los remedios temporales antes de implementar la actualización.

Las pruebas de regresión deben ser exhaustivas, con diversas comprobaciones respecto de una base de referencia de resultados actuales. Las comprobaciones de coherencia ayudan a identificar cambios y, en última instancia, determinan el éxito o fracaso de una actualización. La siguiente tabla presenta una lista de elementos clave que se deben comprobar para evaluar una actualización. Es una lista parcial y representativa de las características importantes. Tenga en cuenta que se deben tomar en consideración tanto la funcionalidad como el rendimiento. Las pruebas, preferentemente automatizadas, que abarcan estos aspectos hacen que el proceso sea más accesible. Toda mejora de las pruebas realizadas como parte del proceso de actualización se debe integrar en el flujo de trabajo de desarrollo. Por último, el plan de prueba se debe escalar como corresponda en las diferentes fases.

Resultado	Ideal	Aceptable	Notas
Resultados de simulación	Resultados idénticos	Dentro del límite de tolerancia	Las señales registradas deben incluir todos los resultados de los modelos. Se pueden registrar señales adicionales si se considera que son resultados intermedios importantes.
Métricas de cobertura	Resultados idénticos	Explicado por los cambios en las herramientas de cobertura	Los resultados de simulación de las pruebas que alcancen cerca del 100% de cobertura ofrecen una mayor confianza. NOTA: Un cambio de herramienta de cobertura de pruebas puede generar cambios en las métricas de cobertura entre versiones. Es importante tenerlo en cuenta para poder confiar en los resultados de las pruebas.
Generación de código	Equivalente	Sin errores	Las pruebas de SIL pueden generar resultados aceptables, aunque no ideales, pero aún adecuados.
Pruebas de SIL	Resultados idénticos	Dentro del límite de tolerancia	Las señales registradas deben incluir todos los resultados de los modelos. Se pueden registrar señales adicionales si se considera que son resultados intermedios importantes.
Duración de la simulación	Menor que en la versión anterior	Dentro del límite de tolerancia	Un ligero aumento de la duración de la simulación en la versión nueva respecto de la versión anterior puede ser aceptable.
Duración de la generación de código	Menor que en la versión anterior	Dentro del límite de tolerancia	Un ligero aumento de la duración de la generación de código en la versión nueva respecto de la versión anterior puede ser aceptable.
Compatibilidad con terceros	Equivalente	Equivalente	El software, las herramientas y el posible hardware de terceros deben funcionar con nuevas versiones de MATLAB.
Errores y advertencias	Resultados similares		Se deben solucionar los nuevos errores. Se deben investigar y comprender las nuevas advertencias.

3.4.2 Pruebas de equivalencia

Se recomienda ejecutar pruebas de equivalencia para comparar los resultados de la simulación de cada modelo y código de la versión anterior con los del modelo y código de la nueva versión. El ejemplo de proceso de la sección 3.3.5.1 tiene una fase de pruebas de equivalencia. Es importante que solo se incorporen nuevas funcionalidades de MATLAB y Simulink al modelo después de completar las pruebas de equivalencia. En caso contrario, es posible que los modelos y el código generado no sean funcionalmente equivalentes entre versiones. Simulink Test permite [ejecutar pruebas en varias versiones de MATLAB](#), [pruebas de equivalencia](#) y [pruebas de código entre versiones](#). Con Simulink Design Verifier, se pueden generar casos de prueba que den como resultado una cobertura del 100% de pruebas de equivalencia.

3.4.3 Pruebas de integración continua

Se recomienda un entorno de integración continua para actualizar, integrar, simular, generar código, realizar pruebas de equivalencia y generar informes de pruebas automáticamente. Este debe ser el objetivo a largo plazo para organizaciones de gran tamaño, ya que hará que el proceso de actualización sea más fluido que convertir y probar manualmente cada modelo. También conseguirá que sea rentable y tenga un impacto mínimo en los usuarios finales. Utilizar un servidor de integración continua reduce notablemente la cantidad de esfuerzo de ingeniería manual que se necesita durante la actualización. El mismo flujo de trabajo que se detalla en la sección 3.3.5.2 se puede utilizar en todas las actualizaciones de modelos y herramientas. El servidor de integración continua debe implementar el proceso personalizado que se haya establecido para actualizar los modelos y archivos de MATLAB. Estas tareas se repiten para cada componente en el servidor de integración continua hasta que se hayan intentado actualizar todos los modelos y archivos de MATLAB.

3.4.4 Pruebas beta

En la fase final de pruebas, encargue a un pequeño grupo de usuarios que realice pruebas beta de las herramientas personalizadas y del proceso de actualización de modelos. Seleccione usuarios con experiencia en MATLAB y Simulink, y conocimiento profundo de los modelos de Simulink pertinentes. Incluya un método para proporcionar feedback y así mejorar el proceso y las herramientas. El grupo debe comenzar a utilizar la nueva versión de MATLAB y Simulink, las herramientas personalizadas y las herramientas de terceros que sean necesarias en sus tareas y proyectos diarios.

3.4.5 Pruebas de actualizaciones

Pruebe las actualizaciones de MATLAB y Simulink e incorpórelas en los planes de prueba de la organización. MathWorks recomienda aplicar la última actualización en cuanto esté disponible y sea conveniente para su organización o programas. Antes de lanzar la versión de destino de MATLAB y Simulink en la organización, compruebe si existe una actualización de esa versión y repita las pruebas de regresión realizadas previamente con la versión más reciente. Lo ideal sería que la organización haya automatizado este tipo de pruebas con un servidor de integración continua y se pueda evaluar la actualización con una prueba rápida. Si no es así, se recomienda realizar las pruebas de regresión documentadas en la fase de evaluación de este white paper para identificar los problemas que podría provocar la introducción de la actualización en este punto del proceso de actualización. Las actualizaciones incluyen correcciones de fallos, de modo que migrar a la actualización más reciente no debería ocasionar ningún problema. Si detecta algún problema, comuníquese con el servicio de soporte de MathWorks [creando una solicitud de servicio en el sitio web de MathWorks](#), para que se pueda incluir la corrección en una futura actualización. Después de actualizar a la nueva versión, estas pruebas de regresión pueden repetirse rápidamente en un servidor de integración continua cada vez que MathWorks lance una nueva actualización, con objeto de evaluar el impacto de lanzar esa actualización en la organización.

3.5 Lanzamiento

El propósito de la fase de lanzamiento es publicar la nueva versión de MATLAB y Simulink, las herramientas personalizadas y de automatización, así como ofrecer formación a los usuarios.

3.5.1 Formación

Se recomienda ofrecer dos tipos de clases de formación a los usuarios durante una actualización. El primer tipo es un seminario o workshop en el que se resuman algunas de las principales funcionalidades nuevas que están disponibles en las versiones actualizadas de MATLAB y Simulink, y que destaque las funcionalidades y los beneficios más relevantes para la organización. El segundo tipo debe centrarse en el proceso de actualización, soluciones temporales recomendadas para determinados problemas, nuevas pautas de estilo y nuevas funcionalidades para las herramientas personalizadas.

3.5.2 Lanzamiento

Cuando se hayan completado las pruebas, todo estará listo para lanzar las herramientas personalizadas a los usuarios. Una práctica recomendada es incluir notas de la versión de las herramientas personalizadas. Otra práctica recomendada es ofuscar el código de MATLAB *utilizando código P* en los archivos antes de lanzarlos, aunque las herramientas solo se vayan a utilizar internamente. Esto evitará que el equipo de ingeniería corrija problemas por su cuenta sin informar a otros usuarios, que utilice distintas versiones de las herramientas personalizadas, y que se produzcan problemas en futuras actualizaciones. Coloque las herramientas personalizadas o el instalador en una ubicación a la que puedan acceder todos los usuarios de la organización. Cuando notifique a los usuarios que están disponibles las nuevas versiones de las herramientas personalizadas y de MATLAB y Simulink, recuerde incluir instrucciones de instalación y actualización.

Se recomienda que el equipo de ingeniería actualice sus propios modelos de Simulink en lugar de encargar a otro grupo que realice la migración. El equipo de ingeniería que haya desarrollado los modelos tiene la experiencia necesaria para realizar la validación. Además, conocen los plazos de producción a los que se enfrentan y qué partes del modelo tendrán que modificar para versiones futuras del software de producción. Si deciden actualizar el modelo a la nueva versión, deberán ponerse de acuerdo para utilizar el modelo validado por otros grupos de interés clave, que puede incluir otros grupos que se verán afectados posteriormente durante el proceso. El objetivo es garantizar que la migración a una nueva versión de MATLAB y Simulink no provoque retrasos en el cronograma de producción. Se debe exigir que todos los nuevos proyectos de modelado utilicen la nueva versión de MATLAB y Simulink. Es buena idea establecer un plazo para que todos los usuarios completen la migración de sus modelos a la nueva versión de MATLAB y Simulink. Entre tanto, es posible que deba admitir varias versiones de MATLAB y Simulink.

El propósito de realizar pruebas exhaustivas y crear scripts de MATLAB es automatizar el proceso lo más posible. Si por alguna razón el equipo de ingeniería no puede actualizar sus modelos, se necesitarán casos de prueba que produzcan el nivel deseado de cobertura de pruebas de modelos (se recomienda el 100%). Pruebe el modelo y el código generado en un entorno de SIL, y verifique que los resultados de la simulación y la generación de código coincidan con los resultados de la versión anterior de MATLAB y Simulink, a través de automatización, si es posible.

3.6 Soporte

El propósito de la fase de soporte es realizar de manera continua actividades posteriores a la actualización, con el fin de reducir el volumen de trabajo en la siguiente actualización y prestar soporte continuo a los usuarios.

3.6.1 Actividades posteriores a la actualización

Después de completar la actualización, se pueden realizar varias actividades para facilitar la siguiente actualización. Si no ha creado documentos de diseño que detallen los requisitos y la funcionalidad de las herramientas personalizadas durante la fase de migración, debería hacerlo en la fase de soporte.

Aproveche la ocasión para crear planes y casos de prueba para las herramientas personalizadas y los flujos de trabajo típicos que se utilizan en la organización. Los podrá utilizar para probar las herramientas personalizadas cuando actualice a una nueva versión de MATLAB y Simulink. Los casos de prueba deben incluir los resultados que se esperan de cada prueba. Si los planes de prueba están documentados y existen casos de prueba, será más fácil automatizar las pruebas la próxima vez que se realice una actualización. Puede ejecutar pruebas de regresión automatizadas en cada versión preliminar y compartir sus impresiones con MathWorks, incluso si no tiene intención de actualizarse a esa versión. Este feedback permite que MathWorks mejore el producto o corrija fallos antes de migrar a la siguiente versión de MATLAB y Simulink. También puede ejecutar pruebas de regresión automatizadas en la nueva versión de destino de MATLAB y Simulink que vaya a utilizar, así como en cualquier actualización que lance MathWorks.

Después de automatizar las pruebas de las herramientas personalizadas, el siguiente paso es automatizar los procesos de migración y validación de modelos, para poder ejecutar pruebas por lotes en los modelos en un entorno de integración continua y generar automáticamente informes sobre los errores. El modelo y el código generado deben probarse en un entorno de SIL con el nivel deseado de cobertura de pruebas, para verificar que los resultados de la simulación y el código generado coinciden con los resultados de la versión anterior de MATLAB y Simulink. Este tipo de pruebas de SIL también se puede automatizar para reducir el volumen de trabajo en la siguiente actualización. La siguiente es una lista parcial de pruebas que se pueden ejecutar en un entorno de integración continua:

- Validar que los resultados de las herramientas personalizadas coinciden entre las versiones de MATLAB y Simulink
- Migrar un grupo de modelos a la nueva versión de MATLAB y Simulink
- Ejecutar actualizaciones de modelos y simulaciones, y generar código en un grupo de modelos en la nueva versión de MATLAB y Simulink
- Diferenciar el código generado entre las versiones de MATLAB y Simulink
- Comparar los resultados de la simulación entre las versiones de MATLAB y Simulink
- Comparar los resultados de la simulación con los resultados del código generado en la misma versión de MATLAB y Simulink
- Comparar los resultados del código generado entre las versiones de MATLAB y Simulink

Realizar una evaluación retrospectiva puede ayudar a documentar las lecciones aprendidas. Requiere esfuerzo y experiencia determinar qué lecciones de arquitectura, estilo de modelado y personalización se aplican a una actualización en general y cuáles a una versión en concreto. Cree un repositorio o una página wiki para documentar esas lecciones con enlaces a las lecciones específicas de cada versión. Documente las nuevas lecciones en la página específica de la versión correspondiente, y revísela periódicamente para identificar y divulgar las cuestiones más generales.

4 Sostenimiento: Una filosofía de actualización continua

MathWorks recomienda adoptar una filosofía de actualización continua. Realizar actividades de actualización de manera continua garantiza que la siguiente actualización sea más fácil que la anterior. Como ayuda para adoptar esta filosofía, considere la posibilidad de aprovechar las pruebas de versiones preliminares y las pruebas de modelos industriales, así como seminarios, webinars y conferencias de MathWorks.

4.1 Pruebas de versiones preliminares

Cuando sea posible, pruebe los modelos y las herramientas personalizadas para cada versión preliminar y comunique los posibles problemas al servicio de soporte de MathWorks [creando una solicitud de servicio en el sitio web de MathWorks](#). Es mejor no esperar a probar los modelos solo en la versión a la que desea actualizar, ya que se incorporan muchas funcionalidades nuevas entre la versión actual y la futura. Si prueba las versiones preliminares cada seis meses, los comentarios que envíe después de esas pruebas iniciales permitirán que MathWorks corrija los problemas cuando llegue el momento de la siguiente actualización. Probar cada versión preliminar no solo permite evaluar la versión más exhaustivamente que leyendo las notas de la versión, sino que disminuye el esfuerzo necesario para actualizar a una nueva versión de MATLAB y Simulink. Además, facilita el aprendizaje de nuevas funcionalidades y proporciona orientación a quienes toman la decisión sobre el momento indicado para realizar la actualización.

4.2 Pruebas de modelos de clientes

Considere someter sus modelos al sistema IMT de MathWorks. El siguiente es un fragmento de la ficha informativa de las pruebas de modelos de clientes:

El sistema IMT de pruebas de modelos de clientes forma parte de una iniciativa de calidad estratégica de MathWorks. Está diseñado para identificar, aislar y evitar regresiones directamente en el software de MathWorks de clientes. Como parte del sistema, MathWorks prueba modelos de clientes reales en un entorno seguro, integrado en los procesos de compilación, pruebas y lanzamiento de software de MathWorks. Incluir un modelo en el proceso IMT reduce notablemente las incompatibilidades de versión.

Si este proceso no es adecuado para su organización, puede crear uno similar internamente. Un primer paso es implementar la automatización de pruebas en un entorno de integración continua. Una forma de emular el sistema IMT de MathWorks consiste en emplear una versión de destino específica de MATLAB y Simulink, al menos para las versiones preliminares y generales. De ese modo, se obtiene conocimiento de posibles problemas y mayor confianza en una nueva versión con antelación.

4.3 Seminarios, webinars y conferencias

Asistir a seminarios, webinars y conferencias de MathWorks permite mantenerse al día de las nuevas funcionalidades y ayuda a tomar decisiones bien fundadas sobre qué versión de MATLAB y Simulink conviene a la organización.

5 Soporte de MathWorks

MathWorks ofrece una gran variedad de opciones de soporte autoservicio para el proceso de actualización. Entre esas opciones autoservicio se incluyen [documentación de productos online](#), [notas de la versión](#) que se pueden filtrar para buscar nuevas funcionalidades e incompatibilidades, [ejemplos](#) sobre cómo utilizar esas nuevas funcionalidades, [MATLAB Answers](#), [informes de fallos](#) y [MATLAB Central](#), una comunidad de usuarios activa. Si necesita soporte adicional, comuníquese con el servicio de [soporte](#) para resolver preguntas específicas sobre los productos. Comuníquese con [formación](#) para cursos sobre productos. Comuníquese con los [servicios de consultoría](#) para recibir asistencia para la actualización.

Lista de comprobación
EVALUACIÓN: COMPRENDER EL IMPACTO DE UNA ACTUALIZACIÓN
<input type="checkbox"/> Identificar el desencadenante de la actualización
<input type="checkbox"/> Evaluar la situación actual, incluyendo posibles incompatibilidades
<input type="checkbox"/> Seleccionar una versión de destino de MATLAB y Simulink
<input type="checkbox"/> Probar algunos modelos en la nueva versión con flujos de trabajo típicos
<input type="checkbox"/> Ejecutar pruebas de regresión en las tareas más frecuentes
<input type="checkbox"/> Crear un informe de evaluación de la actualización
<input type="checkbox"/> Decisión de proceder o no proceder
PLANIFICACIÓN: DEFINIR EL ALCANCE GENERAL DEL PROYECTO DE ACTUALIZACIÓN
<input type="checkbox"/> Crear un caso de negocio que refleje la rentabilidad sobre la inversión
<input type="checkbox"/> Establecer el alcance identificando los modelos, proyectos y organizaciones afectados
<input type="checkbox"/> Crear un plan detallado con los plazos, los recursos y los grupos de interés, entre otros elementos
MIGRACIÓN: CONVERTIR UN PROYECTO DE MANERA ITERATIVA A LA NUEVA VERSIÓN
<input type="checkbox"/> Migración inicial: migrar un pequeño conjunto de modelos y probar los flujos de trabajo típicos
<input type="checkbox"/> Migración de herramientas personalizadas: sustituir la funcionalidad de las herramientas personalizadas por la funcionalidad integrada de MATLAB y Simulink
<input type="checkbox"/> Migración automática: ampliar las pruebas a modelos con diferentes estilos de modelado y utilizar las herramientas de automatización para actualizar el resto de los modelos
PRUEBAS: VALIDAR LOS MODELOS ACTUALIZADOS
<input type="checkbox"/> Ejecutar pruebas de regresión en las tareas más frecuentes
<input type="checkbox"/> Realizar simulaciones de lazo abierto, MIL, SIL y PIL en equipos de escritorio utilizando productos de Simulink para la verificación y validación
<input type="checkbox"/> Realizar pruebas de HIL, de prototipado rápido y de prototipado rápido de destino en el laboratorio
<input type="checkbox"/> Probar las herramientas de terceros
<input type="checkbox"/> Validar la equivalencia funcional de los modelos en la nueva versión
<input type="checkbox"/> Realizar pruebas beta de la nueva versión con usuarios seleccionados
LANZAMIENTO: LANZAR LA NUEVA VERSIÓN DE MATLAB Y LAS HERRAMIENTAS PERSONALIZADAS
<input type="checkbox"/> Programar e impartir clases de formación
<input type="checkbox"/> Lanzar la nueva versión de MATLAB y las herramientas personalizadas
<input type="checkbox"/> Recomendar que expertos conviertan sus modelos
<input type="checkbox"/> Establecer un plazo para que todos los usuarios completen la migración
SOPORTE: AUTOMATIZAR EL PROCESO DE ACTUALIZACIÓN Y COMUNICAR LOS ERRORES
<input type="checkbox"/> Considerar automatizar el proceso de actualización utilizando un servidor de integración continua
<input type="checkbox"/> Realizar una evaluación retrospectiva y documentar las lecciones aprendidas de la actualización
<input type="checkbox"/> Probar cada versión preliminar y comunicar los errores al servicio de soporte de MathWorks