



ENTERPRISE ARCHITECT

用户指南系列

建模团队

Author: Sparx Systems

Date: 13/11/2024

Version: 17.0

创建于  **ENTERPRISE
ARCHITECT**

目录

建模团队	3
典型的项目角色	6
典型任务总结	7
Enterprise Architect	9
业务分析师	10
系统工程师	11
软件架构师	12
软件工程师	13
开发者	14
项目经理	16
测试人员	17
实施经理	18
技术开发人员	19
数据库开发人员	20

建模团队

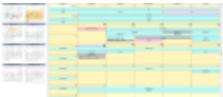


Enterprise Architect已作为团队建模平台从头开始，广泛支持在同一项目上共享信息、想法的人群共享，团队支持的特征包括基线和可重复的模型、版本控制和资产。服务，可保护团队环境中宝贵的建模资产，以及论坛、库窗口和甘特图等工具，以促进项目成员之间的协作。基于角色的安全系统还旨在鼓励协作，使团队成员能够一起工作，确信在访问或更改模型数据时不会发生冲突。

部署选项A选择将支持任何团队开发环境，允许人们在高度分布式的环境中集中或远程工作。企业政策和标准也可以通过使用 workflow 脚本构建到模型中。Enterprise Architect A免费 精简版”版本为团队成员提供 仅查看”其模型的访问权限，同时也允许他们以多种格式生成高质量的企业文档，以便与建模平台之外的人进行交流。

概述

功能	描述
 <p>团队发展</p>	设置协作建模环境，利用安全性、工作流和共享参考数据，如本主题其余部分所述。
 <p>正式模型评论</p>	A简单而有效的机制，用于实时捕获与特定事件一致的模型部分评论。通常，项目经理或协调员将创建一个审阅元素，专门讨论一个或多个项目相或其他类别的审阅，在定义的时期内。
 <p>项目管理</p>	探索在Enterprise Architect中管理项目和团队的一些方法。
 <p>项目资源</p>	跟踪和管理项目中的人员和资源。
 <p>词汇表</p>	在您的不同团队之间定义一个共同的词汇表，确保共同理解。
<p>任务分配</p>	在甘视图图中分配和跟踪团队任务。

	
<p>个人任务</p> 	在项目中记录和管理您的个人任务。
<p>模型邮件</p> 	在“聊天和邮件”窗口中使用模型邮件，通过嵌入在模型中的内部电子邮件系统与您的团队进行安全通信。
<p>项目日历</p> 	以日历格式跟踪资源的部署、任务的时间范围以及即将发生的项目事件，例如会议和里程碑。
<p>用例估计</p> 	形成对系统复杂性的估计和实施模型所需工作量的指示。
<p>图书馆窗口</p> 	提供对基于团队的文档库的访问，以记录和讨论项目的开发和进度。

使项目数据在分布式环境中可用

Enterprise Architect提供了一套多样化的功能，专为在基于团队的分布式开发环境中共享项目而设计；例如：基于云的解决方案、模型库的网络部署、复制和Native/XMI导入/导出。

对模型应用安全性

用户安全是一种改进协作设计和开发的方法，它可以防止未被指定为模型作者的用户同时编辑和无意更改模型。

使用内部讨论论坛

Discussions功能提供了多种机制来支持您的开发团队社区，通常用于讨论模型在整个项目中的开发和进展，或者专门用于讨论模型中的各个元素，讨论成为每个元素的组成部分。

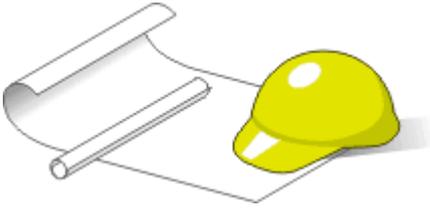
建造公司政策和项目开发指南构建到项目中

您可以创建工作流脚本，通过根据模型本身内的政策和程序验证工作，为应用公司政策和加强项目开发指南提供强大的方法。

标准化和重用项目数据

您可以从相同模型的另一个迭代或不同模型的.XML 文件中导入和导出参考（包括词汇表和问题信息）。

典型的项目角色



Enterprise Architect是一个有效的多学科建模平台，它支持常见的工作实践，并提供特征来协助跨企业、业务、工程和软件项目的角色和学科。每个角色通常会使用工具的不同特征。此处概述了系统支持A许多角色及其职责。

您可以审阅每个角色支持的典型任务的摘要，或审阅相应角色标题的帮助主题，以探索Enterprise Architect如何帮助您在模型驱动项目中执行该角色。

角色和职责

角色	职责
业务分析师	创建业务流程的高级模型。
软件架构师	映射功能需求，执行对象的实时建模，设计部署模型和可交付组件的细节。
软件工程师	将使用案例映射到类图，细节类之间的交互，定义系统部署和定义包。
开发商	执行round代码工程，包括现有代码的逆向工程和从类元素生成代码。
项目经理	为元素分配资源、衡量风险和工作量、估计项目规模、管理元素状态、变更控制和维护。
测试仪	针对建模环境中的元素创建测试脚本。
实施经理	<ul style="list-style-type: none"> 跟踪和分配与维护相关的项目到Enterprise Architect中的元素 快速捕获并保存维护任务的记录，例如特征、变更、文档、问题、缺陷和任务 跟踪系统部署涉及的项目和流程的维护
技术开发者	为Enterprise Architect中已经存在的功能创建自定义添加。
数据库开发者	开发数据库，包括建模数据库结构、从现有数据库导入数据库结构以及生成DDL 以从模型中快速创建数据库。

典型任务总结

在整个设计和开发项目中，有许多不同的任务需要执行，这些任务可以由一个人执行，或者更可能是由具有不同职责的团队成员执行。无论哪种情况，Enterprise Architect都支持您在项目中可能承担的大部分（如果不是全部）职责。本主题中的描述确定了系统支持的许多工作角色。对于那些与您在项目中的角色最相似的人，请参阅该帮助的帮助主题以阅读有关该角色如何使用Enterprise Architect的描述，然后使用这些主题中的参考来探索一些重要的特征到角色。

典型工作角色总结

这些角色中的大多数都使用特定类型的图表，因此您可能希望了解更多关于一般图表类型的信息，尤其是特定类型的图表。

几种类型的项目团队成员可能希望生成有关其工作的文档并报告项目的开发和变化情况。使用Enterprise Architect，您可以生成文档或 Web 格式的项目报告。

角色	职责
业务分析师	对于建模： <ul style="list-style-type: none"> • 需求 • 高级业务流程 • 业务活动 • 工作流程 • 系统行为
数据库开发者	<ul style="list-style-type: none"> • 开发数据库 • 建模数据库结构 • 创建逻辑数据模型 • 生成模式 • 逆向工程数据库
软件架构师	<ul style="list-style-type: none"> • 映射系统的功能需求 • 实时映射对象 • 映射对象的部署 • 定义可交付组件
测试仪	<ul style="list-style-type: none"> • 开发测试用例 • 导入需求、约束和场景 • 创建质量测试文档 • 跟踪元素缺陷和变化
软件工程师	<ul style="list-style-type: none"> • 将使用案例映射到详细的类 • 定义类之间的交互 • 定义系统部署 • 定义包和软件架构
项目经理	<ul style="list-style-type: none"> • 提供项目估算 • 资源管理

	<ul style="list-style-type: none"> • 风险管理 • 维护管理
开发商	<ul style="list-style-type: none"> • 正向、反向和往返工程 • 可视化系统状态 • 可视化包安排 • 映射代码流
实施经理	<ul style="list-style-type: none"> • 对项目推出中的任务进行建模，包括网络和硬件部署 • 分配和跟踪元素的维护项目（问题、更改、缺陷和任务）
技术开发人员	<p>用于创建或自定义：</p> <ul style="list-style-type: none"> • UML Profiles • 模式 • 代码模板 • 标记值类型 • MDG 技术 • 插件

注记

- Enterprise Architect的企业版、统一版和终极版具有可以应用或关闭的用户安全特征；如果打开了安全性，您需要具有适当的访问权限才能使用许多功能

Enterprise Architect

企业架构师可以使用该工具创建企业的深度表示，包括业务、信息、应用程序和技术架构。

您可以将路线图创建为任何图表类型的叠加层，并且功能模型可以确保架构与企业的工作保持一致。您可以创建业务、信息、应用和技术架构，这些以及创建它们的团队都可以通过该工具进行管理。架构师和其他利益相关者可以自动生成业务友好的出版物，包括图表和图表，以可视化关键部分的架构。

Enterprise Architect任务

任务	细节
创建战略路线图	<p>路线图将组织从当前状态引导到目标状态，从而转变企业并通过一系列过渡状态将其引导到战略路线上。</p> <p>Enterprise Architect具有高效且灵活的路线图功能，允许您根据解决方案、细分市场和企业架构的预构建模型模式创建路线图。您可以为任何架构工件（包括功能、应用程序和技术项）制定路线图。</p>
定义业务能力模型	<p>业务能力是其他架构域（包括信息、应用程序和技术架构域）开展工作的基石。它们提供了一种查看企业做什么或需要做什么的方式。与业务流程相比，它们的创建要容易得多且耗时更少，并且与企业的工作直接相关。</p> <p>您可以使用模型功能元素或UML活动和嵌套元素的图表对这些非常重要的业务元素的层次结构进行建模。这些功能可以使用动态图表图例在二维中自动着色。</p>
目录应用程序组合	<p>应用程序架构提供了企业中应用程序的基本目录，描述了它们为转换、传输和存储信息所做的工作。架构还描述了应用程序所需或提供的接口，以及应用程序如何交互以执行业务模型中描述的活动，例如业务流程流程图。</p> <p>您可以在Enterprise Architect中模型应用程序组合并以各种方式可视化列表，包括图表、列表视图、规范管理器以及图表和图形。</p>
信息架构模型	<p>信息架构是企业架构计划成功的关键，因为信息是由构成其他架构的组件创建、使用和销毁的。信息架构通常包括对基线和目标架构的描述，其中定义了一系列可以在路线图上执行和描述的转换。</p> <p>Enterprise Architect是一个非常实用的工具，用于创建和维护信息架构。它对标准的复杂和广泛的支持以及支持信息模型的广泛工具，从高级分类和概念一直到模式级别以及组成它们的元素和列。</p>
创建技术参考模型	<p>技术参考模型(TRM) 引用通用平台服务和技術元素，并充当构建技术架构的基础。TRM提供了一组架构和解决方案构建块，最终将为业务和基础设施应用程序提供平台，以交付应用程序和基础设施服务。</p> <p>您可以使用Enterprise Architect技术，也可以使用任何建模语言来表示设备、节点软件和任何其他技术，工件系统模型。</p>
管理架构治理	<p>架构的治理对于程序及其创建的架构的成功至关重要。再完美的架构，如果没有技术人员正确实施的保证，架构中表达的愿景可能无法实现，对利益相关者的承诺将是空洞的，商业价值永远不会实现。</p> <p>您可以使用Enterprise Architect来治理和管理您的架构实践及其产生的架构，包括治理委员会、治理登记册等。</p>

业务分析师

A Enterprise Architect业务分析师流程并展示业务流程的高层模型，包括业务需求、活动、工作流、系统行为等。

使用Enterprise Architect，业务分析师可以描述管理特定业务的程序。这种模型旨在提供拟议系统的高级概述。

业务分析师任务

任务	细节
高级业务流程模型	<p>使用分析图，您可以对业务的高级流程进行模型。</p> <p>分析图是UML 2.5活动图的子集，不像其他图类型那样正式，但它们提供了一种有用的方法来表达基本的业务特征和需求。</p>
需求模型	<p>收集需求通常是开发解决方案的第一步，无论是开发软件应用程序还是详细说明业务流程；这是项目实施的重要一步。</p> <p>使用Enterprise Architect，您可以定义需求元素，将需求连接到模型元素以进行实施，将需求连接到层次结构中，报告需求，并摆脱需求模型元素的职责。</p>
模型业务活动	<p>您可以使用活动图来模型系统的行为以及这些行为与系统整体流程相关的方式。</p> <p>活动图并不模型系统的确切内部行为，而是在较高层次上显示一般过程和路径。</p>
工作流模型	<p>要将工作流中涉及的协作元素可视化，您可以使用工作交互概览图，它提供了系统中概述的子活动。</p>
显示系统行为	<p>在将系统行为显示为用例图时，Enterprise Architect提供了一个易于理解的工具，用于映射系统的功能需求和行为。</p>

系统工程师

系统工程师可以使用 Enterprise Architect 生成复杂网络机械系统的强大且高效的模型。模型可以是需求模型、结构模型和行为模型，包括使用用例、包图表、块定义图表、序列图表内部块、活动图表、图表、状态机器图表和参数图表。Enterprise Architect 有一套全面的工具来帮助工程师和其他利益相关者。

系统工程师的任务

任务	细节
定义和管理需求及其关系	<p>需求工程领域是解决方案开发生命周期中最关键的学科之一，它对项目的成功有记录的影响。</p> <p>Enterprise Architect 拥有一系列无与伦比的工具，用于开发、管理、可视化和记录需求，包括导入、集成和与外部需求管理系统同步的工具。</p>
用使用用例描述用户目标	<p>系统工程师使用使用用例作为从用户角度表示功能需求的方法。它们是目标驱动的，因为用例定义了用户在与系统交互时试图实现的目标。Enterprise Architect 全面支持用例图的开发，全面支持用例文本的建模和管理；它有一个用于处理使用用例的独特且高效的工具，称为场景生成器。</p>
使用模块模型结构和约束	<p>块是系统结构的基本单位。系统工程师使用块来描述整个系统、子系统、组件、流经系统的项目、约束或驻留在系统外的实体。类似于我们的自然语言，块可以代表抽象的、逻辑的或物理的东西。</p> <p>Enterprise Architect 拥有丰富的工具集，可帮助系统工程师使用 Blocks 并可视化系统定义中这些最重要元素的结构和行为。</p>
协调行为与活动	<p>活动图正式基于称为 Petri 网的数学分支，它使用标记系统来指示动作的序列以及流经系统的项目。流动的项目可以是信息项目、物理项目，甚至是控制信号。</p> <p>Enterprise Architect 提供了一个丰富的工具箱来处理这些行为元素及其关系，包括以活动和行动的形式将系统行为分配给块，并将这些元素与块拥有的行为特征相关联，例如操作。</p>
通过仿真可视化运动中的系统	<p>仿真提供了一种查看运动系统并可视化其在其生命周期中的行为方式的方法。</p> <p>作为领先的系统工程工具，Enterprise Architect 允许系统工程师使用符合行业标准的建模技术和语言来构建模型，以表示网络机械系统。这些模型充当协作工程师、顾问和其他人之间通信的设备，但也可用于使用 OpenModelica 和 MATLAB 的 Simulink 使用的行业标准建模语言生成可视化和仿真。</p>

软件架构师

Enterprise Architect架构师配合使用Casa 对对象的时间建模，使用对象的时间序列，使用交互图，使用实时软件图，使用通讯或交互组件的部件视图（部署视图），设计模型和细节组件图。

软件架构师Tasks

任务	细节
地图系统功能需求	<p>使用Enterprise Architect，您可以采用由业务分析师建模的高级业务流程并创建详细的使用案例。</p> <p>使用案例描述了系统的建议功能，仅用于单个离散工作单元的细节。</p>
实时时间对象	<p>您可以使用交互（序列和通讯）来模型系统的动态设计。</p> <p>序列图详细说明对象之间细节的消息，以及对象的生命周期。</p> <p>通讯图与序列图类似，但显示的是对象与其他object交互的方式。</p>
对象的地图部署	<p>您可以使用部署图提供处理节点的运行时配置和在节点上运行的组件的运行视图。</p> <p>部署图显示了硬件、软件和系统上使用的任何中间件之间的连接。</p>
详细的可交付组件	<p>使用部件图，您可以对系统的物理方面进行模型。</p> <p>组件可以是可执行文件、库、数据文件或属于系统一部分的其他物理资源。</p> <p>组件模型可以从类模型中从头开始开发，也可以从现有项目和第三方供应商中引入。</p>

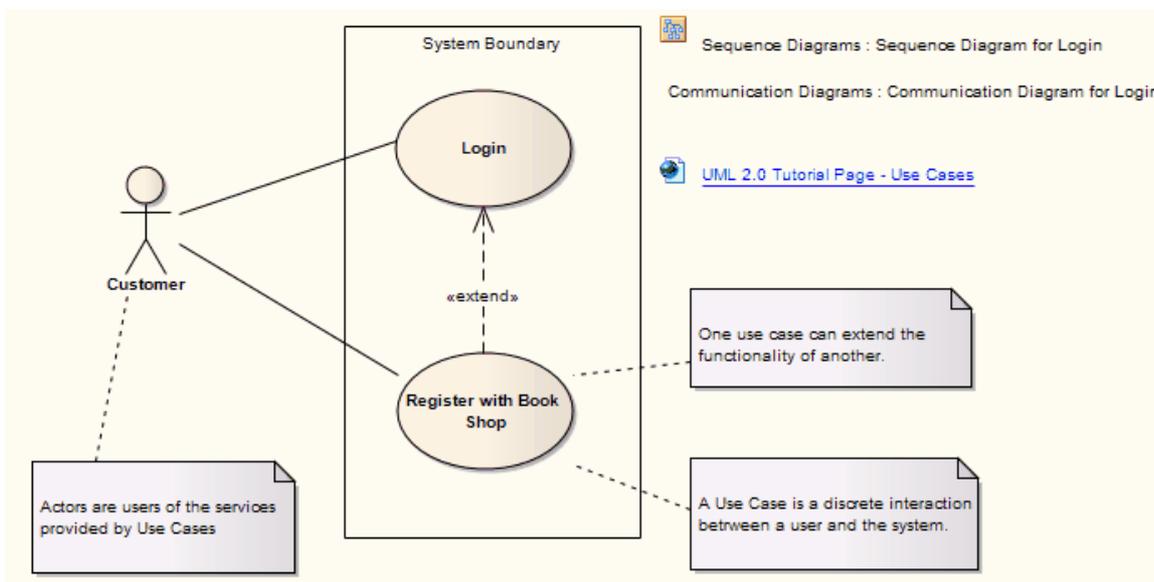
软件工程师

使用Enterprise Architect的软件工程师可以手动将使用案例映射到类图上，细节说明类之间的交互，用部署图定义系统部署，用包图定义包。

软件工程任务

任务	细节
将使用案例映射到详细类	在Enterprise Architect中，您可以研究由软件架构师开发的使用案例，并使用该信息创建满足使用案例中定义的目标的类。 A类是标准的UML结构之一，用于详细说明在运行时生成细节的模式；要记录使用案例和类之间的关系，您可以创建将元素与实现连接器链接的图表，和/或在关系矩阵中映射实现连接器。
详情交互Classes	您可以使用交互（序列和通讯）来模型系统的动态设计。 序列图用于详细说明对象之间细节的消息，以及对象的生命周期。 通讯图类似于序列图，但它显示了对象与其他对象交互的方式。
定义系统部署	部署图提供了处理节点的运行时配置和在节点上运行的组件的运行视图。 部署图可用于显示硬件、软件和系统上使用的任何中间件之间的连接，以解释组件的连接和关系。
定义包	您可以使用包图来细节说明软件架构。 包图用于将图和元素组织成可管理的组，声明依赖关系。

简单用例图



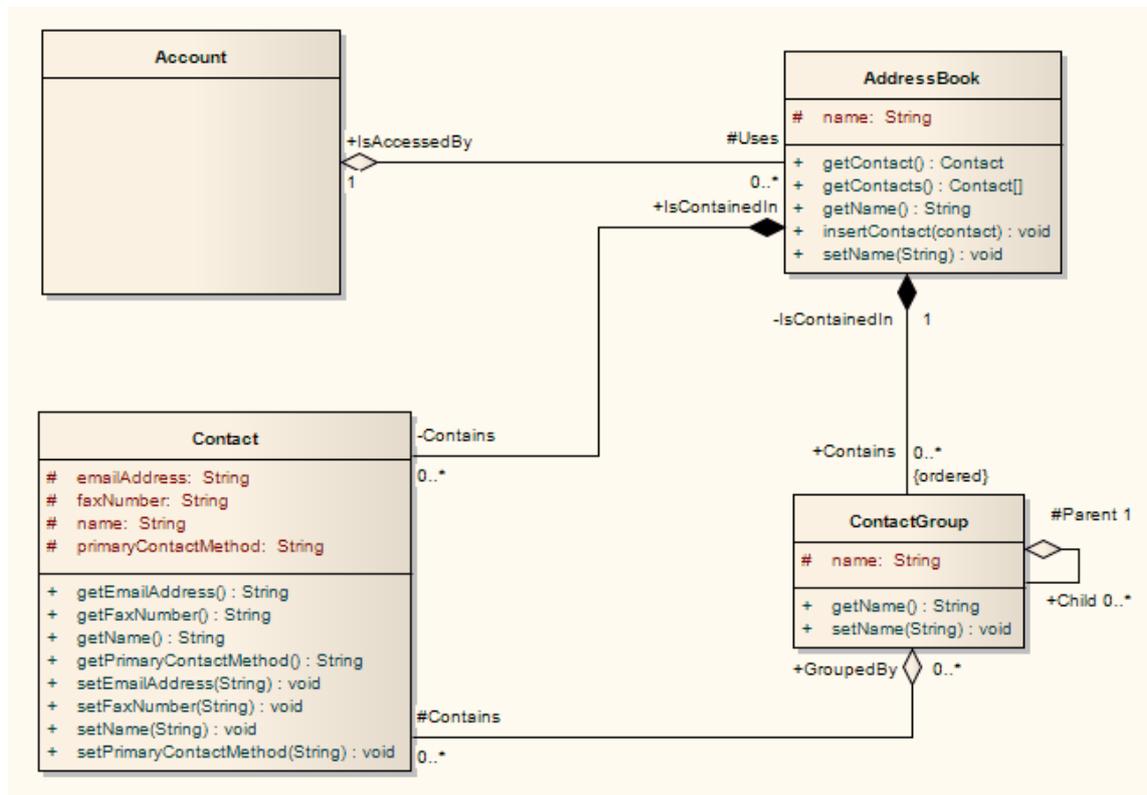
开发者

开发人员可以使用Enterprise Architect执行round代码工程，包括现有代码的逆向工程和从类元素生成代码。

开发人员任务

任务	细节
往返工程	Enterprise Architect为您提供了从现有源代码到UML 2.5 模型并再次返回的 “round”软件的无与伦比的灵活性。 往返工程涉及代码的正向和逆向工程，保持模型和代码同步。
逆向工程	在Enterprise Architect中，您可以对多种支持语言的代码进行逆向工程，并以类图的形式查看现有代码，以说明系统的静态设计视图。 类图显示类和接口，以及它们之间的关系；UML类图中定义的类在编程语言的实现中可以直接对应。
正向工程	除了将您的代码逆向工程到您的模型中，您还可以将您的模型元素转发到代码中（代码生成）。 这样，您可以使用Enterprise Architect对您的模型进行更改，并快速实施源代码中的更改。
确定系统状态	为了可视化系统的状态，您可以使用状态机图来描述元素如何在状态之间移动，根据转换触发器和约束守卫对它们的行为进行分类。 状态机图捕捉系统随时间的变化，通常与特定的类相关联；通常一个类可以有一个或多个状态机图来充分描述它的潜在状态。
可视化包安排	包图帮助您设计系统的架构；它们用于将图表和元素组织成可管理的组，并声明它们的依赖关系。
遵循代码流	活动图帮助您更好地理解代码流。 活动图表说明系统的动态特性；您可以对活动之间的控制流进行模型，并表示系统状态的变化。

简单类图表



笔记

- 可以使用状态机、包和活动图来更好地理解代码元素之间的交互以及代码的排列方式

项目经理

Enterprise Architect为项目管理提供支持。项目经理可以使用该系统为元素分配资源、衡量风险和工作量、估计项目规模、管理元素状态、变更控制和维护。

项目经理任务

任务	细节
提供项目估算	在Enterprise Architect中，您可以访问一个全面的项目估算工具，该工具计算来自使用用例和对象的参与者，以及定义工作环境的技术和环境复杂性的项目配置。
资源管理	在系统组件的设计和开发中管理资源分配是一项重要且有时困难的任务；Enterprise Architect为您提供了一种有效的工具，用于将资源直接分配给模型元素并跟踪一段时间内的进度。
风险管理	您可以使用“风险”窗口将风险分配给项目中的元素；使用风险类型，您可以命名风险、定义风险类型并赋予其权重。
维护	在Enterprise Architect中，您可以将与维护相关的项目分配给元素并对其进行跟踪，从而为特征、更改、文档、问题、缺陷和任务等项目提供快速捕获和记录保存。 您还可以创建和维护流程、程序、术语和描述的项目词汇。

测试人员

Enterprise Architect为测试人员和质量保证人员提供设计测试功能，以针对建模环境中的元素创建一系列测试脚本。

测试任务

任务	细节
测试案例	使用Enterprise Architect，您可以为每个元素模型设置一系列测试。 测试类型包括单元、验收、系统、集成、检查和场景测试。
导入要求、约束和场景	要使用测试来维护整个业务流程的完整性，您可以导入在开发生命周期的早期迭代中定义的需求、约束和场景。 需求指示元素在模型中必须履行的合同义务。 约束是通过测试过程必须满足的条件；约束可以是： <ul style="list-style-type: none">• 前置条件（在处理事件之前必须为真的状态）• 后置条件（事件处理后必须发生的事件）或• 不变的约束（必须在事件的持续时间内保持真实） 场景是object随时间的动作的文本描述，可用于描述测试的工作方式。
创建质量测试文档	Enterprise Architect提供生成 .RTF、DOCX 和 PDF 文件格式的高质量测试文档的功能。
元素缺陷变化	在缺陷跟踪中，您可以将缺陷报告分配给模型内的任何元素，以便所有参与项目的人都可以快速查看缺陷的状态，了解哪些缺陷需要解决，哪些已经处理。

实施经理

Enterprise Architect为项目实施的管理提供支持。您可以在Enterprise Architect中跟踪和分配与维护相关的项目，并快速捕获和更新维护任务的记录，例如特征、更改、文档、问题、缺陷和任务。通过为部署过程中涉及的每个元素提供集中的功能，Enterprise Architect提供了一个方便的解决方案，用于跟踪系统部署中涉及的项目和过程的维护。

实施任务和工具

任务	细节
开发部署图表	<p>使用部署图，您可以模型项目的推出，包括网络部署和 workstation 部署。</p> <p>参与项目部署的用户可以向图表元素添加维护任务。</p> <p>部署图提供了网络上节点或工作站的运行时配置的静态视图，以及在节点上运行或在运行中使用的组件。</p>

技术开发人员

技术开发人员是Enterprise Architect的用户，他们为 Enterprise Enterprise Architect中已有的功能创建定制的补充。

新增内容包括UML Profiles、模式、代码插件模板、标记值类型、脚本、自定义查询、转换、MDG技术和Enterprise Architect。通过创建这些扩展，技术开发人员可以针对特定任务定制Enterprise Architect建模过程并加快开发速度。

开发技术

扩展	细节
UML Profiles	<p>通过创建UML Profiles，您可以创建自定义扩展来构建特定于特定域的UML模型。</p> <p>Profiles存储为XML文件，可以根据需要导入任何模型。</p>
模式	<p>模式是一组协作对象和类，它们为建模问题的可重复解决方案提供通用模板。</p> <p>当在任何新项目中发现模式模式模板</p> <p>模式可以重新使用，并为任何未来的项目修改适当的变量名。</p>
代码模板	<p>代码模板用于自定义Enterprise Architect生成的源代码的输出；通过这种方式，您可以生成Enterprise Architect不特别支持的代码语言，并定义系统如何生成源代码以符合您自己的公司风格准则。</p>
标记值类型	<p>除了UML语言直接支持的信息外，Enterprise Architect中使用标记值来扩展与元素相关的信息。</p> <p>A标记值，严格来说，是一个建模项的一个属性的值，这个属性称为标签；例如：一个名为Person的类元素可能有一个名为'Age'的标签，其标记值为'42'。</p> <p>更松散地说，标签和值的组合可以称为标签标记值。</p> <p>标记值类型是A组参数，它们定义和/或限制标签的可能值，并且在许多情况下，如何将特定值分配给标签；例如，'年龄标记值类型'的整数可能有一个'，因此用户只需键入一个数字值。</p> <p>或者，类型可以是“Spin”，具有下限和上限，例如20和120，因此用户通过单击字段中的箭头来设置值，以在20和120的范围内增加或减少值。</p> <p>通常，标记值在代码生成过程中使用，或由其他工具用于传递用于以特定方式对元素进行的操作的信息。</p>
MDG技术	<p>MDG技术可用于创建资源的逻辑集合，这些资源可以包含通过技术文件访问的UML Profiles、模式、代码模板、图像文件和标记值类型。</p>
Enterprise Architect插件	<p>使用插件，您可以将自己的功能构建到Enterprise Architect中，创建您自己的可以扩展系统功能的小程序，定义您自己的菜单，以及创建您自己的自定义视图。</p>

数据库开发人员

Enterprise Architect支持一系列用于数据库开发的特征，包括建模数据库结构、从现有数据库导入数据库结构以及生成 DDL 以从模型中快速创建数据库。

数据库开发任务

任务	细节
创建逻辑数据模型	借助Enterprise Architect，您可以使用内置的UML数据建模配置文件构建数据库图表。 这支持主键和外键、基数、验证、触发器、约束和索引的定义。
生成架构	通过使用Enterprise Architect的 DDL 生成函数，您可以创建 DDL 脚本以从模型中创建数据库表结构。 Enterprise Architect目前支持： <ul style="list-style-type: none">• DB2• 火鸟• MS Access• MySQL• 女士# 服务器• 甲骨文• PostgreSQL
逆向工程师数据库	使用 ODBC 数据连接，您可以从现有数据库中导入数据库结构以创建数据库模型。 通过直接从数据库生成模型，您可以快速记录您的工作并通过UML的图形优势创建复杂数据库的图表帐户。

示例数据模型图表

