

白皮书

大型企业中的 MATLAB 和 Simulink 版本升级

作者：MathWorks 的 Judy Wohletz、Vinod Reddy 和 Jim Ross

目录

1 简介	4
2 概述	5
2.1 评估	5
2.2 规划	6
2.3 迁移、测试和发布	6
2.3.1 迁移	7
2.3.2 测试	9
2.3.3 发布	10
2.4 支持	10
3 详细的升级工作流	11
3.1 评估	11
3.1.1 工程启动考虑因素	11
3.1.2 选择 MATLAB 和 Simulink 目标版本	11
3.1.3 做好技术环境准备	11
3.1.4 初始测试	12
3.1.5 回归测试	13
3.1.6 升级决策	13
3.2 规划	13
3.2.1 升级工程管理	13
3.2.2 创建业务案例	16
3.2.3 预先设定目标	16
3.3 迁移	16
3.3.1 设置环境	16
3.3.2 确定要迁移的模型	17
3.3.3 初始迁移	17
3.3.4 自定义工具迁移	18
3.3.5 自动迁移	20
3.4 测试	21
3.4.1 回归测试	22
3.4.2 等效性测试	22

3.4.3 持续集成测试.....	23
3.4.4 beta 测试.....	23
3.4.5 测试更新	23
3.5 发布	23
3.5.1 培训	23
3.5.2 发布	23
3.6 支持	24
3.6.1 升级后活动.....	24
4 可持续性：持续升级理念.....	24
4.1 预发行版测试.....	24
4.2 行业模型测试.....	25
4.3 研讨会、网络研讨会和各种会议.....	25
5 MathWorks 支持.....	25
6 附录	26

1 简介

开发团队历来都是在新项目开始之前升级到新的工具版本，然后在耗时多年的整个产品开发生命周期内始终沿用这些版本。但是，随着各公司纷纷以前所未有的速度推出新技术，开发团队很难在现行的产品开发过程中升级或研究新技术。前瞻性的升级策略为企业带来了契机，让他们可以在变更版本与无法采纳新技术和方法分别带来的风险之间取得平衡。

升级到新版 MATLAB 和 Simulink 后，您的企业便能获得基于模型的设计的最新技术，从而提高效率和收益。升级的净收益可通过投资回报 (ROI) 来衡量。回报的计算包括效率的提升以及成本和风险的降低。投资以评估和实施升级的成本以及组织为了有效利用升级的新功能所需的流程和工具变更成本来衡量。

任何升级的成功与否都取决于实施升级所采用的策略。在本白皮书中，MathWorks 建议了一种系统化的企业级升级方法，旨在最大程度地利用 MATLAB 和 Simulink 产品中的先进功能，并减少实现收益所招致的成本、风险和中断。这些建议的依据是指导企业客户升级到新版 MATLAB 和 Simulink 的实践经验。

升级过程包含下面六个主要阶段：

1. 评估
2. 规划
3. 迁移
4. 测试
5. 发布
6. 支持

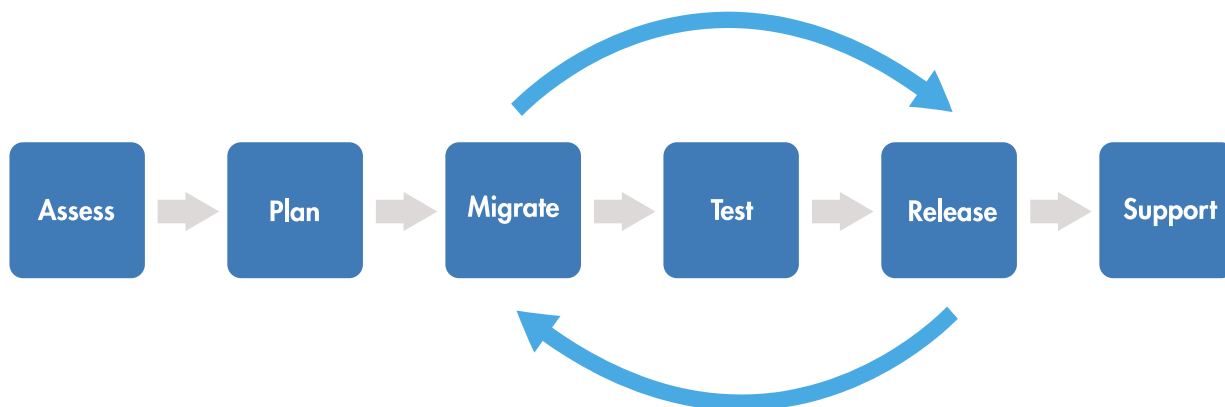
在升级过程中的每个阶段，沟通都是最重要的步骤。在版本升级的整个过程中，所有利益相关方都要充分了解相关情况，这一点至关重要。尽早告知评估、结果和升级决策背后的主要原因有助于用户做好准备。一旦做出升级决策，就应当迅速传达组织计划，以便进行局部规划。在轮到团队进行升级时，随时就时间表、成功体验和此间遇到的困难进行沟通，将有助于他们取得成功。

要实现有效而稳健的升级过程，最终考虑因素是使用持续集成 (CI)。在开发工作流中使用 CI 环境可提高可重复性，并减少执行测试所需的投入。另外，CI 环境的使用有助于日常工作，并可大大减少评估升级所需的工作量。如果 CI 已纳入开发工作流，则可能只需稍加更改，即可对特定 MATLAB 版本进行灵活的测试。如果 CI 未纳入当前的开发工作流，我们建议您不妨一试。

结合这些做法可以最大限度地降低风险，并提高成功升级的几率。

2 概述

要想实现最大投资回报，在进行升级前，必须了解升级过程、可能的升级路径以及潜在收益和相关成本，后面这两点最重要。升级到新版本所需的任务或活动可以分为以下逻辑阶段：评估、规划、迁移、测试、发布和支持。



典型升级工程各阶段的工作流。

升级过程的每个阶段都有触发条件、输入、活动、输出和退出条件。该过程本身由一项维持策略提供支持。此工作流应当由有的放矢的沟通方案提供全程支持。通过这项方案，所有利益相关方随时都能获悉升级目的、状况、决策、时间表以及升级期间发现的重大问题。

2.1 评估

在升级过程中，第一阶段的目标是了解升级所带来的总体影响，以及评估收益是否大于成本、风险和投入。其目的是尽早确定是否存在显著的问题，避免在评估该特定问题上投入额外的精力。在进入升级工程的下一阶段之前，需要获得足够的投资回报。

评估	
目标：了解升级所带来的影响	
触发条件	<ol style="list-style-type: none">1. 操作系统老旧2. 新增 MATLAB 和 Simulink 功能，包括全新的产品和功能3. 新的硬件开发平台4. OEM/供应商/合作伙伴协作5. 新的工程/项目
输入	<ol style="list-style-type: none">1. 待升级工程的现状2. 在您的当前版本和升级版本之间，MATLAB 和 Simulink 发布了多少次？3. MATLAB 和 Simulink、第三方软件和硬件、开发或应用软件和硬件以及计算机和操作系统等开发平台的当前版本4. 升级时间表

活动	<ol style="list-style-type: none"> 1. 选择初始目标版本 2. 编制当前问题列表，以及您期望新版本所能解决的需求列表 3. 找一个团队，让他们使用升级顾问迁移模型、代码和脚本中具有代表性的小样本 4. 迁移模型、代码和脚本中具有代表性的小样本，以便更好地了解版本迁移所带来的影响 5. 对用户通常执行的任务运行回归测试，以收集有关这些任务在新版本中执行所需时间的指标 6. 评估该团队是否有能力成功执行升级工程，是否还需要额外的技术培训 7. 评估该团队是否有足够的资源处理升级工程 8. 准备升级评估报告，其中包括： <ul style="list-style-type: none"> • 升级步骤 • 包括仿真速度、代码大小等效益和指标的结果 • 挑战 • 错误消息 • 基于当前对潜在收益、估计成本和风险的了解所做的 ROI 评估
输出	<ol style="list-style-type: none"> 1. “可行/不可行”决策 2. 初始目标版本 3. 升级评估报告

2.2 规划

规划阶段的目的是确定整个升级工程的总体范围和计划。

规划	
目标：确定升级工程的总体范围	
触发条件	“可行”决策
输入	升级评估报告
活动	<ol style="list-style-type: none"> 1. 创建业务案例 2. 通过了解受影响的模型、工程、组织和主要利益相关方来确定范围
输出	<ol style="list-style-type: none"> 1. 升级的业务案例 2. 升级计划，其中包含：升级时间表；受影响的工程、模型、组织和利益相关方的列表；对所需资源、培训、工作量以及成本和风险的估计；依赖关系；以及硬件和软件的目标版本

2.3 迁移、测试和发布

迁移、测试和发布阶段密切相关，且以迭代的方式进行，以将模型升级到新版本。

2.3.1 迁移

在迁移阶段，通过迭代的方式升级 Simulink 模型、MATLAB 代码、MATLAB App 和自定义工具，以使其适用于新版本。在这里，自定义工具是指某一环境下使用 MATLAB 和 Simulink 创建的任何应用或设置。例如，自定义工具包括：

- 自定义 Simulink 模块库
- S-Function 模块
- MATLAB 启动脚本
- Embedded Coder 字典中的自定义存储类
- 自定义系统目标文件
- MATLAB Report Generator 或 Simulink Report Generator 设置文件
- 自定义模型顾问检查
- 建模指导规范
- 默认配置参数设置
- 用于自定义 MATLAB 和 Simulink 环境的 MATLAB 脚本

迁移阶段分为三个子阶段，通过几次迭代执行：初始迁移、自定义工具迁移和自动迁移。这些迭代通常需要将自动和手动方法结合使用，以升级模型、脚本、用户界面、模板、第三方工具的接口和其他应用。

此阶段的活动只是将模型迁移到新版本。您仍需在升级过程的测试阶段对模型进行测试，如后面章节所述。

2.3.1.1 初始迁移

在初始迁移子阶段，您要迁移一小部分的 Simulink 模型和工程。这些模型是组织内为了制定规程和开发工具以及更深入了解新功能而开发的典型模型。此项迁移是通过 [升级顾问](#) 完成的。该工具旨在帮助您升级和改进当前版本的模型。此子阶段为下一子阶段提供支持，并帮助减少风险和工作量。

初始迁移

目标：迁移模型的子集

触发条件	获批的业务案例
输入	<ol style="list-style-type: none">1. 升级评估报告2. 升级计划3. 模型和其他相关工作件
活动	<ol style="list-style-type: none">1. 对模型运行升级顾问2. 成功更新图、运行仿真并为模型生成代码3. 使用 Simulink 中的诊断查看器记录任何错误或警告消息4. 解决错误（如果有）5. 比较警告消息，以确定这些消息是新版本独有的还是存在于当前版本中6. 评估警告消息的严重性7. 决定是否要解决警告消息8. 将所有其他问题都记录在升级问题列表中

输出	<ol style="list-style-type: none"> 1. 更新的升级评估报告 2. 新的升级问题列表 3. 更新的升级计划 4. 迁移到新版本的模型的子集
-----------	--

2.3.1.2 自定义工具迁移

在自定义工具迁移子阶段，除了迁移自定义工具，还需要采用内置功能来淘汰自定义工具功能。长期目标是采用内置功能来移除自定义工具。这样做将使未来的升级更加容易，也更加自动化，还将大大减少未来升级的工作量和成本。

自定义工具迁移	
目标：升级模型和自定义工具并开发自动化工具	
触发条件	<ol style="list-style-type: none"> 1. 获批的升级计划 2. 迁移模型的子集
输入	<ol style="list-style-type: none"> 1. 升级评估报告 2. 升级问题列表 3. 模型和其他相关工件 4. 自定义工具
活动	<ol style="list-style-type: none"> 1. 将自定义工具升级到新版本 2. 尽可能将自定义工具替换为 MATLAB 和 Simulink 的内置功能 3. 在新版本中验证自定义工具 4. 开发自动化工具，通过自动执行升级过程中的手动步骤来减少工作量并确保一致性
输出	<ol style="list-style-type: none"> 1. 更新的升级评估报告 2. 更新的升级问题列表 3. 更新的升级计划 4. 升级的自定义工具 5. 自动化工具

2.3.1.3 自动迁移

在自动迁移子阶段，使用在上一子阶段制定的规程和开发的自动化工具对剩余模型进行升级。此阶段的两个主要目标是解决迁移问题和改进自动化工具，以便其他团队可以使用这些工具。

自动迁移	
目标：解决与模型、自定义工具和自动化工具有关的已知问题；升级模型、自定义工具和自动化工具	
触发条件	<ol style="list-style-type: none"> 1. 升级了模型和自定义工具 2. 开发了自动化工具
输入	<ol style="list-style-type: none"> 1. 自动化工具 2. 升级的自定义工具 3. 剩余模型和其他相关工件 4. 升级问题列表

活动	<ol style="list-style-type: none"> 1. 对所有剩余模型运行自动化工具以及升级顾问（自动化工具应调用升级顾问） 2. 成功更新图、仿真并为模型生成代码 3. 使用 Simulink 中的诊断查看器记录任何错误或警告消息 4. 解决错误（如果有） 5. 解决阻碍迁移的警告 6. 将所有问题都记录到升级问题列表中 7. 更新自定义工具和自动化工具的帮助文档 8. 编写自定义工具和自动化工具的发行说明 9. 通过改进自定义工具和自动化工具来解决问题 10. 编制或更新培训材料，包括新旧版本之间的相关差异、自定义工具和自动化工具
输出	<ol style="list-style-type: none"> 1. 改进的自动化工具 2. 更新的升级评估报告 3. 更新的升级问题列表 4. 更新的升级计划 5. 升级到新版本的模型 6. 自定义工具和自动化工具的发行说明 7. 更新的自定义工具和自动化工具帮助文档 8. 新的或更新的培训材料

2.3.2 测试

测试阶段的目标是，确保在可接受的限度内，使用新版本时由模型和代码生成的结果与使用上一版本时生成的结果在功能和数值上相同。

测试	
目标：验证升级的模型	
触发条件	升级了模型
输入	<ol style="list-style-type: none"> 1. 升级的自定义工具 2. 模型和其他相关工作件 3. 输入测试数据和预期输出
活动	<ol style="list-style-type: none"> 1. 对用户通常执行的任务运行回归测试，并将得出的结果与使用上一版本时得出的结果进行比较 2. 在桌面端执行开环、模型在环 (MIL)、软件在环 (SIL) 和处理器在环 (PIL) 仿真 3. 在实验室执行硬件在环 (HIL) 仿真、快速原型构建和目标系统快速原型构建 4. 执行常规的任何系统测试或硬件测试 5. 查看结果并接受验证的模型 6. 根据需要定义其他测试数据和预期输出
输出	<ol style="list-style-type: none"> 1. 在新版本中验证的模型 2. 其他测试数据和预期输出（如果需要） 3. 测试报告和相关交付物

2.3.3 发布

在发布阶段，需要发布升级的模型、自定义工具和新版 MATLAB 和 Simulink。

发布	
目标：发布新的 MATLAB 版本和自定义工具	
触发条件	测试阶段完成
输入	<ol style="list-style-type: none">1. 验证的模型2. 输入测试数据和预期输出3. 自定义工具的帮助文档4. 自定义工具的发行说明5. 升级计划
活动	<ol style="list-style-type: none">1. 更新升级计划2. 安排和开设培训课程3. 创建所有材料的存储库4. 通知用户
输出	<ol style="list-style-type: none">1. 更新的升级计划2. 发布的文档、工具和培训材料版本

2.4 支持

在支持阶段，如果出现问题，组织会为模型、自定义工具以及 MATLAB 和 Simulink 产品的用户提供持续支持。

支持	
目标：解决报告的问题	
触发条件	用户发现问题
输入	<ol style="list-style-type: none">1. 自定义工具2. 出现问题的模型3. 帮助文档4. 培训材料
活动	<ol style="list-style-type: none">1. 重现问题2. 查明根源3. 解决问题（存在于模型、自定义工具内或其他地方）4. 就新版本通知用户5. 更新存储库6. 更新问题跟踪列表7. 根据需要更新自定义工具的发行说明8. 根据需要发布带修复程序的更新版本

输出	<ol style="list-style-type: none"> 1. 更新的发行说明 2. 模型或自定义工具的修复版本 3. 更新的问题跟踪列表
----	--

3 详细的升级 workflow

本章将回顾第 2 章定义的六个流程阶段，并详细介绍每个步骤。

3.1 评估

评估阶段的目标是了解升级所带来的总体影响，确定升级的收益是否大于成本、风险和投入。其目的是尽早确定是否存在显著的问题，避免在评估该特定问题上投入额外的精力。如果升级的投资回报不足，组织可能会决定推迟此次升级。每推迟一次升级，升级到未来版本的障碍就会增添一分。升级过程可能由多个事件触发，包括新工程、与新公司合作、新硬件或新操作系统。

3.1.1 工程启动考虑因素

升级过程的第一步是评估当前情况。此项评估包括以下大部分或全部步骤：

- 为您公司当前使用的软件和硬件创建清单
- 记录您公司以及合作公司目前使用的 MATLAB 和 Simulink 版本
- 记录目前与 MATLAB 和 Simulink 产品一起使用的第三方软硬件工具
- 确定第三方供应商打算为哪些版本的 MATLAB 和 Simulink 产品提供支持
- 确定您公司目前以及未来打算为哪些计算机和操作系统提供支持
- 与您公司将会受此升级影响的主要利益相关方协作，就估计的升级时间表达成共识
- 查看以往升级产生的问题

3.1.2 选择 MATLAB 和 Simulink 目标版本

在决定您的组织应升级到哪个 MATLAB 和 Simulink 版本之前，请先筛选和搜索 [发行说明](#) 了解特定版本的新功能，查阅 [Bug 报告](#) 中的已知问题，并阅读 [系统要求](#) 和 [支持的编译器](#)。另请查看 [按产品列出的平台可用性](#) 和 [选择用来运行 MATLAB 和 Simulink 产品的计算机](#)（如果适用于您组织的情况）。选择 MATLAB 和 Simulink 目标版本时，还应考虑“更新”。“更新”可通过单击 MATLAB 窗口右上角的铃铛图标来访问。只要有最新更新可用，并适用于您的组织或项目，MathWorks 就会建议您使用该更新。如果您使用第三方软件，请联系供应商，确认他们当前支持以及打算支持哪些版本的 MATLAB 和 Simulink 及其更新。同样重要的是，在新的 MATLAB 和 Simulink 版本和更新发布后，需要了解他们提供支持的时间，尤其是您的 MATLAB 和 Simulink 目标版本尚未发布的情况下。

请记住，在升级过程中，您可能想要在未来某一时间更改 MATLAB 和 Simulink 目标版本。您可能遇到意料之外的问题，或者 MathWorks 可能发布对您的组织有益的新功能。在升级过程中，与其过早地锁定某个特定版本，**不如根据自身具体情况灵活选择会带来最高投资回报的版本**。如果您的升级频率超过一年，则可通过变更目标版本随时获得这些好处，而不是等待几年才能使用新功能。

3.1.3 做好技术环境准备

下一步是确保您有权访问公司用于受该升级影响的工程的计算机、操作系统、MATLAB 和 Simulink、开发用软件和硬件以及第三方软件和硬件。为评估和迁移工作留出一个专用沙盒，因为在新版本中保存的模型很难在原始版本中打开。最后，新版本意味着您需要考虑其他设置，才能确保得到满意的结果。

沙盒代表着存储库中的一个或多个新的专用分支，可用于：

- 实现多个用户或团队之间的协作
- 共享工具或库的更新
- 轻松使用现有 CI 工作流
- 简化在升级过程中开发的全新或改进 CI 工作流的重用

务必能够扩展沙盒，以供在评估阶段的初始测试中使用，同时能够将该方法更大规模地应用于以后的迁移、测试和发布阶段。评估阶段的测试工作量要做到最少，重点应放在检查不容易解决的问题、了解警告或发现性能问题上。建议在之后的阶段（如迁移和测试阶段）进行更完整的测试。

3.1.4 初始测试

设置好技术环境并确定升级条件后，选择一个要从上一版本升级到目标版本的模型。最好选择这样一个大型模型：代表着您公司目前使用的大多数模型，包含常用建模模式，并且符合您的建模风格。

选择了模型后，执行下面的步骤来完成初始测试。虽然此过程通常需要执行多个手动步骤，但是发现升级到 MATLAB 和 Simulink 目标版本时出现的问题十分重要。“按原样”测试模型、自定义库和自定义工具时，只需执行最少量的修改，即可消除将会阻止进一步测试的错误消息。请记住，此评估阶段的目的是为了快速了解升级的可行性，同时找出需要解决的问题，以实现更大规模的成功迁移。

建议在评估阶段开展的活动：

1. 测试当前版本
 - a. 打开当前 MATLAB 和 Simulink 版本
 - b. 将自定义库或自定义工具添加到 MATLAB 路径
 - c. 在当前 MATLAB 和 Simulink 版本中打开 Simulink 模型或工程。接受任何推荐的 Simulink 模型配置设置。
 - d. 更新图
 - e. 模拟模型
 - f. 解决所有错误消息并确定警告的正确处理方法（如果有）
2. 测试目标版本
 - a. 打开 MATLAB 和 Simulink 目标版本
 - b. 将自定义库或自定义工具添加到 MATLAB 路径
 - c. 在 MATLAB 和 Simulink 目标版本中打开 Simulink 模型或工程
 - d. 对模型运行升级顾问
 - e. 记录警告消息和错误消息
 - i. 纠正错误消息
 - ii. 纠正警告消息（只在阻碍进一步测试的情况下）
 - iii. 记录并跟踪遇到的所有问题（其他模型也可能遇到同样的问题）
 - f. 在发布 MATLAB 和 Simulink 目标版本前解决问题

如果将选定模型升级到 MATLAB 和 Simulink 目标版本时遇到无法解决的问题，请在中间版本中测试该模型。如果是从旧版本升级的，则可能需要升级到 MATLAB 和 Simulink 中间版本。可能需要执行这个额外步骤，才能确保了解关于不兼容问题的警告和错误消息。如果升级的间隔时间超过解除警告和错误消息的时间范围，您可能会错过有助于查明问题根源的重要消息。虽然这种方法可以帮助解决或隔离问题，但是中间版本通常只是升级到新目标版本过程中的权宜之计。推荐的做法是选择 MATLAB 和 Simulink 的当前版本与目标版本中间的 MATLAB 和 Simulink 过渡版本。

3.1.5 回归测试

对用户通常执行的任务运行回归测试，以便估计这些任务在新的 MATLAB 和 Simulink 版本中执行所需的时间。尽可能在上一版本中运行相同的任务，并比较测试结果。建议您对当前使用的一些更大模型运行这些测试。这样做将有助于您在新版本发布之前发现潜在问题。

在评估阶段，回归测试是一项快速测试，旨在了解特定 MATLAB 和 Simulink 版本是否为要升级到的可行候选版本。理想情况下，无论您打算升级到哪个目标版本，都要对每一个预发行版本快速执行此回归测试。下表列出了在评估阶段评估特定 MATLAB 和 Simulink 版本时所应查看的重要项目。这份列表并非面面俱到，只列出有代表性的重要项目。在评估阶段，要评估的最重要方面是性能，以及是否存在无法解决的显著问题。关于回归测试的其他信息将在测试阶段进行介绍。

结果	理想	注释
仿真	仿真时没有错误	模型仿真时出错，但错误可以解决。
代码生成	生成代码时没有错误	模型生成代码时出错，但错误可以解决。
仿真时间	比上一版本快	仿真时间略微增加是可接受的，但始终应以较短的时间作为标准。
代码生成时间	比上一版本快	代码生成时间略微增加是可接受的，但始终应以较短的时间作为标准。

3.1.6 升级决策

在完成回归测试并根据升级条件评估测试结果后，您可以决定是继续升级过程的下一阶段，还是推迟升级以后再继续进行。

3.2 规划

规划阶段的目的是确定升级工程的总体范围和计划。其中应该涵盖升级过程的其余阶段，包括迁移、测试、发布和支持阶段。

3.2.1 升级工程管理

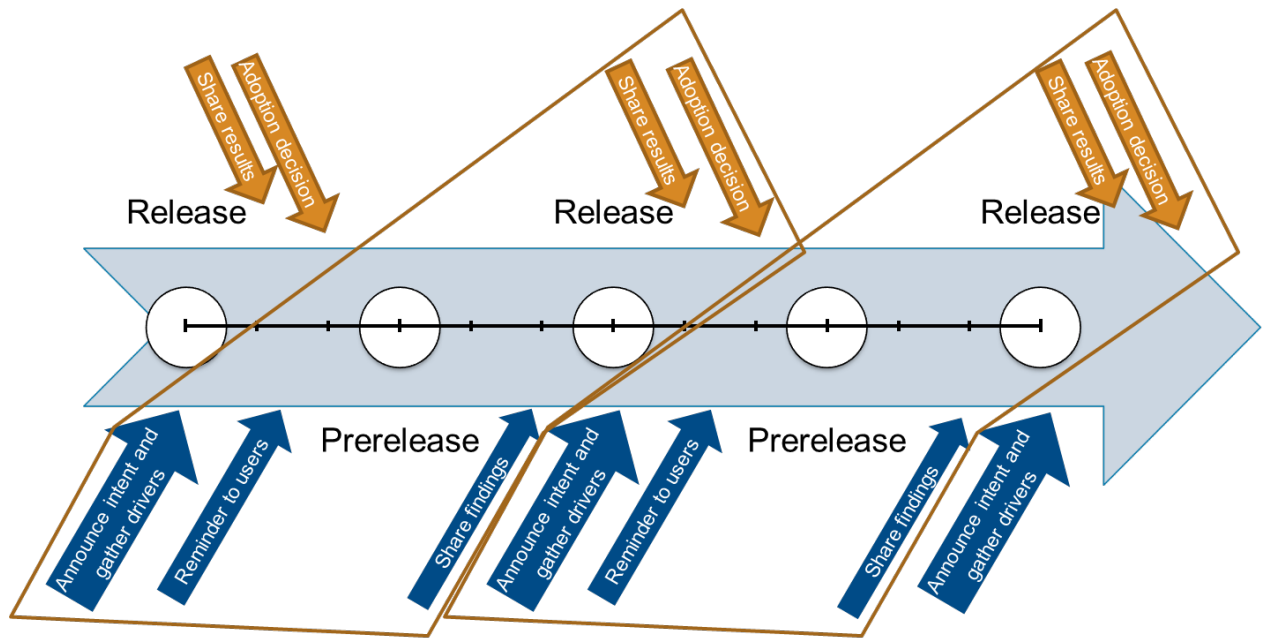
制订升级的总体计划，包括升级的目标日期以及 MATLAB 和 Simulink 目标版本。该计划要明确定义各阶段以及满足每阶段退出条件时所需执行的活动。此外，还要定义升级过程中涉及各个团队以及每位团队成员的角色和对他们的期望。典型角色包括升级负责人、工程团队、管理层发起人、信息技术 (IT) 部门、项目经理、工具团队、经理、第三方供应商和 MathWorks 顾问。请参见下面的泳道图，了解更多详情。

阶段	升级负责人	工程团队	管理层发起人	IT 部门	项目经理	工具团队
评估	领导	通知	通知	支持	通知	援助
规划	领导	援助	援助	援助	援助	援助
迁移	领导	援助	通知	支持	通知	援助
测试	领导	援助	通知	通知	通知	援助
发布	援助	领导	通知	通知	通知	领导
支持	领导	援助	通知	通知	通知	援助

沟通至关重要，贯穿于整个升级过程：应定期将计划、状况、决策和后续步骤告知利益相关方。就可能的升级进行沟通，越早越好。将能够推动升级的新功能通知用户群，并提供评估和迁移阶段的时间表。分享每个阶段的结果，包括哪些方面进展顺利，以及工具团队或用户发现的任何难题。持续提供有关进展情况和决策的更新以及时间表的任何变更。请注意，沟通应该持续进行，贯穿于整个升级过程。最后，为了获得最大影响力，可考虑每个利益相关方群体的需求，据此进行相应的沟通。

安排里程碑会议和进度汇报会议，向主要利益相关方提供有关升级的定期更新和进展情况报告。在这些会议中，让每位团队成员都能了解进度情况。凡在升级过程中遇到的问题，都应在这些会议中与团队分享。定期对每个阶段已完成的活动进行技术审核。下表和下图举例说明了与 MATLAB 和 Simulink 预发布和发布计划有关的沟通时间线。

沟通时间线		
日期	受众	主题
MATLAB 和 Simulink 预发布前 3 个月	用户群和管理层	近期预发行版测试提醒 升级目的，包括： <ul style="list-style-type: none"> • 已知升级推动因素清单 • 已知升级障碍清单 预发行版测试志愿者邀请
MATLAB 和 Simulink 预发布前 1 个月	用户群	近期预发行版测试提醒 近期预发行版测试时间安排 已知推动因素/障碍的变更
MATLAB 和 Simulink 预发布	活跃的预发行版测试参与者	预发行版测试进度和时间安排分享
MATLAB 和 Simulink 预发布后 1 个月	用户群	预发行版测试结果分享，包括取得的成功和遇到的困难 其余预发行版测试计划分享（如果有）
MATLAB 和 Simulink 发布前 1 个月	用户群和管理层	预发行版测试结果分享 升级评估计划分享
MATLAB 和 Simulink 发布	活跃的升级评估参与者	升级评估进度和时间安排分享
MATLAB 和 Simulink 发布后 1 个月	用户群	升级评估结果分享，包括取得的成功和遇到的困难 其余升级评估计划分享（如果有）
MATLAB 和 Simulink 发布后 2 个月	用户群和管理层	准备就绪后的升级决策分享，包括计划和时间安排



与 MathWorks 发布相关的沟通时间线

3.2.2 创建业务案例

在升级到新版本之前创建业务案例，将有助于阐明升级的好处和相关成本。升级的好处可能包括通过 MATLAB 和 Simulink 产品中的新功能可改进工作流。如果当前版本的 MATLAB 和 Simulink 不支持新操作系统或第三方软件，则操作系统或第三方软件升级可能也会推动创建升级的业务案例。

3.2.3 预先设定目标

务必限定升级过程的范围。尝试一次执行太多操作会增加延迟的风险，甚至导致升级工程失败。

尽可能“按原样”升级模型，而不引入新功能。在引入新的自定义工具以及新的 MATLAB 和 Simulink 版本时引入新功能，会使升级过程复杂化，也会使验证模型变得难上加难。等到模型已完全升级并在新的 MATLAB 和 Simulink 版本中得到验证后再引入新功能。

在升级过程中，着重对您组织的典型工作流进行升级测试，例如更新、仿真以及从模型生成代码。在模型和自定义工具升级之后，需要由负责开发这些模型的工程师在新版本中对其进行验证。

如若可能，最好将自定义工具功能替换为 Simulink 的内置功能，以此作为您的目标之一。例如，您可能打算将自定义库模块替换为具有相同功能的新 Simulink 模块。设定一个目标，移除不再适用于新版本的建模指导规范，并为您打算使用的 Simulink 中的新功能添加新规范。考虑将增加对自定义工具文档和培训材料的评估视为目标。

3.3 迁移

在迁移阶段，以迭代方式将模型和自定义工具升级到新版本。在每次迭代中，通常都需要执行自动和手动步骤来升级模型、脚本、用户界面、模板、第三方工具的接口和其他应用。这种迭代方法为升级过程的每个阶段都奠定了坚实的基础，有助于建立对新版本的信心，同时减少整个组织和最终用户面临的风险和工作量。

在迁移阶段，模型只是升级到新版本，尚未进行测试。模型在测试阶段进行测试。

3.3.1 设置环境

在将模型迁移到新版本之前，您需要设置建模环境。此步骤可能包括访问自定义工具，设置 MATLAB 路径，设置编译器，以及启动工程，具体视您的环境而定。在新的 MATLAB 和 Simulink 版本中设置环境时，务必尽

可能少做更改。推迟对自定义工具和 MATLAB 脚本的更改，直到升级过程的自定义工具迁移阶段开始。扩展评估阶段所用的沙盒方法，以供在多个工程中使用。

在迁移阶段尽早创建标准配置设置通常是有利的：可以最大限度地避免重复工作，并减少后期集成问题。初始测试提供的建议设置可作为好的起点，但要审核每个代码目标的任何新设置或修改后的设置，以便确保这些设置是适合的。在启动迁移过程时，将这些经过审核的设置应用于模型。

3.3.2 确定要迁移的模型

在升级过程的评估阶段，您选择并升级了一个模型，然后在新版本中对其进行了测试，以评估升级过程并发现潜在问题。在迁移阶段，要将测试扩展到更多模型。如果您无法测试所有可用模型，则建议您从各个团队中选择具有各种大小和不同建模风格的模型。在升级过程的这一阶段，需要选择边缘情形和不常见的建模模式。如果您知道特定的组或人员创建了不符合规范的模型，请将其模型纳入这个阶段的测试中。测试边缘情形有助于您在升级过程中发现意外问题并加以解决，而不是在发布新的 MATLAB 和 Simulink 版本后才来解决问题。这样，您就有更多时间解决任何问题，而不会影响生产最后期限。下面是模型的典型高级示例，包括：

- 嵌套库
- 模型引用模块
- 可配置子系统或变体
- 模型引用模块及嵌套库

3.3.3 初始迁移

初始迁移阶段的目的是测试您的典型工作流，具体方式包括更新图、运行仿真和生成代码。使用以下活动列表作为完成此阶段的指南。

在旧版本中：

- 打开工程
- 成功更新图、运行仿真并为模型生成代码
- 使用 Simulink 中的诊断查看器记录所有警告消息

在新版本中：

- 使用工程在 MATLAB 中设置路径和环境
- 对选定模型运行升级顾问：
 - 查看报告
 - 应用所需的最少建议修复（对于不妨碍迁移选定模型的任何修复，可在升级过程的后期应用）
 - 成功更新图、运行仿真并为模型生成代码
 - 使用诊断查看器记录错误和警告消息
 - 解决错误
 - 比较警告消息，以确定这些消息是新版本独有的还是存在于旧版本中
 - 评估警告消息的严重性
 - 评估解决警告消息的方法是对模型进行更新还是采用新功能
 - 将所有其他问题都记录在升级问题列表中（例如，记录有问题未被升级顾问检测到的任何自定义工具）

在初始迁移阶段，仅对 MATLAB 文件、数据字典、自定义模块库和自定义工具做出必要的最低限度更改，以确保模型能够更新。模型所依赖的任何文件都应添加到 MATLAB 路径。新功能、优化项、配置集和 S-Function API 更改应在自定义工具迁移子阶段的后期完成。

对于数据字典，加载与模型或数据字典相关的数据，解决加载数据时或在数据加载后成功更新模型时遇到的任何问题。

对于自定义模块库，升级顾问会自动更新模型中已有的 Simulink 自定义库模块。升级顾问会更新从 Simulink 的内置模块创建的自定义模块（如果自定义模块库中有这些模块并且库链接处于活动状态），更新模型或自定义库中已有的自定义 S-Function 模块，并为升级顾问的每次检查生成结果。查看结果后，您可以让升级顾问自动对模块进行更改。升级顾问不会更新封装，因此您必须手动更新。

在极少数情况下，从比目标版本旧得多的某个版本升级时，升级顾问可能无法更新您的模型，也无法找出问题的根源。大多数此类情况是由于 MATLAB 和 Simulink 的自定义项或 Simulink 预期或意料之外的边缘情形建模模式所造成的。在这些情况下，您可能必须先升级到 MATLAB 和 Simulink 的中间版本，以便隔离和调试问题，等这些问题得到解决后再升级到目标版本。请参阅本文档中的“初始测试”部分，了解关于如何在中间版本中调试问题的更多信息。

3.3.4 自定义工具迁移

除了迁移自定义工具，还需要寻找机会，通过采用内置功能来淘汰自定义工具功能。

3.3.4.1 升级自定义工具

在升级到新的 MATLAB 和 Simulink 版本时，您可能需要更新自定义存储类、自定义系统目标文件、MATLAB Report 和 Simulink Report Generator 设置文件、自定义模型顾问检查、建模指导规范和任何自定义 MATLAB 脚本。如果您有模板或演示模型，则可以考虑采用 Simulink 中对您有益的新功能来修改它们。这是将“配置参数”设置迁移到 MathWorks 建议的设置的好时机。这可能也是在整个组织内将常用“配置参数”设置保持一致的好机会。

建模指导规范需要针对新版本进行更新。不妨考虑移除不再适用的建模指导规范，并为您打算使用的 Simulink 中的新功能添加更新的规范。

在开发和迁移您的自定义工具时，建议您创建一个设计文档，详细描述这些工具的需求和预期功能。对于您的自定义工具，遵循与您的生产软件开发流程相似的开发流程。升级到新的 MATLAB 和 Simulink 版本时，创建测试计划和测试用例，然后使用它们测试您的自定义工具。如有可能，自动执行测试计划，对每个预发行版以及 MATLAB 和 Simulink 目标版本运行自动测试。升级到新的 MATLAB 和 Simulink 版本后，按需更新设计文档、测试计划和测试用例。

请查阅以下各节，了解需要升级的任何文件。然后，查看 Simulink 中的新功能、优化项和文档化的 API，找机会使用 Simulink 的内置功能淘汰自定义工具功能。

MATLAB 脚本和文件

升级成功更新、运行仿真并从模型生成代码而无错误所需的 MATLAB 脚本和文件。示例包括：

与 MATLAB 交互的 MATLAB 脚本	与 Simulink 交互的 MATLAB 脚本
<ul style="list-style-type: none">• MATLAB 工程• 环境设置 MATLAB 脚本• 编译器设置脚本• 数据分析脚本• 用户构建的 GUI	<ul style="list-style-type: none">• 加载参数的脚本• 加载配置集脚本• 设置配置选项的脚本• 自定义菜单的脚本• 用户界面脚本• 模块回调中的脚本• 封装内的脚本

如果您已使用 GUIDE 开发了 GUI，请将其转换到 App 设计工具。MathWorks 建议您使用工程，而不使用模型回调。如果仍需要使用模型回调（例如预加载和后加载脚本），请确保打开回调追踪，以帮助调试由回调引起的问题。

Simulink 自定义和配置

升级成功更新、运行仿真并从模型生成代码而无错误可能需要的任何 Simulink 文件。示例包括：

仿真	代码生成	验证和确认
<ul style="list-style-type: none"> • 仿真数据 • 记录自定义项 • 可视化自定义项 • 报告生成设置文件 • 数据比较工具 	<ul style="list-style-type: none"> • 系统目标文件 (STF) • 模板联编文件 (TMF) • 目标语言编译器 (TLC) 文件 • 代码生成模板 (CGT) 文件 • 代码替换库 (CRL) 文件 • SIL/PIL 环境 	<ul style="list-style-type: none"> • 单元测试自定义项 • 测试数据 • 报告 • 信号和测试编写工具 • 自定义模型顾问检查 • Simulink Design Verifier 自定义项 • Polyspace 静态分析工具配置 • SIL/PIL 环境

配置设置

升级时总是要对配置设置进行更改：新设置、修改的设置和弃用的设置。这些更改需要进行确定、了解、记录和应用。在较新的版本中，模型的配置设置可从模型资源管理器导出。有关更多详情，请参见 Simulink 文档中的[保存配置集](#)。对于没有此功能的较旧版本，可以编写自定义脚本来导出这些设置。

更新配置设置应该在迁移阶段的早期完成，以确保设置接受充分测试，每个人都从更新中受益，并且模型适用于集成。建议落实采用配置设置更改的流程。

数据字典

加载数据字典并解决加载或成功更新模型时遇到的任何问题。将 MATLAB 脚本和基于 ASCII 的数据字典升级到 Simulink 数据字典和 Embedded Coder 字典。有关更多信息，请参阅 [Simulink 数据字典](#) 和 [Embedded Coder 字典](#) 文档。

自定义模块库

将自定义模块库升级到新版本时，利用这个机会将自定义模块迁移到 Simulink 的内置模块或功能。您可以使用[发行说明](#)筛选和搜索新的模块和功能。如果有新功能或增强功能可用，这也是改进和优化自定义模块的好机会。

具有相同功能的新增内置模块可能与当前自定义模块的对话框参数值并不相同。在这种情况下，您可以将模块对话框参数从旧参数字段复制到新参数字段。此步骤可以手动完成，也可以通过 MATLAB 脚本自动完成。如果您打算逐步停用不想让团队继续使用的模块，则需要创建位于 MATLAB 路径上的旧库，但需要标注清楚，以让用户不再访问该库。在新的 MATLAB 和 Simulink 版本中更新您的模型并运行升级 MATLAB 脚本之后，

您可以移除此库。对您的库运行升级顾问。如果 S-Function 进行了任何更改，而且您打算支持更多平台，则可能需要重新编译您的 S-Function。

自定义模型顾问设置

模型顾问检查和设置往往随每个版本而变化。升级让您有机会在查看当前模型顾问设置的同时，对新检查进行调查。任何自定义检查也都应进行测试，并在必要时予以更新。应该预先开发一组符合您组织的需要和规范且特定于版本的检查，以便减少各团队花费的时间。

3.3.4.2 自定义工具测试

使用测试计划和测试用例对您的自定义工具进行测试，以验证自定义工具在新的 MATLAB 和 Simulink 版本中能否按预期方式执行。接下来，开始针对您的模型测试升级过程。最好是在多台 PC 上测试自定义工具和升级过程，这些 PC 安装了您的组织支持的各种操作系统。如果您还没有为自定义工具开发自动化回归测试，现在就可以创建这些测试，以便您在每次更新和升级自定义工具时都可以重用它们。

3.3.4.3 第三方工具测试

在发布新的 MATLAB 和 Simulink 版本之前，建议您测试所有第三方工具与新的 MATLAB 和 Simulink 版本的兼容性。在有些情况下，第三方可能不会立即发布与新版 MATLAB 和 Simulink 兼容的新版本软件。与您的供应商确认发布时间，并做出相应的安排。示例包括：

升级时要考虑的第三方产品

- 需求管理工具（例如 IBM Rational DOORS）
- 标定支持工具
- 与 MATLAB 和 Simulink 产品集成的配置管理工具
- 与 MATLAB 和 Simulink 产品集成的变更管理工具
- 编写工具/系统工程工具
- 协同仿真工具
- IDE
- 编译器
- 硬件

3.3.5 自动迁移

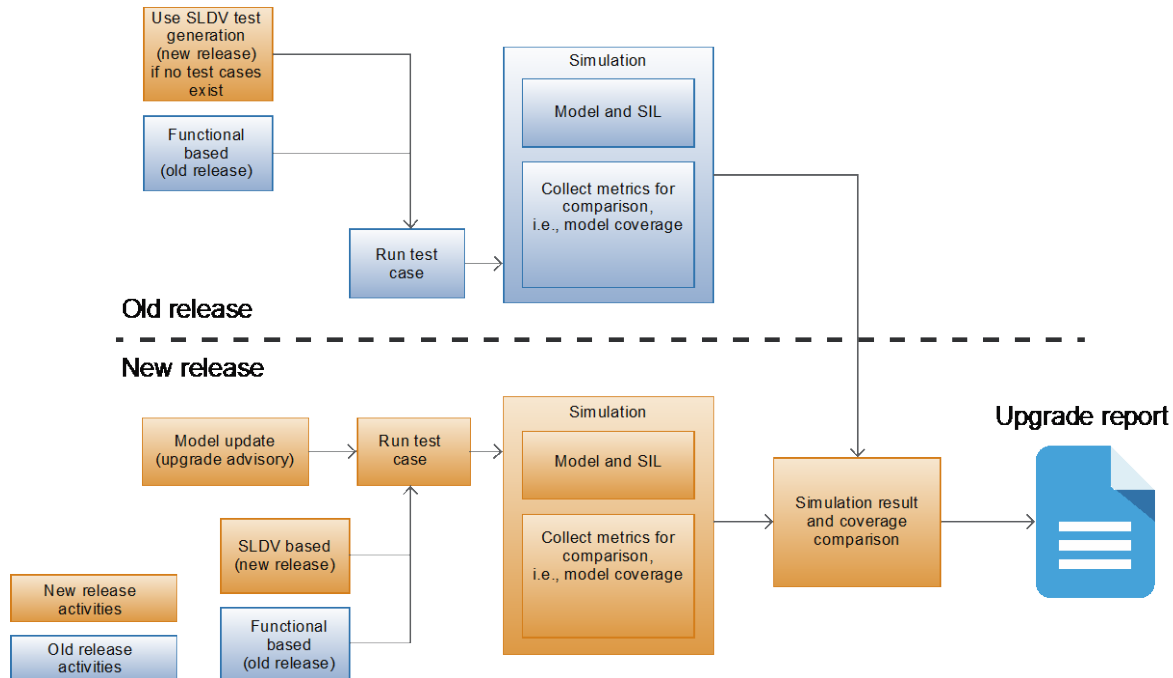
在执行完初始迁移和自定义工具迁移后，下一步是尽可能多地自动执行升级过程。

3.3.5.1 升级自动化工具

建议您自动执行模型和自定义工具升级过程。便捷的方法是创建单个主升级 MATLAB 脚本，用来调用所有其他升级脚本，这样，您只需运行一个 MATLAB 脚本，即可升级任何模型。此过程应尽可能自动化。此 MATLAB 脚本应调用各种函数，以自动执行下列任务：

- 升级顾问
- 自定义库内容
- 配置参数
- 工作区对象
- 升级后的回归测试
- 自定义工具升级过程

这些自动化工具会生成一个报告，详细列明每个阶段遇到的错误和警告。下面是一个可使用 Simulink Design Verifier (SLDV) 和 SIL 仿真完成此任务的工作流示例。此工作流可通过完全基于脚本的方法来实现，也可通过 UI 来实现。该 UI 可自动执行工作流并引导用户完成升级过程。下图举例说了可用来将 Simulink 模型从一个版本自动升级到另一个版本的升级过程。



升级过程示例

3.3.5.2 测试自动化工具

在组织内部发布新的 MATLAB 和 Simulink 版本之前，对所有模型运行包含自定义工具的升级脚本。如果无法升级您的所有模型，请从各个组中选择具有不同建模风格和模型大小的模型。升级了模型后，您应该更新图、运行仿真并生成代码。查看警告消息并记录遇到的任何错误。

自动执行此测试过程是个好做法，这样您便可以批量模式对一组模型运行这些测试，并自动记录问题。在发布新版 MATLAB 和 Simulink 之前，您可能需要找到解决方法或编写 MATLAB 脚本来解决其中一些问题。此外，也可以在持续集成服务器上实现此批量模式。通过使用 CI 服务器，可以池化计算资源，并在组织准备好开始升级过程后自动执行相关活动。

3.3.5.3 迁移剩余模型

到目前为止，您已经升级自定义工具和自定义模块库，并且开发了升级自动化工具。下一步是升级剩余模型。此升级活动可以在本地计算机上完成，也可以扩展到 CI 服务器上自动完成升级过程。

在模型升级时，记录出现的问题。问题可能出在模型、自动化工具、自定义模块库或自定义工具上。按照迁移阶段中推荐的步骤解决问题，然后再次运行自动化工具。以迭代方式重复这些步骤，直到解决所有已知问题。此迭代循环操作也可以在 CI 服务器上执行。在这种情况下，CI 服务器将会执行升级活动。

3.4 测试

测试阶段的目标是，确保在可接受的限度内，新版 MATLAB 和 Simulink 中的模型和代码与上一版本中的模型和代码在功能和数值上相同。

3.4.1 回归测试

对用户通常执行的任务运行回归测试，以便估计这些任务在新的 MATLAB 和 Simulink 版本中执行所需的时间。如有可能，在上一版本中运行相同的任务，并比较测试结果。建议您对当前使用的一些较大模型运行这些测试。

这样做将有助于您在新版本发布之前发现潜在问题。推迟升级到新的 MATLAB 和 Simulink 版本比撤消发布新版本或自定义工具要容易。测试对于最大限度减少升级对生产项目的影响至关重要，因为测试让您能够发现问题，然后在实施升级之前根据需要找到解决方案或解决办法。

回归测试应以当前结果为基线，通过各种检查进行全面彻底的评估。一致性检查有助于识别必须了解的变更，并最终决定任何升级成功（或失败）。下表列出了在评估升级时所应查看的重要项目。这份列表并非面面俱到，只列出有代表性的重要项目。请注意，功能和性能应兼顾考虑。很显然，如果对这些方面进行测试（最好自动执行），则更易于执行此过程。在升级过程中执行的测试如有任何改进，都可以并且应该纳入开发工作流程中。最后，测试计划应针对不同阶段进行相应的调整。

结果	理想	可接受	注释
仿真结果	完全相同的结果	在容差范围内	记录的信号应包括所有模型输出。如果认为其他信号是重要的中间结果，也可以记录它们。
覆盖率指标	完全相同的结果	通过覆盖率工具变更进行解释	仿真结果如果来自覆盖率达到近 100% 的测试，则有助于增强信心。 注意：测试覆盖率工具的变更可能会变更各版本的覆盖率指标。务必了解这些情况，并确保对自己的测试有信心。
代码生成	等效	无错误	SIL 测试可让此处的“可接受”结果令人满意。
SIL 测试	完全相同的结果	在容差范围内	记录的信号应包括所有模型输出。如果认为其他信号是重要的中间结果，也可以记录它们。
仿真时间	比上一版本快	在容差范围内	对比上一版本与新版本，仿真时间略有增加，这是可以接受的。
代码生成时间	比上一版本快	在容差范围内	对比上一版本与新版本，代码生成时间略有增加，这是可以接受的。
第三方兼容性	等效	等效	第三方软件、工具和可能的硬件必须能与新版 MATLAB 协同工作。
错误和警告	类似结果		任何新错误都应得到解决。任何新警告都应进行调查和了解。

3.4.2 等效性测试

建议您运行等效性测试，将上一版本中每个模型和代码的仿真结果与新版本中的相应模型和代码进行比较。在升级过程示例中，3.3.5.1 小节中提到的过程包含等效性测试阶段。重要的是，您只有在完成该等效性测试后，才能向模型中添加任何新的 MATLAB 和 Simulink 功能。否则，两个版本中的模型和生成的代码在功能上可能并不等效。Simulink Test 支持[在多个版本的 MATLAB 中运行测试](#)、[等效性测试](#)和[跨版本代码测试](#)。可以使用 Simulink Design Verifier 生成测试用例，从而使等效性测试的测试覆盖率达到 100%。

3.4.3 持续集成测试

建议使用持续集成环境自动升级、集成、仿真、生成代码、测试等效性和生成测试报告。对于大型组织，这应该是一个长期目标。与手动转换和测试每个模型相比，使用持续集成将会使升级过程更加无缝，成本效益更高，并且对最终用户的影响也最小。使用 CI 服务器可以大大减少升级过程中所需的手动工程工作量。

3.3.5.2 小节中列出的相同 workflow 可用于所有模型和工具升级。CI 服务器应该实现为了升级模型和 MATLAB 文件而已建立的自定义流程。在 CI 服务器上对每个组件重复这些任务，直到尝试过升级每个模型和 MATLAB 文件。

3.4.4 beta 测试

在测试的最后阶段，由一小组选定用户对自定义工具和模型升级过程进行 beta 测试。beta 测试人员包括经验丰富的 MATLAB 和 Simulink 用户，他们深入了解相关的 Simulink 模型，能够提供有用的反馈来改进流程和工具。这群用户应该开始使用新的 MATLAB 和 Simulink 版本、自定义工具以及任何必要的第三方软件工具处理日常工作。

3.4.5 测试更新

测试 MATLAB 和 Simulink 更新，并将其纳入您组织的测试计划。只要有最新更新可用，并适用于您的组织或项目，MathWorks 就会建议您使用该更新。在组织内发布 MATLAB 和 Simulink 目标版本之前，确认哪个更新适用于该版本，然后在使用最新更新的情况下重复以前执行的回归测试。理想情况下，您的组织已经使用 CI 服务器自动执行此类测试，这是用来评估更新的快速测试。否则，建议您执行本白皮书的评估阶段中所述的回归测试，找出在升级过程的后期引入更新可能会引起的问题。更新包含 Bug 修复，因此，迁移到最新更新应该不会引入任何新问题。如果您的组织确实遇到问题，可通过[在 MathWorks 网站上创建服务请求](#)，向 MathWorks 支持部门报告该问题，以便考虑在未来更新中将其修复。在您升级到新版本后，无论 MathWorks 何时发布新的更新，您都可以在 CI 服务器上快速重复此回归测试，以评估发布新更新对您组织的影响。

3.5 发布

发布阶段的目的是发布新的 MATLAB 和 Simulink 版本、自定义工具和自动化工具，并向用户提供培训。

3.5.1 培训

建议您在升级过程中为用户提供两种类型的培训课程。第一种类型是研讨会或讨论会，概要介绍目前随同升级的 MATLAB 和 Simulink 版本一起提供的一些主要新功能，重点介绍与您组织最相关的功能和优点。第二种类型侧重于升级过程、建议的特定问题解决办法、新指导规范，以及适用于您的自定义工具的新功能。

3.5.2 发布

完成测试之后，您随时可以向用户发布自定义工具。好的做法是在自定义工具软件的每个版本中包含发行说明。另一个不错的做法是在发布前通过对文件进行 *P 编码* 来混淆处理 MATLAB 代码，即使这些工具仅在内部使用。这种做法建议工程师不要在不通知他人的情况下自行修复问题，因为这可能导致各个团队使用不同版本的自定义工具，在将来可能造成升级问题。将自定义工具或安装程序放在组织内所有用户都能访问的位置。在通知用户新版本的自定义工具以及 MATLAB 和 Simulink 可用时，同时提供安装和升级说明。

建议工程师升级他们自己的 Simulink 模型，而不是由单独的小组执行迁移。开发模型的工程师具备执行验证所需的专业知识。他们还知道自己面临的生产最后期限，以及需要在未来版本的生产软件中修改模型的哪些部分。如果他们决定要将自己的模型升级到新版本，则需要一致同意使用其他主要利益相关方的已验证模型。这些利益相关方可能包括该流程下游将受影响的其他工程组。这是为了确保迁移到新的 MATLAB 和 Simulink 版本不会造成生产计划的延迟。所有新的建模工程都应该需要使用新的 MATLAB 和 Simulink 版本。最好是为每个人设定将其模型迁移到新 MATLAB 和 Simulink 版本的最后期限。在他们完成该迁移之前，您可能需要支持多个 MATLAB 和 Simulink 版本。

广泛测试和创建 MATLAB 脚本的目的是尽可能多地自动执行该过程。如果出于某种原因，工程师无法升级他们自己的模型，则您需要模型测试覆盖率达到期望水平（建议 100%）的测试用例。如有可能，以自动化方式在 SIL 环境中测试模型和生成的代码，验证仿真和代码生成输出是否与上一 MATLAB 和 Simulink 版本的输出相符。

3.6 支持

支持阶段的目的是继续执行升级后活动，以最大限度减少下一次升级的工作，并向用户提供持续支持。

3.6.1 升级后活动

在完成升级后，您可以执行多项升级后活动，使下一次升级更容易。如果您在自定义工具迁移阶段尚未创建设计文档，以详细说明自定义工具的需求和功能，则应在支持阶段完成这项任务。

利用这个机会为您的组织中使用的自定义工具和典型工作流创建测试计划和测试用例。然后，在升级到新的 MATLAB 和 Simulink 版本时，您便可使用这些测试计划和测试用例来测试您的自定义工具了。测试用例必须包括每个测试的预期输出。如果制定了测试计划并且有测试用例，下次升级时将更容易自动执行测试。您可以对每个预发行版运行自动回归测试，并向 MathWorks 提供反馈，即使您不打算升级到该版本。此反馈将会帮助 MathWorks 在您迁移到下一 MATLAB 和 Simulink 版本之前改进产品或修复您遇到的任何问题。当然，您也可以对有意部署的新的 MATLAB 和 Simulink 版本以及 MathWorks 发布的任何更新运行自动化回归测试。

在自定义工具测试自动执行完后，下一步是自动执行模型迁移和模型验证流程，以便您可在 CI 环境中以批量模式对您的模型运行测试，并自动生成问题相关报告。模型和生成的代码需要在 SIL 环境中进行测试，并达到期望的测试覆盖率水平，才能验证仿真和生成的代码输出是否与上一 MATLAB 和 Simulink 版本的输出相符。这种类型的 SIL 测试也可以自动执行，旨在减少下一次升级的工作量。下面是可以在 CI 环境中运行的测试的不完全列表：

- 验证两个 MATLAB 和 Simulink 版本的自定义工具输出是否相符
- 将一组模型迁移到新的 MATLAB 和 Simulink 版本
- 在新的 MATLAB 和 Simulink 版本中运行模型更新、运行仿真并基于一组模型生成代码
- 区别两个 MATLAB 和 Simulink 版本的生成代码
- 比较两个 MATLAB 和 Simulink 版本的仿真结果
- 在同一个 MATLAB 和 Simulink 版本中比较仿真结果与生成的代码结果
- 比较两个 MATLAB 和 Simulink 版本的生成代码结果与仿真结果

升级回顾可以揭示一些值得汲取的教训。确定哪些架构、建模风格和定制方面的教训适用于一般升级，哪些教训与特定版本相关，这是需要一些精力和经验的。创建存储库或 wiki 来记录这些教训，并提供特定于版本的教训的链接。在特定于版本的相应页面上记录任何新教训，并定期回顾这些页面，查找（并推进）更普遍的事项。

4 可持续性：持续升级理念

MathWorks 建议采用持续升级理念。持续执行升级活动可确保升级越来越轻松。为了便于采用这种理念，可以考虑利用预发行版测试和行业模型测试，以及各类 MathWorks 研讨会、网络研讨会和会议。

4.1 预发行版测试

如有可能，针对每个预发行版测试您的模型和自定义工具，并通过[在 MathWorks 网站上创建服务请求](#)，向 MathWorks 支持部门报告问题。由于您的当前版本与未来版本之间引入了很多新功能，最好提前测试您的模型，而不要等到在您打算升级到的目标版本上才进行测试。如果您每六个月测试一次预发行版，您提供的早期测试反馈有助于 MathWorks 在您下次升级之前已经修复您遇到的问题。测试每个预发行版不仅使您能够比阅读发行说明更透彻地评估该版本，而且可以最大限度减少升级到新的 MATLAB 和 Simulink 版本所需的工作量。此外，还便于您了解新功能，并为企业决策者提供有关恰当升级时机的指引。

4.2 行业模型测试

可以考虑将您的模型提交进行行业模型测试。下面的内容摘自行业模型测试信息表：

行业模型测试 (IMT) 是 MathWorks 的战略质量计划的一部分。IMT 体系旨在确定、隔离并阻止客户在使用 MathWorks 软件时遇到的退化问题。作为该体系的一部分，MathWorks 在安全环境中测试真实客户模型。此环境与 MathWorks 软件构建、测试和发布过程相集成。将您的模型纳入 IMT 流程可以大大降低版本不兼容的几率。

如果 IMT 不适合您，您可能需要在内部设计一种类似方案。在 CI 环境中实现测试自动化是第一步。通过增加面向特定 MATLAB 和 Simulink 版本的灵活性，您至少可对预发行版和正式发行版本模拟 MathWorks 行业模型测试。这样，您便可以尽早知晓潜在问题，或建立对新版本的初步信心。

4.3 研讨会、网络研讨会和各种会议

参加 MathWorks 举办的研讨会、网络研讨会和各种会议，可帮助您及时了解新功能，并对您的组织应采用哪个 MATLAB 和 Simulink 版本做出明智的决策。

5 MathWorks 支持

MathWorks 针对升级过程提供了各种自助支持选项。其中一些自助选项包括[在线产品文档](#)、[发行说明](#)（可进行筛选以搜索新功能和兼容性问题）、[示例](#)（说明如何使用新功能）、[MATLAB Answers](#)、[Bug 报告](#)和活跃的用户社区 ([MATLAB Central](#))。如果需要其他支持，请联系[支持部门](#)询问关于产品的特定问题，联系[培训部门](#)了解产品培训，或联系[咨询服务部门](#)获得升级协助。

6 附录

检查清单内容
评估：了解升级所带来的影响
<input type="checkbox"/> 确定升级的触发条件
<input type="checkbox"/> 评估现状，包括可能的不兼容问题
<input type="checkbox"/> 选择 MATLAB 和 Simulink 目标版本
<input type="checkbox"/> 在新版本中使用典型工作流测试一些模型
<input type="checkbox"/> 对用户通常执行的任务运行回归测试
<input type="checkbox"/> 创建升级评估报告
<input type="checkbox"/> “可行/不可行”决策
规划：确定升级工程的总体范围
<input type="checkbox"/> 创建业务案例，阐述投资回报
<input type="checkbox"/> 通过了解受影响的模型、工程和组织来确定范围
<input type="checkbox"/> 制订详细计划，其中包括时间安排、资源和主要利益相关方
迁移：以迭代方式将工程转换到新版本
<input type="checkbox"/> 初始迁移：迁移一小部分模型并测试您的典型工作流
<input type="checkbox"/> 自定义工具迁移：将自定义工具功能替换为 MATLAB 和 Simulink 的内置功能
<input type="checkbox"/> 自动迁移：将测试扩展到具有不同建模风格的模型，并使用自动化工具升级其余的模型
测试：验证升级的模型
<input type="checkbox"/> 对用户通常执行的任务运行回归测试
<input type="checkbox"/> 使用 Simulink 产品在桌面端运行开环、MIL、SIL 和 PIL 仿真，以进行验证和确认
<input type="checkbox"/> 在实验室开展 HIL、快速原型构建和目标系统快速原型构建测试
<input type="checkbox"/> 测试第三方工具
<input type="checkbox"/> 在新版本中验证模型的功能等效性
<input type="checkbox"/> 邀请选定用户完成新版本的 beta 测试
发布：发布新的 MATLAB 版本和自定义工具
<input type="checkbox"/> 安排和开设培训课程
<input type="checkbox"/> 向用户发布新的 MATLAB 版本和自定义工具
<input type="checkbox"/> 建议模型专家转换他们的模型
<input type="checkbox"/> 为每个人设定迁移最后完成期限
支持：自动执行升级过程并报告问题
<input type="checkbox"/> 考虑使用 CI 服务器自动执行升级过程
<input type="checkbox"/> 回顾升级过程并记录从中汲取的教训
<input type="checkbox"/> 测试每个预发行版并向 MathWorks 支持部门报告问题