



Enterprise Architect

User Guide Series

# Enterprise Architecture

How to model Enterprise Architecture? Sparx Systems Enterprise Architect helps visualize an enterprise at different levels of abstraction, the strategic context of its architecture, its evolution from a current to a future state and its implementation.

Author: Sparx Systems &  
Stephen Maguire

Date: 2021-09-02

Version: 15.2

CREATED WITH  ENTERPRISE  
ARCHITECT



# Table of Contents

Enterprise Architecture .....	9
Introduction .....	13
Enterprise Architecture Overview .....	22
What is Enterprise Architecture .....	24
Context for Enterprise Architecture .....	26
Levels of Architecture .....	30
Types of Architecture .....	37
Scope of Architecture .....	47
Characteristics of Good Architecture .....	56
Lists, Diagrams and Matrices .....	63
Meet the Enterprise Architecture Tools .....	68
Activity Diagram .....	70
Auditing .....	74
Balanced Scorecard .....	77
Business Process Diagram .....	80
Calendar .....	84
Class Diagram .....	87
Component Diagram .....	90
Dashboard Diagrams .....	94
Decision Tree Diagram .....	97
Deployment Diagram .....	101
Documentation .....	105
Gap Analysis Matrix .....	108

Heat Map .....	112
Import and Export Spreadsheets .....	115
Organizational Chart Diagram .....	118
Patterns .....	122
Relationship Matrix .....	126
Requirements Diagram .....	130
Roadmap Diagram .....	135
Specification Manager .....	139
Strategy Map .....	142
Library .....	146
Time Aware Modeling .....	149
Traceability Window .....	155
Value Chain .....	159
Uses of an Enterprise Architecture .....	162
Mergers and Acquisitions .....	164
Corporate Divestiture .....	165
Architecture Oversight .....	167
Business and Systems Improvement .....	168
Communication .....	170
Enterprise Transitions .....	172
Implementation Guide .....	174
Portfolio Management .....	176
Architecture Program Setup .....	177
Management Structure .....	178
Architecture Framework .....	187
Architecture Process .....	189
Architecture Repository .....	195



Tool Setup	200
Architecture Principles	221
Managing an Enterprise Architecture	224
Architecture Governance	225
Governance Process	226
Architecture Steering Committee	228
Architecture Review Board	232
Governance Register	237
Developing an Enterprise Architecture	239
Architectures	241
Business Architecture	244
Strategic Plans	246
Mission and Vision	250
Drivers	256
Goals and Objectives	259
Capabilities	262
Business Processes	265
Information Architecture	269
Conceptual Information Model	271
Logical Data Model	274
Schemas and Messages	276
Physical Data Model	278
Application Architecture	281
Application Lists Diagrams and Matrices	283
Application Communication	288
Interface Lists	291
Capabilities and Applications	293

Business Processes and Applications	296
Technology Architecture	298
Application Platform Services	299
Technical Reference Model	301
Infrastructure Facilities	303
Stakeholder Modeling	306
Requirements Modeling	309
Documenting an Enterprise Architecture	311
Architecture Description	313
Architecture Requirements Specification	314
Architecture Vision	316
Communication Plan	317
Compliance Assessment	319
Project Glossary	320
Enterprise Architecture Techniques	321
Architecture Governance	322
Architecture Requirements Management	327
Architecture Partitioning	333
Application Portfolio Management	338
Balanced Scorecard	344
Baselines and Versioning	348
Business Goals and Objectives Modeling	352
Business Scenarios	357
Capability Based Planning	363
Capability Modeling	366
Compliance Assessment	372
Concept Modeling	373

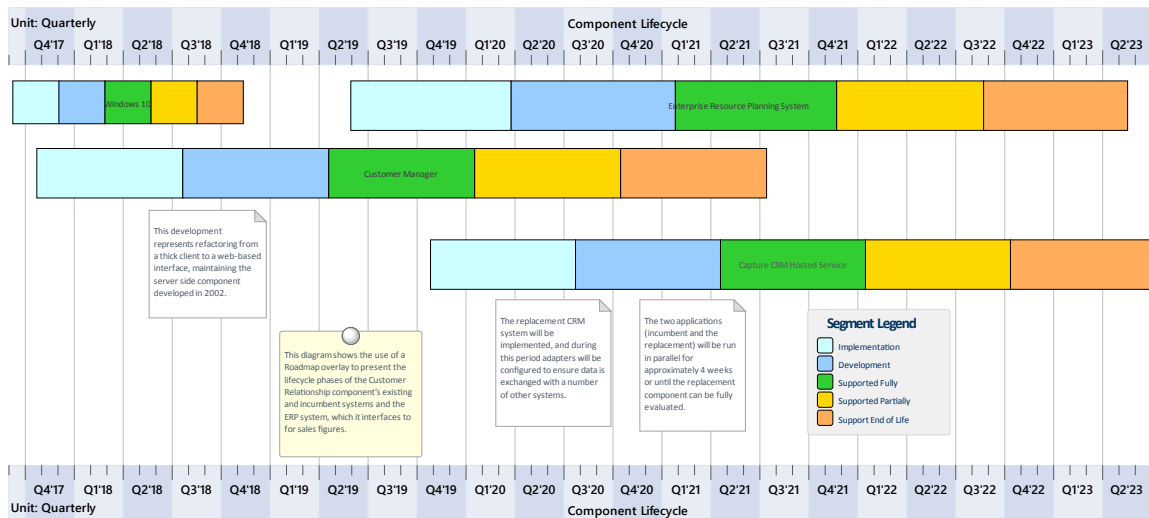
Data Modeling	378
Driver Modeling	383
Functional Decomposition	386
Gap Analysis	392
Glossary	394
Interface Analysis	397
Mind Mapping	401
Organizational Modeling	404
Pattern Analysis	407
Principles Management	410
Process Analysis	415
Process Modeling	419
Reviews	426
Risk Analysis and Management	429
Roadmaps	432
Stakeholder Management	437
Standards Modeling	450
Technical Reference Model	454
Time Aware Modeling	462
Use Cases and Scenarios	468
Views and Viewpoints	473
Workshops	482
Additional Enterprise Architecture Tools	490
Auto-Names and Counters	491
Baseline Tool	494
Boundary	497
Document Artifact	501

Discussions .....	505
Glossary .....	508
Image Manager .....	511
List View .....	514
Mind Mapping Diagram .....	517
Model Mail .....	521
Model Views .....	524
Model Search .....	527
Package Browser .....	531
Pan and Zoom .....	535
Browser Window .....	538
Requirements Checklist .....	542
Requirement Properties .....	544
Risk Taxonomy .....	548
Security .....	552
Stereotyping .....	556
Tagged Values .....	560
Visual Filters .....	563
Working Sets .....	567

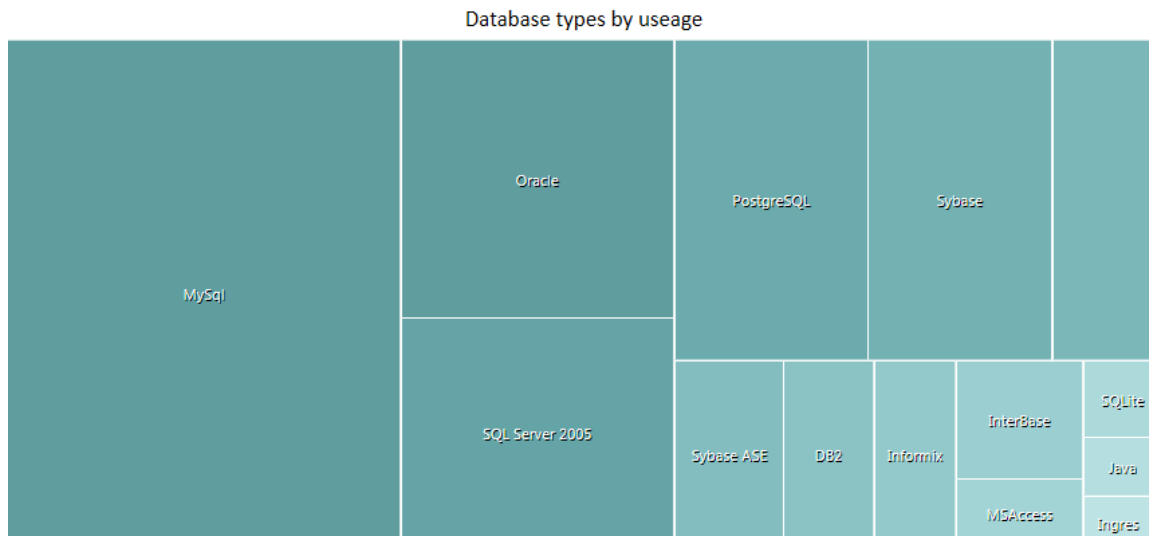
# Enterprise Architecture

Enterprise Architecture has emerged as a discipline that can help steer the 'ship' of the enterprise through both quiet and turbulent waters, charting a course from its current location to a future location in a safe and streamlined way. The discipline has become more prevalent in recent years, but the precepts go back almost to the beginning of what is often termed the information age. An enterprise is one of the most complex man-made systems and is composed of human, political, social, software, hardware and technology components. In an enterprise of any appreciable size, it is impossible for a single person to understand the way the parts all work together, let alone understand its position in relation to the system of other organizations that form its environment, or to determine how it can evolve.

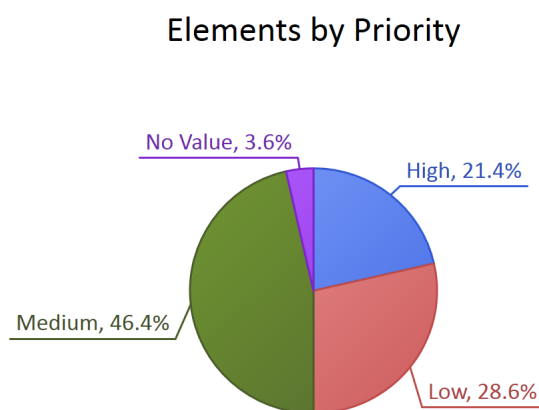
Enterprise Architecture can be used to create visualizations of the enterprise at different levels of abstractions and to create Roadmaps that show how the enterprise can be transitioned from its baseline (current) state to a target (future) state.



Enterprise Architect is a powerful platform that can be used to define the strategic context for an Enterprise Architecture, the Enterprise Architecture itself, and the Implementation initiatives that realize the designs and that finally deliver the business value. It can serve both as the architectural repository and as a tool for managing the process by which architectures are created and maintained, including an architectural requirements management platform. Powerful visualization capabilities allow models to be transformed and presented in a variety of compelling ways that will delight stakeholders, from the executive level through to the implementation teams.



The tool can be used to define Strategic, Tactical and Solution Architectures and to provide compelling views for a wide range of stakeholders, from senior executives through to implementation partners. Business, Information, Application and Technology architectures can be created and managed, and baseline and target architectures defined, allowing transitions to be visualized.



This diagram shows a Pie Chart element depicting element priorities for all the Requirements in a selected Package. It provides a useful summary for a Requirements Manager and is dynamically updated when the priority changes and the diagram is reopened. A range of other pre-defined Charts and user-defined Charts can also be added. A filter has been added to exclude all elements other than Requirements.



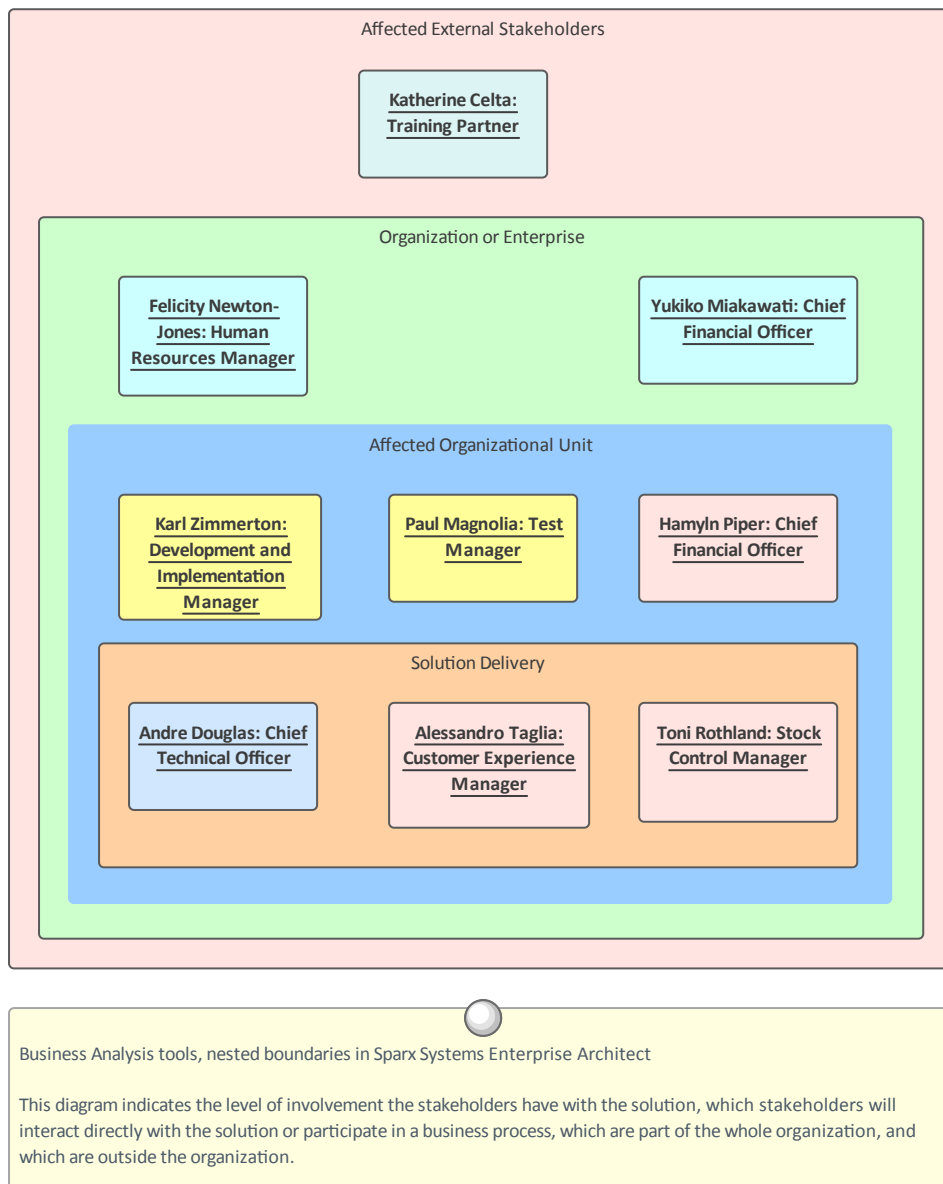


# Introduction

The discipline of Enterprise Architecture dates back to the mid-eighties when John Zachman recognized the need to manage the complexity of distributed technology systems. Enterprise Architecture has since become one of the most important disciplines in the business and information technology professions and has become increasingly relevant in an era dominated by digital disruption. When done well, Enterprise Architecture can be used to guide an organization through the complex business and digital landscape that currently confronts business and technology leaders.

Enterprise Architect has become the tool of choice for many leaders in the industry because of its flexible, extensible and pragmatic approach to modeling complex systems. As a platform, Enterprise Architect offers a unique capability in supporting the integration of strategic, business and technology models, from motivation models down to the implementation of systems. The tool allows the architect to create Strategic models (including diagrams such as the Balanced Scorecard), Capability models, Tactical models such as Gap analysis, and Roadmaps and Operational models, appealing to stakeholders from the senior executives through to line managers and solution and implementation teams.

## Stakeholder Onion Diagram



## How it will help you

Readers will typically come to the topic of Enterprise Architecture with some existing knowledge or experience, even if it is something that has been learnt in lectures or by on the job training, or perhaps by using a different tool. Readers will benefit by understanding the product's features

and the tools that are available to develop and manage Enterprise Architectures in Enterprise Architect, and this will enable them to be more productive as an individual and as a member of a team.

## **Who will benefit**

Anyone involved in the development or management of architectures, whether at a strategic level, a business value level or a technology level will benefit from reading this information. This includes a wide range of roles including Strategic Thinkers, Senior Management, Business, Information, Application and Technology Architects, Solution Architects and Implementation teams whose work and decisions will ultimately be guided by the architectures.

## **What you will learn**

This topic will teach you how to use the powerful features of Enterprise Architect to develop and manage Enterprise Architectures, to create documentation and to work collaboratively as a member of a team using a formal or informal architecture framework. You will learn what tools are available, how to use them and which tools should be used to perform a particular technique. For example, regardless of the process or framework that is adopted, at

some point *Architectural Partitioning* will need to be performed; this topic will describe the technique and how to best achieve it using the tools and facilities you have at your fingertips using Enterprise Architect.

## Overview of the Documentation

This table provides a list and a description of the subjects that are included in this work, giving an overview of the material.

<b>Enterprise Architecture Overview</b>	This topic gives a high level view of Enterprise Architecture, describing what it is and the levels, types and styles of architecture. It describes some of the characteristics of good architecture and the outputs and content of an architecture process such as catalogs, matrices and diagrams. It also puts Enterprise Architecture into the context of other disciplines, from the strategic level through to the operational level, including Solution Architectures.
<b>Meet the Enterprise Architecture Tools</b>	This topic lists the key tools that are used in Enterprise Architecture, including a picture of the tool in action, where to find the tool, how to use it and how to become

proficient in using the tool. There are many additional tools that will be useful, which are described in the last topic entitled *Additional Enterprise Architecture Tools*.

### **Uses of an Enterprise Architecture**

This topic describes how an Enterprise Architecture can be utilized, who will get benefit and how that benefit can be realized. This includes the use of an architecture as a communication tool and strategic guide for senior management, or as a guide for solution architects and implementation teams and for the purposes of the oversight of implementation projects. It will also describe the use of an architecture for portfolio management, business systems improvement and for analyzing, defining and documenting enterprise transitions, including Acquisitions and Mergers.

### **Architecture Practice Setup**

Many architects suffer from the blank canvas syndrome, because much of the literature describes how to develop architectures but is silent about setting up an architecture practice. This topic will guide you in using Enterprise Architect to set up an Architecture Practice, including Framework options and how to set up the

tool to support the practice. This includes repository structure, principles, team collaboration, documentation generation using templates, meta-model definition, governance structures, language support and more. It loosely aligns with the Preliminary phase of The Open Group Architecture Framework Architecture Development Method (TOGAF ADM).

### **Planning an Enterprise Architecture**

An architecture is a project and needs to be planned. This topic explains how to set up an architecture project, including the definition of the scope, constraints and the desired enterprise outcomes, including problems or opportunities that the architecture is addressing. It describes how the architecture aligns with enterprise strategy and how it relates to solution architectures that will ultimately implement the architecture in practice. It loosely aligns with the Vision phase of the TOGAF ADM.

### **Managing an Enterprise Architecture**

The management of an architecture involves the set up, monitoring and control of the architecture project from its inception through to its delivery. Team capabilities, tool usage, the quality of architectural artifacts, communication

mechanisms, governance structures and stakeholder management are all described in this topic. The use of Enterprise Architect as a tool for the management of an architecture will be fully described and exemplified.

**Developing  
an  
Enterprise  
Architecture**

This topic describes how Enterprise Architect can be used to create and maintain an Enterprise Architecture. It is the centerpiece of the discussion and articulates how the architecture is used to describe the way the organization will transition from a baseline to a target state, resulting in the attainment of business goals and objectives and new or augmented capabilities.

**Documentin  
g an  
Enterprise  
Architecture**

This topic describes the powerful documentation features of Enterprise Architect, allowing Architects, Analysts, Managers and others to generate a range of documentation, from ad-hoc reports to publication of quality documents directly out of the architecture repository. It will describe creating organization-specific documentation using a sophisticated and flexible template system for generating documents into a range of formats, including PDF, DOCX and HTML.

**Enterprise Architect's Frameworks and Standards** This topic describes Enterprise Architect's support for frameworks, languages and standards. This consists of a wide range of frameworks, including The Open Group Architecture Framework (TOGAF), the Federal Enterprise Architecture Framework (FEAF) and the Zachman Framework. Languages such as the Unified Modeling Language (UML), ArchiMate and Business Process Model and Notation (BPMN) are described, as are standards such as the National Information Exchange Model (NIEM).

**Enterprise Architect Techniques** This topic describes the important techniques that are used by Architects when working with architectures, from the setup phase through planning, managing, developing and documenting architectures. Each technique is described, with a list of the tools available in Enterprise Architect that can be used to perform the technique, explaining the use of the tool in the context of the technique.

**Additional Enterprise** This topic lists a series of additional tools that can be used for Enterprise



**Architecture Tools**    Architecture, including a picture of the tool in action, where to find the tool, how to use it and how to become proficient in using the tool. There are a number of key tools that are considered to be most important for enterprise architecture, which are described in the *Meet the Enterprise Architecture Tools* section.

# Enterprise Architecture Overview

Enterprise Architecture has emerged as a critical discipline to ensure that an enterprise and the organizations it comprises have an understanding of the significant elements from which it is made, from strategic goals through to the business and information technology components that assist in achieving those goals. The discipline also allows enterprises to create architectures that will transition from where they are to where they need to be. Now more than ever in this age of digital disruption, when organizations can no longer rely on length of tenure in a field or being bigger than their competition as a safeguard against competitive forces, Enterprise Architecture as a discipline is vital.

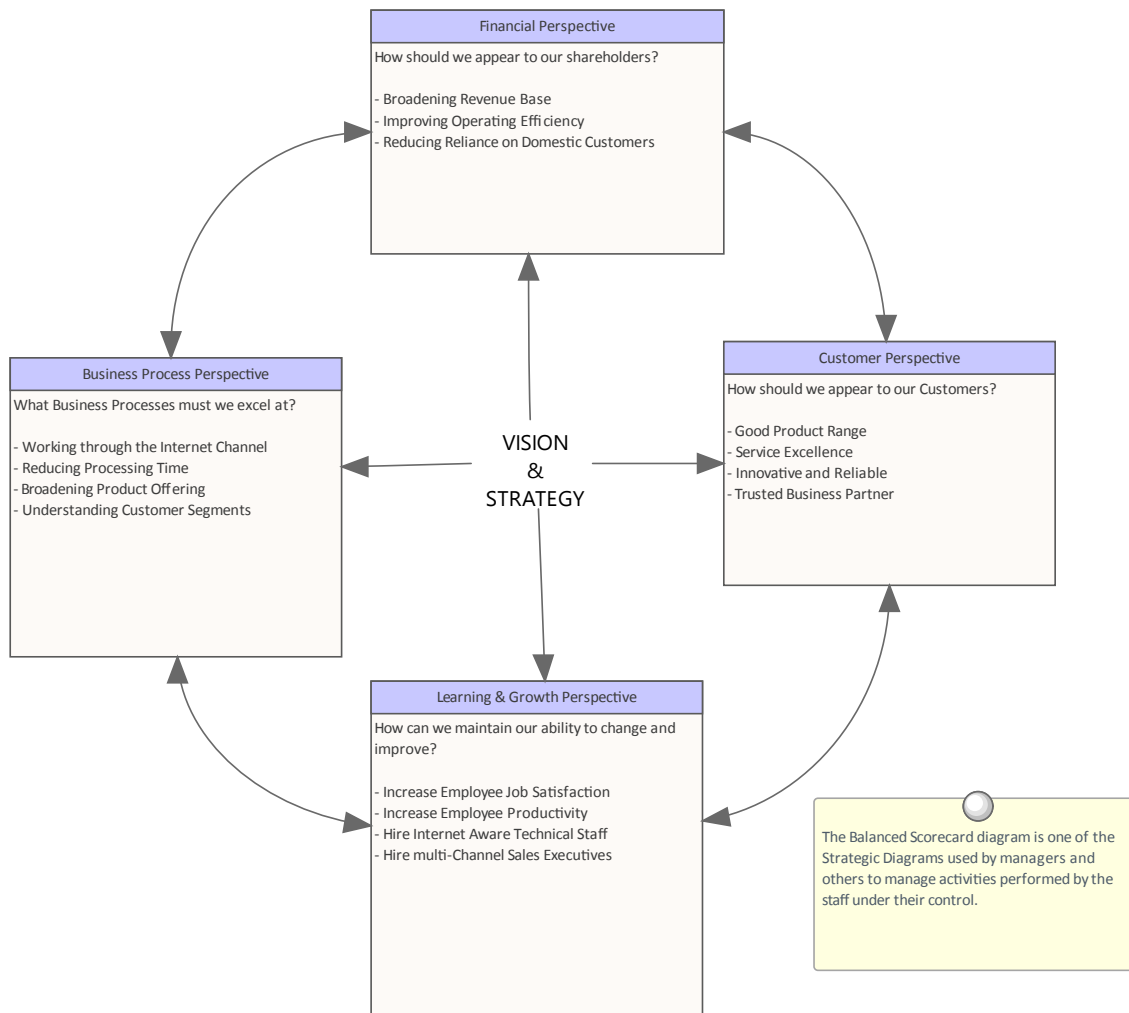
The profession of Enterprise Architect is rarely taught as a separate degree course at tertiary institutions, and it also suffers under the proliferation of large, conflicting and at times burdensome frameworks and a lack of tool support. This has led to architecture practitioners finding it difficult to articulate or demonstrate the value of their 'profession'.

This section addresses the questions:

- What is Enterprise Architecture?
- Where does it fit into the context of other disciplines?
- What are the characteristics of good architecture?

The section also discusses the levels, types and styles of architecture that exist, and describes the notational mechanisms that are at an architect's disposal.

Enterprise Architect's pragmatic approach to modeling, and the extensive set of facilities available to the architect and others, make it a powerful tool both as an architecture repository and as a platform for creating, managing and disseminating architectural work.



# What is Enterprise Architecture

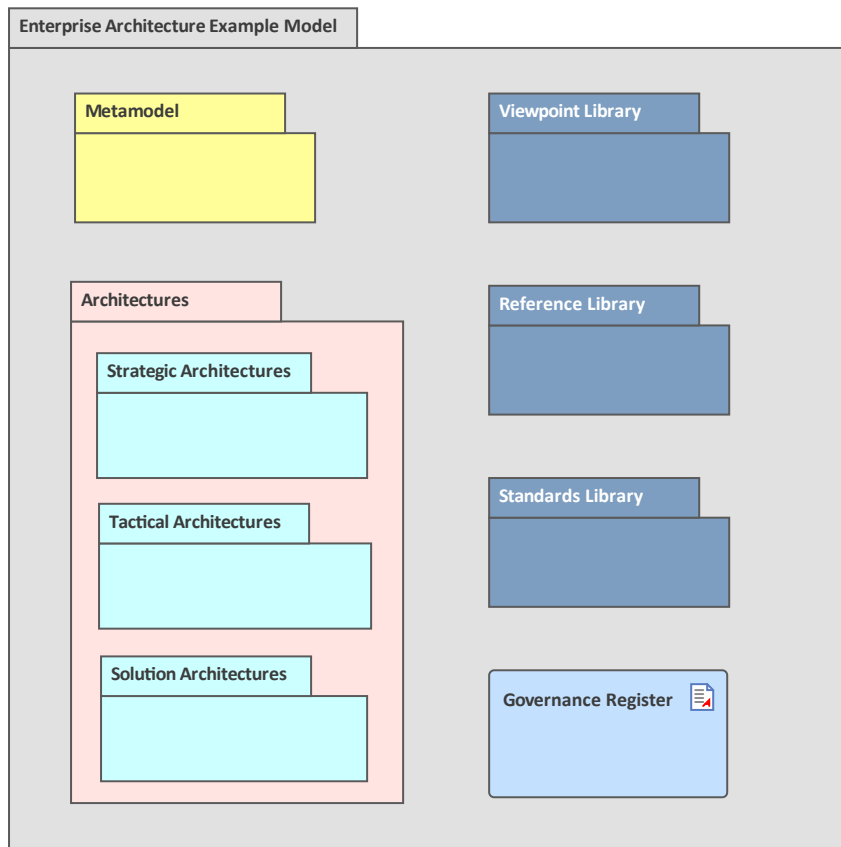
There is still robust and lasting debate about what architecture is in the context of an enterprise, which makes it difficult to provide a universally accepted definition of the term 'Enterprise Architecture'. This is not helped by the fact that there is not a standard curriculum for architecture or any of its flavors in our tertiary institutions, nor do many institutions teach architecture at a graduate or post graduate level. People come to the discipline from a wide range of backgrounds and produce an equally wide range of work products that come under the general heading of architecture.

The ISO/IEC 42010 standard defines architecture as:

'The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution'.

In this context the system that is under discussion is the enterprise. Using this definition it is important to separate the fact that there is a description of the architecture (often called an architecture) and the architecture that has been built. Enterprise Architect is used to define or document the description of the architecture and therefore provides a representation or way of visualizing a planned or built architecture. The powerful facilities available in Enterprise Architect have been designed by practitioners and are continually improved to create a formidable tool that is both

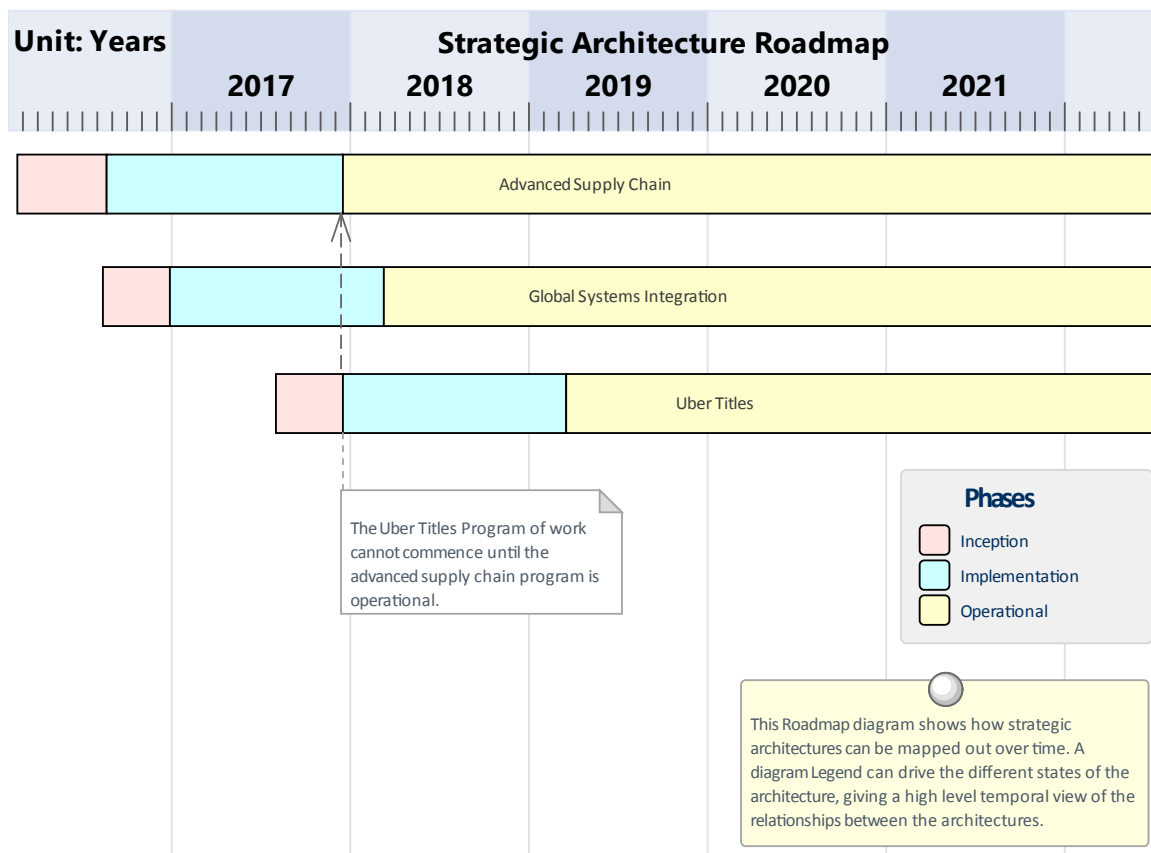
robust and flexible, and that makes creating and managing Enterprise Architectures easy, regardless of the definition that is applied.



# Context for Enterprise Architecture

An Enterprise Architecture Program is best viewed as an operational unit of the business, and as such it has a context. The program must provide value to the business; it does this by ensuring that architectural effort is aligned with the strategic plans of the organization and that the implementation initiatives are carried out in such a way that honors the enterprise architecture.

Enterprise Architect has a number of powerful features to model and visualize the alignment of the architectures and the strategic plans. These visualizations can be used when assessing which Information Technology initiatives are contributing to the strategic business goals and objectives. Enterprise Architect has a number of useful facilities for guiding the implementation projects and assessing their level of compliance, including the **Team Library** facility and the definition of how principles are applied in the context of the solution architectures.

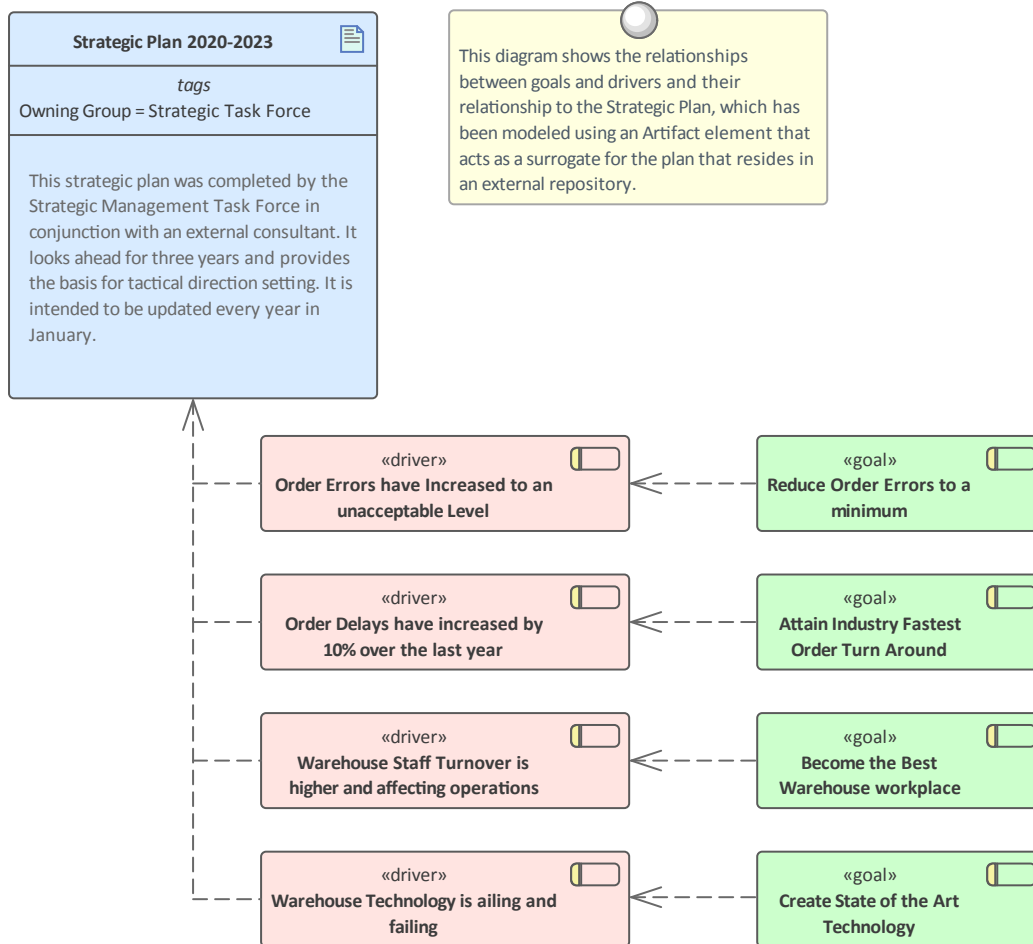


## Strategic Context

The Business Architecture must be related back to the Strategic Plans to ensure that all the other architecture domains that describe the architecture in more detail are ultimately going to benefit the enterprise and deliver value. Business Architects typically gather information from the company strategists and should be privy to the high-level discussions and decisions about the future plans for the enterprise and its organizations.

Enterprise Architect has a number of tools that help to ensure that the architecture's alignment to the strategic plans can be visualized, and that the drivers forming part of the Business Architecture are related back to elements of the

plan. The business plan will typically reside in a corporate repository but a hyperlink can be created from an Artifact within Enterprise Architect that will launch the Strategic Plan or other documentation. Elements of the Business Architecture can be related to the Artifact that acts as a surrogate for the external plan.



## Implementation Context

The purpose of the architecture is to ensure that initiatives and projects deliver the business value and benefits that have been described in the architectural models, so



monitoring the compliance of the implementation projects is critical to the success of the architectures and ultimately the architectural program. Implementation Governance is a key part of the architectural process and needs to be formally managed to ensure that the architecture acts as a guide for the implementation teams, but also to ensure that the architecture is clearly understood and followed. The Implementation initiatives are ultimately what transforms an organization from a baseline (current) state to a target (future) state, and ensuring that these initiatives comply with the principles and the designs is critical to the success of the program.

Enterprise Architect has a number of powerful tools to assist with the governance of Implementation initiatives, including the Formal Review facility that can be used to conduct one-off or repeating reviews of projects, ensuring that their level of compliance can be determined. The fact that Strategic, Tactical and Solution Architectures and implementation projects can be managed in the same tool makes the governance process more streamlined. Even if the Enterprise Architecture and the implementation projects are located in separate repositories, content can be imported into the architecture repository for the purpose of making the assessment. Examples of how the principles are applied in the context of each initiative can be modeled using Instances of Principles, providing a useful guidance for implementation teams.

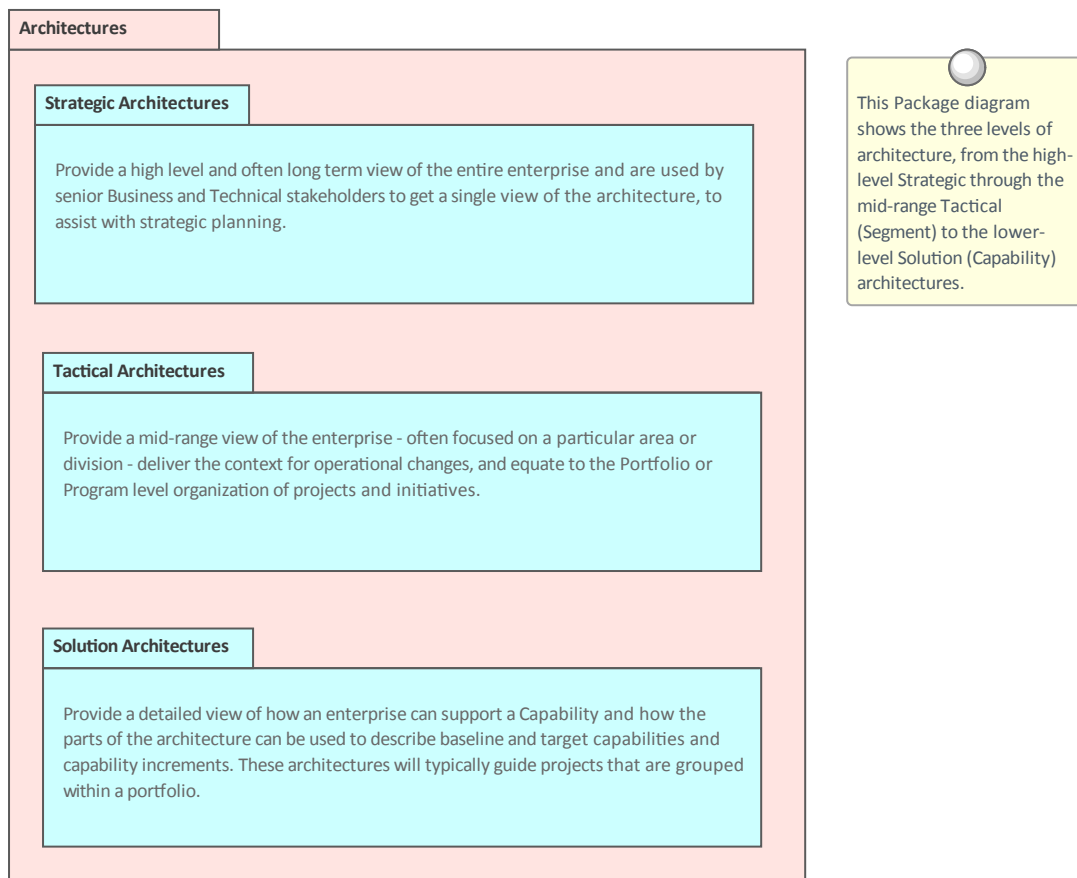
# Levels of Architecture

An Enterprise has a complex and typically hierarchical structure, and you will need to create architectures at discrete levels of this structure. This hierarchy of architectures is analogous to the hierarchies of goals and capabilities, and intuitively aligns with Strategic, Program and Project level divisions. In a small organization it might be possible to create a single architecture that covers the Strategic level and the Project or Capability levels, but in an enterprise of any appreciable size at least three separate levels are typically needed. The naming of the levels has been influenced by The Open Group Architecture Framework (TOGAF).

- Strategic - Long term in the range of 3 - 5 years
- Tactical - Mid term in the range of 1 - 2 years
- Solution - Short term in the range of 6 - 12 months

The different levels of architecture will address different levels of concerns and have different audiences. The architecture framework and the repository must assist in ensuring these architectures are leveled and synchronized so that they form a cohesive and balanced view of the entire enterprise.

Enterprise Architect has a number of powerful features that will assist the architecture program to partition and maintain these levels of the architecture and their inter-relationships.

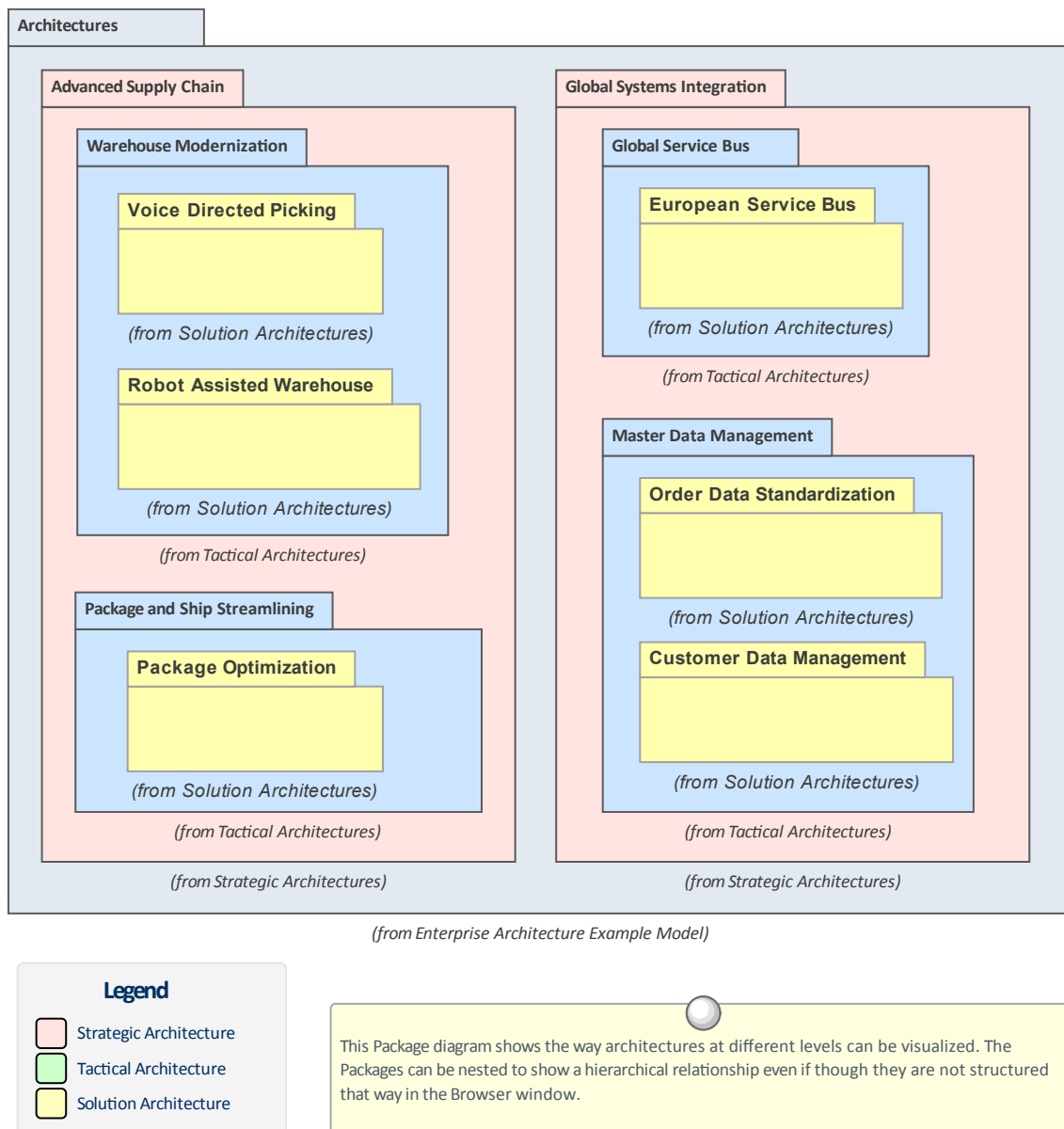


## Strategic

Strategic architectures describe strategic plans and initiatives and typically run for years rather than months. A Strategic architecture will provide a long-range plan that is typically a view of the future over a three to five year period; this period can be longer for industries or enterprises that are not affected by dynamic and disruptive environments. The Strategic architectures must support (or align with) the enterprise's strategic goals, and Tactical and Solution architectures must help achieve the Strategic architecture or run the risk of not being funded.

Enterprise Architect can be used to define and manage

Strategic-level architectures, and can also help to ensure that Tactical and Solution architectures are aligned to support the strategic direction. The Strategic Modeling technology has a number of tools that can be utilized, such as the Balanced Scorecard diagram that can help to identify goals related to Information Technology. There are a number of tools, including the **Relationship Matrix**, diagrams and the **Traceability window**, that can be used to show the relationships between elements of the Business, Information, Application and Technology architectures and to ensure that they all demonstrably contribute to the achievement of the strategic goals.

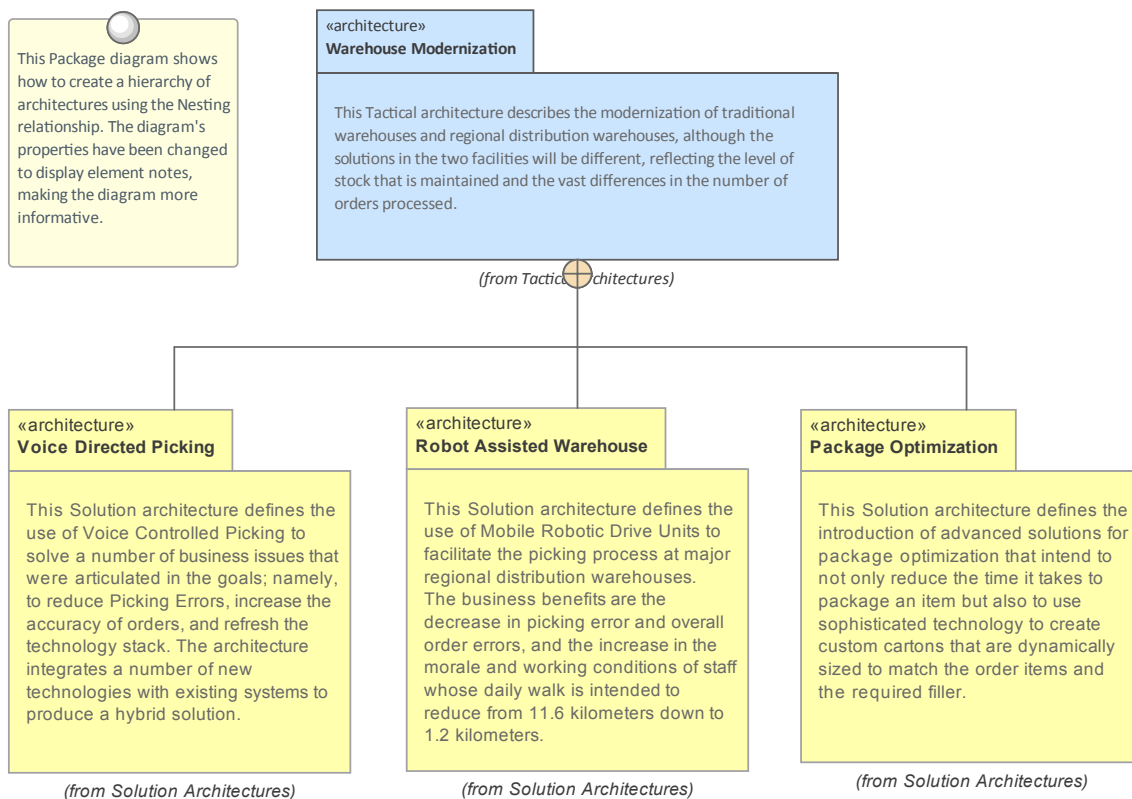


## Tactical

Tactical architectures describe mid-range plans that help to partition the Strategic-level architectures into manageable groups. They typically could run for a number of years and represent a portfolio or program-level plan of how to achieve the goals expressed in the strategic architectures

they relate to. They act as a framework for organizing solution-level initiatives and ensuring that capabilities are developed that ultimately create business value.

Enterprise Architect has tools to support the Tactical-level architectures and to ensure that the partitioning of the Strategic-level architectures - and in turn the partitioning to Solution-level architectures - can be visualized. Roadmap diagrams can be used at all levels of the Tactical architecture, including Business, Information, Application and Technology architectures, showing the time sequencing of the initiatives at a portfolio or program level.



## Solution

Solution architectures describe specific project- or capability-level initiatives that can typically be completed in months rather than years. From a business perspective they are usually focused on a particular problem or opportunity. Similarly, from a technical level they usually involve a slice through Information, Application and Technology areas, but might in some circumstances require a number of these to be addressed.

Enterprise Architect has a rich set of tools that can assist at the Solution architecture level, from defining the business goals and objectives and relating these to information and application components, to the technology devices that underpin the applications. Business Architecture can be defined and managed using stereotypes and profiles to create representations of the Business Drivers, Goals and Objectives, and these can be demonstrated to stakeholders using diagrams, matrices and documentation published automatically from the models. Tools such as the **Schema Composer**, **Database Builder** and **UML Class** diagram will assist with working with the Information architecture, and the elements created can be related to the Business architecture. Application Services, Applications and Interfaces can be modeled, and their relationships to each other and to elements of the Business and Technology architectures can be defined and presented in a variety of visually compelling representations such as diagrams, matrices and lists. The technology services and technology nodes and devices can be managed and, where applicable, they can be derived from the Technical Reference Model.

Enterprise Architect can also be used as the platform for Architectural Requirements Management, which can be related to elements that make up the Business, Information, Application, Technology and other specific architectures. The powerful and flexible Kanban facilities can be used to manage these projects and ensure that the business value is delivered in a timely manner.



# Types of Architecture

The overall architecture of an enterprise can be described by four integrated sub-architectures. These are:

- Business architecture
- Information architecture
- Application architecture
- Technology architecture

Most frameworks describe analogous or similar subsets of an Enterprise Architecture, as the division is based largely on organizational units performing work in these areas.

There are a number of other architectures that could best be described as views, as they cross-cut and are contained by the other architectures, but because of their importance they are often raised to the level of an architecture. These include:

- Security architecture
- Geospatial architecture
- Social architecture

The seven types of architecture are described in this section.

## Business Architecture

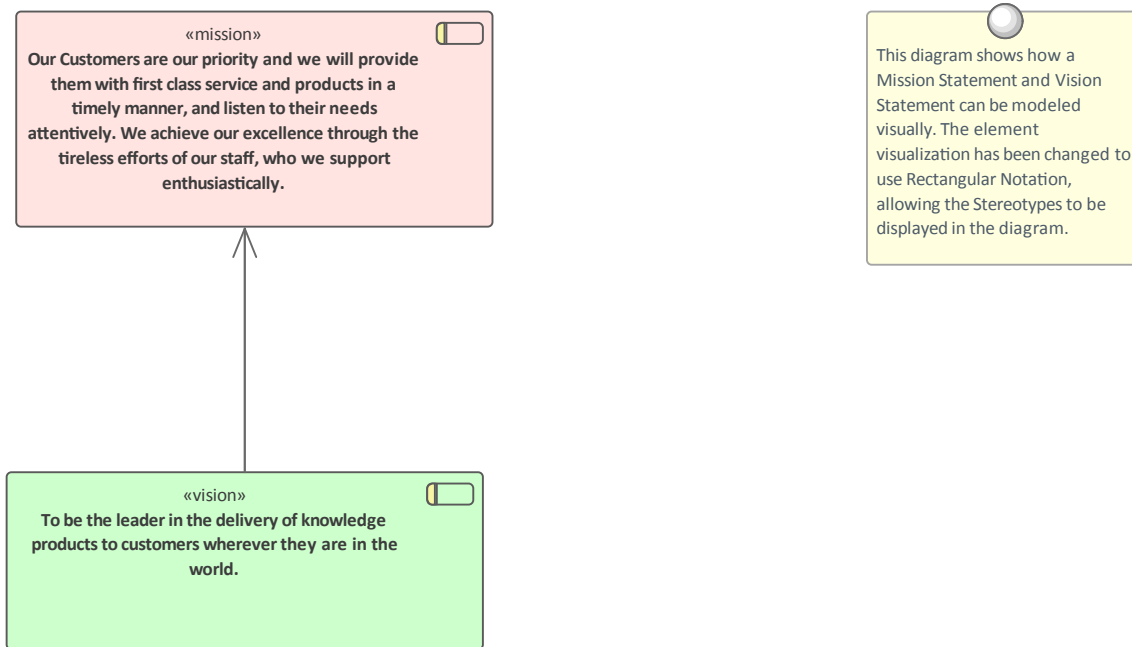
A well articulated Business architecture is the cornerstone of a successful outcome of the overall Enterprise Architecture. It defines the business drivers, the business strategy,

operational models, goals and objectives that the organization needs to achieve to transition in a potentially competitive and disruptive business environment. Architects working in the other architecture disciplines need to understand the Business architecture as the basis for their own architecture descriptions and as a guide to the business outcomes that have to be achieved.

The Business architecture will typically consist of a description of the baseline and target architectures, and definitions of a series of transitions that can be executed and that would be described on Roadmap diagrams.

Enterprise Architect has a wide range of tools that can be used for modeling Business architecture from the Strategic level down to the Operational level. Even when Strategic Plans reside in a corporate document repository, they can be represented and launched from inside Enterprise Architect. This allows other elements such as Drivers, Goals and Objectives to be traced back to their origins in the strategy documents. Mission and Vision statements can also be modeled alongside Business Capabilities, Processes and Functions. Stakeholders and Organizational Roles can be represented and used to show who owns elements of the architecture, such as requirements and applications. Business and Stakeholder Requirements can be created and managed at any level of detail, and the **Document Generator** can be used to create elegant and comprehensive publications that describe aspects of the Business architecture.

Learn More: [Business Architecture](#)



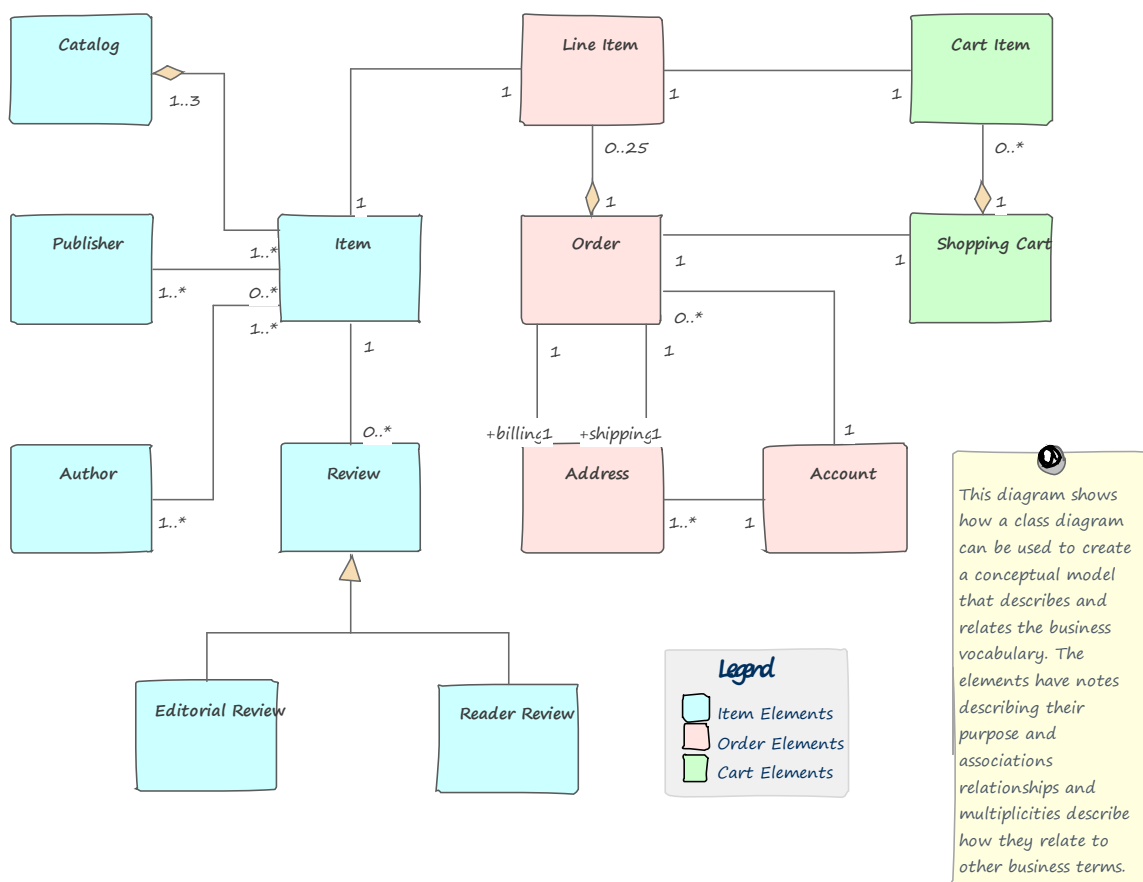
## Information Architecture

Information Architecture is key to the success of an Enterprise Architecture Program, as information is created, consumed and destroyed by the components that make up the other architectures. Understanding which business functions and processes utilize information, which applications serve as the master record, where information is created and destroyed, and which technology components store and manipulate the information, is critical to achieving the business outcomes.

The information architecture will typically consist of a description of the baseline and target architectures, with a series of transitions defined that can be executed and that would be described on Roadmap diagrams.

Enterprise Architect is a profoundly useful tool for creating

and maintaining information architectures, with its sophisticated and extensive support for standards and its wide range of tools to support information models, from high level classifications and concepts right down to the level of schemas and the elements and columns they are composed of. Tools such as the **Schema Composer** and the **Database Builder**, along with the Unified Modeling Language (UML) Class diagram and Glossary, and the powerful Model Transformation facility, will be invaluable. Learn More: [Information Architecture](#)

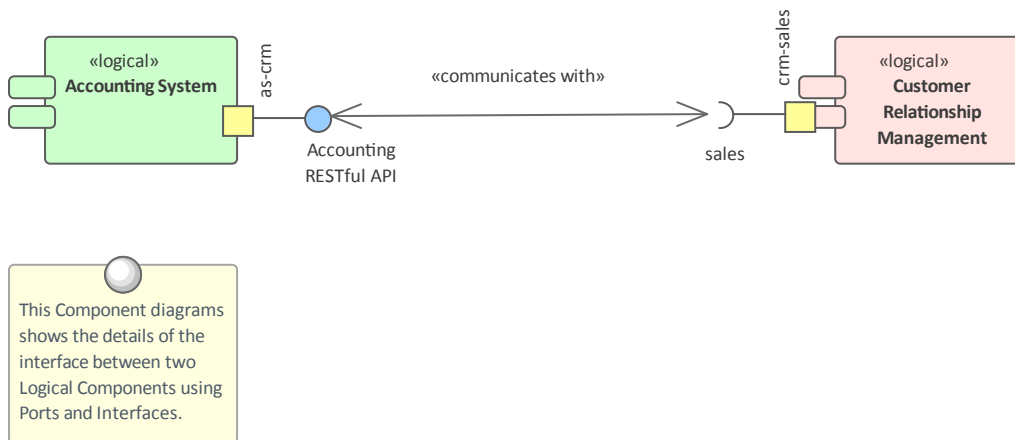


## Application Architecture

The application architecture provides an important catalog of the applications in the enterprise, describing the work that they do to transform, transmit and store information. The architecture also describes the interfaces that are required or provided by the applications and the way the applications interact to carry out the activities described in the business models, such as the Business Process diagrams and Capability models. The catalog of applications, interfaces and the diagrams and matrices that describe their interaction only need to be defined once at the enterprise level. An application architect will be able to draw upon this inventory of existing artifacts to create new architectures, classifying them as part of the baseline and potentially the future state architecture. Where an architecture introduces new applications, these can be added to the description of the target state.

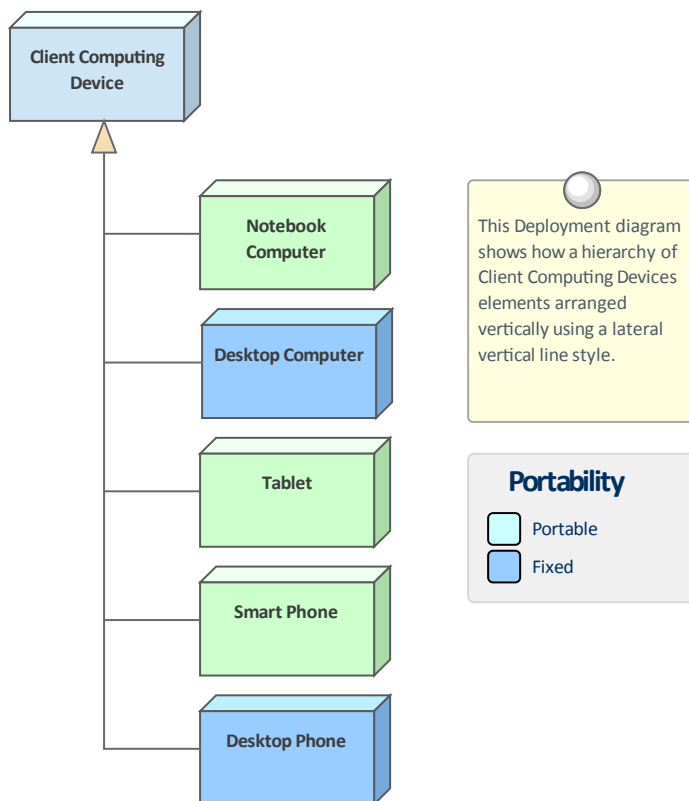
The application architecture will typically consist of a description of the baseline and target architectures, with a series of transitions defined that can be executed and that would be described on Roadmap diagrams.

Learn More: [Application Architecture](#)



## Technology Architecture

The technology architecture underpins the other architectures, providing a description of the logical, physical and virtual infrastructure that supports the execution of application services that in turn support information and business functions and services.



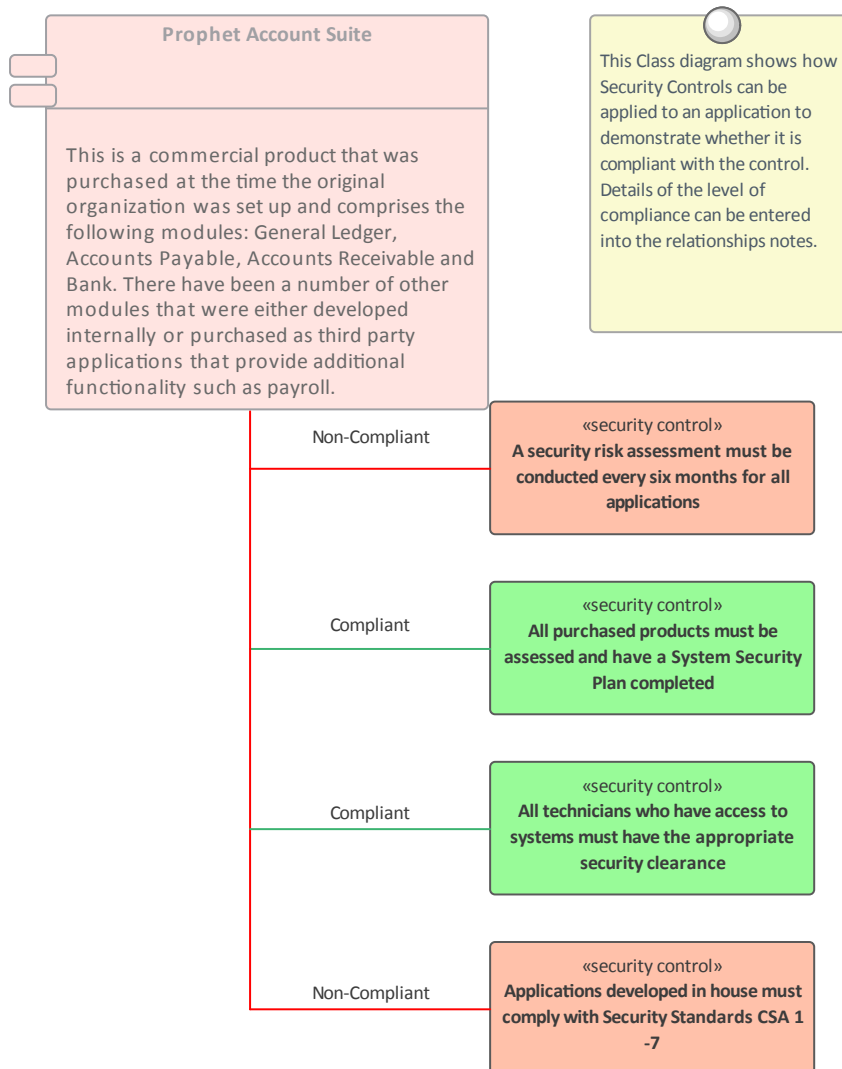
Learn More: [Technology Architecture](#)

## Security Architecture

The security architecture is a slice through all of the other architectures from a security viewpoint. It is listed as a separate architecture because of its importance in ensuring that the enterprise security policies are implemented through the architecture. A breach of security could occur at any point from the business architecture through to the technology architecture. This could include demonstrating how the architectures comply with security controls published by the enterprise or available as part of an

industry compliance regulation.

Enterprise Architect can be used to model the security controls maintained as a list of elements in the repository and when new architectures are created the applicable controls could be applied to aspects of the architecture such as applications, technology devices, communication paths and more. Significant reuse can be achieved by the use of Patterns, and the use of existing building blocks that have already been certified as compliant with the controls.





## Geospatial Architecture

The Geospatial Architecture is a slice through all of the other architectures from a geo-spatial or location-based viewpoint. It is listed as a separate architecture because of its increasing importance in a world dominated by location-based applications and business technology functions. Not every architecture program will have the need to develop separate geospatial architectures and in these cases it can be relegated to a view of the other architectures.

Enterprise Architect is well placed to be a repository for geo-spatial architectures, with its integration with some of the leading standards and tools in the market place, including support for the Geographic Markup Language (GML), ISO 1900 series standards and tools such as ArcGIS and other geodatabases.

## Social Architecture

Social Architecture has been born in an era that depends on social media and social behavior to achieve a wide range of outcomes in built and digital environments. Increasingly in this period of digital and social disruption, understanding the social aspects of an organization's interaction is critical.

This can include the way it interacts with its customers, suppliers and observers and other individuals and communities through what is commonly described as social media.

Enterprise Architect can be used to model the social context of the organization and if significant information is published to social media sites or information is gleaned from these sites these can be modeled in the repository and an architect can create visualizations that show how this information relates to other parts of the business and information architectures.

# Scope of Architecture

Architectures will only be successful if they are scoped correctly. The Enterprise Architecture Body of Knowledge (EABOK) describes three important aspects of scope, but another one can be added that addresses the importance of the stakeholders in the success of the architecture program and the architectures it creates and manages:

- Time Scope
- Organization Scope
- Detail Scope
- Stakeholder Scope

Architecture time frames, organizational context, levels of detail and appeal to stakeholders must all be appropriately set for the architecture to be relevant and successful.

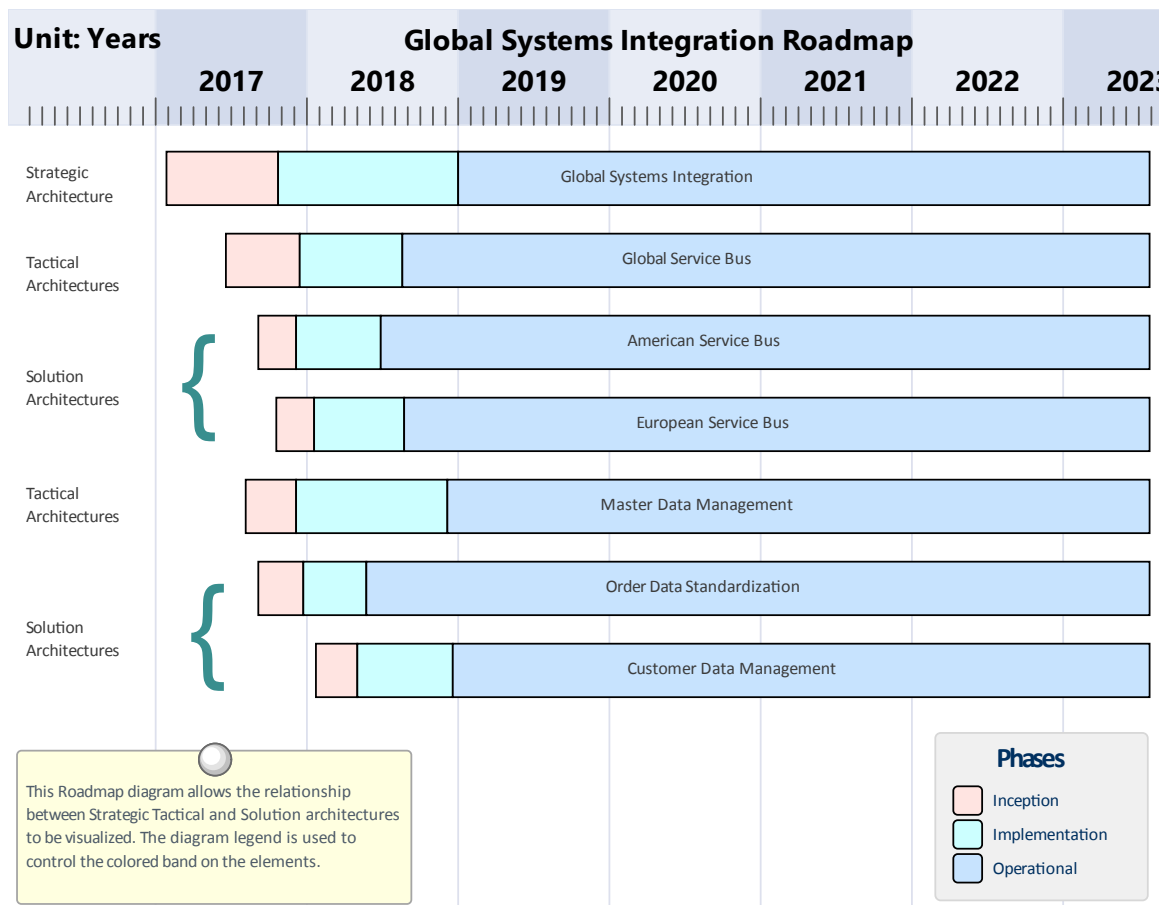
Enterprise Architect has tools that support all of these types of scope, from the powerful Roadmap overlay feature for time modeling, the Organization chart to show which parts of the enterprise will be affected, and the wide range of diagrams and matrices that can be used to allow stakeholders to visualize the architectures at the appropriate level of detail.

## Time Scope

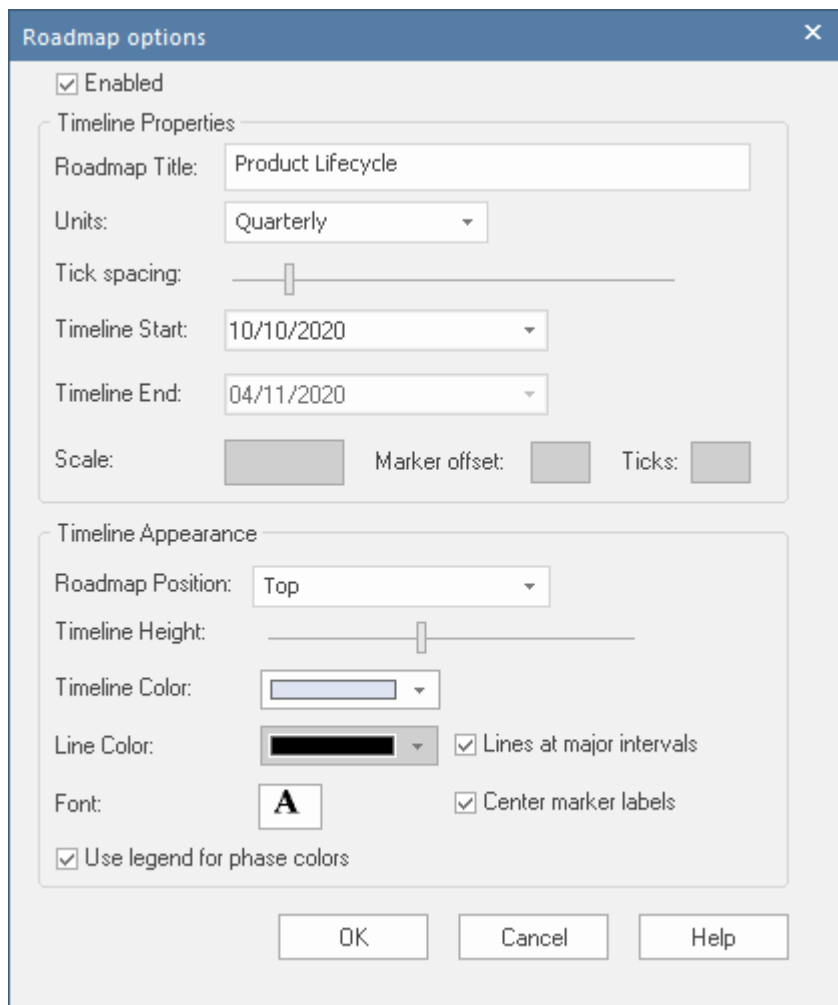
Time Scope is important because the business typically works in cycles, and it is critical that the architectures

respect this time dimension of the organization's management and operation. Strategic plans for medium to large enterprises typically span a period of between three and five years, so it is common for strategic architectures to describe a similar time period, while implementation projects typically run for a period of between three and twelve months. Tactical architectures that group a number of implementation projects can span a period of between one and two years.

Enterprise Architect has some useful features that can help with managing time, including the Roadmap overlay that allows a time scale and extent to be defined and that can indicate the phases any element passes through set against the backdrop of that timescale. The tick spacing can be set from days up to years, allowing any time extent to be represented. Any architectural elements can be represented on Roadmap diagrams, including architectures themselves, principles, capabilities, applications, information, technology devices and more.



The flexibility of the overlay allows any diagram to be converted into a Roadmap, and there is a wide range of settings that can be used to configure the visualization of the Roadmap, including the Diagram Legend, which can be used to define the segmentation of the elements into a series of phases. The Roadmap Options can be used to change the time scale from years, quarters, months, or days down to very fine graduations (used for engineering diagrams). The start and finish times can be set, and the scale can be changed to stretch or collapse the time scale. The position and height of the time ruler can be changed, and fonts and colors can all be configured to make the diagrams more appealing.



The screenshot shows a 'Roadmap options' dialog box with a close button (X) in the top right corner. The dialog is divided into two main sections: 'Timeline Properties' and 'Timeline Appearance'.

**Timeline Properties:**

- ☒ Enabled
- Roadmap Title: Product Lifecycle
- Units: Quarterly
- Tick spacing: A slider control.
- Timeline Start: 10/10/2020
- Timeline End: 04/11/2020
- Scale: A color selection box.
- Marker offset: A color selection box.
- Ticks: A color selection box.

**Timeline Appearance:**

- Roadmap Position: Top
- Timeline Height: A slider control.
- Timeline Color: A color selection box.
- Line Color: A color selection box (currently black).
- ☒ Lines at major intervals
- Font: A font selection box (currently 'A').
- ☒ Center marker labels
- ☒ Use legend for phase colors

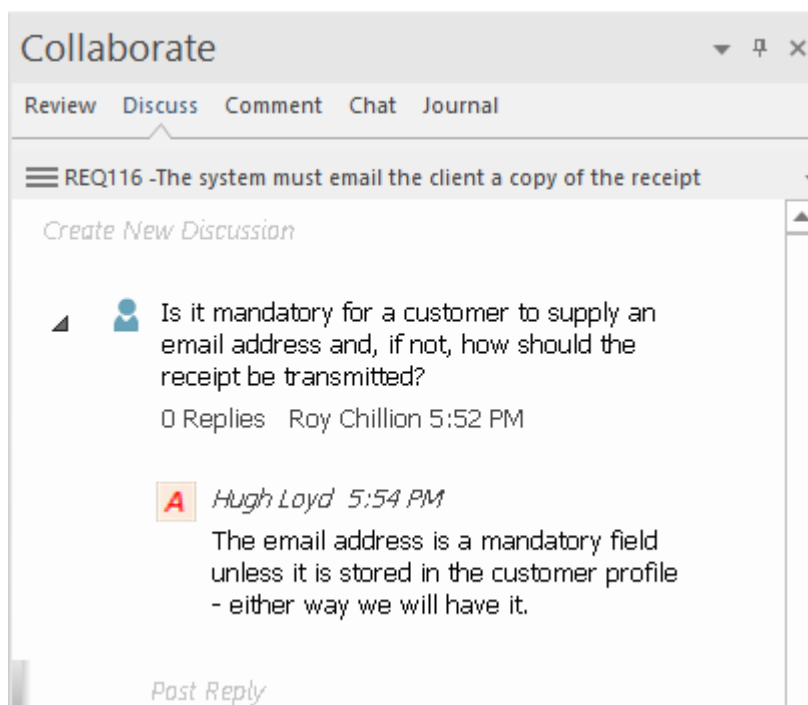
At the bottom of the dialog are three buttons: OK, Cancel, and Help.

## Detail Scope

Selecting the correct level of detail for an architecture is critical to its success; this is particularly true when it comes to the Implementation teams. Creating architectures that are too lofty or aspirational will result in Implementation teams making important design decisions themselves which, while they might be appropriate for their solution, might not be the best outcome for the entire enterprise. On the flip side of this argument, creating architectures that are too prescriptive and detailed can constrain an Implementation team and

result in the team not having the flexibility to select the best solution.

Enterprise Architect is a tool based on the concept of collaboration, and there are many facilities that will help the architecture team members work with each other and with all stakeholders, including implementation teams, to determine the most appropriate level of detail for the architectures. The **Team Library** facility allows in-model reviews to be created, where elements from the architectures such as Goals, Objectives, Applications, Technology nodes and more can be dragged in as references for the reviews. The Discussion and **Discussion Review** windows allow architects and stakeholders to deliberate about the architectures and the consequent implementations. The Diagram Filtering facility and a wide range of tools for changing the visualization of elements in diagrams allow the appropriate level of detail to be set for the architectures and the views that are created for stakeholders.

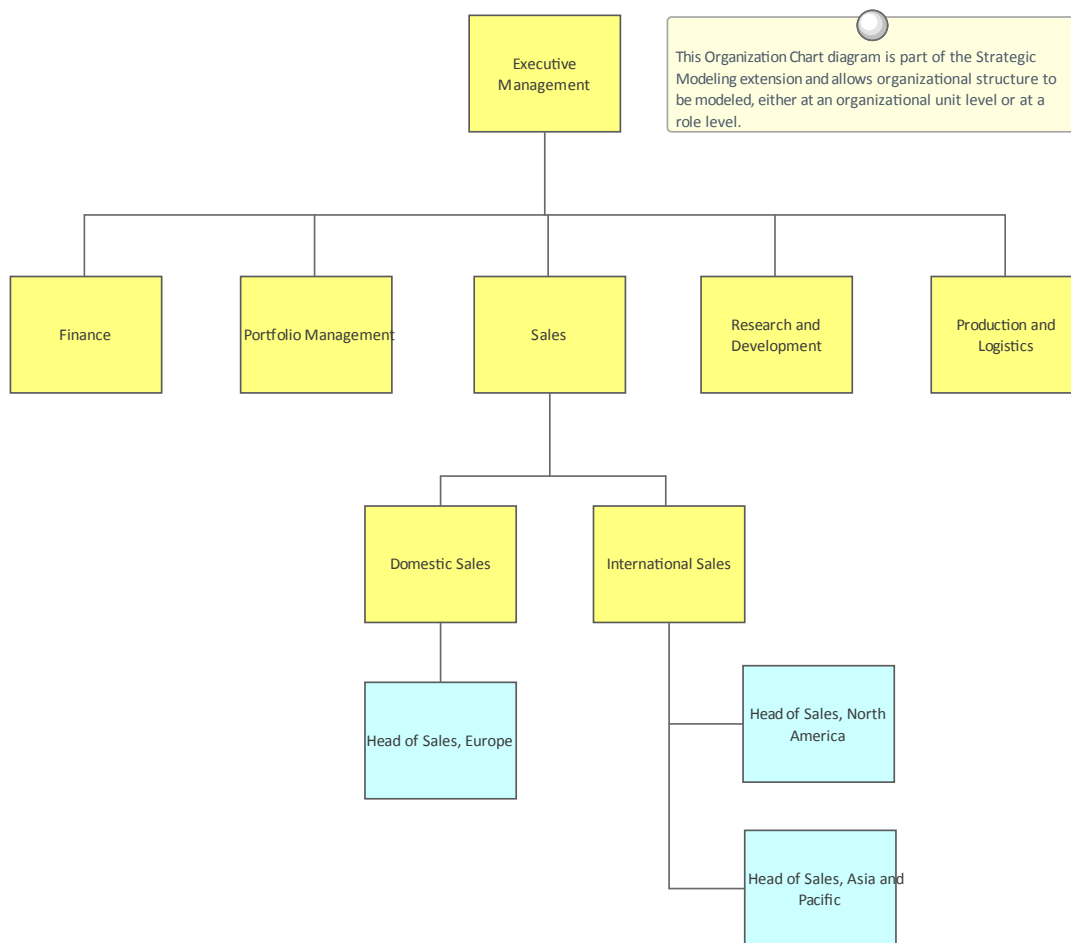


## Organization Scope

Enterprise Architecture is a non-trivial and costly discipline, and it is critical that value is delivered to the business. The best outcomes will be achieved if the architecture touches all parts of the enterprise, but it is quite common for some parts of the enterprise to receive greater emphasis in the architectural descriptions than others. Having a clear understanding of the structure of an enterprise and its organizations and how strategic plans relate to this structure is critical for the success of any enterprise architecture effort.

Enterprise Architect has an Organizational Chart within the Strategic Modeling Technology, which can be used to model the structure of an enterprise and its organizations. The architectures can be related to this structure, which allows the organizational scope to be visualized.



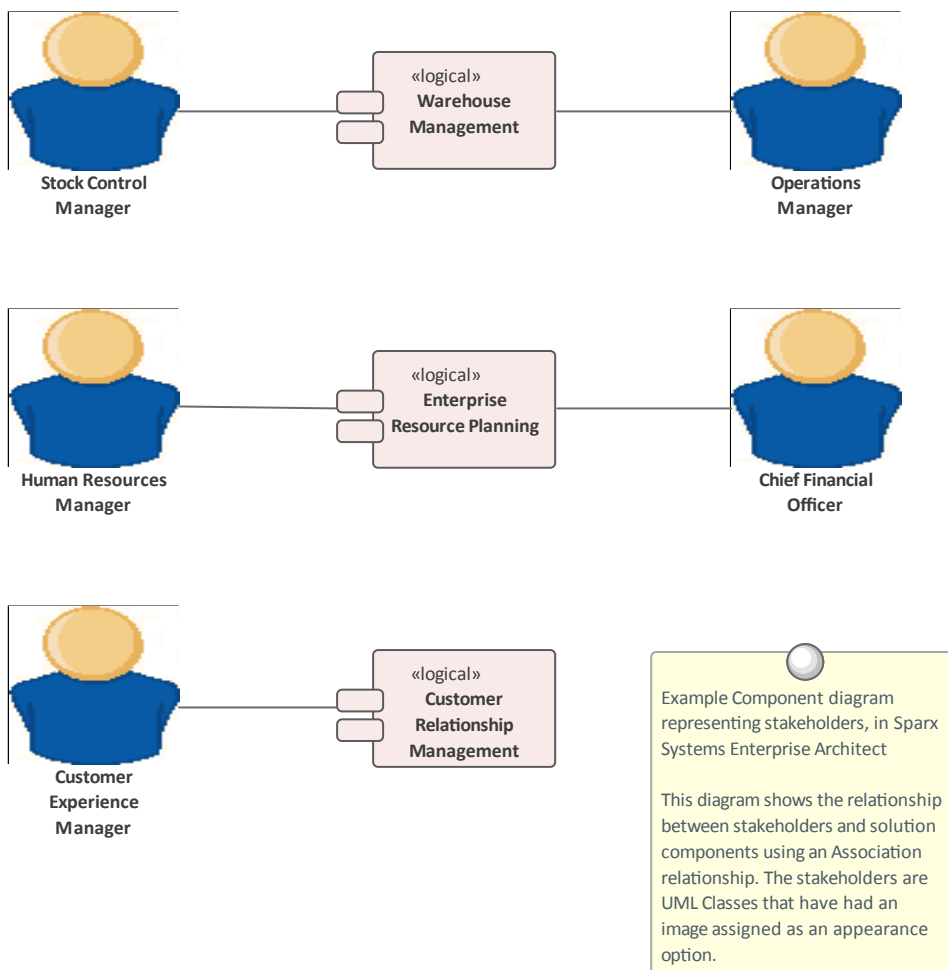


## Stakeholder Scope

The stakeholders and the shareholders or organization owners that they represent are the ultimate beneficiaries of the enterprise architecture, and it is important that the right stakeholders are selected and communication is managed to ensure they are kept updated with the progress of the architectural work and the governance of the Implementation Initiatives.

Enterprise Architect has a number of facilities to ensure that the stakeholder scope is determined and that the

architectures are created in accordance with what these individuals or groups require. The stakeholders themselves can be modeled inside the tool and their relationship to elements such as Drivers, Goals, Objectives, Applications and Architectural Requirements can be maintained. This allows impact analysis to be visualized, so that when changes occur that affect any of these elements, the stakeholders who have an interest in the change can be determined. The visualization can be through diagrams, matrices or lists of elements and can be viewed directly in the model, or publications can be generated in a variety of formats including PDF, DOCX and Web Pages.



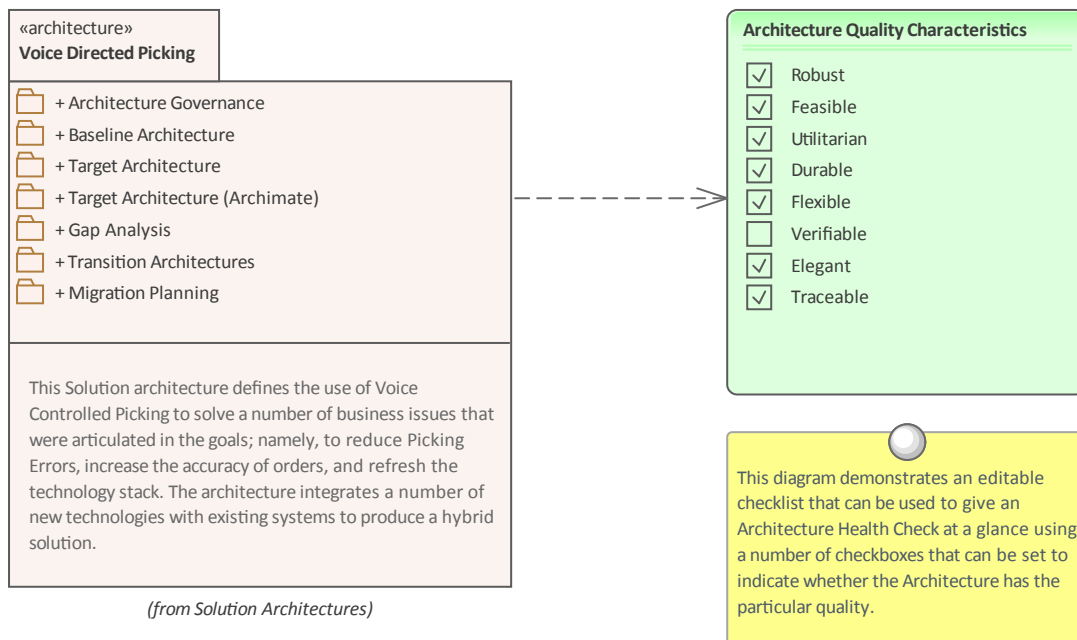


# Characteristics of Good Architecture

It is difficult to define what the characteristics of good architecture are when there is still vigorous and lasting debate about what the term 'architecture' actually means in the context of enterprise systems in the Twenty-First Century. The Roman architect Vitruvius defined three characteristics of good architecture in his treatise *De Architectura* more than 2,000 years ago. Interestingly it is the only surviving text from antiquity describing architecture. These principles are:

- Durability (Firmatis) – It should stand up robustly and remain in good condition
- Utility (Utilitas) – It should be useful and function well for the people using it
- Beauty (Venustatis) – It should delight people and raise their spirits

These ancient characteristics can be elaborated on and expanded to apply to the Enterprise Architectures that are developed in the Twenty-First Century.



## Qualities of Good Architecture

To be effective, an architecture must have a number of qualities or characteristics. Enterprise Architect provides an extensive set of features and tools for helping the Architect produce architectures that are of high quality. This table contains some of the most important qualities, with a description of how Enterprise Architect can be used to ensure the qualities are built into Architecture created and maintained in the tool.

Quality	Description
Robust	An architecture should be strong and not be vulnerable to minor changes in the

	<p>business, information, application and technology systems. Enterprise Architect can assist in ensuring that the architectures are well integrated and related to each other and provides a number of tools such as the <b>Traceability window</b>, the <b>Relationship Matrix</b> and the <b>Insert Related Elements</b> feature that can be used for this purpose</p>
Feasible	<p>An architecture that cannot be implemented will mean that the goals and objectives of the enterprise will not be met. It is best to identify these requirements as quickly as possible so as not to disappoint the party who requisitioned the architecture work. Enterprise Architect can assist by allowing architects, designers and developers to discuss the architecture and determine its feasibility using the <b>Collaborate window</b> and by mapping the Enterprise Architecture to Capability or Solution Architectures.</p>
Utilitarian	<p>An architecture must have utility which, in turn, when implemented will result in practical outcomes. Architectures that are elegant but do not provide demonstrable and measurable value to the stakeholders</p>

	<p>or the parties that requisitioned them will ultimately not be successful. Enterprise Architect has tools that allow an architecture to be visualized and understood by a diverse group of stakeholders, allowing any problems with utility to be discovered early in the architecture process.</p>
Durable	<p>An architecture is a living entity that describes a target state and - once implemented - will become the new baseline state. The architectures should prove to be durable with the passage of time and be resilient to changes in the business and technical environments that might occur over the lifetime of the architectures. This implies that they must - as much as possible - pre-empt the future conditions and environments.</p>
Flexible	<p>The architectures must be flexible and be able to adapt to changing conditions and also provide enough guidance for implementation teams that have the knowledge of their discipline to make the important and necessary decisions about technical problems and opportunities. Architectures that are created with too much detail will often result in brittle and</p>

	<p>inflexible designs and implementations resulting in systems that cannot adapt to changing circumstances and environments. Enterprise Architect has a wide range of features that can assist with change including the Change element a powerful Baseline facility and the Kanban diagrams that allow Requirements Features, User Stories and more to be visualized and prioritized.</p>
Verifiable	<p>It should be possible to verify that the architecture will perform as designed and that there would not be side effects that result from the architecture and the parts of the enterprise that it impacts. The ultimate test of this is whether it delivers the business value that was promised in the Vision Statement. Enterprise Architect can be used to model the measures that are defined to verify that the Business Objectives (and therefore the Goals) have been attained.</p>
Elegant	<p>Architectures must have both form and function and it is a good test of an architecture to measure its elegance. An architecture that is well designed will</p>



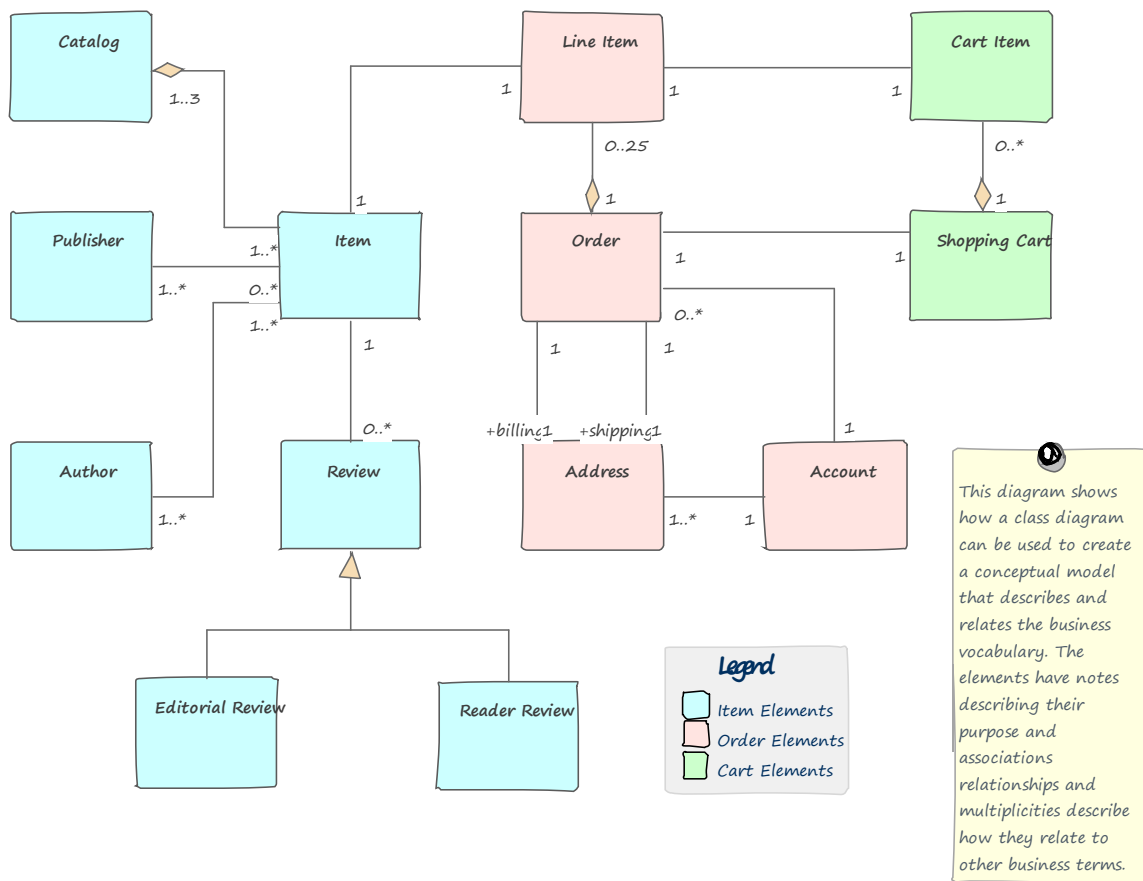
	<p>tend to be elegant and have a simplicity of form that will be obvious to those that take the time study it. Enterprise Architect has powerful features that allow the elegance of an architecture to be visualized including the ability to create professional publications that can be generated automatically from the tool using a series of built-in or user defined templates.</p>
Traceable	<p>An architecture is a description of the an enterprise at a particular level of detail and does not exist in isolation but is typically related to business drivers and goals and other architectures at the same level or higher or lower and to implementation programs and projects. Enterprise Architect allows elements to be traced in any direction and provides a number of powerful tools to visualize the traces including the <b>Relationship Matrix</b>, the Traceability Window and diagrams. The <b>Insert Related Elements</b> facility can be used to automatically construct a diagram of traces almost magically creating expressive and never before seen views of the repository.</p>



# Lists, Diagrams and Matrices

Lists, Diagrams and Matrices are the three main ways of presenting architectural information to stakeholders. These three representations can be used in isolation or together to provide a rich communication of the architectures; they can also be tailored to suit individuals or groups of stakeholders. They are often combined in repeatable sets called Views that provide consistent, coherent and relevant information to an audience.

Lists provide a simple catalog of items that can be displayed in a tabular format, where relevant properties and metadata can be viewed for each item and compared between items. Diagrams are a graphical projection of items connected as a graph and provide a compelling visual representation of the elements and their relationships. Matrices are grids that show the relationship between two sets of items from a particular point of view; they are a powerful visual device because they allow gaps and overlaps to be identified easily. Enterprise Architect has a wide range of tools that support all three of these representations, with many extended features such as searching, sorting, layout, filtering, alternative images and simulations that allow an architect to create visually compelling representations of the architectural content.



## Lists

Enterprise Architect has a number of tools for working with elements in a list. These tools can be applied to any type of element, including Principles, Business Drivers, Requirements, Applications, Interfaces, Technology Devices and more. The **Specification Manager** is a powerful tool that can be used to create and view any type of element in a visually appealing, word processor-type or spreadsheet-type format. The properties of the listed elements can be displayed and any number of properties can be added or removed from the display, including Tagged

Values. The properties can be edited for each row of the list, including selecting values from drop-down lists such as selecting a status. Filters appear below the header of each column and can be applied to restrict the display to elements that meet a particular condition. Changing any of the details of an element, including its name and description, will change it in every other part of the repository, including the **Browser window** and all diagrams in which it appears.

Item

## 1 REQ019 - Manage Inventory

The system **MUST** include a complete inventory management facility to store and track stock of books for the on-line bookstore.

### 1.1 REQ122 - Inventory Reports

Inventory reports are required that detail the available stock for each item including back orders. Future stock level reports should be able to predict the quantity of stock at a specified future date.

### 1.2 REQ023 - Store and Manage Books

A book storage and management facility will be required.

#### 1.2.1 REQ022 - Order Books

A book order facility will be required to allow on-line ordering from major stockist's.

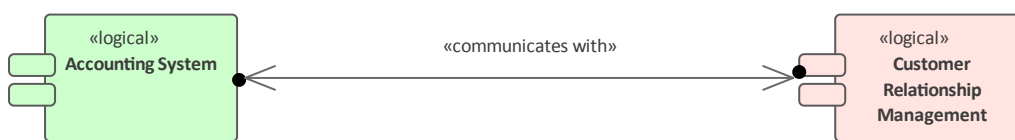
#### 1.2.2 REQ021 - List Stock Levels

A facility will exist to list current stock levels and to manually update stock quantities if physical checking reveals inconsistencies.

## Diagrams

Diagrams are an expressive and visually compelling way of

presenting information and can be carefully constructed to deliver a particular message. Diagrams can be used to communicate with stakeholders to provide a view of the architecture from a particular viewpoint. The same elements can appear in multiple diagrams and color, styles, filters, layout, alternative images and more can be used to convey meaning and to create compelling diagrams that will help stakeholders engage with the architecture. The diagrams can be converted to a hand drawn style and a white board mode to create further appeal and to soften the edges of the formal modeling languages that can deter some stakeholders.



The same diagram can be altered to show the detail of the interface that has been rolled-up in the previous diagram. This is a powerful facility that allows repository content to be automatically altered to create alternative views for different stakeholders.



This Component diagrams shows the details of the interface between two Logical Components using Ports and Interfaces.

## Matrices

The **Relationship Matrix** and Gap Matrix are grids that allow the relationships between two sets of element to be visualized in a matrix format with one set of elements on the horizontal axis and the other on the vertical axis. Markers at the intersection of a column and a row indicate information about the relationship between the two elements.

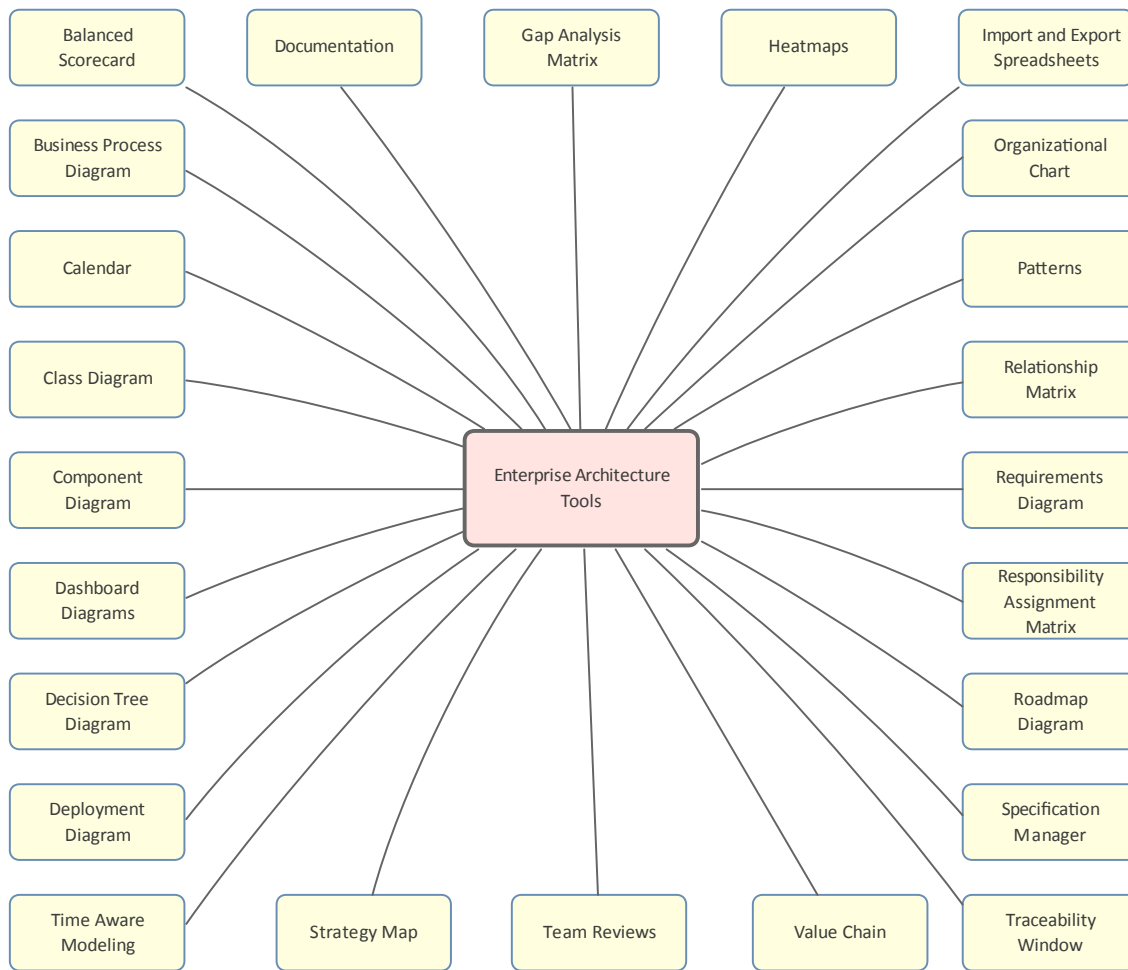
Relationship Matrix			
Source:	Customer Relationship Management	Type: Requirement	Link Type: Realization
Target:	Customer Relationship Management	Type: Component	Direction: Target -> Source
		Target +	
		Capture CRM Hosted Service	
		Customer Manager	
		Summit CRM	
+ Source			
REQ153 - The solution must allow customers to securely manage their own contact information		↑	↑
REQ154 - Customers must be able to use the solution without the need for training or help		↑	↑
REQ156 - The solution must be able to manage leads from initial enquiry through to a customer order		↑	
REQ157 - All user interfaces in the solution must be web based and not require additional browser plugins.		↑	
REQ158 - The solution must allow users to create ad-hoc reports with out the need for scripting.		↑	

# Meet the Enterprise Architecture Tools

Enterprise Architect is a sophisticated and flexible Enterprise Architecture platform that can be used as both an architecture repository and a tool for managing architecture projects. It can be used across the entire life cycle from setting up an architecture program or office to planning, managing, developing and documenting architectures through to the governance of implementation projects that consume the architectural output. The tool can be used with any single or combination of Architecture Frameworks, processes and languages of representation. There is a wide range of facilities and tools that allow the architect to work using their preferred methods such as Word Processor views, Spreadsheet views, Diagrams, Relationship Matrices or a range of other core and extended features.

This Mind Map shows the landscape of the key Enterprise Architecture tools that can be used to set up and maintain the architecture office and to plan, create, manage and document architectures. While these are the primary tools there are a series of other tools described in the Help topic *Additional Enterprise Architecture Tools*.

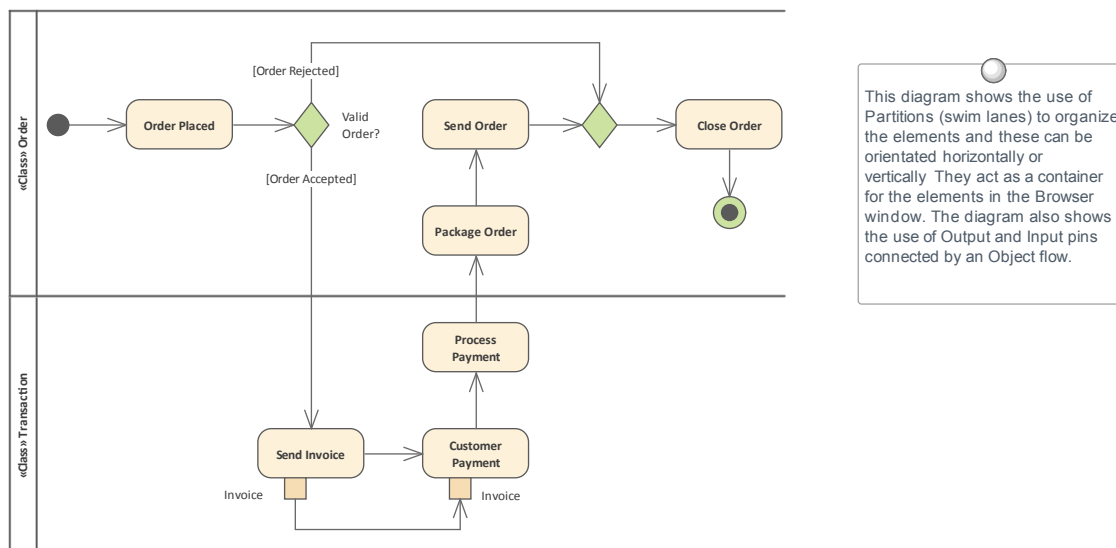




# Activity Diagram

The Activity diagram is one of the Unified Modeling Language (UML) Behavioral diagrams that can be used to model a process or algorithm as a sequence of steps. It is a more sophisticated version of its close cousin the Flowchart diagram. Activity diagrams can be used to model Business Processes as a UML alternative to the **BPMN Business Process** diagram; they have the same ability to create a hierarchy of Activities in the **Browser window**.

Activity Diagram showing the use of Partitions



The elements can be given a name and detailed descriptions can be added to the Notes. By connecting the Activities, Decisions and Forks with connectors (Control Flows) a sequence of elements can describe the business process. A process hierarchy can be constructed by nesting Activities in the Browser window and using the child diagram functionality to enable drill down from the value chain level to the lowest level processes.

## Getting to know the Activity Diagram

**Where to find the Activity Diagram** Ribbon: Design > Diagram > Add > UML > Behavioral > UML Behavioral > Activity  
**Browser window Toolbar** : New Diagram icon > UML > Behavioral > UML Behavioral > Activity  
**Browser window context Menu** | Add Diagram... > UML Behavioral > Activity

**Usage of the Activity Diagram** The Activity diagram can be used to model any business or technical activity or notion that has a series of steps. This includes business and technical processes and also computer algorithms. The steps are connected by Control Flow relationships that show the sequencing of the steps. Decisions and Merges can be used to model choice and to further control the flow through the Activity. Forks and Joins can be added to split and reunite the flow of control and objects added to show how data is supplied and consumed.

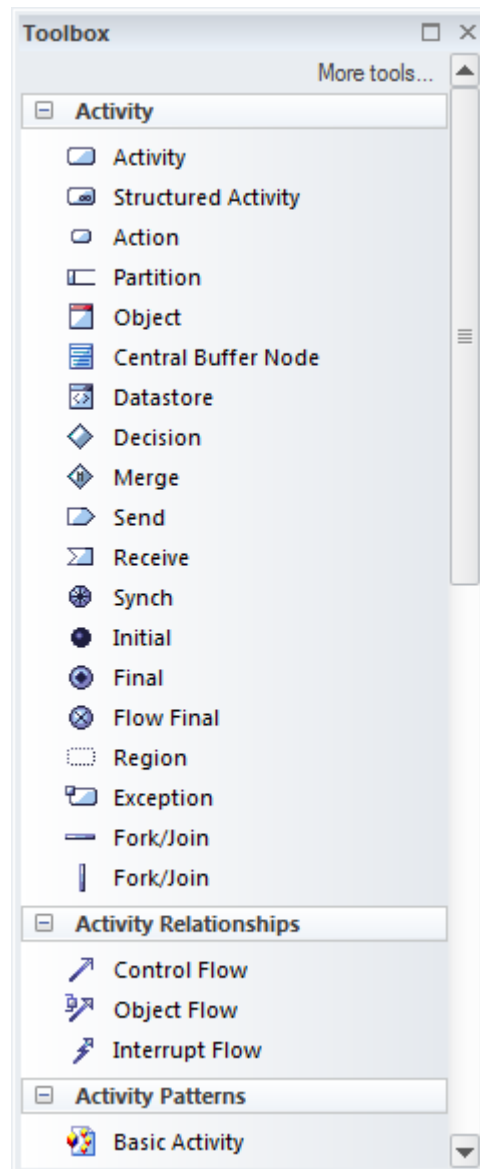
**Options for** Activity diagrams can be drawn at

**the Activity  
Diagram**

different levels of formality, from a Basic Flow Chart style of diagram used to represent a simple Business Process to a sophisticated Action-based diagram that can be used to model a complex system. There is a toolbox that contains a range of elements, relationships and Patterns for creating the models.

The Activity diagram (like any diagram) can be viewed as an Element List, which makes working with element properties easier.

Diagram Filters can also be used when presenting the diagrams, to draw attention to parts of the diagrams, and the diagrams can be presented in hand drawn or whiteboard style by changing the properties of the diagram.



**Learn more  
about the  
Activity  
Diagram**

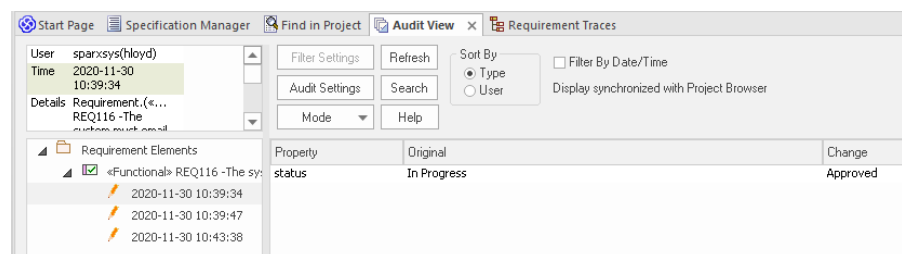
[Activity Diagram](#)

# Auditing

## Getting to know Auditing

### Introducing Auditing

The **Auditing** feature can keep track of the changes to Requirements including what was changed, when it was changed and by whom. Auditing is by default disabled and must be enabled before the changes to requirements will be recorded. Once enabled it is a passive tool that silently records the changes to elements. It does not replace **Version Control** or **Baselines** and in contrast to these tools it can not be used to return to a previous state of the model. Change management, governance and quality control are all aided by the use of Auditing.



### Where to find Auditing

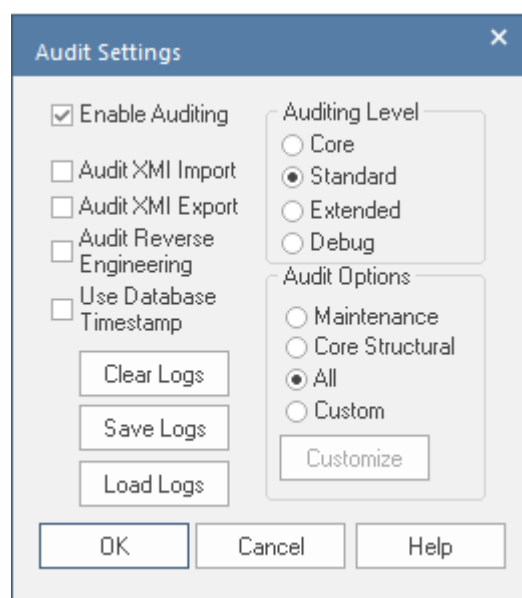
Ribbon: Configure > Model > **Auditing**

## Use of Auditing

**Auditing** can be used to track what was changed in a model, who changed it and when. There are a number of modes and a repository administrator can use the settings to specify what is recorded in the audit. While a baseline can be used to show the difference between a model and a snapshot at a point in time, the Auditing tool records each individual change; it can not, however, be used to revert to a previous state.

## Options for Auditing

There is a wide range of settings to configure auditing, starting with enabling or disabling the settings that determine which elements have an audit trail and the level of detail recorded. Audit logs can be exported from the repository to increase performance.

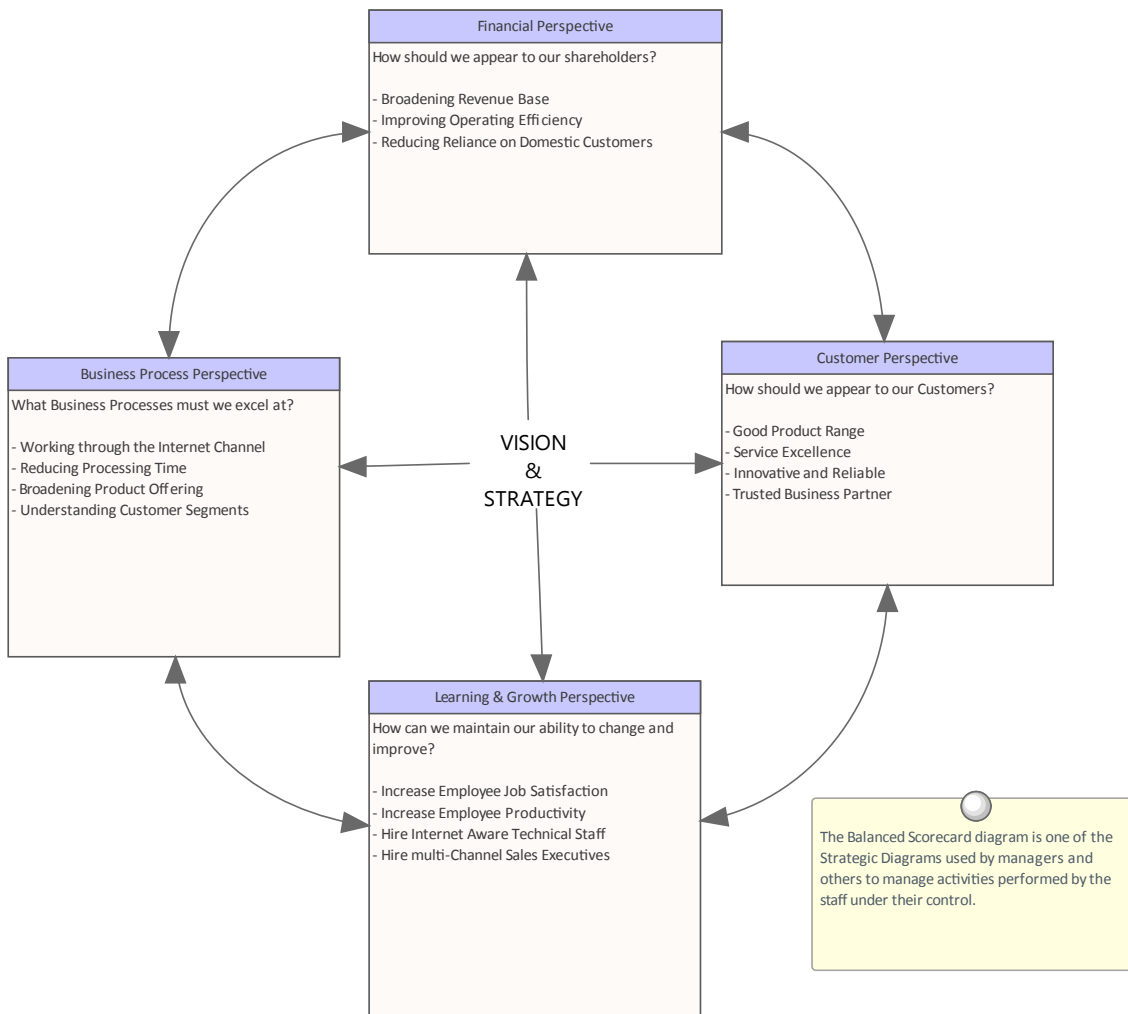


**Learn more** [Auditing](#)  
**about**  
**Auditing**



# Balanced Scorecard

Balanced Scorecard is a strategic diagram that allows a balanced approach to performance measures to be modeled. The diagram can be created from a Pattern that adds and connects the four perspectives: Financial, Customer, Internal Business Process, and Learning and Growth. Most organizations of any appreciable size will be using some type of Balanced Scorecard approach to align business activities to the vision and strategy of the organization, to monitor performance against strategic goals and to improve communication. The Balanced Scorecard diagram is part of a set of strategic diagrams that allow many aspects of an organization's business to be modeled.



## Getting to know the Balanced Scorecard

### Where to find the Balanced Scorecard

Ribbon: Design > Diagram > Add > Strategic Modeling > Balanced Scorecard

Browser window Toolbar: New Diagram icon > Strategic Modeling > Balanced Scorecard

Browser window context menu | Add Diagram... > Strategic Modeling >

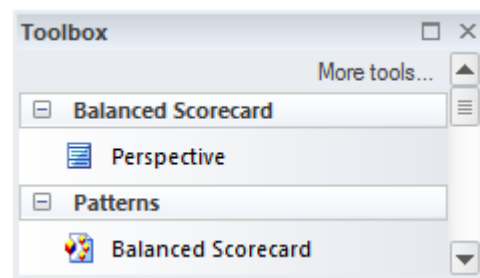
## Balanced Scorecard

### Usage of the Balanced Scorecard

A Balanced Scorecard can be used to ensure business activities are aligned to the vision and strategy of the organization, to monitor performance against strategic goals and to improve communication.

### Options for the Balanced Scorecard

The Classes representing the four **Perspectives** can be traced to other elements in the models, such as Goals and Business Capabilities, and also to Linked Documents that allow the Perspectives to be documented.

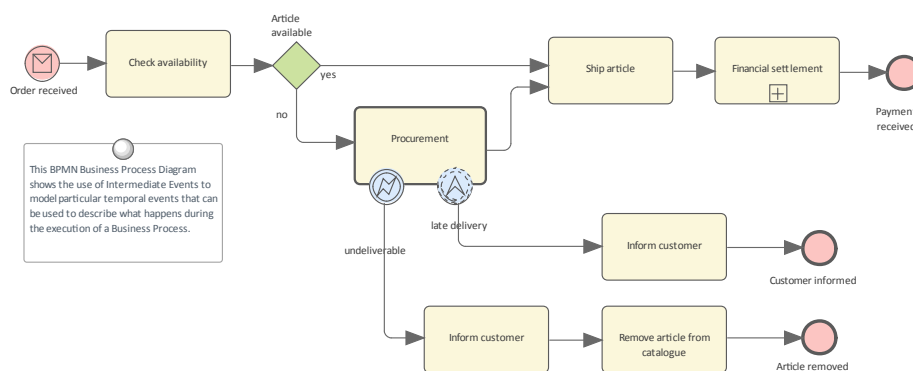


### Learn more about the Balanced Scorecard

[Balanced Scorecard](#)

# Business Process Diagram

The Business Process diagram is part of the Business Process Model and Notation (BPMN) and is the main diagram type for defining business processes. The diagram can include Start, Intermediate and End Events, Business Processes, Activities, Gateways, Pools and Lanes, and more. The elements can be given a name and detailed descriptions can be added to the notes. By connecting the flow objects with connectors, a sequence of activities, gateways and events can describe the business process.



A process hierarchy can be constructed by nesting Business Processes and Activities in the **Browser window** and using the child diagram functionality to enable drill down from the value chain level down to the lowest level processes.

## Getting to know the Business Process Diagram

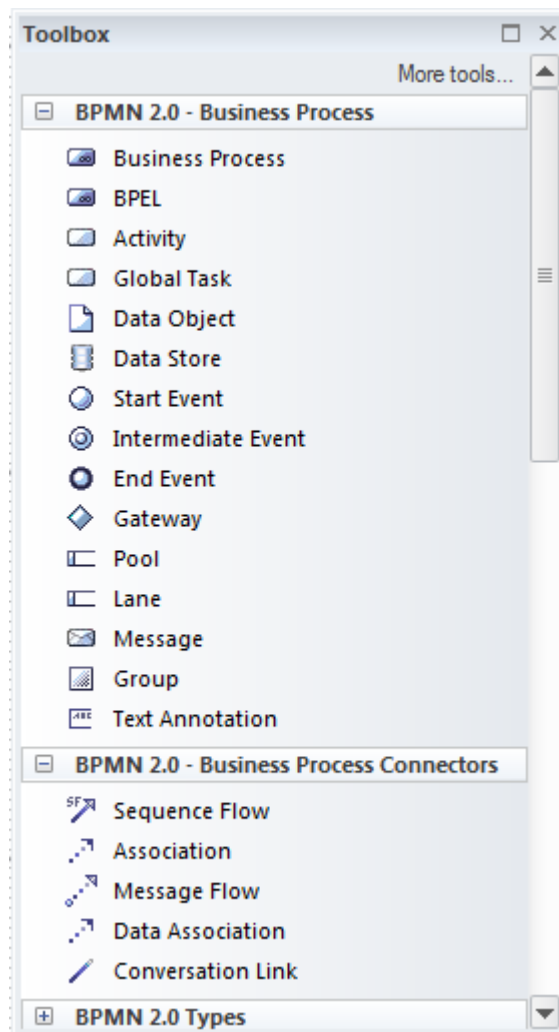
**Where to** Ribbon: Design > Diagram > Add >

**find the Business Process Diagram** BPMN x.y > Business Process  
**Browser window Toolbar : New Diagram icon > BPMN x.y > Business Process**  
**Browser window context menu | Add Diagram... > BPMN x.y > Business Process**

**Usage of the Business Process Diagram** Business Process diagrams can be used to model the Business Processes in an entire organization or part of an organization. Business Processes can be created to represent the current or future state of the organization at any level of detail from the value chain down to a process performer level. A process hierarchy would typically be defined that would set the standards for the number of levels, naming, organization of the processes and more.

**Options for the Business Process Diagram** Business Process diagrams can be drawn at different levels of formality, from a Basic Flow Chart style of diagram used to represent a simple Business Process to a sophisticated diagram making use of many of the markers for Events and Activities to describe complex business processes. There is a toolbox that

contains a range of elements, relationships and Patterns for creating the models.



The Business Process diagram can also be used to generate Business Process Execution Language (BPEL), which is an XML language that can be ingested by a number of tools.

The Business Process diagram (like any diagram) can be viewed as an Element List, which makes working with element properties easier.

Diagram Filters can also be used when presenting the diagrams to draw attention to parts of the diagrams.

**Learn more  
about the  
Business  
Process  
Diagram**

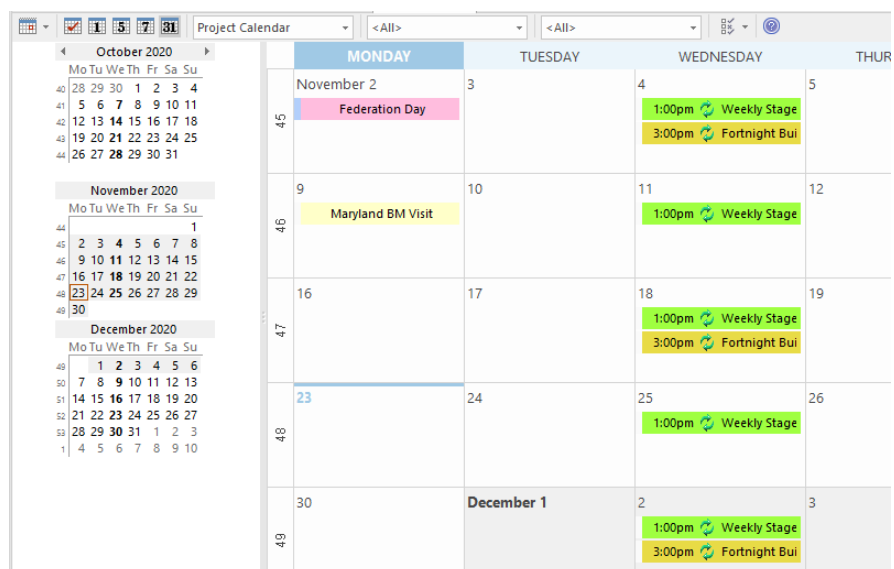
[Business Process Diagram](#)

# Calendar

## Getting to know the Calendar

### Introducing the Calendar

The **Calendar** is a fully featured mechanism for recording the important events in an initiative and displaying other information such as resource allocation. There are day, week and month views and the display can be set to show Calendar entries, **Project Tasks** and Resource Allocation. When a resource has been allocated - for example to analyze a set of requirements - a user can drill through from the Calendar to the requirements' location in the **Browser window**.



There are also fully configurable Event



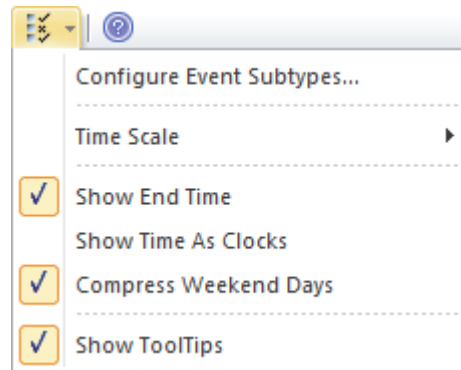
Types, Categories and colors. The work of a Business Analyst will involve a wide range of events including things like: workshops, interviews, focus groups, collaborative games, brainstorming sessions, reviews, observations and meetings. All of these events can be conveniently recorded and managed in the Calendar. When resources have been allocated to elements and tasks have been assigned to individuals these can be displayed in the Calendar.

**Where to find the Calendar** Ribbon: Start > Collaborate > **Calendar**

**Usage of the Calendar** The **Calendar** can be used to schedule and view events such as meetings, milestones, reviews, workshops and more. It can be used to view the allocation of resources to elements in the repository such as who is analyzing a set of requirements. It can also be used to view **Project Tasks**. An analyst can conveniently click through to the elements in the **Browser window** or the **Project Tasks**.

## Options for the Calendar

The **Calendar** has a number of options including the ability to create recurrent events. There is an options toolbar icon that allows aspects of the Calendar's appearance to be configured.



**Learn more  
about the  
Calendar**

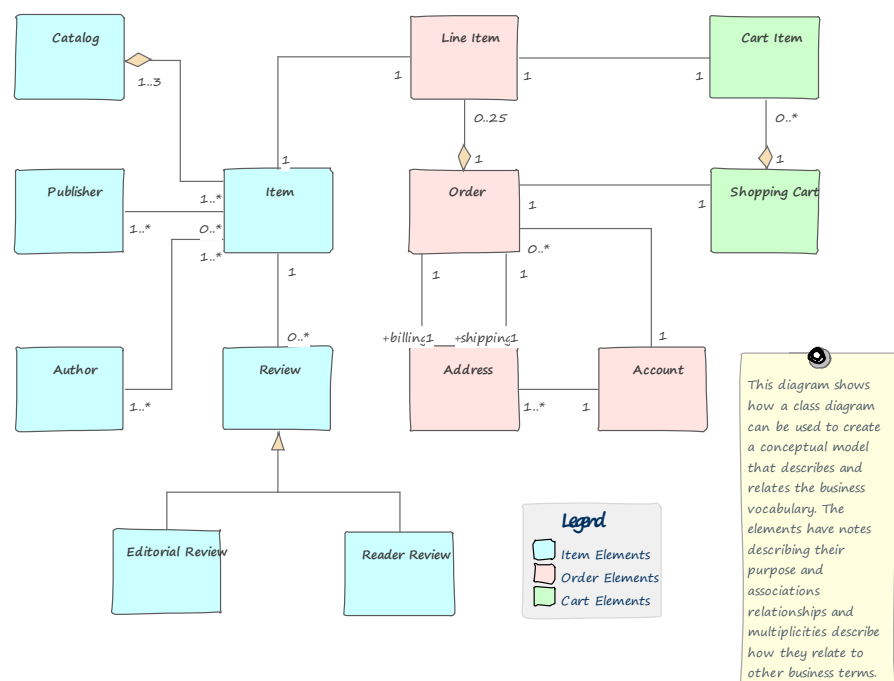
[Project Calendar](#)

# Class Diagram

## Getting to know the Class Diagram

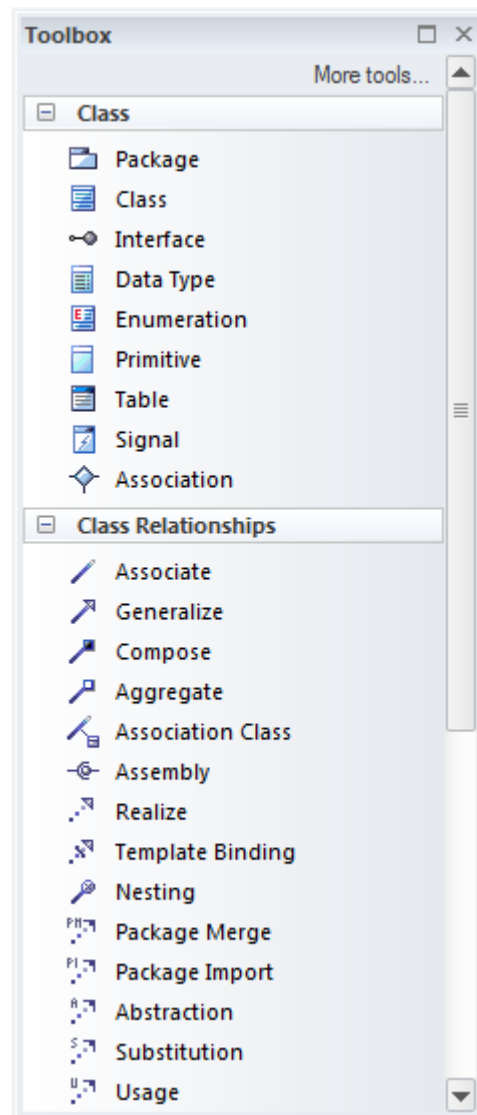
### Introducing the Class Diagram

The Class diagram is one of the Unified Modeling Language (UML) Structural diagrams that can be used to model a wide range of things. It is a general purpose diagram for modeling entities in the business and technical domains, including terms and concepts, Business Rules, and Capabilities in XML and Database Schemas.



**Where to** Ribbon: Design > Diagram > Add >

<b>find the Class Diagram</b>	UML Structural > Class <b>Browser window</b> Toolbar : New Diagram icon > UML Structural > Class Browser window context menu   Add Diagram... > UML Structural > Class
<b>Usage of the Class Diagram</b>	The Class diagram can be used whenever a logical or structural representation of a system is required. It has applicability for modeling both business and technical concepts and can be used to model information and structures such as XML and database schemas.
<b>Options for the Class Diagram</b>	The Class diagram (like any diagram) can be viewed as an Element List, which makes working with the element's properties easier. Diagram Filters can also be used when presenting the diagrams to draw attention to parts of the diagrams and the diagrams can be presented as hand drawn or in a whiteboard style by changing the properties of the diagram.



**Learn more  
about the  
Class  
Diagram**

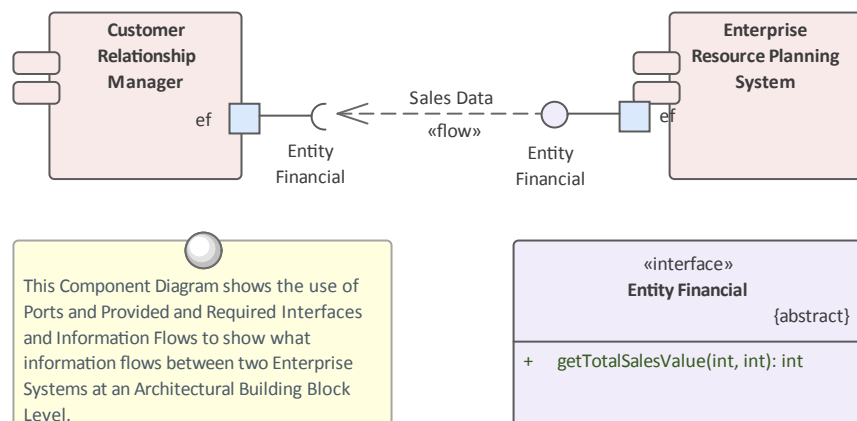
[Class Diagram](#)

# Component Diagram

## Getting to know the Component Diagram

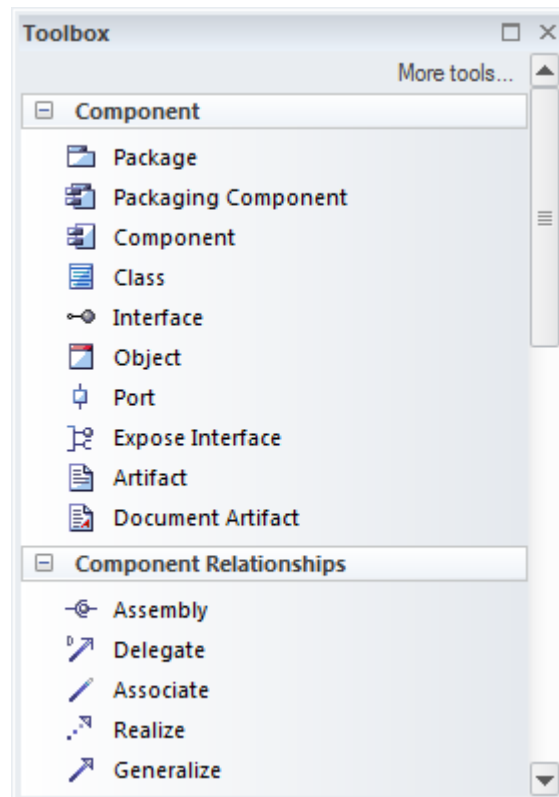
### Introducing the Component Diagram

The Component diagram is one of the Unified Modeling Language Structural diagrams, and it can be used to model the logical components that make up a system. It can be used to model the applications of an organization, including their Provided and Required Interfaces and the information that is exchanged between the interfaces.



Component hierarchies can be created that show how top levels systems or applications are broken down to lower level Components. The Components can be given a name, detailed descriptions can be added and additional properties can be added using Tagged Values.

<b>Where to find the Component Diagram</b>	<p>Ribbon: Design &gt; Diagram &gt; Add&gt; UML Structural &gt; Component</p> <p><b>Browser window</b> Toolbar : New Diagram icon &gt; UML Structural &gt; Component</p> <p>Browser window context menu   Add Diagram... &gt; UML Structural &gt; Component</p>
<b>Usage of the Component Diagram</b>	<p>The Component diagram can be used to model logical or physical parts of a system including current and future state Components. Application Architectures often use a number of Component diagrams to describe the architecture of the applications and how they interact. The interaction between the applications can be shown using a combination of Ports and Provided or Required Interfaces that describe how the Components are wired together.</p>
<b>Options for the Component Diagram</b>	<p>Component diagrams can be drawn at a number of levels of formality, from simple diagrams that show the dependencies between Components to sophisticated diagrams using Ports, Interfaces and Information flows.</p>



The Component Toolbox pages contain a range of elements, relationships and Patterns for creating Component diagrams.

The Component diagram (like any diagram) can be viewed as an element list that makes working with element properties easier.

Diagram Filters can also be used when presenting the diagrams, to draw attention to parts of the diagrams, and the diagrams can be presented in hand drawn or whiteboard style by changing the properties of the diagram.

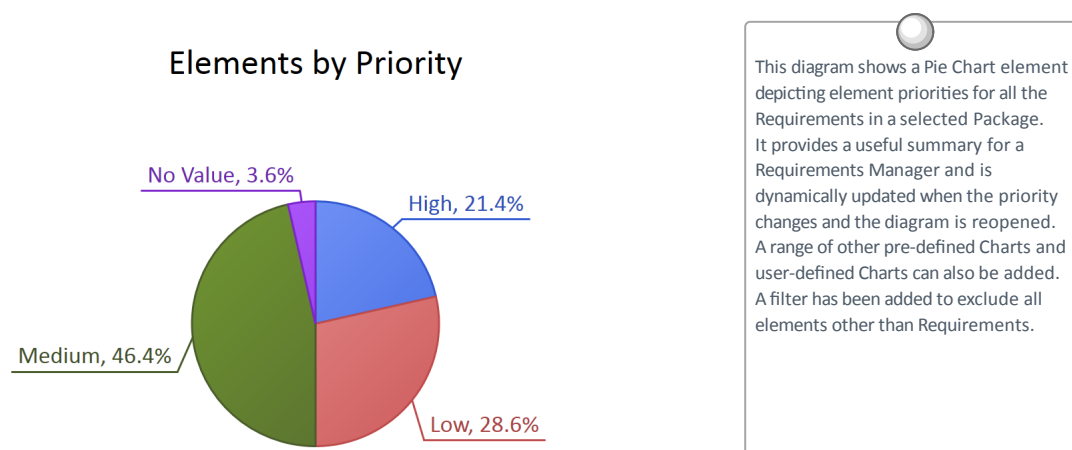
**Learn more** [Component Diagram](#)



# **about the Component Diagram**

# Dashboard Diagrams

Dashboard diagrams allow you to create high quality Charts and graphs to display repository information in a visually compelling way. This diagram is an example of creating a Dashboard **diagram** in Sparx Systems Enterprise Architect; it illustrates the ratio of Requirement Priorities in a Pie Chart.



Enterprise Architect provides a Toolbox page of pre-configured Charts and graphs, but you are free to create and save any number of Charts, sourcing data from anywhere in the repository. The Charts and graphs provide valuable summary information that assists in the management of Requirements. High level reporting and project status can be easily tracked and documented using the numerous Charts and report elements available, which tightly link in with the model content and status.

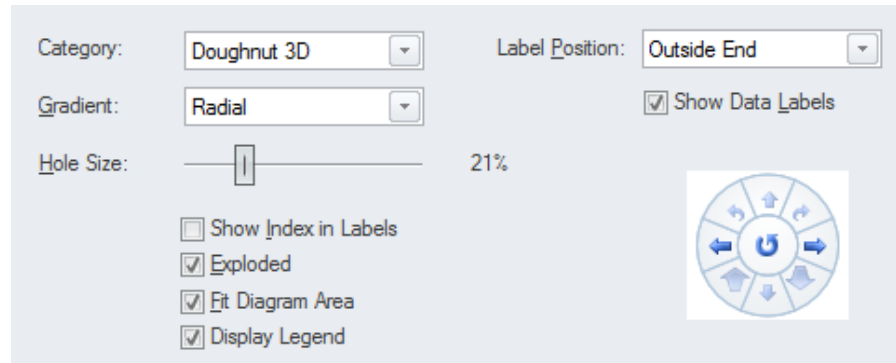
# Getting to Know Dashboard Diagrams

**Where to find Dashboard Diagrams**     **Browser window** | Right-click on Package | Add Diagram :  | Manage | Show All Perspectives | Extended | Dashboard

**Usage of Dashboard Diagrams**     Dashboard diagrams present rich yet easily understood views of information - such as the status of Requirements in a particular release of the system - that can be opened inside the model or conveniently copied directly into management or project team presentations. They are useful for planning an iteration such as an Agile sprint to view how ready the Requirements are for the implementation team; for example, to view what percentage of the Requirements have been approved and are of high priority.

**Options for Dashboard Diagrams**     The standard Charts and graphs available from the Toolbox can be configured in a number of ways, including changing the source, applying filters or modifying the appearance of the Chart as indicated in this diagram, available from the Chart's

## Properties window using the 'Appearance' section.



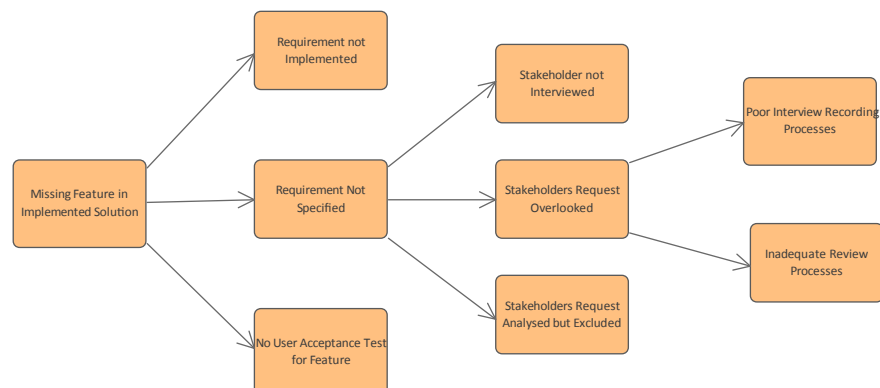
**Learn more** • [Standard Charts](#)  
**about**  
**Dashboard**  
**Diagrams**

# Decision Tree Diagram

## Getting to know the Decision Tree Diagram

### Introducing the Decision Tree Diagram

Decision Trees are an effective way of graphically representing a number of options and provide a mechanism to investigate the possible outcomes and benefits of choosing those options. They can also assist the analyst to form a balanced picture of the risks and benefits associated with each possible course of action. They are a close cousin of the **Decision Table** but have the benefit of being graphical. Enterprise Architect has a purpose-built diagram allowing complex decisions to be modeled and displayed including probabilities and uncertainty.



**Where to find the Decision Tree Diagram** Ribbon: Design > Diagram > Add > Strategic Modeling > Decision Tree  
**Browser window Toolbar** : New Diagram icon > Strategic Modeling > Decision Tree  
Browser window context menu | Add Diagram... > Strategic Modeling > Decision Tree

**Usage of the Decision Tree Diagram** Decision Trees can be used to help in decision making processes, particularly when the decision involves a complex set of conditions that have different likelihoods of occurrence. They can be used for strategic or operational decision analysis and can help to formalize the basis of decision making particularly when it is imperative that actions that are taken are based on formal analysis or have expensive consequences. A Decision Tree can be used to present a graphical picture of a **Decision Table** for stakeholders who are more comfortable viewing diagrams rather than tables and documents.

**Options for the Decision Tree** Decision Trees can be drawn with varying levels of formality from simple trees with a series of decisions resulting

**Diagram** in outcomes to more formal trees that involve uncertainty with probability values assigned or formulaic expressions with input parameters. The 'Decision Tree' toolbox page contains a range of elements that can be used, and two Patterns that can be used to create a diagram giving the analyst a starting point.

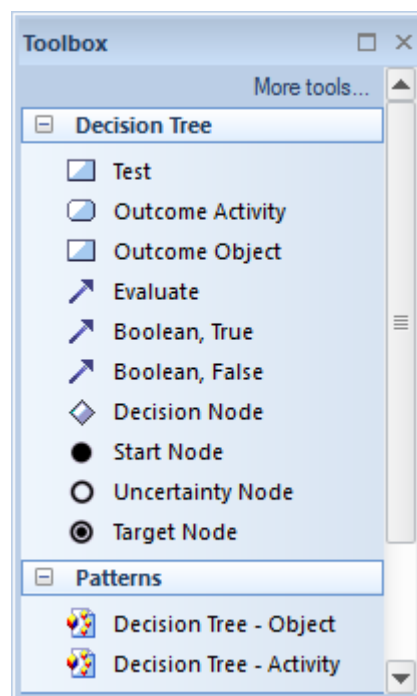


Diagram Filters can also be used when presenting the diagrams to draw attention to parts of the diagrams and the diagrams can be presented as hand drawn or in a whiteboard style by changing the properties of the diagram.

**Learn more  
about the**

[Decision Tree Diagram](#)

# **Decision Tree Diagram**

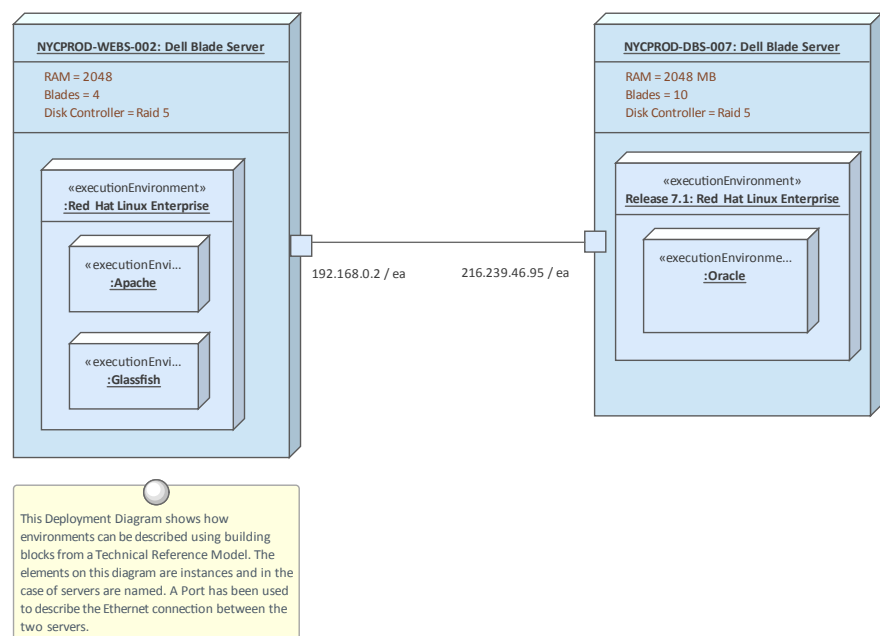


# Deployment Diagram

## Getting to know the Deployment Diagram

### Introducing the Deployment Diagram

The Deployment diagram is one of the Unified Modeling Language (UML) Structural diagrams that can be used to model infrastructure, including a wide range of deployment environments. Both physical and virtual environments can be modeled, including infrastructure and networking services and protocols.

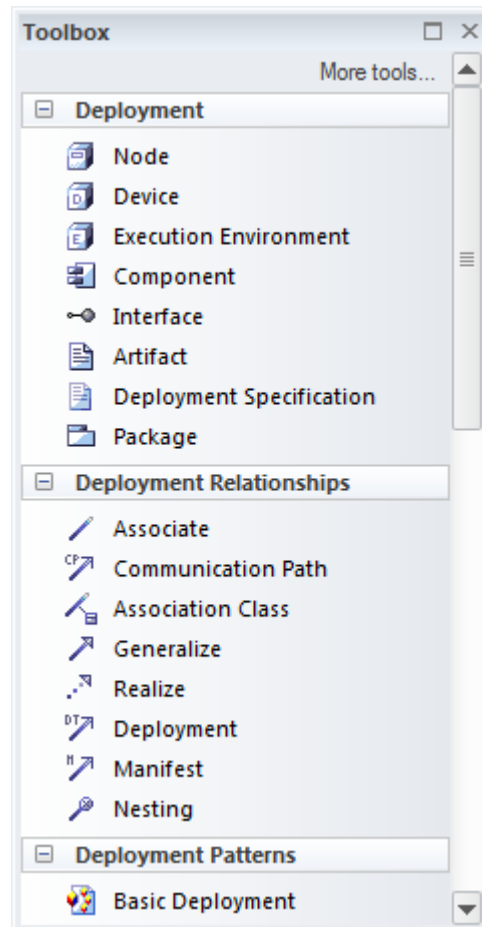


### Where to find the Deployment

Ribbon: Design > Diagram > Add >  
UML Structural > Deployment  
Browser window Toolbar : New

<b>Diagram</b>	Diagram icon > UML Structural > Deployment Browser window context menu   Add Diagram... > UML Structural > Deployment
<b>Usage of the Deployment Diagram</b>	The Deployment diagram can be used to model a wide range of infrastructure and networking environments including physical and virtual environments. It is quite common to model a number of different environments including: production, pre-production, test and development environments. The diagram can be used to model Servers, Devices, Networks, Execution Environments. Switches, Routers, IP Addresses and the software components and Artifacts deployed.
<b>Options for the Deployment Diagram</b>	It is quite common to use the Deployment diagram to model a number of different environments, including production, pre-production, test and development environments. Alternative images can be used in diagrams to represent infrastructure and networking devices, such as Networks, Servers, Routers, Switches and more, making the diagrams more appealing to their users.

The diagrams can be included in documentation generated automatically using the **Document Generator**.



The Deployment Toolbox pages contain a range of elements, relationships and Patterns for creating Deployment diagrams.

The Deployment diagram (like any diagram) can be viewed as an element list, which makes working with element properties easier.

Diagram Filters can also be used when presenting the diagrams, to draw attention to parts of the diagrams; the

diagrams can be presented in hand drawn or whiteboard style by changing the properties of the diagram.

**Learn more  
about the  
Deployment  
Diagram**

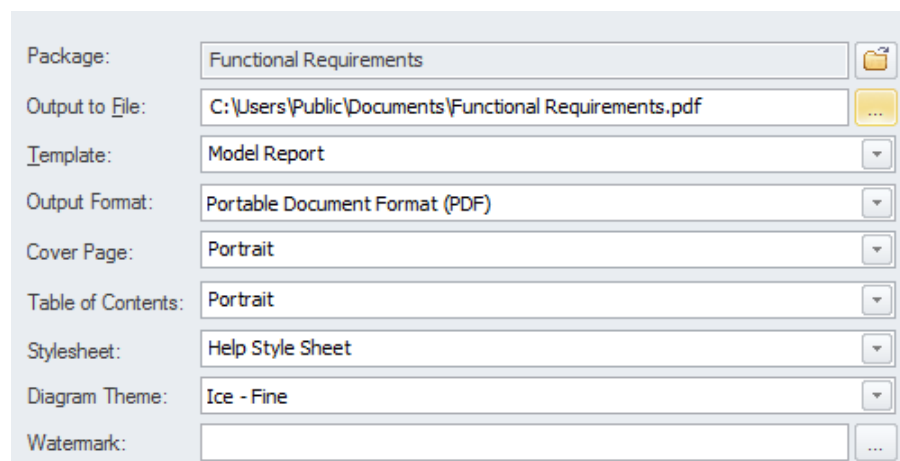
[Deployment Diagram](#)

# Documentation

## Getting to know Documentation

### Introducing Documentation

The Documentation features can be used to automatically generate a wide range of documentation directly from the models. These can be document-based such as PDF and Docx format or HTML-based. Flexible templates can be used to completely tailor the documents that are generated including company logos, tables of content, tables of element information and diagrams. Ad-hoc reports can also be created from a number of tools such as the Glossary and the Search Window.



The screenshot shows a dialog box for generating documentation. It contains several fields and dropdown menus:

- Package:** Functional Requirements
- Output to File:** C:\Users\Public\Documents\Functional Requirements.pdf
- Template:** Model Report
- Output Format:** Portable Document Format (PDF)
- Cover Page:** Portrait
- Table of Contents:** Portrait
- Stylesheet:** Help Style Sheet
- Diagram Theme:** Ice - Fine
- Watermark:** (empty field)

**Where to** Ribbon: Publish > Model Reports >

**find** Report Builder  
**Documentati**  
**on**

**Use of** Modelers, Analysts, Architects, Project  
**Documentati** Managers and others can use the facility  
**on** to produce a wide range of  
document-based publications and  
reports, such as a System Requirements  
Specification, Use Case Report, Data  
Dictionary, Solution Architecture  
Description and more. It can also be used  
for ad-hoc reporting to create reports  
such as a list of the most volatile  
requirements. HTML documentation can  
also be published to allow stakeholders  
who don't have access to Enterprise  
Architect to view the models from an  
Intranet site that can just be placed on a  
file system without the need for a Web  
Server.

**Options for** There are several options that can be set  
**Documentati** to tailor the information that is included  
**on** in a generated document, including the  
ordering of elements and diagrams and  
hiding certain elements. Filters and word  
substitutions and other options can also  
be applied.

**Order**

Order Packages by: Tree Order Ascending

Order Elements by: Tree Order Ascending

Order Diagrams by: Tree Order Ascending

**Options**

- ☐ Hide 'note-less' elements
- ☐ Propagate Package Filters
- ☐ Hide 'note-less' connectors
- ☐ Hide <Anonymous> elements
- ☐ Disable large OLE file support
- ☐ Use style defined in template for notes
- ☐ Insert page breaks when generating a Master Document
- ☐ Include child elements even if the parent element is filtered out
- ☐ Indent Linked Document Headings
- ☒ Hide Diagram Borders
- ☐ Hide non-printable objects
- ☐ Show status colors
- ☐ No bookmarks
- ☐ Skip root package

Diagram Format: Metafile ☐ Set as Default

Adjust Heading Levels: Heading 9

Switch generator

**Filter**

Only include objects: Created After 15/05/2018

Where Package Phase: > 3

With element status:

Connector Direction: Both

Except where Query excludes

☒ Custom SQL ☐ Custom Script

**Learn more  
about  
Documentation**

[Model Publishing](#)

# Gap Analysis Matrix

## Getting to know the Gap Analysis Matrix

### **Introducing the Gap Analysis Matrix**

The **Gap Analysis Matrix** is a specialized **Relationship Matrix** that is used to record the gaps that exist between two versions of some part of an enterprise. The gaps between two different versions of an architecture could be recorded, or the gaps between two versions of Capabilities or Staff Competencies, or two versions of Information or Data. The tool is structured similarly to a spreadsheet with columns and rows. The elements that make up the baseline (starting point) are listed as rows and the elements that make up the target (end point) are listed as columns. There is a column for recording missing or eliminated elements and a row for recording new elements. At the intersection of a baseline element and target element, notes can be added that describe any details of the relationship between the two elements.



Gap Analysis Matrix

Target Architecture: Target1 Filter: ABB Profile: Refresh

Baseline Architecture: Baseline1 Filter: ABB Record Gap As: Options

Target \ Baseline	Video Conferencing Services	Enhanced Telephony Services	Mailing List Services	Missing / Eliminated
Broadcast Services				Retired service : Intentionally eliminated
Video Conferencing Services	Included			
Enhanced Telephony Services		Potential match		
Shared Screen Services				Address Shared Screen Service : Unintentionally eliminated
New		Improve Telephony service : To be enhanced	Mailing List : New-To be produced or developed	

## Where to find the Gap Analysis Matrix

Ribbon: Design > Impact > Gap Analysis

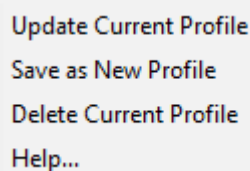
## Usage of the Gap Analysis Matrix

The **Gap Analysis Matrix** can be used for both business and technical analysis. It is a general purpose tool for recording the details of a comparison between different versions of some part of an enterprise. In business analysis it can be used to analyze Staff Competencies, Data and Information, Business

Processes, Lines of Business and more, comparing them within current and future states of an enterprise. In Enterprise Architecture the matrix can be used to record gaps between baseline, transition and target architectures, comparing Capabilities, Architecture and Solution Components and more.

### **Options for the Gap Analysis Matrix**

The **Gap Analysis Matrix** can be configured to display different parts of the repository. Once the appropriate Packages have been chosen for the Target and Baseline, and the types of element have been selected for the filter, the Gap element type can be selected. The element chosen for the gap will restrict the available elements to represent the gap for 'Missing' or 'New' elements in cells in the matrix. There are a number of choices available from the 'Options' menu, including being able to update, delete and save the Gap Analysis Matrix as a profile, giving it a name so that it can be recalled at a later time.



- Update Current Profile
- Save as New Profile
- Delete Current Profile
- Help...

**Learn more  
about the  
Gap Analysis  
Matrix**

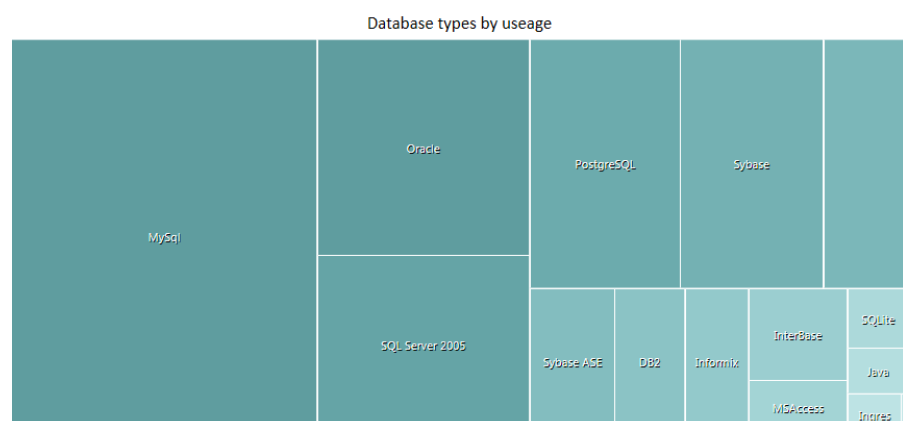
[Gap Analysis Matrix](#)

# Heat Map

## Getting to know the Heatmap

### Introducing the Heatmap

A **Heat Map** is a type of chart that can be used to visualize data in two dimensions. It uses the color of rectangles to indicate a dimension of the data and the relative size of the rectangles to indicate another dimension. They are typically used to create compelling representations of data for strategic or tactical decision making. They can be used at any level of a repository from strategic architecture down to Technology architectures.



### Where to find the

Double-click on Chart element | Chart Details | Source > Package

## Heatmap

### Usage of the Heatmap

Heatmaps are typically used to create compelling representations of data for strategic or tactical decision making. They can be used with Requirements to indicate the statuses of a group of requirements and, if the metrics are available, the estimated implementation cost of each requirement. They could be used with an application or technology inventory to show the prevalence of technologies. For example, which applications were developed in a particular language or run on a particular operating system.

### Options for the Heatmap

As an alternative to specifying the parameters of the **Heat Map** in the fields on the 'Package' tab, you can select the 'Custom SQL' tab and create a customized Heat Map using SQL. You still specify the chart type in the 'Type' field, but the other dialog fields are grayed out.

### Learn more about the Heatmap

[Heat Maps](#)



# Import and Export Spreadsheets

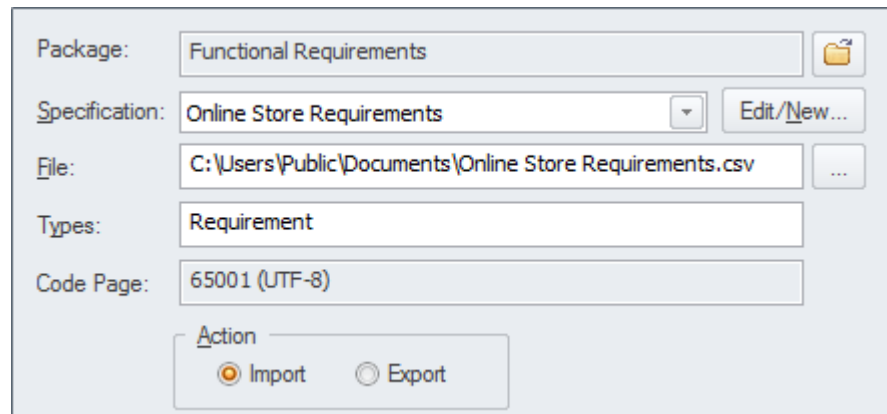
## Import and Export Spreadsheets

### **Introducing Import and Export Spreadsheets**

This facility is a useful mechanism to import Requirements that have been defined in a Spreadsheet or a Word Processor table into Enterprise Architect. Once in Enterprise Architect the Requirements can be managed and traced to elements such as business drivers and Scenarios and Components. Alternatively Requirements in Enterprise Architect can be exported to a Spreadsheet for the purposes of providing them to a third party or for some type of numerical or statistical analysis. The mapping between fields in the Spreadsheet and the analogous properties in Enterprise Architect is completely configurable using a specification.

For more detailed information exchange, the MDG Link for Microsoft Office (available from Sparx Systems) provides additional functionality and integration points useful when dealing with complex

## Requirements.



The screenshot shows a dialog box titled 'Requirements'. It contains several fields and buttons:

- Package:** A text box containing 'Functional Requirements' and a folder icon button.
- Specification:** A dropdown menu showing 'Online Store Requirements' and an 'Edit/New...' button.
- File:** A text box containing 'C:\Users\Public\Documents\Online Store Requirements.csv' and a browse button ('...').
- Types:** A text box containing 'Requirement'.
- Code Page:** A text box containing '65001 (UTF-8)'.
- Action:** A section with two radio buttons: 'Import' (selected) and 'Export'.

### **Where to find Import and Export Spreadsheets**

Ribbon: Publish > Model Exchange > CSV

### **Use of Import and Export Spreadsheets**

This feature can be used to import or export Requirements from a CSV file. Before a tool such as Enterprise Architect was installed, Analysts might have used a Spreadsheet or a table in their favorite word processor to record Requirements; these can conveniently be imported using the CSV import facility. Alternatively, Requirements sometimes have to be provided to a third party who will typically specify that they want them in a Spreadsheet file; this can be achieved using the export facility.

### **Options to Import and**

The import and export facility is completely configurable and has a



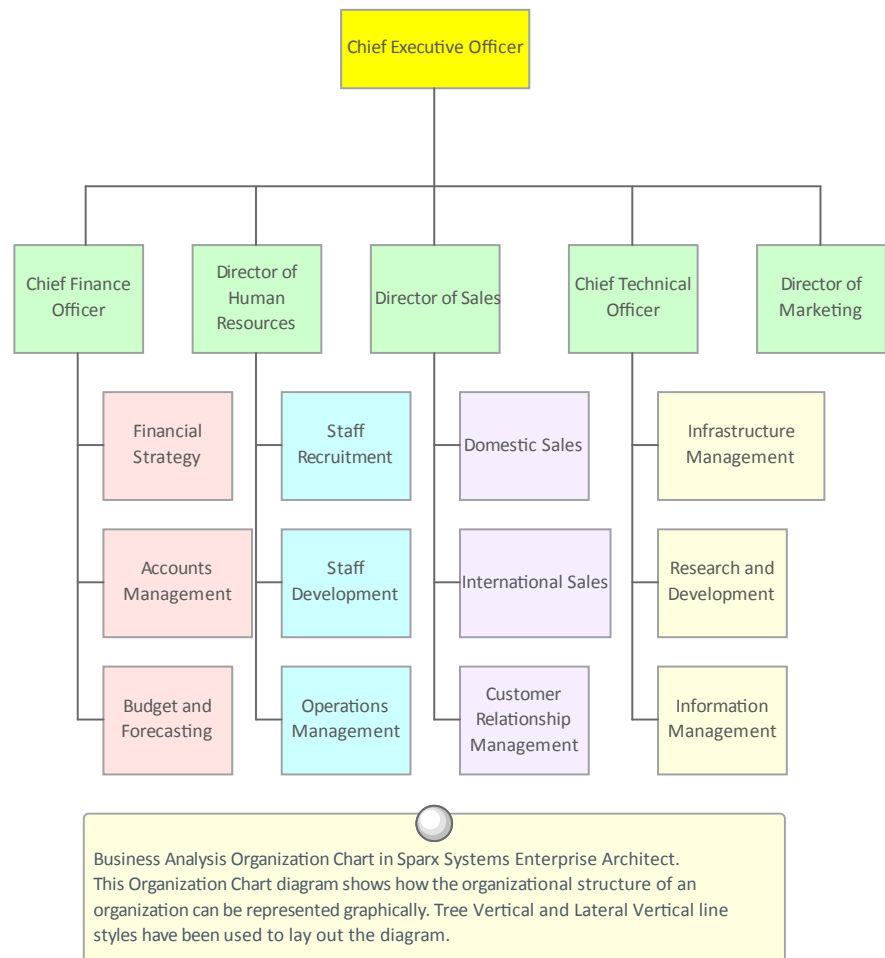
**Export Spreadsheets** user-defined specification to facilitate the mapping of Spreadsheet fields to Requirements properties in Enterprise Architect. This facility also includes the ability to import and export fields in Tagged Values of the Requirement.

**Learn more about Import and Export Spreadsheets** [CSV Import and Export](#)

# Organizational Chart Diagram

## Getting to know the Organizational Chart Diagram

**Introducing the Organizational Chart Diagram** An organizational model is a powerful tool that is a visual representation that describes the roles, responsibilities and the reporting structure of an organization. It is invaluable to a Business Analyst as it allows them to visualize and to understand who they need to communicate with while performing various tasks. The Organization Chart in Enterprise Architect is flexible and can be used to create any type of representation including the inclusion of images representing teams or individuals.



## Where to find the Organizational Chart Diagram

Ribbon: Design > Diagram > Add > Strategic Modeling > Org Chart

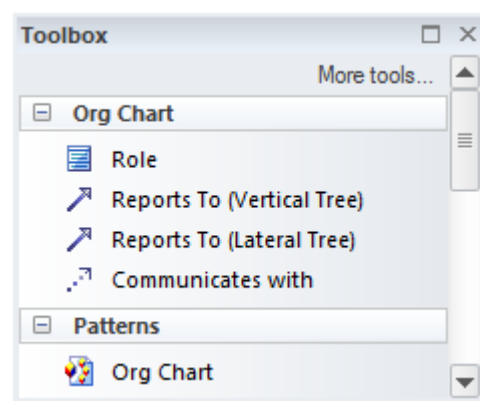
Browser window Toolbar : New Diagram icon > Strategic Modeling > Org Chart

Browser window context menu | Add Diagram... > Strategic Modeling > Org Chart

## Usage of the Organizational Chart

The Organizational Chart can be used to create any type of chart, including a Functionally Orientated, Market

- Diagram** Orientated or Matrix Model chart. A number of charts could be created, representing different current and future states of the enterprise.
- Options for the Organizational Chart Diagram** Apart from the relationships that represent reporting lines in the Organizational Chart, Business Analysts can also add relationships of authority, influence, and communication, which might not overlay the reporting lines. These additional lines can be modeled using named Associations and stereotyped if required.



The Organizational Chart (like any diagram) can be viewed as an element list, which makes working with the element's properties easier.

Diagram Filters can also be used when presenting the diagrams, to draw

attention to parts of the diagrams; the diagrams can be presented as hand drawn or in a whiteboard style by changing the properties of the diagram.

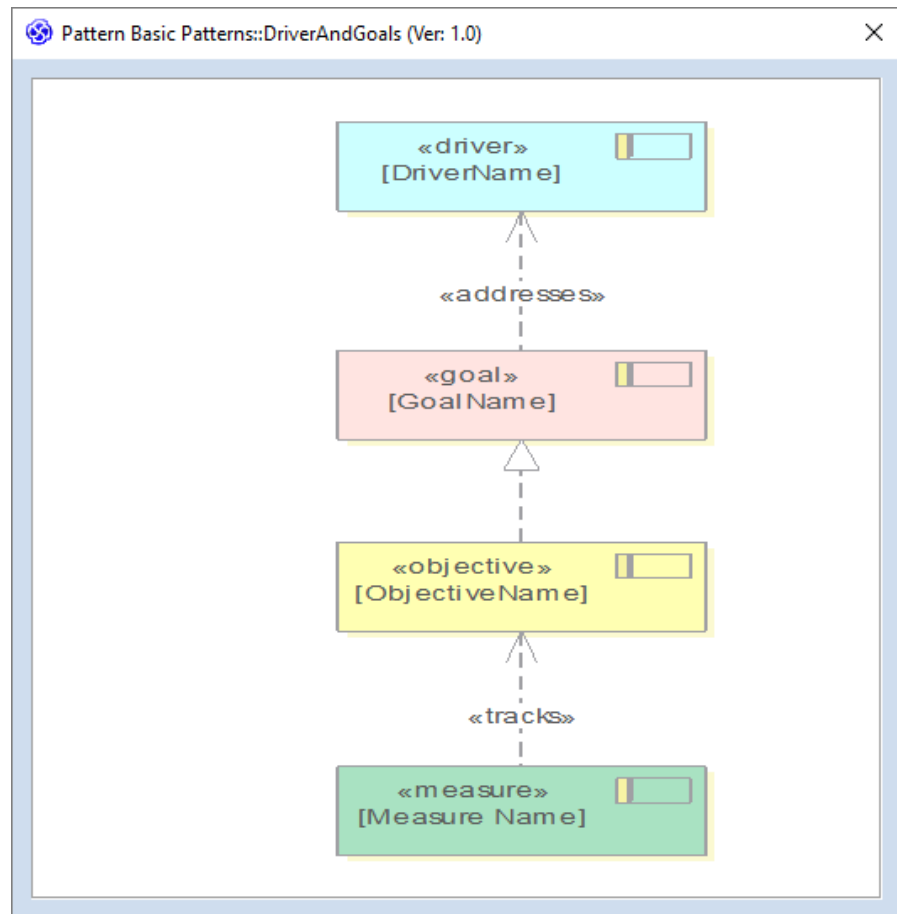
**Learn more  
about the  
Organizational Chart  
Diagram** [Organizational Chart](#)

# Patterns

## Getting to know Patterns

### **Introducing Patterns**

A Pattern is a general reusable design solution to a commonly occurring problem within a given architectural context. Patterns are not resolved designs, but rather templates for how a problem can be solved. The concept originated in the building architecture world and was first published in a book by Christopher Alexander entitled *Design Patterns*. They were then applied to the software industry and were used extensively by the software engineering domain to solve commonly recurring software engineering problems, even though on the surface the nature of the problems seemed quite different.

**Where to find Patterns**

Create a Pattern:

Specialize > Technologies > Publish-Tech > Publish Diagram as Pattern

Use a Pattern:

Start > Desktop > Share > Resources > Patterns <pattern group> > Right-click on Pattern name > Add Pattern to Diagram

**Usage of Patterns**


Patterns can be applied in a wide range of situations from business to technology

architecture, but are always used to apply a common solution to any number of problems or contexts that on the surface might appear quite different. Enterprise Architect has provided a useful mechanism for mining Patterns, which means that any diagram can be published as a Pattern and then reused in the same or a different context. An example of a Pattern and its usage might be a Pattern articulating the relationship between Drivers, Goals, Objectives and Measures. An existing diagram could be published as a Pattern and then any business architecture could reuse the Pattern by simply dragging it onto an empty diagram.

### **Options for Patterns**

Patterns are most commonly available from the 'Resources' tab of the **Browser window** but are also sometimes built into technologies and made available from a Toolbox page. There are a number of options available when publishing a Pattern, including the ability to describe the details of the Pattern overall and to include notes for each of the elements that make up the Pattern.



 Save Diagram as UML Pattern ✕

Pattern Name:

Filename:  ...

Category:  Version:

Notes:  
Intent  
Allow a business architect to define how drivers are resolved o goals which in tum are broken down into Objectives and which in tum are validated by Measures.

Name	Type	Create	Merge	Instance	Type	Default	Comment
[Measure Name]	Requirement	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[Measure Na...	A element for che...
[ObjectiveName]	Requirement	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[ObjectiveNa...	An element that s...
[GoalName]	Requirement	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[GoalName]	An element that s...
[DriverName]	Requirement	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[DriverName]	An element that s...

# Relationship Matrix

## Getting to Know the Relationship Matrix

### **Introducing the Relationship Matrix**

The **Relationship Matrix** provides a visualizing compelling matrix-style view for a convenient analysis of the way that Requirements are related to each other and to other elements in the model. It can be used to view the relationships between Stakeholders and their Requirements, how Use Cases are related to Business Requirements or Functional Requirements, how Capabilities are related to Business Drivers, which Components implement a set of Requirements, and more. Any number of matrices can be defined quickly and then saved to be viewed in workshops, or included in documentation generated automatically from the model or exported to a spreadsheet file. When a matrix is created, connections can be viewed by placing the Requirements on one axis of the matrix and the connected elements on the other axis, then the cells of the matrix will indicate the direction

of the relationship.

Target +	REQ011 - Manage User Accounts	REQ012 - Provide Online Sales	REQ013 - Manage Deliveries	REQ014 - ShoppingBasket	REQ015 - Process Credit Card Payment	REQ016 - Add Users	REQ017 - Remove User	REQ018 - Report on User Account	REQ019 - Manage Inventory	REQ020 - Receive Books	REQ021 - List Stock Levels	REQ022 - Order Books
+ Source												
Add New Titles												
Add To Shopping Basket				↑								
Close Account							↑					
Create Account						↑						
Create Orders												↑
Delete User							↑					

### Where to find the Relationship Matrix

In the **Browser window**, click on Package and select:

- The 'Resources' tab | Matrix Profiles | Right-click on a profile | Open Matrix Profile or
- The Start ribbon > Desktop > Design > Details > Matrix

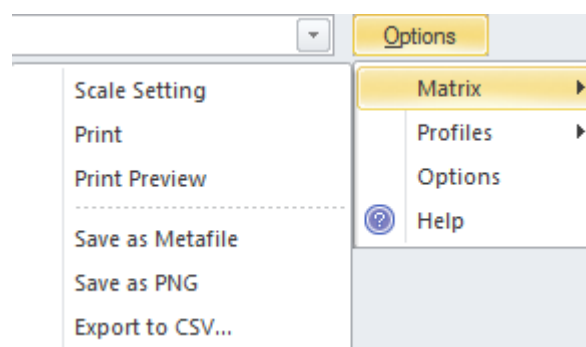
### Usage of the Relationship Matrix

To display the relationships that exist between elements - such as which Requirements are realized by which Use Cases - in two Packages in a visually compelling matrix. It is useful in analyzing missing elements or relationships; for example, to determine which Requirements are not realized by

any Use Case, or which Components do not have corresponding Requirements or Use Cases. It is particularly useful in workshops with Business Stakeholders who might not be familiar with seeing Requirements in Trace diagrams.

### Options for the Relationship Matrix

There is a range of options that can be set for the **Relationship Matrix**, including saving it to the 'Resources' tab of the **Browser window** or to a CSV format for opening in a spreadsheet. The appearance of the Relationship Matrix can also be altered by sorting the elements, showing an outline numbering view, and suppressing Package names. These items are available from the **Options button** on the Relationship Matrix.



### Learn more about the Relationship Matrix

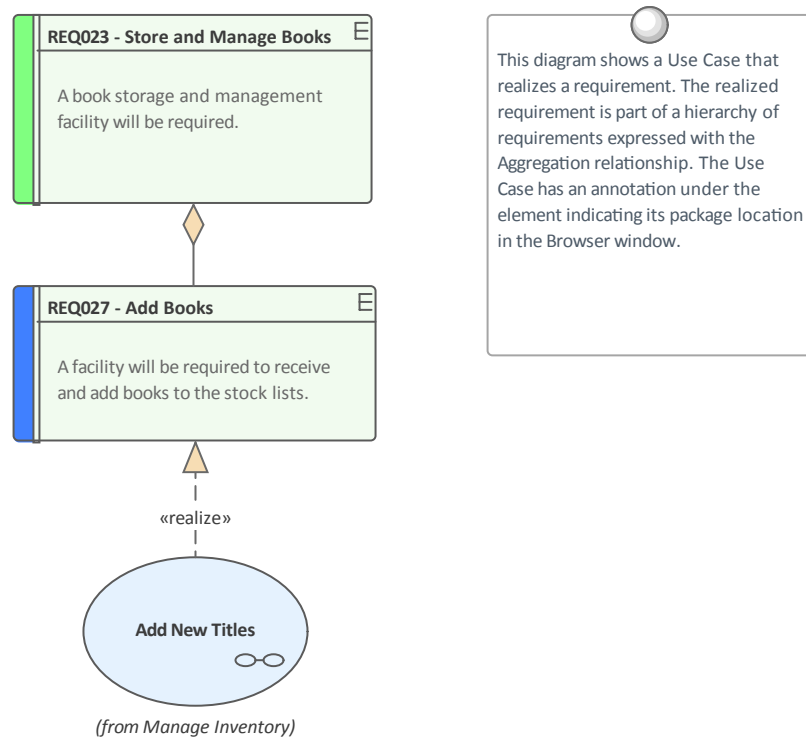
[Relationship Matrix](#)



# Requirements Diagram

## Getting to Know the Requirements Diagram

**Introducing the Requirements Diagram** The Requirements diagram provides a visual representation of how Requirements are related to each other and to other elements in the model, including Business Drivers, Constraints, Business Rules, Use Cases, User Stories, design Components and more. The diagram is one of Enterprise Architect's extended diagram types. It provides an appealing graphical representation of Requirements, that will be a welcome change for Requirements Analysts who are accustomed to working with text based tools.



## Where to find the Requirements Diagram

**Browser window Context Menu :** Add Diagram : | Manage | Show All Perspectives | Extended | Requirements

## Usage of the Requirements Diagram

One usage is to show how Requirements are connected together in a hierarchy or, even more importantly, how Requirements are connected to other elements. The experienced modeler will define and manage the Requirements in the **Specification Manager** and then use the Requirements diagram to show how each Requirement is related to upstream process elements such as Business Drivers, and downstream process elements such as Use Cases, User

Stories, User Experience designs and solution Components.

**Options for the Requirements Diagram** The appearance of a diagram can be changed to suit the audience, and details can be included, suppressed or altered to ensure the diagram meets its main objective of communication. There is a wide range of options, ranging from creating a Hand Drawn style of diagram to filtering diagram content.



**Properties** □ ×

Diagram Compartment

---

**General**

Name	Requirements Model
Type	Requirements
Stereotype	
Author	hbritten
Applied Metamodel	Default
Filter to Metamodel	<input type="checkbox"/>
Filter to Context	<input type="checkbox"/>
Context Navigation	<input type="checkbox"/>

**Version**

Version	1.0
Filter to Version	<input type="checkbox"/>
New to Version	<input type="checkbox"/>

**Appearance**

Display as	Diagram
Hand Drawn	<input checked="" type="checkbox"/>
Whiteboard	<input type="checkbox"/>
Custom Style	<input type="checkbox"/>
Disable fully scoped object names	<input type="checkbox"/>
Display Element Lock Status	<input type="checkbox"/>
Use Info Tip (global)	<input type="checkbox"/>
Theme	Use global theme

**Advanced**

MDG Technology	Extended::Requirements
GUID	{82928D10-B2FA-4314-A1ED-2...
WebEA	

**Connectors**

Show Relationships	<input checked="" type="checkbox"/>
Show Non-Navigable Ends	<input type="checkbox"/>
Show Property String	<input checked="" type="checkbox"/>
Suppress All Labels	<input type="checkbox"/>
Show Stereotype Labels	<input checked="" type="checkbox"/>
Show Feature Linker	<input checked="" type="checkbox"/>
Connector Notation	UML 2.1

**Learn more**

**about the**  
**Requirement**  
**s Diagram**

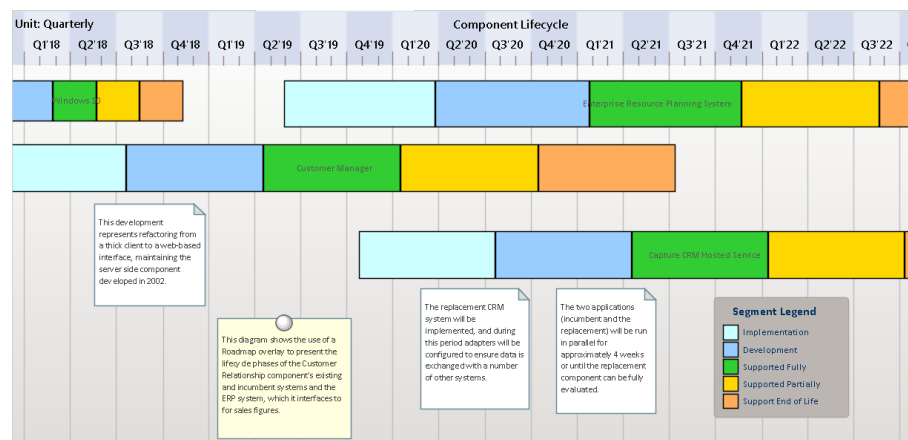
[Working In Diagrams](#)

# Roadmap Diagram

## Getting to know the Roadmap Diagram

### Introducing the Roadmap Diagram

The Roadmap diagram is an overlay that can be applied to any diagram to describe significant phases in elements and how they change with the passage of time.



There is no restriction on the type of element that can appear on the diagram, and any diagram can have a Roadmap overlay defined. Significant user defined phases in the element's lifetime are represented by colored bars, which can be set to show **duration**. The colors and the phases can be configured using a Diagram Legend, which automatically applies them to the elements in the diagram. They are particularly useful in

Enterprise Architecture diagrams for describing capability and application Roadmaps.

**Where to  
find the  
Roadmap  
Diagram**

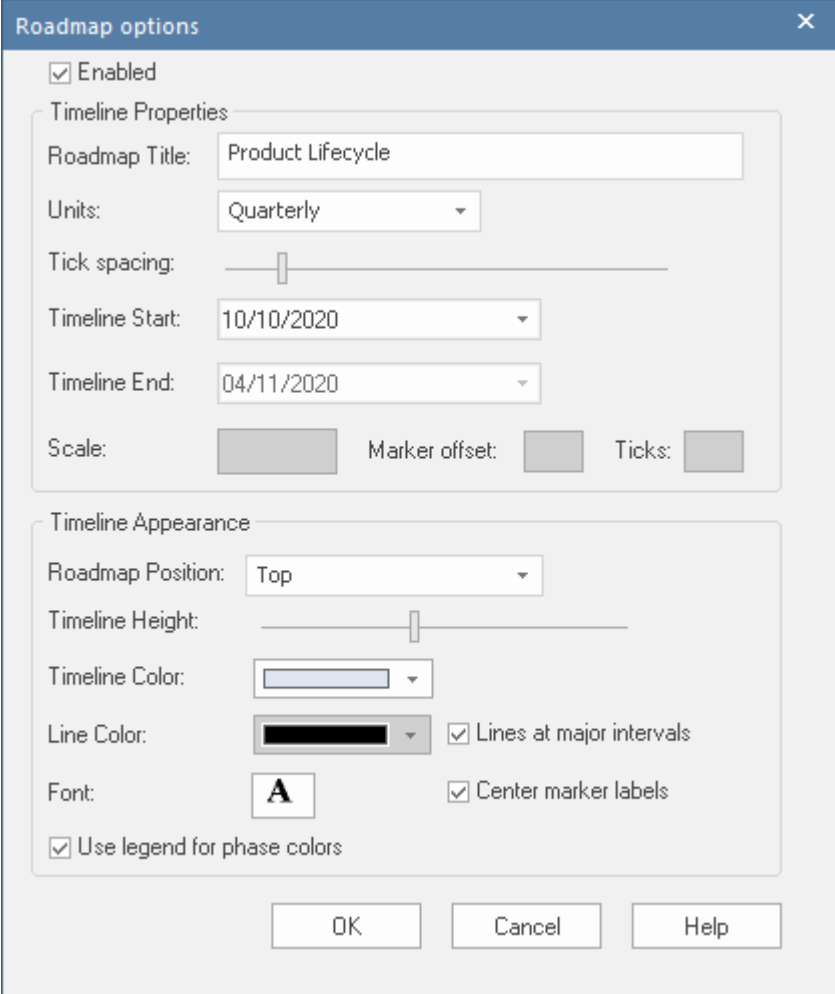
Ribbon: Layout > Diagram > Roadmap  
Diagram Context Menu: Roadmap

**Usage of the  
Roadmap  
Diagram**

The Roadmap diagram has a wide range of uses in Enterprise Architecture where they can be used to show application and capability roadmaps to Systems Engineering, where they are used to show timing in low level components.

**Options for  
the Roadmap  
Diagram**

The Roadmap overlay has a range of options that determine the properties of the timeline, such as the scale of the time rulers, units, their positions, and the appearance of the time line including fonts and colors. The height and position of the timeline can also be configured to suit the diagram and display.

The image shows a 'Roadmap options' dialog box with a blue title bar and a close button. It contains two main sections: 'Timeline Properties' and 'Timeline Appearance'. In the 'Timeline Properties' section, there is a checked 'Enabled' checkbox, a 'Roadmap Title' text field with 'Product Lifecycle', a 'Units' dropdown menu set to 'Quarterly', a 'Tick spacing' slider, 'Timeline Start' and 'Timeline End' date dropdowns set to '10/10/2020' and '04/11/2020' respectively, and three color swatches for 'Scale', 'Marker offset', and 'Ticks'. The 'Timeline Appearance' section includes a 'Roadmap Position' dropdown set to 'Top', a 'Timeline Height' slider, a 'Timeline Color' dropdown, a 'Line Color' dropdown set to black, and three checked checkboxes: 'Lines at major intervals', 'Center marker labels', and 'Use legend for phase colors'. A font preview shows the letter 'A'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

The Diagram Legend can be configured to define the phases in the element's lifetime, to set the specification of the colored bands and more. Roadmap segments can be shown or hidden on individual elements in cases where a particular segment might not apply to one or more of the elements on the diagram.

**Learn more about the** [Roadmap Diagram](#)

# **Roadmap Diagram**

# Specification Manager

## Getting to Know the Specification Manager

### **Introducing the Specification Manager**

The **Specification Manager** is the central tool for working with Requirements; it provides an interface resembling a Word Processor or Spreadsheet tool for entering, maintaining and viewing Requirements. New Requirements can be created with names and detailed descriptions and properties such as Status and Priority can be added from drop-down lists. Existing Requirements can be viewed and managed in a convenient view, and changing them in the Specification Manager will change them in all other places in the repository such as diagrams and windows. It is the perfect tool for those analysts more comfortable working with text rather than diagrams and who are accustomed to working in a Word Processor or Spreadsheet. It has the added advantage that the requirements are part of a model and can be traced to other elements, including Business

## Drivers, Stakeholders and Solution Components.

Item

### 1 REQ019 - Manage Inventory

The system **MUST** include a complete inventory management facility to store and track stock of books for the on-line bookstore.

#### 1.1 REQ122 - Inventory Reports

Inventory reports are required that detail the available stock for each item including back orders. Future stock level reports should be able to predict the quantity of stock at a specified future date.

#### 1.2 REQ023 - Store and Manage Books

A book storage and management facility will be required.

##### 1.2.1 REQ022 - Order Books

A book order facility will be required to allow on-line ordering from major stockist's.

##### 1.2.2 REQ021 - List Stock Levels

A facility will exist to list current stock levels and to manually update stock quantities if physical checking reveals inconsistencies.

**Where to  
find the  
Specification  
Manager**

**Browser window | Right-click on  
Package | Open Package in |  
Specification Manager**

**Usage of the  
Specification  
Manager**

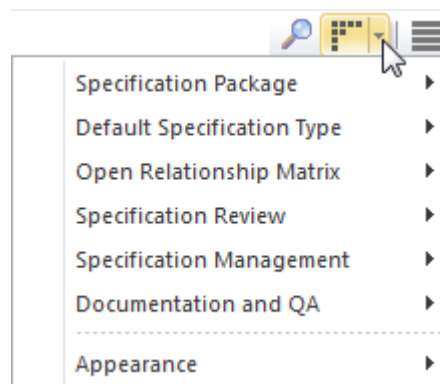
To create, view and maintain Requirements in a text based tool that resembles working in a word processor or spreadsheet. Details can be added to the Requirements and Requirement properties can be added from drop-down lists. When the Requirements are changed in the **Specification Manager**



the changes are conveniently reflected in the **Browser window** and all other windows.

### **Options for the Specification Manager**

There are a wide range of options available from the options menu, to tailor the way you use the **Specification Manager**. These include Level (hierarchical) Numbering, Auto Naming, Spell Check, Documentation, Import and Export of Requirements, access to various related tools and more.

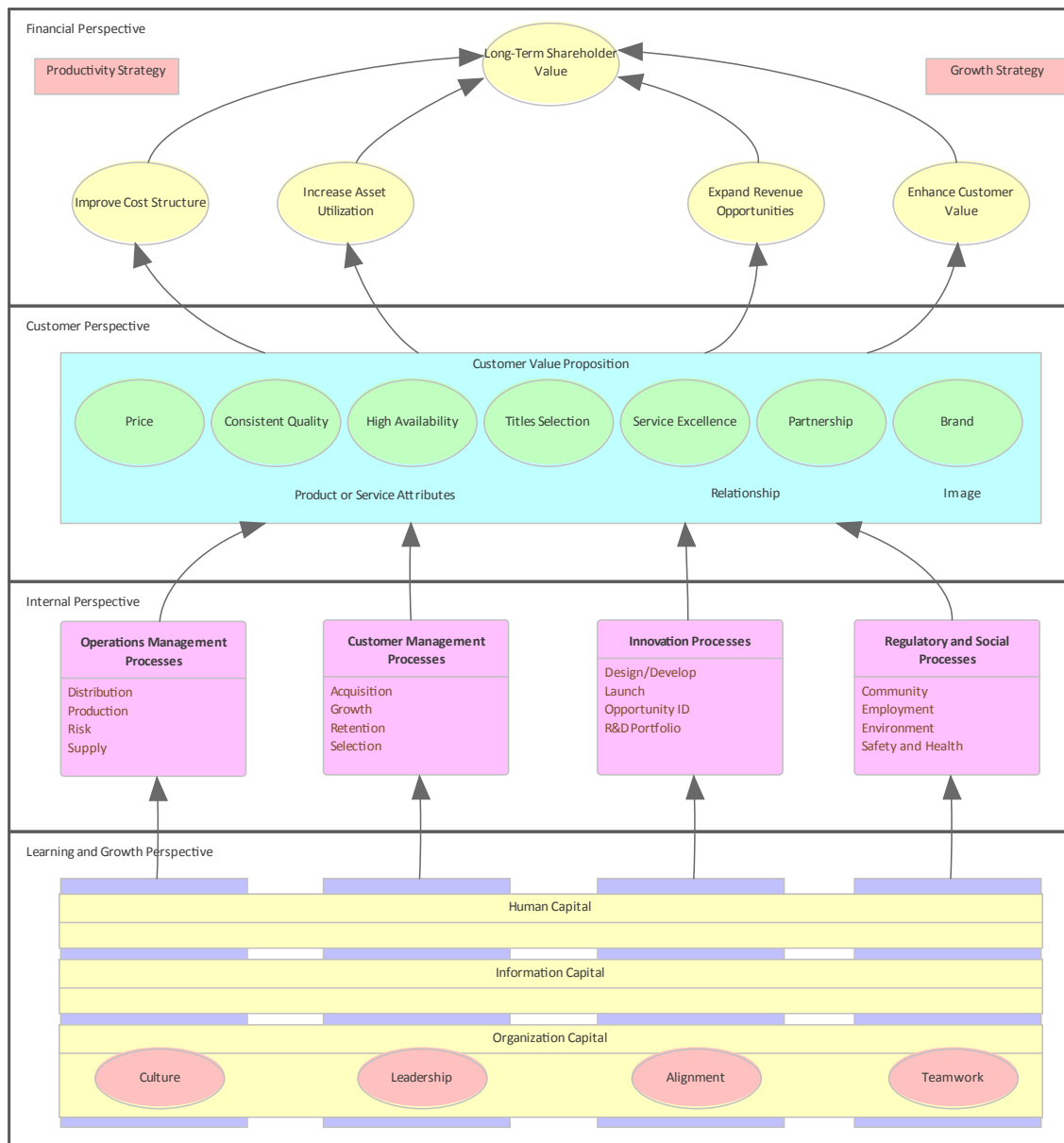


### **Learn more about the Specification Manager**

[The Specification Manager](#)

# Strategy Map

A Strategy Map is a diagram that is used to describe the primary strategic goals that are important to an organization or business team. The diagram shows four important perspectives that are the significant questions that provide the definition of a strategy. The defined perspectives are: 'Financial', 'Customer', 'Internal Business Processes' and 'Learning and Growth'. The diagram is used as a communication device to ensure there is a common understanding of the strategy, to focus organization effort and to assist with the assessment of progress.



## Getting to know the Strategy Map

**Where to find the Strategy Map**

Ribbon: Design > Diagram > Add > Strategic Modeling > Strategy Map

**Browser window Toolbar :** New Diagram icon > Strategic Modeling > Strategy Map

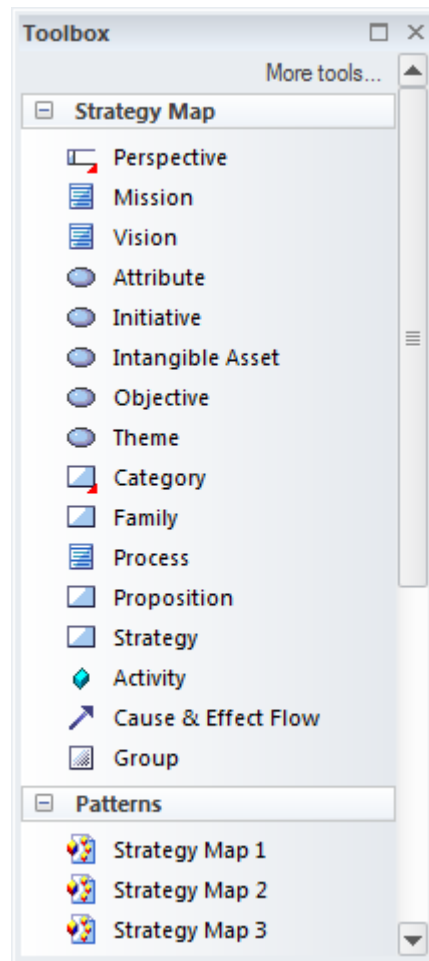
Browser window context menu | Add  
Diagram... > Strategic Modeling >  
Strategy Map

**Usage of the  
Strategy  
Map**

The Strategy Map is used to model the key strategic goals that an organization or management team intend to achieve. Elements in each of the four perspectives can be linked to other elements in the repository to show how they could be implemented at a business, application or technology level.

**Options for  
the Strategy  
Map**

A Strategy Map can be created using Patterns that automatically create elements and a diagram that can be used as a starting point for the Strategy Map. There are three Patterns available, ranging from a very simple expression with a single element in each perspective to a completely worked expression with multiple elements in each perspective. A toolbox provides a range of additional elements and relationships to extend the base maps created using the Patterns.



**Learn more  
about the  
Strategy  
Map**

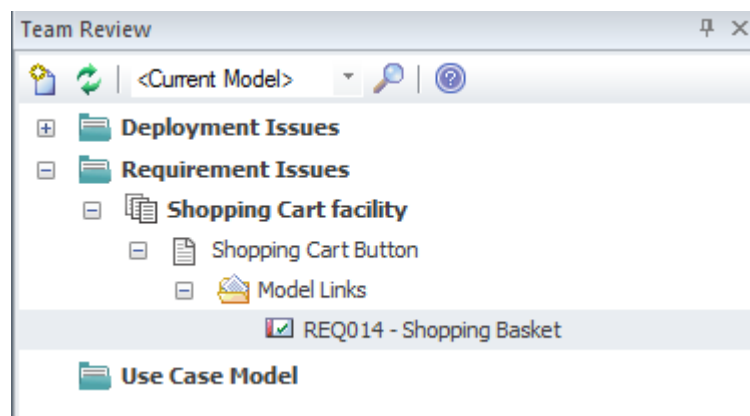
[Strategy Map](#)

# Library

## Getting to know The Library

### Introducing The Library

The **Library** window provides an opportunity for developers, modelers, customers and stakeholders to comment and provide feedback on the work in progress or at the completion of a milestone or project.



### Usage of The Library

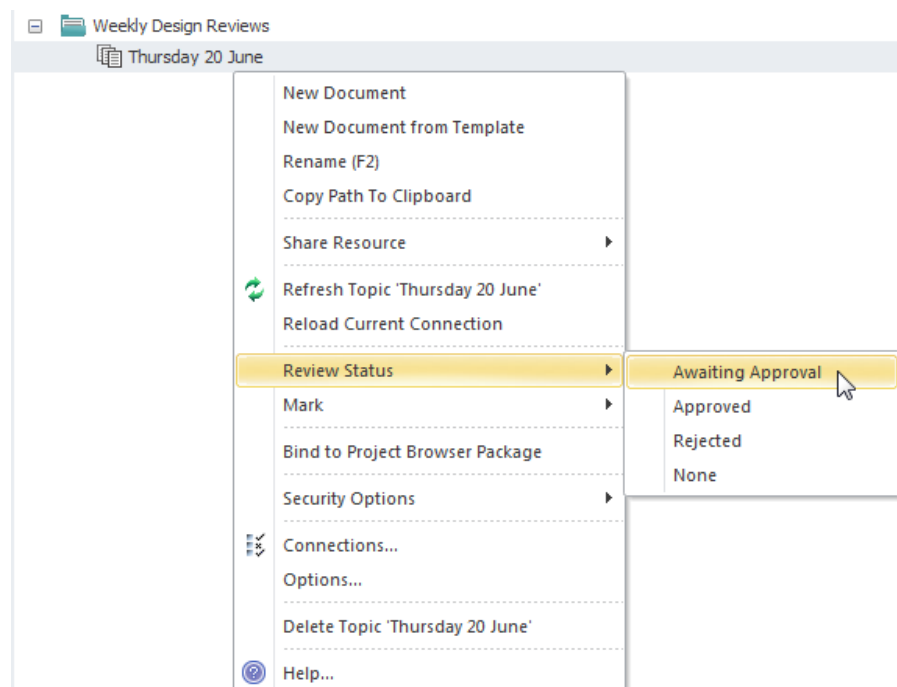
The Library feature can be used to conduct model reviews from any number of perspectives, including walk-throughs, formal model reviews, or ad-hoc reviews.

### Where to find The Library

To post or view an element's discussion  
Ribbon: Start > Desktop > Share > Library

## Options for The Library

There is a wide range of settings available to configure the Library available from the Category and Topic context menus, including setting the status of the category or topic and other options. Diagrams, elements and element features can be conveniently dragged from the **Browser window** to create model links that can be used by team members to hyperlink directly from the **Library window** to these items in the Browser window.



**Learn more  
about Team  
Library**

[Library Window](#)





# Time Aware Modeling

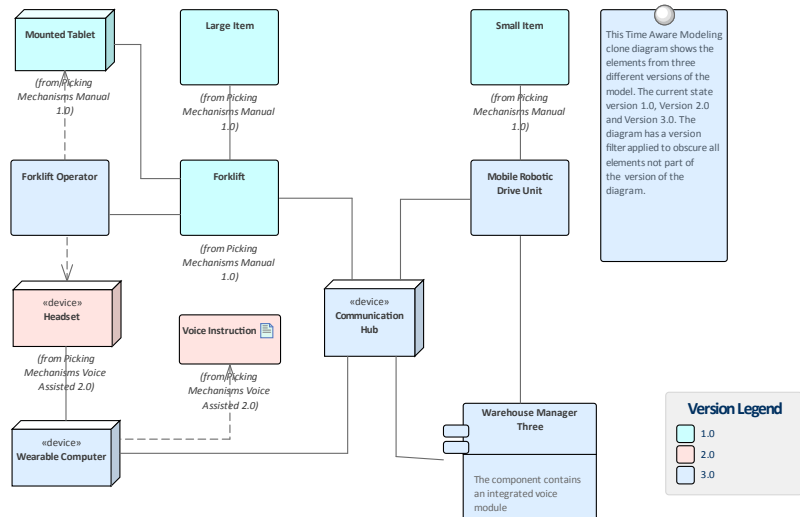
## Getting to know Time Aware Modeling

### **Introducing Time Aware Modeling**

The Time Aware Modeling feature allows analysts, architects and others to create incremental versions of their models by providing tools that facilitate the migration of elements, diagrams and Packages through the dimension of time allowing multiple transitions or versions to be created. The baseline ('As-is', current state) models remain unaffected and any number of target ('To-Be', future state) models can be constructed for the purpose of visualization and analysis. It is particularly useful and powerful when a number of future options need to be represented and compared effectively allowing 'what-if' analysis to be conducted. Time is one of the most important dimensions in architecture as it is the substrate upon which all changes occur. Architects are aware of its importance and have traditionally created models with duplicated elements; time aware modeling allows the existing

elements, diagrams and Packages to be cloned.

Integrated Mobile Robotic Drive Units and Voice Assisted Forklifts Version 3.0 Q4 2019



**Where to find Time Aware Modeling**

Clone Package Structure as New Version

Ribbon: Design > Model > Manage >

Clone Structure as New Version

**Browser window** Context Menu: Clone Structure as New Version

Clone Diagram as New Version

Ribbon: Design > Diagram > Manage >

Clone as New Version

Clone Element as New Version

Ribbon: Design > Element > Manage >

Clone Element as New Version

Diagram Context Menu: Clone Element

## as New Version

### **Usage of Time Aware Modeling**

Time Aware Modeling can be applied in a wide range of situations from business to technology models. It provides a powerful way of showing how the entities represented by the model change over time. The baseline ('As-Is', current state) models can be left unaffected while any number of target ('To-Be', future state) models can be constructed to reflect the possible evolution of the baseline models over time. There is a wide range of situations where an analyst or architect will find this tool useful; for example:

- Architects use models to document the current state and then move on to the more challenging and rewarding task of defining what the future state will look like, often in a series of transitions
- When new customers are acquired, a Business Analyst is often challenged with describing how a base product should be configured for different customer groups, resulting in different versions of the same product
- Business Strategists typically prescribe

what a Capability model will look like for the organization of the future, resulting in at least two versions of the Capability model

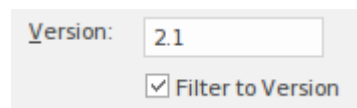
- During Mergers and Acquisitions, Business Architects are tasked with describing the possible states of the enterprise after the takeover is complete
- Engineers are required to develop better or more efficient solutions to meet the challenges of the future, so define newer and better versions of the solution
- Testers need to be aware of different versions of a product when designing and running Test Cases
- Infrastructure engineers need to define future environments in response to performance or security concerns creating multiple versions of servers, devices and even whole facilities

All of these situations require that time is incorporated into the models so that it can be reasoned about and made explicit. Enterprise Architect's Time Aware modeling facilities can be used in all these situations to ensure that time is included as a first class citizen in the

models. Time is not measured or modeled in absolute or relative terms but by representing any number of future states or differences in the form of versions.

### **Options for Time Aware Modeling**

The Time Aware Modeling features allow a modeler to clone Packages, diagrams and elements. Most models are not trivial and Enterprise Architect provides a wide range of tools that will assist in the visualization of the models and how they change over time. The **Traceability window** will be particularly useful for viewing the connection between elements in the time aware models and other parts of the repository. A very useful feature is the ability to apply a filter to a diagram based on version, thus obscuring elements that are not part of a particular version.



### **Learn more about Time Aware Modeling**

[Time Aware Modeling](#)

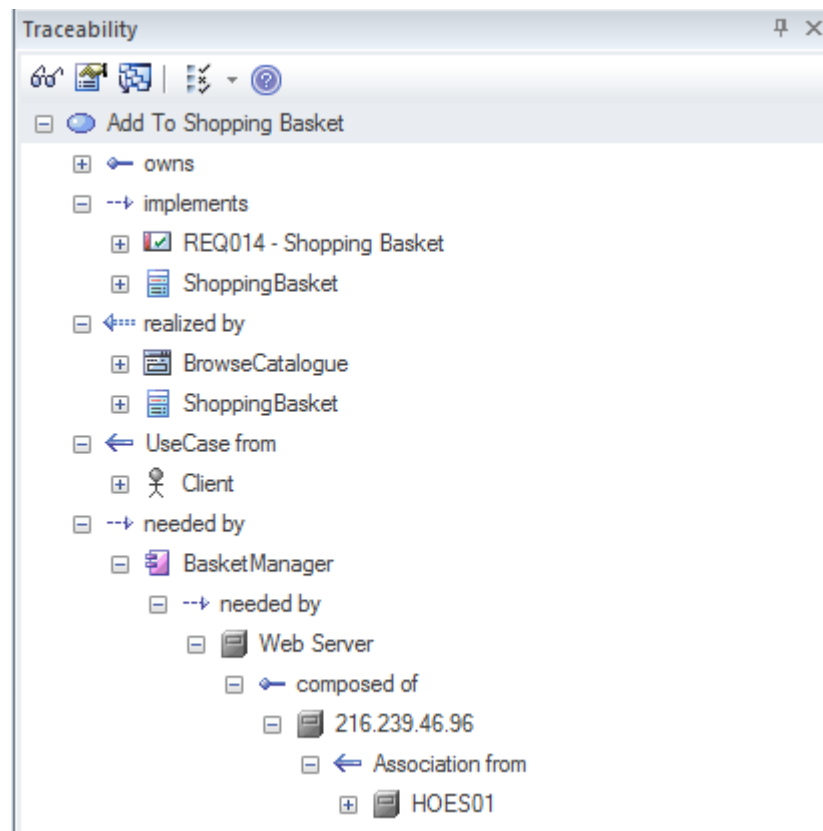


# Traceability Window

## Getting to Know the Traceability Window

### **Introducing the Traceability Window**

The **Traceability window** provides a hierarchical view of element connections, allowing traceability to be visualized and queried as elements are traversed in the model. This tool is particularly powerful because a modeler will often choose to hide diagram relationships, but by selecting an element in the diagram and viewing its connections in the Traceability window all its relationships will be revealed.



### Where to find the Traceability Window

Start > Desktop > Design > Traceability

### Usage of the Traceability Window

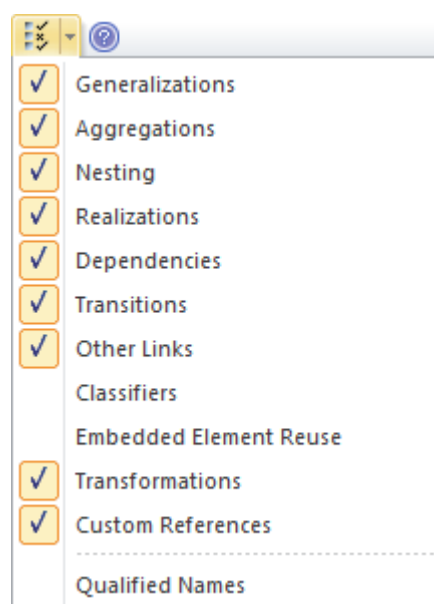
The **Traceability window** provides a hierarchical view of the way an element is connected to other elements in the repository, along with the type of each relationship. This window gives a complete list of all relationships that cannot be seen by viewing elements in the **Browser window** and that also might not appear in any diagrams. It is very useful for managing Requirements and



tracing how a Requirement is related to upstream process elements such as Business Drivers and downstream process elements such as Components. It is a useful tool, enabling newcomers to a model to gain a quick understanding of which are the important and well connected elements. Before you delete an element in the model, you should use the Traceability window to ensure that you understand that element's existing relationships.

### **Options for the Traceability Window**

There is a series of options that restrict traceability to specified connector types; these options can be set to alter what is displayed in the window. The options are available from the toolbar at the top of the window.



**Learn more  
about the  
Traceability  
Window**

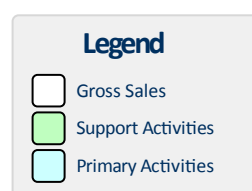
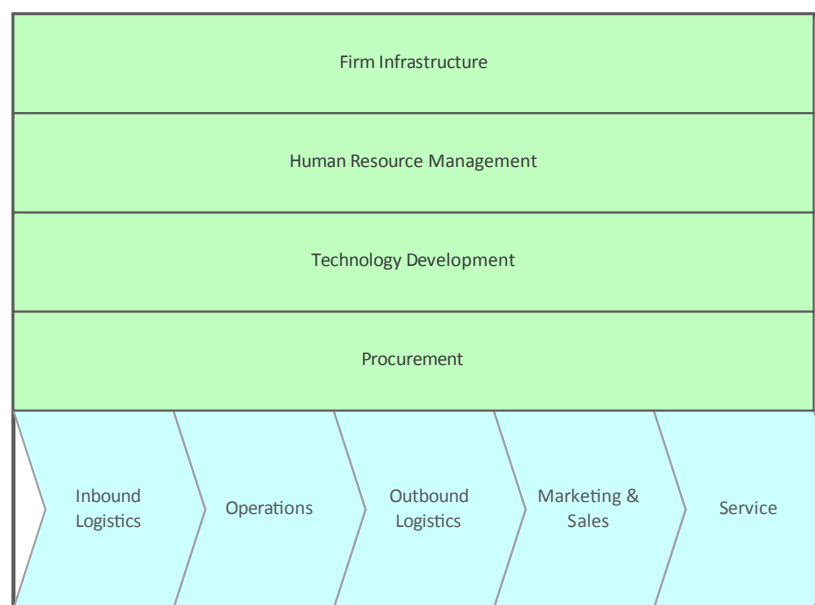
[The Traceability Window](#)

# Value Chain

## Getting to know the Value Chain

### Introducing the Value Chain

The Value Chain is a strategic diagram that allows the primary and secondary activities in an organization to be modeled. The diagram can be created from a Pattern that adds and connects the five primary activities in a chain and the four supporting activities underpinning them.



The Gross Sales element acts as a frame for the Primary Activity and Support Activity elements.

An Analyst working at the strategic business unit level will often be asked to model the activities the business unit performs to provide value to its customers. The Value Chain is the preferred tool for creating this strategic representation of the sequence of activities that an organization performs.

**Where to find the Value Chain**

Ribbon: Design > Diagram > Add > Strategic Modeling > Value Chain

**Browser window** Toolbar : New Diagram icon > Strategic Modeling > Value Chain

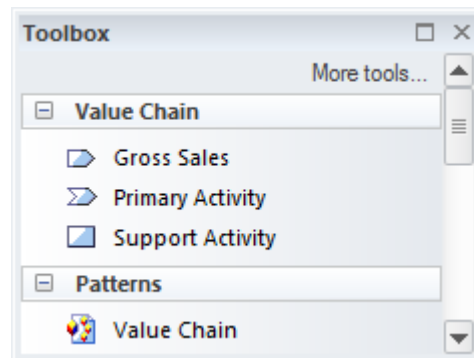
Browser window context menu | Add Diagram... > Strategic Modeling > Value Chain

**Usage of the Value Chain**

The Value Chain is an important tool to assist with strategic planning allowing the whole sequence (or chain) to be understood. It also allows the chain to be broken down into its constituent activities allowing the evaluation of costs, resource and value to be determined and potentially improved.

## Options for the Value Chain

Each one of the Primary and Supporting Activities can be linked to other elements in the model including a Linked Document and elements that define benchmarks.



The Value Chain diagram (like any diagram) can be viewed as an element list which makes working with the element's properties easier.

Diagram Filters can also be used when presenting the diagrams to draw attention to parts of the diagrams.

## Learn more about the Value Chain

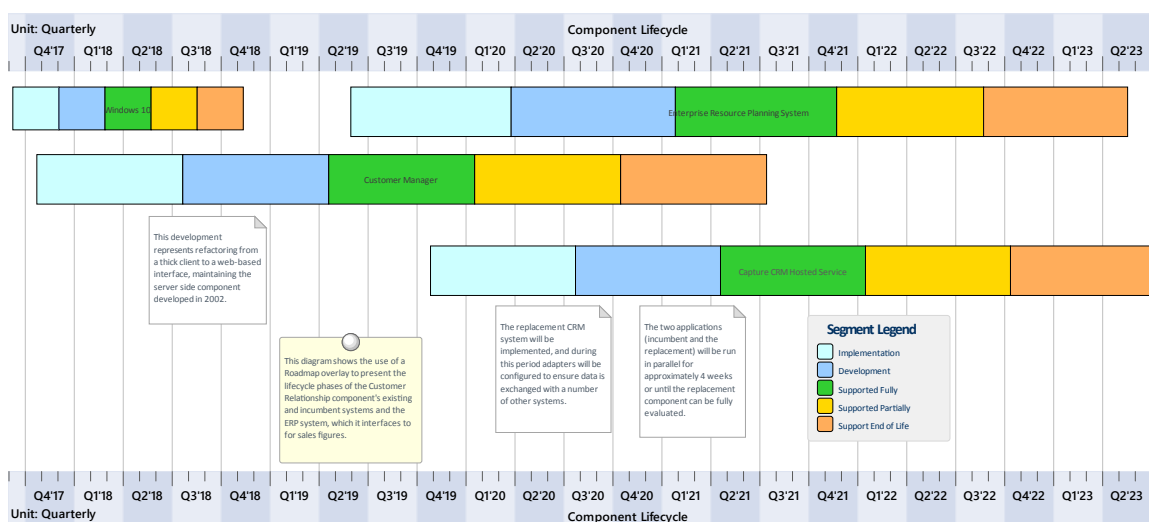
[Value Chain](#)

# Uses of an Enterprise Architecture

An Enterprise Architecture can be used in a number of contexts but all of these can be summarized in three categories:

- Prescriptive - *Articulates the designs to transform an enterprise.*
- Restrictive - *Acts to govern, guide or constrain the implementations.*
- Descriptive - *Describes the enterprise so as to facilitate decision making.*

In practice an architecture can be used in the context of Mergers and Acquisitions, Corporate Divestiture and Business and Systems Improvements. It can also be used as a powerful communication mechanism to inform stakeholders of aspects of the enterprise and how they are related. Architectures can assist with Portfolio Management and act as a guide for implementation initiatives and can be used for a range of other uses.





# Mergers and Acquisitions

Mergers and Acquisitions (takeovers) have become an important part of the business strategy for many medium to large enterprises in an effort to adapt in a business and technology environment dominated by digital disruption, globalization, regulatory changes, technology breakthroughs, and global and local economic forces. Enterprise Architecture has a key role to play in describing and planning the way the takeovers are organized and performed, ensuring they have the best chance of achieving the desired business outcomes.

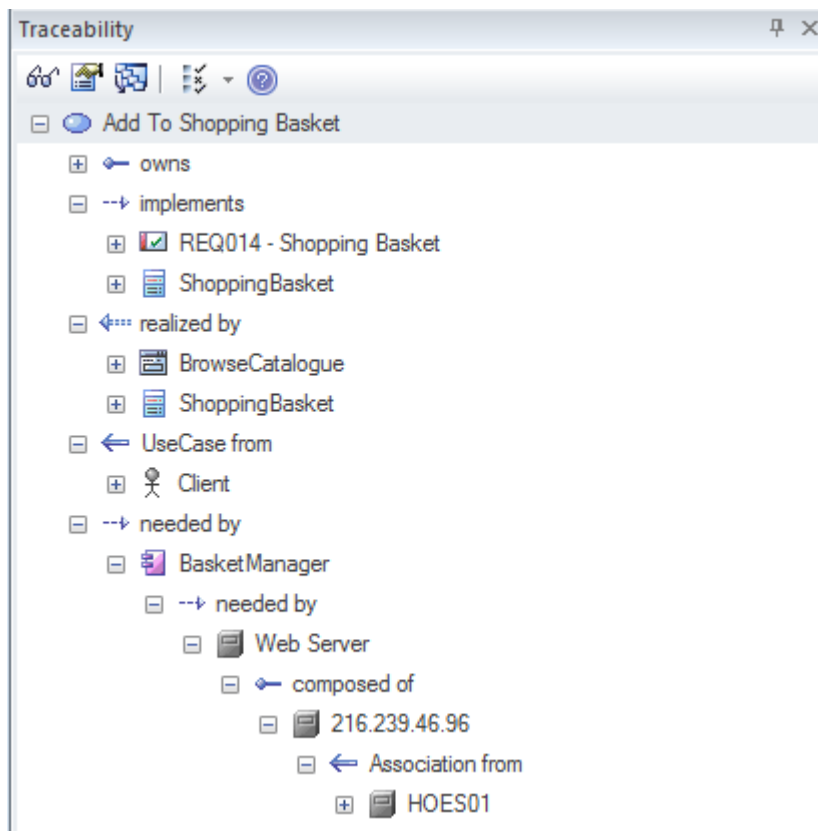
Enterprise Architect has a wide range of tools that can assist in the planning and execution of architectures designed to transition the enterprise during a Merger or Acquisition. These include tools such as the **Specification Manager** to view catalogs of Capabilities, Objectives, Applications and more, and the Roadmap diagrams that can be used to plan and visualize the sequence of changes to Capabilities, Applications, Technology Nodes and other elements.



# Corporate Divestiture

Corporate divestiture is an important business strategy to reduce operating losses and achieve higher efficiency, by shedding some of the enterprise's assets in its business portfolio. The need for the divestiture can be varied, including overlapping functions resulting from mergers or acquisitions, business units identified as under-performing, or regulatory requirements or market pressures forcing the enterprise to change. When the need for the divestiture has been identified, the selected business units can be removed and revenues from the sale, spin off or carve out can be channeled to develop other business functions.

Enterprise Architect has a number of tools that can assist with the planning and management of the divestiture. The removal of a business function will result in the concomitant need to identify the technical assets that need to be removed or decommissioned or untangled from other parts of the system. Enterprise Architect has powerful traceability and visualization facilities that can be used to understand the impact on the Business, Information, Application and Technology architectures.



# Architecture Oversight

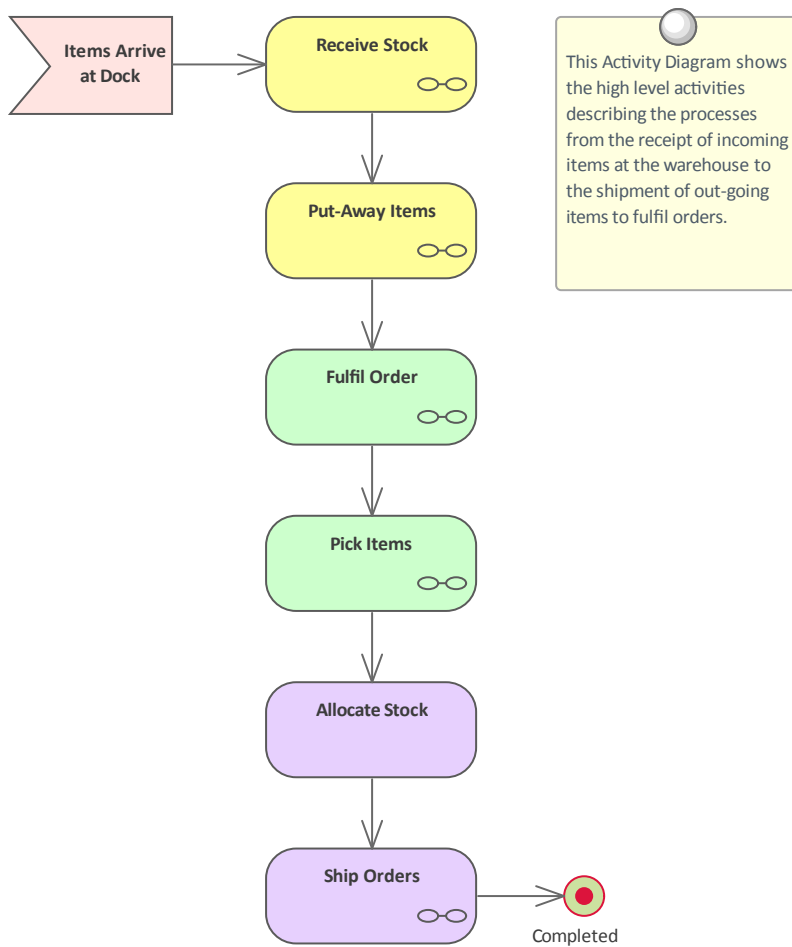
The success of an architecture will depend largely on the way it is implemented. The implementation could be at a business or technology level, but will typically involve both. The implementation teams are therefore important stakeholders to the enterprise architecture program and the architectures developed must be relevant, useful, meaningful, understandable and implementable. A practical way that this can be achieved is to involve the Solution Architects and Development Leads in the creation of the architectures, or at least seek their input. One of the key issues with the governance or oversight of implementation projects is visibility.

Enterprise Architect is well positioned to assist in the oversight of implementation projects by virtue of the fact that both the Enterprise Architecture and the Implementation projects can reside in the same repository, allowing visibility and traceability between the two disciplines. As work is completed at an implementation level it can be reviewed for compliance, and interventions or dispensations can be applied as required.

# Business and Systems Improvement

One or more architectures can be created that describe the business operational and technology systems for the purposes of improving the systems from any one of a number of perspectives such as efficiency or responding to business or technical forces. An architecture provides a fundamental understanding of the way the systems are structured and when analyzed can be used to determine where the opportunities for change exist, what the constraints are and how best to transition to an improved state.

Enterprise Architect has a number of tools that can assist in creating architectures for the purpose of business and information systems improvement.



# Communication

The success of an Enterprise Architecture program and the architectures that it produces will depend largely on how well the program's worth is communicated to interested stakeholders. The very strategic nature of the work and its importance as a guide to ensure the strategies are executed, means that the program's outputs will be critical to a wide range of stakeholders from senior executives down to business and technology implementation teams.

The management of the Enterprise Architecture program should include a communication plan that identifies the important stakeholders who are impacted by the program's work, determines their needs, and devises the best way of communicating with them, either as a group or as individuals. The communication plan will typically take the form of a document and while there would be an overall program level plan each architecture could have its own version of this plan, which identifies the stakeholders affected by that work. The types of communication device can include: formal or informal oral presentations, oral or written status reports, one page summary documents, walk-throughs, slide shows and one page diagrams.

Enterprise Architect has a wide range of tools that can assist with the development of the communication plan and the dissemination of information to stakeholders. The most obvious of these is to provide access to the repository so stakeholders can be directed to the parts of the model that

are of interest to them. The **Model Views** facility can be used to tailor views that are relevant and meaningful to individuals or groups of stakeholders. The search facility can also be used to locate information in the repository for example the Applications supplied by a particular vendor or that have a particular sunset date.

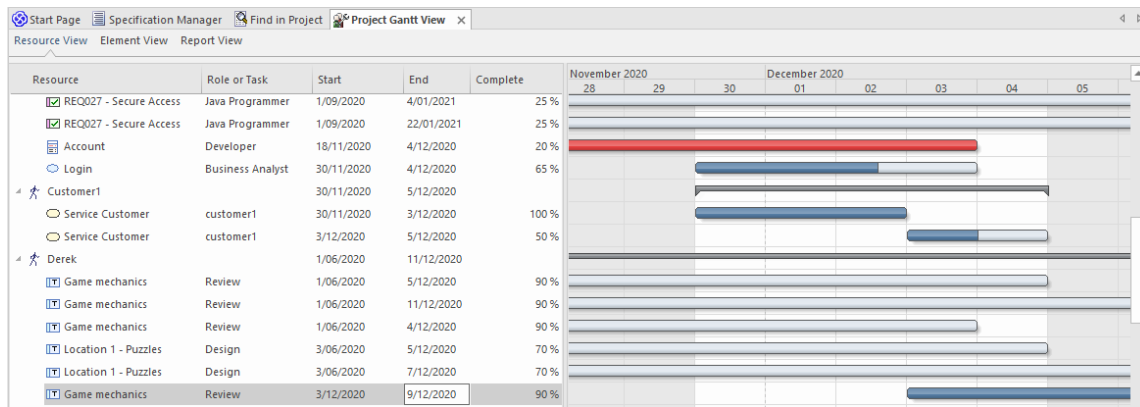


# Enterprise Transitions

When implemented, the architectures will transition an organization from the baseline (current) architecture to the target (future) architecture. These transitions must be carefully planned and orchestrated and should be implemented as part of an overall program or project level management plan. The transitions will typically involve a series of well orchestrated changes that involve existing and new business functions, information models, applications and technology platforms.

Enterprise Architect has a powerful and flexible Roadmap overlay that can be applied to any diagram and that can be used to show how the various parts of the architecture need to be implemented and integrated with existing architectures over a given time period. The Roadmap can be applied to any diagram and can show the phases of change for elements such as Business Capabilities, Information Models, Applications and Technology platforms. There is a wide range of other tools that will assist with transition planning and execution, such as the Document Artifact that can be used to create a transition plan and a Gantt Chart that can be used for scheduling architecture and other resources, and more.

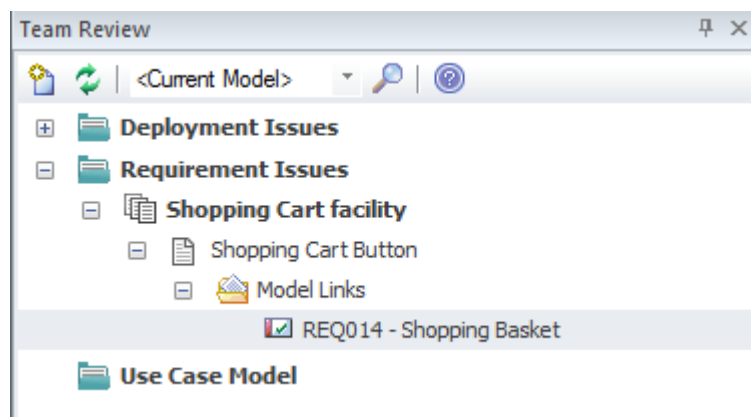




# Implementation Guide

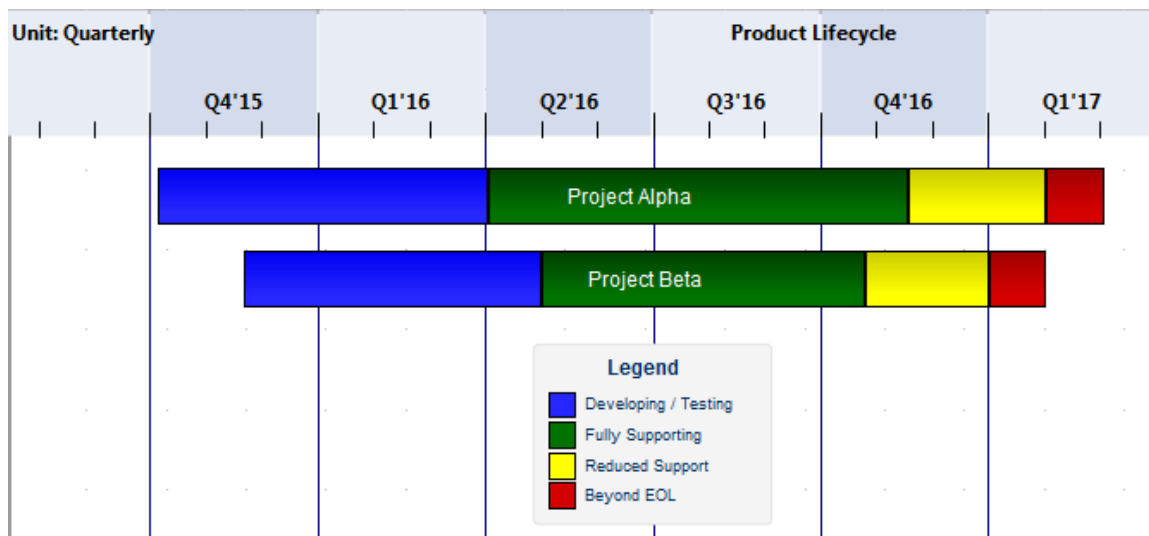
The architectures that are created and maintained by the Enterprise Architecture Program must be implemented to realize the business goals and objectives they set out to achieve. These architectures will be an invaluable guide to implementation teams irrespective of whether a solution is being purchased or developed in-house. The implementation will typically involve both business and technology components and the architecture will act as a guide to both groups to ensure that the implementation projects align with higher level enterprise initiatives. The guidance and governance might be formal and require contracts, or informal based on regular meetings, and will typically involve a liaison with the Project Management Office or its equivalent.

Enterprise Architect is well positioned to assist in this guidance and governance as both the architecture models and the implementation models can reside in the one repository, allowing visibility and traceability between the two disciplines. Principles can be applied at an implementation level showing the significance and applicability of the principle at the implementation level, including any dispensations if they have been granted.



# Portfolio Management

The architecture program must work closely with program and project management offices to ensure the target architectures are implemented within the context of the plans and schedules defined by these groups. While the architectural office will have an import role in the oversight of implementation projects they must work closely with the program and project management offices to plan and schedule the execution of their architectures.

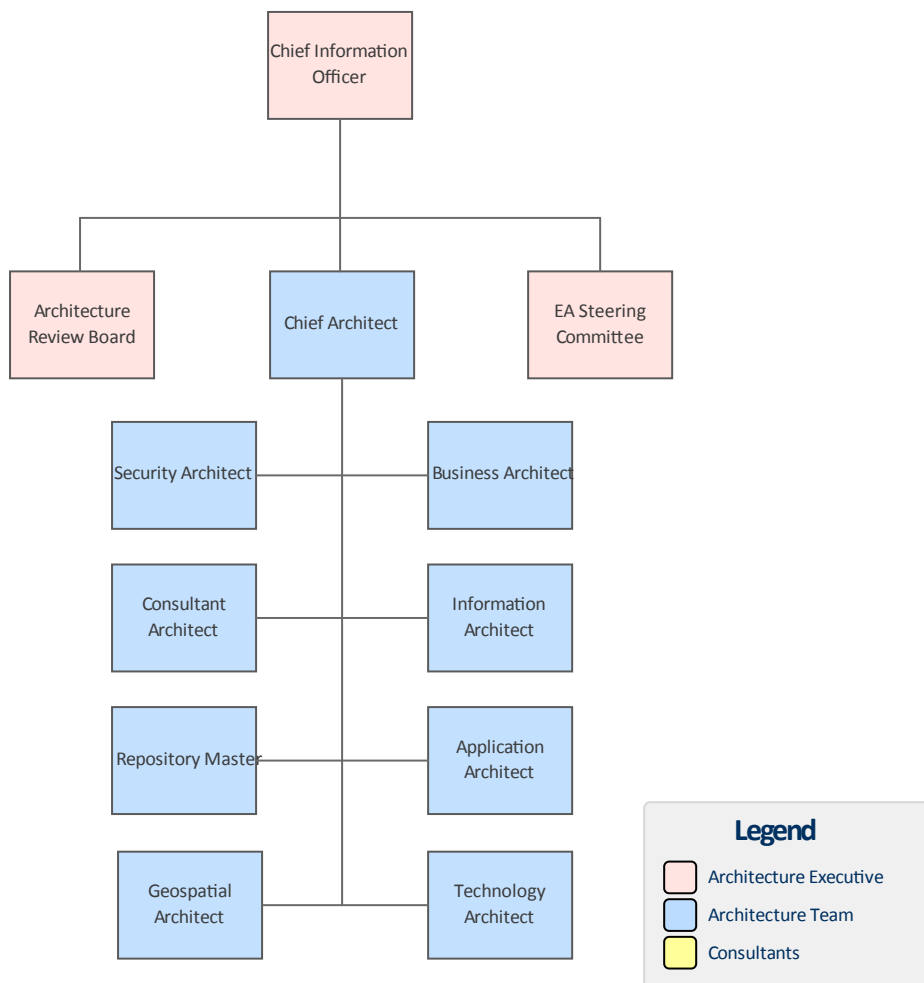


# Architecture Program Setup

Critical to the initiation and continuing success of an Enterprise Architecture Program is buy-in from the executive level. This requires them to have an appreciation of the strategic importance and value the Enterprise Architecture Program can bring to an enterprise, and to understand that for it to be effective the Enterprise Architecture team must have access to the entire organization and all its divisions.

# Management Structure

The Enterprise Architecture Office needs to work closely with the Program Management Office or its equivalent. The Program Management Office typically has a strategic planning role that evaluates the strategic benefit, risk and cost associated with programs and projects. It is critical that they are able to understand the outputs of the Architecture Office so as to appreciate the strategic value proposed by the architectures. The Chief Architect has an important role to liaise with this Program Management Office to ensure that the architectures are valued and that programs and projects are created that realize the architectures in implementation initiatives.



## Architecture Office Setup

An Architecture Office should enjoy the sponsorship of the executive and should be fully budgeted and funded as a program and monitored against performance criteria. The Chief Architect should have a budget that allows the right resources to be assigned and if necessary procured and research to be conducted where necessary to determine the best architectures. Architectures should be considered as work items that need to be monitored and their value assessed.

Enterprise Architect has powerful and flexible Kanban

support that can be used to manage the creation and development of architectures. Phases can be set up and work items can be moved through the Kanban board assigning architecture resources and viewing their progress visually.

## Enterprise Architecture Steering Committee

An Enterprise Architecture Steering Committee is an important body that will guide and direct the Architecture program and most importantly will be the decision making body. The members of the Steering committee should have the authority to enforce the decisions and be able to provide or acquire financial, material and human resources to achieve the outcomes required. The committee should be made up of members representing all major business and technology divisions of the enterprise, and given the power as overseers of the Enterprise Architecture Program from a strategic point of view and to act as a liaison with the Enterprise Portfolio Management Office.

Enterprise Architect can be used to define the structure of the Steering Committee, and to provide the members of the Committee with the necessary visualizations of the architectures, so that when important decisions are required the members can be well informed and have custom views that provide relevant and necessary information about the architectures, enabling them to make informed decisions.



## Chief Enterprise Architect

The Chief Enterprise Architect is the leader of the Architecture Team and is the liaison between the executive level stakeholders, the line managers of the business units and the architects that make up the Architecture Team. This key person should possess all the skills of a Strategic thinker, Architecture Designer, Planner, Policy Manager, Systems Engineer, Diplomat and Communicator. The Chief Architect should have a strong command of the architecture framework and the Repository and be able to understand how to use the tool to create and manage the architectures, govern the implementation initiatives, and create visualizations that communicate how the architectures are moving the enterprise in the direction of its goals.

Enterprise Architect has a number of tools that will support the Chief Architect in doing their work. These include Dashboard diagrams that give an overview of a property or facet of the repository and provide compelling charts that help information to be visualized. The Roadmap diagrams can be used to show how the architecture landscape will change over time, to describe the transition from one state to another, or even to describe the relationship between the architectures themselves. Diagrams can be created using images for the purpose of communication with executives, the Steering Committee and Line Managers. The Chief Architect is typically the first appointment and has the responsibility of selecting the architects who will make up the Architecture Team.

## Architecture Team

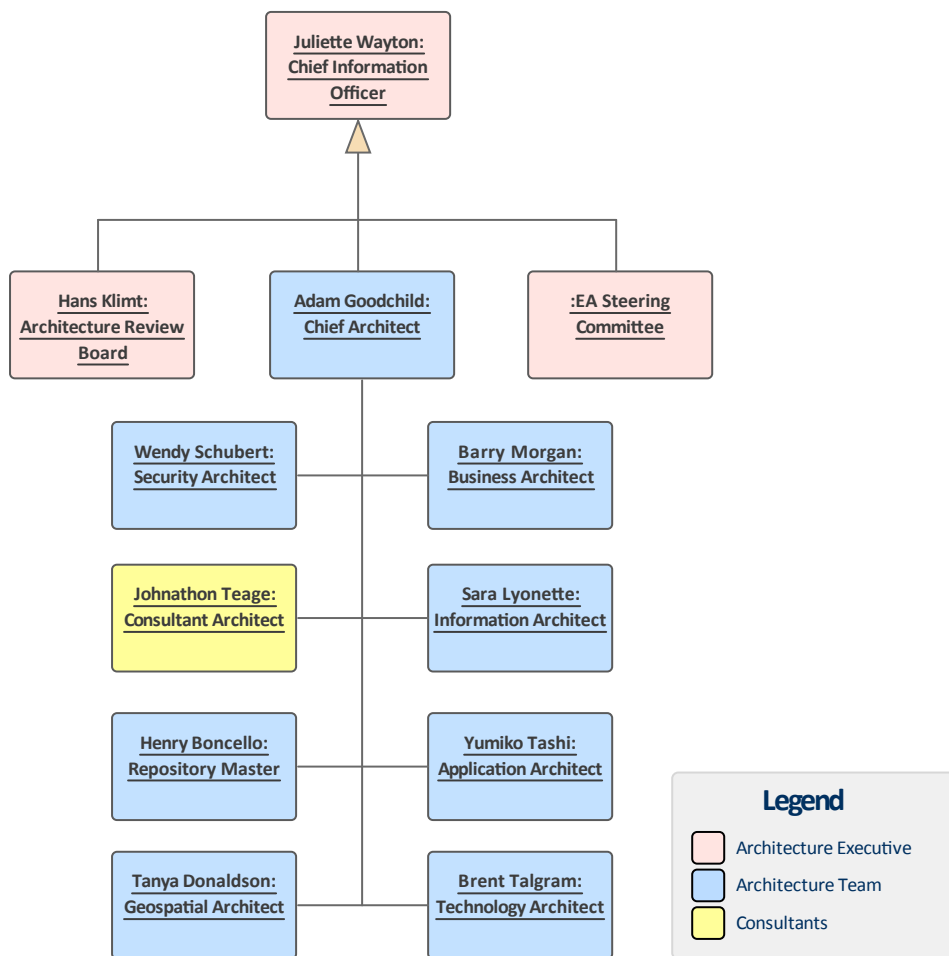
The Architecture Team is typically chosen and assembled by the chief architect, who must ensure that the members have the right competencies and discipline to carry out the work and to collaborate as a team to ensure the work in the architecture domains is integrated into a coherent, comprehensive and propitious enterprise architecture. It is quite common to assign a single person to an architecture domain, but is also common and fortuitous to have some overlap between the roles; for example, a Business Architect having some knowledge of Information Architecture and an Application Architect having knowledge of Technology Architecture. The most common positions for the team are:

- Business Architect
- Information Architect
- Application Architect
- Technology Architect

In addition there are several other architects that are often appointed where there is the need for a focus on a particular domain. The focus could be on security, such as in financial organizations, or geospatial concerns in an organization that deals with distributed infrastructure services, such as an energy distribution organization, or on social aspects where there is a significant social media bias to an organization, such as a marketing or publicity organization.

- Security Architect
- Geospatial Architect
- Social Architect

Enterprise Architect has a wide range of tools to assist the architects in creating and managing the domain specific architectures and the management of the architects that make up the team. This includes an Organizational chart that can be used to define the structure of the team, who holds the positions, and potentially their relationships to key business and technical stakeholders. Security can be enabled in the tool to encourage collaboration between the architects and to protect parts of the model from change. The powerful **Virtual Documents** facility can be used to create compelling and cohesive architectural publications and documentation that spans the architecture domains.



## Program Management Plan

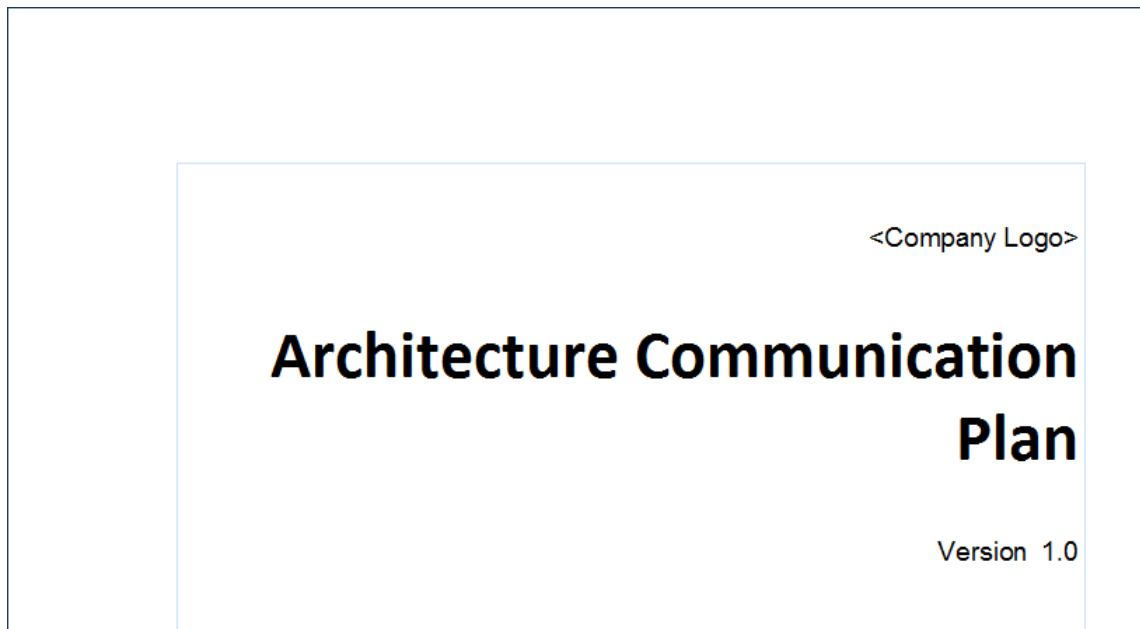
The Enterprise Architecture Office needs to work closely with the Program Management Office or its equivalent. The Program Management Office typically has a strategic planning role that evaluates the strategic benefit, risk and cost associated with programs and projects. It is critical that they are able to understand the outputs of the Architecture Office so as to appreciate the strategic value proposed by the architectures. The Chief Architect has an important role to liaise with this Program Management Office to ensure that

the architectures are valued and that programs and projects are created that realize the architectures in implementation initiatives.

## Communication Plan

Timely and frequent communication with the people and parties that hold a stake in the Enterprise Architecture effort is critical to the success of the program and the architectures it creates and disseminates. Each stakeholder group and also occasionally some individual stakeholders will require different information and will need to be kept informed at different times and with varying update frequencies. A plan should be created for each architecture but an enterprise plan could be devised that catalogs generalized communication needs of the stakeholder groups; this could be copied and modified for each distinct architecture where there is variation from the general plan.

Enterprise Architect has a Document Artifact facility that can be used to create any number of Communication Plans directly inside the repository. These plans can be viewed inside the repository or exported to an external file and saved into a variety of formats. A template can be created and be used as the basis of new plans, and there is the ability to include links to repository content within the document.

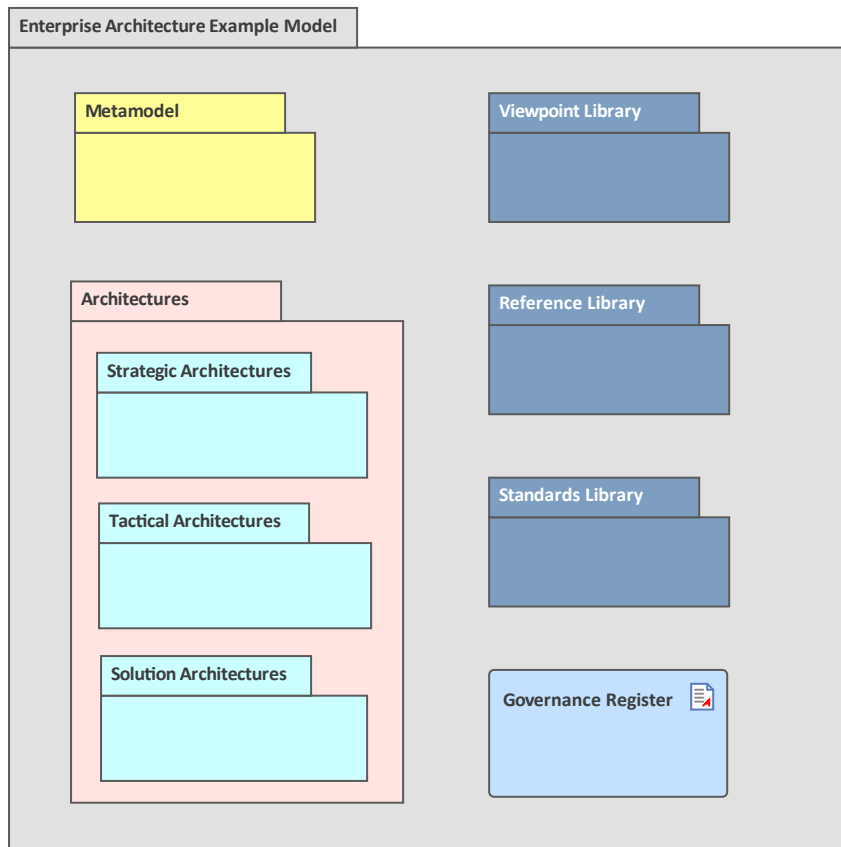


# Architecture Framework

An architecture Framework contains both the tools and the methods for creating and managing enterprise architectures. This includes the processes and tools that are used to create the architectures, the repository that stores the architecture content, and the organization of the team, including guidance on how to create the architectures and the governance of the teams that implement the architectures.

Enterprise Architect is a powerful and fully featured platform that provides all the facilities required to create, operate and manage an architectural framework. The tool can be used to define and operate an architecture process or method and has support for the leading industry frameworks such as The Open Group's Architecture Framework (TOGAF) and the Zachman Framework. Enterprise Architect can also act as the repository for architectural content and can store architectures at Strategic, Tactical or Solution levels and across the Business, Information, Application and Technology Domains.

The tool has been developed by practitioners and has a pragmatic and practical feature set that can be used to create content and views that will appeal to executive, management and operational level stakeholders alike. Content from a wide range of sources can be imported into Enterprise Architect allowing the architect to reuse existing architectural assets that reside in other tools.

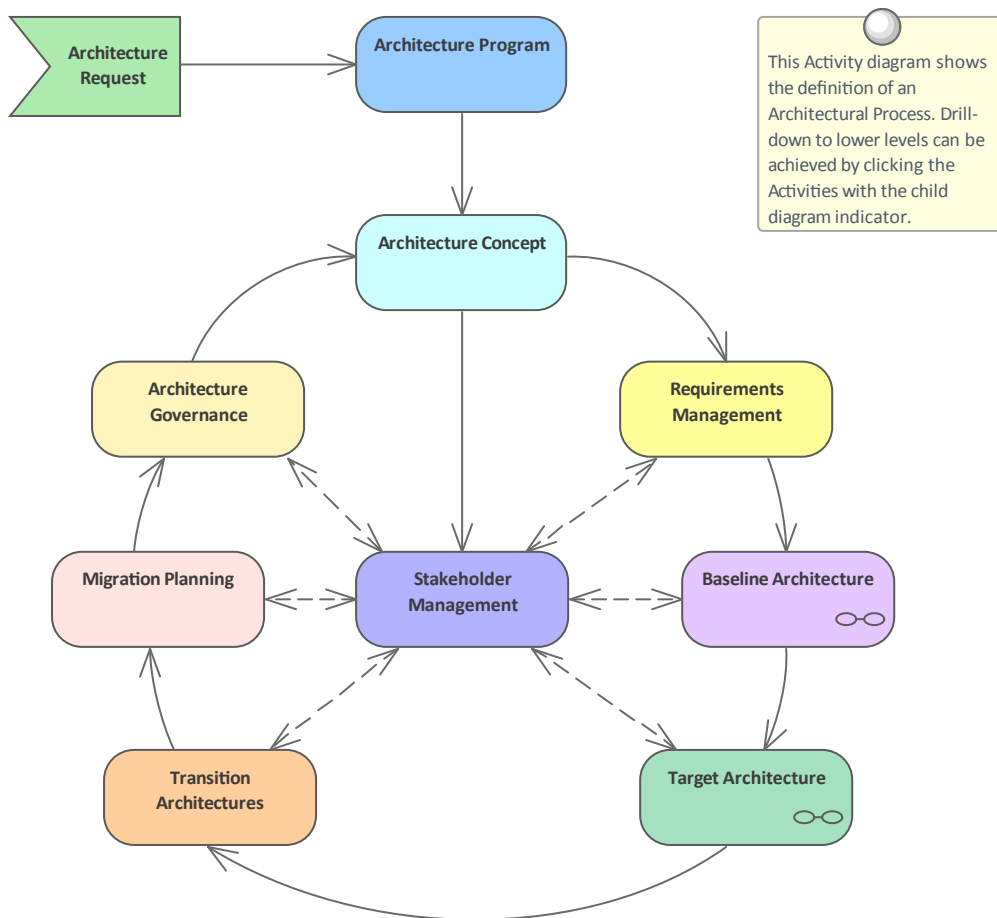




# Architecture Process

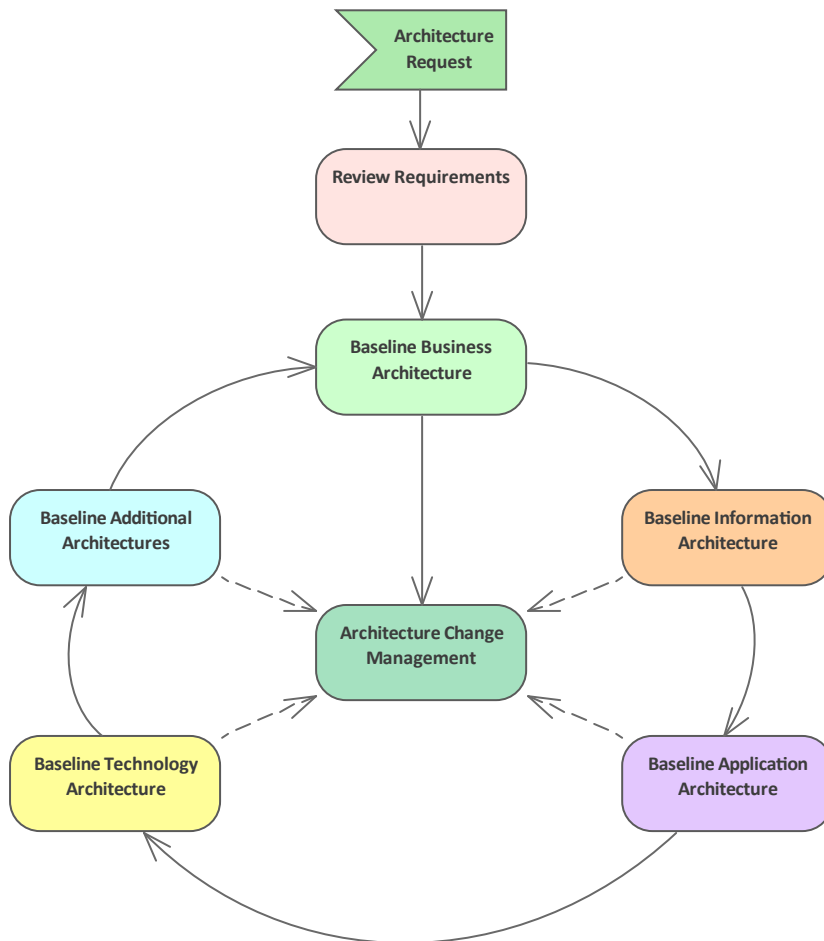
An architecture process or method is required to prescribe the way architectures should be developed. Not all frameworks provide a defined process, leaving it up to an organization to create and configure their own process. Enterprise Architect can be used to define a process at any level of detail and, using the drill-down to child diagram feature, more granular aspects of the process can be elaborated. UML Activity diagrams can be used to create a suite of diagrams that express the process of architecture creation, including the processes, the tasks, the inputs and outputs, and the people that perform the various steps in the process. Enterprise Architect also has an extension called the Software Process Engineering Metamodel (SPEM) that can be used to define the process with great rigor if required. In most cases the UML Activity diagram will be sufficient to create a detailed process.

Learn More: [Activity Diagram](#)



Baseline Architectures are often difficult to justify to executive and management level stakeholders who have a greater appetite for target architecture and roadmaps. The importance of the baseline architectures is to establish the starting point that will allow transitions to the target architectures to be defined. It is often the case that documentation and models will exist that can be mined to collect material to populate the baseline repository. For example most organizations have had at least a couple of attempts at modeling the existing processes, perhaps as part of a business re-engineering effort and one or more information models and hardware diagrams will often exist. Enterprise Architect can be used to import existing content

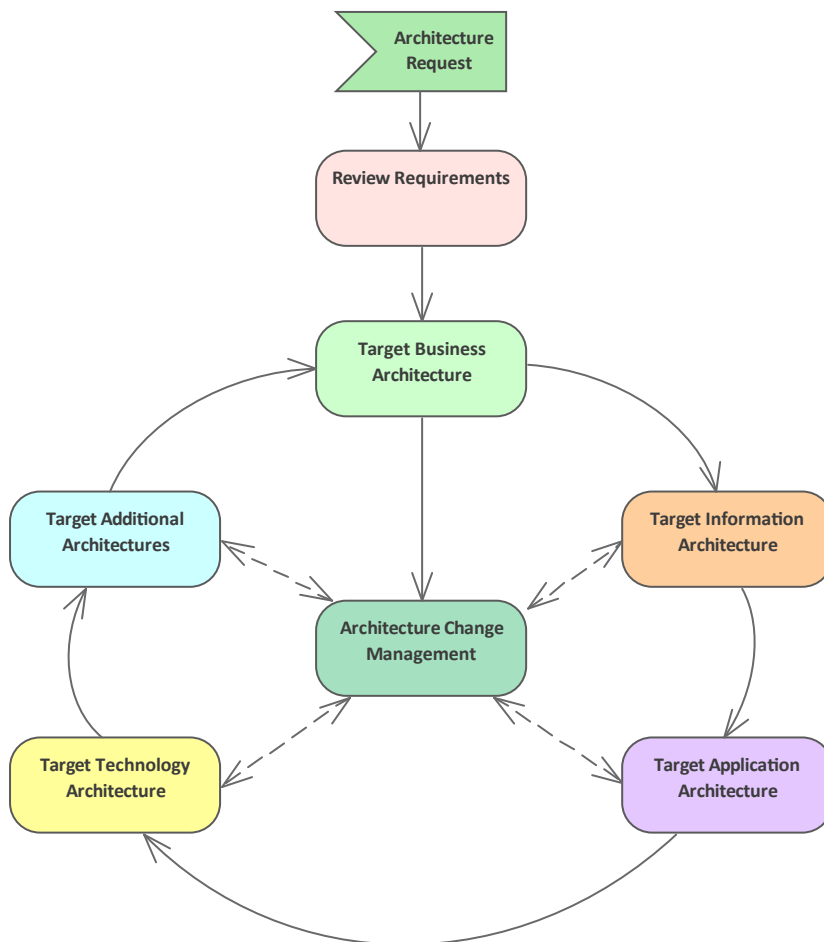
or models from other repositories and to reverse engineer data models that can form the basis of information architecture descriptions.



Target Architectures are vital for the executives and line managers because they define the architectures that will realize the business strategies and deliver value to the enterprise. Once these are known and there is enough elaborated in the baseline architectures, the architecture team can set about the more difficult task of defining the transition architectures and creating the roadmaps that will prescribe how the target architecture can be achieved in

practice using transitional steps.

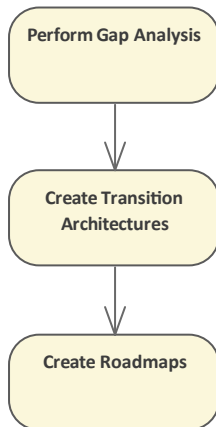
Enterprise Architect has a wide range of tools that allow the target architectures to be defined for all architecture domains, including Business, Information, Application, Technology, Security, Geospatial and Social architectures. Diagrams can be created and presented in a wide range of visualizations and styles, to be relevant to the executive and implementation stakeholders alike. Tools such as the **Specification Manager** and the List View enable you to work with catalogs in an appealing word processor or spreadsheet view. Impact and relationships can be analyzed using the Relationship and Gap Analysis Matrices and the **Traceability window**.



Transition Architectures are the stepping stones from a baseline architecture to a final target architecture, and are theoretically target architectures themselves. They represent the practical steps for moving from a current state to some aspired-to future state, and will often be represented as projects or phases in a project at an implementation level. Enterprise Architect allows Transition Architectures to be defined and related in a sequence of Roadmaps to be created so the transitions from one state to the next can be visualized and planned. Any number of Roadmaps can be defined, and they can be used for all the architecture

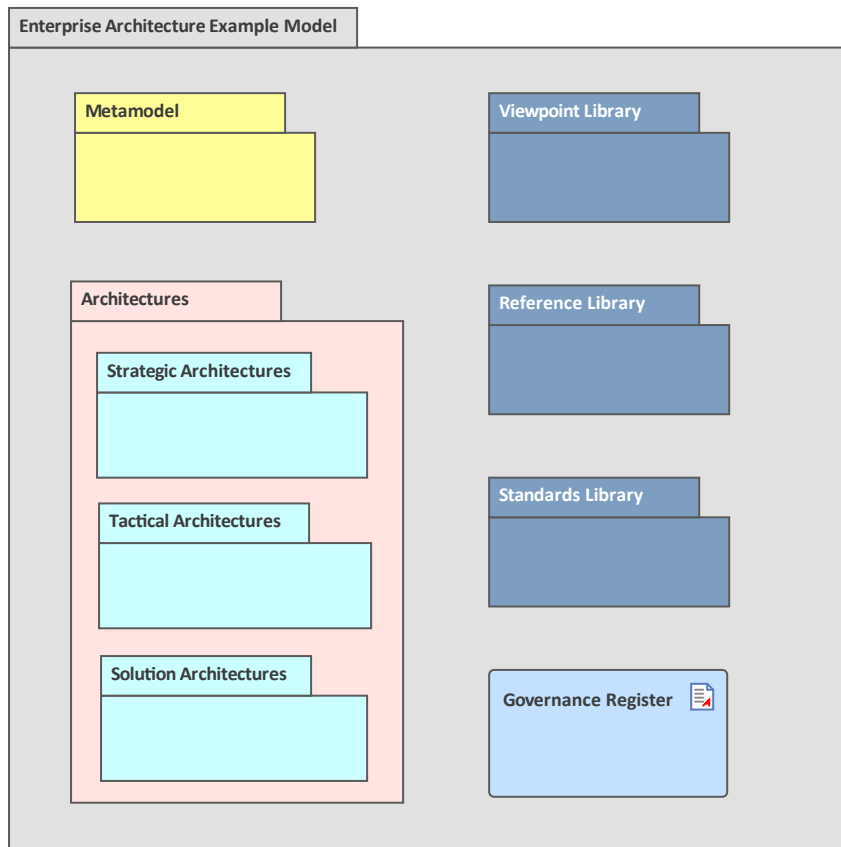
domains so there can be, for example, Capability, Application, and Technology Roadmaps.

Learn More: [Roadmap Diagram](#)



# Architecture Repository

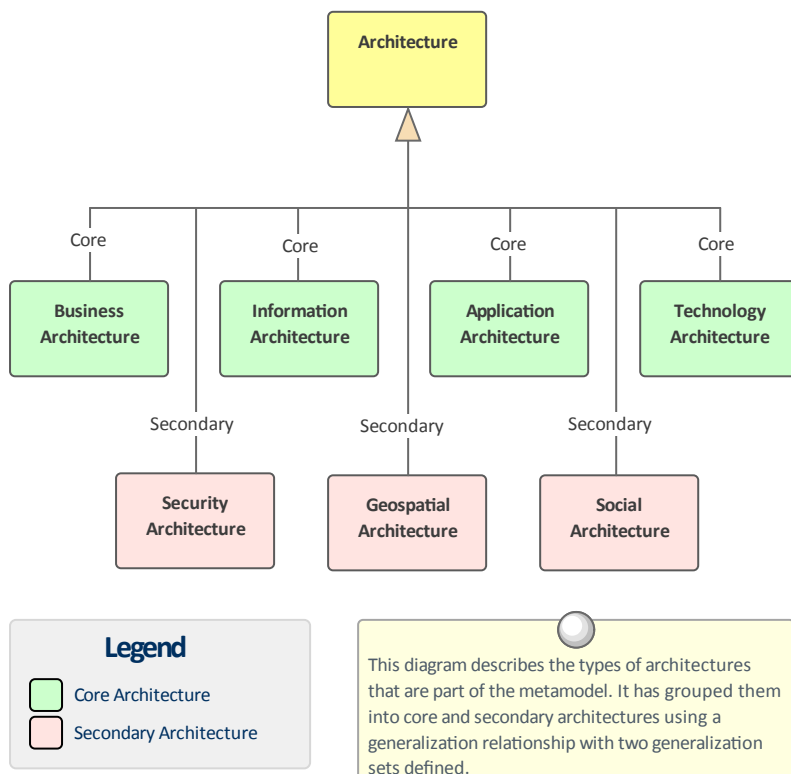
The Architecture Repository is a software tool that stores the important architectural input and output, including Architectures themselves, the elements of which they are composed, standards, references, principles and the Governance Register. Regardless of the Architecture Framework that has been selected, Enterprise Architect is a fully featured Architecture Repository that can store and manage all architectural content. The tool has a powerful set of features that allows a program to be set up efficiently, content to be imported from other sources, views and viewpoints to be defined and created, high quality publications to be generated, and much more. The repository contains a number of key parts, as represented on this diagram.



## Metamodel

The metamodel is a model of a model that describes the elements and relationships that can be used to construct the architectures. The metamodel functions as a grammar that defines the types and how they can be related to each other. It is critical to have defined the metamodel before any architectural models are created, as it will inform architects what elements they should use and how they can be used together. The metamodel in this format is a passive model and only acts as a guide and communication tool; if a more formal model is required a UML Profile could be created.





## Profiles

Enterprise Architect has a facility to create a Profile, which is a UML extension mechanism for creating domain and platform specific extensions of the UML. A Profile contains a collection of stereotypes, Tagged Values and constraints that refine the basic UML grammar but remain compliant with the fundamental aspects of the language. The Profile can be created in Enterprise Architect using a toolbox of stereotyped elements, and then imported into the repository. The Profile defines stereotypes that extend primitive UML elements and relationships, which can contain Tagged

Values that define platform or domain specific properties. The Profile allows any number of Toolbox pages to be defined, which can contain the stereotyped elements and relationships, which in turn can be dragged and dropped onto diagrams.

Profiles can also be included in an MDG Technology, which is a more elaborate extension mechanism that can bundle a range of reusable assets such as Element, Toolbox and Diagram Profiles, Patterns, Document Templates, Searches, Scripts, Images, Workspace Layouts and more.

Stereotypes extend the deep grammar of the Unified Modeling Language (UML), so some thought should be given to whether a new Profile is really needed or if an existing Profile could be used.

## Principles

Principles are a fundamental set of architecture axioms that help ensure that the strategy is embodied in the architectural designs and concomitantly in the implementation solutions. They form a set of guidelines that are prescriptive rules intended to guide and constrain the way a solution is architected and implemented. Critical to the success of the principles is the way they are applied and made applicable in a particular context.

Enterprise Architect allows Principles to be modeled using a stereotyped **UML Class** where the properties of the

principles can be defined using Tagged Values. Instances of these Classes can then be created describing how the principle can be applied in the context of a Strategic, Tactical or Solution architecture. This is a critical step as the principles are typically abstract and defined in theoretical terms and often the principles are included in an appendix or another type of list with little or no explanation of their applicability to an architecture or solution.

## Architectures

The Architectures are the centerpiece of the Architecture Repository; they contain the designs and solutions to the problems and opportunities defined by the business in Strategic Modeling and are guided and constrained by the architecture principles.

## Tool Setup

Enterprise Architect is a sophisticated and powerful platform that can be used as an Architectural Repository. The platform consists of a wide range of tools that can be used from the set up and management of the program, through the creation of the architectures themselves to the governance of the implementation initiatives that realize the architectures. The platform has great flexibility and can be configured in a variety of ways, allowing each architecture program and team to get the most benefit from the tool. It is recommended to have an administrator and librarian role responsible for setting up the tool with the most appropriate configurations. These sections list some of the most important things to set up, which can be done prior to starting to develop architectures or after some of the work has commenced.

## Security

Enterprise Architect has a powerful role-based security system that is designed to encourage collaboration between modelers. The security system can be used to restrict updates to portions of the model, but it cannot restrict the view of a part of the model. Users and Groups can be defined, and both can be assigned any number of built-in permissions. The security system has two different modes (a completely locked or a completely open model) and at set

up a choice can be made between the two modes.

Model users can be set up by importing users from Active Directory, which then allows single sign in to Enterprise Architect (a user will not be challenged for credentials when opening Enterprise Architect) using Windows Authentication. Typical Groups might be: Administrators, Librarians, Modelers and Viewers.

Surname	Firstname	Login
Administrator	The	admin
Dean	Pauline	paulinedean
Howard	Tim	timhoward
Nichols	Greg	gregnichols
Nielsen	Ken	kennielson

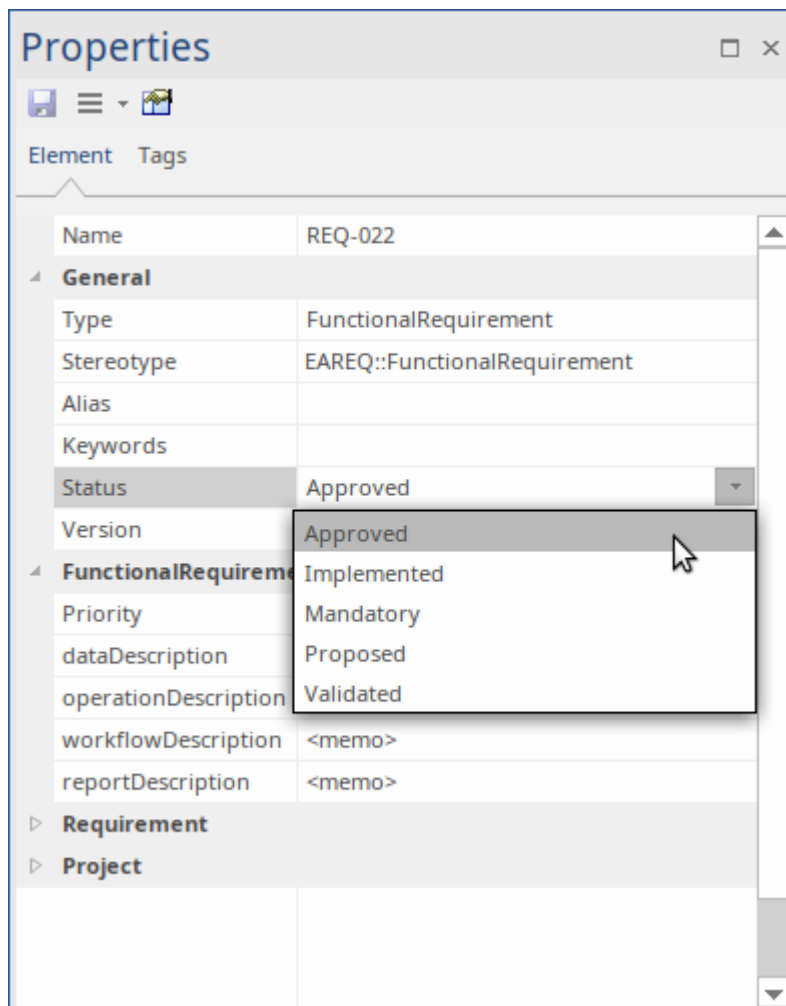
It is typically the role of an administrator or librarian to manage security. There are a number of other facilities that require security to be set up before they can be used, including **Model Mail** and Resource Allocation.

Learn More: [Security](#)

## Reference Data

Reference data is used to configure a number of aspects of Enterprise Architect, such as drop down lists. The tool

comes with a set of reference data that can be used out of the box, but it is recommended to review this data and configure it to make it fit for a particular team's purposes. An example is that each element has a Status property that is displayed in a drop down list on the element's property sheet (or in the **Properties window**).



The list of statuses and other reference data can be configured, and any number of statuses can be defined and then assigned to elements in the repository.

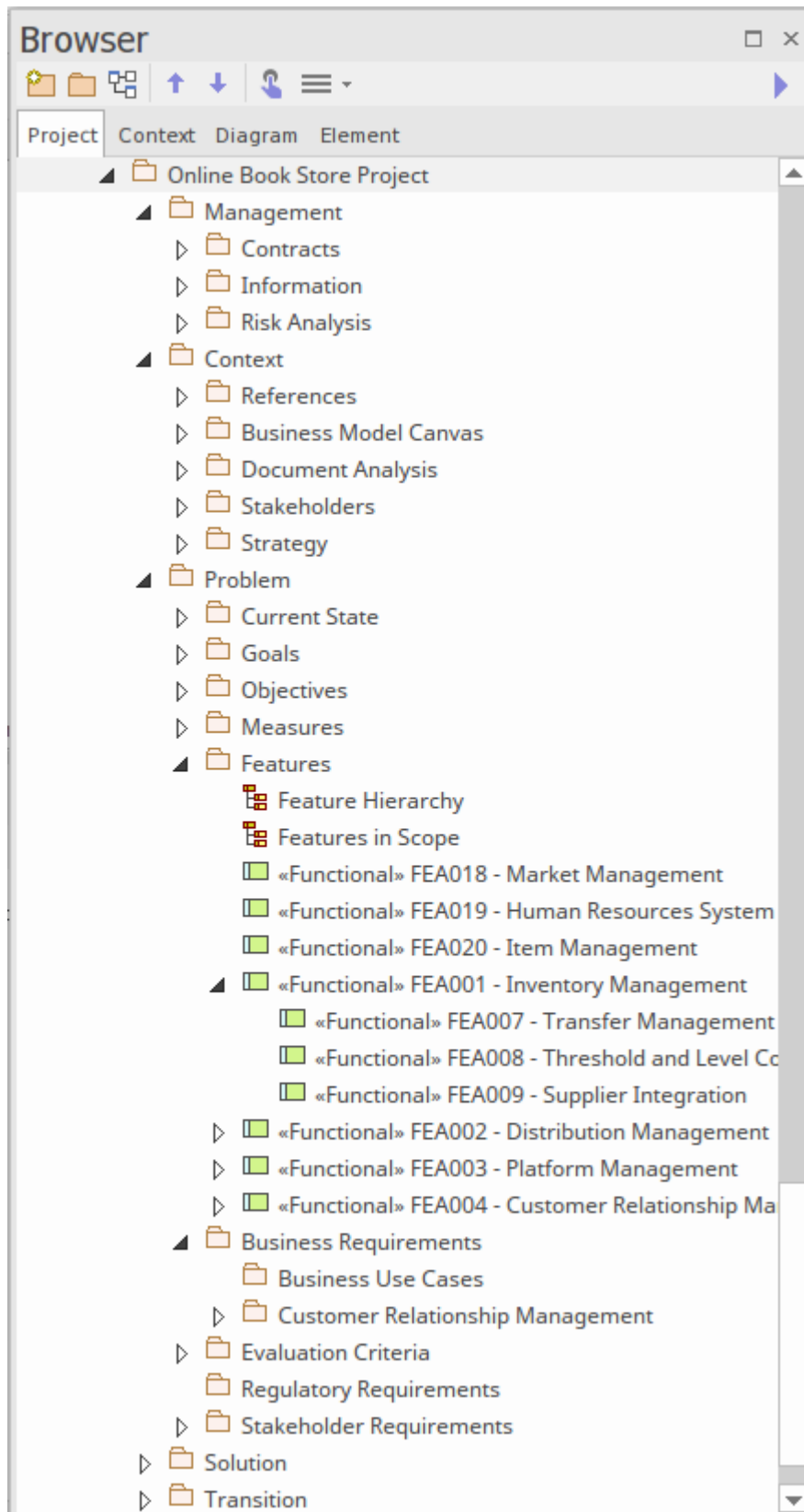
Type	Description
Approved	Item is approved
Implemented	Finished
Mandatory	Required
Proposed	Item has been proposed
Validated	Approved and Checked

Learn More: [Reference Data](#)

## Package Structure

The Package Structure defines the anatomy of the repository. Defining the layout of this structure will help with navigation and other facilities that work at a Package level, such as **Baselines**, **Version Control**, Documentation and more. The structure of the Packages in the repository can be changed at any time, and elements can be freely moved around between Packages if needed. It is, however, good practice to spend some time creating a well structured repository at the time of set up, as this will facilitate good modeling behavior and assist newcomers with locating elements and Packages in the repository.

It is typically a librarian or administrator who will set up and maintain the Package structure and who will receive requests to add, delete or merge Packages and elements.



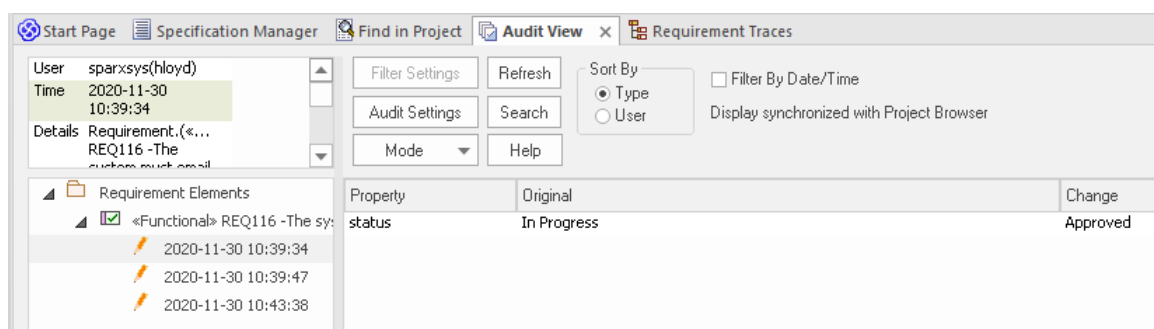


It is quite common to use an existing Package structure by importing a model from an XML file that has been exported from another team or organization.

Learn More: [Browser Window](#)

## Auditing

**Auditing** keeps a log of all the changes that have been made on a repository and once enabled will work silently in the background. It is a useful tool not so much for policing but for tuning the use of the repository. When parts of the repository have been changed incorrectly it is useful to be able to go to the auditor and view when and by whom the change was made. The audit facility can be configured to collect a range of data about a change and a number of different styles of presentation are available. The audit feature is by default disabled so if it is needed it must first be enabled.

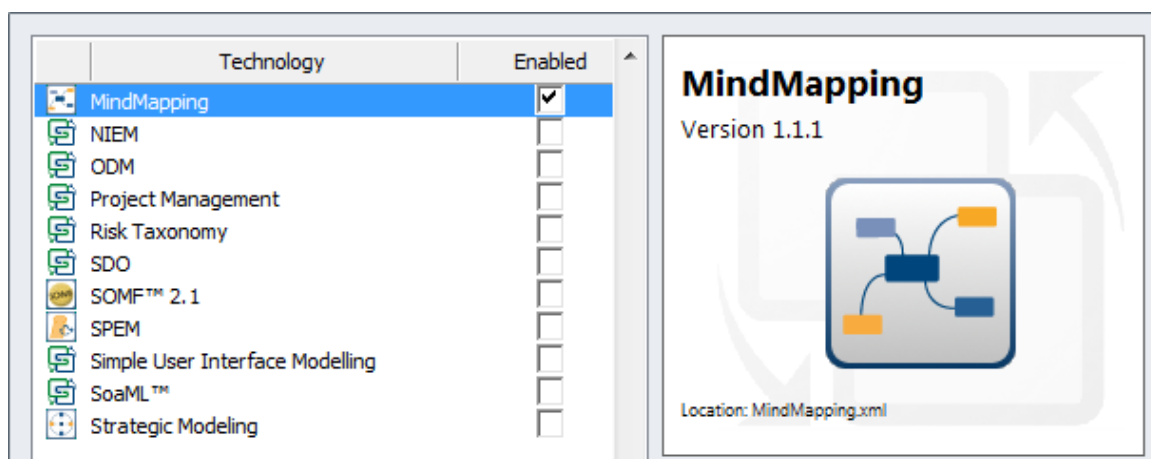


The audit logs are stored in the repository and so it is advised to archive them periodically to keep the repository trim.

Learn More: [Auditing](#)

## Extensions

Model Driven Generation Technology Extensions are a way of enabling additional features of the tool and are commonly referred to as Technologies or Extensions. There are a wide range of built-in extensions available ranging from Strategic Modeling - which contains facilities such as Balanced Scorecards - to Risk Taxonomies, Wire Framing, Mind Mapping and many more. A team can also create their own extensions (including modeling languages) to meet a particular modeling need. It is good practice to make a decision about which of these extensions will be used for a given program or initiative.



By disabling the technology in the **MDG Technologies window** any diagrams, toolboxes, images, workspace layouts, Patterns, templates and more will be hidden from the user.

Learn More: [Extensions](#)

## Profiles and Technologies

Enterprise Architect has a powerful extension facility based on the UML Profile, which allows the language to be extended to make it suitable for a variety of modeling domains. A profile can be created in the model and then imported, allowing new elements, toolboxes and diagrams to be created. The new elements can be given additional properties in the form of Tagged Values and assigned graphical representation based on the domain being modeled. There is also a more elaborate extension mechanism called Model Driven Generation (MDG) Technologies that allow a profile to be bundled with a wide range of other assets such as Patterns, Document Templates, Searches, Scripts, Images, Workspace Layouts and more.

A team needs to make a decision whether it is appropriate to create a profile or MDG Technology at the time the repository is set up.

Learn More: [Profile](#)

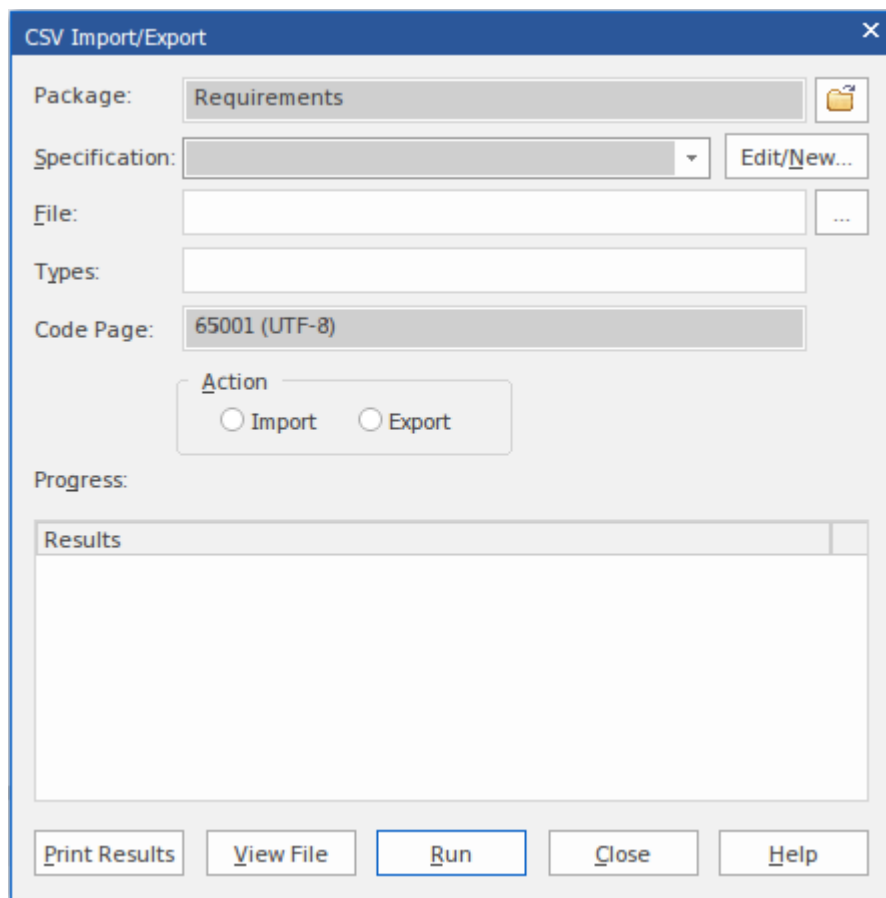
## Content Import

A quick way to get a new repository started is by importing content from existing files. It is quite common for a team to have started working before the tool has been set up, and much of the content that has been created can be

automatically imported into elements in the repository. This could include Principles, Requirements, Capabilities, Applications, Interface lists and more. The easiest way to import these lists of things is to use a spreadsheet file where the rows define the elements to be imported and the columns define properties of the elements; for example, the first columns could be the Name, the second column the Description and so on.

The screenshot shows the 'Import Specification' dialog box in Enterprise Architect. It is configured for importing a CSV file named 'Requirements.csv' with a comma delimiter. The 'Notes' field contains a description of the import. The 'Default Filename' is set to 'E:\Projects\Online BookStore\Requirements\RFT Phase One\Requii'. The 'Default Direction' is 'Import'. The 'Default Types' field is empty. The 'Preserve Hierarchy' checkbox is unchecked. The 'Available Fields' list includes 'Type', 'GUID', 'Priority', 'Stereotype', and 'Fully Qualified Stereotype'. The 'Add Tagged Value Field' button is visible. The 'File Specification' section has 'Up' and 'Down' arrows. The 'Select Element Field' list includes 'Name', 'Notes', 'Phase', 'Version', and 'Status'.

Enterprise Architect defines a mapping that can be configured to describe how columns in the spreadsheet (CSV file) map to element properties and extended properties known as Tagged Values.

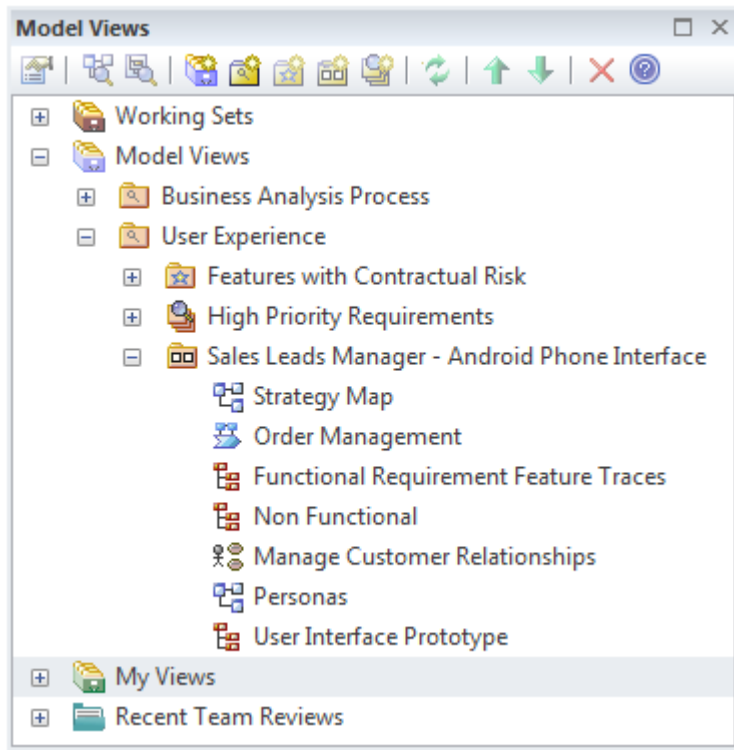


Learn More: [Import and Export Spreadsheets](#)

## Model Views

**Model Views** is a powerful facility that allows a modeler to create any number of different representation of the Packages and elements in the repository. The repository Package structure will typically be set up to assist with navigation and ease of content creation and will not provide the views that are important for some stakeholders. The model view facility is a convenient way to create views of the elements in the repository to assist stakeholders working or viewing the model. For example a **Model View** could be

set up to list all Applications with a status of proposed that will be decommissioned before the end of the financial year.



Models views that are based on searches can be set up at the time the repository is created, model views based on slide shows and favorites will need to be added as the elements and diagrams are added to the repository.

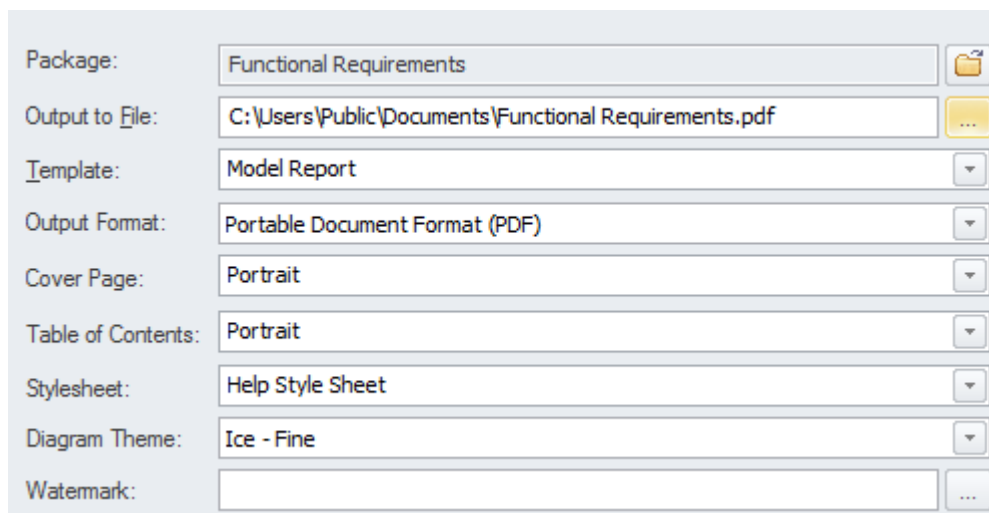
Learn More: [Model Views](#)

## Document Templates

Enterprise Architect has a sophisticated and flexible document and web page generation engine that can be used to create a wide range of content from the model, from high fidelity and polished publications and web sites to ad hoc reports. While it is good practice to encourage stakeholders

to view content directly in the model, it is inevitable that some form of document or web page reports or publications will be required. Apart from being able to configure the content it is also possible to completely tailor the style of the output based on a rich system of templates.

It is good practice to get a small group of people to spend some effort at the beginning of an initiative setting up templates for others to use. This allows the model to be seen as an important source of information, and stakeholders will return to it when they need information about the initiative.



The screenshot shows a configuration window with the following settings:

Package:	Functional Requirements	[Folder icon]
Output to File:	C:\Users\Public\Documents\Functional Requirements.pdf	[More options icon]
Template:	Model Report	[Dropdown arrow]
Output Format:	Portable Document Format (PDF)	[Dropdown arrow]
Cover Page:	Portrait	[Dropdown arrow]
Table of Contents:	Portrait	[Dropdown arrow]
Stylesheet:	Help Style Sheet	[Dropdown arrow]
Diagram Theme:	Ice - Fine	[Dropdown arrow]
Watermark:		[More options icon]

There is a wide range of options that can be set for each template; these can be stored in the repository and are available from the 'Resources' tab of the **Browser window**.

**Virtual Documents** can also be created once the Package structure of the repository has been defined; these can be collected together in a Publications Package for each initiative.

Learn More: [Documentation](#)

## Model Navigation

Model Navigation can be set up by creating any number of diagrams with images and hyperlinks to content in the model. A model home page can be created by setting a diagram as the model default diagram. This can link to other navigation pages that are tailored for particular stakeholder groups. The diagram-based navigation creates a soft entry point for people who are not familiar with the structure of the repository and who might not be familiar with using Enterprise Architect. Images from the domain that are familiar to users can be included, and these can have hyperlinks added. The pages will be carried through to HTML documentation and the hyperlinks will work on the generated web pages.

It is good practice to set these pages up as early as possible to encourage people to access the repository without fear of being disoriented.

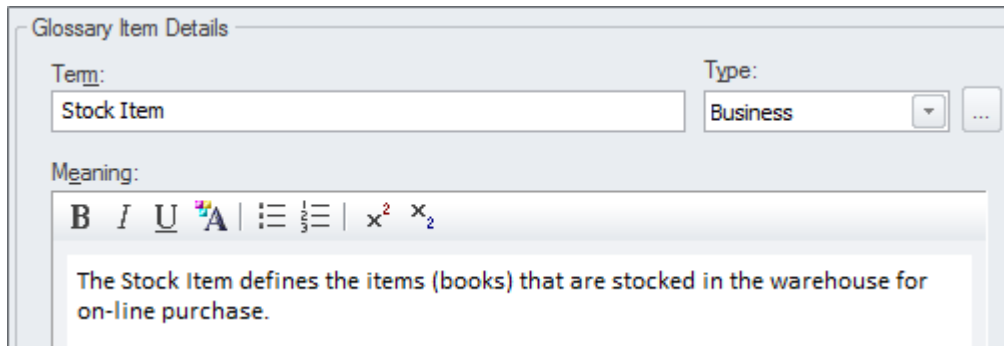
Learn More: [Model Navigation](#)

## Glossary

Enterprise Architect has a built-in Glossary where terms, their types and their definitions can be entered and stored in the Repository. The terms can be used in element notes and when a modeler rolls over the term the definition will be displayed in a small window. The terms can be also



included in publications and reports automatically created by the **Document Generator**.



Often an enterprise will have a list of terms that are stored in a document or spreadsheet and these can be imported into the tool. Alternatively a team might make a decision to model the important terms as a Class model, which allows more detailed information to be stored and relationships to be created between the terms. Either way it is good practice to set up the list of terms in Enterprise Architect at the time the repository is being defined.

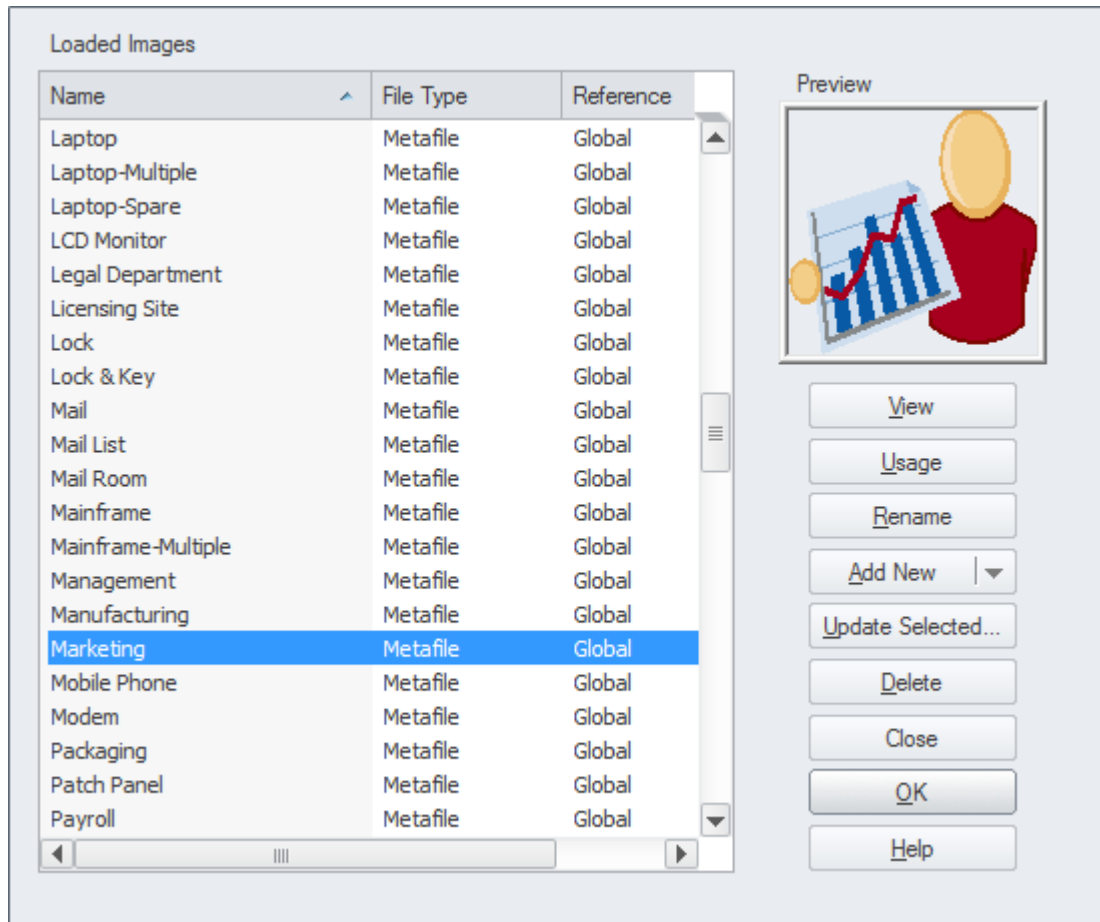
Learn More: [Glossary](#)

## Images

Images provide a welcome alternative to modeling language element representations such as rectangles and diamonds. Any element can be assigned an alternative image, creating compelling diagrams that are often more acceptable to some stakeholders groups, particularly business and senior management.

The images can simply be pasted onto any diagram, or they can be imported into the **Image Manager** for reuse within

the repository. It is recommended that some effort is spent at set up time collecting and importing images that will soften the diagrams intended for certain stakeholder groups. This will also have the effect of creating consistency between diagrams.



It is best to use a vector based format such as Windows Metafiles as these will allow the images to be scaled in diagrams.

Learn More: [Image Manager](#)

## Patterns

Patterns provide a powerful way of supplying redefined content for a diagram. Any number of Patterns can be defined and saved in the repository, relieving novice modelers of the responsibility of devising the structure of a diagram and resulting in model consistency.

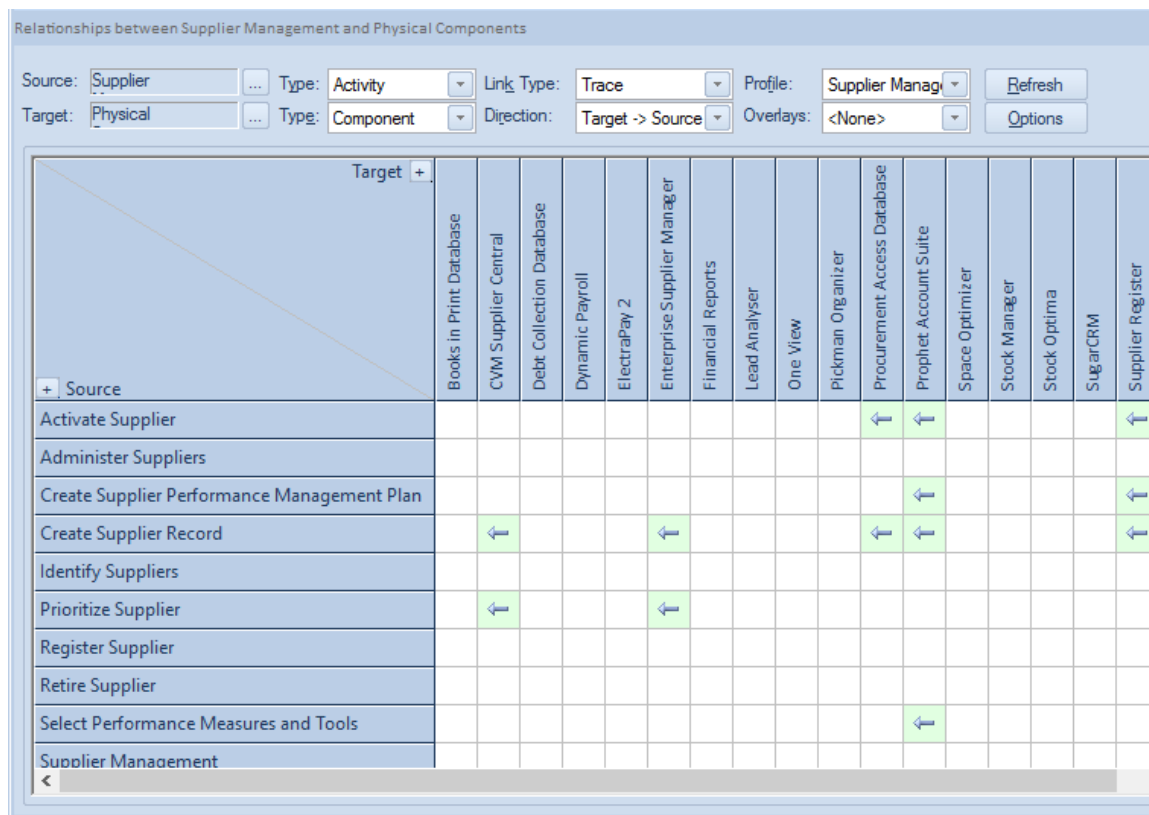
It is good practice to define a number of Patterns at the time of repository set up, giving them appropriate descriptions that define the context, their intent and how they can be applied.

Learn More: [Patterns](#)

## Matrix Profiles

Matrix Profiles define Relationship Matrices that provide a convenient way of presenting the fact that two elements (one on each axis of the matrix) have some type of relationship. An indicator will be drawn at the intersection of the two elements, showing the direction of the relationship. Overlays can be created, allowing user defined letters to represent the nature of the relationship.

It is good practice to set up the Matrix Profiles at the time the repository is set up, and save them as profiles that can be accessed from the 'Resources' tab of the Browser Window. They will not have elements listed, but as the repository is fleshed-out the elements will appear in the matrices.



Learn More: [Relationship Matrices](#)

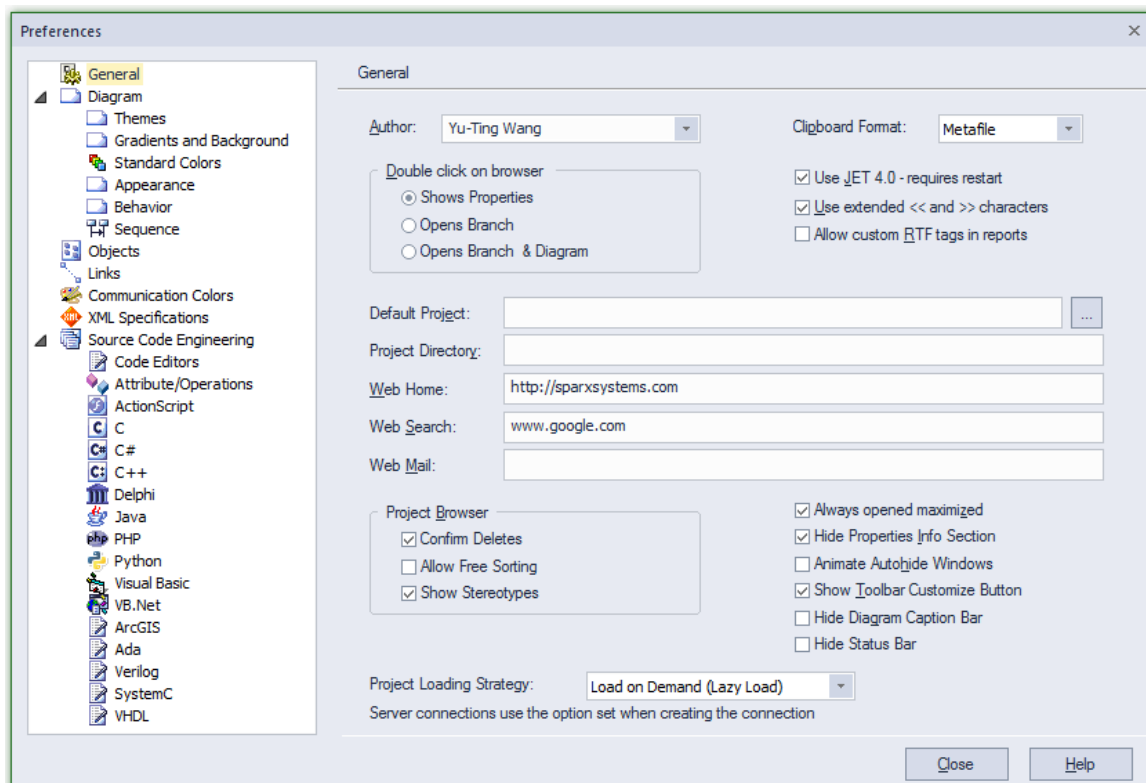
## Localization

Enterprise Architect is used all over the world, therefore it has a number of places where localization settings can be defined. It is good practice to define these at the time the repository is set up. They include a page setup that can be defined for each diagram, and the Dictionary used for **Spell Checking**.

Learn More: [Spell Checking Dictionaries](#)

## Preferences

Enterprise Architect is a highly configurable product and allows a user to set up a wide range of preferences to tailor the way the various tools work. Some settings apply to the entire repository and will be stored in the repository itself whilst others apply to the individual modeler and are typically stored in a user's Application Data.

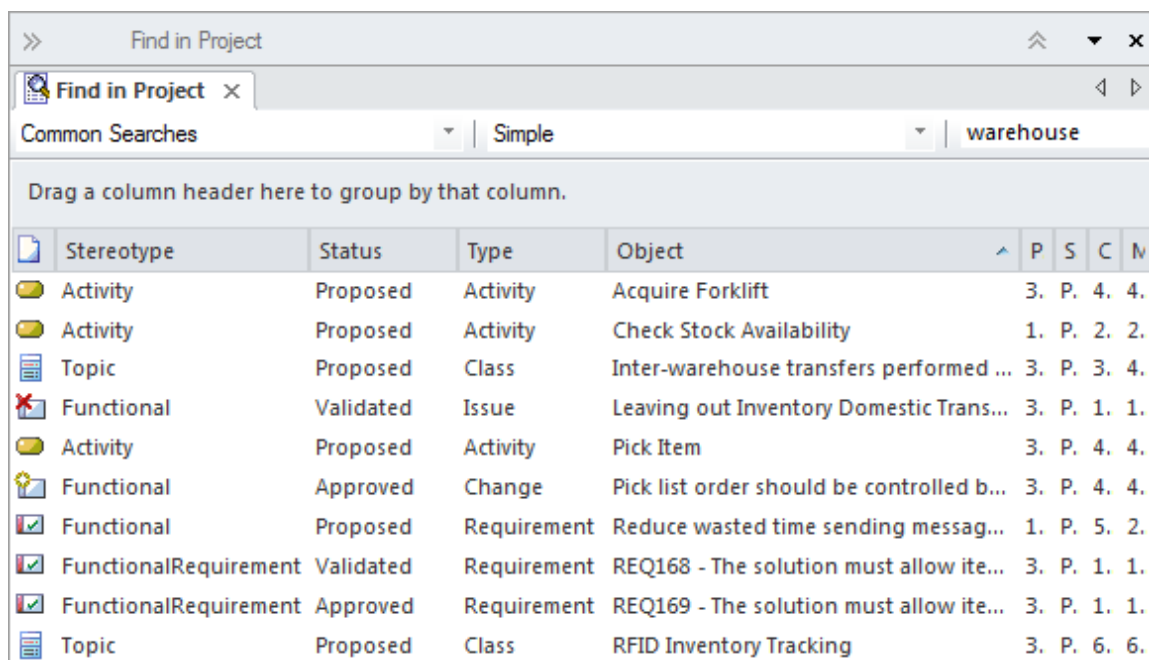


Learn More: [Local Options](#)

## Searches

As modeling progresses, a repository will become large quite quickly and finding things in the model will be important. Enterprise Architect has a flexible and powerful search facility that can be used to find any model content.

The results are returned to a window where individual items can be found in the **Browser window** or diagrams, and reports can be generated for one or more selected elements. Searches are also important because they are used in a variety of places, such as **Model Views**. There is a wide range of built-in searches that can be used to locate objects in the model, but you can also set up your own searches that can be saved and reused.



Stereotype	Status	Type	Object	P	S	C	N
Activity	Proposed	Activity	Acquire Forklift	3.	P.	4.	4.
Activity	Proposed	Activity	Check Stock Availability	1.	P.	2.	2.
Topic	Proposed	Class	Inter-warehouse transfers performed ...	3.	P.	3.	4.
Functional	Validated	Issue	Leaving out Inventory Domestic Trans...	3.	P.	1.	1.
Activity	Proposed	Activity	Pick Item	3.	P.	4.	4.
Functional	Approved	Change	Pick list order should be controlled b...	3.	P.	4.	4.
Functional	Proposed	Requirement	Reduce wasted time sending messag...	1.	P.	5.	2.
FunctionalRequirement	Validated	Requirement	REQ168 - The solution must allow ite...	3.	P.	1.	1.
FunctionalRequirement	Approved	Requirement	REQ169 - The solution must allow ite...	3.	P.	1.	1.
Topic	Proposed	Class	RFID Inventory Tracking	3.	P.	6.	6.

It is good practice to set up a number of common and frequently run searches at the time the repository is created. This will provide useful results for stakeholders but will also test the repository structure and metamodel definitions. It is quite common for new searches to be created as the model develops. A common trigger for this is when a stakeholder needs to interrogate the architecture to get information required for decision-making.

Learn More: [Model Search](#)

## Stereotypes

Stereotypes are one of the UML's extension mechanisms, used to create additional types to extend the language to make it more useful in a given domain or context. They can be given an alternative representation either as an image or a Shape Script. Stereotypes should be created by a repository librarian or administrator as they are effectively changing the grammar of the UML.

The screenshot shows a configuration window for a UML stereotype. At the top, there are three input fields: 'Stereotype:' with the text 'stakeholder', 'Group name:' which is empty, and 'Base Class:' with a dropdown menu showing 'class'. Below these is a 'Notes:' section containing the text: 'A stakeholder is a group or person who has interests that may be affected by an initiative or who have influence over the initiative.' Under the notes are three buttons: 'New', 'Save', and 'Delete'. The 'Override Appearance' section has three radio buttons: 'None', 'Metafile' (which is selected), and 'Shape Script'. To the right of these are 'Assign' and 'Remove' buttons. Below the radio buttons is a 'Default Colors' section with three color pickers for 'Fill:', 'Border:', and 'Font:', each with a 'De...' dropdown and a 'Reset' button. To the right of the color pickers is a 'Preview' window showing a pixelated icon of a person with a yellow head and a blue body.

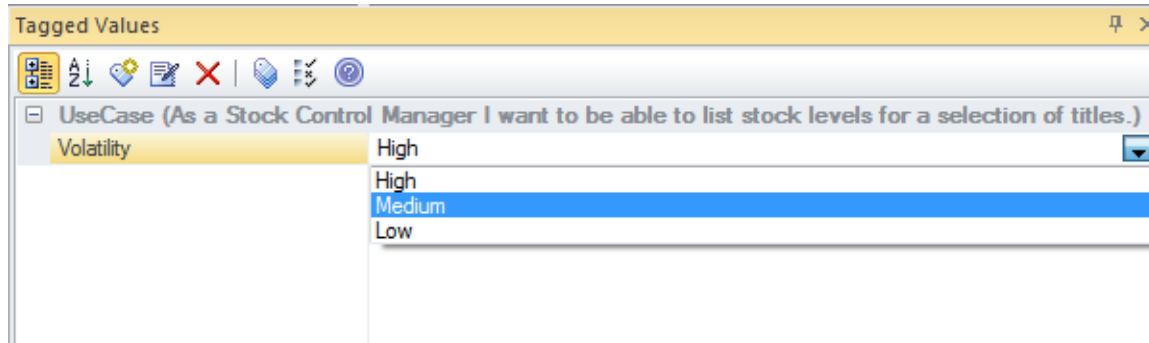
It is good practice to decide on what stereotypes, if any, are required at the time the repository is set up. They can, of course, be added as the program or the architecture matures

and can also be created as part of a Profile, which in turn can be incorporated into a Model Driven Technology extension.

Learn More: [Stereotyping](#)

## Tagged Values

Tagged Values are additional properties that can be added to a variety of items in the repository including elements, connectors, attributes and operations. There are a large number of built-in properties available for these items but a user is free to create Tagged Values when extra properties are required.



Learn More: [Tagged Values](#)

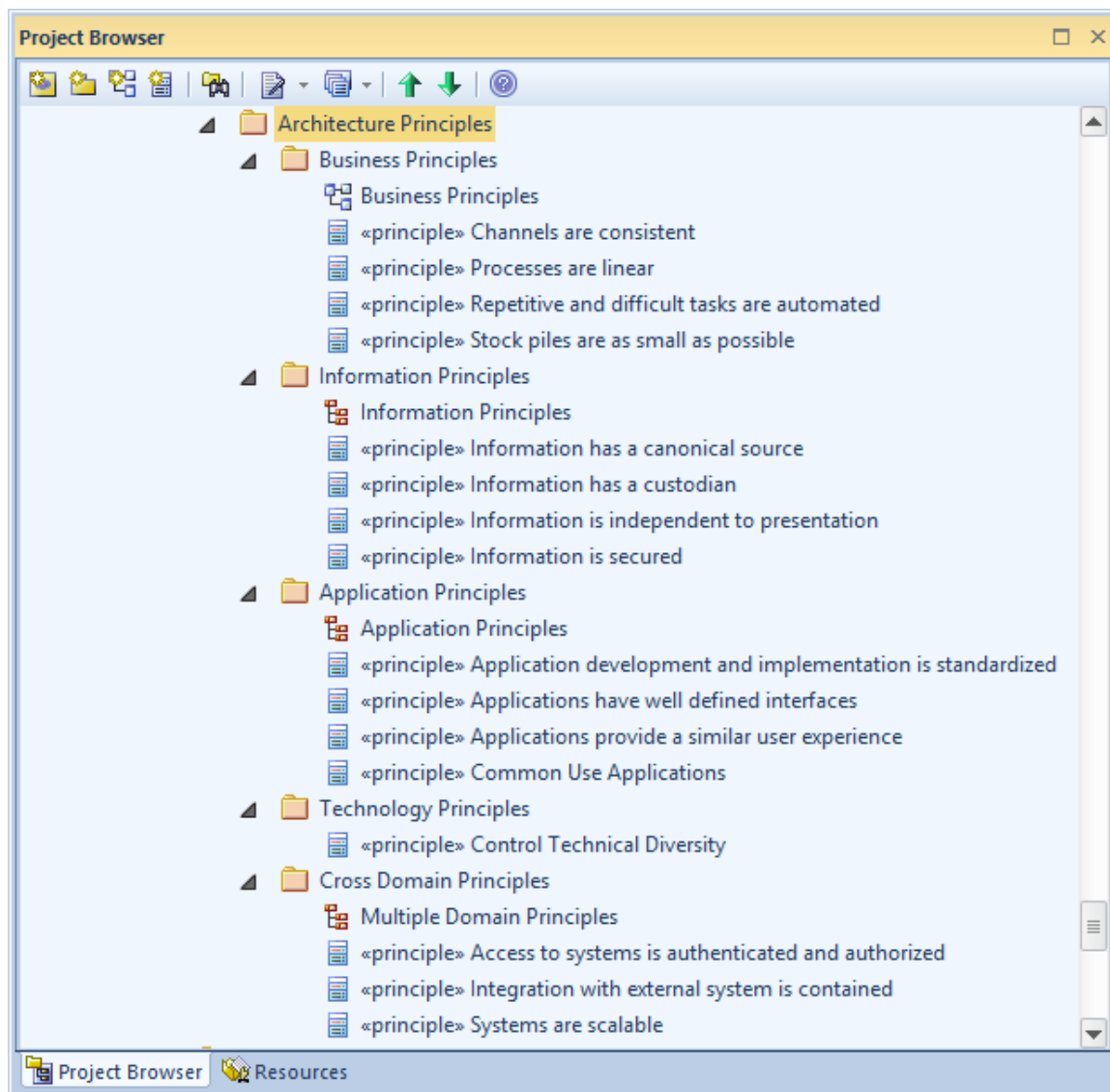


# Architecture Principles

Architecture Principles play a critical role in guiding the architecture work that ultimately has the responsibility of defining the Enterprise's future direction and the transitions it needs to reach that future state. The Architecture Principles are typically created at the time the architecture program is set up, and are reviewed and ratified by the Architecture Steering Committee. It is important that they are aligned with any existing Enterprise Principles, but also that they interpret these in a way that makes them meaningful and applicable at the architecture level. It is common to define a set of principles for each of the architecture domains, with a common group that spans more than one domain. These groups of principles typically exist:

- Business Principles
- Information (Data) Principles
- Application Principles
- Technology Principles
- Security Principles
- Cross Domain Principles

The principles must be applied in the context of the architecture and the implementation initiatives that consume that architecture. It is best practice for the domain architects to work with the implementation teams to carefully define which principles apply to a particular initiative and how they should be interpreted and applied.



## Dispensations

There will be situations where the application of one or more principles would be detrimental to the desired outcomes. In these situations the chief architect can issue a dispensation exempting the implementation team from having to comply with the principle. These dispensations

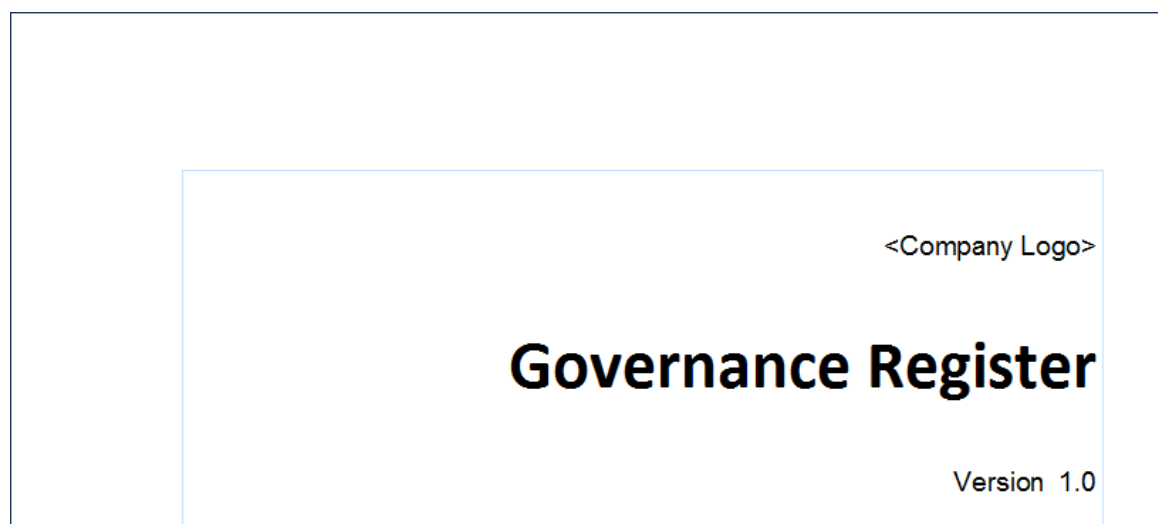
should be recorded in the Governance Register.

# Managing an Enterprise Architecture

The management of an Enterprise Architecture is critical to the overall success of the business and unless the architecture is managed through from its inception to its implementation and beyond it will not deliver the value promised to the business.

# Architecture Governance

The governance of the architectures is critical for the success of the program and the architectures it creates. Regardless of how perfect an architecture is, without the assurance that it has been implemented correctly the vision can fail to be realized, the promise made to the stakeholders will be empty and the business value will never eventuate.



# Governance Process

The architectural artifacts are a critical enterprise asset and should be managed accordingly. They resemble the plans of a building or the charts that guide a ship on its course. The Governance process can be defined in Enterprise Architect using UML Activity diagrams, and the various bodies can be defined and the members described. The Governance Register can also be created and maintained inside the Repository. The most important bodies in the process are:

- Architecture Steering Committee
- Architecture Review Board
- Chief Architect
- Architecture Team
- Governance Register

## Architecture Steering Committee

The Architecture Steering Committee is the legislature that governs the entire Enterprise Architecture program and effort. It will typically report to the Chief Information or Technical Officer, or directly to the Board of Directors. The Steering Committee provides high level guidance and is responsible for managing the program and the enforcement of compliance, the approval of principles and resolving disputes, ambiguities and conflicts.

## Architecture Review Board

The Architecture Review Board performs the primary function of reviewing and assessing architectures, and creating and maintaining standards, and references.

Enterprise Architecture can be used during the review process to present and assess the architectures, standards and references and to record the progress and results of the review.

## Governance Register

The Governance Register records the activities and decisions of the Architecture Steering Committee and Architecture Review Board, and important milestones. This includes the acceptance of Principles, Standards, References, Architectures, dispensations for the application of principles and more. It is a living register that is vital as a ledger or diary of the architecture program.

# Architecture Steering Committee

The Architecture Steering Committee is the legislature that governs the entire Enterprise Architecture program and effort. It will typically report to the Chief Information or Technical Officer, or directly to the Board of Directors. The Steering Committee is responsible for managing the program and the enforcement of compliance, the approval of principles and resolving disputes, ambiguities and conflicts. The Steering Committee provides the high level guidance whereas the Architecture Review Board provides expertise to assess the architectures and their compliance and reports this information to the board.

Enterprise Architect has a variety of facilities that can be used to assist with the function of the Steering Committee. The board members might only require read access to the repository and often will rely upon publications or reports created from the repository.

## Enterprise Architecture Charter

The Enterprise Architecture Charter, along with the principles, resembles the constitution that defines the purpose of the program and its relationship to other organizational bodies such as planning, program and project management offices, and implementation teams. It describes how Enterprise Architecture will be conducted to ensure the enterprise's vision has the best chance of success.



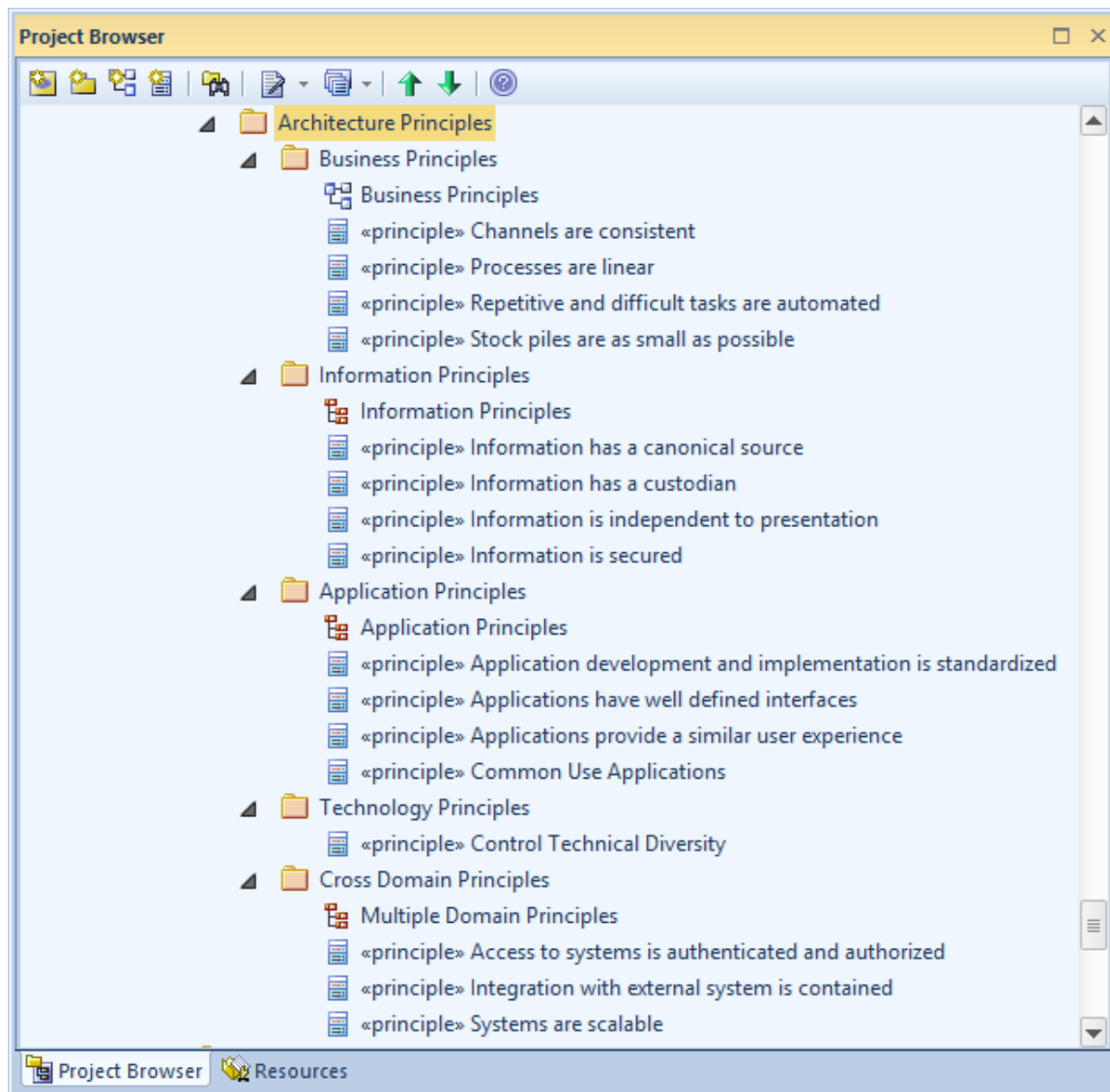
The Charter can be stored inside the Repository using a Document Artifact or, alternatively, if the document needs to reside outside the repository an Artifact can be used that references the external file or Web page using a hyperlink.

## Principles

The principles are a set of guides that prescribe the way an enterprise translates business strategy into architectures and the way those architectures are used and implemented. Many organizations define the principles and then fail to ensure that they are complied with and often they are included in documentation but are not actively used to guide and constrain the architectures and implementations. Critical to the benefit that they provide rests in the application of the principles in a context that unambiguously demonstrates how the principle applies to a given architecture. Only a subset of the principles will apply to any given context and it is the architects responsibility to ensure that the principle is explained and exemplified in the context and level of the prescribed architecture.

Enterprise Architect can be used to model principles and their application using the concept of Classifiers and Instances. The principles are defined as Unified Modeling Language (UML) Classes and Instances of the principles can be used to guide individual architectures and implementation teams describing how the principles are applied in the context. This a powerful technique because

any number of Instances of principles can be created and linked back to the parent principle.



## Governance Register

The Governance Register records the activities and decisions of the Architecture Steering Committee and Architecture Review Board, and important milestones. This

includes the acceptance of Principles and the Architecture Process, and the dates and minutes of meetings.

# Architecture Review Board

The Architecture Review Board performs the primary function of reviewing and assessing architectures, and creating and maintaining standards and references. The Board consists of individuals who are experts in their field; typically these will be the domain architects and other technical leaders invited to give opinions where required. The Board should meet regularly to assess the architectures and the implementations, most commonly once a month or more regularly when required. They have an important role at the time the program is set up, to select and create the Standards and References; these are maintained and reviewed on an ongoing basis. Thus the course of Information Technology is set by the Architecture Steering Committee and the Architecture Review Board is there to ensure that the IT remains on that chosen course.

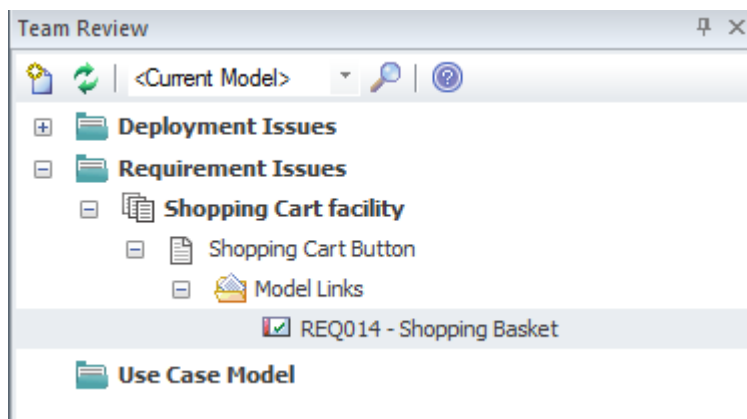
Enterprise Architect has a number of useful tools to assist with the set up and operation of the Board. These include: Artifacts and Document Artifacts to maintain the Standards and References and the Governance Register; a **Team Library** Facility for conducting reviews; informal **Discussions** to create commentary on architectures and their constituents, and more.

## Architecture Reviews

Enterprise Architect has a number of useful tools that can be

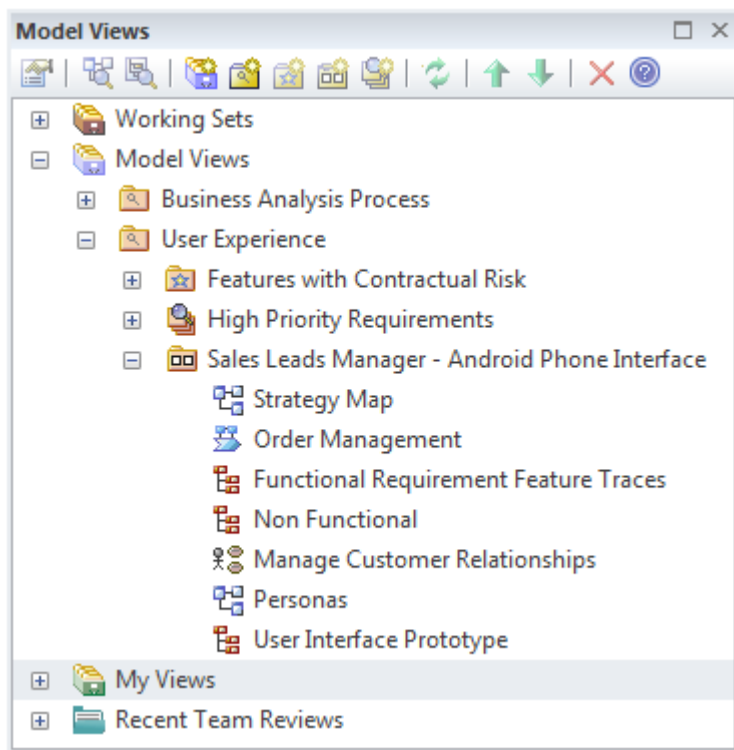
used for conducting reviews including the **Library window** facility that can be used for planning and conducting Architecture reviews.

Learn More: [Library Window](#)



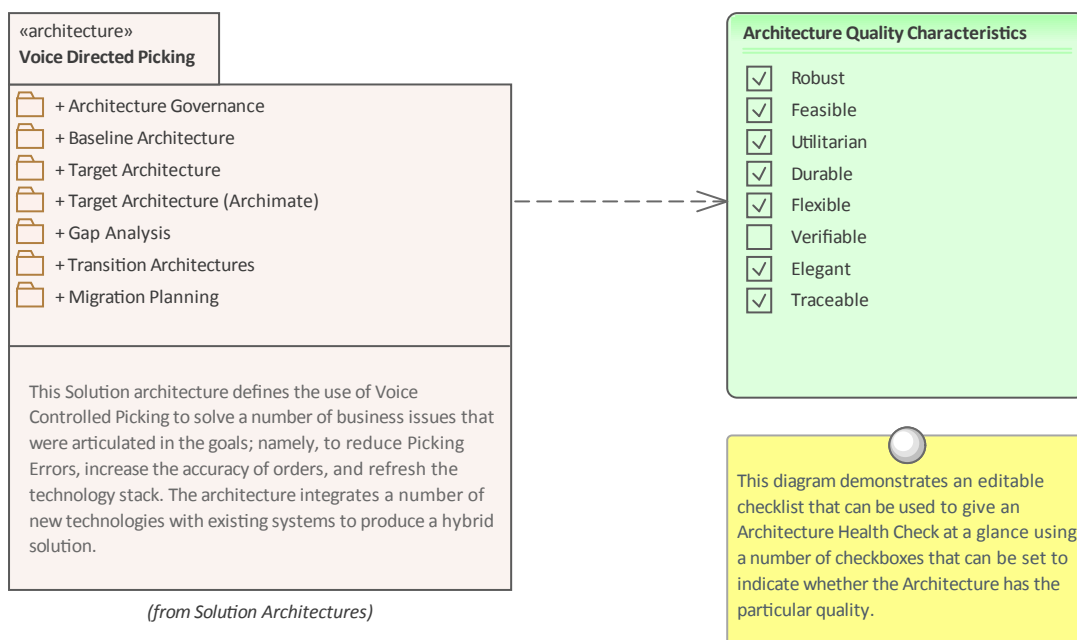
The **Model Views** tool has a slideshow facility that can be used to orchestrate a walk-through of the architectures showing the diagrams that are of interest to the review process. Model Links can be created to any a review that allow a reviewer to link a variety of elements and diagrams to the review.

Learn More: [Model Views](#)



Checklists can be used as part of a review process to ensure that an Architecture, Standard or Reference complies with prescribed criteria for quality and best practice. The list of items can be configured to suit an organization and any number of checklists can be created and applied.

Learn More: [Requirements Checklist](#)

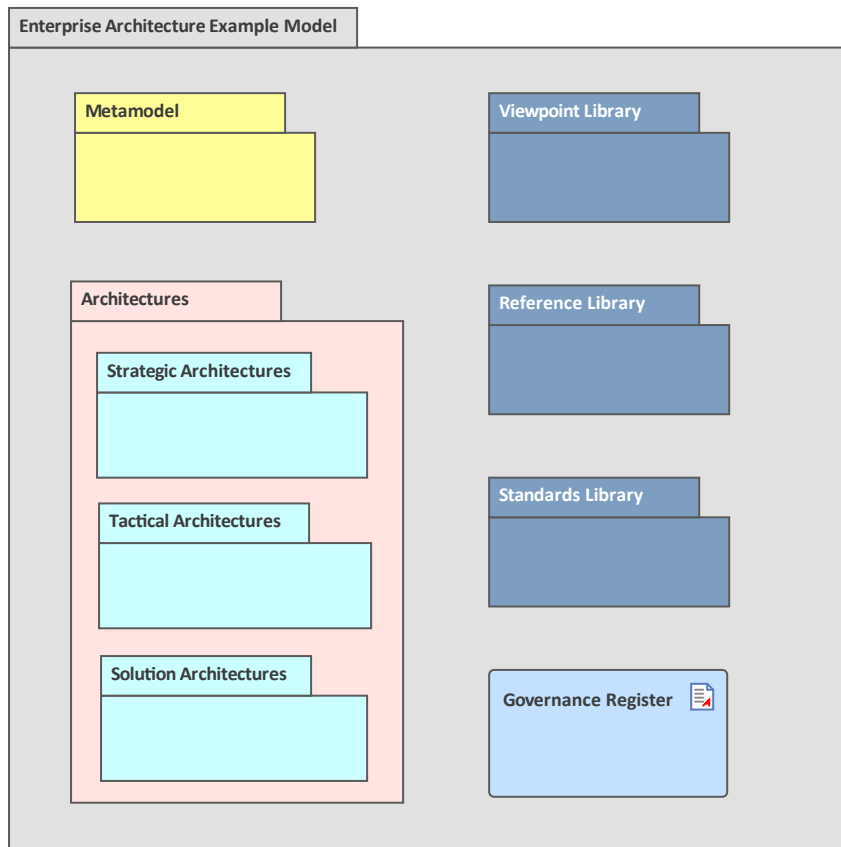


## **Governance Register**

The Governance Register records the activities and decisions of the Architecture Review Board, and important milestones. This includes the acceptance of Principles and the Architecture Process, and the dates and minutes of meetings.

## **Standards and References**

Standards and references are typically developed at the time the architecture program is set up, but are often added to as work on architecture is undertaken. For example, new government or industry geospatial standards could be developed and disseminated after the initial standard library had been created.

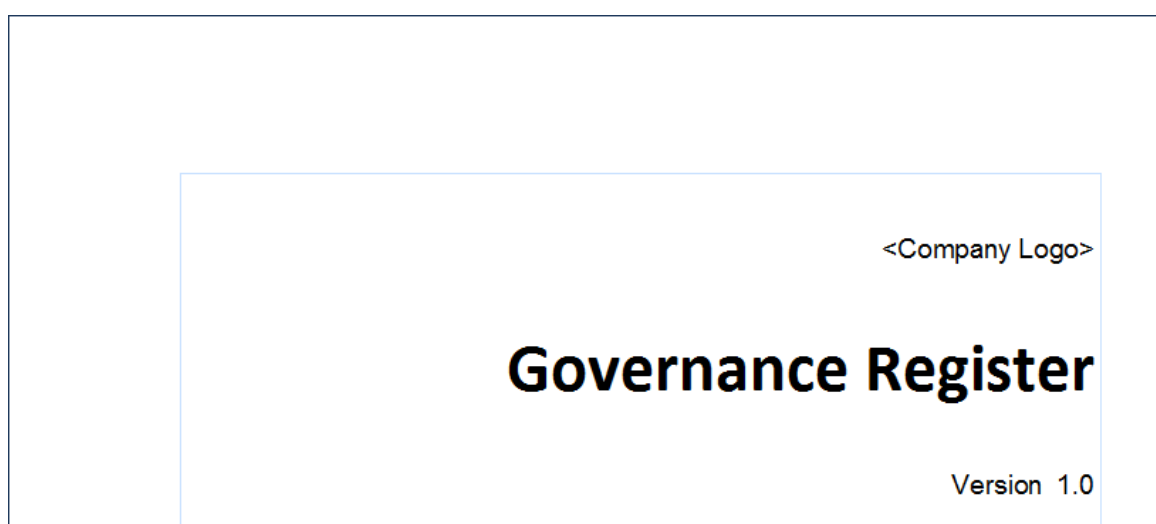




# Governance Register

The Governance Register records the activities and decisions of the Architecture Steering Committee and Architecture Review Board, and important milestones. This includes the acceptance of Principles, Standards, References, Architectures, dispensations for the application of principles, and more. It is a living register that is a vital ledger or diary of the architecture program. It will become indispensable when formal or legal investigations are conducted and human, business or technical issues are raised that concern the architectures or the processes by which they were devised and implemented.

Enterprise Architect can be used to store the Governance Register directly inside the repository using a Document Artifact or, if for some compliance reason it must be stored outside the repository, it can be linked to from an Artifact element within the repository.



## Internal Register

The register can be set up in Enterprise Architect using a Document Artifact, which is an RTF document inside the repository. This has the advantage of being close to the elements of the Architecture, and links can be included to various parts of the Architecture. If security has been enabled, the element Register can be locked for update by the Chief Enterprise Architect, who should have the responsibility of maintaining it. The document can conveniently be exported to an external file if required.

Learn More: [Document Artifact](#)

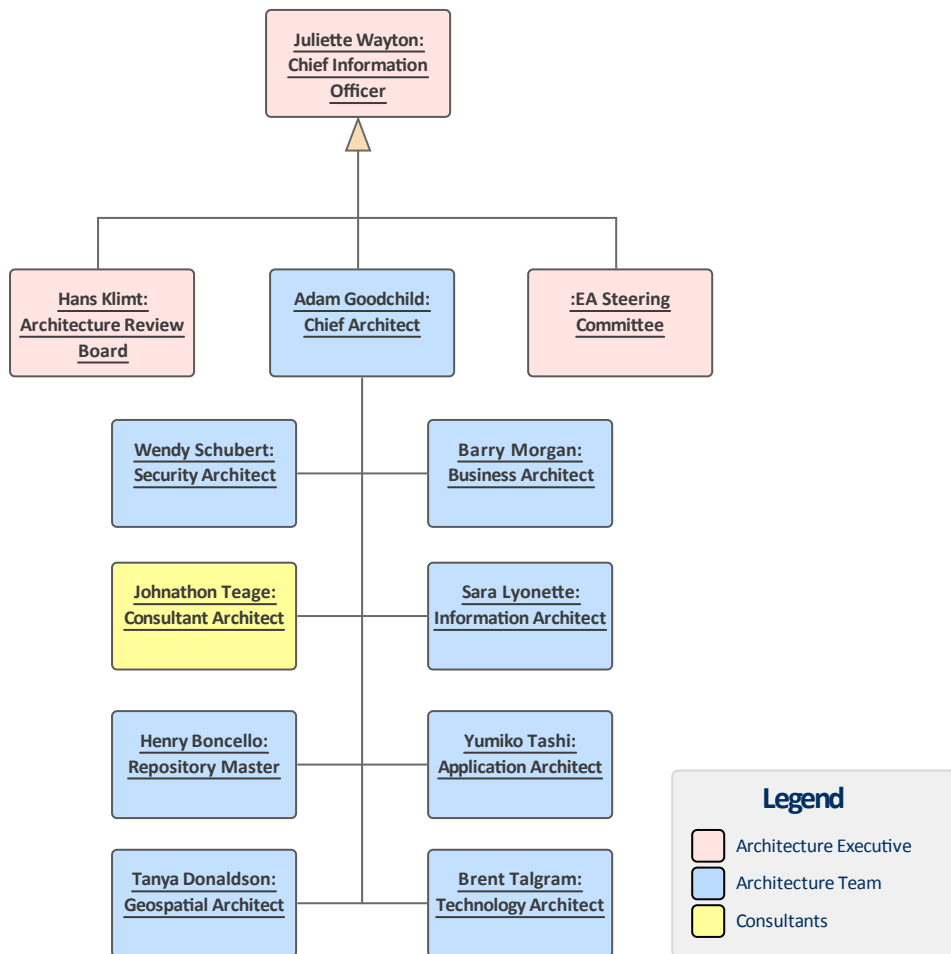
## External Register

There will be occasions when a register already exists or there is a corporate requirement that the register is stored in another tool such as a policy register or Intranet system. In these cases an Artifact element can be used as a placeholder to the external document or Web Page and a reference to it stored as a file or URL reference. This allows the Register to be visible and launched from inside the repository.

Learn More: [Artifact](#)

# Developing an Enterprise Architecture

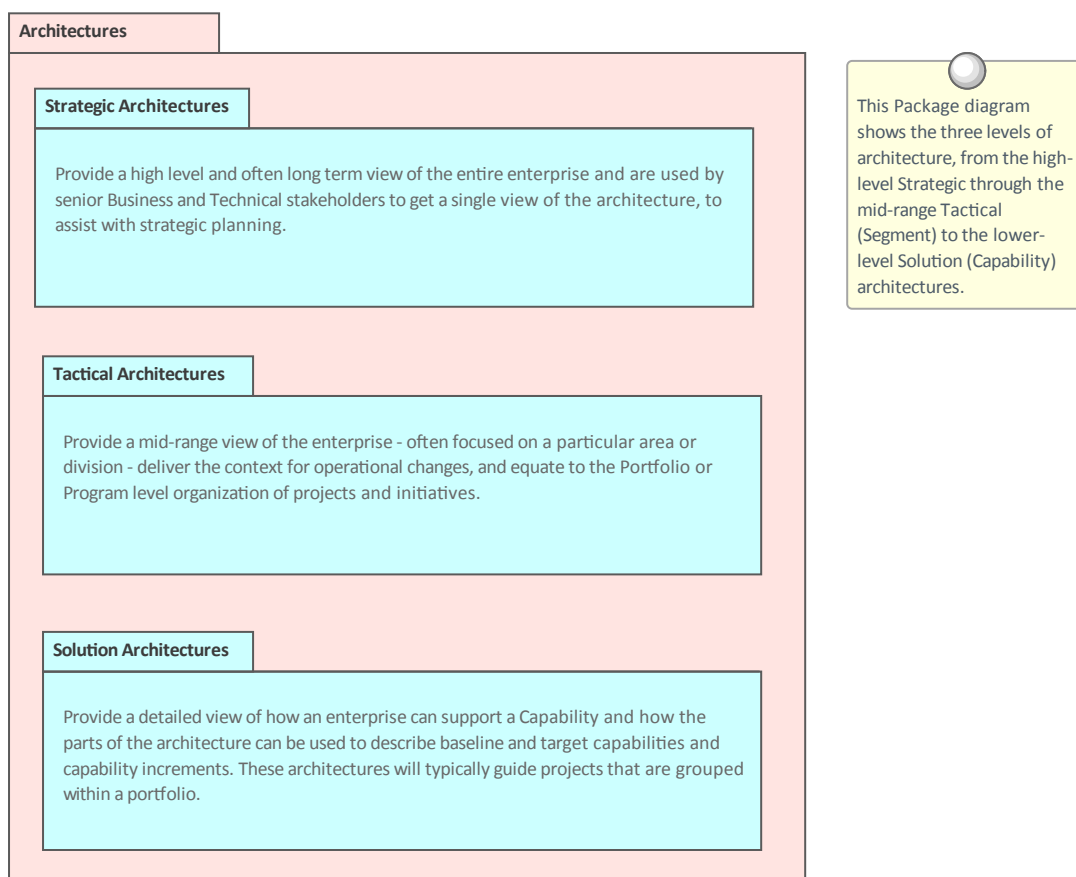
The development of an Enterprise Architecture is a complex and time consuming endeavor involving a multi-disciplined team of architects. Ultimately the architectures must deliver value to the business and this can only be achieved with the engagement of the stakeholders. Without a comprehensive stakeholder engagement and communication plan the architectures will inevitably fail to deliver the envisioned business benefit. The Enterprise Architecture should be a cohesive and well organized product that provides meaning for all stakeholders even though it comprises a number of interlocking and quite separately authored architectural contents. In this way the chief architect needs to act in an editorial role, ensuring that the Business, Information, Application, Technology, Security, Geospatial and Social architectures are merged into a coherent whole. The Architecture must act as a guide for the implementation teams, as source material for decision making and as a description of a system either before or after it is built. The Architecture Process must be well defined, repeatable and flexible to deal with the great variety of problems and opportunities that will be presented to the program. Fundamental to the execution of the process and the resulting architectures is the Architecture Team.



# Architectures

Architectures are the key organizing mechanism for architectural content; they are the design or solution statement to a proposed problem or opportunity, or the documentation of an existing system. The architectures can be defined at a number of different levels, including:

- Strategic - *Long term in the range of 3 - 5 years*
- Tactical - *Mid term in the range of 1 - 2 years*
- Solution - *Short term in the range of 3 - 12 Months*



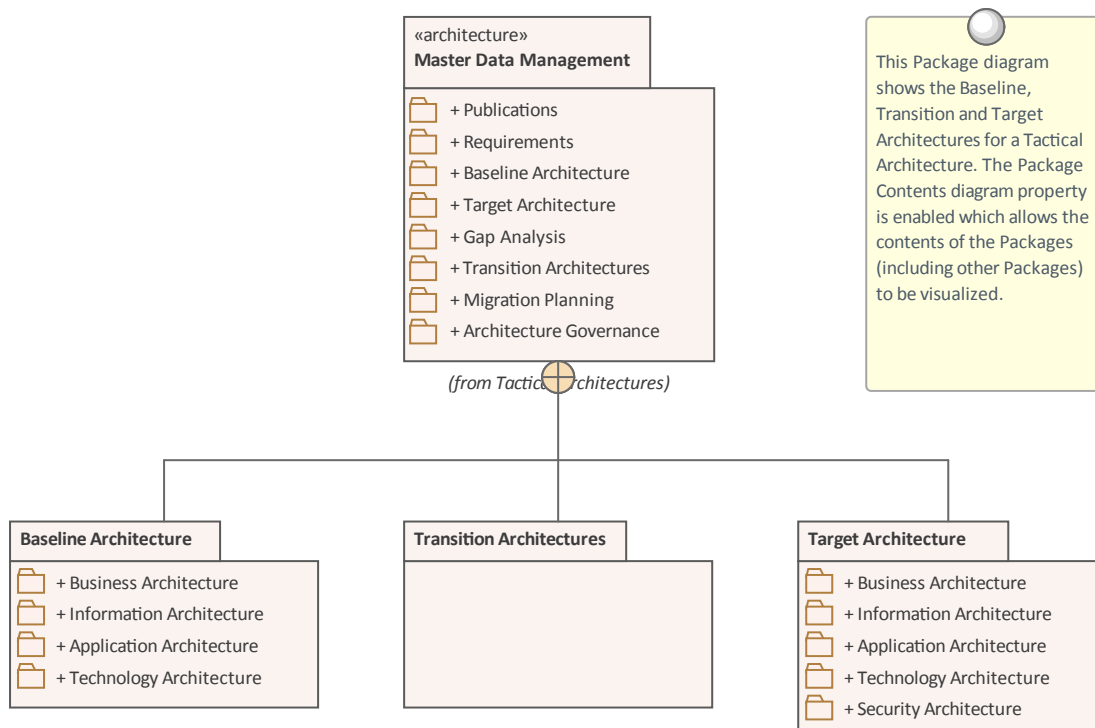
They also span a number of disciplines that are all

incorporated into a single cohesive view of the system design. It is not possible for a single person or a single discipline to understand and to articulate the Architecture at all levels. So a Business Architect is required to interpret the Strategic Plan, an Information Architect would categorize the enterprise's data, an Application Architect would articulate the interfaces between Components, and a Technology Architect is required to define the Servers and Devices that ultimately do the work. Their work results in four core domain architectures:

- Business Architecture
- Information Architecture
- Application Architecture
- Technology Architecture

Most frameworks describe analogous or similar subsets of an Enterprise Architecture, as the division is based largely on organizational units performing work in these areas.

Baseline, Transition and Target Architectures can also be defined.



# Business Architecture

A well articulated Business Architecture is the cornerstone for a successful outcome of the overall Enterprise Architecture. It defines the business drivers, the business strategy, operational models, goals and objectives that the organization must have to achieve transition in a potentially competitive and disruptive business environment. Architects working in the other architecture disciplines must understand the Business Architecture as the basis for their own architecture descriptions and as a guide to the business outcomes that must be achieved.

The Business Architecture will typically consist of a description of the baseline and target architectures with a series of transitions defined that can be executed and that would be described on Roadmap diagrams.

Enterprise Architect has a wide range of features that can be used to model the Business Architecture and to create powerful and compelling visualizations that will help the executive level stakeholders and line managers to understand how their strategies are being addressed by the architectures. When an architecture program is set up or a business architecture is initiated it will typically be found that a number of important architectural materials will be available, such as mission and vision statements, strategic plans, business drivers and goals and - more often than not - business processes and capabilities will be at least started even if not complete. Enterprise Architect has a convenient



facility to import the content of spreadsheets and word processor tables directly into the repository, saving time and effort.

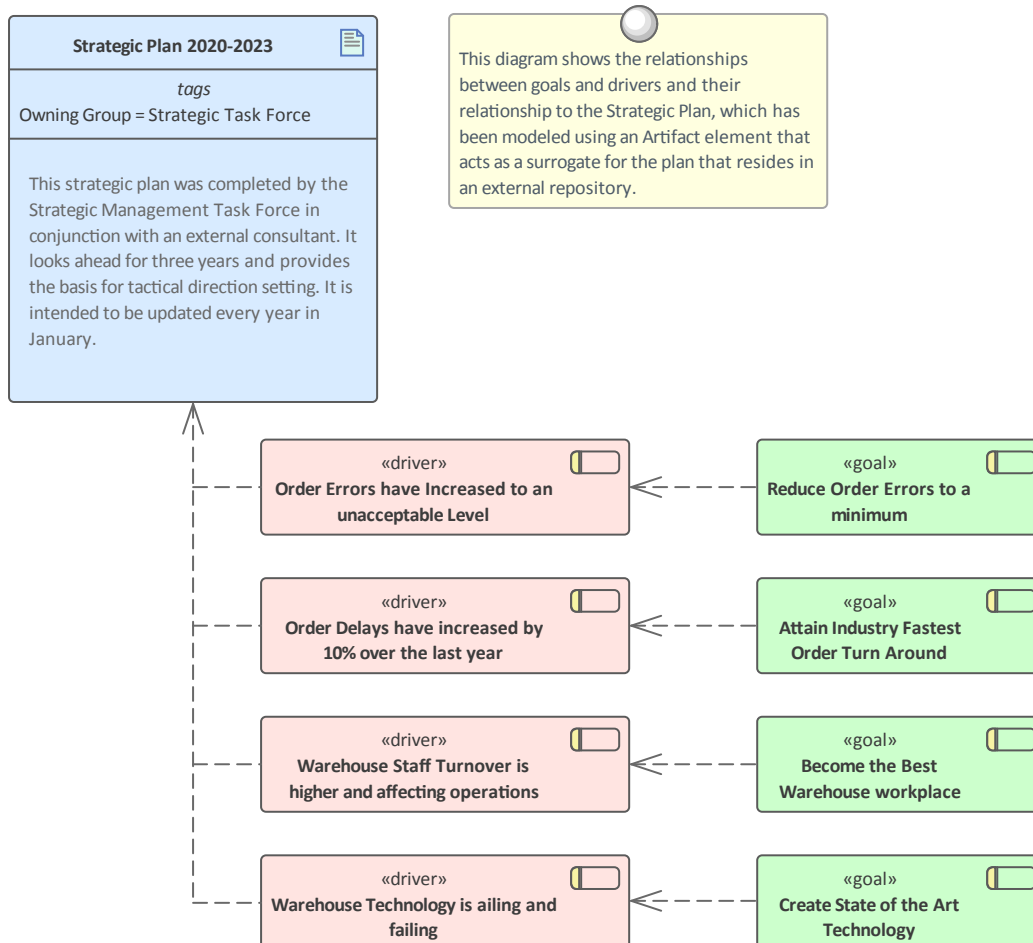
# Strategic Plans

The Strategic Plan is a fertile source of information that will provide the strategic business context for the Enterprise Architectures. It is critical that the strategic architectures are aligned to the content in this document. The strategic plan will typically include business material that might not be relevant to the Enterprise Architecture Program, and this can be omitted; however, it is common in this information age for many business initiatives, strategic goals, competitive analyses, direction settings and SWOT analyses to have some information technology basis. It is also important to remember that the strategic plan is typically updated annually, or more frequently when business environment forces influence or threaten the plan in some way.

Business Architects typically gather information from the company strategists and should be privy to the high level discussions and decisions about the future plans for the enterprise and its organizations. This information is translated into business architecture models that distill the important and relevant information and present it in a model that can be articulated with the other architecture domains.

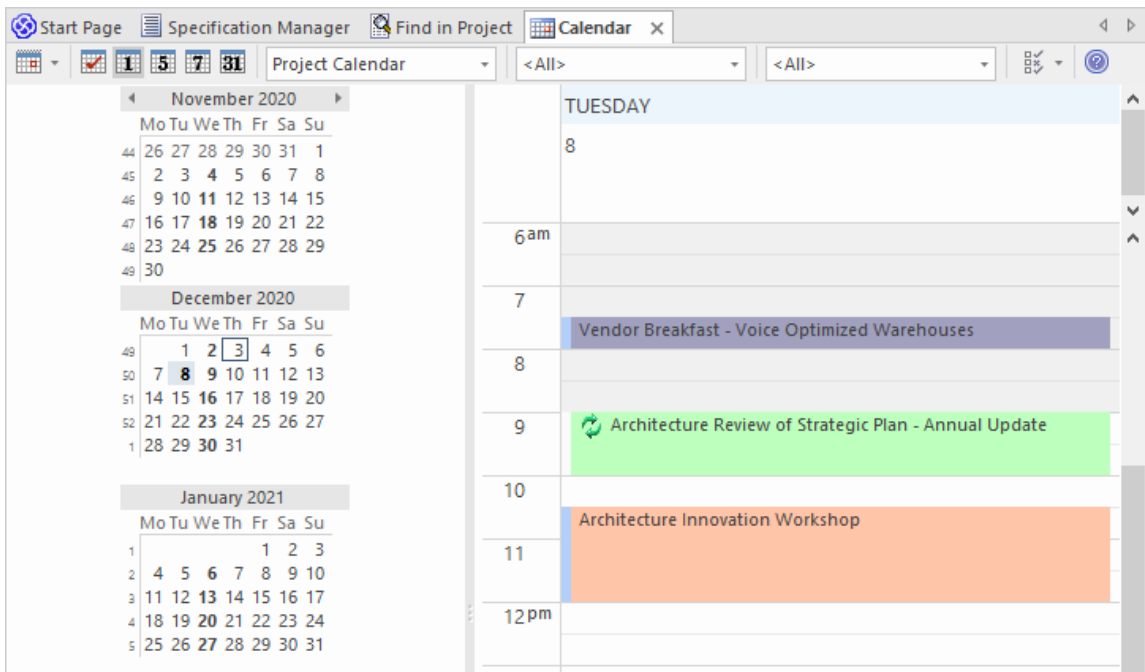
Enterprise Architect has a number of facilities that help ensure that the architecture's alignment to the strategic plans can be visualized in a format that is appropriate for the relevant executive and line of business stakeholders. The business plan will typically reside in a corporate repository, but a hyperlink can be created from an Artifact within

Enterprise Architect that will launch the Strategic Plan or other documentation. The Artifact acts as a surrogate for the external plan and then elements for Mission and Vision statements, Goals, Objectives, Drivers, Competitive Analysis and more can be created in the repository.



The diagrams, matrices and lists that describe this strategic context and the way it has been distilled to form the basis for architecture work should be provided to the executive level stakeholders in a view tailored for them. It will then act as a double check that the Business Architect and Chief Architect have interpreted the strategic content correctly. Failure to perform this step initially and whenever the strategic plan (or its equivalent) is changed could result in architectures and implementation initiatives that solve the

wrong problem. The **Calendar** can be used to schedule meetings and to act as a reminder to review the Strategic Plan at least annually, if not more frequently.



Enterprise Architect is a standards compliant modeling platform that allows rich models to be constructed that assist implementation teams, but it also provides the facilities to create compelling views of the architecture for executive level and business stakeholders who are more comfortable viewing images.

## Warehouse Modernization Tactical Architecture

This tactical architecture describes the modernization of the traditional warehouses and the regional distribution warehouses although the solutions in the two facilities will be different reflecting the level of stock that is maintained and the vast differences in the number of orders processed.

### Current Warehouse Operations



#### Manual Pickers and Forklifts

Current warehouse picking is carried out with a combination of forklifts and hand picking staff requiring workers to walk up to 12 kilometers a day.



3-5 year Strategy

### Future Warehouse Operations



#### Mobile Robotic Drive Units

Future warehouse picking in the major distribution centers will be orchestrated by a team of Mobile Robotic Drive Units that will bring the items to the packers.

# Mission and Vision

The Mission and Vision statements will typically already exist when an architecture program is set up or a Business Architecture is commenced, but there will be some cases where they do not exist, or the Mission statement exists but the Vision statement is not available. These two very simple statements have different purposes:

- A Mission statement describes the line of business the organization undertakes and what its purpose is
- The Vision statement is like a summary of the organization's goals or a synopsis of the strategic plan; it describes where the organization wants to be in the future and what its aspirations are

These two statements are useful to include in a Business Architecture as they provide a very brief and crisp definition of what the organization is and where it wants to be. The business goals should link back to the Vision statement, providing a traceability to the fundamental strategic direction of the organization.

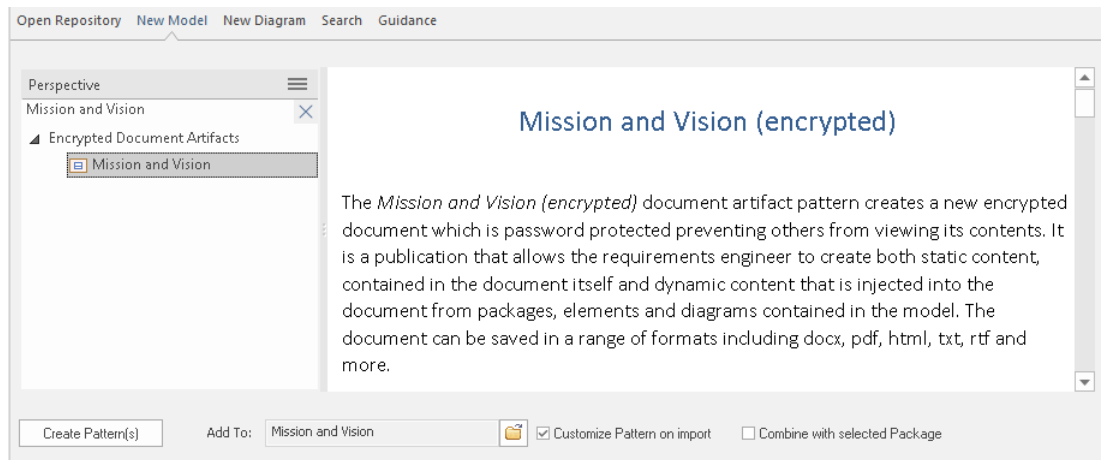
Using Enterprise Architect, there are a number of ways in which you can create Mission and Vision statements in your model, depending on how you want to define and use the statements and the objects representing them. Some of the ways are described in the remaining sections of this topic.

## As an Encrypted Document

If you are defining new Mission and Vision statements, and want to record the background to each statement, what the statements encapsulate, how they were devised and who contributed, you can generate these broader and more detailed statements as a model document from a Pattern selected from the **Model Wizard**. This document is core to the business and to developments within the business, so it is generated as an encrypted document with password protection. This safeguards the document from uncontrolled access and editing; when you want to publish it you can save it to an external file location for public access.

Follow these steps:

1. In the **Browser window**, right-click on the Package to hold the document, and select the 'Add a Model using Wizard' option. The Model Wizard displays.
2. In the 'Perspective' field, click on the icon and select 'All'.
3. In the **Filter** Bar directly below the Perspective field, type 'Mission and Vision'. The option 'Encrypted Document Artifacts' displays in the left-hand panel, expanded to show 'Mission and Vision'.
4. Click on 'Mission and Vision', and carefully read the Pattern description displayed in the right-hand panel.



5. Click on the **Create Pattern(s) button** to create the document in the selected Package. The element shows in the Browser window in this format:

 «EncryptedDocument» EncryptedDocumentTest

6. Double-click on the element to decrypt and open the document in the document editor. A prompt displays to enter the password for the document.
7. Type in the password (as indicated in the Pattern description you read in the Model Wizard). The template document opens for you to complete.

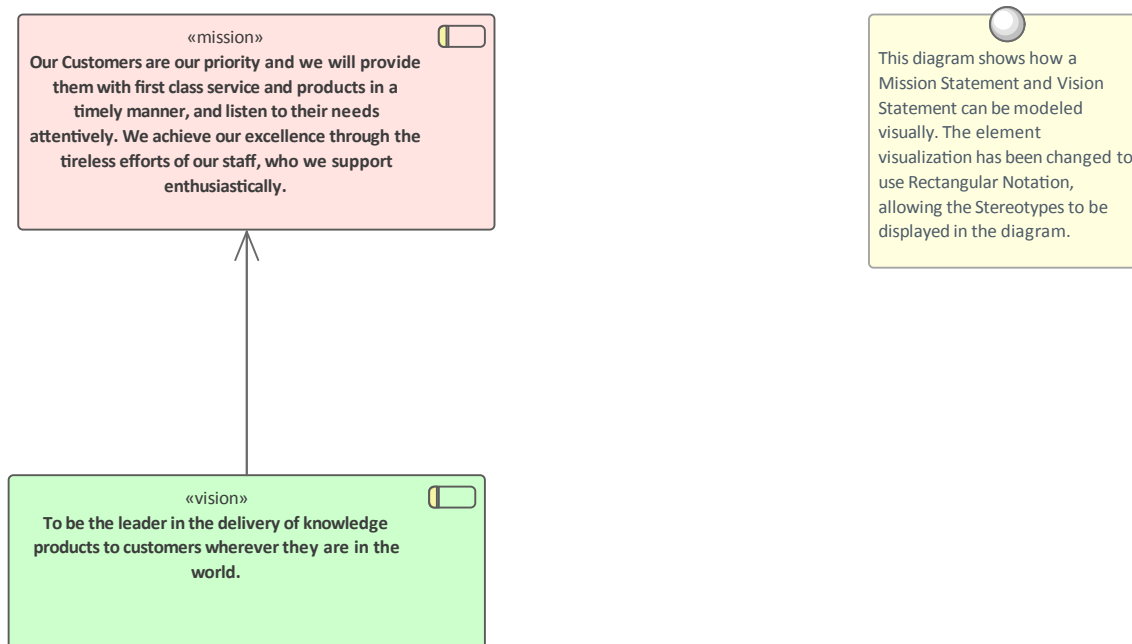
The encrypted document is also a Custom Document. The Mission and Vision template provides a number of styles and boiler-plate sections and text, but you can type in and format your own text as Static text, and also drag in objects from the Browser window, which become Dynamic text sections in the document. Each Dynamic text section has its own user-selected template or template fragment to format the information drawn from the object, and that information is refreshed from the model whenever the document is opened or whenever you specifically prompt the system to update the information.



For more information, see the *Custom Document* and *Artifact Help* topics.

## As Stereotyped Requirements

If you want to use short, direct and essential Mission and Vision statements, you can model them using the UML Extension mechanism of stereotypes on elements. The statement elements can be included in diagrams and related to other elements such as Goals and, in turn, Objectives. These diagrams are compelling for senior executives and line managers, who can see clearly that the architecture work and the subsequent implementation initiatives can be traced back to the strategic primitives.



In this illustration, a Logical (Class) diagram has been created, and two Requirement elements have been dragged onto the diagram from the Common page of the Toolbox.

Through the **Properties window**, in the 'Stereotype' field, one Requirement element has been given the stereotype 'Enterprise Architecture::mission', and the other the stereotype 'Enterprise Architecture::vision'. The name of each element is the Mission or Vision statement itself.

## As Other Types of Element

Depending on which modeling language or tool you are using, Mission statements and Vision statements can be represented by various types of element. For example, in:

- TOGAF, in the Conceptual Framework and the Business Motivation Model, a Mission or Vision can be defined in the unencrypted Linked Document of a Document Artifact
- A Strategic Map, the statements can be represented by stereotyped Class elements
- A Business Motivation Model, the statements are represented by stereotyped Requirement elements
- UPDM 2.0, the statements can be rendered as stereotyped Activity elements
- The Zachman Framework, the statements are defined by simple Artifact elements

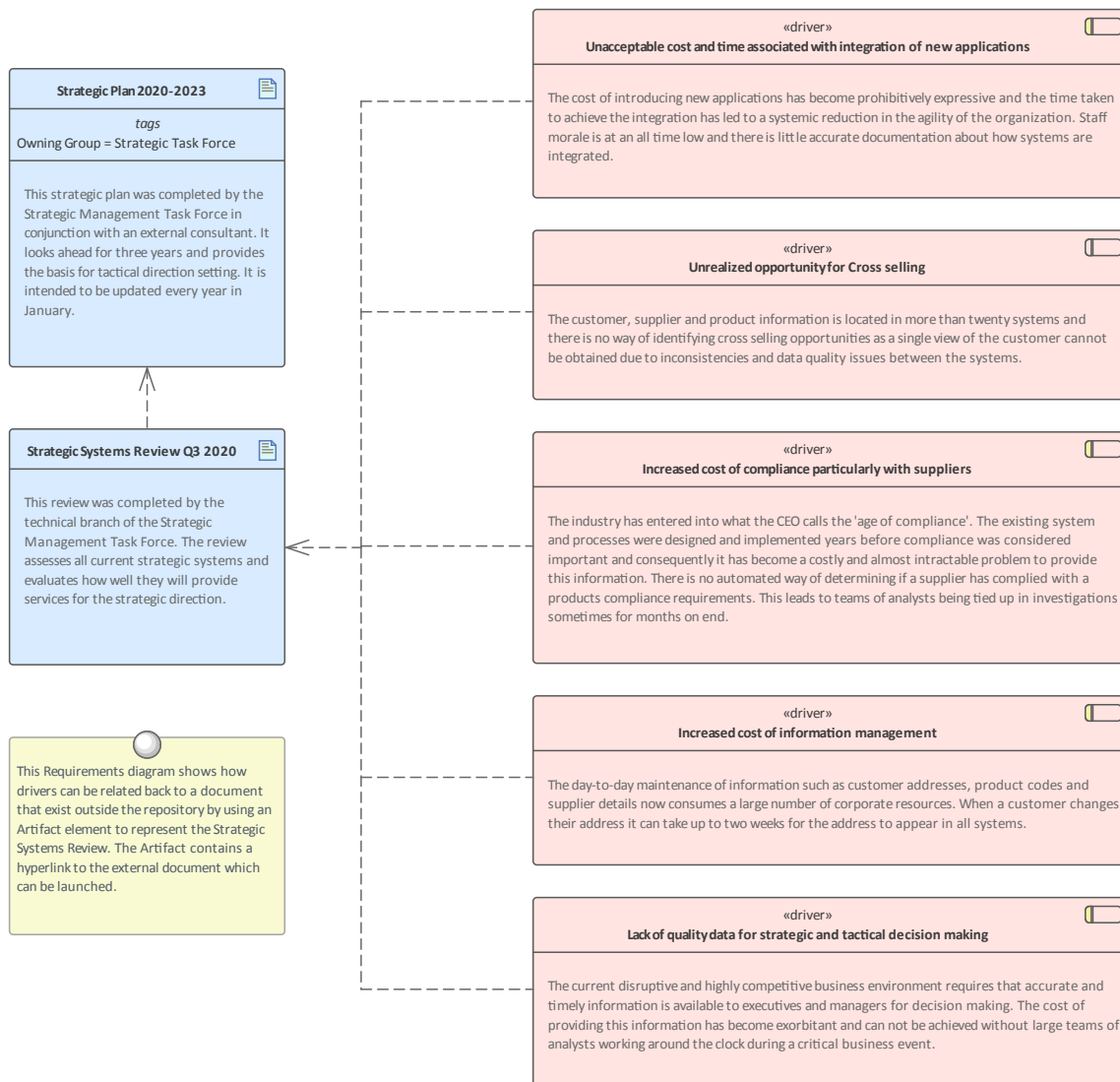
In each case, you drag the element icon from the appropriate Toolbox page and then either apply the statement as the element name or define a more detailed statement in a Linked Document on the element.



# Drivers

Drivers are the events, conditions or circumstances that motivate an enterprise to change some aspect of its goals and consequently its objectives, and potentially business activities or technical solutions. It is common for only the Goals and Objectives to be tabled, but understanding the Drivers is critical to understanding the high level needs (goals) articulated by the executive level stakeholders. Assumptions and Constraints should also be analyzed in conjunction with the Drivers to ensure the entire context for the goals is well understood. There are many different drivers, including:

- Legal or Regulatory - *for example, a new law governing Credit Card Usage*
- Cultural or Social - *for example acceptance of a product in a new market*
- Financial and Economic - *for example, reduction in profit due to changes of an exchange rate for borrowed foreign capital*
- Organizational or Cultural - *for example, high staff turn-over in a particular division or business unit*



The drivers can originate internally, such as financial or budget pressures, or externally such as regulatory obligations or forces from competitors. They can be top-down, in the form of strategic drivers such as regulatory compliance and rule changes, or bottom-up in the form of tactical drivers such as technology changes or opportunities that have been discovered by audits, performance assessment, surveys or other mechanisms.

Enterprise Architect has support for modeling both strategic and tactical drivers with the use of a stereotyped driver

element that can be linked to goals and other organizational elements to show how the organization must respond. They can also be linked through the Goal or Objectives to a stakeholder who can help validate that the Driver and its relationships to Goals, Constraints and Assumptions are correct. This should be done collaboratively and stakeholders can review the models directly in the repository or by looking at documentation that has been automatically generated from the models using built-in or custom templates.

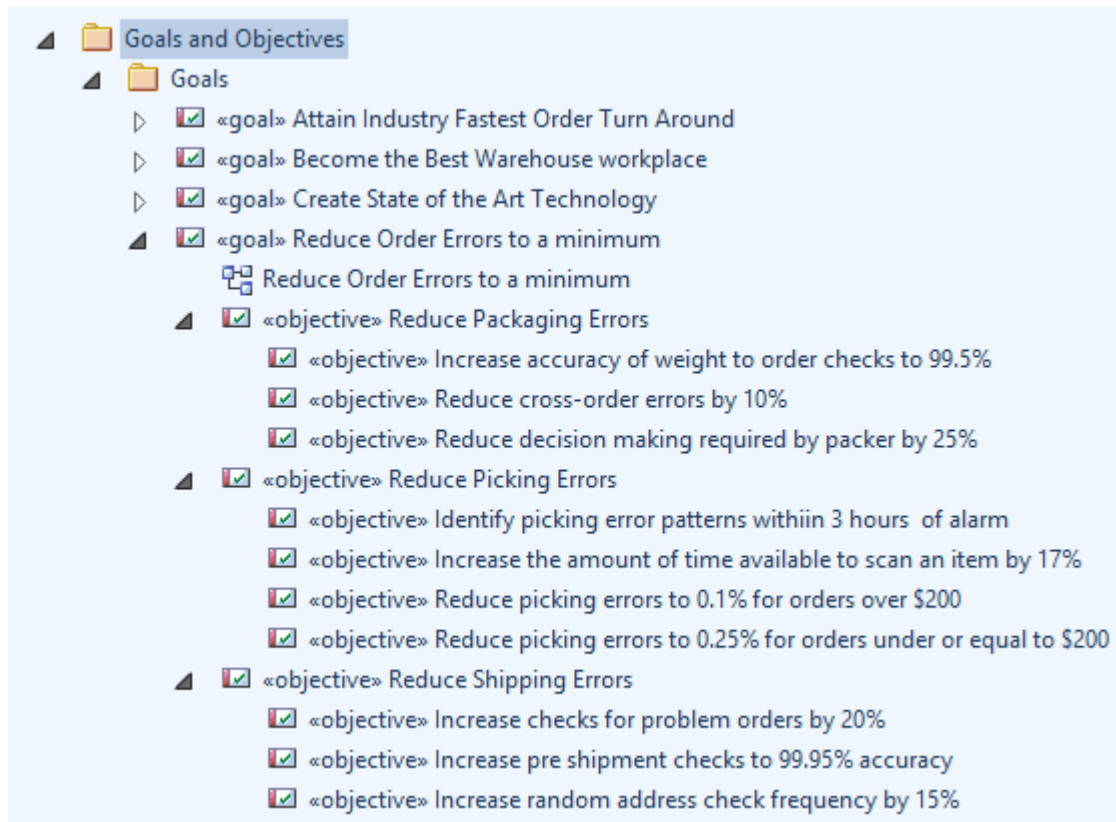
# Goals and Objectives

Goals represent the decomposition of the strategy and are aspirations that the enterprise intends to achieve. They are important because they are statements that can be used as anchor points for the architecture. Some frameworks and methods will consider these to be outside the scope of the architecture and just providing a context. Either way it is critical that they are modeled to provide a reference point for the business architecture. Once the goals are modeled it will most likely become apparent that some of them will not be able to be achieved by technology alone but will typically require pure business solutions to address them in part or in whole.

The goals will still be too high level to act as a starting point for the other architecture domains and so they need to be broken down into a series of objectives that can be stated in terms of measurable outcomes. Each goal will be an aggregate of one or more objectives and these can be related back to the parent goal.

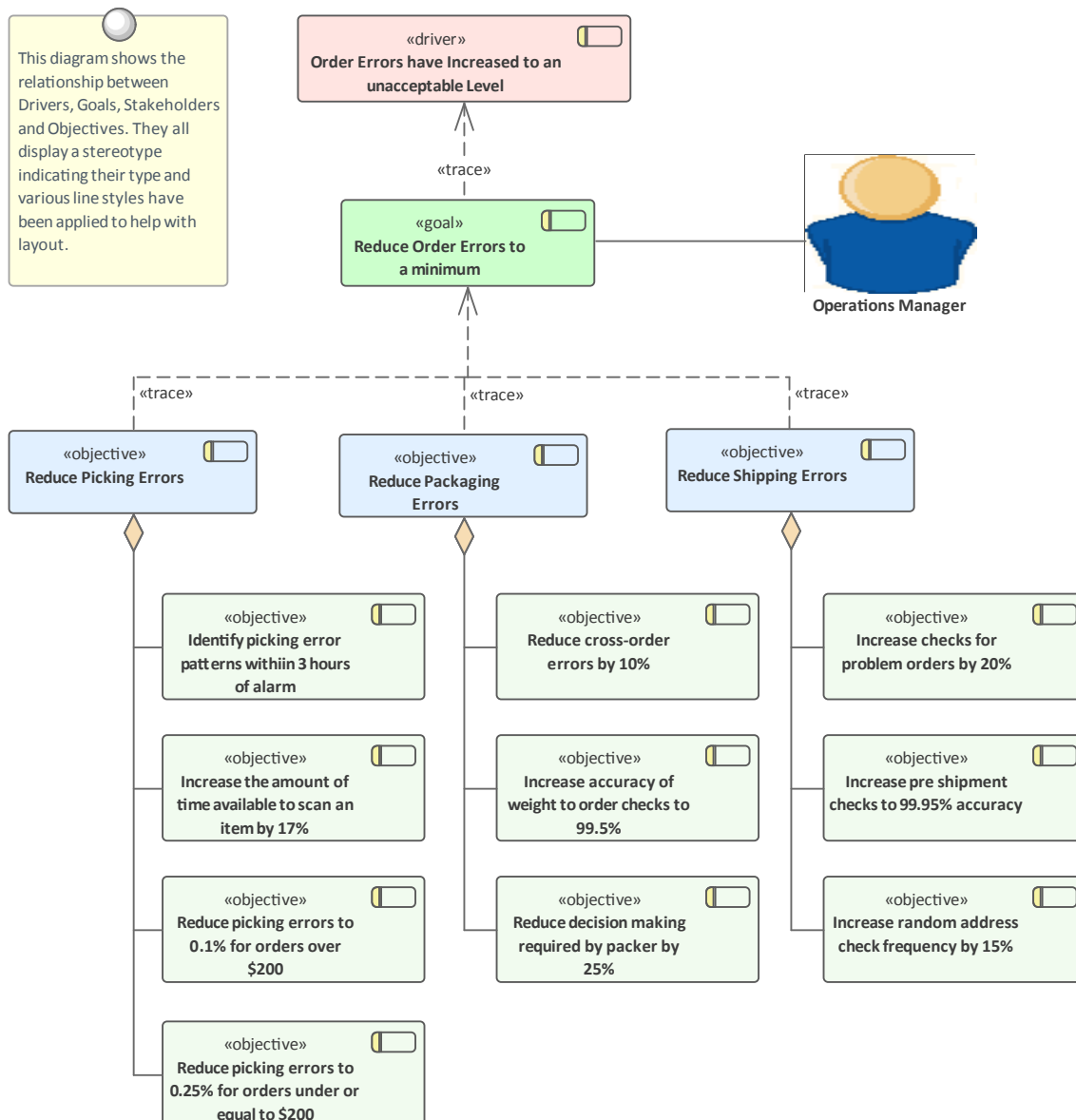
Enterprise Architect allows both the Goals and Objectives to be modeled using the UML Extension mechanism of stereotypes and these elements can be included in diagrams and lists and related to other elements such as the Vision statement and Measures. These diagrams are compelling for senior executives and line managers who can see clearly that the architecture work and the subsequent implementation initiatives can be traced back to the Objectives and Goals

and other strategic primitives.



These elements and relationships can also be visualized in a diagram which provides a welcomed view for some stakeholders particularly the executives and senior business managers.



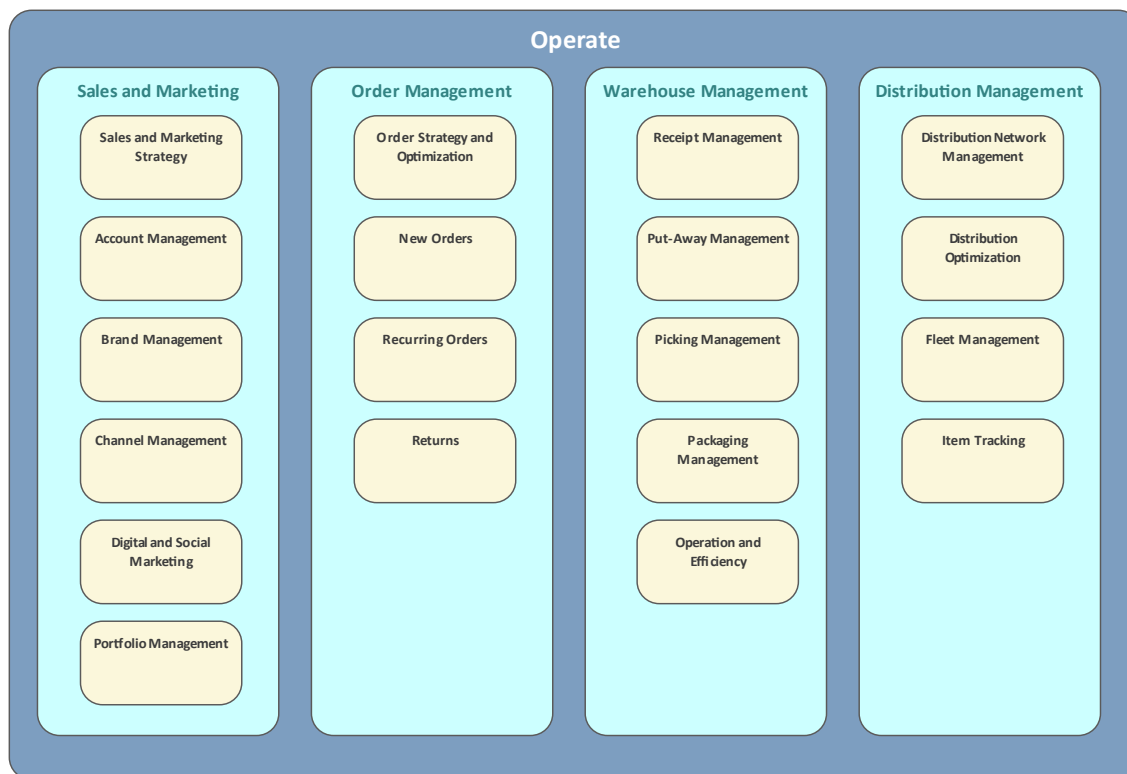


# Capabilities

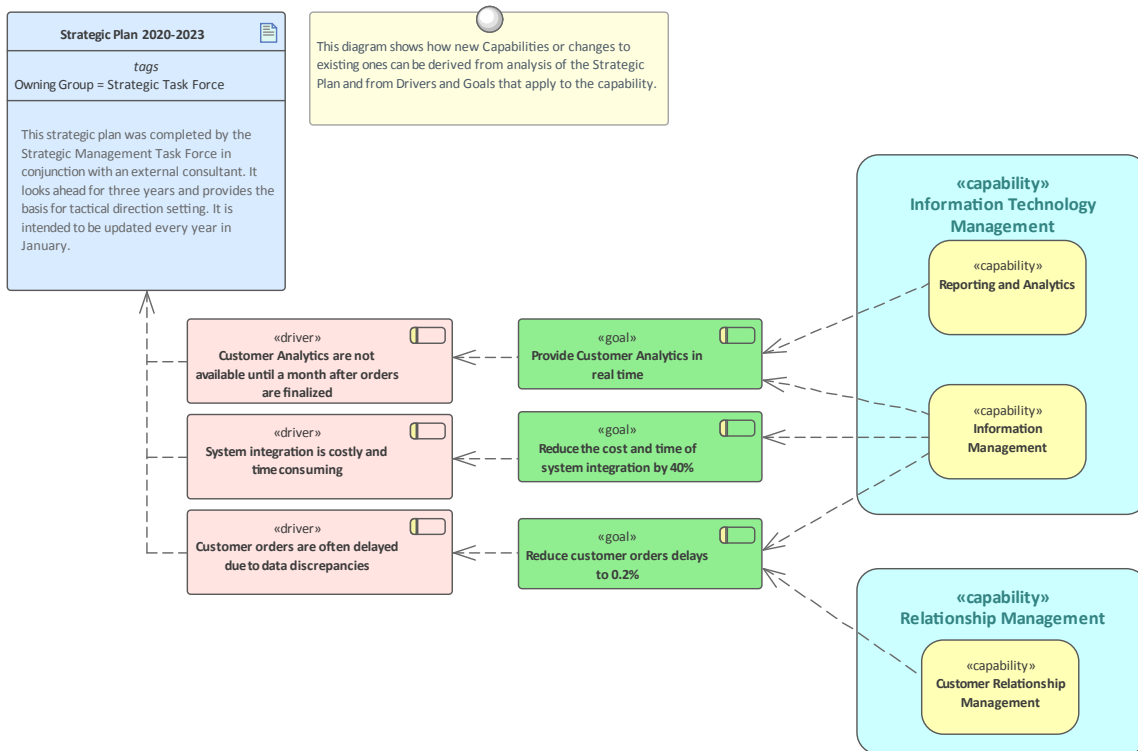
Business Capabilities are the cornerstone for the work carried out by the other architecture domains, including Information, Application and Technology Architecture. They provide a way of viewing what the business does or needs to do. They are considerably easier and less time consuming to create than business processes, and provide a direct linear relation to what the business does. One thing that makes them easier is that they can be seen as an abstraction above the business processes, as they are modeled independent of how the Capability is achieved or is going to be achieved.

Enterprise Architect allows Capabilities to be modeled using the UML extension mechanism of stereotypes. Like any text based element, the **Specification Manager** can be used when working with Capabilities and provides a compelling spreadsheet or word processor view that is appealing to many business people. Capabilities are usually created in a hierarchy that allows high level Capabilities to be decomposed into more granular elements, usually down to three levels. Capabilities can be viewed in a diagram, where the hierarchy can be visualized using nested elements that can be colored to create visual appeal and printed onto large format paper in A1 or A0 to create an eye-catching poster. The Capability stereotype can be hidden using the diagram properties, making the diagram more appealing.

## Operate Capability Hierarchy



Capabilities provide a convenient anchor point in the Business Architecture for both the Information and Application Architecture domains. The connection to the Application Architecture is usually to an application service or to a logical application that is at least one level of abstraction above the systems that actually carry out the work specified in the Capability. Capabilities also provide a connection point for the pure business processes or services that are needed to deliver the Capability.



It is often the case that new Capabilities defined by the business to achieve a strategic goal or objective are too large to be used as the basis for architectural and implementation project definitions. These high level Capabilities can be made more granular by defining a series of Capability Increments that act as stepping stones between the existing enterprise and the attainment of the new Capability. The Capability Increments can be represented on a Roadmap diagram to show the time-based sequencing of their implementation.

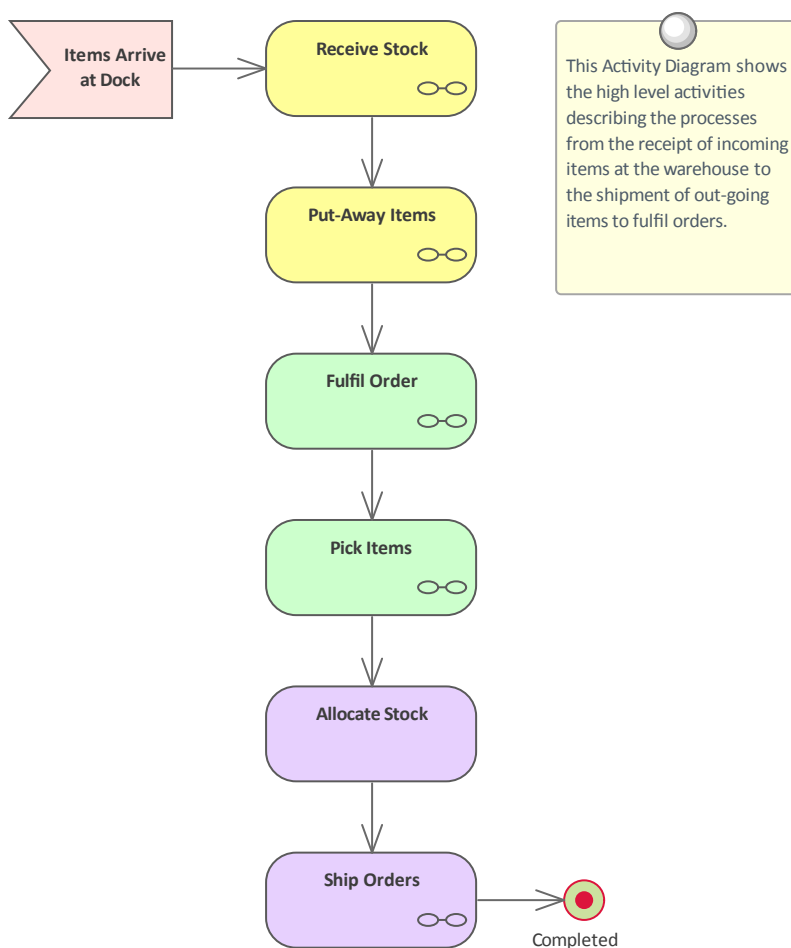
# Business Processes

Business Processes describe the Activities that are carried out by an organization to deliver a product or service. They are typically represented as a series of sequential steps, and are decomposed into hierarchies that often start at the highest level with the value chain and go down to the level just above the task level. When used with Enterprise Architecture it is important to keep these diagrams simple and descriptive. They are a useful view of how an organization goes about achieving its desired business goals and objectives. They are typically represented as a series of diagrams organized in a hierarchy reflecting the way the business sees its operation.

These provide an alternative view of what the business does, and can be compared to both Business Functions and Business Capabilities, both of which are represented by hierarchies of elements. The Business Functions typically deliver business value and are organized by required skill and organizational structure. Capabilities are more abstract and focus on what abilities are required, and as a result they tend to be less volatile. One of the issues with Businesses Process modeling is that by the time the task is complete the processes are likely to have changed. Even though it is more common to pin architectural effort to a well developed capability model, the Businesses Process Model still has its purpose and helps the architect understand how the business value is achieved and the flow of activities that creates it.

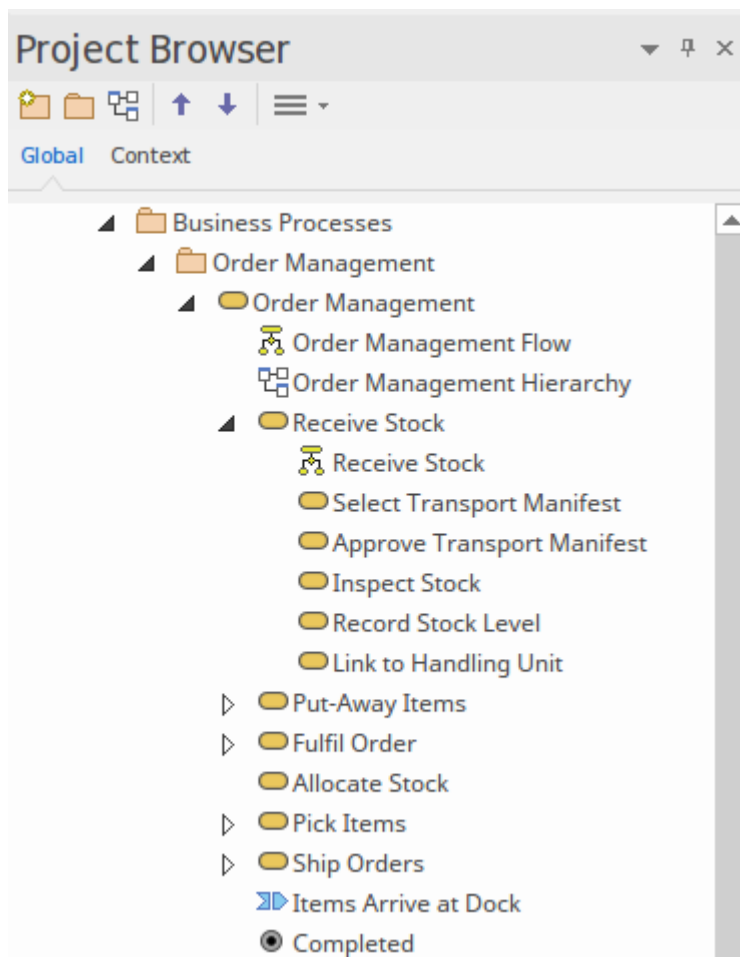
Enterprise Architect has a wide range of tools that can be used to create models of the business activities. The three main languages that can be used to model the Business Processes are:

- ArchiMate Business Layer - *Business Process*
- Unified Modeling Language - *Activity diagram*
- Business Process model and Notation - *Business Process diagram*

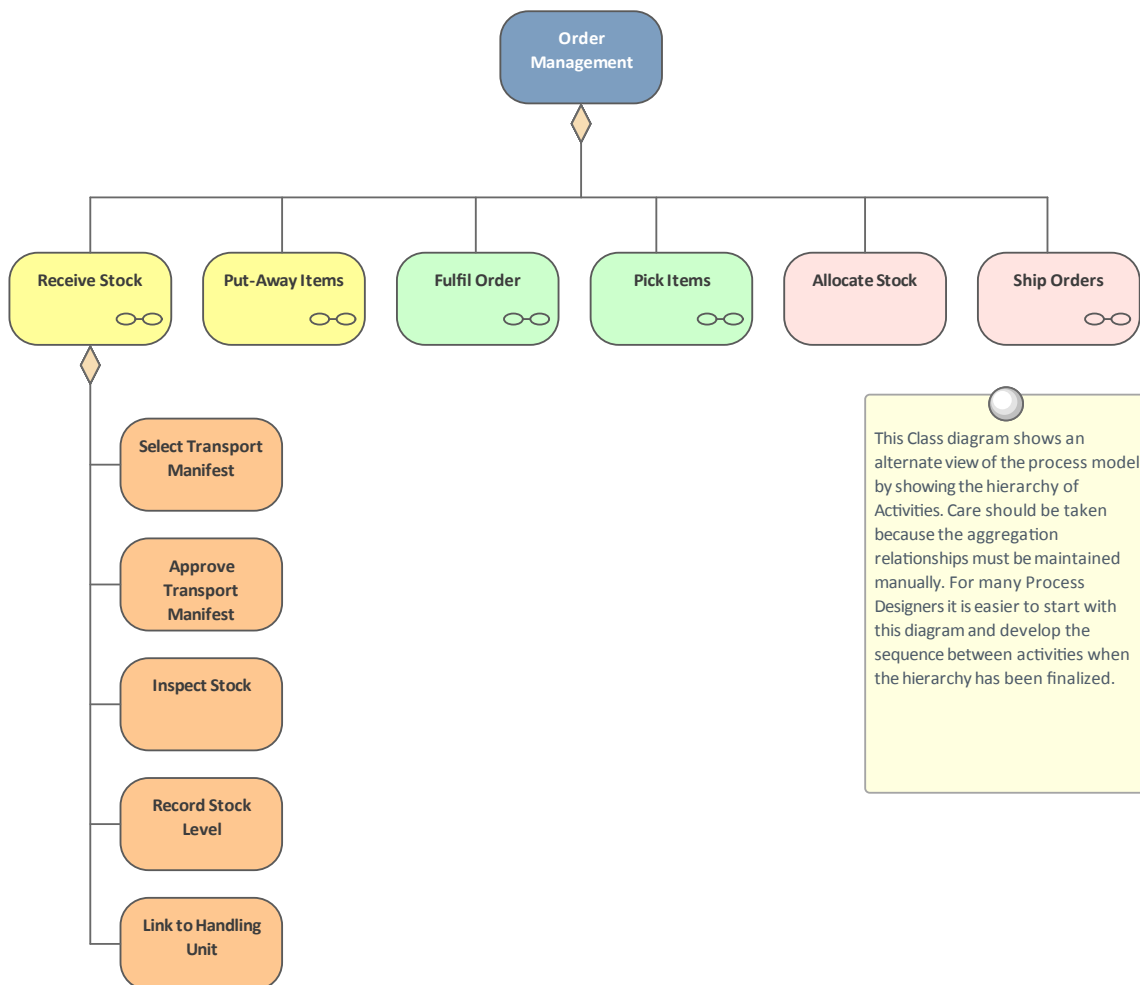


The Activity that is the behavioral unit in each of the notations can be organized into a hierarchy using the **Browser window**, which provides a way of rolling up from low level activities and a method to decompose a higher level activity into more granular Activities. This provides a

usual mechanism for viewing the hierarchy and also acts as a paradigm for communication with the appropriate stakeholders - executives will typically be interested in the Value Chain or highest level Activities, line of business managers with the next level down, and operation managers will be interested in the lower level diagrams.



An alternative way of visualizing the hierarchy of activities that make up a business process is to create a Business Process Hierarchy diagram that shows the hierarchy in a structural diagram. This provides a compelling view but care must be taken as this diagram must be updated manually if the structure of the hierarchy is changed.



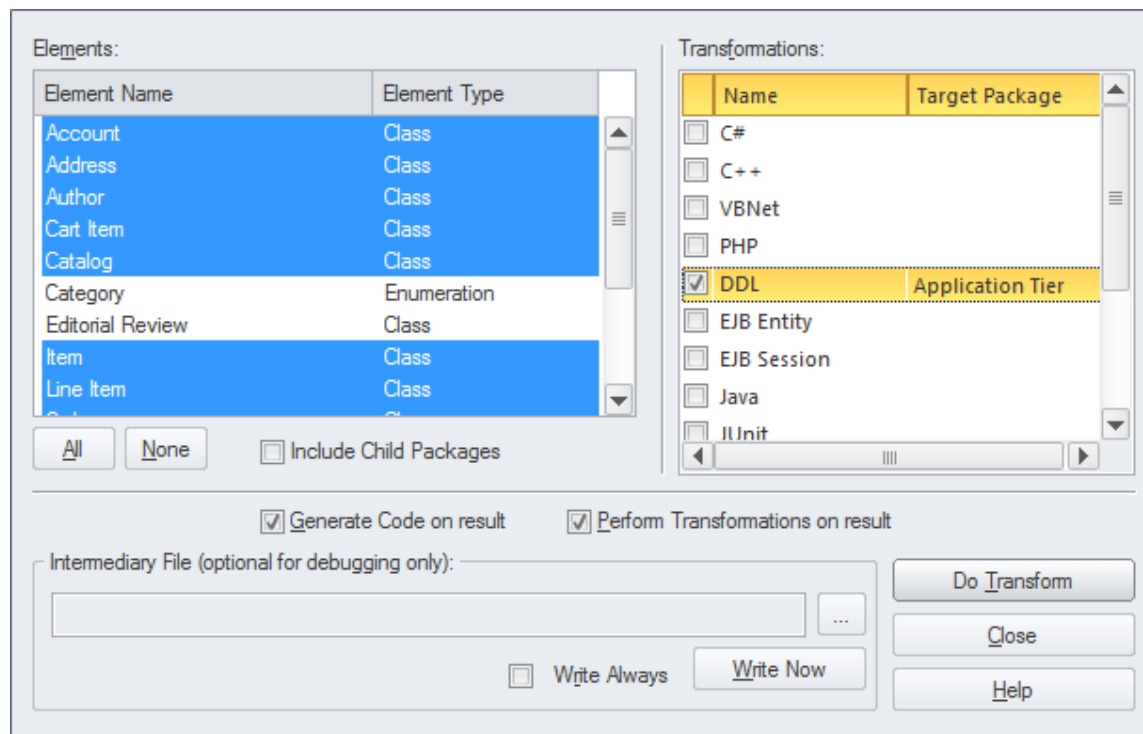


# Information Architecture

Information Architecture is key to the success of an Enterprise Architecture Program, as information is created, consumed and destroyed by the components that make up the other architectures. Understanding which business functions and processes utilize information, which applications serve as the master record, where information is created and destroyed, and which technology components store and manipulate the information, is critical to achieving the business outcomes.

The information architecture will typically consist of a description of the baseline and target architectures, with a series of transitions defined that can be executed and that would be described on Roadmap diagrams.

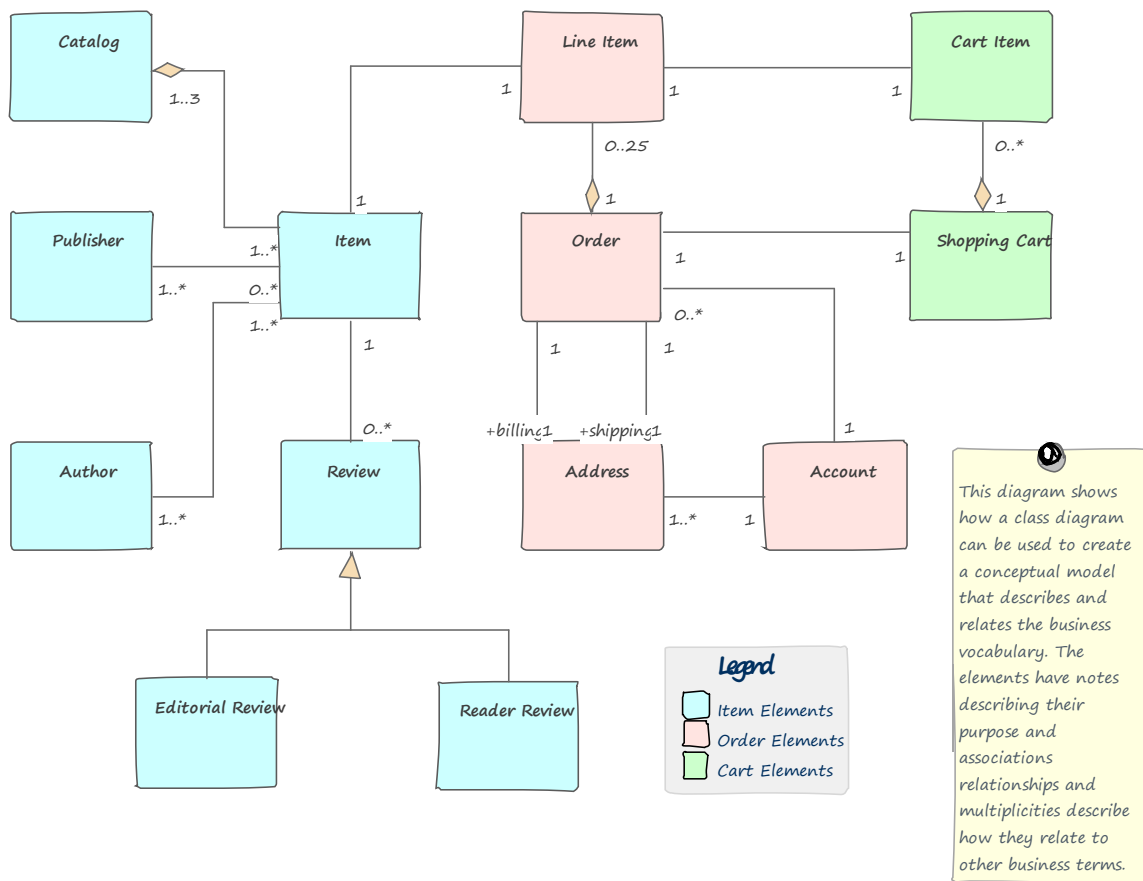
Enterprise Architect is a profoundly useful tool for creating and maintaining information architectures, with its sophisticated and extensive support for standards and its wide range of tools to support information models, from high level classifications and concepts right down to the level of schemas and the elements and columns they are composed of. Tools such as the **Schema Composer** and the **Database Builder**, along with the **UML Class** diagram and Glossary, and the powerful Model Transformation facility, will be invaluable.



Learn More: [Information Architecture](#)

# Conceptual Information Model

A Conceptual Information Model is a high level diagram describing the important information in an enterprise or system; it is typically useful for communicating ideas to a wide range of business and technical stakeholders. Any number of diagrams can be created representing the information at a line-of-business level, but it is also quite common to create a large articulated diagram comprising information entities from a variety of divisions or lines of business of the enterprise or systems. There are also occasions when cross-enterprise information models are useful, particularly in cases such as supply chain modeling or business-to-business models. The diagram typically consists of named entities and their relationships to each other.



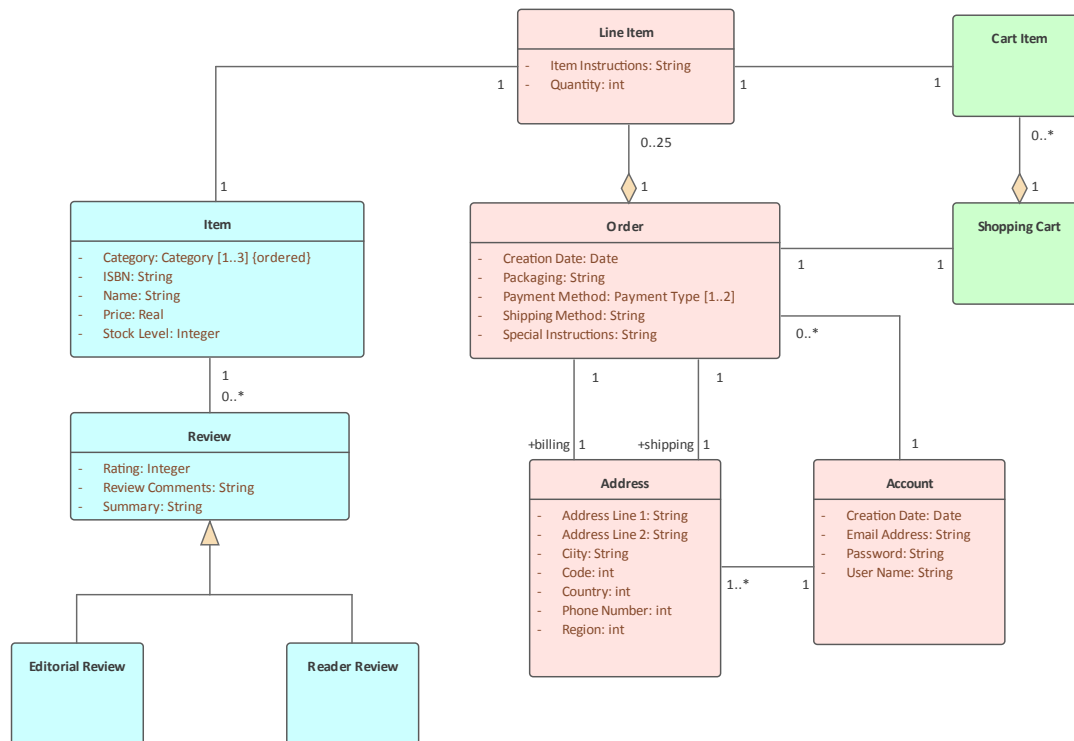
Enterprise Architect has a powerful set of tools for working with conceptual information models, allowing any number of models to be created and viewed and analyzed. A range of tools allow compelling visualizations to be created that can be tailored to provide views that are meaningful to individual or groups of stakeholders. The **UML Class** diagram is the main diagram used for representing conceptual models, and the appearance of the entities can be changed, adding color, styles, graphic images, shapes and more to create appealing diagrams that can also be copied into presentations or included in the Slide Show that is a part of the tool. Element filters can obscure elements with specified properties, and diagram layout and alignment tools can be used to make the diagrams easier to read. There is a

useful Hand Drawn and Whiteboard mode that can be used to make the diagrams appear hand sketched, which often makes them more accessible and attractive to business stakeholders.

Learn More: [Concept Modeling](#)

# Logical Data Model

Logical Data models are an evolution of the Conceptual Information Models and a precursor to the platform and system specific Physical Data models used during database design. The entities they contain should be aligned with their more abstract counterparts in the conceptual model. The data elements are typically described in business language and provide a level of detail that is useful to consumers of the information, such as the teams charged with describing the business process, who often annotate their models with data that is both produced and consumed. The elements typically have attributes with primitive types defined, but the entire model is independent of any technology platform or specific Relational Database Management System.

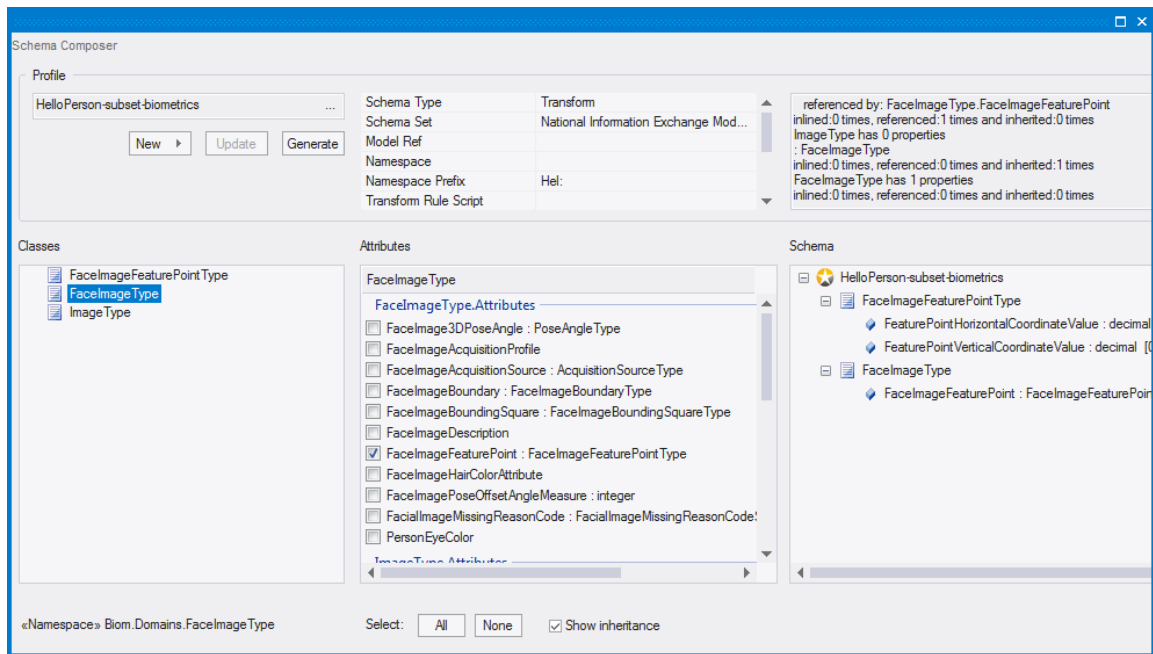


# Schemas and Messages

Industry standards are important in all architecture domains, but they are particularly important in Information Architecture. Information is typically exchanged between organizations and between divisions of an organization. To facilitate the exchange of information, industry bodies and other groups have spent considerable time and effort defining standards in the form of industry and domain specific schemas that codify the way that information should be represented and transmitted between interested parties. These schemas can be large and unwieldy but provide a necessary and often regulated method of exchanging information between disparate parties.

Enterprise Architect can be used to assist an architect when working with these industry schemas, and it has a number of tools including the **Schema Composer** that will make it easy to work with the schemas and create and maintain compliant messages. The Schema Composer can be used to shield the information architect or designer from the details of the schema but will allow them to create and view schema compliant messages. Quite typically the information that an organization wants to share will be a small subset of the information addressed by the entire schema; Enterprise Architect allows a modeler to create messages based on a subset of the schema and in a variety of standards, including JSON, **XSD** and RDF Schemas.



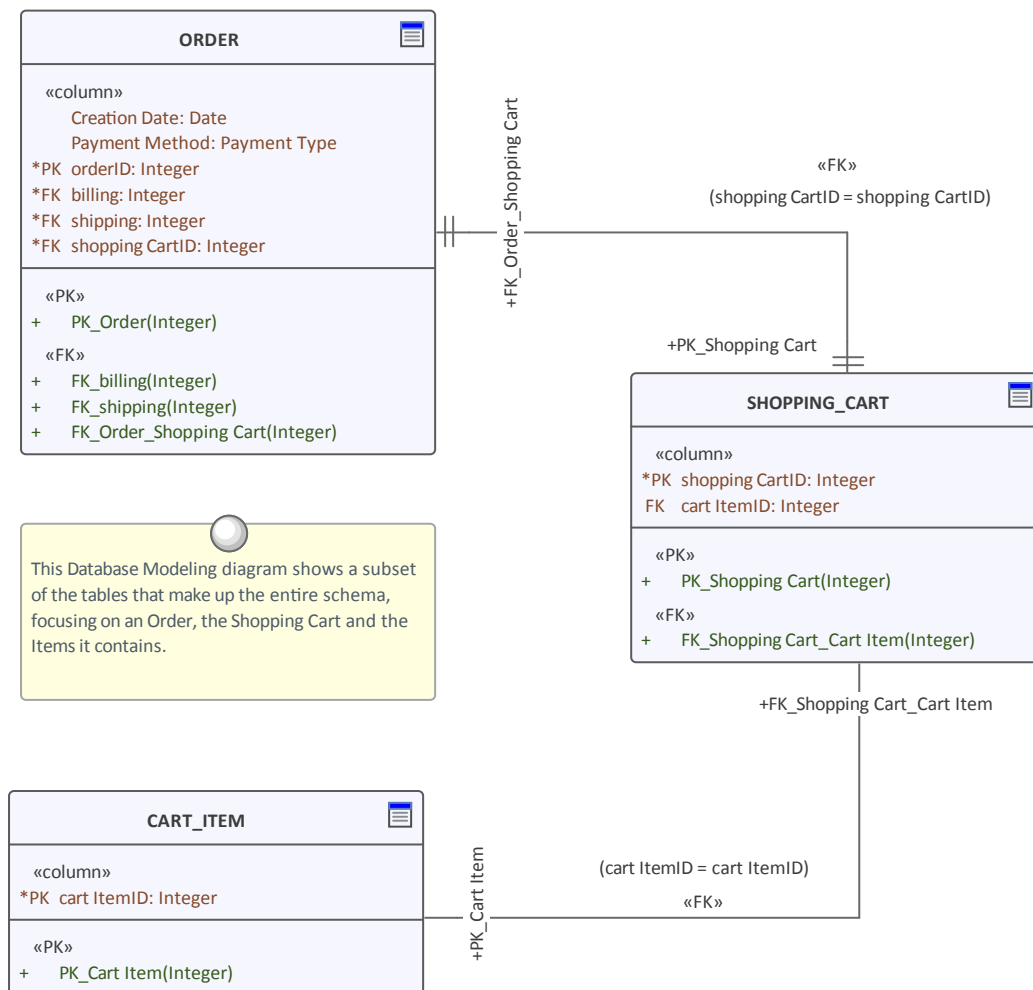


Learn More: [Schema Composer](#)

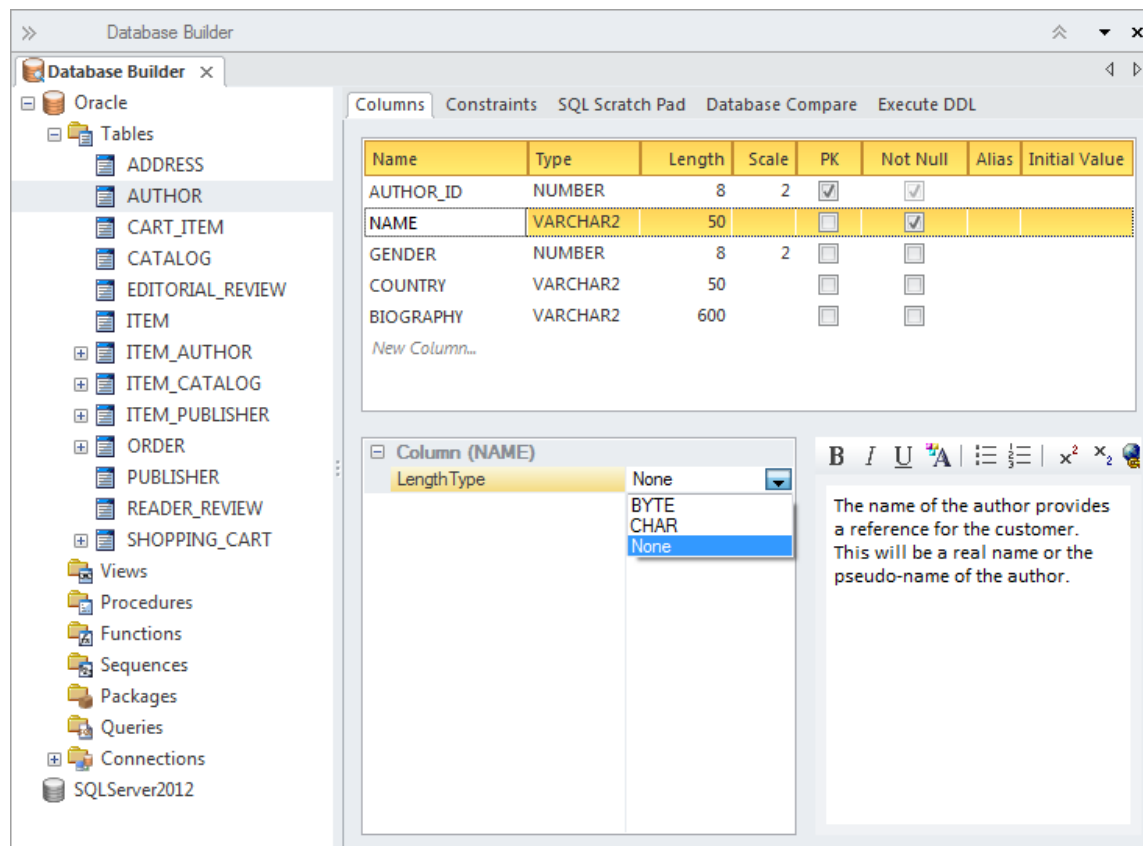
# Physical Data Model

Physical Data Models are typically not themselves included in an Enterprise Architecture because of their level of detail; nevertheless they provide valuable information that can be used to create abstractions and often provide a useful starting point for the Information Architect. The models are often highly normalized with de-normalization added for transaction efficiency and the names of database objects such as tables and views are often abbreviated and abstruse. The Information Architect can, however, use these models to derive the logical data models and then in turn the conceptual information models for baseline architectures. Even when the information models are initiated from the top-down creating the bottom-up models can provide a useful check to ensure nothing has been overlooked.

Enterprise Architect has some powerful tools for working with live databases and can be used to reverse engineer existing databases in a wide range of technologies into UML based physical data models. A variety of database objects can be imported and diagrams can be automatically generated representing the Tables, Views, Stored Procedures, Functions and other objects and the way they relate to each other. Enterprise Architect can also be used to create database schemas from scratch or to create them by using the Model Driven Transformation facility within the tool to transform a more primitive model such as a business domain model into a database schema.



Apart from the ability to reverse engineer a physical database schema Enterprise Architect has a powerful tool called the **Database Builder** that can be used to connect to multiple Relational Database Management Systems and to reverse engineer and synchronize models and live databases. Learn More: [Data Modeling](#)

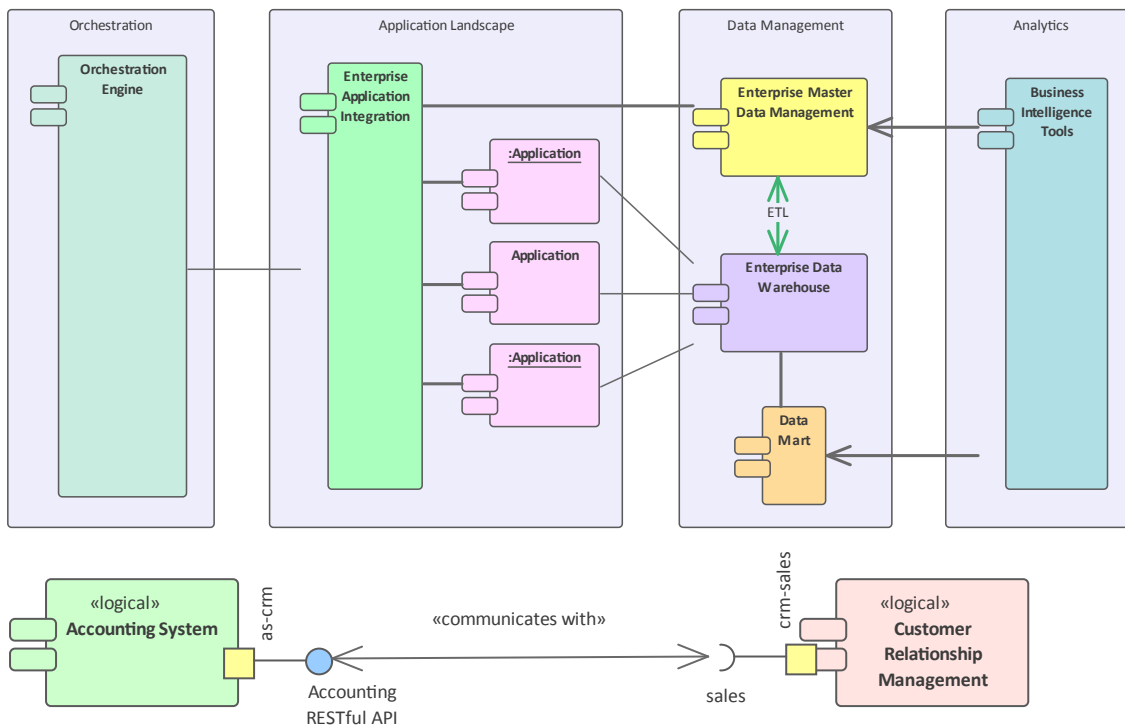


# Application Architecture

The Application Architecture provides an important catalog of the applications in the enterprise describing the work that they do to transform, transmit and store information. The architecture also describes the interfaces that are required or provided by the applications and the way the applications interact to carry out the activities described in the business models such as the Business Process diagrams. The catalog of applications, interfaces and the diagrams and matrices that describe their interaction only need to be defined once at the enterprise level. An application architect will be able to draw upon this inventory of existing artifacts to create new architectures, classifying them as part of the baseline and potentially the future state architecture. Where an architecture introduces new applications, these can be added to the description of the target state.

The Application Architecture will typically consist of a description of the baseline and target architectures with a series of transitions defined that can be executed and that would be described on Roadmap diagrams.

Learn More: [Application Architecture](#)



This Component diagrams shows the details of the interface between two Logical Components using Ports and Interfaces.

# Application Lists Diagrams and Matrices

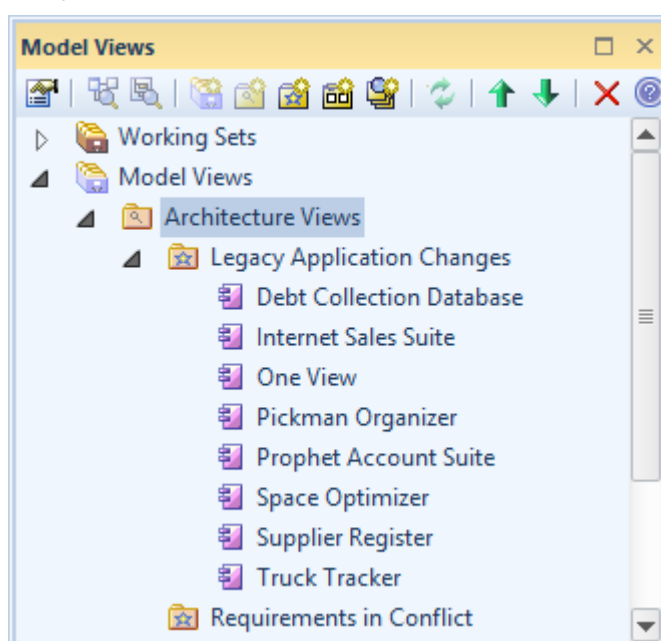
## Application Lists

Lists, sometimes referred to as catalogs, provide a simple way of viewing a group of applications. The list can be based on any number of criteria including a list of baseline applications, a list of target applications, or a list based on complex criteria such as the applications owned by a particular stakeholder that were installed before a certain date and that have a lifetime of retirement. The lists can be used simply to view the applications or to add new ones and change existing ones.

Enterprise Architect has a wide range of tools for working with lists of applications and provides ways of constructing lists dynamically, running searches across a selected part or the entire repository, applying filters and sorting mechanisms that can restrict and order a list based on particular criteria. Diagrams can be viewed as a list of elements or even viewed as a Gantt chart. There is also a very flexible **Specification Manager** that allows any Package to be viewed as a list of elements; this has built-in filtering, and notes and other properties can be edited inline.

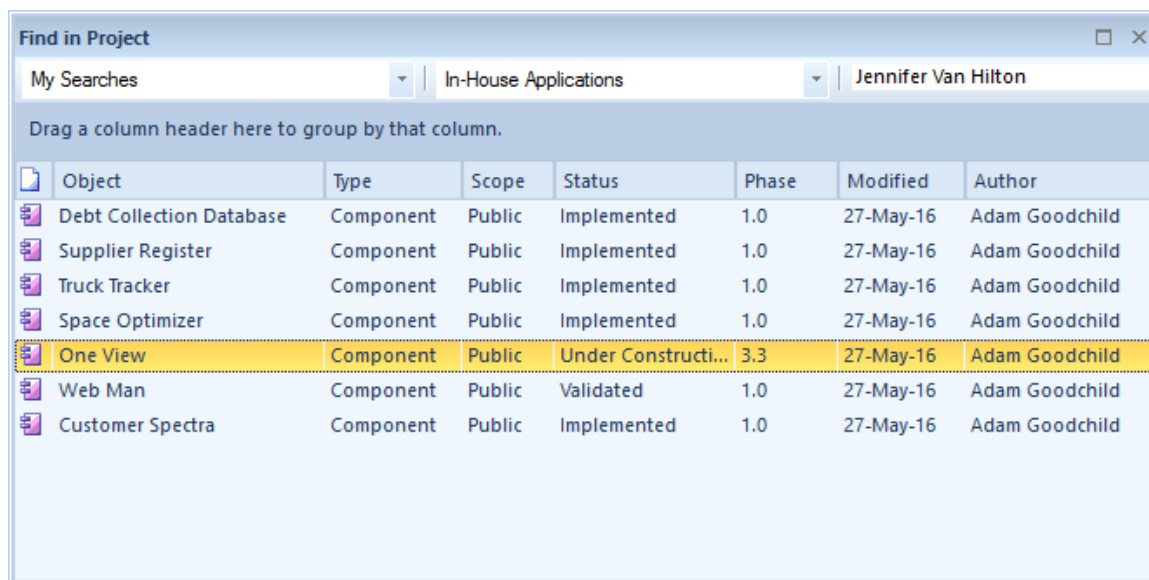
Component	Notes	Status	Phase
Prophet Account Suite	This is a commercial product that was purchased at the time the original organization was set up and comprises the following modules: General Ledger, Accounts Payable, Accounts Receivable and Bank. There have been a number of other modules that were either developed internally or purchased as third party applications that provide additional functionality such as payroll.	Implemented	1.0
Space Optimizer	This is a warehouse space optimizing application that was originally an Open Source project that we contributed to but has since been discontinued. A group of consultants worked on it subsequently and integrated it with the Stock Manager and Stock Optima and the Warehouse Suite.	Implemented	1.0
Stock Manager	This is an off the shelf package that is highly configurable and has got an open interface that allows it to be connected with a number of other products	Implemented	1.0
Stock Optima	This is a custom built application that was developed by a consulting group we own the IP but it is maintained as part of a ten year support contract. It was developed when the organization purchased a number of smaller sellers and there was a need to optimize inventory across a group of dispersed warehouses.	Implemented	3.3

The **Model Views** facility is another way of creating a list of elements in a tree structure similar to the **Browser window**, but the elements can be added freely to a favorites Package or generated by a simple or sophisticated search. This is particularly powerful when working with applications and any number of tree based lists can be created.





The **Model Search** facility is another tool that can create a list of applications or interfaces based on predefined or user created criteria. This facility can function as an analytics tool to answer questions posed about the Application Architecture. For example - find the list of applications that were built in-house that are critical to the business and have a particular owner.



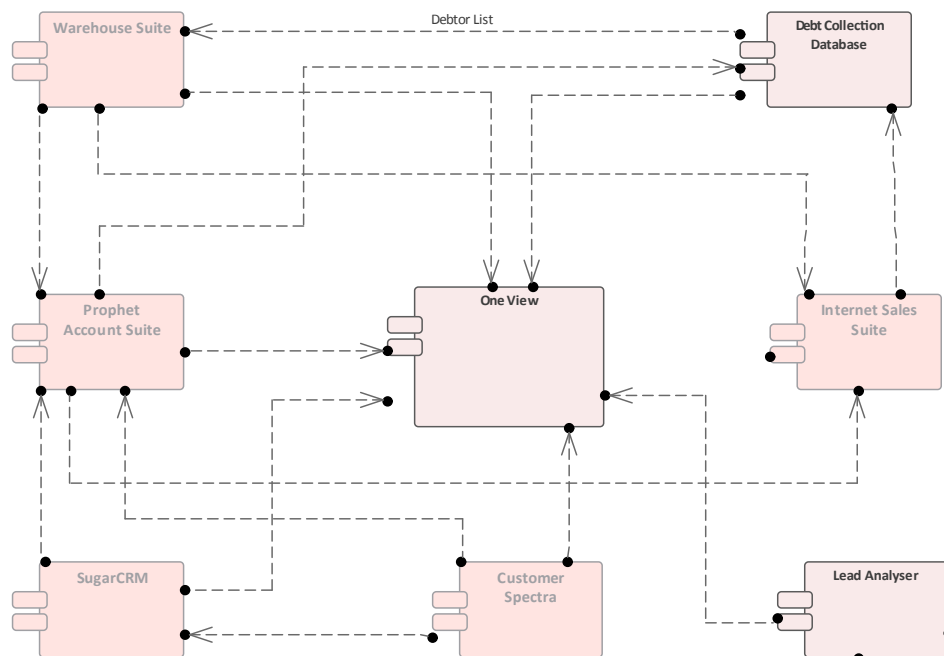
The screenshot shows a window titled "Find in Project" with a search bar containing "My Searches", "In-House Applications", and "Jennifer Van Hilton". Below the search bar is a table with columns: Object, Type, Scope, Status, Phase, Modified, and Author. The table lists several applications, with "One View" highlighted in yellow.

Object	Type	Scope	Status	Phase	Modified	Author
Debt Collection Database	Component	Public	Implemented	1.0	27-May-16	Adam Goodchild
Supplier Register	Component	Public	Implemented	1.0	27-May-16	Adam Goodchild
Truck Tracker	Component	Public	Implemented	1.0	27-May-16	Adam Goodchild
Space Optimizer	Component	Public	Implemented	1.0	27-May-16	Adam Goodchild
One View	Component	Public	Under Constructi...	3.3	27-May-16	Adam Goodchild
Web Man	Component	Public	Validated	1.0	27-May-16	Adam Goodchild
Customer Spectra	Component	Public	Implemented	1.0	27-May-16	Adam Goodchild

## Application Diagrams

Diagrams are one of the most commonly used mechanism for expressing ideas in the Application Architecture. They can be large format diagrams intended to give a view of an entire landscape of applications and printed onto poster size paper, or smaller diagrams intended for documentation. An application Communication diagram is a typical candidate for a large A0 size diagram where all (or at least the most significant) applications are printed in a single diagram with lines showing how they are wired together, including

payloads if required. Equally an application Communication diagram could be created with just a constellation of applications that realize a business service or capability.



## Application Matrices

Matrices provide an alternative view to diagrams and lists and allow stakeholders to view the relationship between two sets of elements in a familiar spreadsheet-like view. The matrix could be between applications and capabilities or applications and business processes or any other elements, including between the same set of applications to show connectivity. The matrix profiles can be set up and saved at the time the repository is created, and as elements are added

the relationships will become visible. If further analysis is required a matrix can also be conveniently exported to a CSV file so it can be opened in a spreadsheet application.

Relationships between Supplier Management and Physical Components

Source:  ... Type:  Link Type:  Profile:  Refresh

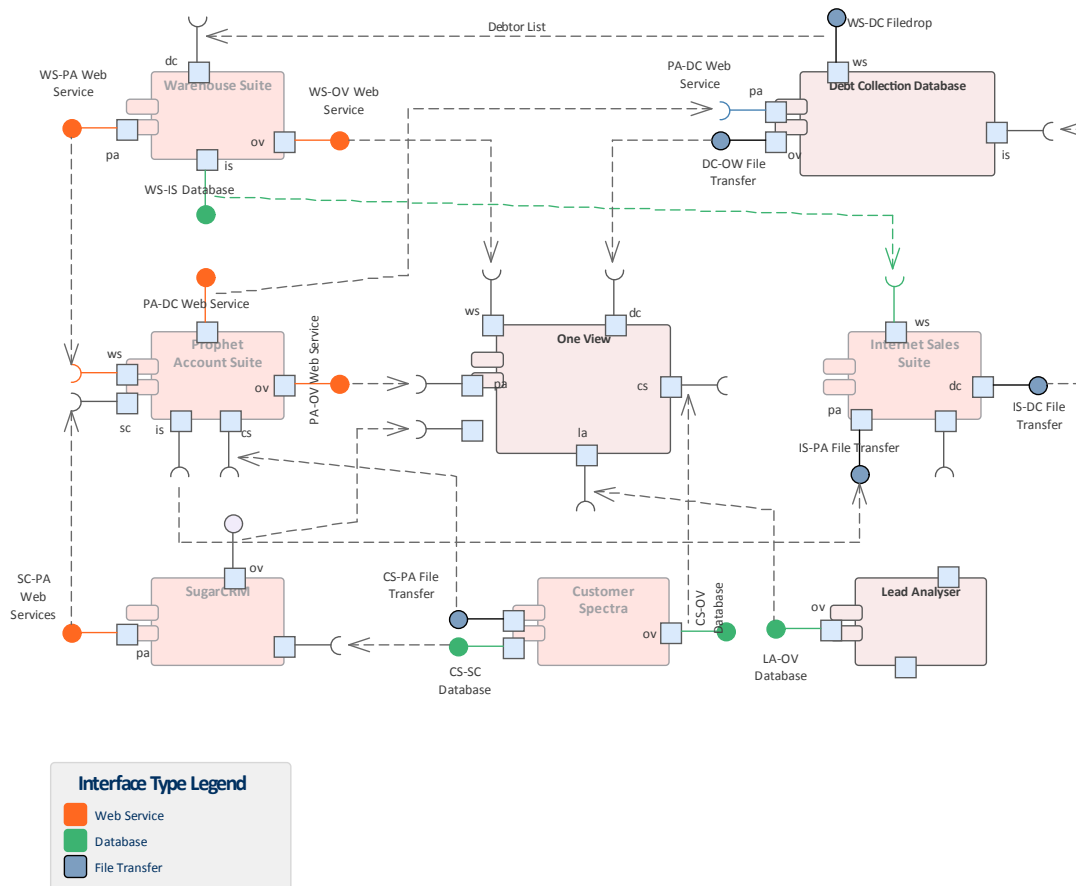
Target:  ... Type:  Direction:  Overlays:  Options

Target +	Books in Print Database	CVM Supplier Central	Debt Collection Database	Dynamic Payroll	ElectraPay 2	Enterprise Supplier Manager	Financial Reports	Lead Analyser	One View	Pickman Organizer	Procurement Access Database	Prophet Account Suite	Space Optimizer	Stock Manager	Stock Optima	SugarCRM	Supplier Register
+ Source																	
Activate Supplier											↑	↑					↑
Administer Suppliers																	
Create Supplier Performance Management Plan												↑					↑
Create Supplier Record		↑				↑					↑	↑					↑
Identify Suppliers																	
Prioritize Supplier		↑				↑											
Register Supplier																	
Retire Supplier																	
Select Performance Measures and Tools												↑					
Supplier Management																	

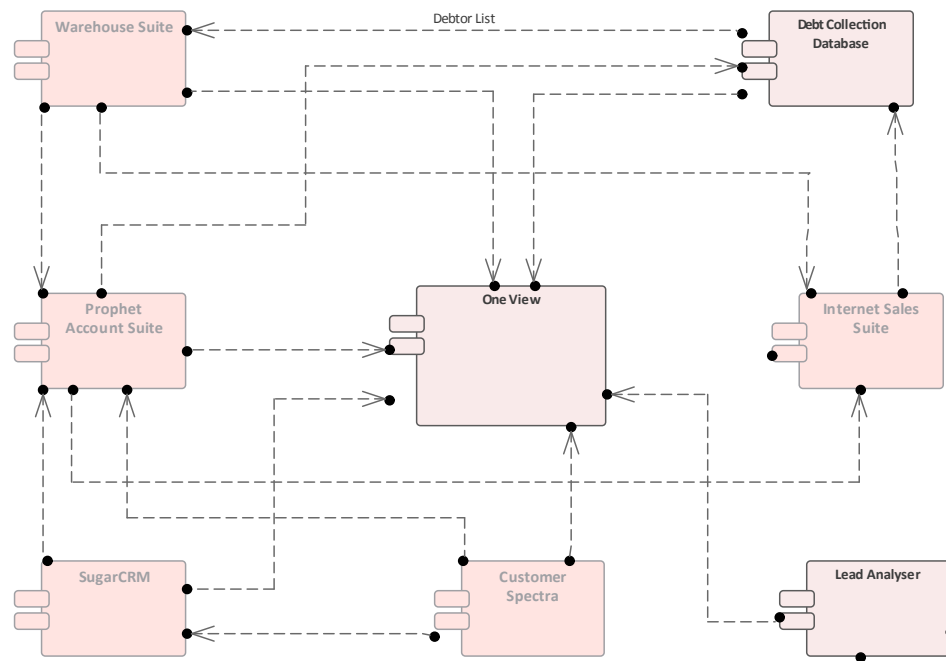
# Application Communication

Application Services are not typically delivered by single applications, but are more commonly delivered by a concert of applications working together. The way these applications communicate to carry out the work described in the business capabilities is often built up over many years and can be quite complex. More often than not the individuals or the teams that devised the connectivity between the applications have moved on, and typically little documentation exists. If documentation does exist it is usually specific to a single application and does not describe the type of overview needed by an architect investigating the landscape. In this way the architect is often working to understand and describe how the landscape of applications came to be connected and how they currently communicate. One of the issues an architect will face is the balance between not describing application communication in too much detail but also not creating trivial and meaningless representations. There is also a need to be able to toggle between views suitable for high level executives and those suitable for Implementation teams.

Enterprise Architect has a powerful feature that allows a diagram that displays the details of application communication including Interfaces, Ports and Payloads to be collapsed, so that it simply shows the applications and the communication lines.



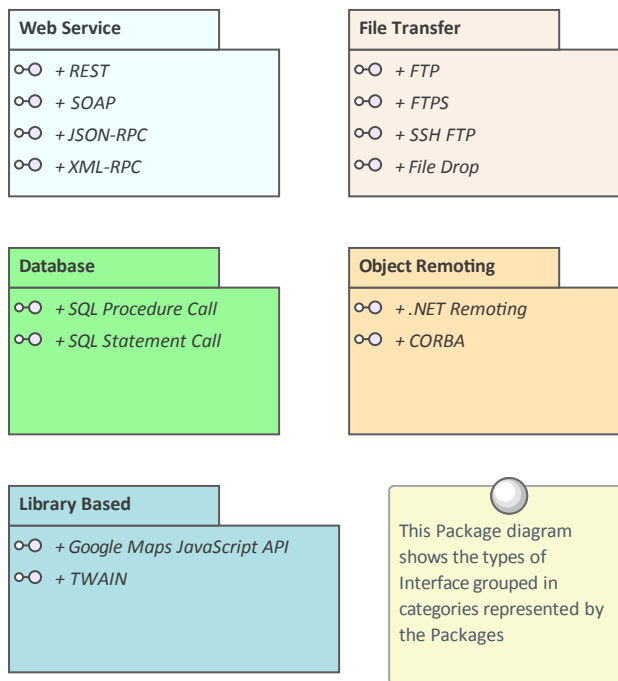
An architect might decide to create a single diagram and to toggle between the two views by changing the diagram properties or, alternatively, a separate diagram could be constructed. Either way the two diagrams are just two different views of the same underlying repository elements, and changing an element or connector's properties in one diagram will result in changes in the other. The appearance of the elements, including color and layout, will be independently maintained in the two diagrams.



# Interface Lists

An Interface List and the related diagrams provides a useful view of the complexity of the enterprise. The number and types of the interfaces helps the architect understand how applications are connected and the level of complexity that exists. When these diagrams are drawn for baseline architectures the result is what people often term a 'Spaghetti diagram', because of the numerous lines that connect applications together. This diagram is quite often the architectural evidence for what has been obvious in symptomatic failure of systems, and in the costs and problems associated with integrating new applications into the landscape.

Often these diagrams are drawn in a general purpose drawing tool by an external consultant, and once they have served their purpose of highlighting the systemic problems are discarded. With Enterprise Architect there is the opportunity to create the diagrams and, once the message is conveyed to management, to reuse the diagram and the associated repository elements to begin the process of re-designing the application integration architecture and untangling the spaghetti that has built up over years.





# Capabilities and Applications

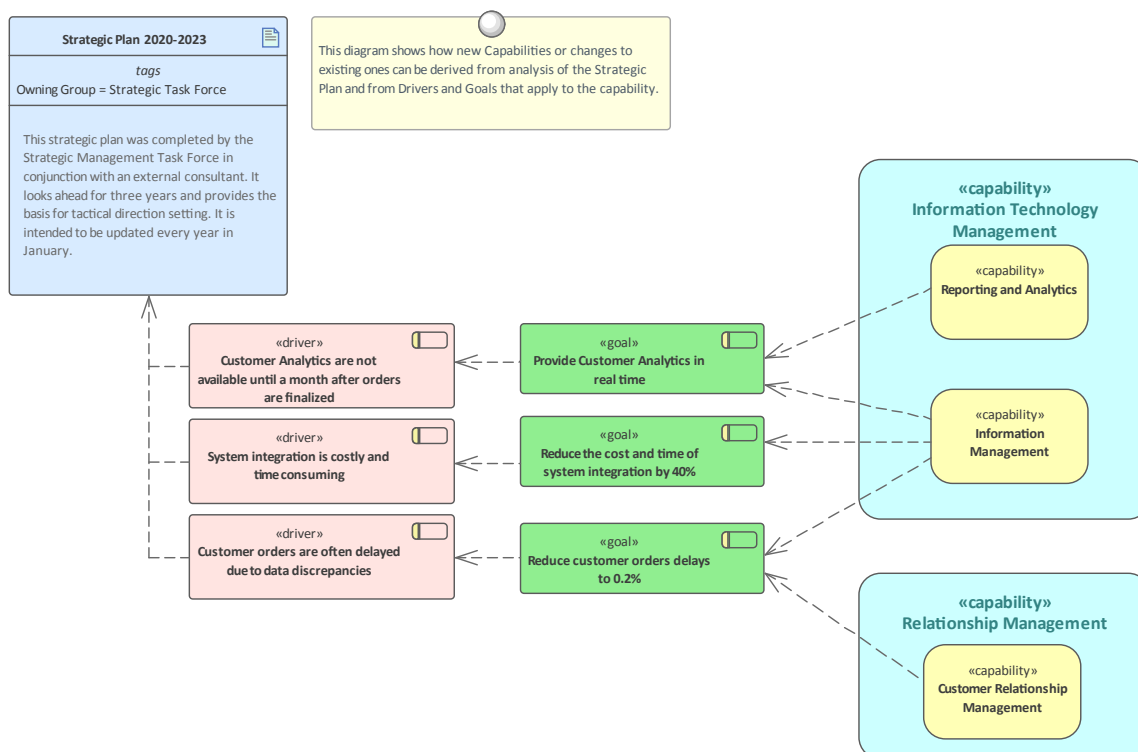
Enterprise Architecture, as with any type of design practice, requires a context or point of reference. The Capability Model is a profoundly useful starting point for describing architectures as it is often readily available (or, if not, it is relatively easily constructed) and provides the illusive bridge between the business and information technology disciplines. The business community implicitly understands this model and it has the advantage of being organizationally agnostic - it can be devised without reference to or dependency on the process models. While it is possible to pin architecture work to a process model, this is often fraught with issues related to the model such as:

- The process model is often time consuming to create
- Definition requires access to operational staff, who are often busy
- The process model is volatile and has often changed before it has been completed, and
- The process model is expensive and time consuming to maintain

The Capability Model, on the other hand, is stable, relatively easy to maintain, and its definition typically doesn't require access to operational staff. The Capability Model belongs to the business and its construction is typically business led.

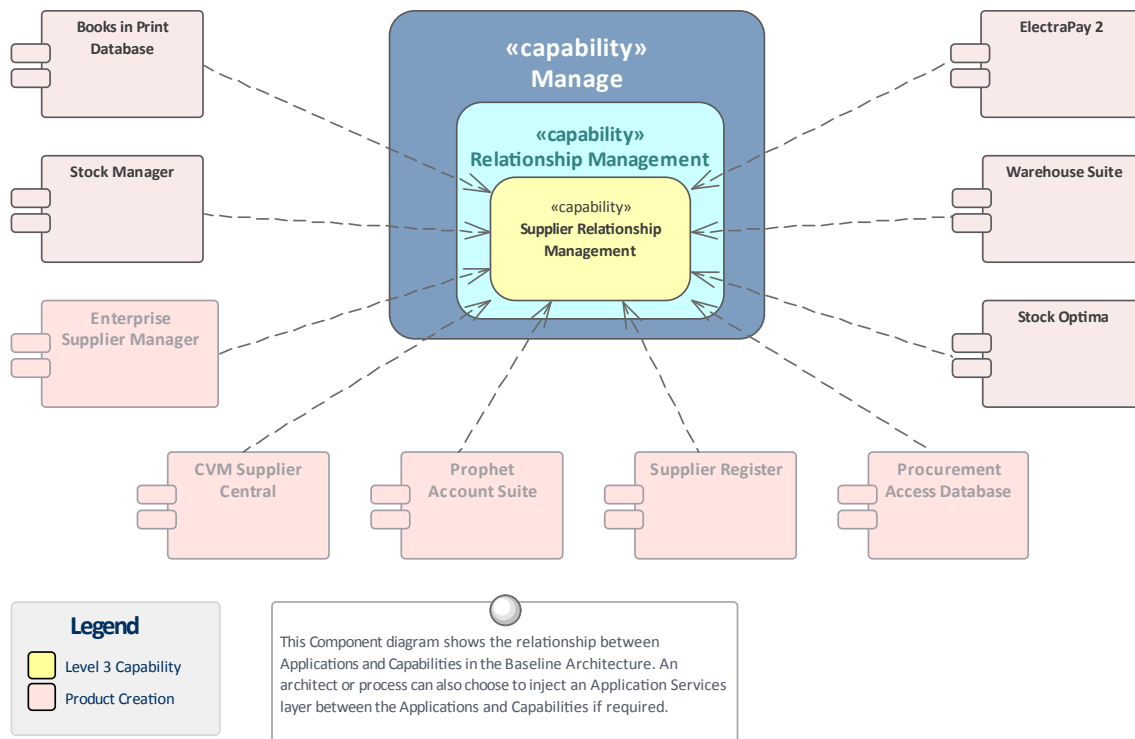
The Strategic Plan sets the direction for the enterprise over a horizon of three to five years and this document - in conjunction with Annual Updates or an Annual Business

Plan - will be an invaluable source for the derivation of capabilities. A large number of the capabilities will span lines of business and thus it will be difficult to find a single owner or manager who is responsible for them. Enterprise Architect can be used to create diagrams that will provide clarity for a range of stakeholders, including the senior Executives, the Business Line Managers and senior Information Technology staff. The Strategic Plan can be modeled as an Artifact and hyperlinked to the actual document in an external document repository. This allows the Architect to use the Artifact as a surrogate for the Strategic Plan and to link Drivers and Goals to the document, which in turn can be linked to Capabilities.



Enterprise Architect can also be used to define the baseline architecture by describing the existing applications that deliver a capability, or to define a target state by prescribing the planned applications that will deliver the capability.

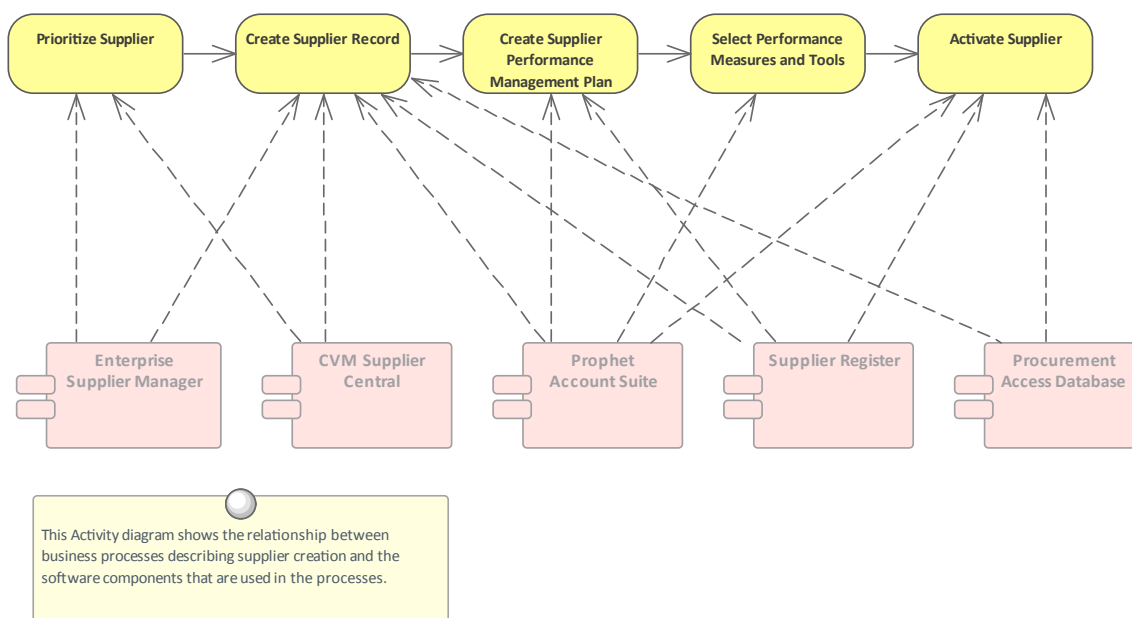
Regardless of the purpose of the architecture, the capabilities provide a sound and business-driven anchor point for the applications.



Capabilities are delivered by a range of business, information technology and personnel changes, so an Enterprise Architect will typically be involved in a range of activities that might include change management, staff training, information management, and application and technology stack changes.

# Business Processes and Applications

An architecture can provide profound insights into the relationship between the business processes that model how an organization carries out its work, and the application components that assist them. Enterprise Architect can be used to create expressive visualizations of these connections, which are particularly useful when analyzing current state models. Organizations have typically been formed through a diverse set of changes and developments, resulting in a complex set of processes and an equally complex set of applications that interact together. This often leads to redundancy, duplication and inefficiencies that cannot be seen unless an architecture is constructed to visualize the connections and to highlight the issues.



The same relationships between business processes and

applications can be visualized using the powerful **Relationship Matrix**, which provides a spreadsheet-like interface that some stakeholders will find more accessible and expressive. Application redundancy can be determined by viewing a particular process and seeing the number of markers it has indicating the applications that support that process.

Relationships between Supplier Management and Physical Components

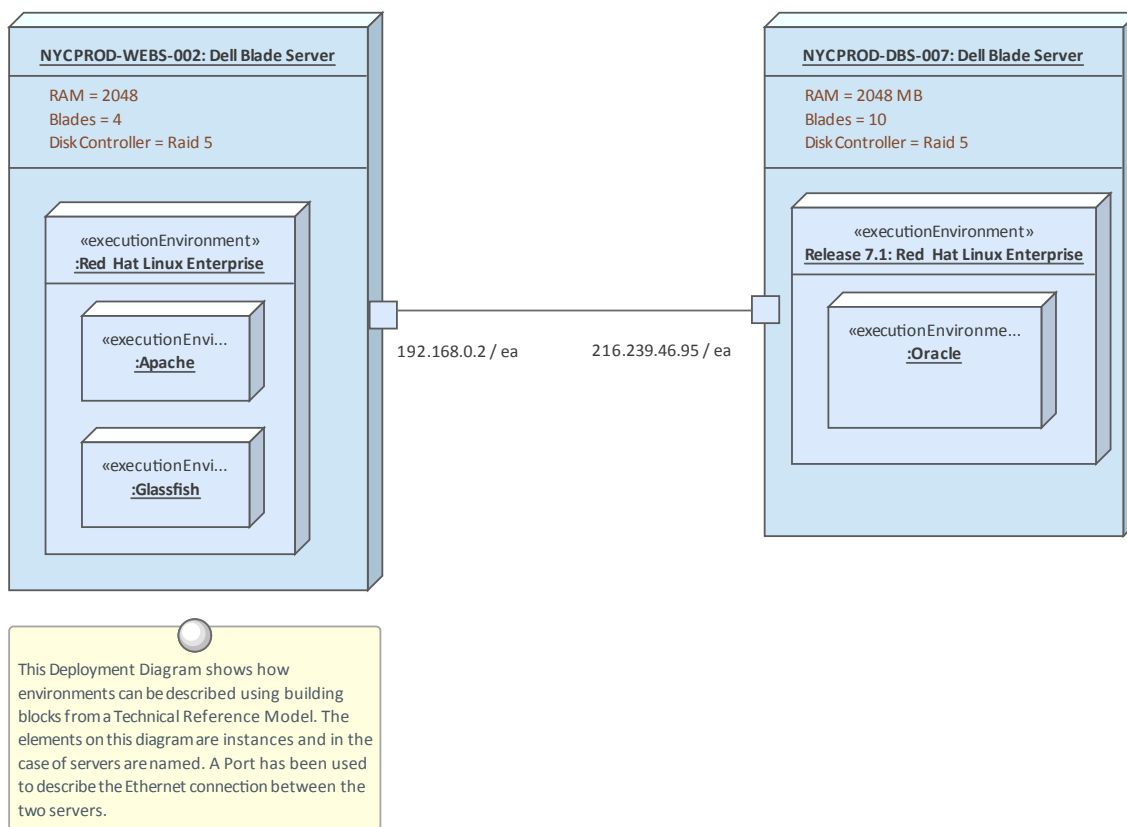
Source:  ... Type:  Link Type:  Profile:  Refresh

Target:  ... Type:  Direction:  Overlays:  Options

Target +	Books in Print Database	CVM Supplier Central	Debt Collection Database	Dynamic Payroll	ElectraPay 2	Enterprise Supplier Manager	Financial Reports	Lead Analyser	One View	Pickman Organizer	Procurement Access Database	Prophet Account Suite	Space Optimizer	Stock Manager	Stock Optima	SugarCRM	Supplier Register
+ Source																	
Activate Supplier											↑	↑					↑
Administer Suppliers																	
Create Supplier Performance Management Plan												↑					↑
Create Supplier Record		↑				↑					↑	↑					↑
Identify Suppliers																	
Prioritize Supplier		↑				↑											
Register Supplier																	
Retire Supplier																	
Select Performance Measures and Tools												↑					
Supplier Management																	

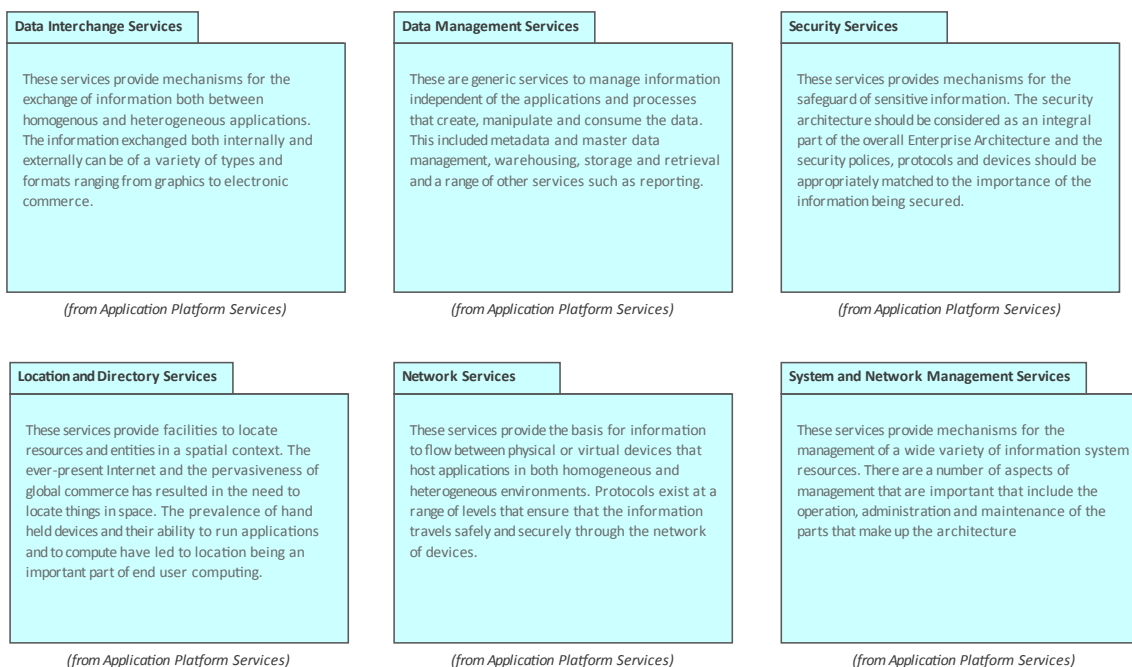
# Technology Architecture

The technology architecture underpins the other architectures, providing a description of the logical, physical and virtual infrastructure that supports the execution of application services, which in turn support information and business functions and services.



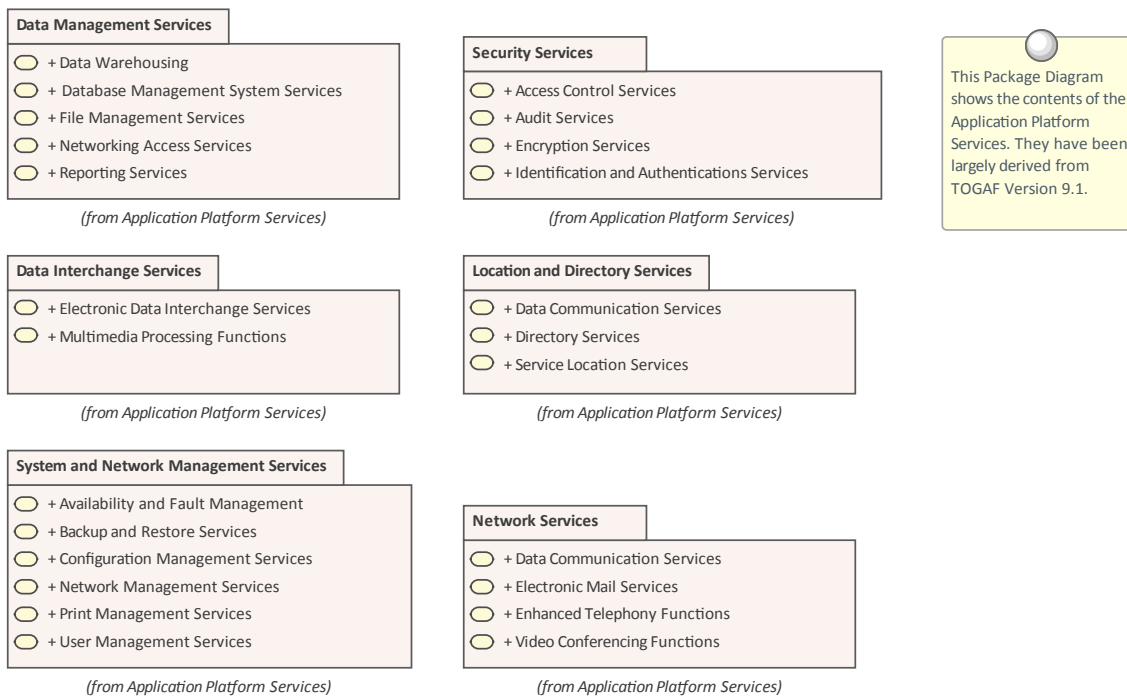
# Application Platform Services

The concept of an Application Platform is described in TOGAF; it provides the benefit of portability and interoperability by defining a set of services that are made available to applications in a standard and repeatable way, and also by the fact that the platform is underpinned by communication infrastructure made accessible to the application platform through a standards-based communication infrastructure interface. This Package diagram illustrates the taxonomy of Platform Services.



This second Package diagram shows the individual Platform Services that are grouped within each Package. The Platform Services are elements that can be linked to other model elements, including the applications that rely on the service and the underlying mechanisms that provide the

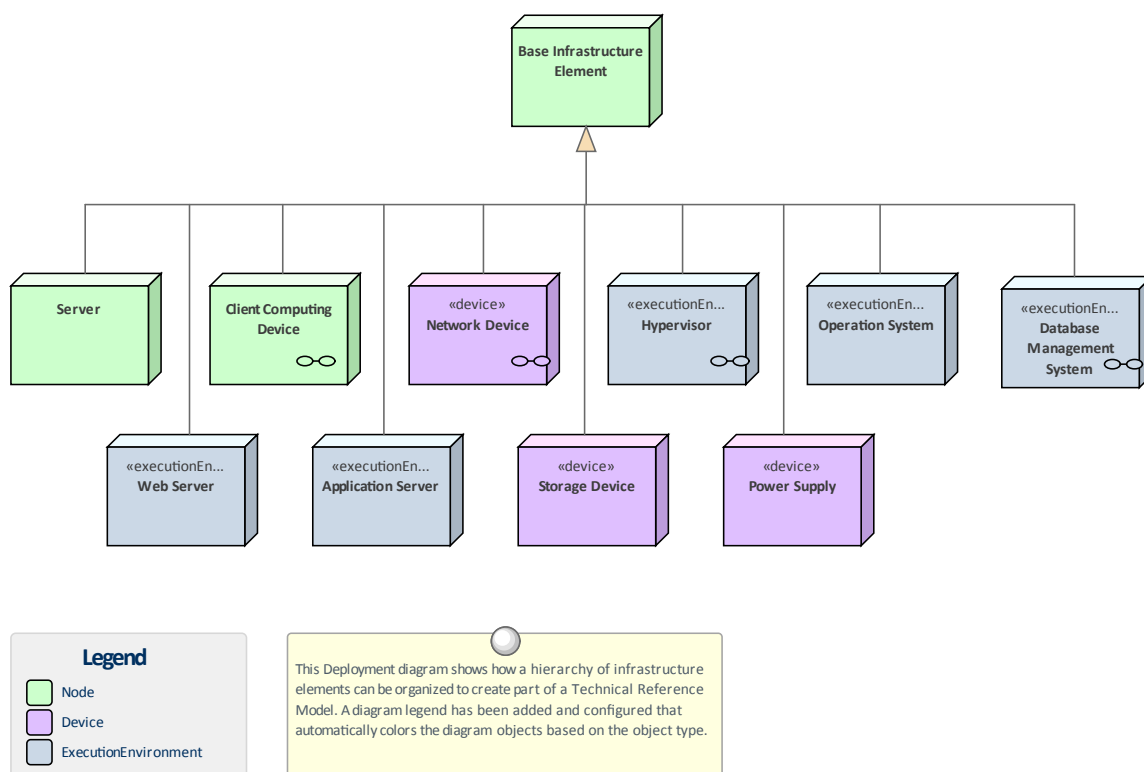
## service.



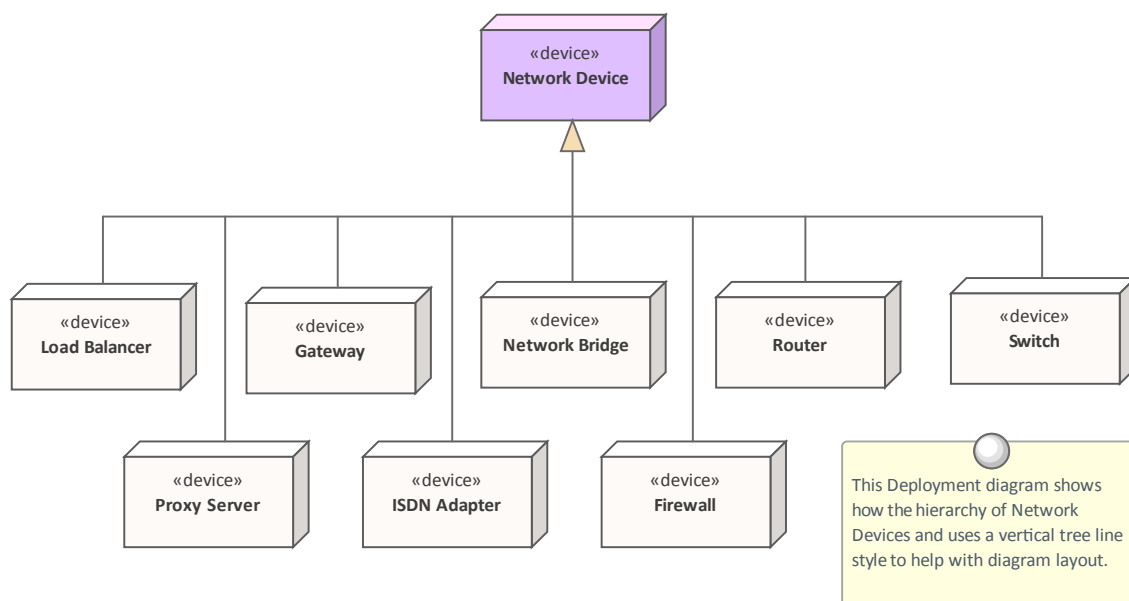


# Technical Reference Model

The Technical Reference Model (TRM) provides a reference of generic platform services and technology elements and acts as a substrate upon which technology architectures can be built. The TRM provides a set of architectural and solution building blocks that will ultimately provide the platform for business and infrastructure applications that will deliver the application and infrastructure services. The Technical Reference Model ensures that architectures are created consistently and repeatedly based on a standard set of elements. The model should be created as part of the set up of the architecture programs but will typically require extending as technology standards are introduced and retired.



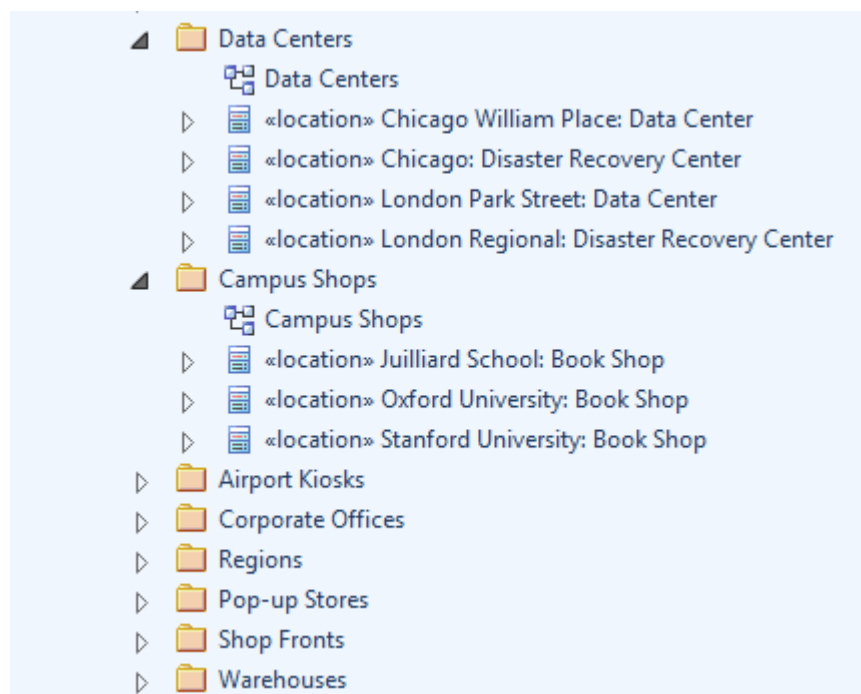
The Technology elements section of the TRM is a hierarchical structure and classifies parts of the infrastructure based on their function as a Physical or Virtual Node, Device or Execution Environment. This allows any type of technology concept, physical or virtual to be modeled including a wide range of infrastructure pieces such as servers, routers, operating systems, hypervisors, database management systems, web servers and more.



Once defined, the Technical Reference Model can be used as the basis for all Infrastructure Architecture models by creating instances of the Infrastructure elements; for example, a particular data center or facility might contain a specific manufacturer's switch or router.

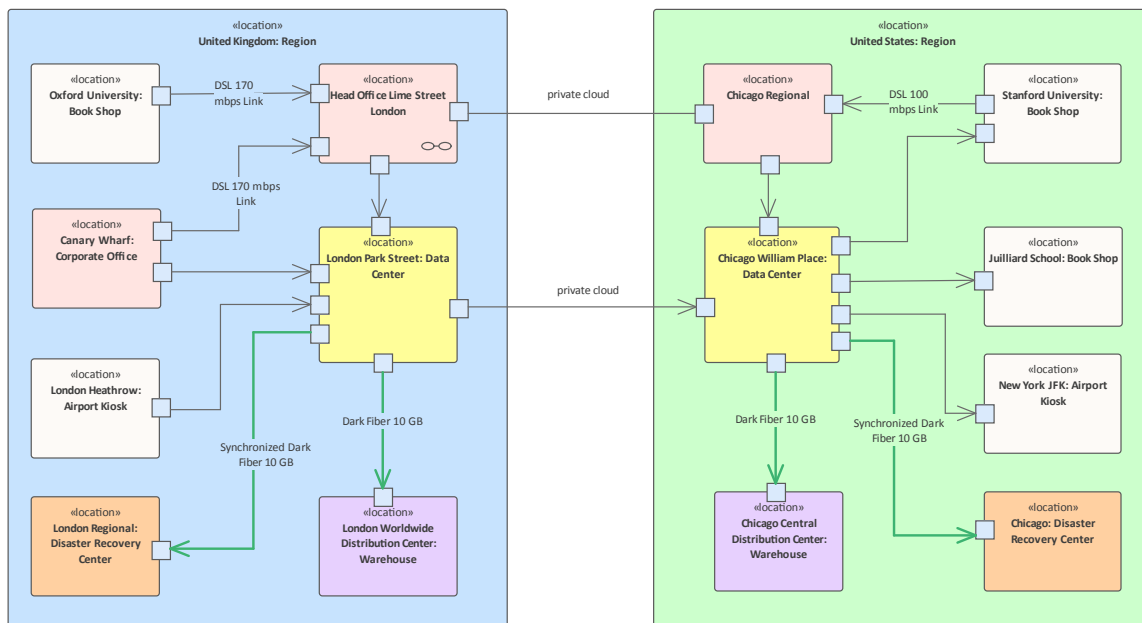
# Infrastructure Facilities

An organization of any appreciable size will typically operate a number of facilities or locations. These will include structures such as corporate offices, data centers, disaster recovery centers, shop fronts and warehouses, and a variety of specific locations such as hotels, airports, transmission centers, vehicles, vessels, car parks and more. These locations will typically be modeled at a business level to model concerns such as where business is performed, the location of processes and personnel, and which locations are required for particular capabilities and business functions. For example, what business processes are performed in an ambulance? Similar models will be required at an Information level to indicate where data is created, stored and utilized; for example, in which locations can a patient record be created? At an application architecture level, the relationship between applications and particular locations will be important; for example, which applications are used in robot assisted warehouses and which applications are required for a pop-up kiosk?



At an infrastructure level, where physical or virtual machines are operating, location is critical for reasoning about accessibility, bandwidth, performance, security and more. It should be noted that, from the perspective of Cloud computing, the location of infrastructure becomes immaterial other than for mapping accessibility zones or their equivalents. Indeed, in a Cloud environment it is mostly impossible and even irrelevant to determine the location of the physical hardware that is executing a process. Enterprise Architect provides a wide range of tools for modeling locations at all levels, and this is particularly true at an infrastructure level. The same elements that are used to model the business locations can be augmented to create expressive models that represent how the infrastructure relates to facilities, including high level models of the communication fabric between different locations and the protocols used. Unified Modeling Language Ports allow drill-down to the internal structure of a facility. For

example, a high level diagram might describe the communication path between a corporate office and a data center using a Port on the boundary of each facility. A child diagram could then describe the routers, switches, servers, and other infrastructure elements that make up the data center or corporate office.



# Stakeholder Modeling

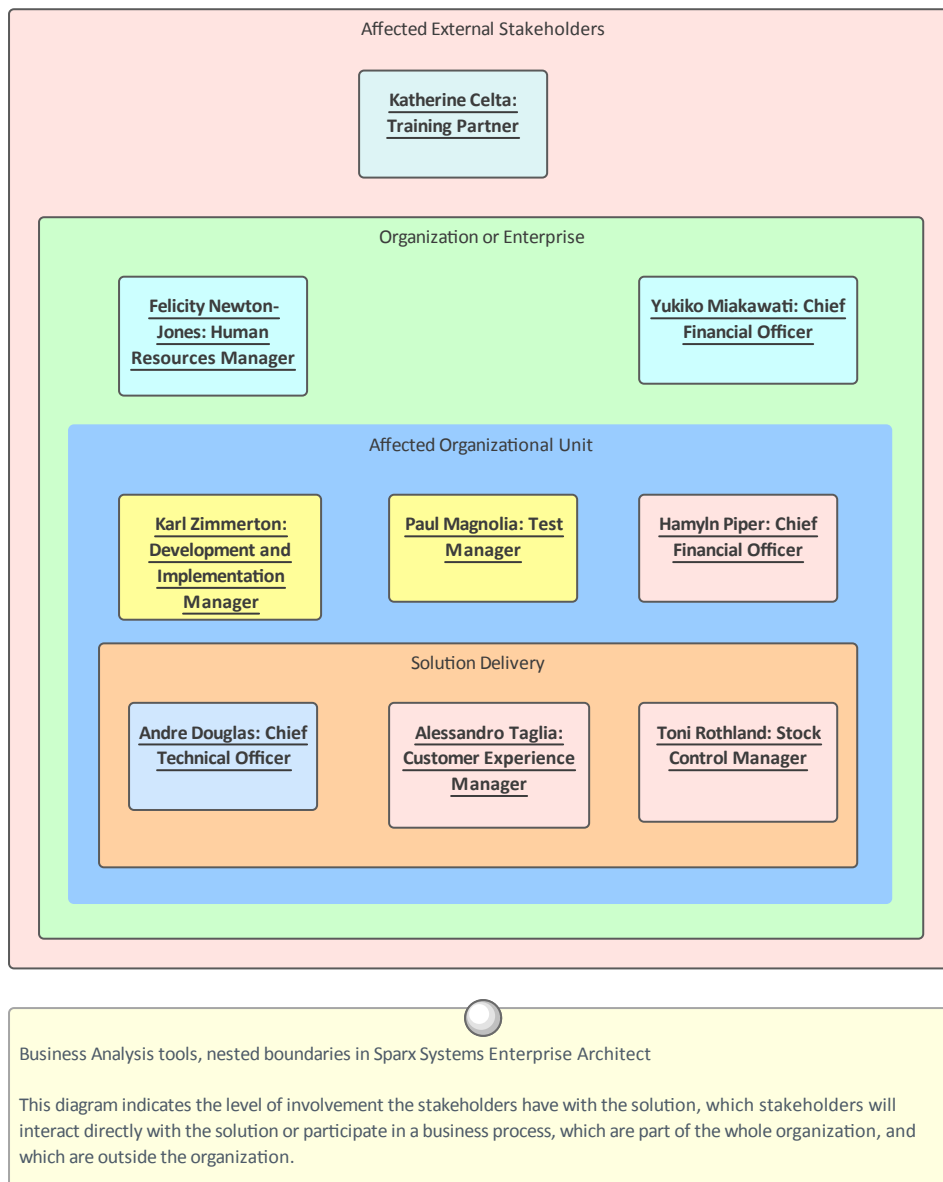
Stakeholder Management is critical to successfully initiating and maintaining an architectural program. Architects will need to engage with a wide range of stakeholders, from senior executives through to implementation staff. The engagements will often require political sensitivity, diplomacy and flexibility to ensure that the stakeholders' needs and concerns are dealt with appropriately. Providing relevant and tailored views of the architectures will be critical to ensuring that the stakeholders are kept informed and that they spend the requisite time needed to understand the impact the architectures will have on their domain. Having a comprehensive communication plan is also critical to ensuring that the stakeholders receive the information they require and that they maintain an interest and have input into the architectures. Stakeholder analysis should be undertaken early in the set up of an architecture program and early in the development of an architecture, as an individual stakeholder's position with respect to a particular architectural initiative can vary. A matrix can be used to describe the positions key stakeholders or stakeholder groups hold with respect to aspects of the architecture program, such as how supportive they are, or what commitment they have to the architecture. The matrix can be updated and monitored over the lifetime of an architecture and remedial efforts made to manage the relationships with key stakeholders.

Target +	Ability to Disrupt	Commitment to Architecture	Flexible	Supportive	Understanding of Domain
+ Source					
Chief Financial Officer	M	H	L	H	L
Chief Technical Officer	M	H	H	H	L
Customer Experience Manager	H	L	M	M	H
Development and Implementation Manager	H	H	L	M	M
Facilities Manager	L	L	H	M	M
Human Resources Manager	L	M	L	H	H
Operations Manager	H	H	L	M	H
Stock Control Manager	H	L	M	L	M
Test Manager	L	M	M	H	H
Training Partner	L	L	H	M	H

Enterprise Architect has a wide range of facilities and tools that can assist with Stakeholder Management. These include the ability to model the individual and groups of stakeholders, to classify them in a taxonomy and to show the extent of their influence using a series of nested Boundary elements.

There is a wide range of lists, diagrams and matrices that will be relevant to certain stakeholders, including lists presented in the **Specification Manager**, Component diagrams describing Applications, Class diagrams used to present the Information Architectures, principles and a range of other ideas. The **Calendar** and **Model Mail** are useful tools for keeping stakeholders informed about things of interest and important events in the architecture program.

## Stakeholder Onion Diagram



There is also a wide range of additional tools that assist when tailoring the information in the repository for individual or groups of stakeholders.

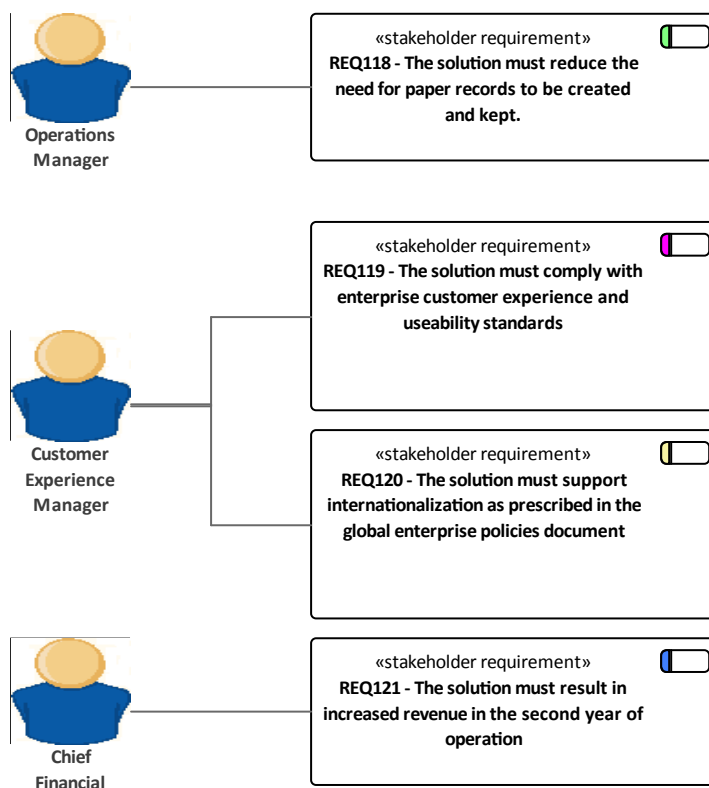


# Requirements Modeling

Requirements engineering is one of the most important disciplines in the system lifecycle and when done well it will set the foundation for a successful architecture or program of work, ultimately ensuring that great value is delivered to the users and other stakeholders.

## Stakeholder Requirements










This diagram shows a number of stakeholders and their needs (requirements). A stereotype has been created for the stakeholders, that has an alternate image assigned to it. The requirements are displayed using a rectangular presentation style, so as to display the stereotype <<stakeholder requirement>> in the diagram.



and capabilities to detailed functional and non-functional requirements. Requirements can be prioritized, traced and tracked, and changes can be recorded, baselined, versioned and audited. Analysts can work together in a collaborative platform with role based Security, **Discussions**, **Team Library**, **Model Mail** and a range of other tools to encourage best practice and productivity.

# Documenting an Enterprise Architecture

The Documentation features can be used to automatically generate a wide range of documentation directly from the models. These can be document-based such as PDF and Docx format or HTML-based. Flexible templates can be used to completely tailor the documents that are generated including company logos, tables of content, tables of element information and diagrams. The success of an architecture and ultimately the entire Architecture Program will depend on how well communication with stakeholders is managed. Many stakeholders will be content to view architecture information including lists, diagrams and matrices directly in the repository but others will want electronic or printed documentation delivered to them. The documentation generator can be used to create high quality corporate publications automatically from the repository. This includes a wide range of standard publications such as the Architecture Vision, Compliance Assessment, Communication Plan and more. Ad-hoc reports can also be created from a number of tools such as the Glossary and the Search Window.

Package:	<input type="text" value="Functional Requirements"/>	
Output to File:	<input type="text" value="C:\Users\Public\Documents\Functional Requirements.pdf"/>	
Template:	<input type="text" value="Model Report"/>	
Output Format:	<input type="text" value="Portable Document Format (PDF)"/>	
Cover Page:	<input type="text" value="Portrait"/>	
Table of Contents:	<input type="text" value="Portrait"/>	
Stylesheet:	<input type="text" value="Help Style Sheet"/>	
Diagram Theme:	<input type="text" value="Ice - Fine"/>	
Watermark:	<input type="text"/>	

# Architecture Description

The Architecture Description describes the central aspects of the architecture including the business drivers and goals that led to the architecture being requisitioned, the requirements and constraints, the domain architectures including baseline and target architectures and any required transition architectures. The document will typically be viewed by a number of quite diverse stakeholders and so should be created with these audiences in mind. Senior management such as executive and board level stakeholders will be keen to get an overview of the architecture and how it will contribute to business success and flexibility.

Learn More: [Documentation](#)

# Architecture Requirements Specification

This document describes the Architecture Requirements of the system, its behavior under defined conditions, and the constraints that it must operate under; it will typically be read by a variety of stakeholders. There is a built-in Requirements template that can be used to generate the document, although the modeler is free to create a new template that could be either based on this or created from a blank template. When the document holds content from a variety of locations in the **Browser window**, it would be most expedient to use the **Virtual Documents** facility. This allows a user to create a model of the document (similar to a Master document in a Word Processor) which consists of a number of sections called Model Documents. The Model Documents in turn can have content cherry-picked from anywhere in the Browser window.

## Software Requirements Specification

Online Bookstore

Version 1.0 • Proposed

Learn More: [Documentation](#)



# Architecture Vision

The Architecture Vision provides an overview and guide that describes how the enterprise will be transformed at all levels by the proposed architecture. It provides a high level view of the problem or opportunity, and a description of the way that the architecture will address the requirements to achieve the desired outcome. It will typically describe an overview or outline of the Business, Information, Application, Technology and other relevant architectures, describing their baseline and target states. It can best be seen as a precursor document to the much more complete Architecture Description, but should continue to be valid and relevant throughout the lifecycle of architecture development. The Architecture Vision can be generated automatically from the repository content, including model content such as Principles, Constraints, Requirements and high level diagrams that describe the solution.

Learn More: [Documentation](#)



# Communication Plan

A Communication Plan is a critical document that describes the approach that will be taken to ensure that architectural stakeholders are informed about the architectures that are being developed and implemented and how they will affect them. Stakeholder involvement and commitment to the development and implementation of architectures is unquestionably one of the most important factors contributing to the success or failure of a specific architecture, or more generally the entire architecture program.

Enterprise Architect has a significant role to play in this communication, allowing access to be given to the stakeholders. Views, workspaces and sets of lists, diagrams and matrices can be provided that are relevant to them. The Communication Plan can be created using a Document Artifact, which allows a formatted document to be created and edited directly inside the repository. This document should list the individual and stakeholder group's needs, and articulate the mechanisms that will be used to communicate the architecture to them. The document would typically describe the number and combination of the powerful presentation and visualization techniques and tools to be used, including Roadmaps, Heat Maps, Dashboard Charts, Diagrams, Documentation, Relationship and Gap Analysis Matrices, Visual Filters, **Baselines** and more.



Learn More: [Document Artifact](#)

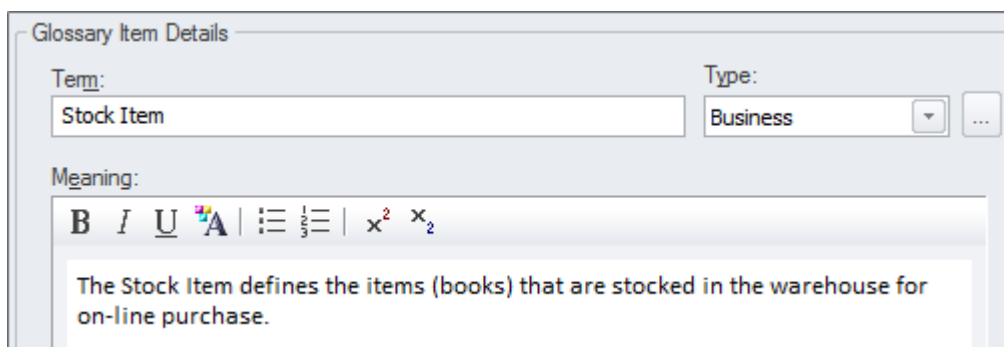
# Compliance Assessment

A Compliance Assessment is created by the architecture team and is used to monitor the way the architecture is realized by the implementation teams. These teams are typically under pressure to complete their project work and achieve the project's desired outcomes, which often means that corners are cut and rules are bent. An architect needs to work with the team and understand these pressures, but also needs to have a governance role that can assess whether the fundamental tenets of the Architecture are being implemented. A Compliance Assessment is a useful mechanism for documenting the observations made about the implementation team's compliance with the architecture. It is a way of formally recording the findings of an architectural review of the implementation initiatives, to ensure that they are implementing the architecture as it was specified in the architecture vision and in the definitions of the architecture in the models describing the Business, Information, Application and Technology Architectures. There are three ways the assessment could be produced in Enterprise Architect: using a Document Artifact, the **Team Library** facility or an Artifact.

Learn More: [Documentation](#)

# Project Glossary

A **Project Glossary** lists and defines the terms that are important for a project or program of work. The Project Glossary can be generated as an isolated document or it can be included as a section in one or more other documents. It provides a single point of truth for the important project terms and their meanings; when new documentation is generated the terms will automatically be updated. The Glossary can be generated to a Docx or PDF format, or to HTML that could be included in a project or organization level web site. The Glossary allows the modeler to categorize the terms into user defined Types, which can have styles applied when they are generated in documentation. An alternative way of using the Glossary is to define the important terms in the domain of the Enterprise in a Business **Object Model** and only list project-related terms in the Glossary.



Glossary Item Details

Term:  Type:

Meaning:

**B** *I* U A |  $\frac{1}{2}$   $\frac{1}{3}$  |  $x^2$   $x_2$

The Stock Item defines the items (books) that are stocked in the warehouse for on-line purchase.

Learn More: [Glossary](#)

# Enterprise Architecture Techniques

Enterprise Architect supports a very large number of techniques that can be used with the discipline of Enterprise Architecture. The techniques are supported by an even larger set of tools that will assist the Architect when performing these techniques. The techniques range from the strategic level down to the implementation level and include items such as Balanced Scorecard, Roadmaps, Gap Analysis, Principles Management, Pattern Analysis, Stakeholder Analysis and more. All of these techniques are discussed in the set of topics within this section.

The related tools are discussed in the topics in the [Additional Enterprise Architecture Tools](#) section.

# Architecture Governance

Architecture Governance is concerned with the management and control of the architectural practices and the architectures that are created. It should be viewed in the context of the wider governance models of the enterprise and the more detailed governance models of business and technical implementations. These governance structures are typically hierarchical and can be distributed across geographies and lines of business. The concept of governance is more about guidance and ensuring that good practice is followed and that the architectural processes are appropriate and repeatable, ultimately to ensure value is delivered to the stakeholders.

Enterprise Architect has a number of facilities that can assist with Architectural governance, including the governance of implementation projects. These include an Organizational Chart that can be used to model the structure and the relationships of the governance roles. A Document Artifact can be used as the Governance Register and Process Models can be used to model the Governance processes. The **Calendar** can be used to schedule important events in the governance lifecycle.

This section lists the main tools available in Enterprise Architect that can be used to perform Architecture Governance. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools

is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## Organizational Chart

An Organizational Chart is part of the Strategic Modeling extension that can be used to model the structure and the relationships of the governance roles and to make this explicit to the Architecture Team. These roles can be reused when defining the Governance Process by applying them as an Instance Classifier for the Pools and Lanes in a Process Diagram used to describes the Governance Process. Relationships in addition to the structural connections can be added to show important political or historical associations between team members.

Learn More: [Organizational Chart](#)

## Document Artifact

A Document Artifact is the perfect tool to hold the Governance Register allowing it to be accessed inside the model and close to the Architectures and elements, diagrams, matrices and list that describe them. The Document Artifact is a word processor file that is stored inside the repository and can be structured with headings, sections and tables to store the details of the important events and decisions that are made by the architecture

executive. Links to a variety of model content can be added to the document ensuring that the log is relevant and its application to architectural content is made explicit.

Learn More: [Document Artifact](#)

## Business Process Diagram

The Business Process diagram can be used to define and communicate the governance process to members of the Architecture team to ensure they all understand the sequence of activities that make up the process. Roles can be indicated by the use of Pools and Lanes, and Data Objects and Data Stores can be used to represent inputs and outputs of the process, including when the Governance Register is updated.

Learn More: [Business Process Diagram](#)

## Calendar

Then **Calendar** is a useful tool for recording important events in the governance lifecycle such as meetings, reviews and milestones. The Calendar is visible to all members of the Architecture team and if the solution architectures are located in the same repository it will also be visible to the solution teams. Event types and categories can be defined and recurring events can be set up for events such as the



Architecture Steering Committee meetings.

Learn More: [Calendar](#)

## Model Mail

**Model Mail** is a mail system that exists entirely inside the model and is available to all users. It can be used to exchange mail messages with any users or groups of users in the team, and has the added advantage of being able to attach links to model content, including diagrams, elements, operations, attributes, matrices, images and more.

Learn More: [Model Mail](#)

## Team Library

A critical part of the governance of the Architectural repository is the review process. The **Team Library** facility is a powerful in-model tool for managing all aspects of architectural reviews. The reviews can be held at any stage, including at the time the program is set up or during the creation and maintenance of architectures and other ancillary content, such as principles, standards and references. It is quite common for organizations to store these reviews separately to the repository, but the compelling thing about using the powerful Team Library facility is that the reviews are housed inside the repository

and there is the ability to relate parts of the repository - such as diagram elements - as hyperlinks to the review.

Learn More: [Team Library](#)

# Architecture Requirements Management

Architecture Requirements Management is critical to the creation of architectures that are fit for purpose and that address the concerns, drivers, constraints, goals and other forces that express the needs of the enterprise or its organizations. These requirements effectively define the design problem that the architecture needs to solve. Each architecture domain will have requirements defined that will be the result of architectural analysis of the high level domains. The requirements will have a variety of sources, and requirements need to reference these sources or the more primitive elements from which they are derived.

Enterprise Architect is a sophisticated platform that can be used for architectural Requirements development and management. The powerful text-based **Specification Manager** helps you create, view and manage Requirements in a word processor or spreadsheet-like interface that provides a compelling alternative to viewing requirements in the **Browser window** or in diagrams. Any number of Requirements diagrams can be created and used to visually represent the relationship between requirements and other elements, including other requirements. **Baselines** provide a snapshot of the requirements and then, as time goes on, you can compare the current model to this or any other baseline and display the differences visually. If a publication such as a Request for Tender or a requirements specification is

needed, it can be conveniently generated automatically using the Documentation engine.

This section lists the main tools available in Enterprise Architect that can be used to perform Architecture Requirements Management. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

Item

## 1 REQ019 - Manage Inventory

The system **MUST** include a complete inventory management facility to store and track stock of books for the on-line bookstore.

### 1.1 REQ122 - Inventory Reports

Inventory reports are required that detail the available stock for each item including back orders. Future stock level reports should be able to predict the quantity of stock at a specified future date.

### 1.2 REQ023 - Store and Manage Books

A book storage and management facility will be required.

#### 1.2.1 REQ022 - Order Books

A book order facility will be required to allow on-line ordering from major stockist's.

#### 1.2.2 REQ021 - List Stock Levels

A facility will exist to list current stock levels and to manually update stock quantities if physical checking reveals inconsistencies.

## Specification Manager

The **Specification Manager** is the central tool for working with Requirements; it provides a Word Processor or Spreadsheet interface for entering, maintaining and viewing Requirements. Architecture Requirements can be viewed as a catalog and detailed descriptions and a range of properties can be added directly through the interface. Changing Requirements in the Specification Manager will change them in all other places in the repository, such as diagrams and windows. The Specification Manager can be used with any set of model elements such as business drivers or applications where viewing the elements in a list is preferred, but it is particularly useful for creating and managing Architectural Requirements.

Learn More: [Specification Manager](#)

## Requirements Diagram

The Requirements diagram can be used to create a visual representation of the relationship between requirements and other elements in the model, including Principles, Business Drivers, Constraints, Business Rules, Use Cases, User Stories, design Components and more. For architects who are accustomed to working with requirements in a text based tool it will provide a powerful and expressive visual representation allowing the requirements to be tied to elements of the business, information, application and technology architectures.

Learn More: [Requirements Diagram](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. Architectural Requirements stored in the repository can be conveniently generated and included in documentation. The formatting in an element's Notes and diagrams can also be carried through to the documentation. A sophisticated template facility exists that provides a range of built-in templates and also allows the user to create their own templates, defining styles, cover pages, tables of contents, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Requirement Properties

Requirement properties are metadata about each Requirement. They allow a series of aspects of the Requirement to be specified, which are used to manage the Requirements as a set. Properties such as Status, Priority, Difficulty, Author, Phase and Version can be used to prioritize and group Requirements for the purpose of defining work Packages, releases and architecture work

allocation.

Learn More: [Requirement Properties](#)

## Traceability Window

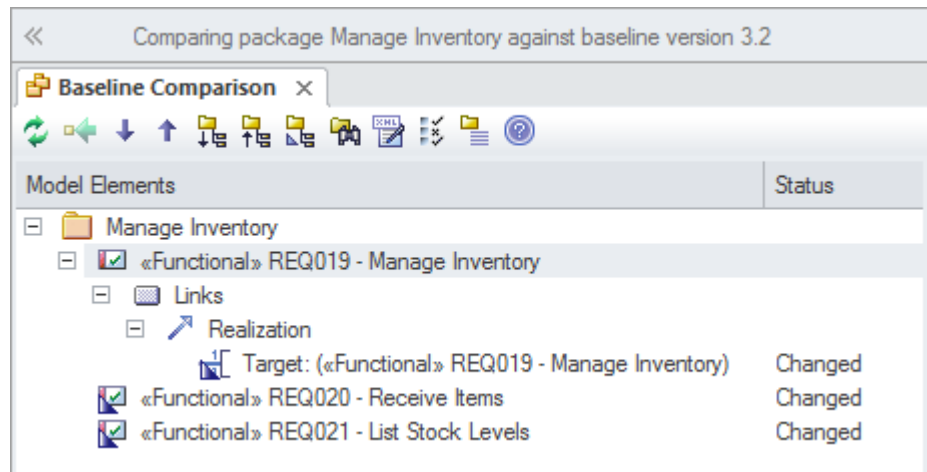
The Traceability Window displays a dynamic hierarchical view of a model element's relationships to other elements in the model. An architect will find the tool indispensable for examining how requirements trace to items higher up in the architecture such as Business Drivers and lower level items such as designs and solution options.

Learn More: [Traceability Window](#)

## Baselines

The Baseline Tool can capture a snapshot of the Architectural Requirements at a point in time and a comparison can be made between this or any other snapshot and the model at a later point in time. The comparison tool will enable an architect to visualize what has changed and if necessary revert back to content contained in the baseline. Any Package in the requirements hierarchy can be baselined and any number of baselines can be created.

Learn More: [Baseline Tool](#)



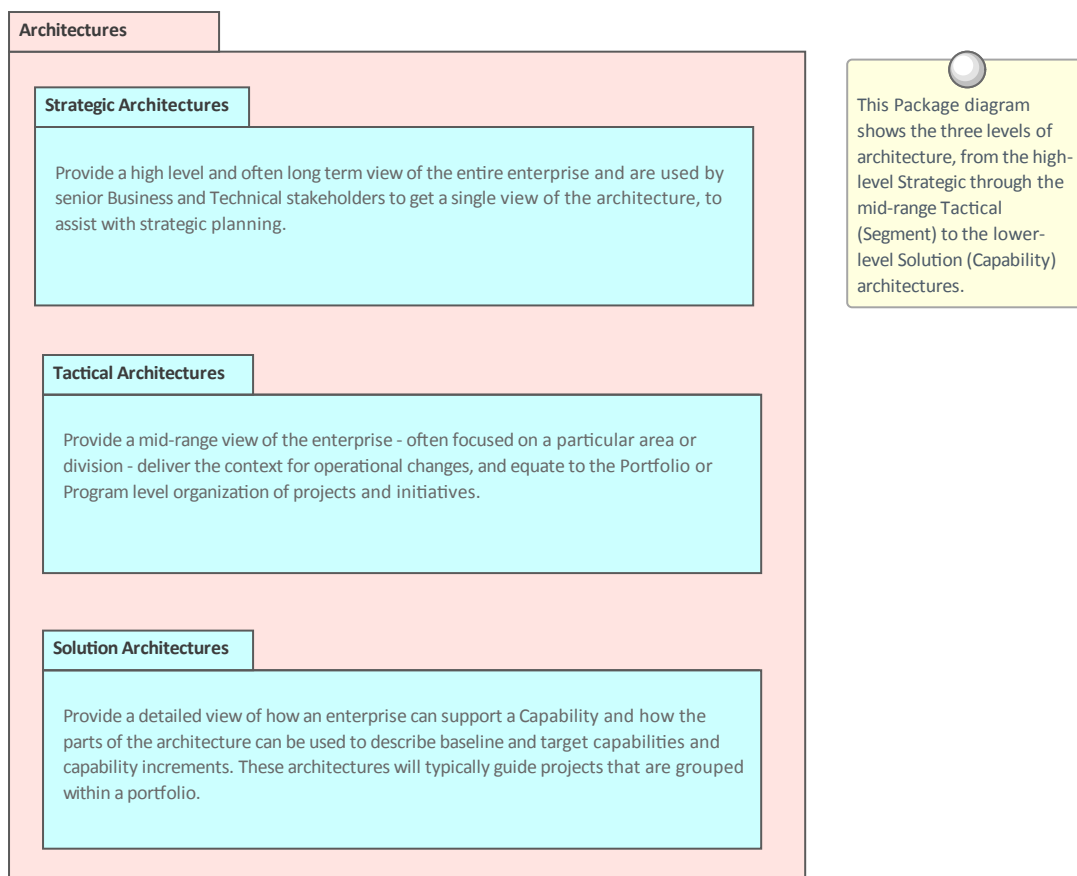


# Architecture Partitioning

An enterprise of any appreciable size will typically have a number of architects working at different levels and on different Strategic, Tactical and Solution architectures. The architects will be part of programs of work and projects, which could cut across the other architecture levels and groupings. To ensure that the architects' work is not duplicated or in conflict, Architecture Partitioning is required. This forms part of Architecture Governance that attempts to ensure that the architecture resources are working efficiently and effectively, and that the architectures can be integrated to create the best result for the Enterprise.

Enterprise Architect allows all architecture work at an enterprise, segment and solution level to be housed in the one repository, which facilitates the governance and integration of the architectures. The **Browser window** can be set up to ensure that enterprise level notions can be modeled and reused across any number of separate architectures, which helps to ensure that the architectures can be aligned and integrated. **Model Views** are useful to create views of cross cutting concerns such as reuse or shared architectural representations or stakeholders' views. A Roadmap diagram could be used to represent the **duration** and time sequencing of architectures, which will help manage the dependencies. Organization Charts can be used to understand the way teams and resources are allocated to architecture work.

This section lists the main tools available in Enterprise Architect that can be used to perform Architecture Partitioning. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Model Views

**Model Views** is a facility that allows a modeler to create lists of elements in alternative views to the way they are organized in the **Browser window**. This mechanism is

particularly useful for Architectural Partitioning where it might be expedient to create lists of cross-cutting elements and architectures independent of the structure defined in the Browser window. Any number of Favorite folders can be created and grouped into one or more higher level folders. This provides a mechanism for manually defining lists of elements by simply dragging elements from the Browser window into a Favorites folder and ordering them using the up and down arrow keys in the toolbar. Alternatively, elements can be listed using a built-in or user defined search that can be used to select elements from anywhere in the repository based on properties such as Complexity, Version and Phase, or user defined properties expressed in Tagged Values.

Learn More: [Model Views](#)

## Roadmap Diagrams

The Roadmap is an overlay that can be applied to any diagram and that can be conveniently enabled or disabled. The overlay presents a timeline that is used to indicate the starting time, **duration** and finishing time of important phases in the lifetime of the elements on the diagram. It can be used to show the duration of each of the architectural initiatives that will assist with understanding the dependencies as part of Architecture Partitioning. Any diagram object can appear on a Roadmap diagram, including Packages that could be used as the organizing

element for an Architecture. The appearance and properties of the timeline can be altered to set the Units, Tick Spacing (Year, Month and so on), Start and Finish time, Colors, Fonts and more. The phases on the elements can be configured in a Diagram Legend, allowing names and colors to be defined dynamically.

Learn More: [Roadmap Diagram](#)

## Organizational Chart

The organization chart is useful for understanding the makeup of the architectural team, the reporting lines and the structure of the governance bodies. This will help ensure that the governance side of the architectural Partitioning is being managed and that there is only a single team assigned to an architecture and to get an understanding of the distribution of the work load and the team make-up and the boundaries between architecture work.

Learn More: [Organizational Chart](#)

## Browser Window

The **Browser window** can be used to create a series of Packages that can form the structural basis for Architecture Partitioning. The placeholders can be created for Strategic, Tactical and Solution architectures, and within these

Packages the architectures themselves can be placed. The arrangement of the Packages will be determined by an individual architecture program.

Learn More: [Browser Window](#)

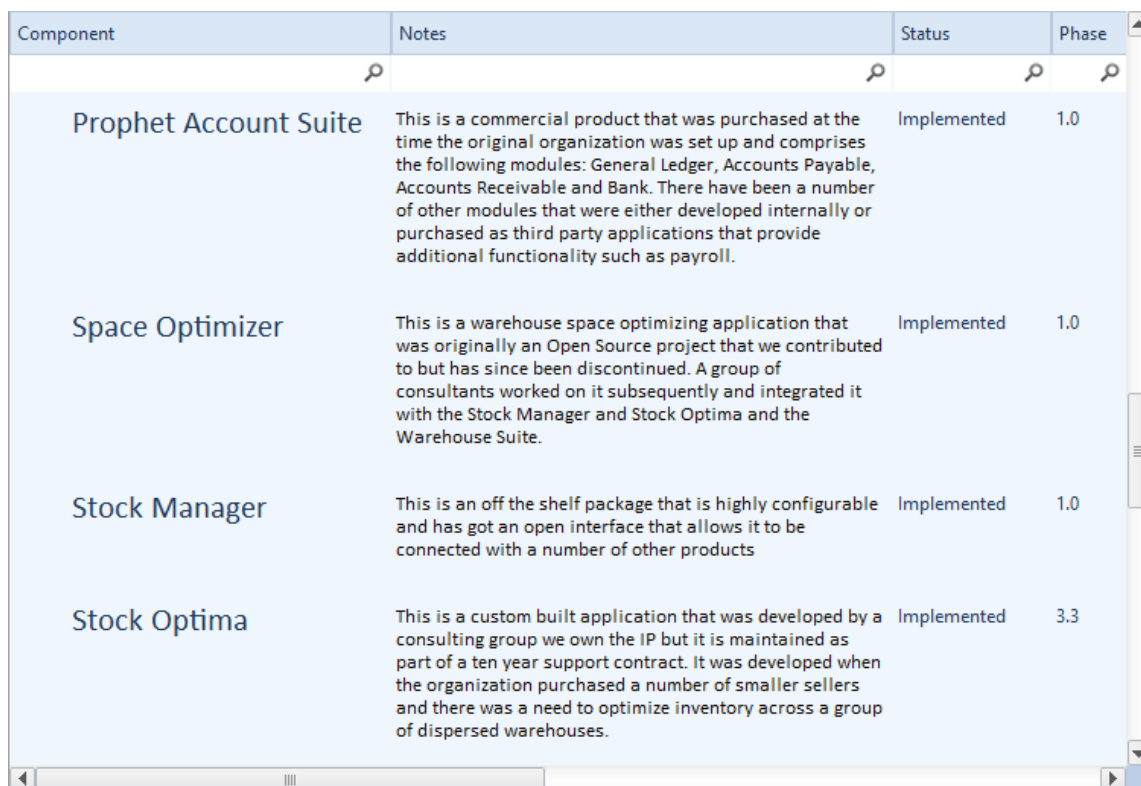
## Traceability Window

The Traceability Window can be used to view the relationships between the architectures and the elements that make up those architectures. This is a useful mechanism to show the dependencies between parts of the architectures and to show where there are opportunities to reuse a part of one architecture in another architecture or to partition out a separate architecture. These relationships are visible through the **Traceability window**.

Learn More: [Traceability Window](#)

# Application Portfolio Management

Application Portfolio Management is used to create and maintain a catalog of Applications and Application Services.

A screenshot of a software application window displaying a table of application components. The table has four columns: Component, Notes, Status, and Phase. There are four rows of data, each representing a different application suite. The interface includes a search icon in the top right of the table area and a scroll bar on the right side.

Component	Notes	Status	Phase
Prophet Account Suite	This is a commercial product that was purchased at the time the original organization was set up and comprises the following modules: General Ledger, Accounts Payable, Accounts Receivable and Bank. There have been a number of other modules that were either developed internally or purchased as third party applications that provide additional functionality such as payroll.	Implemented	1.0
Space Optimizer	This is a warehouse space optimizing application that was originally an Open Source project that we contributed to but has since been discontinued. A group of consultants worked on it subsequently and integrated it with the Stock Manager and Stock Optima and the Warehouse Suite.	Implemented	1.0
Stock Manager	This is an off the shelf package that is highly configurable and has got an open interface that allows it to be connected with a number of other products	Implemented	1.0
Stock Optima	This is a custom built application that was developed by a consulting group we own the IP but it is maintained as part of a ten year support contract. It was developed when the organization purchased a number of smaller sellers and there was a need to optimize inventory across a group of dispersed warehouses.	Implemented	3.3

This section lists the main tools available in Enterprise Architect that can be used to perform Application Portfolio Management. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## Import and Export Spreadsheets

It is common for an analyst to have started modeling Applications - including Interfaces - in a Spreadsheet, or to want to manipulate existing elements in a Spreadsheet. Enterprise Architect has a flexible and configurable tool for importing and exporting elements from a CSV file, which can be imported and exported from a Spreadsheet. Any type of element can be imported or exported to the Spreadsheet file, but it is particularly common to use the facility with Applications, including Interfaces. Names, descriptions, built-in properties, and extended properties in the form of Tagged Values can be imported or exported. The tool provides a flexible Specification window where the mapping between element properties and the columns in the Spreadsheet and other parameters can be defined and saved. Essentially the columns of the spreadsheet define the properties of the Applications or Interfaces, and each Application or Interface is specified in a row.

Learn More: [Import and Export Spreadsheets](#)

## Model Views

**Model Views** is a facility that allows a modeler to create lists of elements, including Applications and Interfaces, in alternative views to the way they are organized in the **Browser window**. Any number of Favorite folders can be created and grouped into one or more higher level folders. This provides a mechanism for manually defining lists of



Applications by simply dragging elements from the Browser window into a Favorites folder, and ordering them using the up arrow key and down arrow key in the toolbar.

Alternatively, elements can be listed using a built-in or user defined search that can select elements from anywhere in the repository based on properties such as Complexity, Version and Phase, or user defined properties defined in Tagged Values.

Learn More: [Model Views](#)

## Component Diagram

A Component diagram can be used to describe any number of the Applications that make up the Application Portfolio Catalog. Expressive diagrams can be created and viewed that show the relationships between Components including Interfaces and Ports that describe the services the application offers to its environment, including other applications. Any number of diagrams can be created showing groups of applications that collaborate to provide a service that is ultimately required to meet a business service or capability. An enterprise of any appreciable size will typically have hundreds if not thousands of applications and their relationships will be complex and varied. It is possible to create a large diagram that shows all these applications in one view but it is more common to divide the portfolio up into a number of groups of applications. Both logical and physical application components can be described.



Learn More: [Component Diagram](#)

## Specification Manager

The **Specification Manager** is a powerful tool for working with lists or catalogs of elements, including applications; it provides a word processor or spreadsheet interface for entering, maintaining and viewing Applications.

Architecture applications can be viewed as a catalog, and detailed descriptions and a range of properties can be added directly through the interface. Changing an application's details in the Specification Manager will change them in all other places in the repository, such as Component diagrams and windows. The Specification Manager can be used with any element or group of elements, and shields a user from needing to know about the underlying representation as they are simply working with a list as they would in their favorite spreadsheet or word processor tool.

Learn More: [Specification Manager](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. An application or application service portfolio stored in the

repository can be conveniently generated and included in documentation. The formatting in an element's Notes and diagrams can also be carried through to the documentation. A sophisticated template facility exists that provides a range of built-in templates and also allows the user to create their own templates, defining styles, cover pages, tables of contents, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Browser Window

Applications and Application Services can be created and added to a catalog directly in the **Browser window**, without the need to create the elements on a diagram. Elements created on a diagram will always be visible in the Browser window and changing them in one location will automatically update them everywhere in the repository. The Applications can be ordered in the Browser window using the **Up Arrow** button or **Down Arrow button** in the Browser window header. They can also be grouped into categories by the use of Packages, to make the list more manageable.

Learn More: [The Browser Window](#)

## Stereotyping

Stereotypes can be used to create additional types that can further classify Applications by types or category, such as 'Logical Application Component'. Stereotyping is a powerful mechanism but it should be used sparingly as each stereotype practically adds to the grammar of the language and makes the model more difficult for outsiders - such as implementation partners - to understand. Stereotypes can also be created as part of the powerful prototyping facility, which allows Tagged Values to be added when an element with a stereotype is created.

Learn More: [Stereotyping](#)

# Balanced Scorecard

Balanced Scorecard is a strategic planning and monitoring mechanism to ensure that business activity aligns with strategic goals and - where it is found not to align - to adjust the business activities in an attempt to improve performance. The idea was originally expounded by Robert Kaplan and David Norton as a performance measurement framework that created a more 'balanced' view of organizational performance by adding a number of non-financial performance measures to traditional financial metrics. It was discovered that just focusing on financial measures was not sufficient for modern enterprises in the information age and so three additional measures were added. The Balanced Scorecard suggests that an organization should be viewed from these four perspectives:

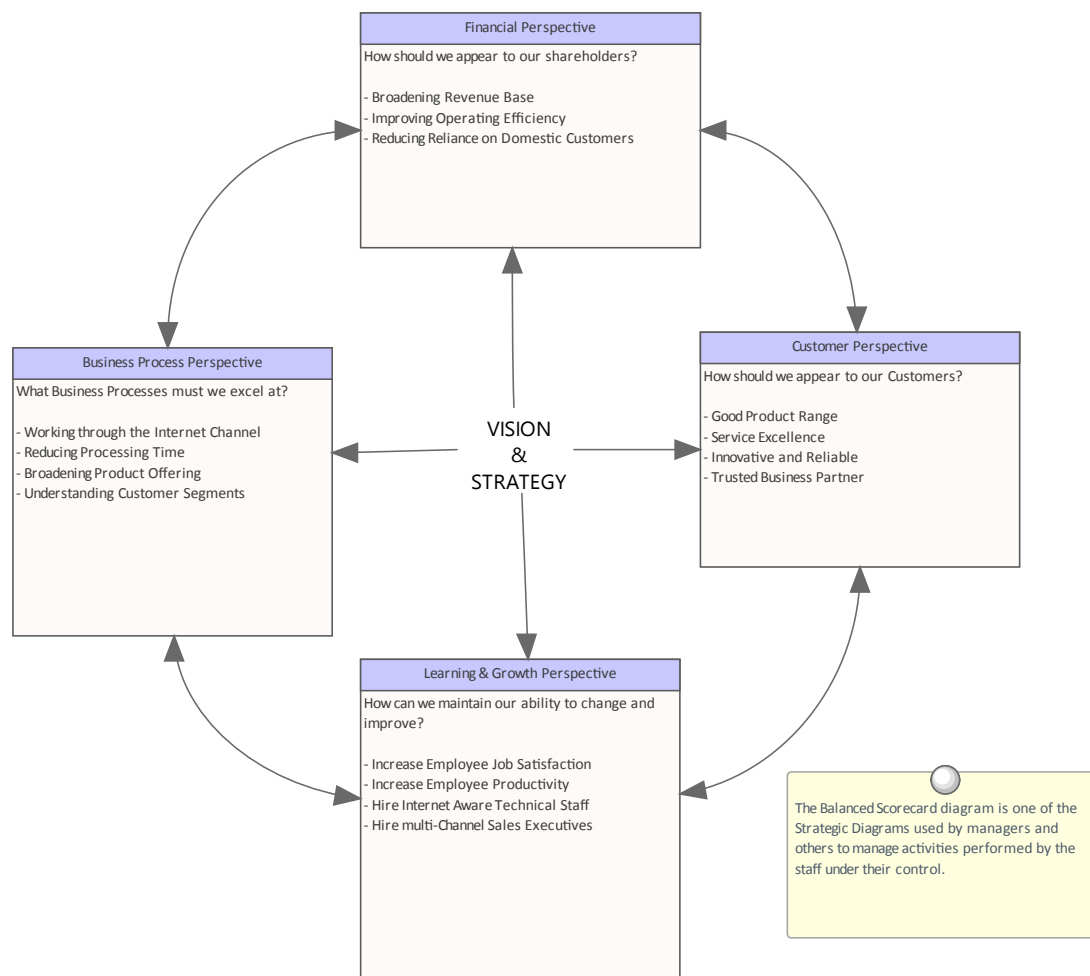
- The Learning & Growth Perspective
- The Business Process Perspective
- The Customer Perspective
- The Financial Perspective

Metrics should be defined for each perspective, and data collected and analyzed on a regular and ongoing basis to provide the information required by managers to intervene by adjusting business activity in an attempt to increase performance.

Enterprise Architect has support for the Balanced Scorecard with its Strategic Modeling extension, which allows a modeler to create a wide range of strategic diagrams

including the Balanced Scorecard. The Strategic Modeling extension can be enabled by using the **MDG Technologies window**.

This section lists the main tools available in Enterprise Architect that can be used to create and manage a Balanced Scorecard. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the Help topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Balanced Scorecard Diagram

A Balanced Scorecard diagram can be created at any level, from an organization level through to an initiative (project) level. Any number of diagrams can be created within a project. The diagrams and the elements they contain can be placed into a Package at appropriate levels in the **Browser window**. The Toolbox page for the diagram comes with a time-saving Balanced Scorecard Pattern that can be used to create the elements and the diagram, ready for the analyst to complete. All four dimensions are supported, and Objectives, Measures, Targets and Initiatives can be added as text and formatted as required.

Learn More: [Balanced Scorecard Diagram](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce DOCX and RTF Word Processor files, PDF and HTML reports. Any Balanced Scorecard stored in the repository can be conveniently generated and included in documentation. The formatting used to describe the Objectives, Measures, Targets and Initiatives can also be carried through to the documentation. A sophisticated template facility exists that provides a range of in-built templates and allows the user to create their own templates, defining styles, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Baselines

The Baseline Tool is used as a type of backup and recovery tool and can capture a snapshot of the Balanced Scorecard at a point in time and then at a later time the repository can be compared to this (or another baseline) for the purpose of determining what has changed. It is possible to revert the current model back to a state captured in the baseline at the level of a granular change.

Learn More: [Baselines](#)

# Baselines and Versioning

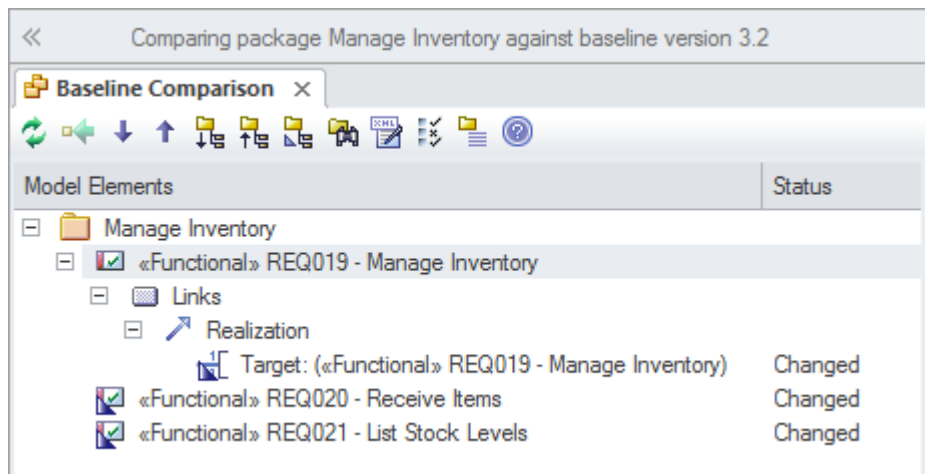
**Baselines and Versioning** is an important technique with any model but is particularly important with Enterprise Architecture. A model is used as a representation of reality and attempts to describe the things of the enterprise in a simple, coherent, clear and correct way. The ideas and notions described in the model are likely to change as the enterprise changes and it is important at times to keep a history or memory of these changes. This is particularly important when an idea is explored but subsequently found not to be worth pursuing. With a baseline or version the changes can be rolled back and the model reverted to its previous state.

Enterprise Architect has a number of tools that can assist with Baselines and Versioning, including the Baseline tool, which can take any number of snapshots of Packages at a point in time, and the **Version Control** facility that allows any number of Packages to be versioned and keeps a complete atomic change history. There is also a built-in feature to allow any Package (including child Packages) to be exported to an XML file and re-imported at a later time.

This section lists the main tools available in Enterprise Architect that can be used to perform Baselines and Versioning. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise](#)



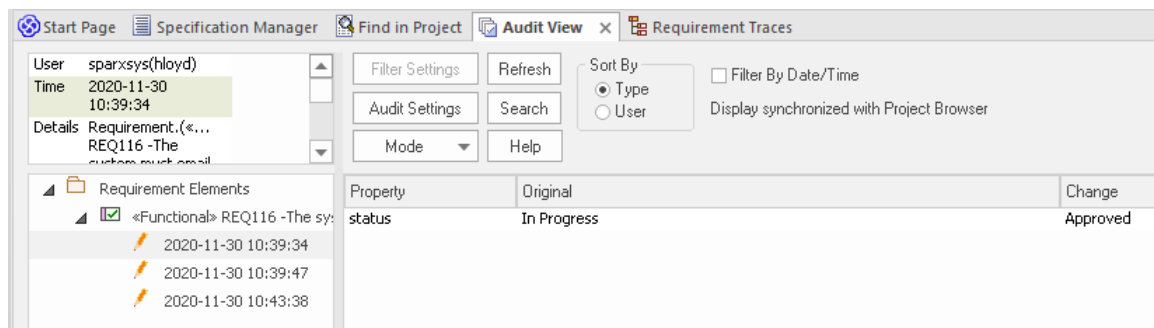
## [Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Auditing

The **Auditing** feature can keep track of the changes to architecture content including what was changed, when it was changed and by whom. Auditing is by default disabled and must be enabled before the changes to architecture elements will be recorded. Once enabled it is a passive tool that silently records the changes to elements. It does not replace **Version Control** or baselines and in contradistinction to these tools it can not be used to return to a previous state of the model. Change management, Architecture Governance and quality control are all aided by the use of Auditing.

Learn More: [Auditing](#)



## Baselines

The Baseline Tool is used as a type of backup and recovery tool and can capture a snapshot of any Package containing architectural content at a point in time. At a later time the repository can be compared to this (or another baseline) for the purpose of determining what has changed. It is possible to roll the current model back to a state captured in the baseline at the level of a granular change. This is a powerful mechanism that can be built into the Architectural Process and conducted at milestones during the creation and management of architectures.

Learn More: [Baseline Tool](#)

## Version Control

The **Version Control** facility allows a history of changes to be captured and, if required, rolled back to. The tool requires an external Version Control server to be set up, and Enterprise Architect communicates with this service to

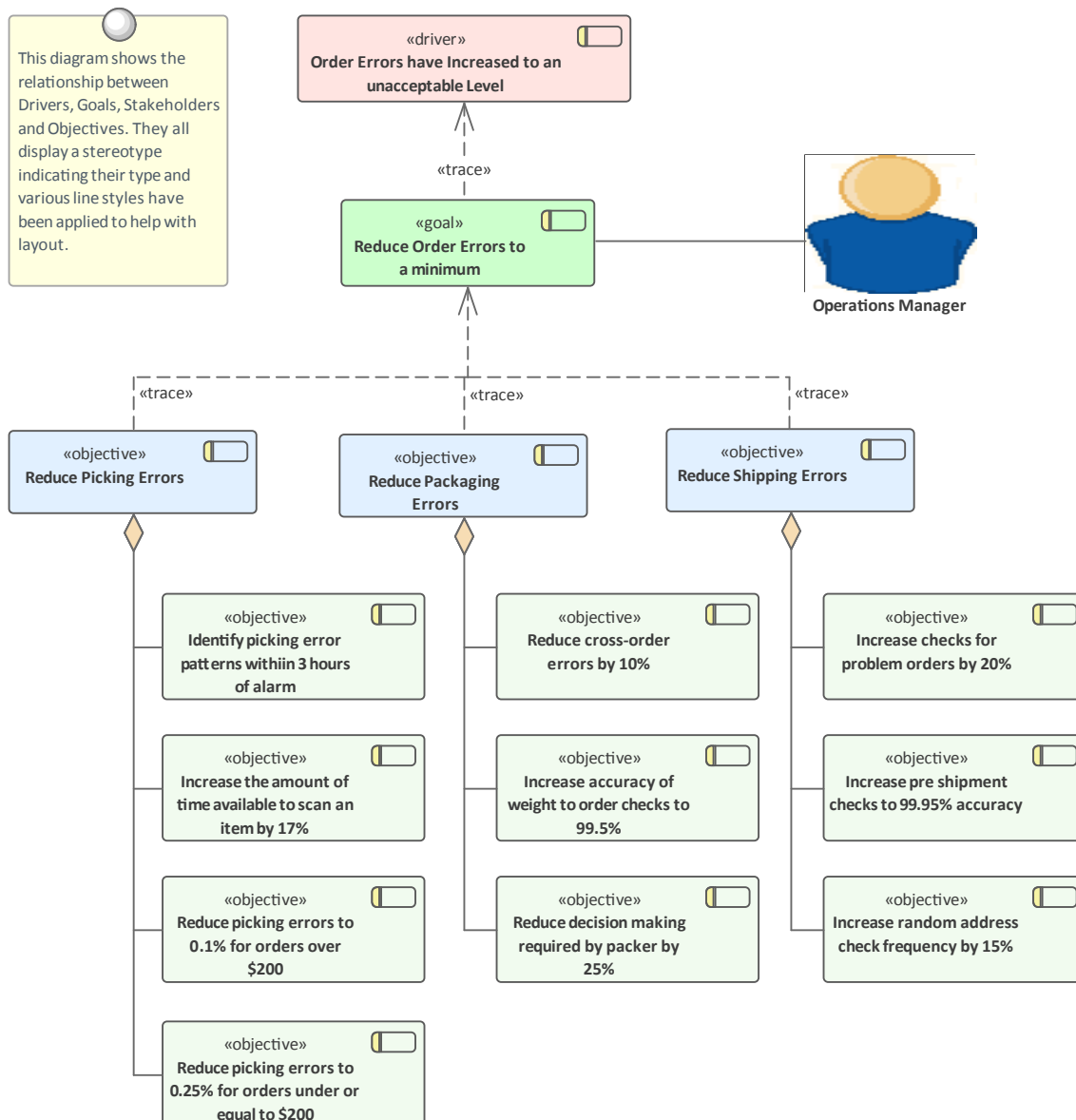
allow check-out and check-in of content. The unit of Version Control is the Package, and any number of Packages can be versioned, also using multiple different configurations and Version Control servers. The tool does not use optimistic locking; once a Package is checked-out by one user it is locked for update by other users until it is checked back in. Version Control is a powerful tool but requires discipline and good practice in model management. Learn More: [Version Control](#)

# Business Goals and Objectives Modeling

Business Goals and Objectives Modeling is arguably the most important input to the development of an architecture, as it sets the right context for the alignment of architecture with the business strategy.

Enterprise Architect can be used to model the Goals and Objectives and relate them to each other, with each Goal being decomposed into a number of Objectives that can be measured. Stereotypes can be used to create Goal and Objective elements, and these can be added to any number of diagrams. The **Browser window** provides a way of structuring the Goals, adding the Objectives as child elements placed under the applicable Goal. The **Specification Manager** is a useful tool for creating, viewing or presenting the Goals and Objectives in a workshop. It is a text based tool and allows the Goals and Objectives to be viewed in a word processor or spreadsheet interface.

This section lists the main tools available in Enterprise Architect that can be used to perform Business Goals and Objectives Modeling. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



elements such as Business Drivers can be viewed as a catalog and detailed descriptions and a range of properties can be added directly through the interface. Changing these elements in the Specification Manager will change them in all other places in the repository, such as diagrams and windows. The Specification Manager can be used with any set of model elements such as Requirements, and applications where viewing the elements in a list is preferred, but it is particularly useful for creating and managing content in the Business Architecture where users might not be as familiar with diagrammatic representations. Learn More: [Specification Manager](#)

## Browser Window

Business Goals and Objectives can be created and added to a catalog directly in the **Browser window**, without the need to create the elements on a diagram. Elements created on a diagram will always be visible in the Browser window, and changing them in one location will automatically update them everywhere in the repository. The elements can be ordered in the Browser window using the **Up Arrow** button or **Down Arrow button** in the Browser window header. They can also be grouped into categories by the use of Packages, to make the list more manageable.

Learn More: [The Browser Window](#)

## Requirements Diagram

The Requirements diagram can be used to create a visual representation of the relationship between Goals and Objectives and other elements in the model, including Principles, Business Drivers, Constraints, Business Rules, Use Cases, User Stories, design Components and more. For architects who are accustomed to working with lists in a text based tool, the Requirements diagrams will provide a powerful and expressive visual representation, allowing the Goals and Objectives to be tied to elements of the business, information, application and technology architectures.

Learn More: [Requirements Diagram](#)

## Stereotyping

The Unified Modeling Language does not itself contain elements to represent Goals and Objectives, but they can be included by using stereotypes, which is a mechanism for extending the core language. These stereotypes can be applied to a base element such as a Requirement and then the stereotype can be applied to create new Goals and Objectives. The stereotype will be visible in the **Browser window** and diagrams if these options have been set. These stereotyped elements can become available if other

Technologies or profiles have been enabled.

Learn More: [Stereotyping](#)

## Traceability Window

The Traceability Window allows a modeler to visualize the connections between elements in the repository. This is very useful to show how Goals and Objects relate to each other, how they are connected to other elements in the model and, in turn, how those elements are connected. Business Goals and Objectives can be related to a wide range of elements, including Policies, Requirements, Business Process, and more; all of these relationships would be visible through the **Traceability window**.

Learn More: [Traceability Window](#)



# Business Scenarios

Business Scenarios are a powerful technique for deriving an understanding of the significant business needs, which can then be used to determine the important requirements and ensure solutions meet the overall business needs. They help to make sure that the architectures are developed holistically, that the problem and concomitantly the solution do not become fragmented, and that architects do not lose sight of the overall business problem or opportunity being solved.

There is a wide range of tools available for creating and managing Business Scenarios in Enterprise Architect. Probably the most compelling tool is the Document Artifact, which allows a full textual description to be written defining the Business Scenario in detail. A template can be created and stored in the repository and can be reused whenever a new Scenario is created. When a more model based representation is preferred a Requirements diagram can be used to model the Business Scenarios and a Scenario can be related to Principles and Constraints, Actors, Processes and Components. A combination of the text and model based approaches also produces convincing results.

This section lists the main tools available in Enterprise Architect that can be used to develop Business Scenarios. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available

from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## Specification Manager

The **Specification Manager** is the central tool for working with Business Scenarios, providing a word processor or spreadsheet interface for entering, maintaining and viewing Requirements. Architecture requirements can be viewed as a catalog, and detailed descriptions and a range of properties can be added directly through the interface. Changing requirements in the Specification Manager will change them in all other places in the repository, such as diagrams and windows. The Specification Manager can be used with any set of model elements, such as business drivers and applications where viewing the elements in a list is preferred, but it is particularly useful for creating and managing Architectural Requirements.

Learn More: [Specification Manager](#)

## Deployment Diagram

A Deployment diagram can be used to describe infrastructure elements and services that support the problem or opportunity that has been identified in the higher level architectures. The diagram can describe virtual, logical or physical infrastructure elements and their relationships,

including Nodes (such as servers and storage area networks), Devices (such as routers and switches) and Execution Environments (such as operating systems and database management systems).

Learn More: [Deployment Diagram](#)

## Document Artifact

A Document Artifact can be used to document a Business Scenario, conveniently storing the document inside the repository. A template could be specified for the document, which could be reused for other Business Scenarios within this and other initiatives. This provides a convenient way to reference a Business Scenario as the element can be dropped from the **Browser window** onto a diagram and other elements such as derived requirements and infrastructure servers and devices or applications supporting the current architecture could be related to the Document Artifact.

Learn More: [Document Artifact](#)

## Requirements Diagram

The Requirements diagram can be used to create a visual representation of the relationship between a Business Scenario and requirements and other business architecture elements in the model, including Principles, Business

Drivers, Constraints, Business Rules, Design Components and more. For architects who are accustomed to working with requirements in a text-based tool it will provide a powerful and expressive visual representation, allowing the requirements and other elements to be visually related to the Business Scenario.

Learn More: [Requirements Diagram](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. Business Scenarios stored in the repository can be conveniently generated and included in documentation. The formatting in an element's Notes and diagrams can also be carried through to the documentation. A sophisticated template facility exists that provides a range of built-in templates and also allows the user to create their own templates, defining styles, cover pages, tables of contents, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Traceability Window

The Traceability Window displays a dynamic hierarchical

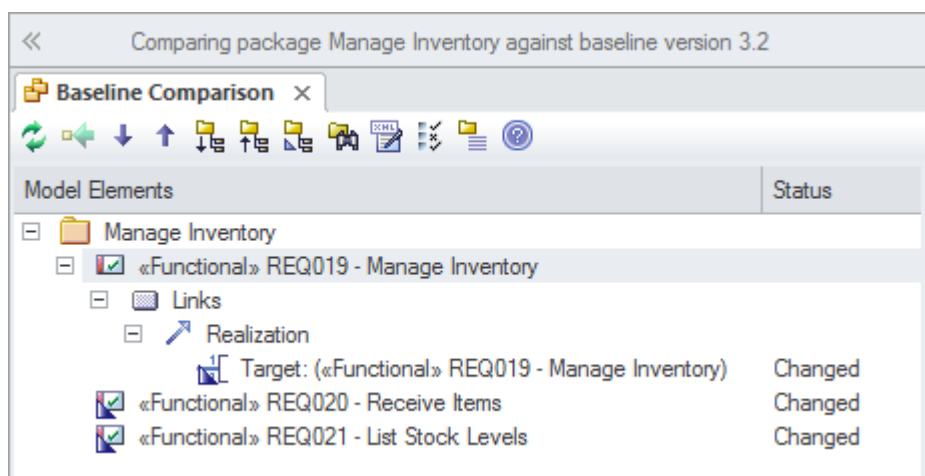
view of a model element's relationships to other elements in the model. An architect will find the tool indispensable for examining how requirements trace to items higher up in the architecture such as Business Scenarios and lower level items such as designs and solution options.

Learn More: [Traceability Window](#)

## Baselines

The Baseline Tool can capture a snapshot of the Business Scenarios at a point in time and a comparison can be made between this or any other snapshot and the model at a later point in time. The comparison tool will enable an architect to visualize what has changed and if necessary revert back to content contained in the baseline. Any Package in the Business Scenarios hierarchy can be baselined and any number of baselines can be created.

Learn More: [Baseline Tool](#)



## Component Diagram

A Component diagram can be used to describe any number of the Applications, Application Services or Interfaces that make up the Technical Reference Model. Expressive diagrams can be created and viewed that show the relationships between Components such as Interfaces and Ports that describe the services the application offers to its environment including other applications. Any number of diagrams can be created showing groups of applications that collaborate to provide a service that is ultimately required to meet a business service or capability. An enterprise of any appreciable size will typically have hundreds if not thousands of applications and their relationships will be complex and varied. It is possible to create a large diagram that shows all these applications in one view but it is more common to divide the portfolio up into a number of groups of applications. Both logical and physical application components can be described.

Learn More: [Component Diagram](#)

# Capability Based Planning

Capability Based Planning is ideally a business driven and led mechanism for creating value by planning, creating and delivering strategic business capabilities. It has become important as a way of ensuring that an organization focuses its activities on delivering business capabilities that in turn ensure business objectives and goals, and ultimately the vision, are met.

Capability Based Planning and Enterprise Architecture are well aligned because they both tend to work against the normal line-of-business management and reporting styles that exist in most organizations. Capability Planning is often difficult due to the fact that many lines of business and projects can be simultaneously involved in the delivery of the capability. The planning will typically require a capability to be delivered by a series of increments that span vertical organization structures, projects and time frames.

Enterprise Architect is ideally positioned to assist with Capability Based Planning, with its ability to create any number of views of the repository including a view that shows how a capability is implemented at business, data, application and technology levels. This view will typically cut across other line-of-business views allowing executives and managers to see both a horizontal and a vertical view of the capability.

This section lists the main tools available in Enterprise Architect that can be used to perform Capability Based

Planning. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#) Help topics.

## Stereotyping

The Unified Modeling Language does not itself contain an element to represent a Capability or Capability Increment, but they can be included by using stereotypes, a mechanism for extending the core language. These stereotypes can be applied to a base element such as a Requirement and then the stereotype can be applied to create new Capabilities or Capability Increments. The stereotype will be visible in the **Browser window** and diagrams if these options have been set. Capabilities elements can become available if other Technologies or profiles have been enabled.

Learn More: [Stereotyping](#)

## Traceability Window

The Traceability Window allows a modeler to visualize the connections between elements in the repository. This is very useful to show how Capabilities relate to each other and how they are connected to other elements in the model and in turn how those elements are connected. Capabilities can



be related to other business architecture elements such as Goals and Drivers and applications architecture elements such as Application Services and Applications and more and all of these relationships would be visible through the **Traceability window**.

Learn More: [Traceability Window](#)

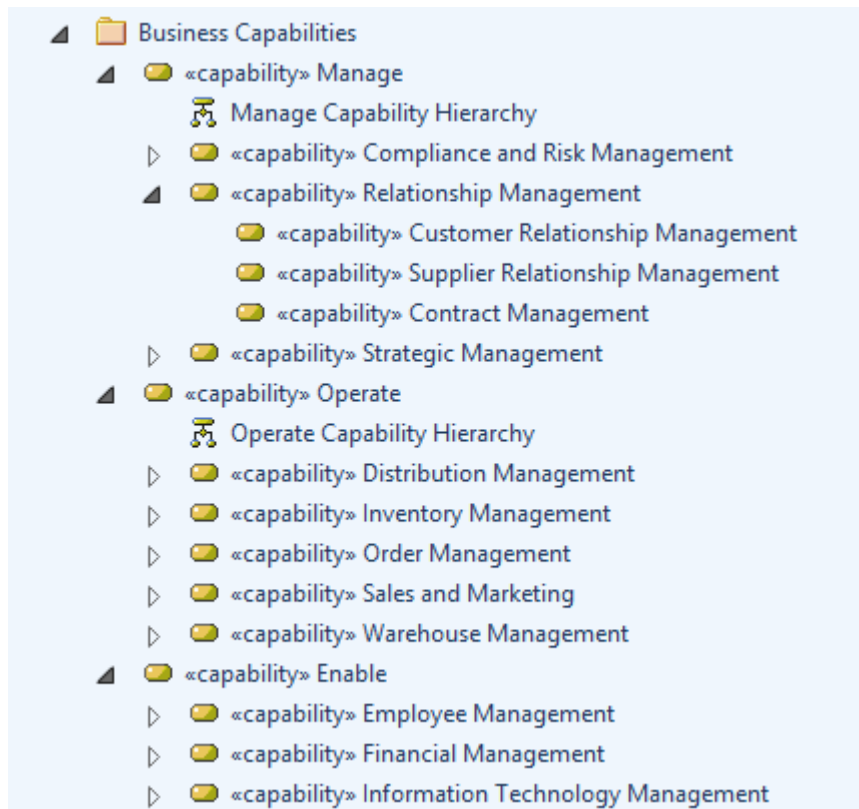
# Capability Modeling

'Capabilities' is an important business concept that describes the abilities or competencies of an organization. They are typically quite stable, and while business processes, functions and roles change quite frequently, capabilities change less frequently. When they do change it is typically in response to a strategic driver or change. Capabilities can be mapped back to strategic goals and objectives. They provide a useful starting point or stake in the ground for enterprise architects to map lower level elements such as business process and functions, applications and technology assets. They typically require a long time frame to deliver and often will span multiple lines-of-business and involve multiple portfolios and projects. For this reason capabilities are often broken down into Capability Increments, which in turn are made up of a number of dimensions including:

- Personnel - including professional development and training
- Processes - including business processes and rules
- Information - including information management
- Facilities - including buildings and structures
- Infrastructure - including components and technology

Enterprise Architect can be used to model Capabilities, including showing their relationships back to strategic concerns such as Drivers, Goals and Objectives. These relationships can be viewed in diagrams, the **Relationship Matrix** or list views, creating powerful communication

mechanisms appropriate for executives, managers and other architects. The capabilities can also be related to tactical concerns such as business processes, logical and physical applications and services, and in turn technology devices and services.



This section lists the main tools available in Enterprise Architect that can be used to perform Capability Modeling. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## List View

The List View is a convenient alternative to viewing the Capabilities (or other elements) in a selected Package or diagram. The view presents the Capabilities in a tabular format where each Capability is a row and properties and Tagged Values are displayed in columns. It is a convenient alternative to a diagram or Package view and allows new elements to be created and properties to be edited. It is useful as it provides a view of the properties of a set of elements in a single table and is often more appealing to business and executive level stakeholders.

Learn More: [List View](#)

## Relationship Matrix

The **Relationship Matrix** is a useful tool for viewing the relationships between sets of elements; it provides a Spreadsheet-like interface for entering, maintaining and viewing these relationships. A business or application Architect will typically want to show how Capabilities or Capability Increments are being realized by application level architecture elements, such as Business Processes or Application Services or Applications. By placing the Capabilities on one axis and the other elements on the other it is possible to create a matrix that displays the relationships. The tool is often favored for use in workshops

and meetings with business stakeholders who are more familiar with Spreadsheet than diagrammatic views.

Learn More: [Relationship Matrix](#)

## Requirements Diagram

The Requirements diagram can be used to create a visual representation of the relationship between Capabilities and other elements in the model, including Business Drivers, Goals, Business Processes and more. The Capabilities can also be created on a diagram by themselves and represented as elements nested inside each other to show a hierarchical relationship. A large format diagram can be used to create a poster of the Capabilities using the documentation engine.

Learn More: [Requirements Diagram](#)

## Stereotyping

The Unified Modeling Language does not itself contain an element to represent a Capability or Capability Increment but they can be included by using stereotypes, which is a mechanism for extending the core language. These stereotypes can be applied to a base element such as a Requirement and then the stereotype can be applied to create new Capabilities or Capability Increments. The stereotype will be visible in the **Browser window** and

diagrams if these options have been set. Capabilities elements can become available if other Technologies or profiles have been enabled.

Learn More: [Stereotyping](#)

## Traceability Window

The Traceability Window allows a modeler to visualize the connections between elements in the repository. This is very useful to show how Capabilities relate to each other, how they are connected to other elements in the model and, in turn, how those elements are connected. Capabilities can be related to other business architecture elements such as Goals and Drivers, and applications architecture elements such as Application Services, Applications and more; all of these relationships would be visible through the **Traceability window**.

Learn More: [Traceability Window](#)

## Specification Manager

The **Specification Manager** is a powerful tool for working with lists of elements including Capabilities and provides a word-processor or spreadsheet-like interface for entering, maintaining and viewing these elements. The Capabilities can be created and editing using this interface without the

need to access elements in the **Browser window** or in diagrams. Changing these elements in the Specification Manager will automatically ensure these changes are visible in all other places in the repository such as diagrams and windows. A Capabilities name and description and its properties can be edited directly through the interface and Tagged Values can be viewed.

Learn More: [Specification Manager](#)

# Compliance Assessment

A Compliance Assessment is conducted to ensure that an implementation project is proceeding in compliance with the defined architecture. This is an important governance mechanism that is used to ensure that the architectural vision is being realized and that the principles and designs are complied with or, if not, dispensations are issued where necessary. A series of checklists can be applied to ensure there is rigor in the assessment and to indicate if the implementation is compliant.

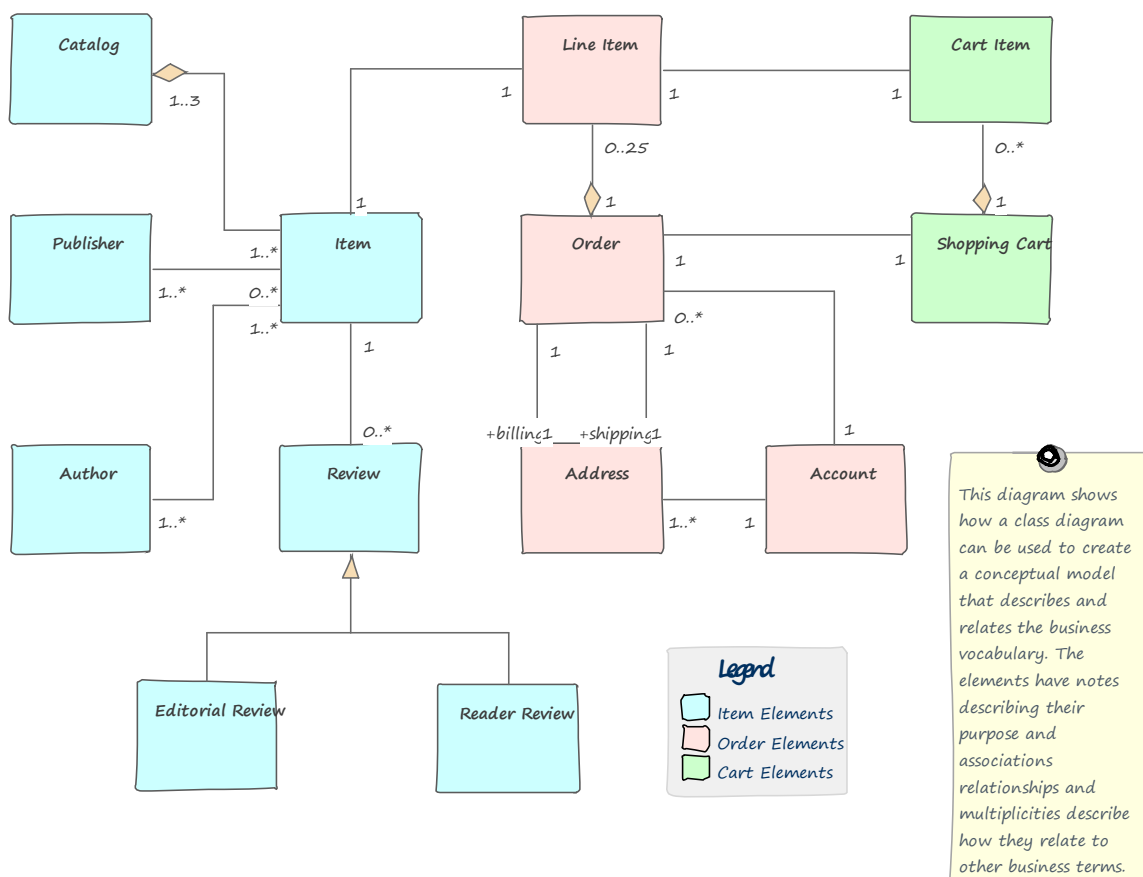
Enterprise Architect has a series of tools that can be used for the assessment, including the powerful **Team Library** facility that can be used to record the results of the assessment. This is a document-based facility that can be tailored to suit each organization or project. One or more checklists can be conveniently applied to give a visual representation to the assessment.

This section lists the main tools available in Enterprise Architect that can be used to perform a Compliance Assessment. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



# Concept Modeling

A Concept Model is an implementation-independent representation of the nouns that are important for an organization, domain or industry. It is a business model and should not be confused with information or data models. Enterprise Architect has tools to create and maintain this model, with the flexibility to present the model graphically in a diagram or textually in a list view, a published document or a web page. The elements in the Concept Model can be linked to any number of up-process or down-process elements, such as business goals and capabilities.



This section lists the main tools available in Enterprise

Architect that can be used to perform this business analysis technique. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the Help topic [Business Analysis Tools](#).

## Class Diagram

A Class diagram can be used to model important concepts in the domain. The concepts are modeled using UML Classes, and Names, Descriptions and details such as Attributes can be entered for each Class. The concepts can be related to each other using relationships such as Association and Generalization. These concepts can then be used throughout the model as a type of Glossary, including in element notes where they can be referenced.

Learn More: [Class Diagram](#)

## Project Glossary

The Glossary is a purpose-built tool for managing important concepts and their meanings, which can be included in documentation. If words defined in the Glossary are used in element or diagram notes they will automatically appear as hyperlinks and the meaning will conveniently appear in a pop-up window.

Learn More: [Glossary](#)

## Entity Relationship Diagram

An **Entity Relationship diagram** is a conceptual or abstract model of information for a system, often created as a precursor for the development of a database schema. The Entities and their attributes can be modeled and the relationships between one or more Entities can be drawn.

Learn More: [Entity Relationship Diagram](#)

## National Information Exchange Modeling (NIEM)

NIEM is an XML-based information exchange framework that is the result of collaboration between all levels of American government. Enterprise Architect has extensive support for the framework allowing organizations to create subsets of the standard for the purpose of exchanging standards compliant messages. Conceptual models can be transformed into NIEM-compliant representations.

Learn More: [National Information Exchange Modeling \(NIEM\)](#)

## Ontology Definition Metamodel

Ontologies are a useful and formal way of describing the concepts in a domain and include the names and definition of the types, their properties, and their relationships with each other. Enterprise Architect implements the Object Management Group's Ontology Definition Metamodel (ODM) and supports both the Resource Description Framework (RDF) and the Web Ontology Language (OWL).

Learn More: [MDG Technology for ODM](#)

## Schema Composer

The **Schema Composer** is a powerful tool that can be used to define a range of formal schemas from a model. The tool allows a Conceptual model such as a Class diagram to be converted to a formal XML schema without the user needing to understand the underlying complexity of the XML machinery.

Learn More: [Schema Composer](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. The terms

defined in the Glossary or the Classes used to define the concepts can both be conveniently generated and included in documentation. In the case of terms defined in the Glossary, this includes the term Name, Description and Type. In the case of terms defined as Classes it includes the term Name, Description and a variety of other information including Attributes, Tagged Values and Connections with other terms. A sophisticated template facility exists that provides a range of in-built templates and allows the user to create their own defining styles, images and a wide range of other formatting options.

Learn More: [Documentation](#)

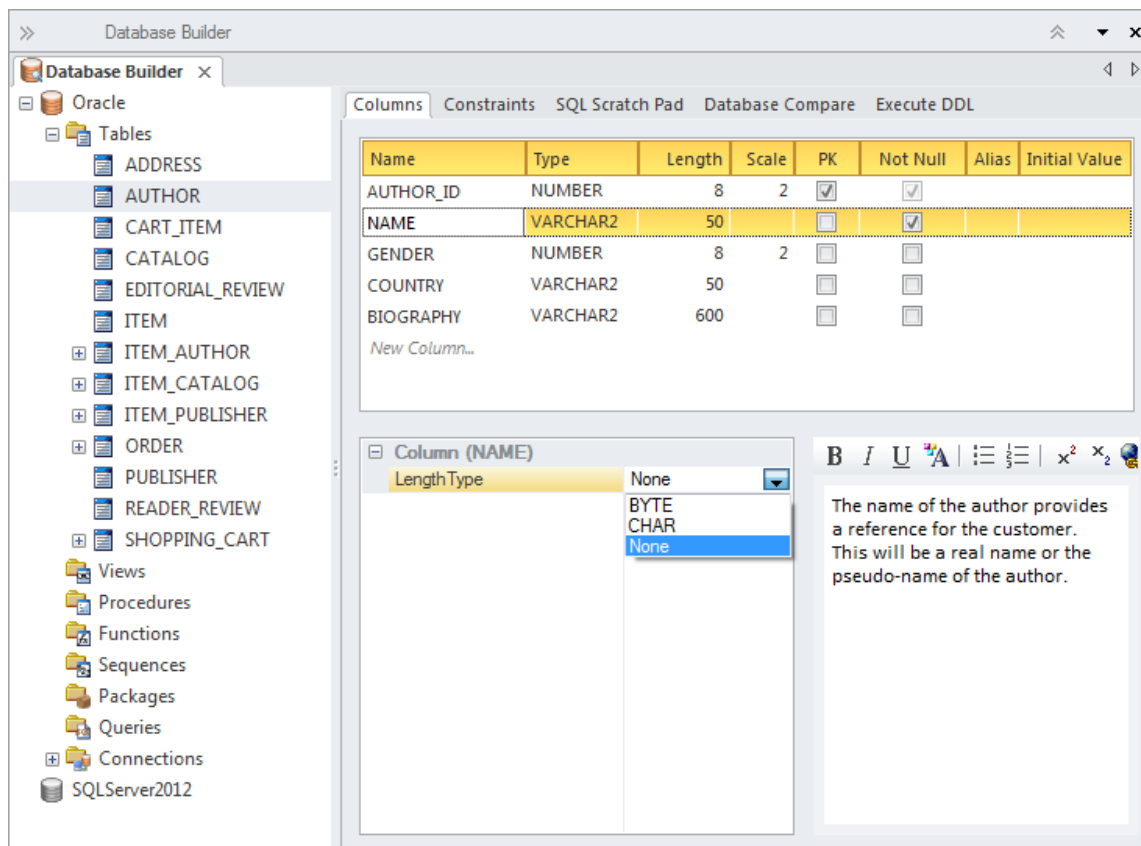
## Visual Filters

Visual Filters can be used to hide or dim parts of the diagram so that other parts can be emphasized. This is particularly useful when presenting a Class diagram of important concepts to a team in a meeting or demonstration. Context Filtering is easy to set up and will emphasize the currently selected diagram element and its directly connected elements.

Learn More: [Visual Filters](#)

# Data Modeling

Enterprise Architect has profound support for Data Modeling. Data models describe the things that are important in a domain or solution, and their attributes (or columns), including their types and the relationships between them. Data modeling can be done for a number of reasons, including to clarify and communicate and also to implement a solution on a particular technology platform. Data modeling can occur at a number of different levels, from the conceptual data models that are analogous to concept models and are used for clarifying and communicating, through logical data models that include data normalization to physical models used for implementation. Enterprise Architect has a number of diagrams such as the Class diagram and the Data Modeling diagram that can be used to visualize the models, and a number of purpose built tools such as the **Database Builder** and the **Schema Composer** that will assist a modeler to be highly productive.



This section lists the main tools available in Enterprise Architect that can be used to perform this business analysis technique. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the Help topic [Business Analysis Tools](#).

## Class Diagram

Conceptual data models can be defined using a Class diagram and these often provide an abstract (conceptual) precursor for logical and physical data models. These information models can be created at any level of detail ranging from a simple list of elements to elements related to each other with connectors and containing attributes.

Learn More: [Class Diagram](#)

## Database Builder

The **Database Builder** is the primary tool for working with live databases and the data models that represent them. The tool can be used to model a database from scratch including a wide range of database objects such as Tables, Views, Stored Procedures and more. Databases can be generated from the data models and data models can be reversed engineered from live database connections. A modeler can work with the database objects in tabular form or in a diagram.

Learn More: [Database Builder](#)

## Entity Relationship Diagram

An **Entity Relationship diagram** is a conceptual or abstract model of information for a system often created as a precursor for the development of a database schema. The Entities and their attributes can be modeled and the relationships between one or more Entities can be drawn.

Learn More: [Entity Relationship Diagram](#)



## Generate or Import XML Schema

XML schemas are typically used as a standards-based data definition for the exchange of information between systems or parts of systems. Enterprise Architect has support for modeling schemas and generating schemas (**XSD**) documents based on these models. There is also the ability to reverse engineer a schema document into a schema model.

Learn More: [Generate or Import XML Schema](#)

## Model Transformation

Model transformations are a powerful productivity tool that can be used with data modeling. The transformation engine has a number of built-in but configurable templates that can be used to transform a conceptual Class model into a logical or physical database schema. The generated schema can then also be generated to Database Definition Language (DDL) or, using the **Schema Composer**, applied directly to an existing database.

Learn More: [Model Transformation](#)

## Schema Composer

The **Schema Composer** allows a modeler to work with industry standard schema definitions to create the structure for messages flowing between organizations, systems, or parts of systems. The Schema Composer supports a number of international standard models and hides much of the underlying complexity inherent with working with complex schemas. It also supports the generation of the messages in a number of formats, including **XSD**, RDFS and JSON.

Learn More: [Schema Composer](#)

## Visual Filters

Visual Filters can be used to hide or dim parts of the diagram so that other parts can be emphasized. This is particularly useful when presenting a Data Modeling diagram to a team in a meeting or demonstration. Context Filtering is easy to set up and will emphasize the currently selected diagram element and its directly connected elements.

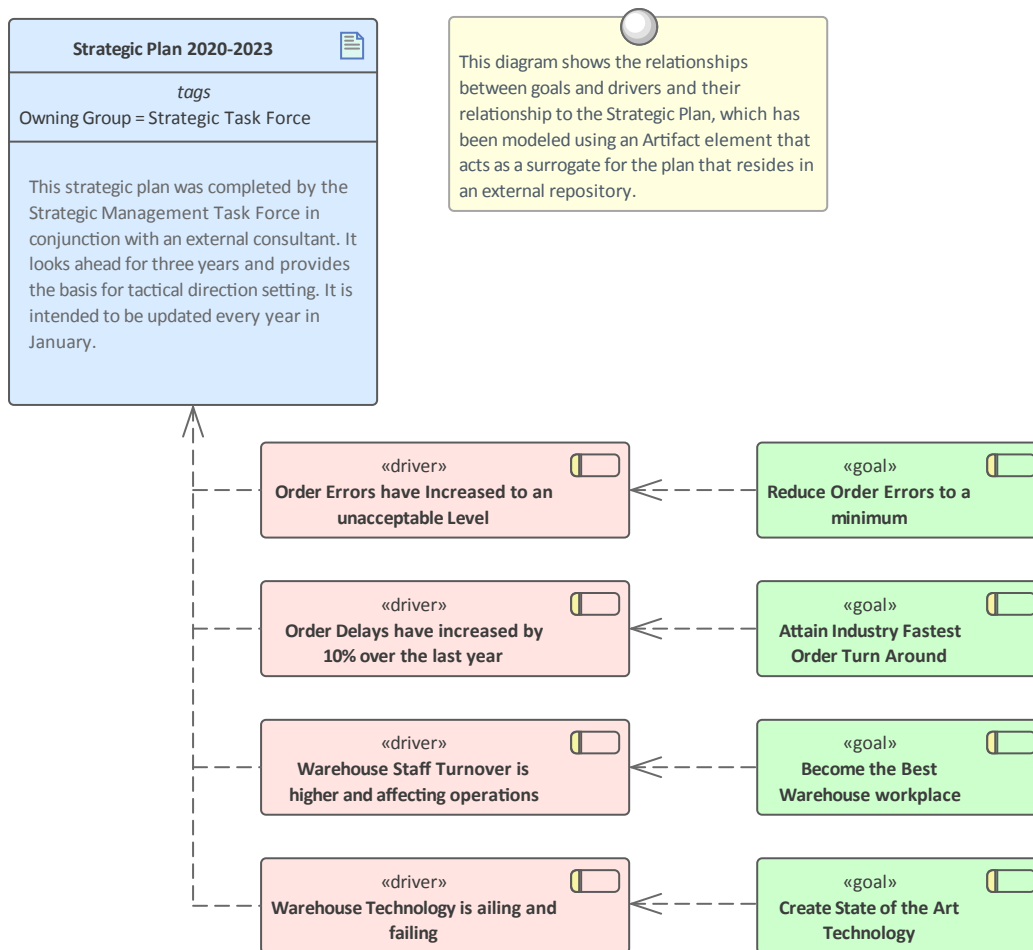
Learn More: [Visual Filters](#)

# Driver Modeling

Drivers are events or conditions that motivate an enterprise to change some aspect of its goals and consequently its objectives, and potentially business activities or technical solutions. The drivers can be top-down in the form of strategic drivers, such as regulatory compliance rule changes, or bottom-up in the form of tactical drivers such as technology changes or opportunities that have been discovered by audits, performance assessment, surveys or other mechanisms.

Enterprise Architect has support for modeling both strategic and tactical drivers with the use of a stereotyped driver element that can be linked to goals and other organizational elements to show how the organization must respond.

This section lists the main tools available in Enterprise Architect that can be used to perform Driver Modeling. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Stereotyping

Neither the Unified Modeling Language nor Enterprise Architect have a built-in element to represent Business Drivers, but they can be included using stereotypes, which is a mechanism for extending the core language. These stereotypes can be applied to a base element such as a Requirement and then the stereotype can be applied to create new Drivers. The stereotype will be visible in the **Browser window** and diagrams if these options have been set. These stereotyped elements might be available if other Technologies or profiles have been enabled.

Learn More: [Stereotyping](#)

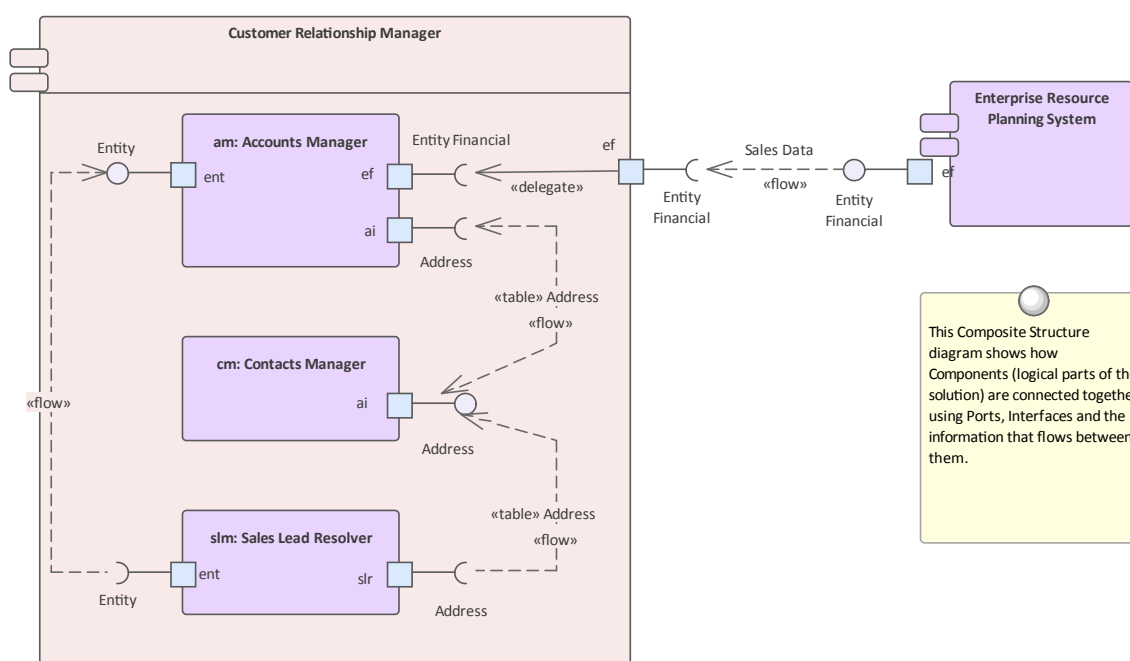
## Traceability Window

The Traceability Window is a powerful view that allows a modeler to visualize the connections between elements in the repository, regardless of whether the relationships exist in diagrams. This is very useful to show how Drivers relate to each other, how they are connected to other elements in the model and, in turn, how those elements are connected. Drivers can be related to other business architecture elements such as Goals and Objectives, and to applications architecture elements such as Application Services, Applications and more; all of these relationships are visible through the **Traceability window**.

Learn More: [Traceability Window](#)

# Functional Decomposition

Enterprise Architect has been built as a platform for managing models of complex systems including business, information and technology systems. The tool allows business concepts such as outcomes, capabilities and processes to be decomposed. Information models can be broken down from high level concepts to details and technology models of components and devices can also be decomposed. There are a wide range of tools for working with and representing these models starting with the **Browser window** that displays elements in a hierarchy allowing the user to expand or collapse levels thus revealing or hiding detail. Decomposition can also be represented in diagrams using elements such as the Aggregation connector on a single diagram or the ability to drill down through a whole chain of connected diagrams from a high level element in a value chain down to the low level processes.



## Browser Window

Functional Decomposition can be achieved by creating a feature hierarchy using the **Browser window** without the need to create a diagram. A system Feature is a good way to capture the high level capabilities of a system and these can be created directly in the Browser window. Additional features can be added under each first level feature creating a second level of features. These second level features can have features nested under them creating a third level. The resulting tree of Features provides a useful way of describing Functional Decomposition that can be presented to and reviewed by stakeholders.

Learn More: [Browser Window](#)

## Component Diagram

The composition of a system can be shown using a Component diagram where systems are displayed in a hierarchy including drill down to other Component diagrams. This allows a complex system that is made up of a number of subsystems to be modeled.

Learn More: [Component Diagram](#)

## Organizational Chart Diagram

The Organizational Chart Diagram is part of the Strategic Modeling group of diagrams and shows the structure of an organization including officials, roles responsibilities, business units or departments. The roles or departments can be displayed in a diagram using a tree layout or in a list view. Any number of Tagged Values can be added to the elements or connectors to add additional information as required. The diagram can be used to show the decomposition of an organization by roles or business units. Learn More: [Organizational Chart Diagram](#)

## Requirements Diagram

Functional Decomposition can be achieved by using a Requirements diagram to create a hierarchy of system features. A system Feature is a good way to capture the high level capabilities of a system, and these can be broken down to a number of levels using a tree structure created with an Aggregation or Composition Relationship. This provides a compelling representation of scope that can be reviewed by the stakeholders and used as a guide through the initiative. Gaps and out-of-scope features should be identified as early as possible and the tree amended to reflect these. Out of Scope features could be left as part of the tree but annotated in some way to indicate that they are out of scope such as by



using a stereotype or by using color with a Diagram Legend.  
Learn More: [Requirements Diagram](#)

## BPMN Business Process Diagram

The **BPMN Business Process** diagram is useful for modeling business processes that form part of a process hierarchy. Enterprise Architect allows the modeler to define a decomposition of processes that can be drilled down to from a higher level. It allows a modeler to document the business process including the way a process starts, what work is performed and how it ends. Gateways and connecting lines determine the sequence of activities. The Business Process diagram is emerging as an important standard for modeling business processes.

Learn More: [Business Process Diagram](#)

## Specification Manager

The **Specification Manager** is the primary tool for working with text based elements such as Features, Requirements and Components. These and other elements can be created directly in the Specification Manager and their name and descriptive text can be added in a compelling Word Processor or Spreadsheet format. Other built-in properties such as status, version and phase can be managed directly in

the columns of the Specification Manager and where property lists are available these can be chosen or changed from drop down lists. Additional properties in the form of Tagged Values can also be managed from within the Specification Manager. The Specification Manager will reflect the hierarchical structure of elements and can use level numbering if this has been turned on.

Learn More: [Specification Manager](#)

## UML Activity Diagram

The Activity diagram, which is part of the Unified Modeling Language, is a useful alternative to other diagrams such as Flow Charts and Business Process diagrams. They allow a modeler to describe the sequence of behaviors including how they start, what work is performed and decisions that change the flow and the way the process ends.

Learn More: [Activity Diagram](#)

## ArchiMate Business Process Diagram

Allows the high level description of a business process showing what triggers the process and the order of processes. If detailed representations are required **BPMN Business Process** diagrams or UML Activity diagrams can be used.

Learn More: [ArchiMate](#)

# Gap Analysis

Gap Analysis is a useful technique that can be used to determine the differences between two states of the architecture. It is typically performed between a baseline architecture and a target architecture, or between a baseline and any intervening transition architectures. Gap Analysis is an important step that should be performed before a roadmap can be created, solution options can be described and migration planning commenced. A gap can be one of three types:

- Unintentionally Omitted
- Intentionally Omitted
- Not yet described

The gaps can occur in any of the architecture domains including Business, Information, Application and Technology architectures. Typical gaps include things such as missing Business Processes, Redundant Facilities, Data not available at the right place and time, applications that are no longer required, new technology services, or devices needed to support an application.

Enterprise Architect has a purpose-built **Gap Analysis Matrix** tool that can be used to conduct Gap Analysis and to present the results to business and technical audiences. Any number of Gap Analysis Matrices can be created for any or all of the architecture domains, and these can conveniently be included into high quality publications generated automatically from the model.

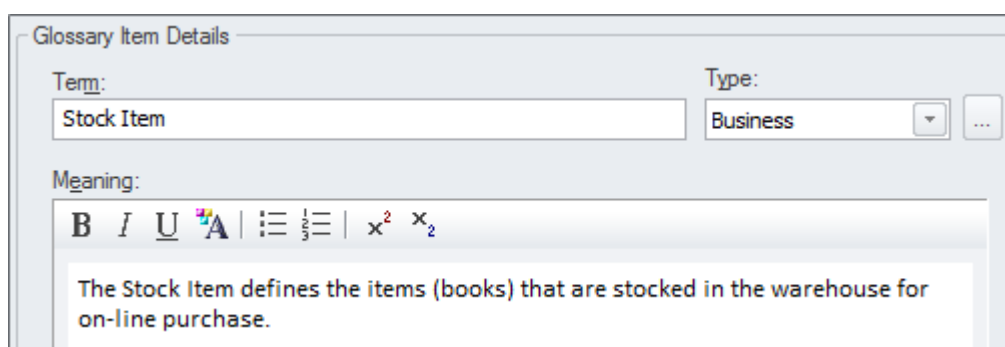
This section lists the main tools available in Enterprise Architect that can be used to perform Gap Analysis. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

Gap Analysis Matrix				
Target Architecture:	Target 1	Filter:	ABB	Profile:
Baseline Architecture:	Baseline 1	Filter:	ABB	Record Gap As:
				<a href="#">Refresh</a>
				<a href="#">Options</a>
Target \ Baseline	Video Conferencing Services	Enhanced Telephony Services	Mailing List Services	Missing / Eliminated
Broadcast Services				Retired service : Intentionally eliminated
Video Conferencing Services	Included			
Enhanced Telephony Services		Potential match		
Shared Screen Services				Address Shared Screen Service : Unintentionally eliminated
New		Improve Telephony service : To be enhanced	Mailing List : New-To be produced or developed	

# Glossary

Enterprise Architect supports two different ways of creating a glossary. The most obvious way is to use the built-in **Project Glossary**, which is available from the Publish ribbon. This provides a convenient way for modelers to store terms and their associated meanings, grouped by user-defined categories. An alternative (and perhaps more sophisticated) method is to create a Class diagram of the terms, linking the elements to create relationships between the terms.

Enterprise Architect has a convenient Project Glossary that can be used to store a list of terms and their definitions. The terms can be grouped into any number of user defined categories such as business, technical, domain specific grouping such as medical, scientific, or aviation. The power of having the terms defined in the model is realized when modelers write description in model elements as these will be linked automatically to terms defined in the Project Glossary.



The screenshot shows the 'Glossary Item Details' dialog box. It has two main sections: 'Term' and 'Meaning'. The 'Term' section contains a text box with 'Stock Item' and a 'Type' dropdown menu set to 'Business'. The 'Meaning' section contains a rich text editor with a toolbar showing bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and subscript/superscript. The text area contains the sentence: 'The Stock Item defines the items (books) that are stocked in the warehouse for on-line purchase.'

## Project Glossary

The Glossary is a purpose built tool for managing project terms and their meanings, which can be included in documentation. If words defined in the Glossary are used in element or diagram notes, they will automatically appear as hyperlinks and the meaning will conveniently appear in a pop-up window.

Learn More: [Glossary](#)

## Class Diagram

A Class diagram can be used to model important concepts in the domain. The concepts are modeled using UML Classes, and names, descriptions and details such as attributes can be entered for each Class. The concepts can be related to each other using relationships such as Association and Generalization. These concepts can then be used throughout the model as a type of Glossary, including in element notes where they can be referenced.

Learn More: [Class Diagram](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. The terms

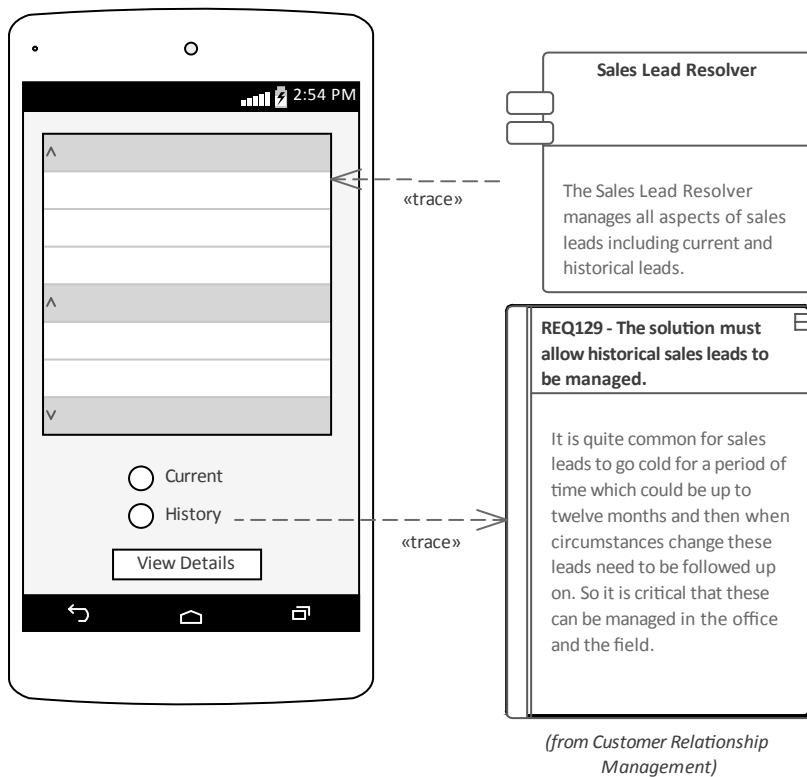
defined in the Glossary or the Classes used to define the concepts can both be conveniently generated and included in documentation. In the case of terms defined in the Glossary, this includes the term Name, Description and Type. In the case of terms defined as Classes, it includes the term Name, Description and a variety of other information including Attributes, Tagged Values and Connections with other terms. A sophisticated template facility exists that provides a range of in-built templates and allows the user to create their own, defining styles, images and a wide range of other formatting options.

Learn More: [Documentation](#)



# Interface Analysis

Any complex system will typically have a number of interfaces; these can be any of a wide range of types, including user interfaces, component interfaces, application programming interfaces and hardware interfaces. Enterprise Architect has built-in facilities to model all these types of interface and they can be connected into a single articulated model. There is support for a wide range of user interfaces including applications, web pages, and hand held devices such as Tablets and Phones, including devices such as iPhones, Android and Windows phones. Business interfaces such as people outside the organization or business processes can also be modeled.



This diagram shows the rich support for creating wireframe models of modern portable devices such as cell (mobile) phones and tablets. The toolbox has rich support for a wide range of pre-built controls applicable to the main types of devices in use including Apple and Android devices. Parts of the interface can be linked to other elements in the repository creating powerful traceability.

## Component Diagram

A system is typically composed of a number of subsystems that are often connected to each other through known interfaces. These interfaces are where data or control signals are exchanged between the Components. These can be modeled using the Unified Modeling Language Interface element, which allows operations and services to be defined. Components can expose these Interfaces graphically in a diagram as Provided or Required interfaces, allowing the Components to be connected.

Learn More: [Component Diagram](#)

## Deployment Diagram

Hardware Nodes and Devices interface to other Nodes and Devices through published Interfaces. These can be described with UML Ports that can be used to describe aspects of the interface such as IP Addresses and Protocols. These describe the physical or virtual connections between the various pieces of hardware that make up the system.

Learn More: [Deployment Diagram](#)

## Wireframe Diagram

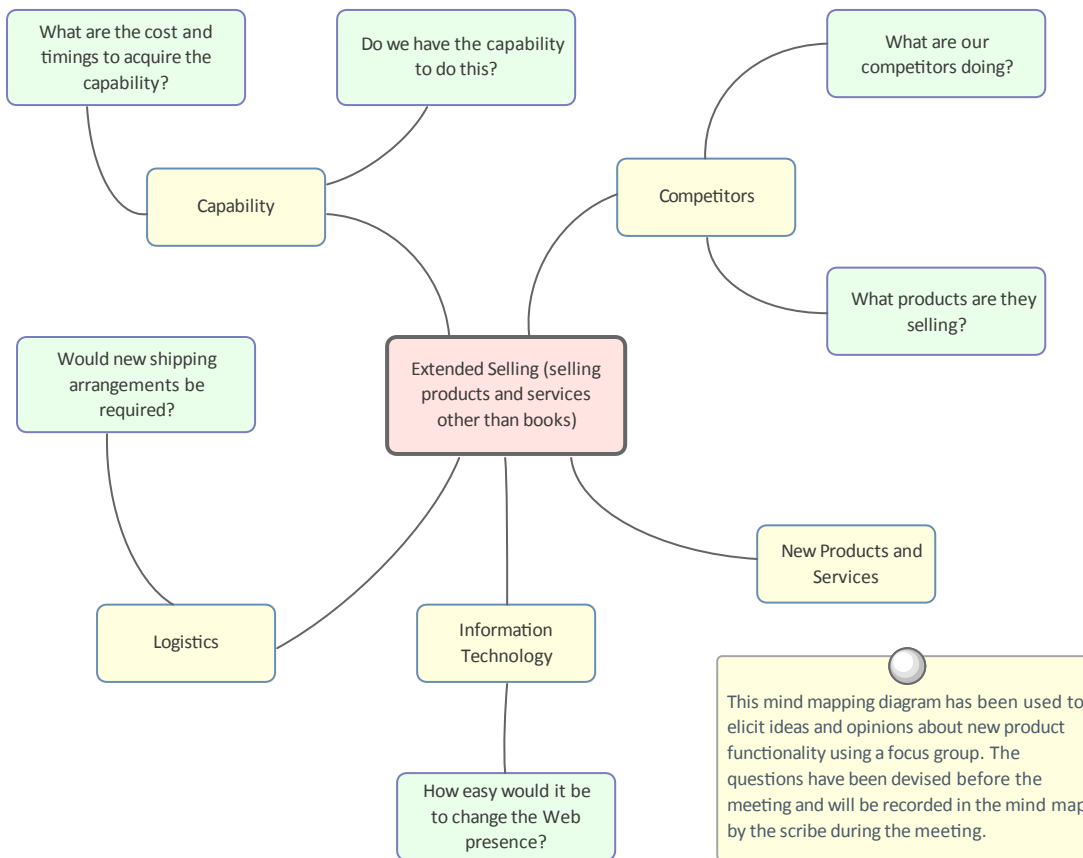
Wireframe diagrams allow a modeler to create powerful and expressive models of the user interface for a variety of hand held devices, including Apple, Android and Windows phones and tablets. Screen dialogs and web pages can also be modeled. The physical appearance of the devices themselves can be represented and configured by the use of a number of Tagged Values. Patterns can be used to automatically create a base model for a device. A range of controls are available applicable to each type of device including Android Widgets, Apple Controls and Windows Tiles.

Learn More: [Wireframe Diagram](#)



# Mind Mapping

Enterprise Architect has a flexible and easy to use Mind Mapping diagram which can be used for structuring thoughts or for note taking in a variety of situations from stakeholder workshops, focus groups, interviews and collaborative games. They can be used to map the thoughts and ideas of an individual or a group of people and provide a non-confronting and appealing way of recording information. The power of Enterprise Architect can be used to create the Mind Map but then other elements such as a set of Requirements that have been derived from analysis of the Mind Map can be linked back to elements in the Mind Map to show how the Requirements trace back to ideas expressed in a workshop. This provides a powerful analysis audit trail of the provenance of Requirements or other solution precursors such as Capabilities, Goals, Business Objectives and more.



## Mind Mapping Diagram

A Mind Mapping diagram can be used to record a wide range of information in a compelling format that mimics the way we think about ideas as a graph of thoughts. It can also be used in Workshops, Focus Groups and other meetings with Stakeholders or team members as a note taking tool. Other elements in the model can be connected to the Topics and Sub Topics in the Mind Mapping diagram to create powerful and expressive relationships between the ideas and their specification or implementation.

Learn More: [Mind Mapping Diagram](#)

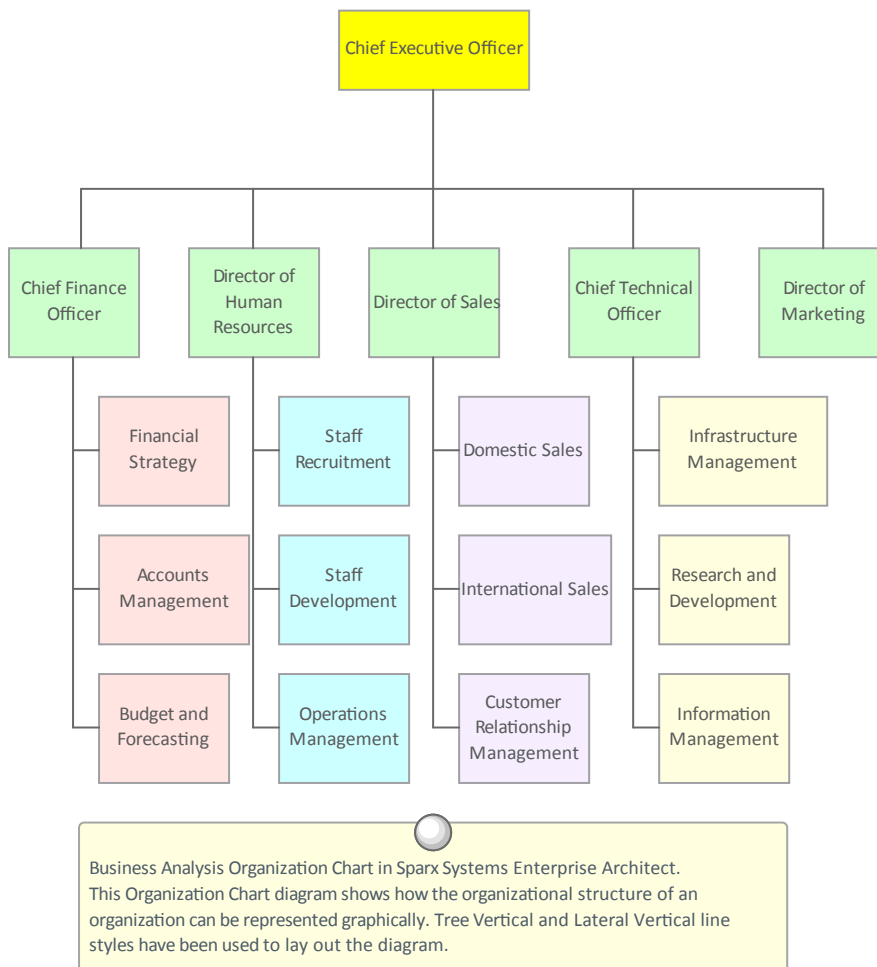
## Traceability Window

The Traceability Window is a useful window that allows a modeler to visualize the connections between elements in the repository. This is useful when used with elements in a MindMapping diagram to show how the topics are related to each other but also to view how the topics and sub topics are related to other elements in the repository such as Requirements or System Components.

Learn More: [Traceability Window](#)

# Organizational Modeling

Enterprise Architect has an Organization Chart diagram as part of its strategic modeling diagram set that allows organizational structures to be modeled; the roles in the organization chart can be linked to any number of model elements, including Vision Statements, Business Goals, Objectives, Processes and Stakeholder Requirements. There is also a useful mechanism to show the different people who occupy the roles over time, using Instances of the Classes.



## Organizational Chart



The Organizational Chart diagram is part of the Strategic Modeling group of diagrams and shows the structure of an organization including officials, roles, responsibilities, business units or departments. The roles or business units can be displayed in a diagram using a tree layout or in a list view. Any number of Tagged Values can be added to the elements or connectors to add additional information as required. The elements that make up the organizational chart can then be used in other parts of the model such as assigning business owners to business processes, business rules, systems and more.

Learn More: [Organizational Chart Diagram](#)

## Baselines

The Baseline Tool is used as a type of backup and recovery tool and can capture a snapshot of the Package that contains the Organizational Chart at a point in time and then at a later time the current Package can be compared to this (or another Baseline) for the purpose of determining what has changed. Changes that are not desirable can be overridden by the values contained in the Baseline at a granular level, thus restoring the model to the state that existed at the time the Baseline was captured.

Learn More: [Baselines](#)



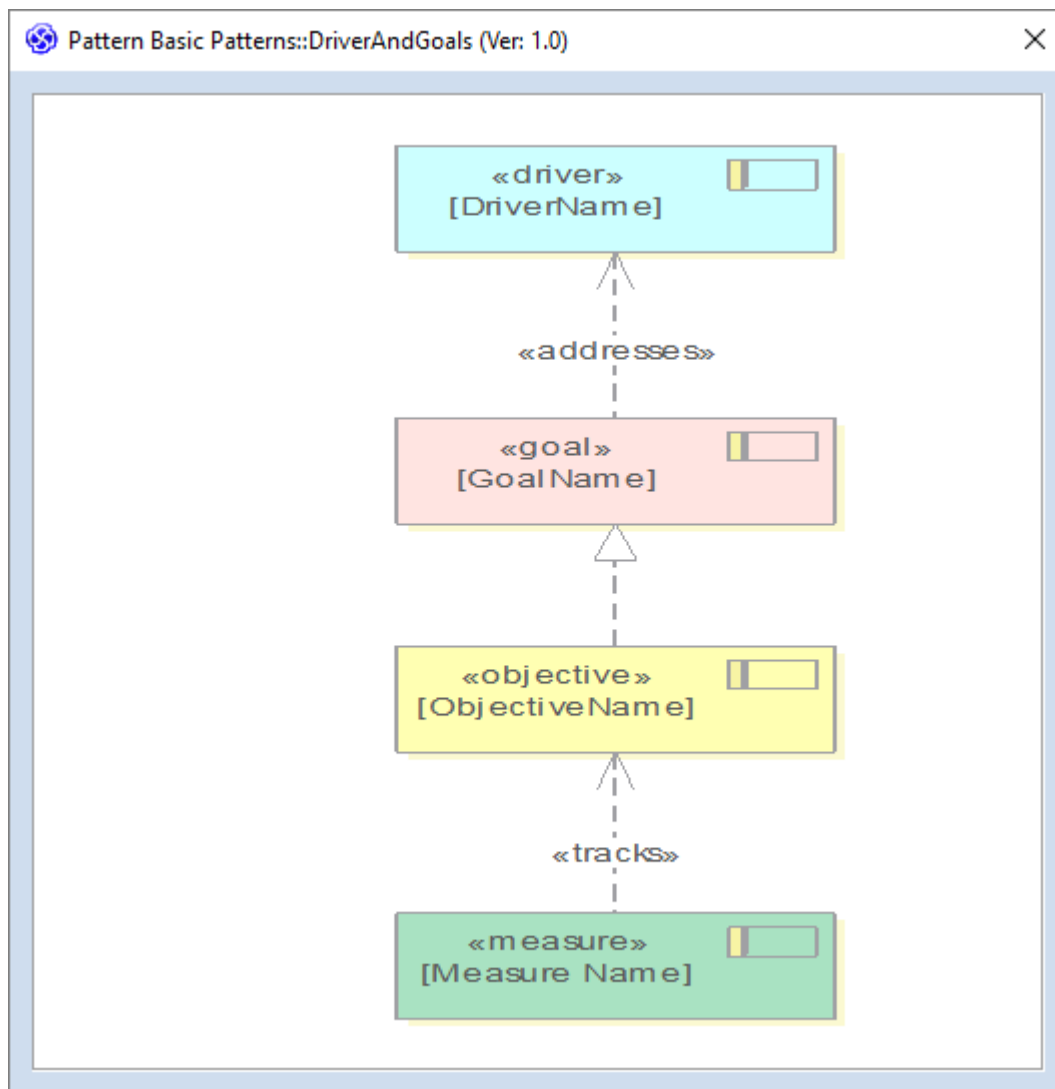
# Pattern Analysis

Patterns are a valuable tool in the architect's toolkit. They allow the architect to reuse a solution or part of a solution that has been proven successful and apply it in the context of their own problem. The use of architecture Patterns is in its infancy and has its origins with Christopher Alexander (a building architect) who wrote a seminal book entitled *A Pattern Language*. Much of the use of Patterns in the information technology industry has been at a software engineering level but interest in them is starting to increase at an architecture level. Patterns can be created and used at any level from a team level right up to industry and cross industry foundation architectures and they can be used for any recurring idea from business level diagrams down to technology architecture.

Enterprise Architect supports the mining, identification, creation, and use of Patterns at any level. Existing architectures can be viewed and diagrams that an architect identifies as being worthy of Pattern status can be saved as a Pattern. A description of the Pattern can be included to assist others who are using the Pattern to understand its purpose and how it should be applied. Patterns developed by other architects, organizations, industries or groups can be imported into the 'Resources' tab of the **Browser window**.

This section lists the main tools available in Enterprise Architect that can be used to perform Pattern Analysis. There are also many additional tools that a modeler might

find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Patterns

Patterns are a productivity and architecture tool for creating reusable fragments of an architecture. Any diagram can be published as a Pattern and then stored in the 'Resources' tab of the **Browser window**. This is the perfect mechanism for

the reuse of architectures and including problem or opportunity definitions and solutions or parts of solutions. Existing architectures and completed projects can be mined for reusable diagrams and ideas and these can then be exported as Patterns including annotations that describe the Pattern's motivation, intent, applicability and more.

Learn More: [Patterns](#)

## Model Search

The **Model Search** facility can be used to find a set of elements that meet a particular set of criteria. This is useful when searching for elements in diagrams that could form the basis of a Pattern. A modeler can use a built-in search but would more typically devise their own search using the powerful Query Builder or, if required, the SQL Editor or an **Add-In Search**. When the search is run a list of elements is returned that meet the conditions and parameters defined in the search. The search can be saved and run again at any time. The elements returned in the search can be located in the **Browser window** and in diagrams, and also generated to documentation.

Learn More: [Model Search](#)

# Principles Management

Principles Management is concerned with the creation, maintenance and governance of architecture principles, their relation to higher level enterprise principles, and their application and adherence at an implementation level. A principle is typically a statement that is used to guide the development of architectures and the solutions that implement the architectures. They are typically created at the time an architecture program is initiated, and are expected to be unchanging, acting as a pillar that the architectures and their implementation are held upon. They are most commonly developed by an Enterprise Architect in cooperation with leading business and technical stakeholders, including the domain architects. They form an important governance device and are typically managed by the Architecture Board or an equivalent body.

Enterprise Architect can be used to model and manage the Architecture Principles and their relation to enterprise level principles, effectively creating a principle catalog. The principles can be used in the context of a solution architecture to guide or constrain the design of a solution, ensuring that it is complied with or a dispensation is issued. Tagged Values provide a way of describing the important aspects of a principle in addition to its name and statement. The Traceability Window can be used to create relationships between the Architecture principles and the higher level Enterprise Principles. Security can be used to ensure the principles are not changed either inadvertently or

intentionally. **Baselines** and auditing can be used to track changes that have been made to the principles.

This section lists the main tools available in Enterprise Architect that can be used to perform Principles Management. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## Auditing

The **Auditing** feature can keep track of the changes to architecture content including what was changed, when it was changed and by whom. Auditing is by default disabled and must be enabled before the changes to architecture elements will be recorded. Once enabled it is a passive tool that silently records the changes to elements. The facility can be particularly useful when managing principles as they are intended to be enduring and are not intended to be changed very often after they have been defined. The audit facility is useful as it will alert a librarian or administrator that a principle has been changed.

Learn More: [Auditing](#)

## Baselines

The Baseline Tool can capture a snapshot of the Architectural Principles at a point in time and a comparison can be made between this or any other snapshot and the model at a later point in time. The comparison tool will enable an architect to visualize what has changed and if necessary revert back to content contained in the baseline. Any Package in the Principle hierarchy can be baselined and any number of baselines can be created. It is typically the Chief Architect that would manage the baselines and update the principles if required.

Learn More: [Baseline Tool](#)

## Class Diagram

The Class diagram can be used to create visual representations of the Principles, including how they are related to each other. The Principles, or instances of the Principles, can be added to any diagram and will provide important guidance for architects who are required to be constrained by these principles when creating their architectures, and also for implementers when creating solutions. The instances of the Principles describe how they apply in a particular context.

Learn More: [Class Diagram](#)

## Security



The **Security System** in Enterprise Architect is intended to encourage collaboration, but it can be used to lock Packages or elements to prevent them from being changed. This is a useful mechanism that can be applied to the Packages containing the Principles, ensuring that they are not changed unintentionally or inadvertently. It would typically be the chief architect or the model librarian acting on their behalf who would explicitly lock and unlock the Principles for update.

Learn More: [Security](#)

## Tagged Values

Tagged Values are used to manage additional properties of elements and other items in the model. They can be used with principles to add the descriptive properties of the principles including Rationales and Implications. They can be viewed through the **Properties window** or in the element's property sheet or in the Tagged Values compartment of a diagram object.

Learn More: [Tagged Values](#)

## Traceability Window

The Traceability Window is a useful when working with

Principles as it allows a modeler to visualize the connections between principles and other elements in the repository.

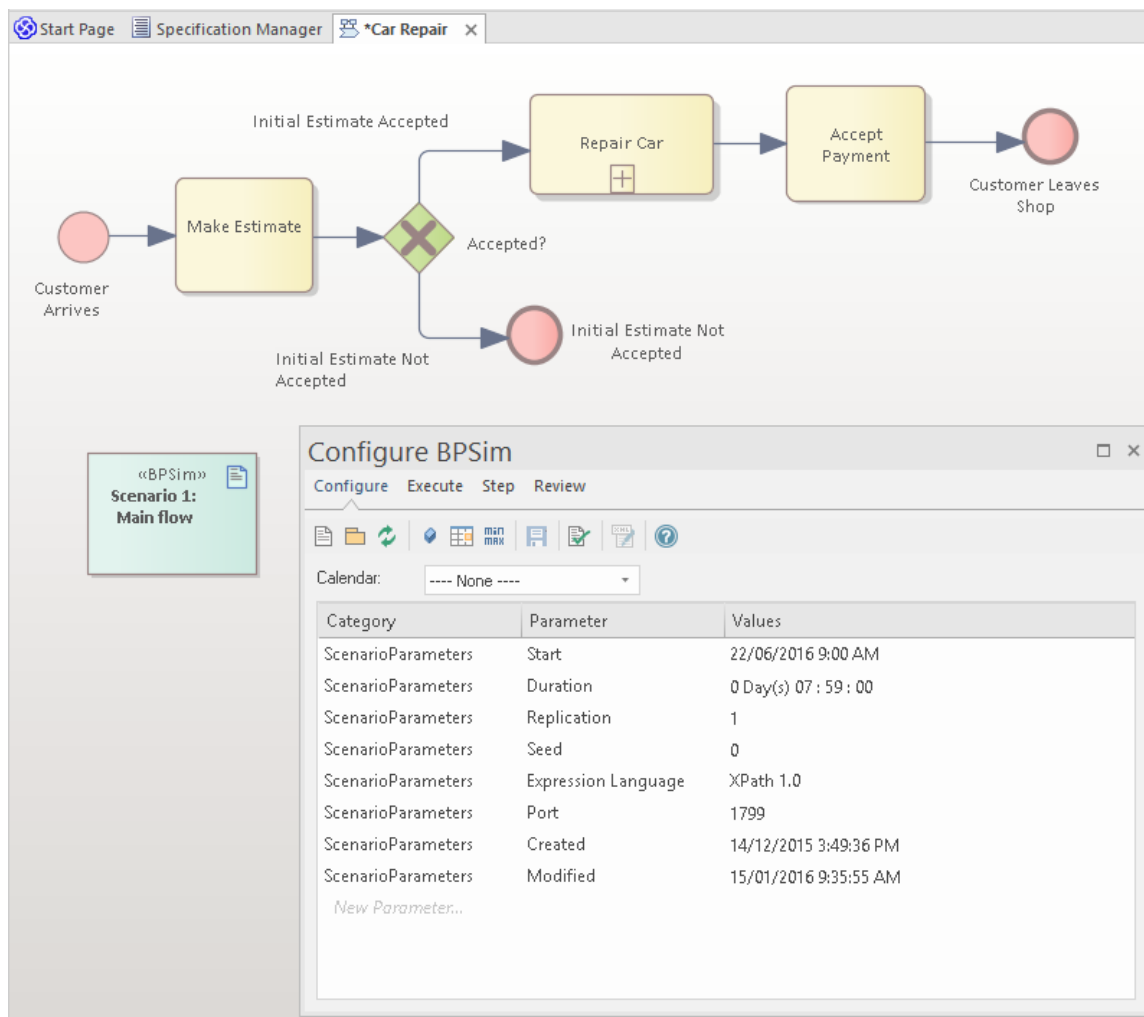
This is useful to show how Principles relate to each other and how they are connected to other elements in the model and in turn how those elements are connected. For

Principles to be effective they need to be applied at the level of architectural representations and implementation projects and the Traceability Window can show how the principles have been applied.

Learn More: [Traceability Window](#)

# Process Analysis

Enterprise Architect allows processes to be modeled and these models can be analyzed to assess them for how effective and efficient they are and to view any opportunities for change or improvement. There is a powerful element discussion facility allowing modelers to discuss processes and analyze the processes collaboratively even when the modelers are geographically dispersed. Enterprise Architect also has a powerful simulation facility to simulate the models and derive information about the models identifying any points in the process that can be improved.



## Business Process Simulation (BPSim)

The BPSim facility provides a way of simulating processes written in Business Process Model and Notation (BPMN), providing valuable results that can be used in process analysis. The BPMN models are augmented with extra data as parameters to the simulation. It allows structural and capacity analysis to be performed, providing for pre and post execution optimization. Enterprise Architect allows you to construct the Process models and enter the appropriate data, which is then sent to an internal or external BPSim Simulation engine.

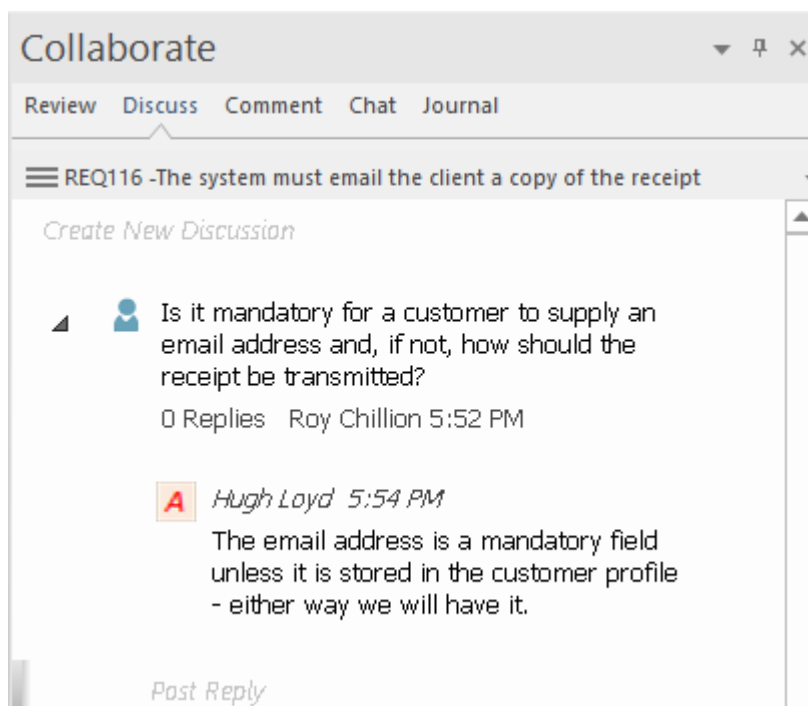
Learn More: [Business Process Simulation](#)

## Discussions

The **Collaborate window** is a convenient facility that allows commentary to be made on processes (or any element) without contaminating the notes with discussions that ultimately don't contribute to the integrity of the model. The Collaborate window allows a modeler to initiate a discussion and for others to reply. It is a perfect way for stakeholders and analysts to discuss aspects of a process as it is being analyzed.

A **Discussions** summary window conveniently displays the Discussions for all elements in the repository.

Learn More: [Discussions](#)



## Maintenance Items

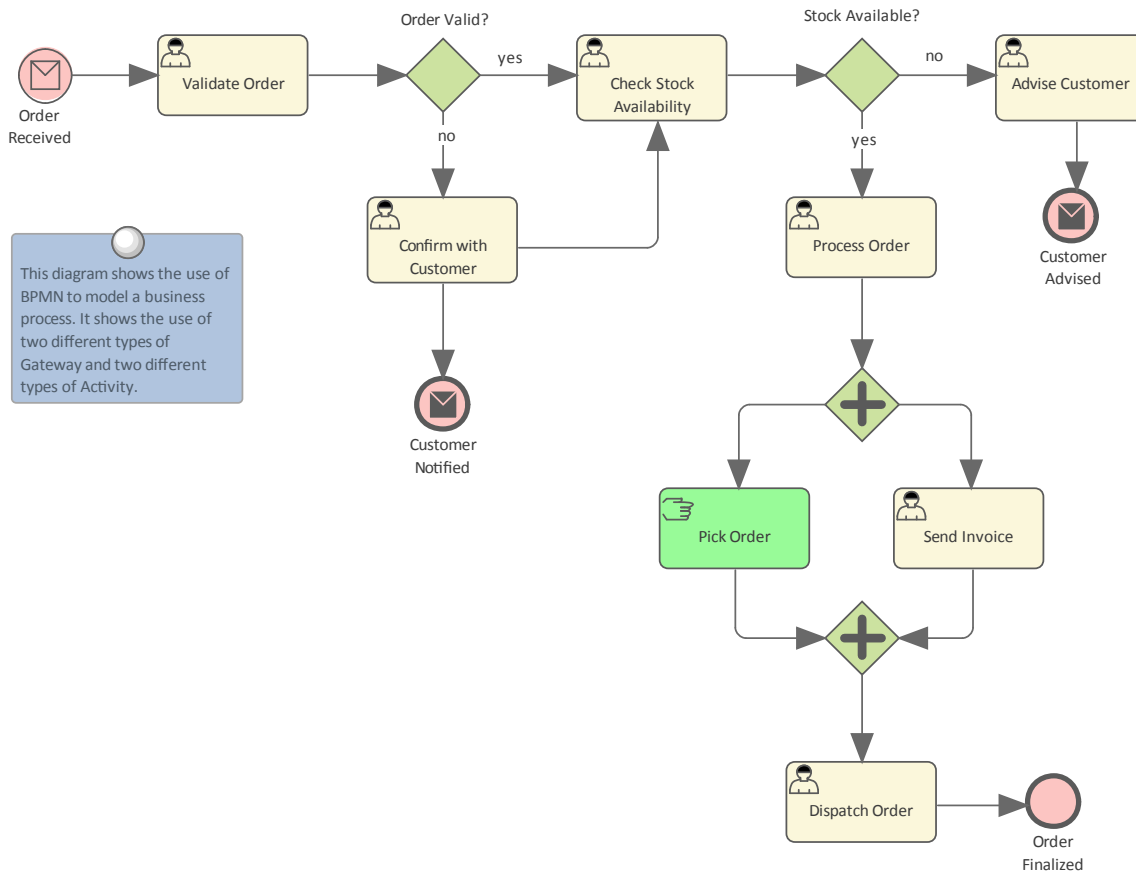
Element Maintenance Items can be used with Business Processes to capture features, changes, documents, issues, defects and tasks that affect individual Activities or whole Processes.

Learn More: [Maintenance Items](#)

# Process Modeling

A Process model can be used to model a wide range of sequential activities, including business processes, system processes or even the flow through an algorithm in a programming module. Enterprise Architect supports a variety of ways to model processes, including the Unified Modeling Language (UML) Activity diagrams, Business Process Model and Notation (BPMN) Business Process diagrams, and Flow Charts as part of the Strategic Diagramming set. These can be drawn at any level; drill down functionality is provided, allowing you to click through from a high-level diagram such as a Value Chain down to the task-level representations. The process elements can be linked to a wide range of other elements, including Business Rules, Policies, Standard Operating Procedures, and Use Cases or User Stories.

The tool supports both Current State and Future State modeling; transitions can be modeled showing the time sequence of changes between the current state and the future state. Enterprise Architect also has the ability to run model simulations from these models, allowing powerful insights to be gained that will assist with improving efficiency and effectiveness.



Enterprise Architect supports a number of different languages and techniques for performing Process Modeling. The choice of language will depend on what has been defined as a standard by the organization, or pragmatic considerations such as what the customer or subject matter experts will be most comfortable with.

## BPMN Business Process Diagram

Business Process diagrams are part of the Business Process Model and Notation (BPMN) standard and allow a modeler to document a business process, including the way the process starts, what work is performed and how it ends. Gateways and connecting lines determine the sequence of activities. Current state and Future state process models can



be created and managed in Enterprise Architect. The diagrams can be organized into a process hierarchy, allowing drill down from high level to lower level diagrams. BPMN has long been an important standard for modeling business processes and is widely used by business and technical communities. It can be automatically generated to the Business Process Execution Language (BPEL), which is an XML-based language that can be ingested by a number of orchestration engines.

Learn More: [Business Process Modeling](#)

## UML Activity Diagram

Activity diagrams are one of the Behavioral Unified Modeling Language diagrams. They allow you to describe the sequence of behaviors, including how they start, what work is performed and how decisions change the flow and the way the process ends. They are a useful alternative to using other diagrams, such as Flow Charts and Business Process diagrams. The syntax of Activity diagrams when Actions and Pins are used can be drawn at the execution level and to express detailed system semantics.

Learn More: [Activity Diagram](#)

## Flow Charts

A Flow Chart is a general purpose diagram for representing a sequence of steps and decisions, and can be used to represent a process, workflow or algorithm. Connecting lines determine the sequence of the steps in the Flow Chart. Flow Charts are an easy to understand representation of the steps in a process. The syntax of the diagrams is much easier to understand than their more advanced cousins, the **BPMN Business Process** diagram and the UML Activity diagrams. Learn More: [Flow Chart Diagram](#)

## ArchiMate Process Diagram

Allows the high level description of a business process, showing what triggers the process and the order of processes. If detailed representations are required, **BPMN Business Process** diagrams or UML Activity diagrams should be used.

Learn More: [ArchiMate](#)

## Data Flow Diagram

A Data Flow diagram provides a diagrammatic representation of the way data (information) flows through an information system. It shows where the data originates and where it is consumed and stored. These diagrams typically show the functions (or processes) that operate on

the data but do not indicate timing or sequencing.

Learn More: [Data Flow Diagrams](#)

## Business Rule Analysis

Business Rules are often associated with the Activities described in a process model and best practice suggests they should be managed separately from the Activities but they should be linked to the Activities they apply to.

Learn More: [Business Rule Model](#)

## Organizational Chart Diagram

The activities (steps) in a business process are typically performed by a role in the organization (or by a system on behalf of a role). Thus an Organizational Chart will provide a useful source for these roles and will help the analyst to express the relationship between the steps in the process and the roles that perform them. It is also common for a business unit or stakeholder to be responsible for a process, and these units or roles can be related to the Activities either using diagram relationships or Tagged Values.

Learn More: [Organizational Chart Diagram](#)

## Baselines

The Baseline Tool is used as a type of comparison and backup and recovery tool. The tool can capture a snapshot of the Process Models at a point in time and then at a later time the repository can be compared to this (or another Baseline) for the purpose of determining what has changed. It is possible to revert the current model to a state captured in the Baseline at the level of a granular change. This is a useful tool when working with Processes or any other model that typically has sign-off milestones. **Baselines** are typically taken at important milestones and given a memorable name such as 'After Stakeholder Final Workshop'. Baselines can be applied at the Package level and can be applied regardless of what type of elements are contained in the Package.

Learn More: [Baseline Tool](#)

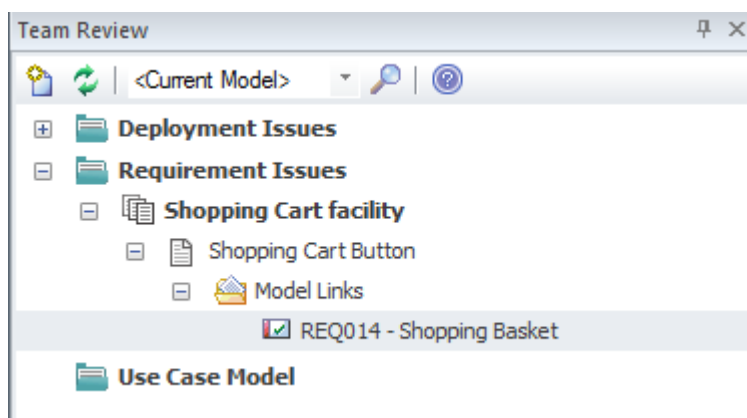
## Visual Filters

Visual Filters can be used to hide or gray-out parts of the diagram so that other parts can be emphasized. This is particularly useful when presenting a Business Process diagram to a team in a meeting or demonstration. Context Filtering is easy to set up and will emphasize the currently selected diagram element and its directly connected elements.

Learn More: [Visual Filters](#)

# Reviews

Enterprise Architect has a flexible review facility called a **Team Library**, which allows any type of review to be conducted, including inspections, formal and informal walk-throughs, single issue reviews, desk checks and pass-arounds. One of the powerful aspects of the Team Library facility is that it is right in the model, and elements - such as the work product being reviewed or related artifacts - can be attached to the review and reviewers and others can open them from the review window. Any amount of text can be added in a document editor, allowing reviews to be described and published.



## Team Library

The **Team Library** facility is a powerful tool for assisting with a review. Creating and recording a Team Library document directly in the model is useful as it allows a reviewer to include links that reference parts of the repository, such as Requirements or Components. There is a

word processor document available that can be used to describe aspects of the review, and a template can be used to define the structure of the document. Any number of categories can be set up for the reviews, and topics and documents can be added to these to form a hierarchy. Resources such as XML files and Images can be added under the Category, Topic or Document. Model Links allow elements and diagrams to be dragged and dropped from the **Browser window**, creating links to these elements from the review.

Learn More: [Team Library](#)

## Calendar

The **Calendar** provides a way to record and present important temporal information, such as events and meetings, in calendar format. Reviews can be entered into the Calendar and set up as recurring if needed. Event subtypes can be configured to add 'Review' as a Meeting Type.

Learn More: [Calendar](#)

## Model Mail

**Model Mail** can be used to notify and remind people of a Review and to notify them after the review about outcomes

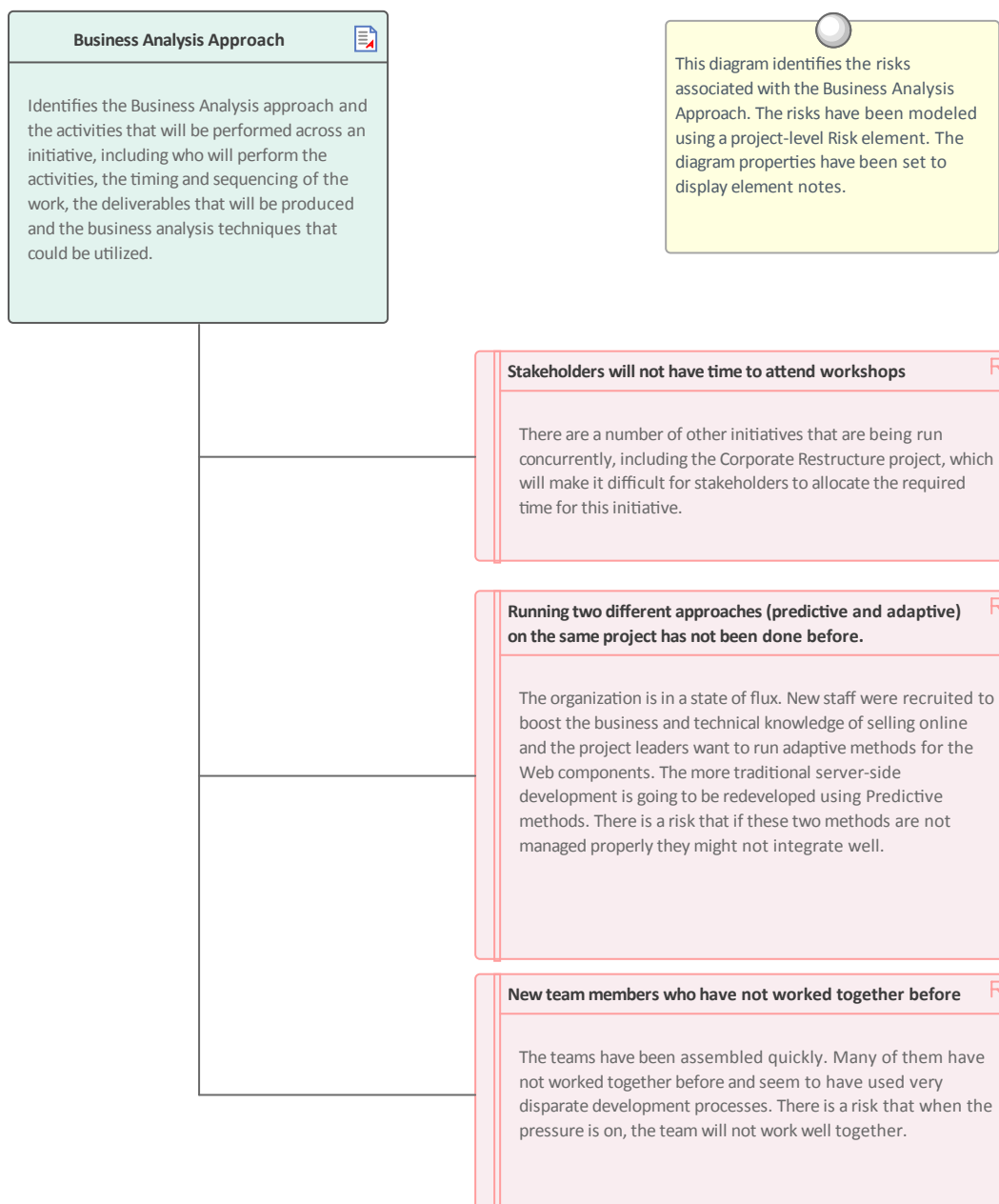
and decisions and required actions. Links to model elements and diagrams, matrices, team reviews and a variety of other items can be added.

Learn More: [Model Mail](#)



# Risk Analysis and Management

Enterprise Architect supports modeling risks at the project or the element level. A Project Manager or Senior Analyst will typically define risks at the project level, whereas a Business Analyst - while they might assist with project risks - would typically record risks at the level of Requirements and modular units of the system such as Components.



## Project Management Items

Risks can be assigned to any element in the repository. They can be managed through the Risks Window, which is part of the Project Maintenance group of windows.

Learn More: [Project Management Items](#)

## Risk Taxonomy

The **Risk Taxonomy** can be used to formally define risks and associated metadata applied universally across one or more initiatives. This includes definitions of Threats, Loss type, Contact Frequency, Loss Magnitude, Risks and more.

Learn More: [Risk Taxonomy](#)

## Requirements Diagram

A Requirements diagram can be created and any number of Risk elements can be added to the diagram. Modeling the risks using an element allows the Risks to be connected to other modeling elements by diagram relationships. This can be useful when the Risk applies to more than one element or

when it is of concern to one or more stakeholders.

Learn More: [Requirements Diagram](#)

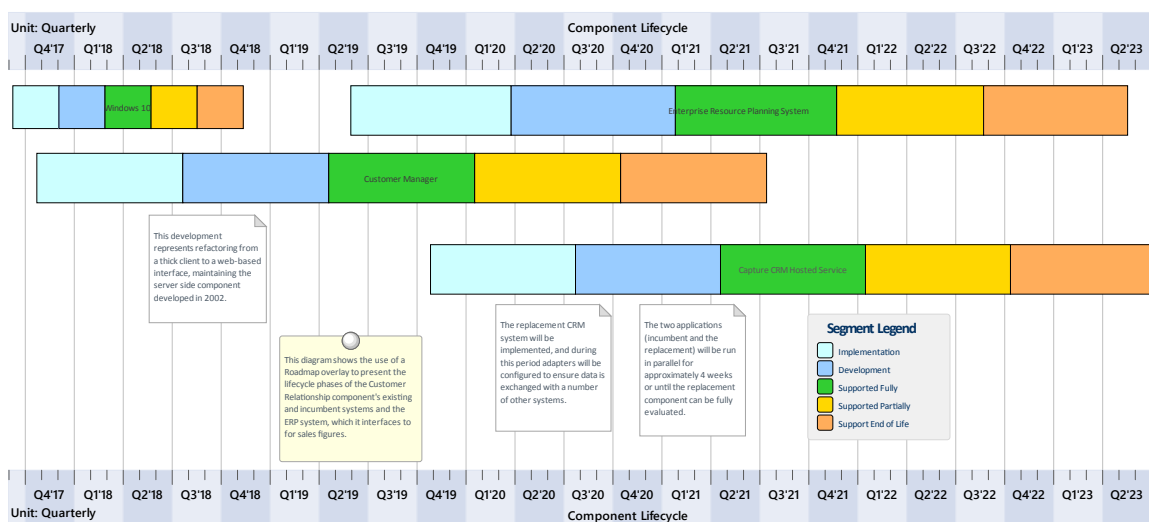
# Roadmaps

Roadmaps are a visual representation of the sequence of changes that must occur when transitioning between two states of an architecture - typically a baseline architecture and a target architecture or an intervening transition architecture. If an enterprise architect is looking for a diagram that will excite executive level stakeholders and senior managers, it is the Roadmap diagram, as it is a tangible deliverable that describes what needs to be done and when.

Each architecture domain will typically have a series of Roadmaps that describe the transition between architectures, so there will be Business Roadmaps, Information Roadmaps, Application Roadmaps and Technology Roadmaps. Each one of the domain specific Roadmaps will contain elements pertinent to that domain. So a Business Roadmap might show how Capabilities will be established, changed or phased out. Similarly, an Application Roadmap might show when one or more applications will be retired, replaced or refactored, or when new applications are introduced. A Roadmap could be created that incorporates elements from some or all of the architecture domains to give an enterprise level view of the sequence of changes. Enterprise Architect has a powerful and flexible Roadmap facility that can be applied as an overlay to any diagram to show how elements of the architecture change over time, including colored bands that represent the state of an

element at a given point in time. The meaning and color of each band can be assigned. The position of the element with respect to the diagram's time scale can be altered, and the length of the element and the bands can be changed to express when a particular stage in the element's lifecycle is started and finished. There is a wide range of options available to configure the look of the diagrams, including the timeline properties and appearance, units, tick spacing and more.

This section lists the main tools available in Enterprise Architect that can be used to create and manage Roadmaps. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Roadmap Diagrams

The Roadmap is an overlay that can be applied to any diagram and can be conveniently enabled or disabled. The overlay presents a timeline that is used to indicate the starting time, **duration** and finishing time of important phases in the lifetime of the elements on the diagram. It is particularly useful for describing transitions between a baseline and a target or transition architecture. Any diagram object can appear on a Roadmap diagram, including Business Capabilities and Capability Increments, Applications and Services, and Physical or Virtual servers described in a Technology Architecture. The appearance and properties of the timeline can be altered to set the Units, Tick Spacing (Year, Months, Days), Start and Finish time, Colors, Fonts and more. The phases on the elements can be configured in a Diagram Legend, allowing names and colors to be defined.

Learn More: [Roadmap Diagram](#)

## Diagram Legends

Diagram Legends can be used with any diagram but they have a particular function with Roadmap overlays to define the phases in the lifetime of the diagram objects. Any number of phases can be added and a unique color can be assigned to each phase. The diagram elements will automatically appear with colored bands that can be adjusted in length to indicate the **duration** of the phase with respect to the timeline. The option 'Use Legend for phase

colors' on the 'Roadmap' dialog must be set for the Legend to function in this way.

Learn More: [Diagram Legends](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. A Roadmap diagram that is stored in the repository can be conveniently generated and included in documentation. The formatting in element or diagram notes can also be carried through to the documentation. A sophisticated template facility exists that provides a range of built-in templates and also allows the user to create their own templates, defining styles, cover pages, tables of contents, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Visual Filters

Visual Filters can be used to obscure or hide elements on the diagram based on user-defined criteria. This is particularly useful with detailed Architectural Roadmap diagrams that are used to communicate ideas to a wide range of stakeholders. Any number of filters can be created and used

to focus attention on a particular part of the diagram, for example to show only applications or capabilities that are being implemented in a particular phase of the architecture or that have a specified status. The filters can be saved and reapplied to the same or any other diagram.

Learn More: [Visual Filters](#)



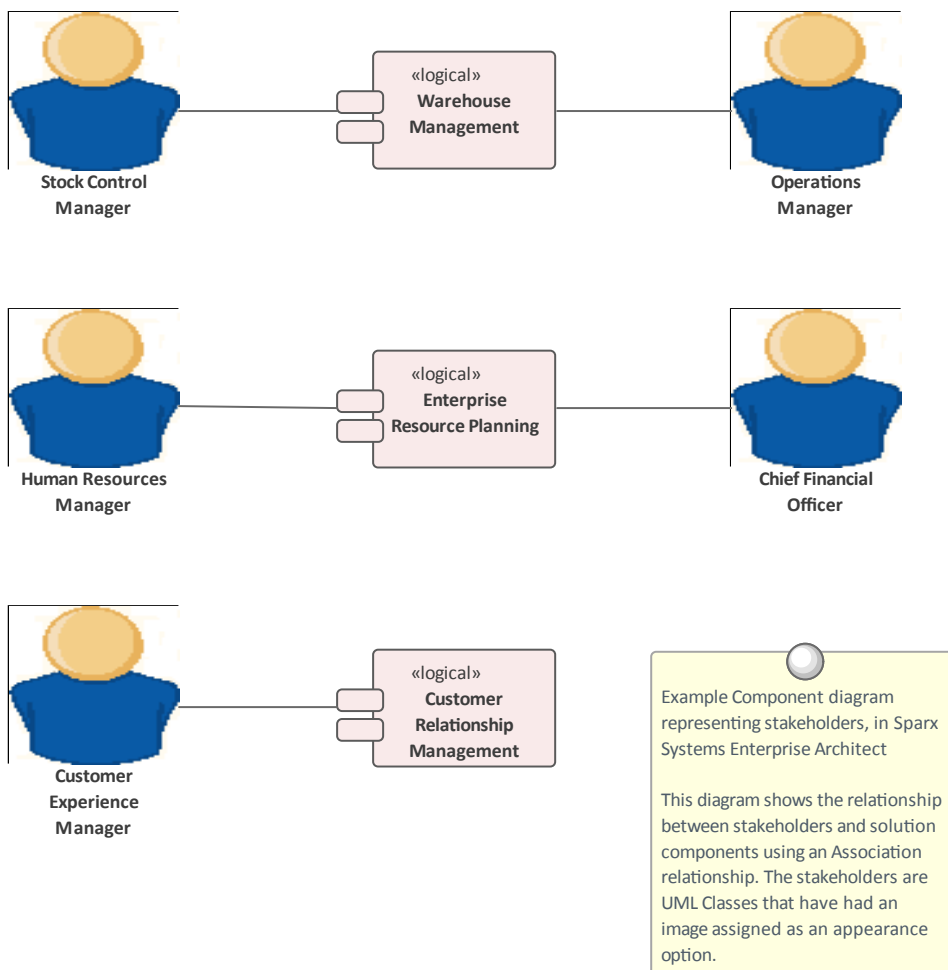
# Stakeholder Management

Stakeholder Management is critical to successfully initiating and maintaining an architectural program. Architects will need to engage with a wide range of stakeholders from senior executives down to implementation staff. The engagements will often require political sensitivity, diplomacy and flexibility to ensure the stakeholders' needs and concerns are dealt with appropriately. Providing relevant and tailored views of the architectures will be critical to ensuring that the stakeholders are kept informed and that they spend the time needed to understand the impact the architectures will have on their domain. Having a comprehensive communication plan is critical to ensure that the stakeholders receive the information they require and to maintain an interest and input into the architectures.

Enterprise Architect has a wide range of facilities and tools that can assist with Stakeholder Management. This includes the ability to model both the individual and groups of stakeholders, to classify them in a taxonomy, and to show the extent of their influence by using a series of nested Boundary elements. There is a wide range of diagrams, matrices and lists that will be relevant to certain stakeholders, including 'Lists' presented in the **Specification Manager**, Component diagrams describing Applications, and Class diagrams used to present the information architectures, principles and a range of other ideas. The **Calendar** and Collaboration facilities are useful tools for keeping stakeholders informed about things of interest and

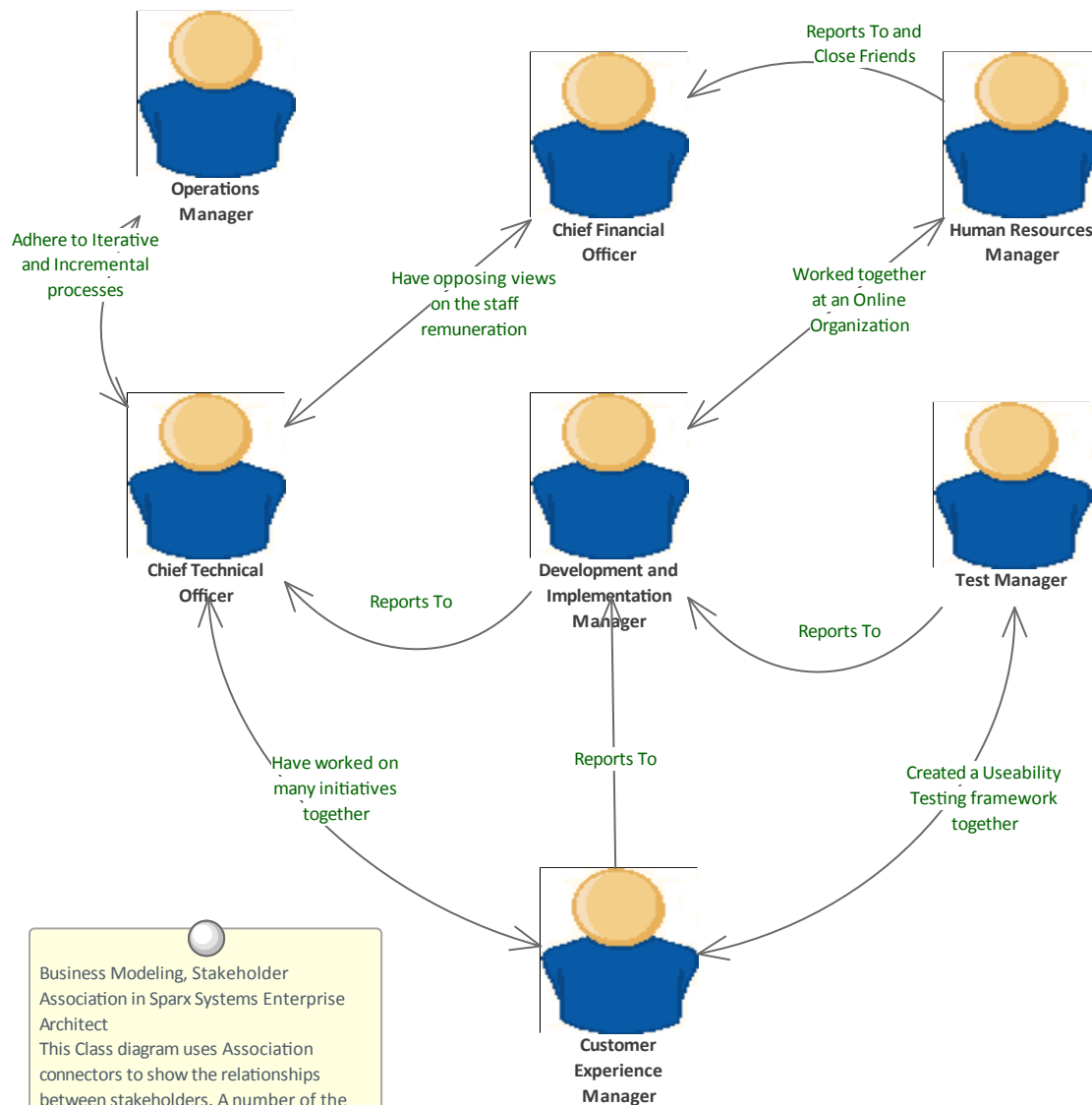
important events in the architecture program.

This section lists the main tools available in Enterprise Architect that can be used to perform Stakeholder Management. There is also a wide range of additional tools that you might find suitable when applying the technique in a particular context. The complete list of tools is available in the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



The Class diagram can be used to create visual representations of the stakeholders, including how they are related to each other. An alternative image can be used for each element to make the diagrams more appealing and to visually represent stakeholder groups. Class diagrams are also useful for presenting parts of the information architecture, such as conceptual and logical models. Principles can be displayed and communicated using a Class diagram, and their relationships to other parts of the model can be demonstrated.

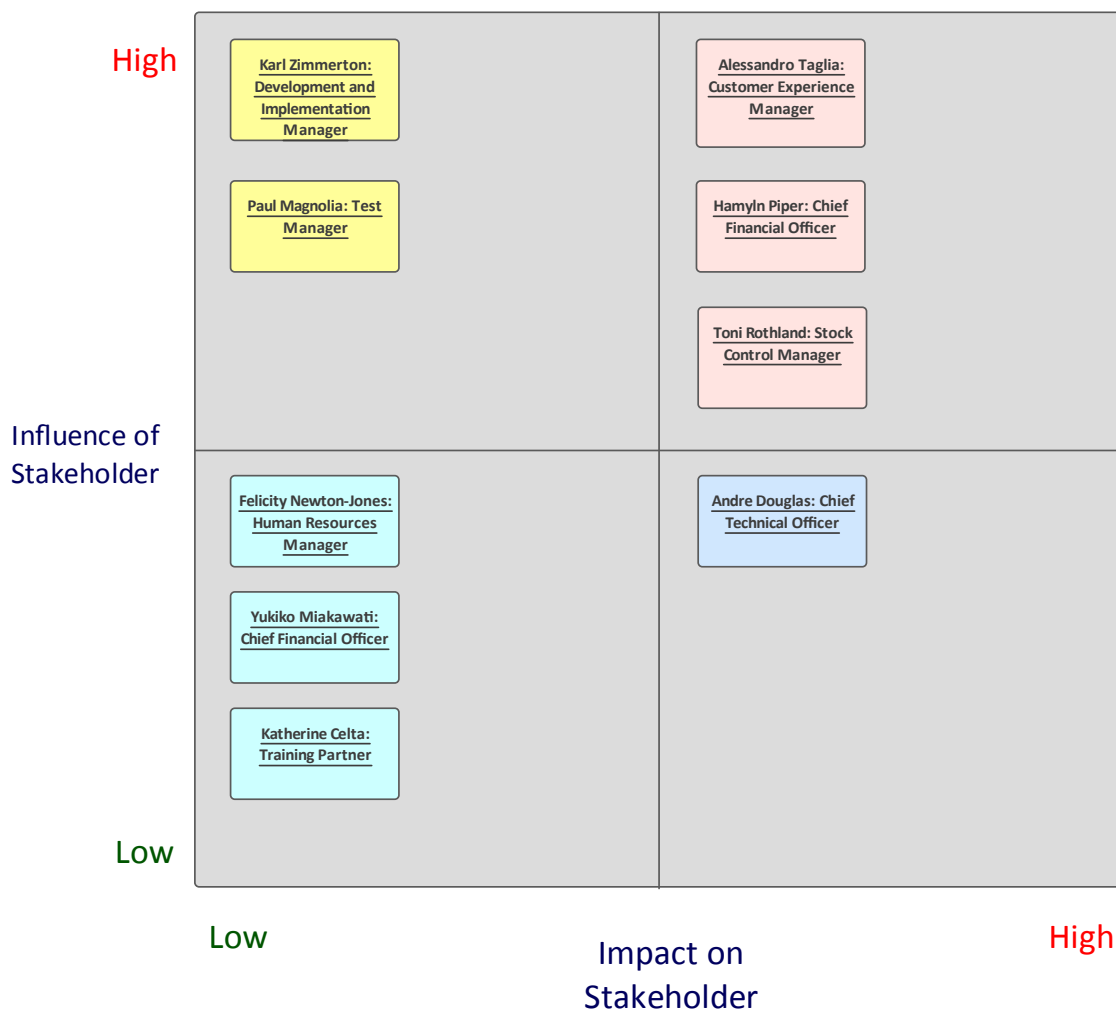
Learn More: [Class Diagram](#)



Business Modeling, Stakeholder Association in Sparx Systems Enterprise Architect

This Class diagram uses Association connectors to show the relationships between stakeholders. A number of the connectors have a Tagged Value (`_bezier=true`) assigned to allow them to be bent in a curve. The Association names indicate the nature of the relationship, and the arrows indicate the direction of the relationship.

## Stakeholder Influence x Impact Matrix



Business Modeling, Stakeholder Management in Sparx Systems Enterprise Architect

This diagram maps the level of stakeholder influence against the level of stakeholder interest. The size of the Boundary element can be changed to accommodate more stakeholders. Instances of the Stakeholders have been used to indicate the person who occupies the role.

## Component Diagram

The Component diagram can be used to show the relationship between stakeholders and the parts of the solution that affect them or that they have an interest in or a concern about. The diagram could explain the reason for

their interest, which helps Analysts and others to notify the stakeholders when important milestones are reached. They can also be useful to show any number of Applications or Interfaces in the context of other parts of the architectures.

Learn More: [Component Diagram](#)

## Boundary

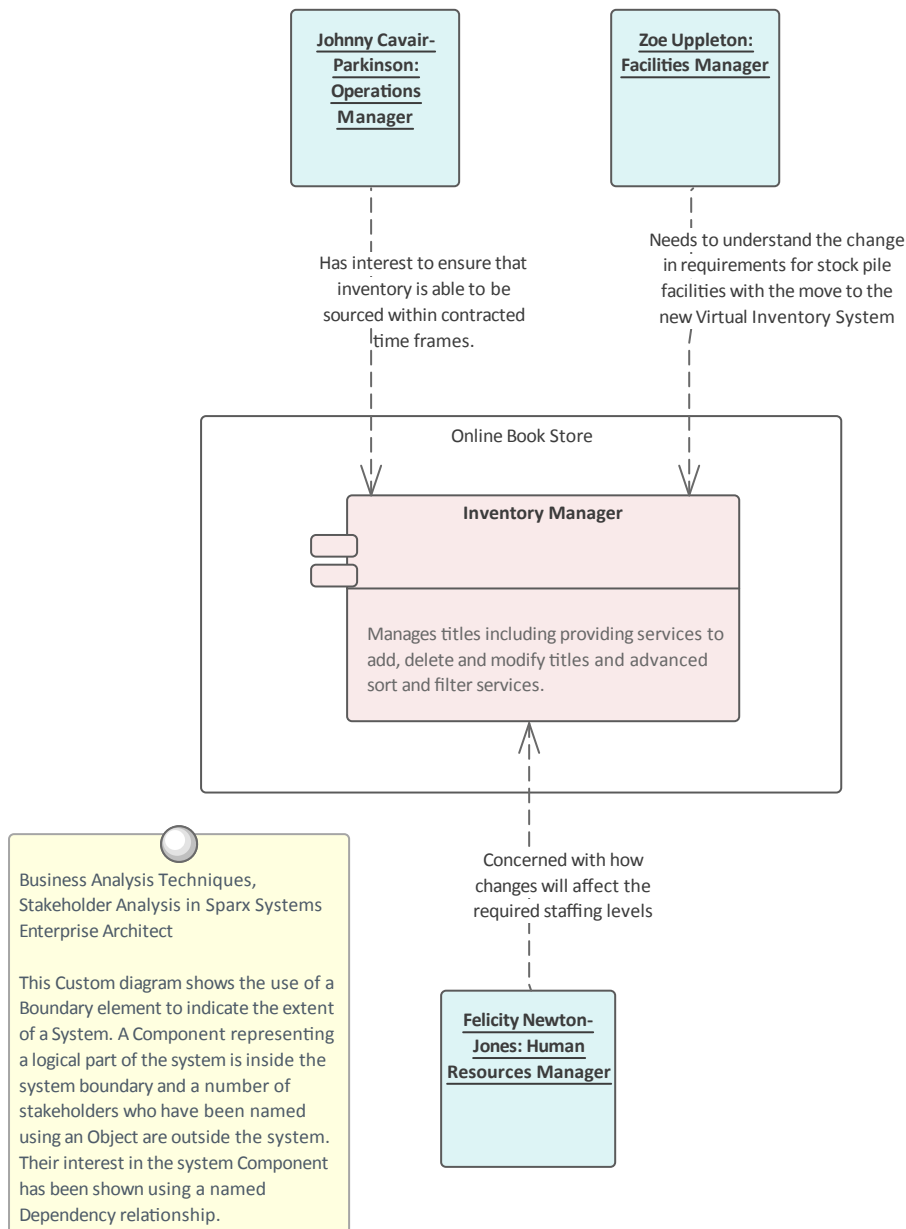
The Boundary element can be used to group stakeholders visually. This is a powerful mechanism and can be used to create an onion diagram showing the relative influence that groups of stakeholders have on an initiative.

Learn More: [Boundary](#)

## Element Appearance

Many of the stakeholders will be managers or non technical staff who will benefit from viewing diagrams with graphics. Images from the standard Image Library could be used, or your organization can create their own graphics to represent individual stakeholders or stakeholder groups.

Learn More: [Element Appearance](#)

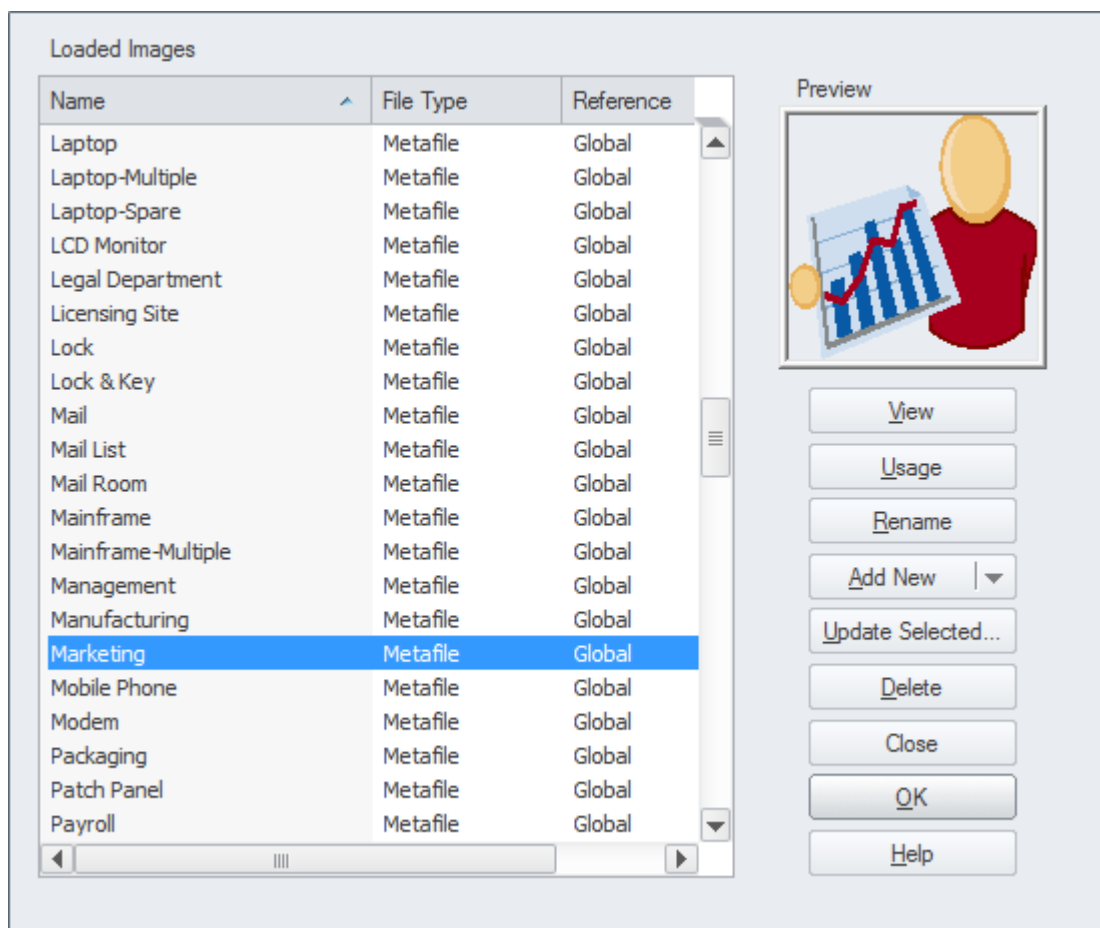


## Image Manager

The Image Library is a global store of images in the repository that can be applied to any number of diagram elements. The images can be in a variety of formats and when used to change the appearance of an element they create a compelling diagram that is often more appealing to

managers or non-technical audiences. Where possible it is recommended to store a vector-based image so that it can be effectively scaled in a diagram. The images can also be used when defining the metafile for a stereotype.

Learn More: [Image Manager](#)



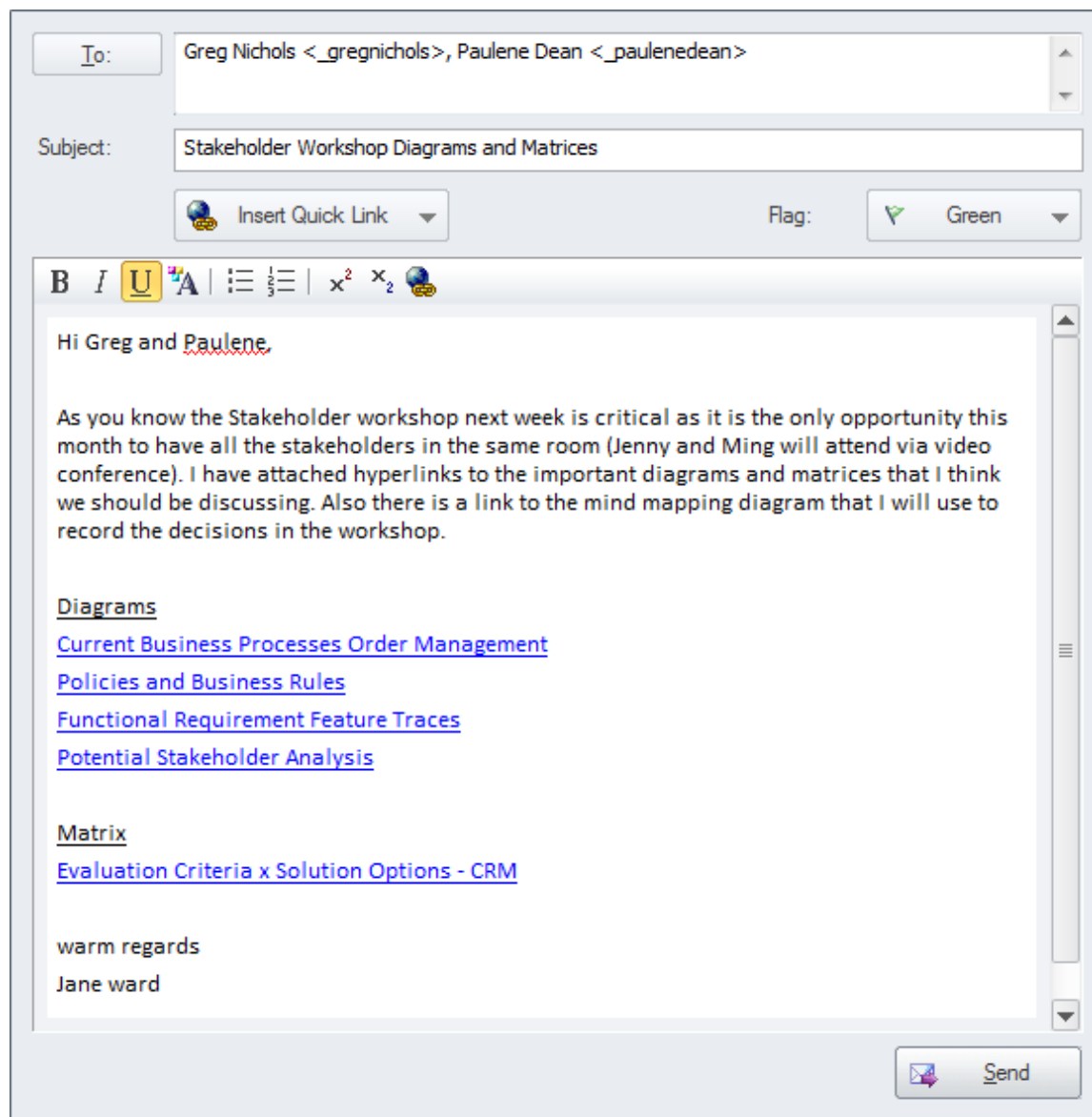
## Model Mail

**Model Mail** can be used to communicate with stakeholders, to alert them of important events that are relevant to them and advise them of outcomes, decisions and required actions. Links to model elements, diagrams, matrices, Libraries and a variety of other items can be added to a mail



message.

Learn More: [Model Mail](#)



## Organizational Chart Diagram

An Organization Chart Diagram is a useful tool to show the structural and organizational relationship between stakeholders and to understand reporting lines in an enterprise or department.

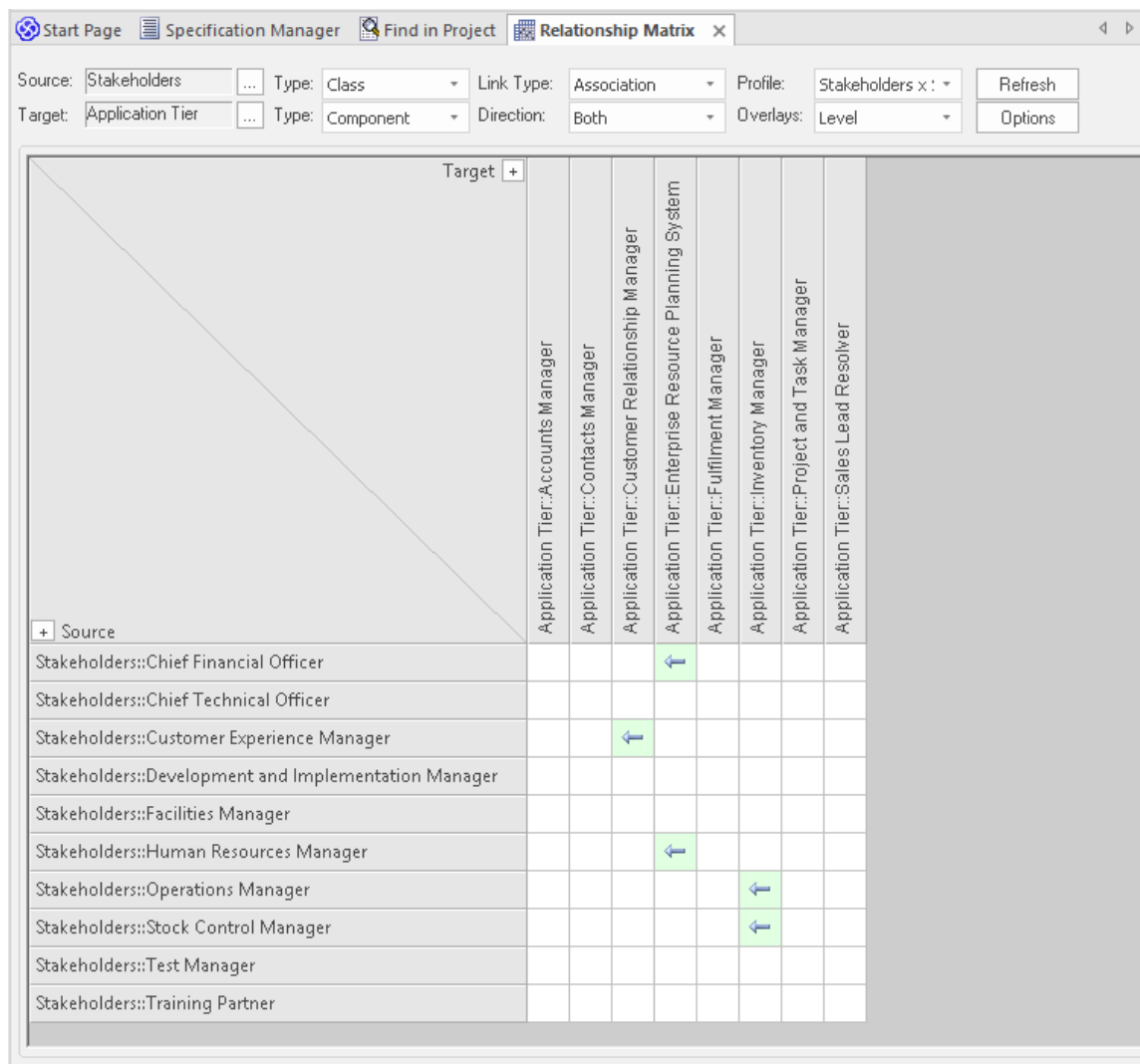
Learn More: [Organizational Chart Diagram](#)

## Relationship Matrix

A **Relationship Matrix** can be used to present a number of views of the stakeholders. The matrix is a spreadsheet view that is often more appealing to many managers and non-technical audiences. Any number of matrices can be created that relate stakeholders to other model elements, including other stakeholders.

Learn More: [Relationship Matrix](#)

Source	Stakeholders::Chief Financial Officer	Stakeholders::Chief Technical Officer	Stakeholders::Customer Experience Manager	Stakeholders::Development and Implementation Manager	Stakeholders::Facilities Manager	Stakeholders::Human Resources Manager	Stakeholders::Operations Manager	Stakeholders::Stock Control Manager	Stakeholders::Test Manager	Stakeholders::Training Partner
Stakeholders::Chief Financial Officer										
Stakeholders::Chief Technical Officer										
Stakeholders::Customer Experience Manager										
Stakeholders::Development and Implementation Manager										
Stakeholders::Facilities Manager										
Stakeholders::Human Resources Manager										
Stakeholders::Operations Manager										
Stakeholders::Stock Control Manager										
Stakeholders::Test Manager										
Stakeholders::Training Partner										



## Requirements Diagram

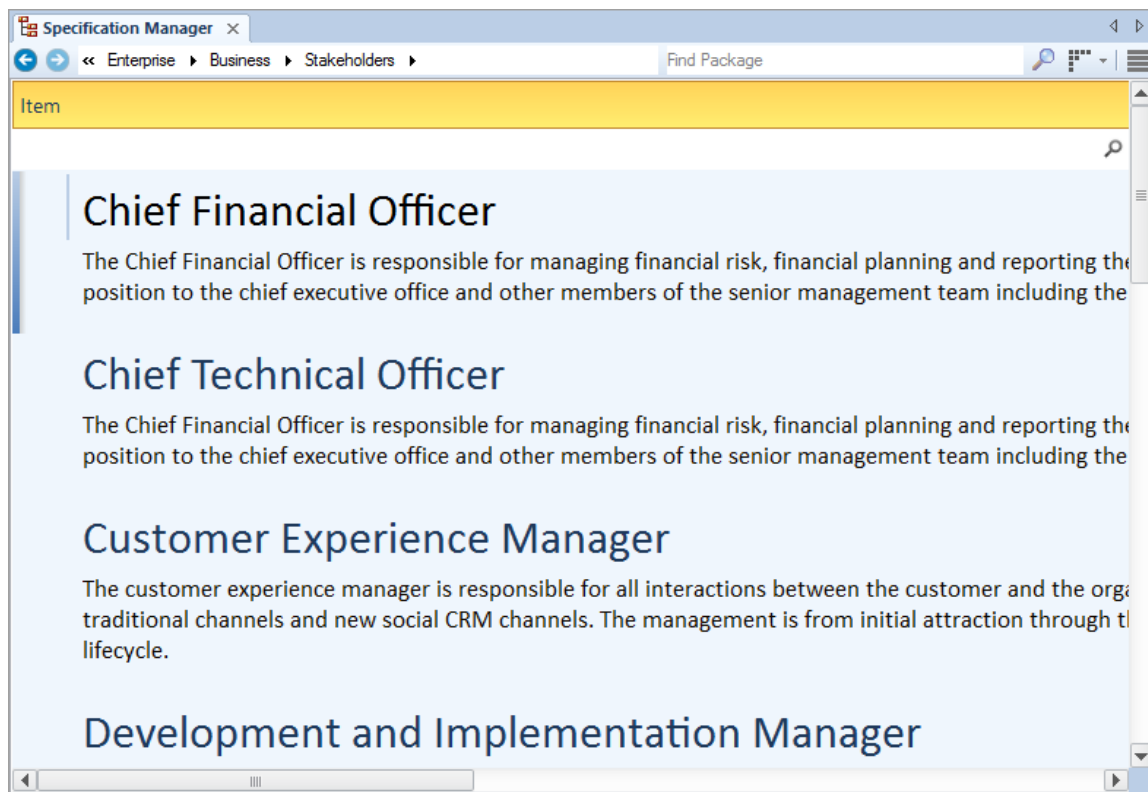
The Requirements diagram is useful for showing the relationship between stakeholders and their Requirements and allowing communication about the way the Requirements are managed. Diagrams that display Trace connectors between the Requirement and the solution Components will also be important for ensuring that the stakeholders understand how their problems or opportunities are being solved.

Learn More: [Requirements Diagram](#)

## Specification Manager

The **Specification Manager** is a useful tool for working with lists inside Enterprise Architect, which makes it the perfect tool for working with lists of stakeholders and personas.

Learn More: [Specification Manager](#)



## Calendar

The **Calendar** is a useful tool for communicating

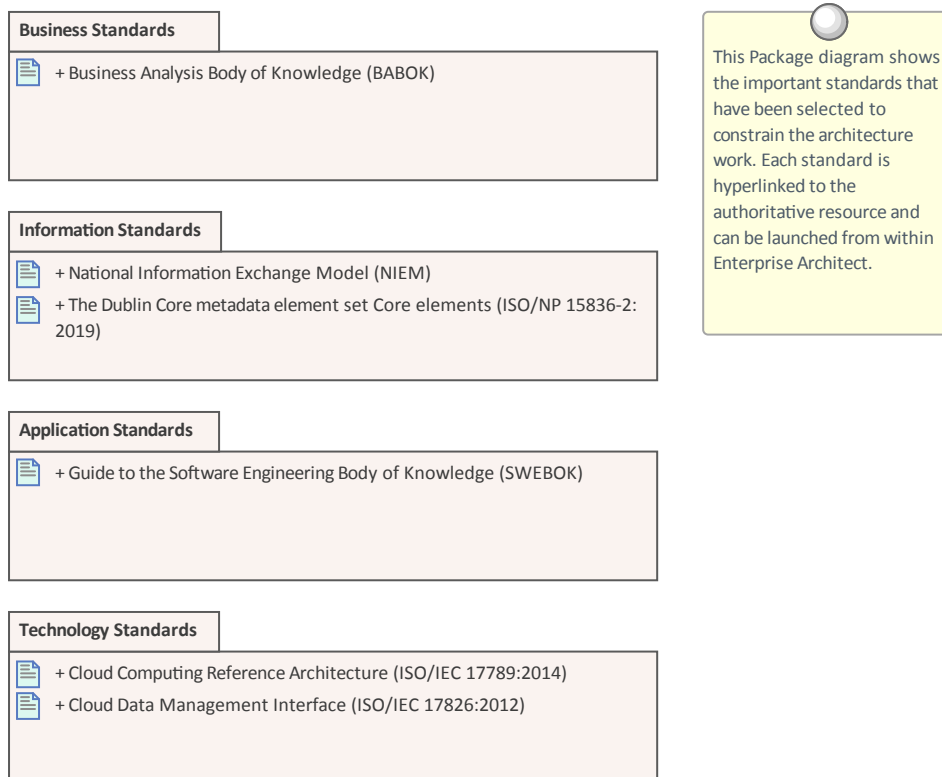
information about project milestones and events that will facilitate good communication with the stakeholders. This includes providing hyper-linked references to the parts of the model - including matrices and diagrams - that are of interest to them.

Learn More: [Calendar](#)

# Standards Modeling

Standards are important specifications with which architectures should comply. They can be either organizational standards (internal to the organization), Industry Standards (governed by Industry bodies) or Regulatory Standards (mandated by international, national or jurisdictional law). All of these standards must be represented in the Architecture Repository and mapped to applicable parts of the architecture to indicate compliance or dispensation. The standards can be categorized further by architecture domain: Business, Information, Application and Technology standards. The process of standards adoption must be managed to show how the standards move through a lifecycle from selection to adoption and eventual retirement.

Enterprise Architect has a number of tools and mechanisms for modeling standards. The most important of these are the Artifact and Document Artifact, which can be used to model External and Internal standards respectively. The Artifact can be used as a surrogate for an external standard inside the model, which provides the advantage of being able to create a hyperlink to the external standard but also being able to relate the Artifact to elements that form part of one or more architectures. The Document Artifact is a word processing document that can be used to create and manage standards inside the repository. It can be created using system or user defined templates, and can be related to elements that form part of one or more architectures.



This section lists the main tools available in Enterprise Architect that can be used to perform Standards Modeling. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## Artifact

An Artifact can be used as a placeholder for an external standard such as a regulatory or Industry standard that is governed and stored in a remote location. The Artifact element can be hyper-linked to the external standard regardless of whether it is accessible as a file or a Web Page. This mechanism allows it to be launched from within

Enterprise Architect. This provides a convenient way to reference standards as the element can be dropped from the Standards Packages onto a diagram and related to elements to indicate compliance or that a dispensation has been granted.

Learn More: [Artifact](#)

## Document Artifact

A Document Artifact can be used to create a standard, conveniently storing the document inside the repository. A template could be specified for the document, which could be reused for other standards initiatives. This provides a convenient way to reference standards as the element can be dropped from the Standards Packages onto diagram and related to elements to indicate compliance or that a dispensation has been granted.

Learn More: [Document Artifact](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. A single standard or group of standards could be generated directly from Enterprise Architect incorporating diagrams matrices



and other artifacts into the document from the repository. This is useful when the Standards need to be provided to an external body such as an implementation partner.

Learn More: [Documentation](#)

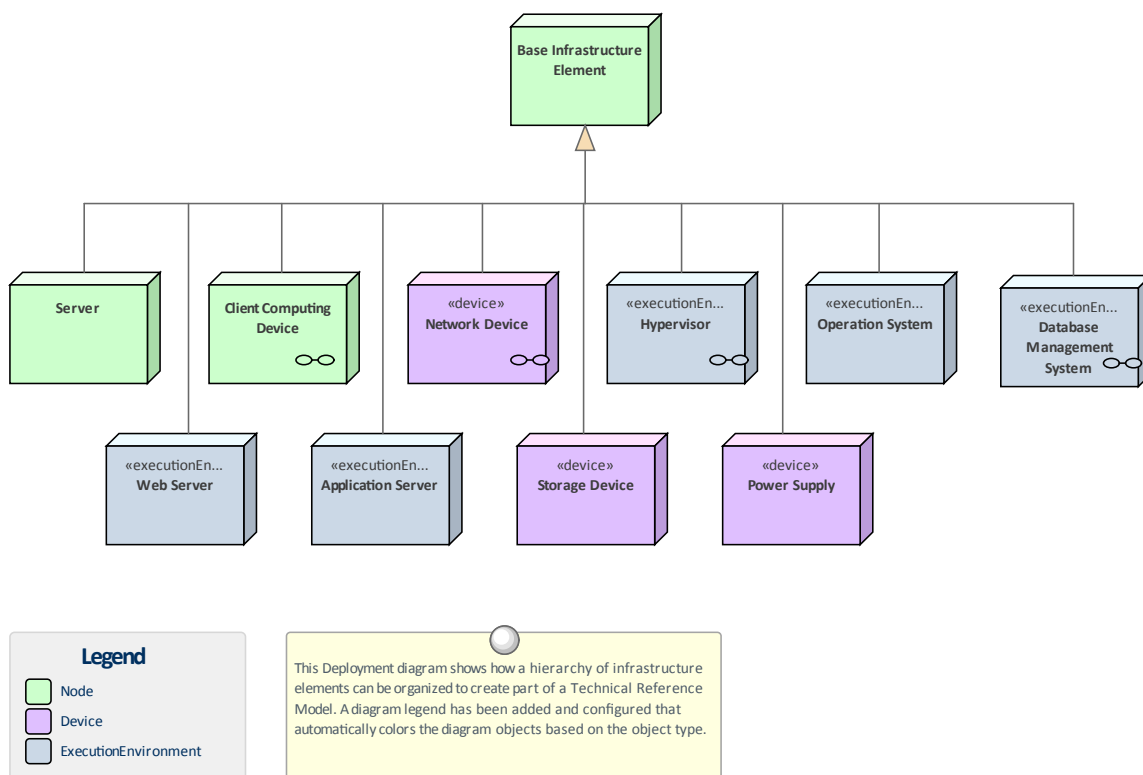
# Technical Reference Model

Architectures are typically not built from scratch, but rather architects make use of existing models as a foundation upon which new architectures are constructed. These models underpin the new architectures and create a consistency and foundation for any type of architecture regardless of the domain or level, from Strategic, Segment or Solutions architectures.

There are a number of published Technical Reference Models (TRMs) available, and while it would be judicious for an enterprise to adopt one of these it might be necessary to adapt the selected TRM to suit an enterprise's domain and level of maturity. The structure and the content of the TRM can often be derived from existing platforms and services, but will typically have to be augmented at various times to meet the demands of new or emerging architectural needs such as new services and interfaces.

Enterprise Architect has a wide range of tools that can be used to create a Technical Reference Model that acts as the foundation for all existing and new architectures. The most compelling of these tools is the Profile tool that can be used to extend the core language constructs and grammar of the Unified Modeling Language. Communication Infrastructure such as Servers can be defined that interface to Application Platforms, which in turn provide interfaces for Business and Infrastructure Applications. Once a profile is created it can be either imported into a repository or bundled together with

a number of other facilities into a Model Driven Generation (MDG) Technology. The visualization and documentation for the TRM can be created and viewed in the model itself or generated out as documentation.



This section lists the main tools available in Enterprise Architect that can be used to create and maintain a Technical Reference Model. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics: [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).

## Profile

The Profile is part of the Unified Modeling Language extension mechanism allowing a modeler to make extensions to the grammar of the language by using Stereotypes and Tagged Values to create domain, industry or organizational level elements. This mechanism can be used to create a Technical Reference Model by creating the foundational elements,, services, interfaces and connectors as stereotypes within the profile. These elements can also have Tagged Values added to capture additional properties. Toolbox pages can be created and groups of elements and connectors can be added that users can apply to create new elements.

Learn More: [Profile](#)

## Deployment Diagram

Communication Infrastructure such as Servers and Network devices can be defined that interface to Application Platforms, which in turn provide interfaces for Business and Infrastructure Applications. The Deployment diagram can be used to model all of these elements and can show the relationships between the different levels of elements in the Technical Reference Model, from the Communication Infrastructure and Application Platforms to the Application. Hardware Nodes and Network Devices need to interface to other Nodes and Devices through published Interfaces. These can be described with Unified Modeling Language

(UML) Ports that can be used to describe aspects of the interface such as IP Addresses and Protocols. These describe the physical or virtual connections between the various pieces of hardware that make up the system.

Learn More: [Deployment Diagram](#)

## Specification Manager

The **Specification Manager** is a powerful tool for working with lists or catalogs of elements, including the elements that make up the Technical Reference Model. It provides a word processor or spreadsheet interface for entering, maintaining and viewing elements. Servers, Network Devices, Operating Systems, Containers, Applications and Infrastructure, and Application Services and Interfaces can all be created and viewed as a catalog, and detailed descriptions and a range of properties can be added directly through the interface. Changing an element details in the Specification Manager will change them in all other places in the repository, such as Deployment and Component diagrams and windows. The Specification Manager can be used with groups of elements and shields a user from needing to know about the underlying representation, as they are simply working with a list as they would in their favorite spreadsheet or word processor tool.

Learn More: [Specification Manager](#)

## Working Sets

Working sets are a useful facility for working with Technical Reference Models, as they allow a series of views to be collected as a set, given a name, saved and reopened at a later time. They are useful for working with different stakeholders who might be interested in a particular set of views. The set can include diagrams, Matrix Profiles, **Team Library** and more, and any number of working sets can be created.

Learn More: [Working Sets](#)

## Component Diagram

A Component diagram can be used to describe any number of the Applications, Application Services or Interfaces that make up the Technical Reference Model. Expressive diagrams can be created and viewed that show the relationships between Components, including Interfaces and Ports that describe the services the application offers to its environment including other applications. Any number of diagrams can be created showing groups of applications that collaborate to provide a service that is ultimately required to meet a business service or capability. An enterprise of any appreciable size will typically have hundreds if not thousands of applications and their relationships will be

complex and varied. It is possible to create a large diagram that shows all these applications in one view but it is more common to divide the portfolio up into a number of groups of applications. Both logical and physical application components can be described.

Learn More: [Component Diagram](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. Any number of views of the repository content can be created by generating documentation and the powerful **Virtual Documents** facility allows content to be cherry picked from anywhere in the repository and grouped into sections presented with common template. The formatting in element or diagram notes can also be carried through to the documentation. A sophisticated template facility exists that provides a range of built-in templates and also allows the user to create their own templates, defining styles, cover pages, tables of contents, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Element Appearance

Many of the elements that make up the Technical Reference Model, such as servers and devices, are commonly represented by images that help the viewer to visualize the type of device or server that is being represented. Many of the manufacturers and Cloud providers make a set of images available that can be used as alternative representations of these elements. Images from the standard Image Library could be used, vendor images imported or the organization is free to create their own graphics to represent concepts in the Technical Reference Model or that have particular meaning to individuals or groups of stakeholders.

See also: [Element Appearance](#)

## Image Manager

The Image Library is a global store of images in the repository that can be applied to any number of diagram elements. There are built in libraries but for the Technical Reference Model it is common for organizations to import libraries of images that can be used to represent particular vendors or providers products and services. The images can be in a variety of formats and when used to change the appearance of an element they create a compelling diagram that is often more appealing than the geometric shapes of the modeling languages. Where possible it is recommended to store a vector based image so that it can be effectively scaled in a diagram. The images can also be used when



defining the metafile for a stereotype.

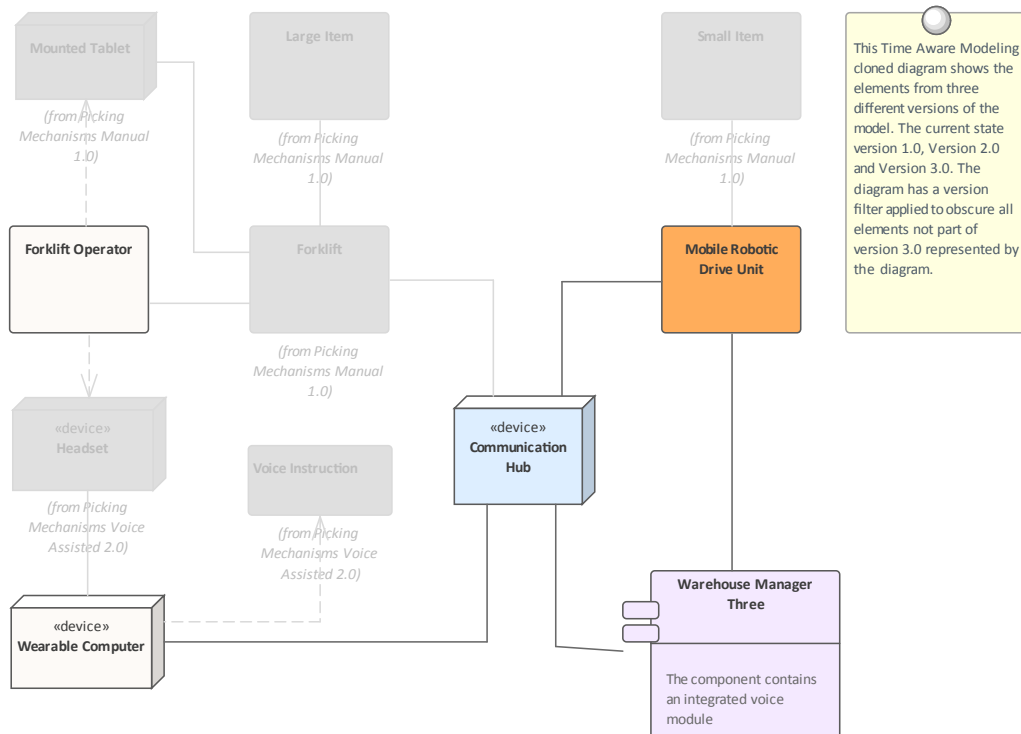
Learn More: [Image Manager](#)

# Time Aware Modeling









Time Aware Modeling is a powerful and expressive technique for defining and visualizing how models change over time. The technique allows different versions or states of a model to be created to represent how the model will change with the passage of time. This includes elements and features that are changed, deleted or added at a point in time. The technique works by allowing a modeler to clone a Package, diagram or element in a model. When this is done, the complex task of maintaining the relationships between previous versions is managed by Enterprise Architect.

This compelling technique can be used by a variety of modelers including analysts, architects and product owners, to describe and to visualize how models change over time. For example, an enterprise or solution architect might want to describe transition architectures that move the organization from a baseline ('As-Is', current state) to a target ('To-Be', future state) architecture or that describe architectural options. The target state models can be visualized as an evolution of the baseline models without in any way altering the baseline models.

### Integrated Mobile Robotic Drive Units and Voice Assisted Forklifts Version 3.0 Q4 2019

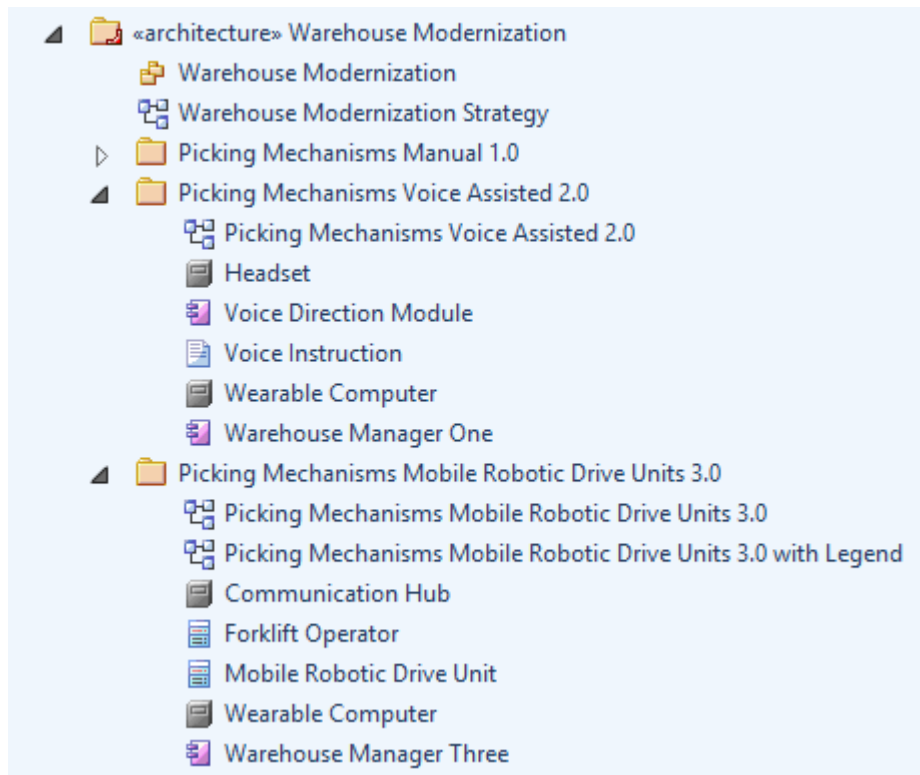


There is a wide range of tools that can be used in conjunction with Time Aware Modeling, such as the ability to filter a diagram to only highlight elements that are part of the version represented by the diagram. The effect of this will be to obscure all elements other than those that have been added or cloned into the specified version. The **Traceability window** can be used to keep track of the complex relationships that exist between different versions of the elements. Diagram Legends can be used to color the diagram's elements based on the version they belong to, effectively creating a colored chronology of change or transition from version to version. Diagram Filters can be used to obscure elements based on more complex criteria, such as to display only the elements that are part of either version 2 or 3 and that have a status of 'Approved'.

List of Elements in Diagram Picking Mechanisms Mobile Robotic Drive Units 3.0			
	Name	Status	Version
		Proposed	3.0
	Communication Hub	Proposed	3.0
	Forklift	Validated	1.0
	Forklift Operator	Proposed	3.0
	Headset	Validated	2.0
	Large Item	Proposed	1.0
	Mobile Robotic Drive Unit	Proposed	3.0
	Mounted Tablet	Proposed	1.0
	Small Item	Proposed	1.0
	Voice Instruction	Approved	2.0
	Warehouse Manager Three	Proposed	3.0
	Wearable Computer	Proposed	3.0
Showing 1 - 12 of 12 items			

## Cloning Packages Diagrams and Elements

Cloning is the primary mechanism for creating Time Aware Models and allows an entire Package, individual diagram or element to be cloned. Typically a Package is cloned that will copy the structure and diagrams to a new Package, which can be assigned a new version number. Elements in the cloned diagrams will not be available for change until they are cloned. When they are cloned Enterprise Architect creates and maintains a Trace relationship back to the original version of the element; changes can then be freely made without affecting the original version of the element.



Learn More: [Time Aware Modeling](#)

## Diagram Filters

Diagram Filters can be used to obscure parts of the model, leaving only the significant elements highlighted in the diagram or visible in a List or Gantt chart view of the diagram elements. This tool is particularly useful with Time Aware Modeling, as complex criteria can be added to the filters allowing modelers to specify a number of conditions, such as all of the version 2 and 3 elements that don't have a status of Validated.

Learn More: [Diagram Filters](#)

## Traceability Window

The **Traceability window** can be used to view all the relationships that a selected element participates in, even when they are not visible in a diagram. The relationships are presented in a hierarchy and a modeler can also explore the relationships of related elements. The tool is very useful when visualizing the complex relationships of Time Aware Models and will allow model elements to be traced back to their earlier versions without needing to create diagrams.

Learn More: [Traceability Window](#)

## Diagram Legend

Diagram Legends provide a way to manually or automatically relate elements and connectors in a diagram to a set of applicable colors. This provides a compelling way to automatically color the elements in Time Aware Model diagrams based on their version and additional properties. This allows the chronology or evolution of the models to be visualized in color and will be a welcomed tool during presentations and workshops.

Learn More: [Diagram Legends](#)

## Insert Related Elements

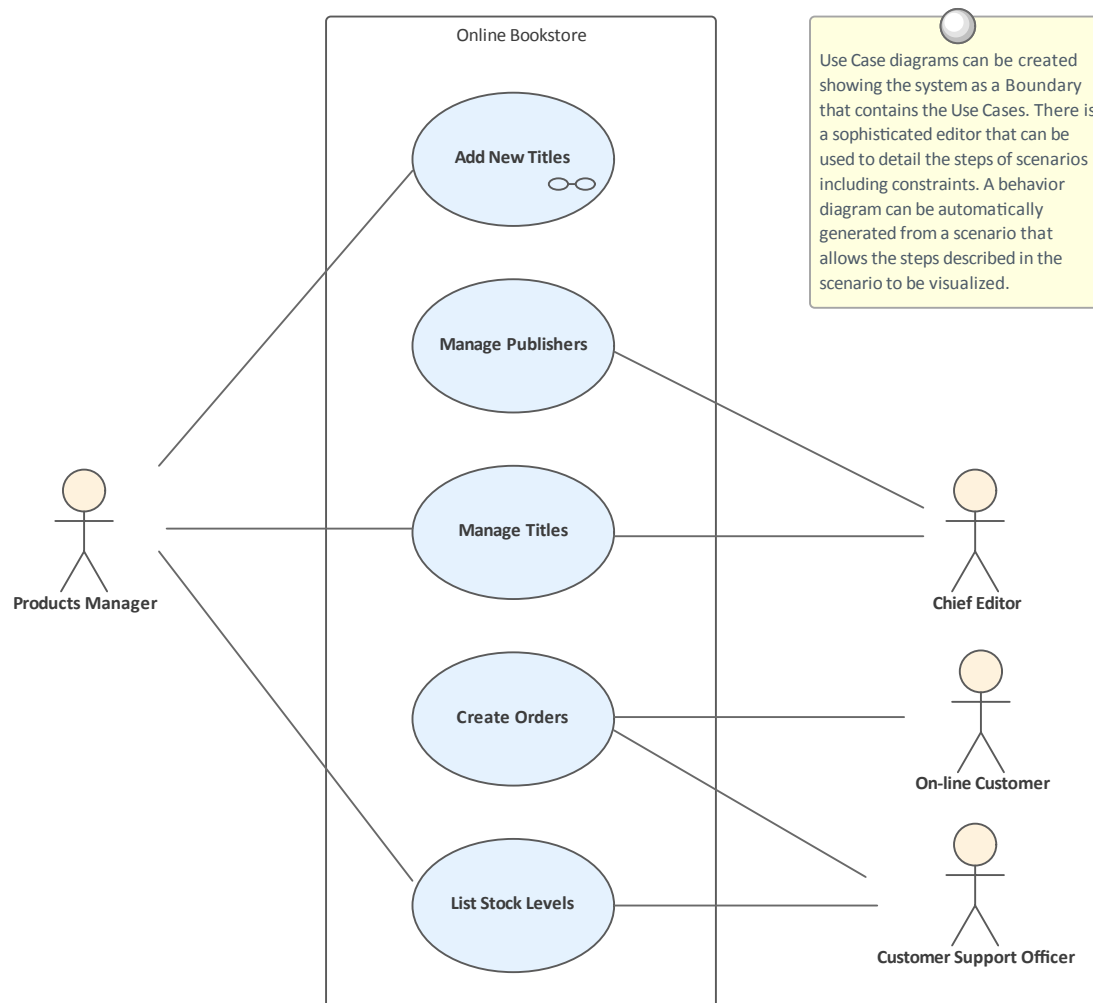
**Insert Related Elements** is a powerful exploration tool that can be used in conjunction with Time Aware Modeling to automatically create diagrams displaying the Trace relationships that exist between different versions of elements. By placing a single element onto a diagram to act as a context (or starting point), the Insert Related Elements facility can be used to explore the element's connections and to automatically draw a diagram using the context element as a starting point. This will allow an analyst or architect to explore how a set of elements changes over time, effectively creating a visual chronology of the elements.

Learn More: [Insert Related Elements](#)

# Use Cases and Scenarios

Enterprise Architect has a full and rigorous implementation of Use Cases and Scenarios including being able to create Use Case diagrams that include Scenarios. The Use Cases and Scenarios can be included on a number of other diagrams to show how business processes are automated or which component realizes the Use Case. There is also an innovative editor where the detailed steps of Use Cases and Scenarios can be created and these can be generated to documentation thus completely dispensing with the need to create Word Processor files to document Use Cases. The steps can also be automatically generated to a number of diagram types including Activity and Sequence diagrams. The generated diagrams can be synchronized with the steps if they change and the diagrams redrawn.





## Scenario Builder

The **Scenario Builder** is a productive and unique tool and editor that allows the analyst to work with the text of Use Cases and Scenarios directly inside the model. Many analysts will be familiar with creating large Word Processor documents describing the details of Use Cases. With the Scenario Builder the descriptions and steps of Scenarios can be entered directly into the repository and linked to other elements. Alternate and Exception paths can be defined,

including branch and re-entry points. Diagrams representing the steps in a scenario can be generated and automatically synchronized. Constraints, including Preconditions and Post-conditions, can be defined and Test Cases automatically generated. Elements that have a relationship to the selected Use Case are listed in a convenient Context Reference list.

Learn More: [Scenario Builder](#)

## Use Case Diagram

The Use Case diagram is a simple and powerful way of describing the goals of the users of a system (or entity). It describes who wants to achieve a particular goal, but not how the goal will be delivered. Use Case diagrams can be simple or structured, and relationships such as Include, Extend and Generalization can be added to refine the model. A system (or entity) boundary can be added with a descriptive name showing clearly that Use Cases reside inside the system and Actors outside.

Learn More: [Use Case Diagram](#)

## Relationship Matrix

The **Relationship Matrix** can be used to define and display the relationships that exist between Use Cases and other

model elements including up-process and down-process elements. The up-process elements could include Business Processes, Stakeholder, Functional or Non-Functional Requirements, Business Use Cases and more. The down-process elements could include Components, Building Blocks, Test Cases and Experience models and more.

Learn More: [Relationship Matrix](#)

## Traceability Window

The Traceability Window automatically displays the relationships that exist between Use Cases and other model elements including up-process and down-process elements. The traceability tree view can be conveniently expanded to see deeper relationships and elements displayed in the window can be located in all diagrams in which they appear.

Learn More: [Traceability Window](#)

## Specification Manager

The **Specification Manager** can be used as an alternative tool for working with Actors and Use Cases. The names, descriptions and properties of the Actors, Use Cases and Scenarios can be conveniently created, viewed and edited in a list resembling a Word Processor or Spreadsheet presentation.

Learn More: [Specification Manager](#)

## Documentation

Even though the **Scenario Builder** provides a productive way of defining Use Cases and Scenarios, there will be situations where formal documentation is required describing the Use Case. The **Document Generator** can be used to create high quality corporate documentation using a built-in Use Case Report template, or an analyst can define their own custom template. An HTML report can also be created, allowing click-through and drill-down.

Learn More: [Documentation](#)

# Views and Viewpoints

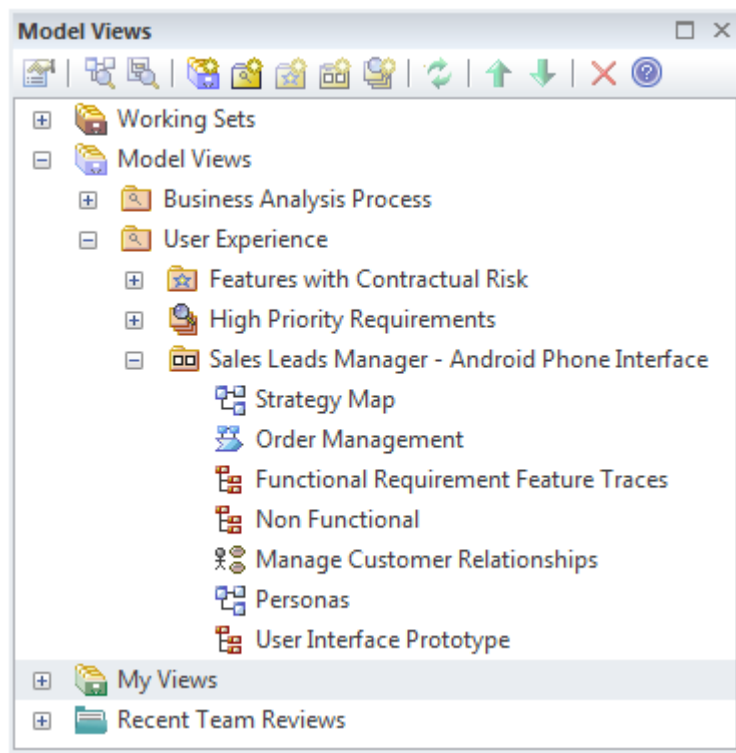
Views and Viewpoints are closely related to the notion of communicating with stakeholders, who commonly have different needs when it comes to understanding the architectures and the way they describe the things of interest to them.

There is theoretically an infinite number of viewpoints and views but in practice it turns out that many groups of stakeholders will benefit from a common set of views and viewpoints. It should be remembered that many stakeholders perform a number of roles and so an individual's needs might be satisfied by a number of different views from different viewpoints. The view is what can be seen from a viewpoint, but it can have different representations depending upon what the stakeholders want to see, including level of detail, fidelity, filtering, stylizations and more. So a Chief Information Officer (CIO) and a Development Manager might want to view the applications that realize a given business capability in the current state architecture, but the CIO will want a simple list or diagram. The Development Manager, on the other hand, might want a detailed diagram of the same applications, showing interfaces and payload information. A number of methods and languages promote the idea of a Viewpoint Library consisting of a series of useful and commonly used viewpoints.

Enterprise Architect has a wide range of tools to assist with

the creation and management of viewpoints, views and representations. There are several tools that can be used to create different views of the elements in the repository; these include the **Working Sets** and the **Model Views**. Working sets allow a collection of diagrams, Matrixes, Team Libraries and other items to be saved and reopened as a set, which is useful when working with different groups of stakeholders. Model Views can be used to create views of elements grouped together irrespective of their location in the **Browser window**. There are several tools that can hide or obscure parts of a diagram to make it more appealing to a particular audience. The appearance of diagrams can be altered by changing the appearance of elements, including using an image, and Diagram Filters can obscure or hide elements from view. A powerful Documentation engine can create high quality publications directly from the model.

This section lists the main tools available in Enterprise Architect that can be used to provide Views and Viewpoints. There is also a wide range of additional tools that a modeler might find suitable when applying the technique in a particular context. The complete list of tools is available from the topics [Meet the Enterprise Architecture Tools](#) and [Additional Enterprise Architecture Tools](#).



## Relationship Matrix

The **Relationship Matrix** can be used to present the relationship between two sets of elements in a matrix style of view. The matrix provides a compelling view of the relationships between the elements allowing stakeholders, analysts and architects to quickly locate missing or overlapping relationships. The matrix can be augmented with overlays that can be used to present specific values in the intersecting cells to indicate the nature of the relationship.

Learn More: [Relationship Matrix](#)

## Specification Manager

The **Specification Manager** is a powerful tool for creating a view of elements contained in a Package or element hierarchy in the repository. It is particularly useful for presenting lists or catalogs of parts of the architecture to non-technical stakeholders. It has the appearance of a simple document editor or spreadsheet and there is a range of options to change the presentation to make it more appealing for different audiences.

Learn More: [Specification Manager](#)

## Working Sets

Working sets are a useful facility for working with views and viewpoints, as they allow a series of views to be collected as a set, given a name, saved and reopened at a later time. They are useful for working with stakeholders who might be interested in a particular set of views. The set can include diagrams, Matrix Profiles, Team Libraries and more, and any number of working sets can be created.

Learn More: [Working Sets](#)

## Model Views

**Model Views** is a facility that allows a modeler to create lists of elements in alternative views to the way they are



organized in the **Browser window**. It is a useful facility for creating views of elements and diagrams that can be saved as a set, viewed as a slide show or dynamically created using a search. Any number of Favorite folders can be created and grouped into one or more higher level folders. This provides a mechanism for manually defining a view by simply dragging elements or diagrams from the Browser window into the Favorites folder, and ordering them using the up arrow key and down arrow key in the toolbar. Alternatively, elements can be listed using a built-in or user defined search to select elements from anywhere in the repository based on properties such as Complexity, Version and Phase, or user defined properties defined in Tagged Values. You can also set up a Slide Show folder where any number of diagrams can be added to a slide show. This is a useful tool for presenting prototypes, particularly when there are a number of diagrams that describe the user interface in different states, such as how the interface responds to a button click. Views can be saved as slide shows that can be configured to run automatically with a specified time interval, or slide progression can be controlled manually.

Learn More: [Model Views](#)

## Roadmap Diagrams

The Roadmap is an overlay that can be applied to any diagram and can be conveniently enabled or disabled. The overlay presents a timeline that is used to indicate the

starting time, **duration** and finishing time of important phases in the lifetime of the elements on the diagram. It is particularly useful for describing transitions between a baseline and a target or transition architecture. Any diagram object can appear on a Roadmap diagram, including Business Capabilities and Capability Increments, Applications and Services, and Physical or Virtual servers described in a Technology Architecture. The appearance and properties of the timeline can be altered to set the Units, Tick Spacing (Year, Months, Days), Start and Finish time, Colors, Fonts and more. The phases on the elements can be configured in a Diagram Legend, allowing names and colors to be defined.

Learn More: [Roadmap Diagram](#)

## Documentation

Enterprise Architect has a powerful and flexible documentation generator that can produce Docx and RTF Word Processor files, PDF and HTML reports. Any number of views of the repository content can be created by generating documentation and the powerful **Virtual Documents** facility allows content to be cherry picked from anywhere in the repository and grouped into sections presented with common template. The formatting in element or diagram notes can also be carried through to the documentation. A sophisticated template facility exists that provides a range of built-in templates and also allows the

user to create their own templates, defining styles, cover pages, tables of contents, images and a wide range of other formatting options.

Learn More: [Documentation](#)

## Element Appearance

Many of the stakeholders will be managers or non technical staff who will benefit from viewing diagrams with graphics. Images from the standard Image Library could be used or an organization is free to create their own graphics to represent concepts in the domain of the architecture or that have particular meaning to individuals or groups of stakeholders.

See also: [Element Appearance](#)

## Visual Filters

Visual Filters can be used to obscure or hide elements on the diagram based on user-defined criteria. This is particularly useful as the same diagram can be presented in different views and used to communicate ideas to a wide range of stakeholders. Any number of filters can be created and used to focus attention on a particular part of the diagram, for example to show only applications or capabilities that are being implemented in a particular phase of the architecture or that have a specified status or both. The filters can be

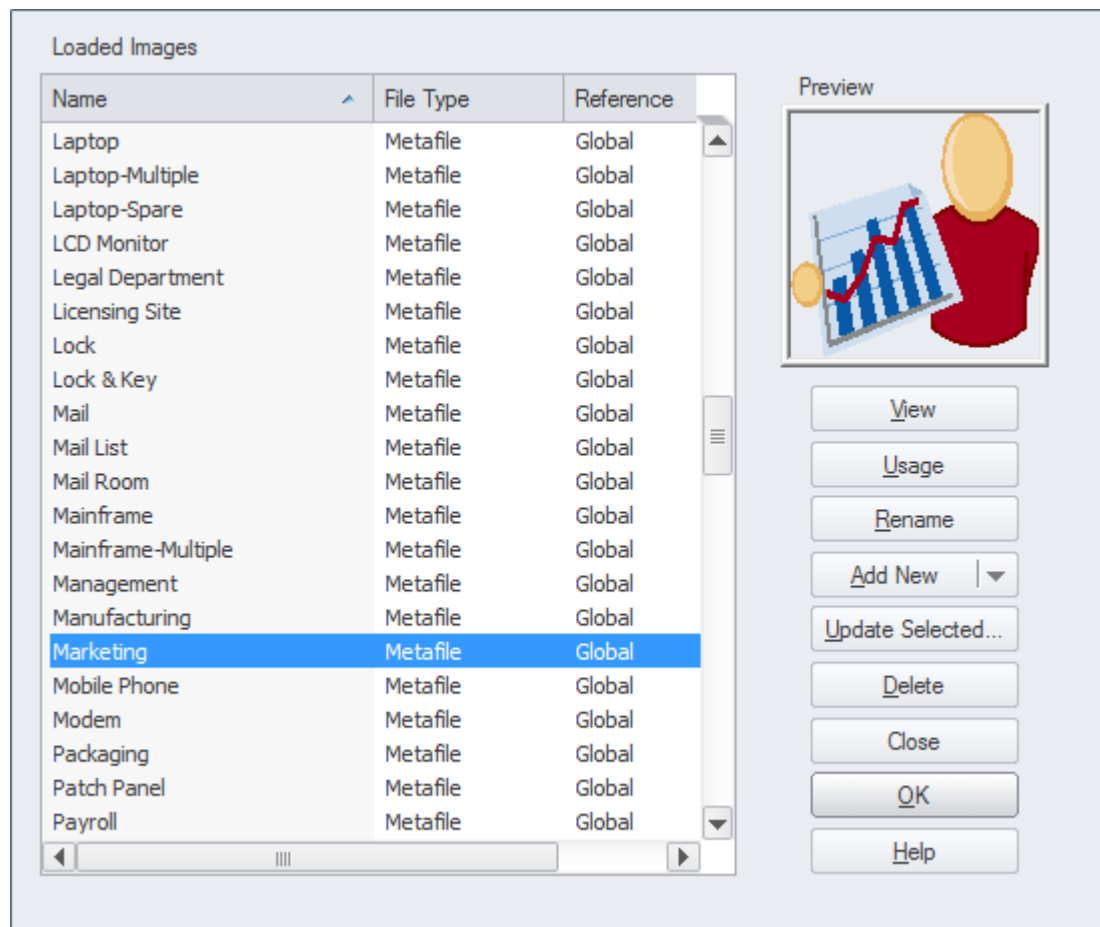
saved and reapplied to the same or any other diagram in the repository.

Learn More: [Visual Filters](#)

## Image Manager

The Image Library is a global store of images in the repository that can be applied to any number of diagram elements. The images can be in a variety of formats and when used to change the appearance of an element they create a compelling diagram that is often more appealing to managers or non-technical audiences. Where possible it is recommended to store a vector based image so that it can be effectively scaled in a diagram. The images can also be used when defining the metafile for a stereotype.

Learn More: [Image Manager](#)



# Workshops

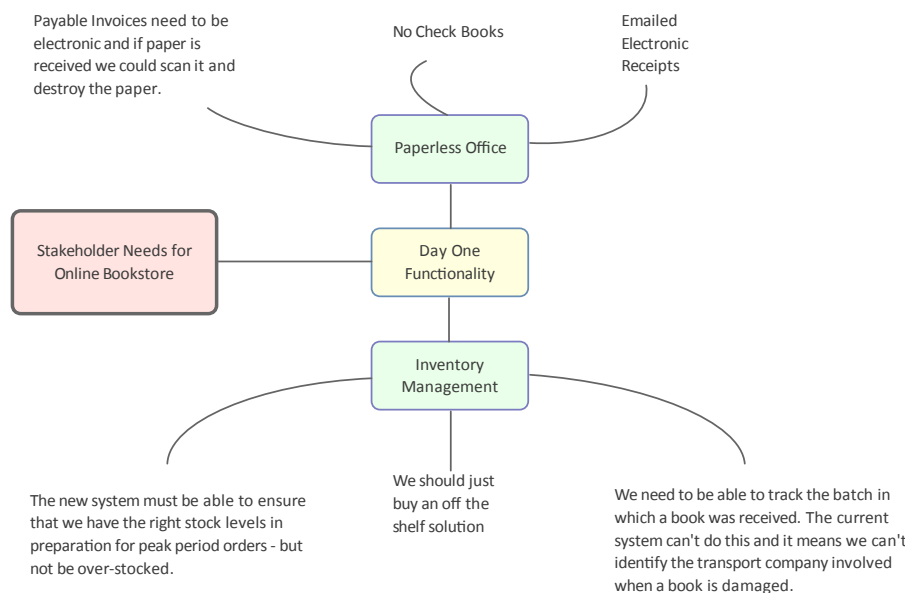
Enterprise Architect has a range of tools that can assist with planning, scheduling and conducting workshops. Depending how formal a workshop is, a plan might be required and an Agenda and Minutes might also be necessary. This can be achieved using a Document Artifact. A **Calendar** could be used to record the date, time and location of the meeting. **Model Mail** could be used to invite people to the workshop and to let them know when the minutes have been completed. Links to important information in the model can also be added to a mail message. During the workshop, Mind Maps could be drawn and **Discussions** posted, new concepts or Glossary terms recorded, and Slide Shows presented.

## Elicitation Workshops - Mind Mapping

This diagram shows the flexibility of Mind Mapping as a technique for recording needs elicited from stakeholders. It allows the modeler to keep a record of the workshops right inside the model. Once the analysis is complete, stakeholder requirements can then be linked back to topics in this diagram.

To create a new Mind Mapping diagram, from the 'Design' ribbon, select the option: 'Diagram > Add > Mind Mapping > Mind Mapping Diagram'.

Make sure that the perspective is set to 'All Perspectives' or select 'Strategy > Mindmap' in the Perspective combo box and that the Mind Mapping technology is enabled in the MDG Technologies dialog.



## Mind Mapping Diagram

A Mind Mapping diagram can be used to record the stakeholder's statements during an elicitation workshop. The statements are not categorized but simply recorded, and later during the analysis phase of Requirements development they can be converted to the appropriate elements or retained and the Requirements can be traced back to the topics, effectively creating a record of how the Requirement was derived. This is a powerful technique that shields the

stakeholders from needing to know the modeling languages, and it allows them to concentrate on articulating their needs; it also frees the analyst up from concerns about which element to use to model the statements. The derivation of requirements from Mind Mapping topics is usually performed in the analysis phase of the Requirement's Development process.

Learn More: [Mind Mapping Diagram](#)

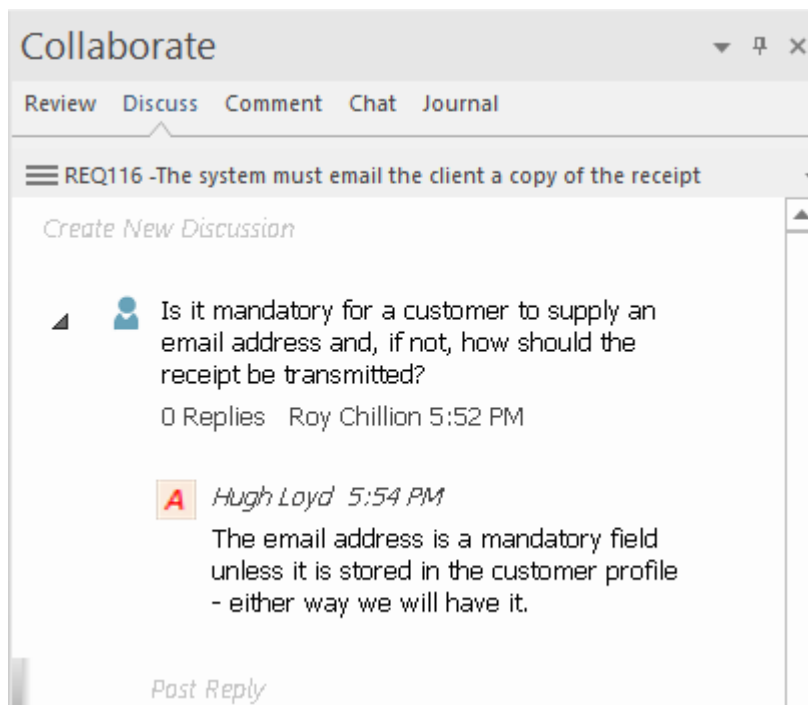
## Discussions

The **Collaborate window** is a convenient facility that allows commentary to be made on elements without contaminating the notes with discussions that ultimately don't contribute to the integrity of the model. Modelers often place notes on diagrams or write questions in the element Notes fields, and these are distracting and must be removed when formal documentation is generated from the model. The Collaborate window allows a modeler to initiate a discussion and others to reply. It is a perfect way to discuss Requirements.

A **Discussions** summary window conveniently displays the Discussions for all elements in the repository.

Learn More: [Discussions](#)





## Document Artifact

A Business Analyst will typically plan a workshop and define the approach, and if it is a structured interview a list of questions will be defined in advance of the workshop. A Document Artifact can be used as a place to store this information directly inside the model. A template could be specified for the document and the answers to structured or unstructured question recorded into the Document. Other model elements that exemplify some aspect of the workshop, such as topics of discussion, can be dragged into the document as links.

Learn More: [Document Artifact](#)

## Model Views

**Model Views** provide alternative views of the elements and diagrams stored in the **Browser window**. The Slide show is particularly useful for presenting a series of diagrams during a workshop and solves the problem of needing to remember the location of the diagrams in the Browser window. Any number of slide shows can be created and diagrams can be added regardless of their type or location in the Browser window. The progression from slide to slide can be automated or triggered manually.

Learn More: [Model Views](#)

## Concept Model

A Concept Model will act as a guiding model for discussions with many stakeholders; ideally a skeleton model should be created prior to the commencement of any workshop. The Concept Model should be kept simple; domain elements should be given a name and a description or a responsibility, and initially only important connections should be made between elements. As the workshop progresses new elements will be uncovered that can be added directly to the model, giving the stakeholders confidence that their needs and concerns are being

addressed and managed well. Enterprise Architect allows domain models to be created using the **UML Class** diagram.

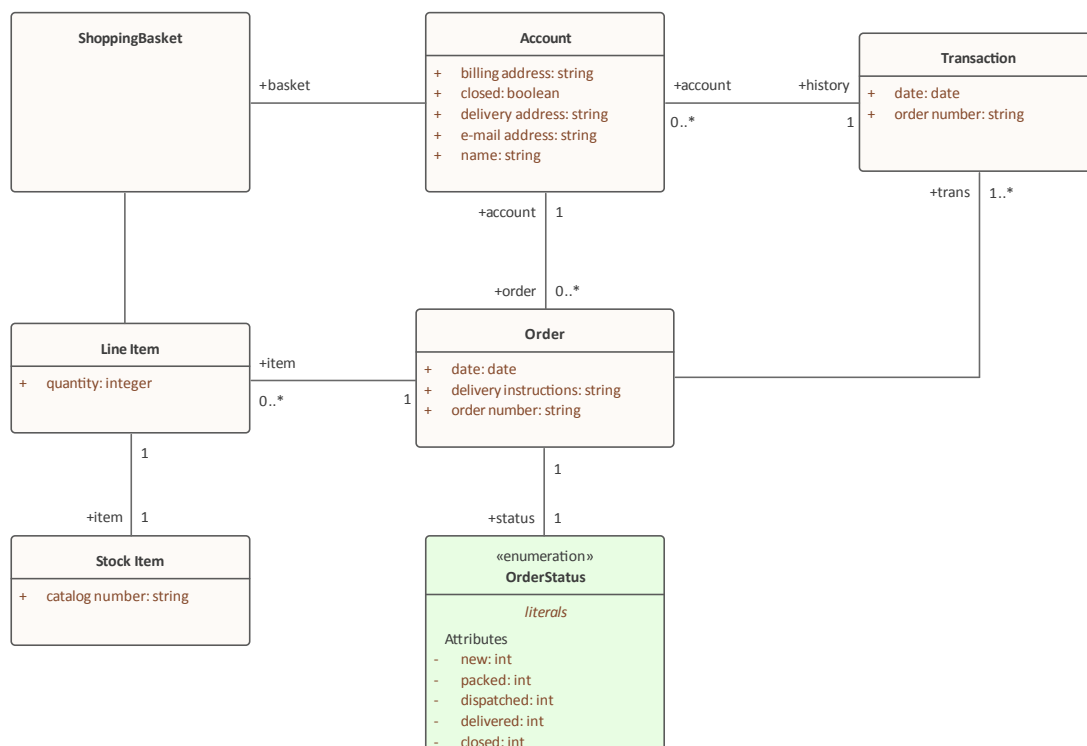
Learn More: [Class Diagram](#)

## Domain Model

The Domain Model is a powerful mechanism for recording and defining business terms that are identified during Requirements analysis. It provides a single definition of the terms and their relationships, that can be referenced from anywhere within the model.

One of the advantages of using a domain model is that the terms are modeled as Class elements, allowing them to be linked to other elements within the domain model itself or to elements in other parts of the model. They can be used on any number of diagrams, and they can be displayed as a list, using the Package List window.

It is also possible to create hyperlinks within the Notes text of Requirements elements, that link directly to relevant domain elements.



## Glossary

Prior to a workshop, an analyst can populate the **Project Glossary** with the existing terms and their meanings that

have been gleaned from reading project documentation such as a Business Case or Vision Document. During the workshops, as new terms are uncovered they can be added to the Glossary and their definitions can be discussed and entered or deferred until later in the analysis phase.

Learn More: [Glossary](#)

Glossary Item Details

Term:  Type:

Meaning:

**B** *I* U A |  $\frac{1}{2}$   $\frac{1}{3}$  |  $x^2$   $x_2$

The Stock Item defines the items (books) that are stocked in the warehouse for on-line purchase.

## Calendar

The **Calendar** provides a way to record and present important temporal information, such as events and meetings, in calendar format. Workshops can be entered into the Calendar and set up as recurring if needed. Event subtypes can be configured to add 'Workshop' as a Meeting Type.

Learn More: [Calendar](#)

## Model Mail

**Model Mail** can be used to notify and remind people of a

workshop and to inform them after the workshop about outcomes, decisions and required actions. Links to model elements, diagrams, Matrices, **Team Library** and a variety of other items can be added.

Learn More: [Model Mail](#)

# Additional Enterprise Architecture Tools

In addition to the tools listed in the *Meet the Architecture Tools* topic there is a wide range of other tools that can be used when working with Enterprise Architectures. These include tools for managing the model itself, such as **Security**, **Model Mail** and **Model Views**, and tools for working with elements, such as the **List View**, **Package Browser**, **Model Search** facility and more.

# Auto-Names and Counters

## Getting to know Auto Names and Counters

### Introducing Auto Names and Counters

To aid, regulate and enforce a naming standard, Enterprise Architect includes some capabilities to configure the default names assigned to new elements of a specific type. This is a useful feature when dealing with complex and large sets of requirements, but is also relevant when dealing with smaller data sets.

**Auto Names and Counters** can be used to assign a sequential number to any element type including Requirements. It includes a prefix definition, a counter and a suffix definition allowing numbers such as: 'REQ007 - Manage Inventory' to be created.

The screenshot shows a configuration dialog box for 'Auto Names and Counters'. At the top, a 'Type' dropdown menu is set to 'Requirement'. Below this, there are two sections: 'Name' and 'Alias'. Each section contains three input fields: 'Prefix', 'Counter', and 'Suffix', followed by an 'Apply on creation' checkbox. In the 'Name' section, the 'Prefix' field contains 'REQ', the 'Counter' field contains '001', and the 'Suffix' field contains '-'. The 'Apply on creation' checkbox is checked. In the 'Alias' section, all three fields are empty, and the 'Apply on creation' checkbox is unchecked. On the right side of the dialog, there are three buttons: 'Save', 'Close', and 'Help'.

**Where to find Auto Names and Counters**

Ribbon: Configure > Reference Data > Settings > **Auto Names and Counters**

**Usage of Auto Names and Counters**

Analysts and others can use the sequential number for communicating unambiguously about the requirements without having to use the often long requirement name. Select the 'Apply on Creation' option to start using the auto numbering feature; this can also be used to temporarily suspend auto naming, for example if other types of requirements are being entered that don't need to have sequential numbers assigned.

**Options for Auto Names and Counters**

There are options to define the prefix, counter and suffix for a requirement.

The screenshot shows a configuration window for 'Auto Names and Counters'. At the top, there is a 'Type' dropdown menu currently set to 'Requirement'. Below this, the window is divided into two main sections: 'Name' and 'Alias'. Each section contains three input fields: 'Prefix', 'Counter', and 'Suffix', and a checkbox labeled 'Apply on creation'. In the 'Name' section, the 'Prefix' field contains 'REQ', the 'Counter' field contains '0001', and the 'Suffix' field contains '-'. The 'Apply on creation' checkbox is checked. In the 'Alias' section, all three input fields are empty, and the 'Apply on creation' checkbox is unchecked.

**Learn more about Auto Names and**

[Apply Auto Naming to Existing Elements](#)



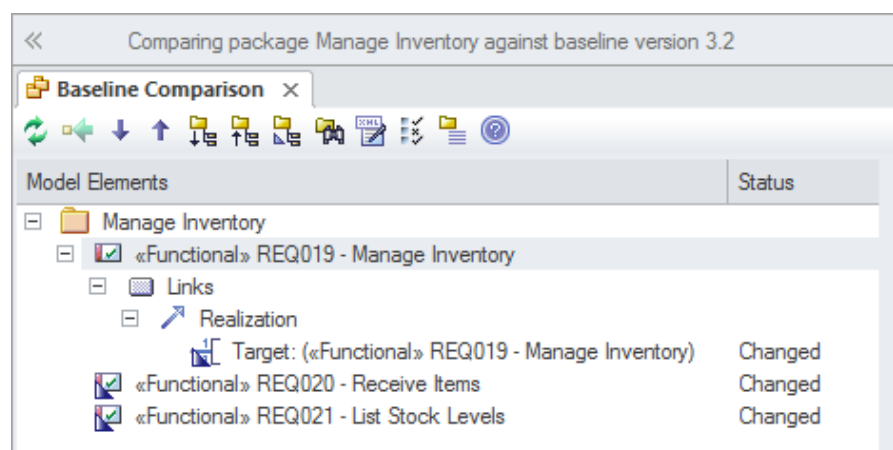
# Counters

# Baseline Tool

## Getting to Know the Baseline Tool

### Introducing the Baseline Tool

The Baseline Tool can capture a snapshot of the Requirements at a point in time and then, at a later time, the repository can be compared to this (or another baseline) for the purpose of determining what has changed. Any number of baselines can be created and labeled, and there is a baseline comparison tool that displays the differences between the baseline and the model and allows the modeler to revert a change in the model to a baseline at a granular level.



**Where to find the Baseline Tool** Ribbon: Design > Model > Manage > Manage **Baselines**  
Keyboard: **Ctrl+Alt+B**

**Usage of the Baseline Tool** **Baselines** are also useful when a formal requirements process is being followed or the Requirements form part of a contract, as the baseline can keep a snapshot of the requirements at important milestones such as contract signing or requirement phase sign off. This is also applicable to iterative and incremental processes such as Agile methods, as the requirements can be baselined before or even after a Sprint. When Requirements are still volatile and the Requirements' owners are still formulating their needs, a baseline can be created to take a snapshot at important points in the analysis phase, such as after an elicitation workshop.

**Options for the Baseline Tool** There are several options that can be applied to configure the way the Baseline Compare tool presents information; these are available from the **Options button** on the **Baselines** window.

☐ Always Expand to Differences

Show Elements that are:

- ☒ Changed
- ☒ In Baseline Only
- ☒ In Model Only
- ☐ Unchanged Items

Suppress these Changes:

- ☐ Suppress Diagrams
- ☒ Suppress Date Modified
- ☒ Suppress Date Created
- ☐ Suppress Children of Missing Items
- ☒ Suppress Advanced Properties

Baseline Diagram Compare Options

- ☐ Always open first parent with a Baseline  
- When comparing from the Project Browser or a Diagram

**Learn more**  
**about the**  
**Baseline Tool**

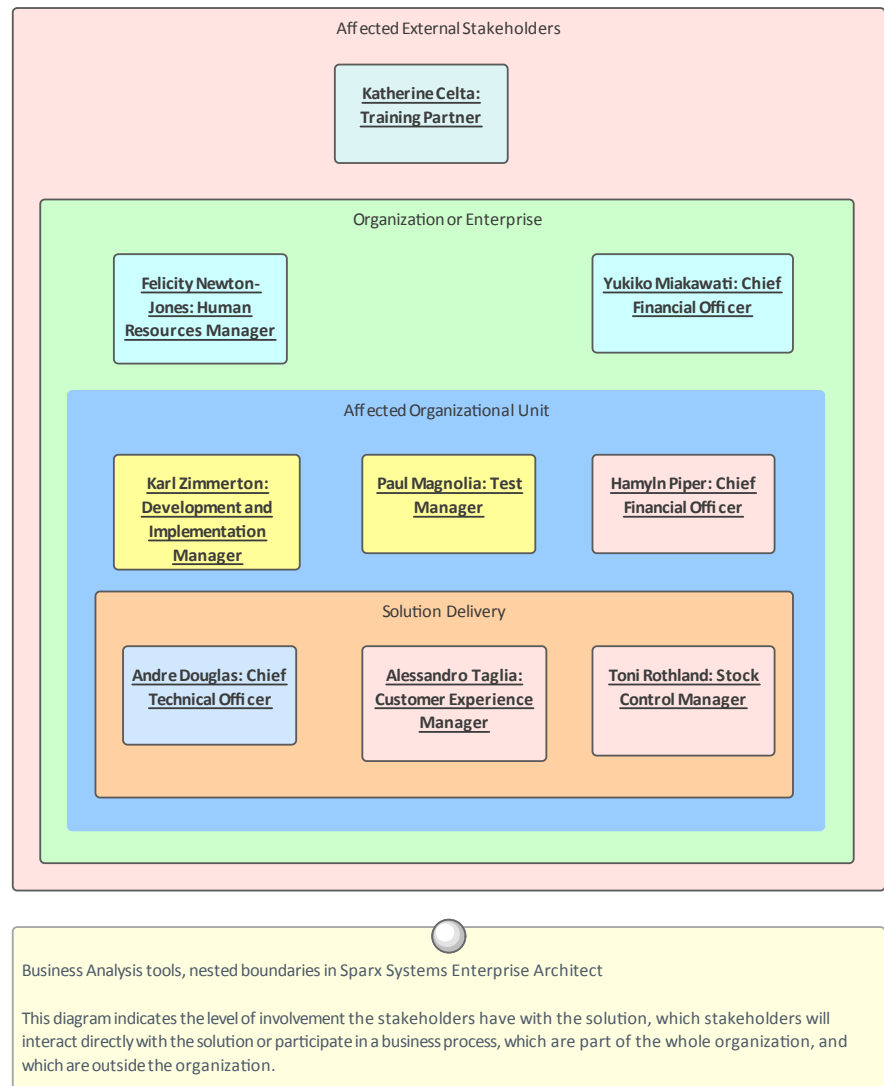
[Baselines](#)

# Boundary

## Getting to know the Boundary

**Introducing the Boundary** The Boundary element is derived from the Use Case system boundary but can be used extensively in other contexts as a way of describing a separation between a system or part of a system and its external environment. Any number of Boundaries can be added to a diagram and other elements such as Use Cases, Features, Requirements, Components and more can be placed inside the boundary. The properties of the Boundary can be altered to show a number of compartments organized into vertical and horizontal swimlanes. The border style and opacity of the Boundary can also be set.

## Stakeholder Onion Diagram



**Where to  
find the  
Boundary**

UML Elements Toolbar | Boundary  
Common Toolbox Page | Boundary

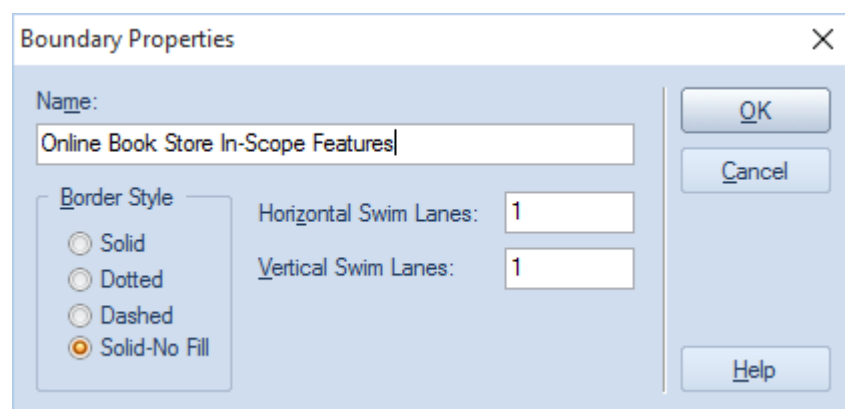
**Usage of the  
Boundary**

The Boundary element is particularly useful for defining what is inside a system (or part of a system) and what is outside. It can be used to show the Use Cases of a system or subsystem, the

in-scope Features or Requirements. Diagrams created for management and non-technical audiences will benefit from the use of Boundaries, which can be colored and nested to have visual appeal and business meaning. It is essentially a diagrammatic device and does not appear in the **Browser window**. If a formal and structural grouping of elements is needed, the modeler could consider the use of a Package.

### Options for the Boundary

The Boundary can be configured to have a number of vertical and horizontal swimlanes, which are useful for grouping elements into different sections of the boundary, creating a matrix effect. The Border Style can also be configured to allow different line styles that can then be colored using the standard element appearance settings. Also, choosing the 'Solid' options allows the Boundary's fill color to be set.



**Learn more  
about the  
Boundary** [Boundary](#)



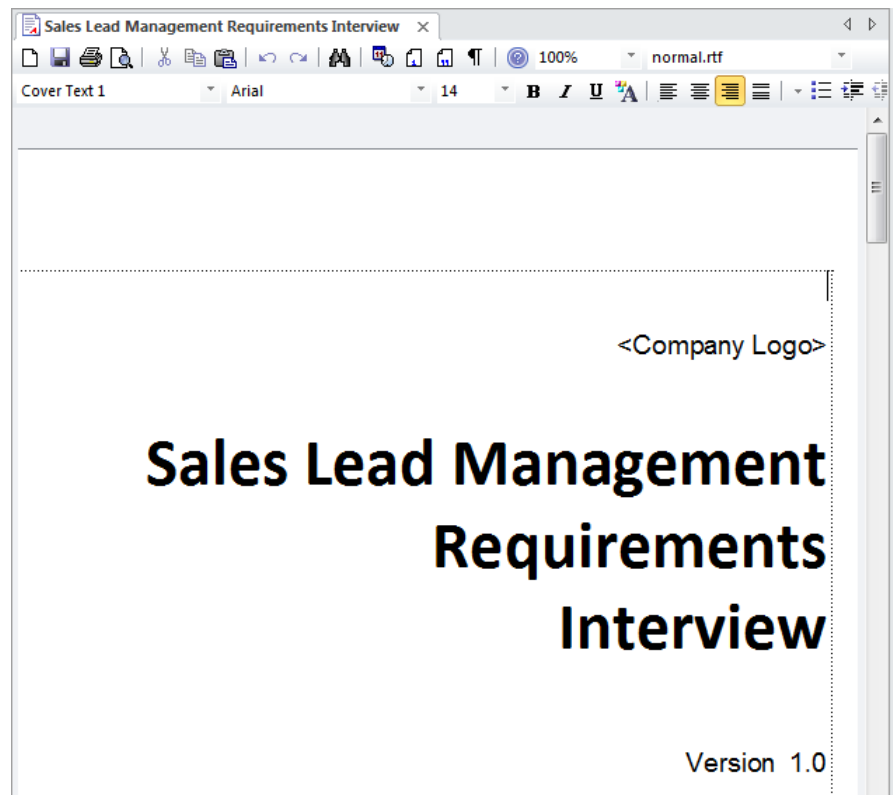
# Document Artifact

## Getting to know the Document Artifact

### **Introducing the Document Artifact**

A Document Artifact is similar to a word processor file that can be created and stored directly inside the model to record structured documentation. It is a light-weight and convenient alternative to creating a document in your favorite word processor. It has many of the features you would expect in a word processing tool, and allows you to create hyperlinks to content in the modeling Repository. An analyst will often be required to create structured documentation in addition to what might properly be placed in an element's notes or in a corporate document repository. This could include things such as interview plans, focus group agendas, surveys and more. There is rarely a requirement to store this type of documentation in a corporate document repository, and storing them inside the modeling repository has many benefits, including being able to include

hyperlinks to elements and diagrams contained in the **Browser window**.



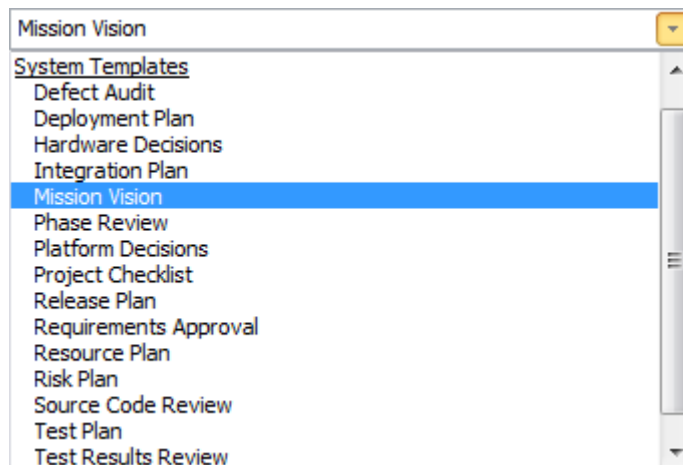
<b>Where to find the Document Artifact</b>	Toolbox: Common, Artifacts, Component, Documentation, Deployment Toolbox Page   Document Artifact
--	---

<b>Usage of the Document Artifact</b>	A Document Artifact can be used to add structured documentation in the form of a word processor (rtf) file. This can be a standalone element or it can be linked to another model element. In large organizations there is often a corporate repository for documents but in smaller organizations or when a document is less formal the document Artifact can be a
---------------------------------------	---

useful mechanism for storing information. For example it can be used to document or plan a stakeholder workshop or focus group.

### Options for the Document Artifact

When a Document Artifact is created there is an option to use a built-in or user defined template. Any number of templates can be created for documents such as focus groups, workshops surveys and plans.



Elements and diagrams can be dragged onto the document from the **Browser window**, which allows readers to click through from the location in the document to the diagram or element.

Many of the features found in Word Processors are available, such as inserting pictures and tables, creating Tables of Contents and headers and footers, tracking changes to the document and more.

**Learn more  
about the  
Document  
Artifact**

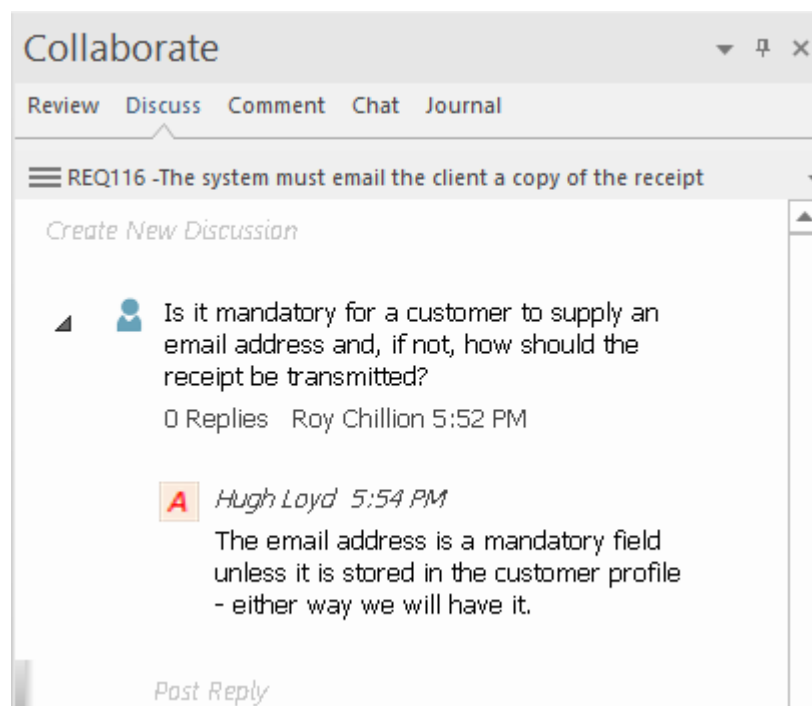
[Create Document Artifact](#)

# Discussions

## Getting to know Discussions

### Introducing Discussions

The **Discussions** facility allows modelers to have conversations about elements, posting discussions and replying to existing posts. The Discussions for all elements in the model are conveniently listed in the Discussions Review window, allowing a modeler to see all the elements with posts.



### Where to find

To post or view an element's discussion  
Ribbon: Start > Collaborate >

## **Discussions**   **Discussions** > Discuss

To view recently discussed elements

Ribbon: Start > Collaborate >

Discussions > Recently Discussed  
Elements

To view all discussions in the repository

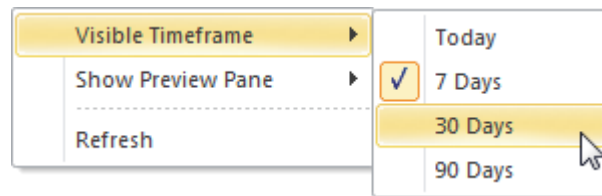
Ribbon: Start > Collaborate >

Discussions > Discussion History

**Usage of Discussions**   **Discussions** allow modelers to have conversations about elements without 'polluting' the element's notes with questions or modeling level comments such as 'Properties need to be added before the first release'. This feature brings the collaborative modeling platform alive, where modelers can add informal discussions about elements, emulating the discussions held in a physical workshop.

**Options for Discussions**   The **Discussion Review** window has a number of options to tailor the **Discussions** that are listed, including defining the timeframe to allow older and perhaps less relevant Discussions to

be hidden.



**Learn more  
about  
Discussions**

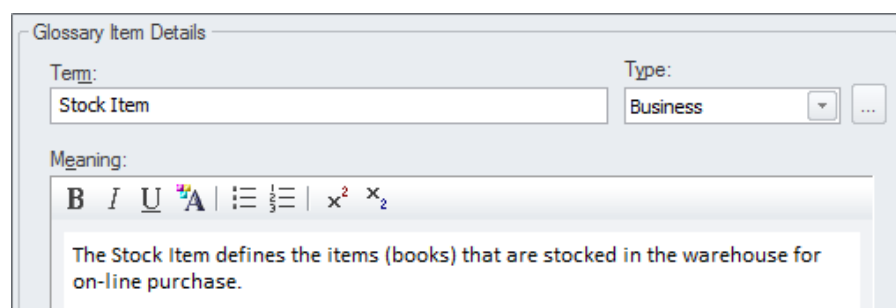
[Discussions](#)

# Glossary

## Getting to know the Glossary

### Introducing the Glossary

The Glossary is a project level lexicon of the important terms and their meanings categorized by type. Any number of terms, their types and meanings can be defined and these can be referenced from the notes of model elements. The terms can be included in documentation or generated as a stand-alone report. When working with domain specific requirement specifications, architectures and other models it is essential that new terms and over-ridden meanings for common words or phrases are kept in a suitable dictionary format to ensure proper understanding of documentation and specifications.

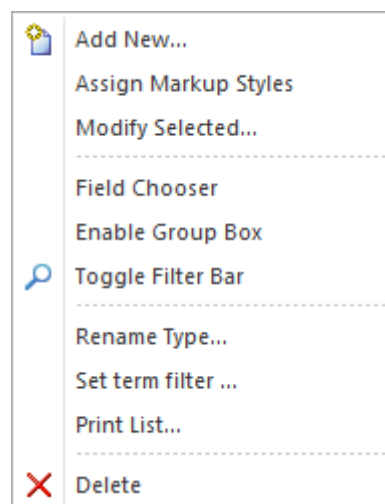




**Where to find the Glossary** Ribbon: Publish > Dictionary > Glossary

**Usage of the Glossary** The project Glossary can be used to record the important terms of a project or domain grouped by the type of term, allowing business, technical and domain specific types to be defined. A Glossary Report can be generated as a stand alone report or the glossary can be included as a section of another document.

**Options for the Glossary** The Glossary has a number of options to determine the terms that are displayed in the list and to define the style that is used for types of terms in documentation.



**Learn more about the Glossary** [Model Glossary](#)

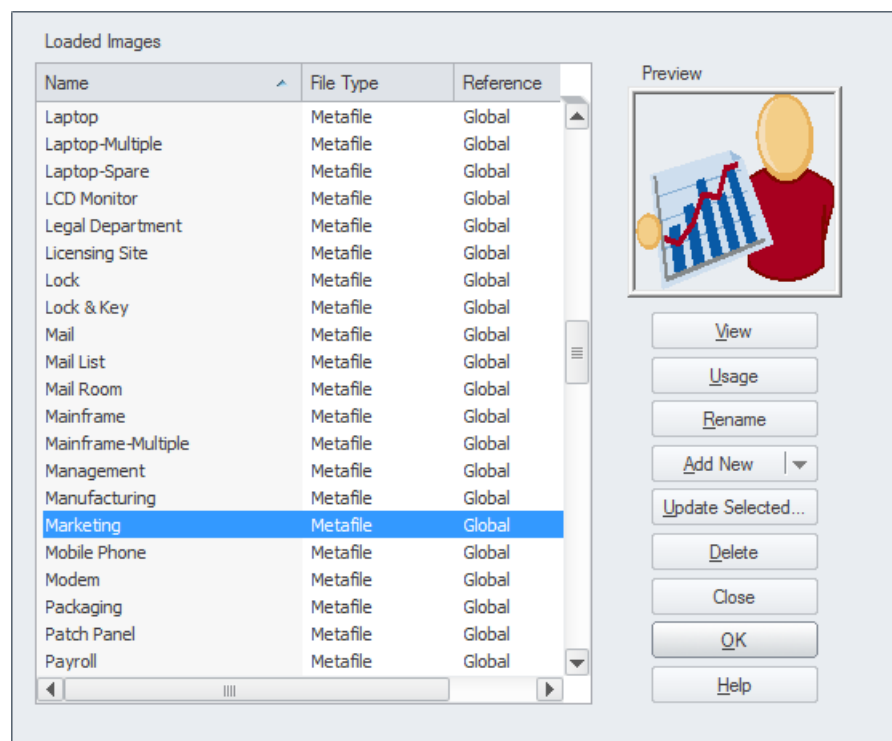


# Image Manager

## Getting to know the Image Manager

### Introducing the Image Manager

The **Image Manager** is used to manage pictures in the Repository, allowing you maintain and reuse them across diagrams. Images can be inserted in a variety of formats - including Bitmaps and Extended Windows Metafiles - and are then used to change the conventional appearance of elements to create more attractive and focused diagrams.



### Where to Defining Images:

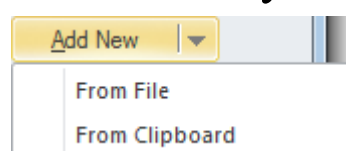
**find the Image Manager** Ribbon: Configure | Reference Data | Images

Using Images:

On diagram, right-click element | Appearance | Select Alternate Image

**Usage of the Image Manager** The **Image Manager**'s main use is to define pictures that can be used as alternative representations for elements in diagrams. For many business and networking diagrams the default appearance of an element such as a **UML Class** or Component can be replaced with an image from the Image Manager, making the diagram more relevant to its users. The Image Manager will also store images that are defined in included technologies.

**Options for the Image Manager** The **Image Manager** allows Images to be imported in a variety of formats including Bitmaps and Extended Windows Metafiles. Images can be imported from the file system or conveniently from the Clipboard.



There is also an option to display the diagrams where the image has been used.

**Learn more  
about the  
Image  
Manager** [Image Manager](#)

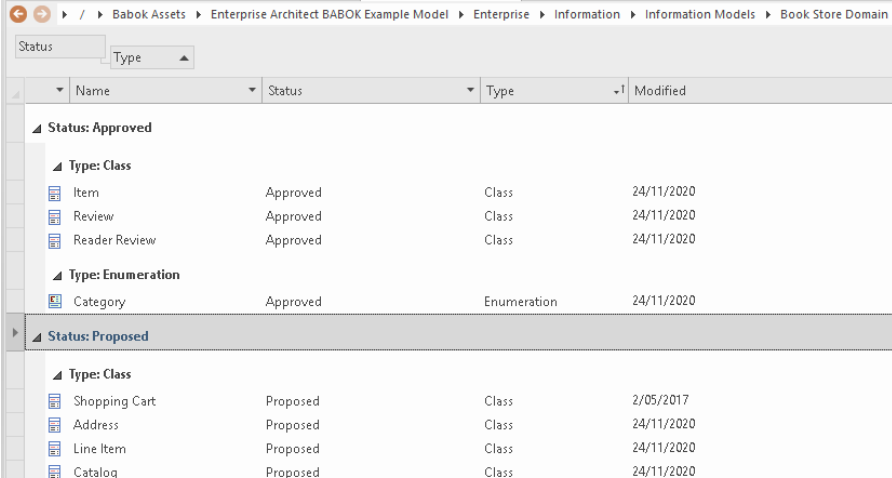
# List View

## Getting to know the List View

### Introducing the List View

The List View is a powerful tool for viewing the contents of a Package or a diagram as a list of elements, allowing the elements' properties to be viewed and edited in a convenient spreadsheet-like view. Many analysts will prefer to view elements in a list, and when details are changed the changes will be effective in every other view of the element, including diagrams and the **Browser window**.

There are a number of options for filtering, sorting and grouping elements in the list based on a wide range of properties and Tagged Values.



Name	Status	Type	Modified
<b>Status: Approved</b>			
<b>Type: Class</b>			
Item	Approved	Class	24/11/2020
Review	Approved	Class	24/11/2020
Reader Review	Approved	Class	24/11/2020
<b>Type: Enumeration</b>			
Category	Approved	Enumeration	24/11/2020
<b>Status: Proposed</b>			
<b>Type: Class</b>			
Shopping Cart	Proposed	Class	2/05/2017
Address	Proposed	Class	24/11/2020
Line Item	Proposed	Class	24/11/2020
Catalog	Proposed	Class	24/11/2020

**Where to find the List View**

Ribbon: Design > Model > List View  
Ribbon: Design > Diagram > View As > List View  
Diagram Context Menu: Switch View | Switch to List View  
**Browser window**, Package Context Menu: Open Package in | List View

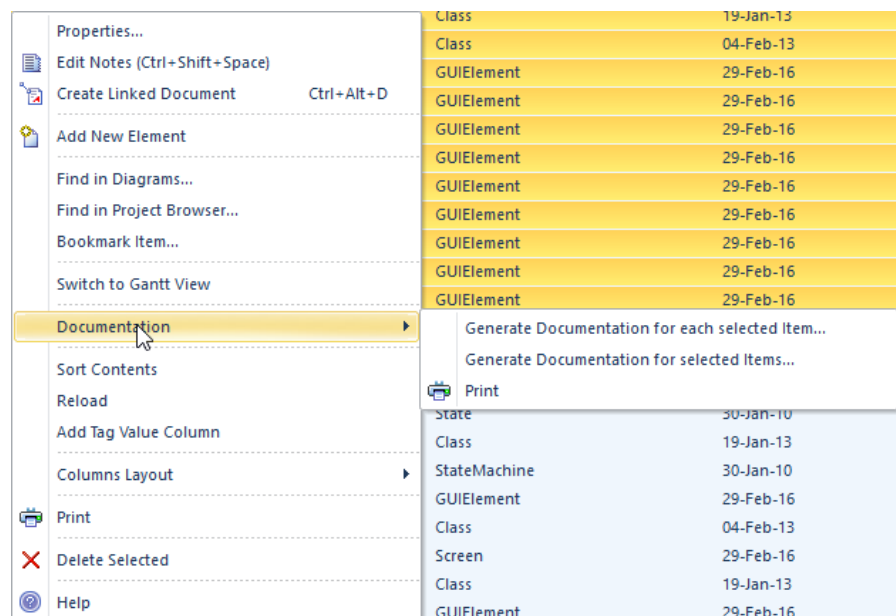
**Usage of the List View**

The List View can be used to get a different view of the elements contained in a Package or a diagram and to visualize their properties in a single spreadsheet-like view. It is particularly useful when dealing with non technical audiences or when the elements need to be sorted, filtered or grouped based on criteria such as Name, Status, Phase, Version and more. Project Managers and others working with sets of elements will be able to analyze in a given Package or diagram. There is also a related **Gantt View** where resource allocations, work completion and timings can be visualized.

**Options for the List View**

Additional properties can be added by using the 'Field Chooser' dialog and Tagged Value columns can also be

added, allowing a modeler to view the elements' native properties and Tagged Values side-by-side. The elements in the list can also be exported to a document report using any number of built-in or user defined templates and generating to a variety of formats including DOCX, PDF and RTF.



**Learn more  
about the  
List View**

[Package Browser \(List View format\)](#)

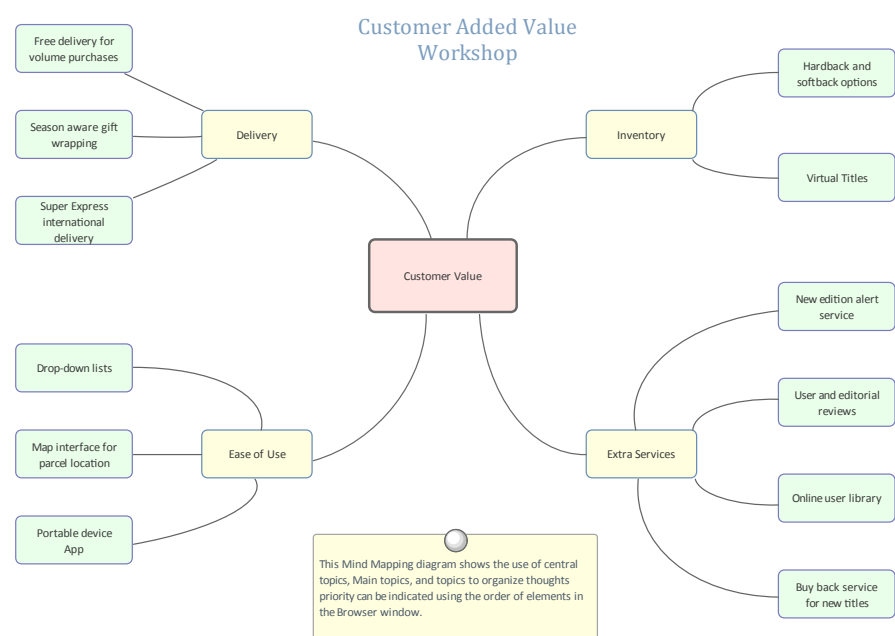


# Mind Mapping Diagram

## Getting to know the Mind Mapping Diagram

### Introducing the Mind Mapping Diagram

The Mind Mapping Diagram is a useful tool for recording ideas, thoughts and information in a simple and understandable format. The diagram is a type of spider (or radial) diagram where a Central Topic or idea is typically located at the centre of the diagram and Main Topic, Topic and Sub-Topics radiate out as the ideas are explored and recorded.



An analyst will often be required to take notes or record ideas (including their own) and information in the context of meetings, workshops and focus groups. The Mind Mapping diagram is a useful tool and its simple and appealing layout makes it accessible to everyone.

Regardless of what formal languages or frameworks are being used to articulate requirements and processes and other artifacts the Mind Mapping diagram requires little or no explanation.

**Where to find the Mind Mapping Diagram** Ribbon: Design > Diagram > Add > Mind Mapping > **MindMapping Diagram**

**Browser window** Toolbar : New Diagram icon > Mind Mapping > MindMapping Diagram

Browser window context menu | Add Diagram... > Mind Mapping > MindMapping Diagram

**Usage of the Mind Mapping Diagram** A **MindMapping Diagram** can be used to record ideas and information in the context of a workshop, focus group, meeting or even to get your own thoughts down. It becomes a compelling visual record of the communication and

discussions and often acts as a precursor to more formal techniques such as requirements articulation, Architecture modeling database design and more. Requirements and other elements such as Capabilities, Acceptance Criteria, design Components, User Stories and more can all be traced back to Topics and Sub-Topics in a MindMapping Diagram.

**Options for  
the Mind  
Mapping  
Diagram**

A MindMapping diagram can be made more expressive with the use of images and colors. The appearance of the Central Topic and Main topics could use an image that visually communicates the central idea. Colors can be applied to communicate concepts such as the importance or the owner of an idea or any other concept. Dependency relationships can be added to show how ideas relate to each other; for example, which stakeholders have specific interactions or which services depend on other services.

The **MindMapping Diagram** (like any diagram) can be viewed as an element list which makes working with the element's properties easier.

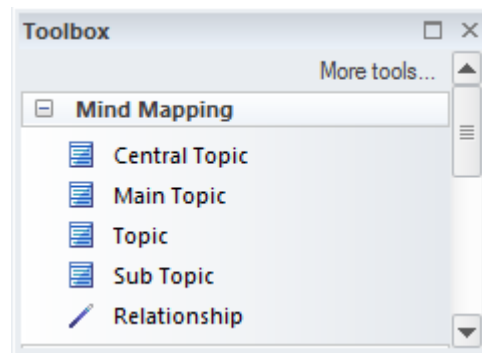


Diagram Filters can also be used when presenting the diagrams, to draw attention to parts of the diagrams. The diagrams can be presented as hand drawn or in a whiteboard style by changing the properties of the diagram.

**Learn more  
about the  
Mind  
Mapping  
Diagram**

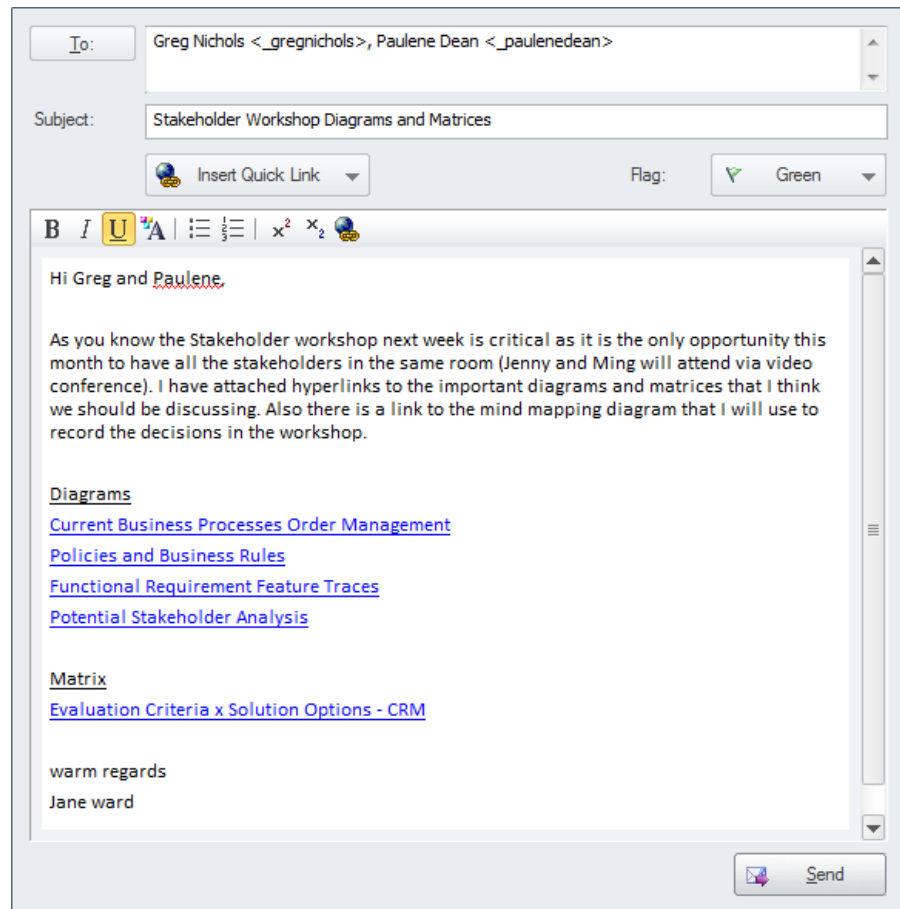
[Mind Mapping Diagram](#)

# Model Mail

## Getting to know the Model Mail

### **Introducing the Model Mail**

**Model Mail** is a mail system internal to the Enterprise Architect repository that allows users to send and receive mail messages. One of the important advantages that Model Mail has over other mail systems is that it is possible to embed hyperlinks to parts of the repository, allowing the recipient to click through to diagrams, matrices, elements, Packages, reviews and more.



Most initiatives are realized by a team of people, and critical to their success is good communication. With a fully featured tool such as Enterprise Architect, it is typical for team members to perform a lot of their work inside the tool; having mail internal to the repository provides a lot of advantages. The Model Mail facility allows team members and others using Enterprise Architect to send and receive mail messages including hyperlinks to repository content.

**Where to** Ribbon: Start > Collaborate > Mail

## **find the Model Mail**

### **Usage of the Model Mail**

**Model Mail** can be used to send and receive mail messages internal to the model, allowing team members and other stakeholders to communicate effectively about the model and its content. A typical scenario might be a team member sending a message with a link to a set of Requirements to a number of people, asking for their comments.

### **Options for the Model Mail**

**Model Mail** has the option to insert links to a variety of repository items, including diagrams, matrices, images, searches, Help topics, attributes, operations, **Team Library** and more.

### **Learn more about the Model Mail**

[Model Mail](#)

# Model Views

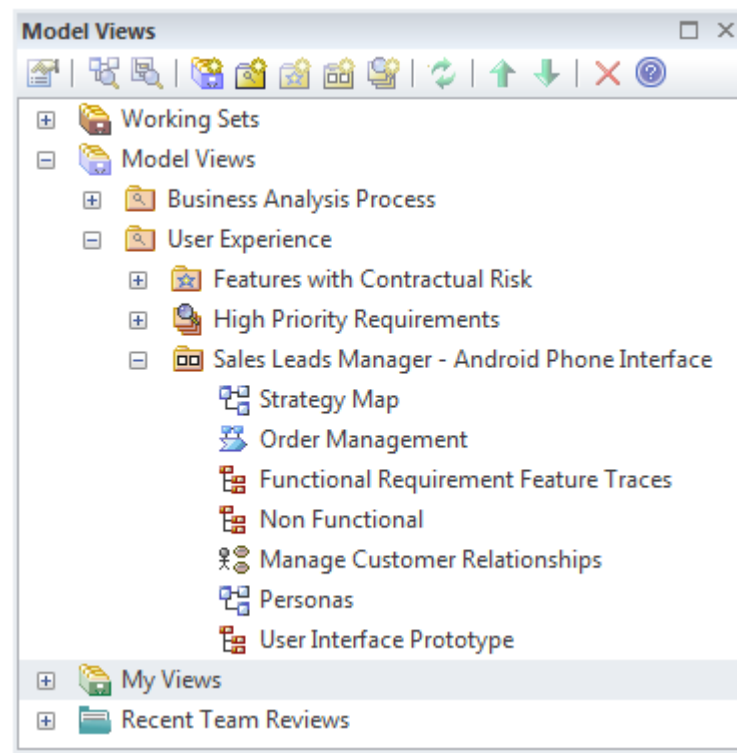
## Getting to know Model Views

### **Introducing Model Views**

**Model Views** provide an alternative view of the elements in the Repository. Whilst the **Browser window** is designed to organize the Packages and elements structurally, the Model Views facility allows the modeler to create a number of views that can group elements and diagrams differently.

The Browser window has been designed to organize the elements structurally, grouping elements and diagrams together into Packages based on factors such as namespaces, type of element and the part of the process. Model Views allow a user to create a structure based on a wide range of criteria including Favorites folders and folders based on a search such as all elements I created last week that have a status of 'Proposed'.





**Where to find Model Views** Ribbon: Start > Desktop > Design > Focus > Views

**Usage of Model Views** Model views can be created for a variety of purposes and an experienced modeler will often use the **Model Views** facility as the access point to the Repository. The Favorites folder is useful to keep track of hand picked elements and diagrams of interest. The Slide show folder is useful to create a diagram slide show for the purposes of demonstrating diagrams to a group of people in a workshop or meeting. The Search folder is useful to keep track of elements that meet specified criteria and to be notified when

new elements are added to the Repository that meet the criteria.

## Options for Model Views

The **Model Views** root folder defines views that every user can see, whereas the My Views root folder is only visible to the current user. Both these root folders can contain any number of user-defined folders down to two levels. User-defined folders under the Model Views folder can contain three types of view: Favorites, Diagram Slide Shows and views based on a Search. User-defined folders under the My View folders can contain only views based on a Search.



The **Model Views window** provides a convenient toolbar for working with the folders and elements.

## Learn more about Model Views

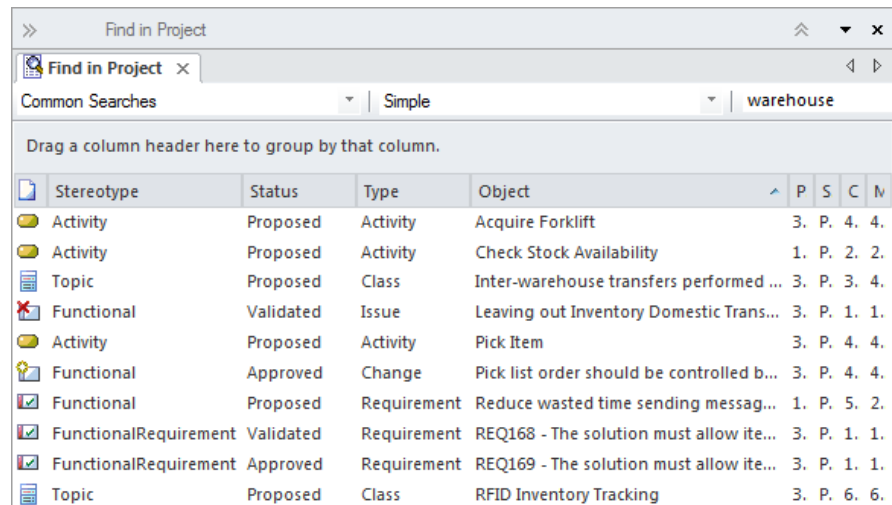
[Model Views](#)

# Model Search

## Getting to know Model Search

### **Introducing Model Search**

The **Model Search** facility is a powerful and flexible tool for finding anything in the modeling repository. An object can be located regardless of what a user is looking for, whether it is the name of an element, text in element notes, a Tagged Value or properties such as when an element was created, who created it, what its status is, and more. The facility includes a wide range of built-in searches that can be used or a user can define their own searches using the Query Builder, SQL Builder or by defining an **Add-In Search**.



With a number of analysts and others working on creating or importing content into the repository the number of elements will increase rapidly and before long even with a well organized model it will not be possible to remember where all the elements are located. The search facility can be used to locate information easily no matter where it is in the model.

### Where to find Model Search

Ribbon: Explore > Search > Model

Ribbon: Design > Element > Manage > Search Model

Keyboard: **Ctrl+Alt+A**

### Usage of Model Search

The **Model Search** facility can be used to find anything that exists in the repository. It can be used to return a set of elements that meet specified criteria for example in preparation for an

iteration to get the list of all Requirements that were created in the last two weeks that have a Status of Validated and that have a Difficulty value of Low and a Priority of High. The result set can be sorted, documentation can be generated and individual elements can be Book Marked, or located in the **Browser window** or in all the diagrams in which they appear.

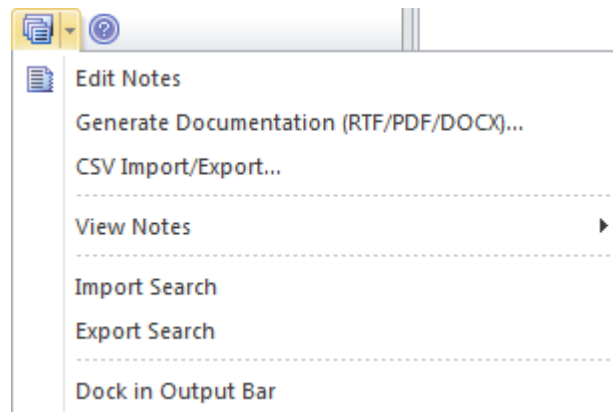
Searches can be used as the basis of a number of other tools such as **Model Views**, the Documentation Generator and more.

### **Options for Model Search**

A user can choose to use predefined searches or create their own user-defined searches. Searches can be defined using three different tools: the Query Builder, which is screen driven and easy for most non-technical analysts to understand; the SQL Builder, which uses SQL; or creating an **Add-In Search**, which requires some programming.

The Query Builder is the most intuitive of the tools and allows a user to build up their own query adding one or more filters to restrict the set of elements that will be returned.

The Search Facility can be configured to query a single Package (and its sub-Packages if required) or to search the entire Repository.



Elements returned in the search results list can also be exported to a word processor or Spreadsheet document.

**Learn more  
about Model  
Search**

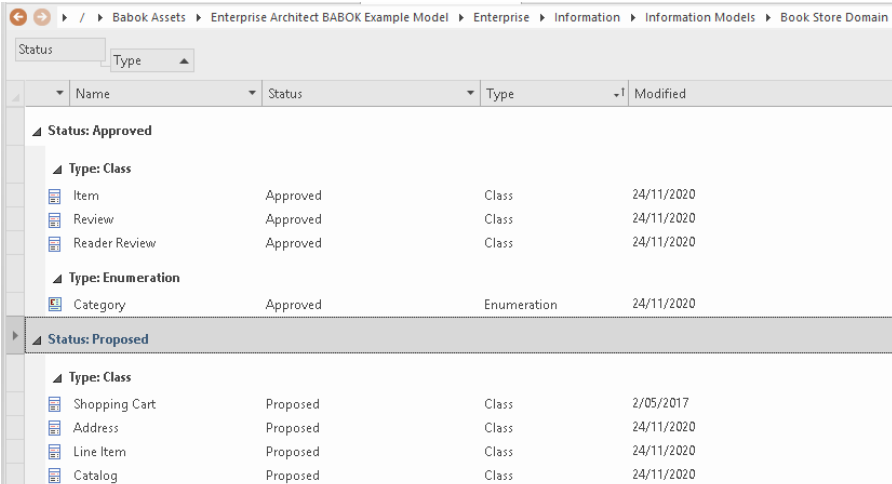
[Model Search](#)

# Package Browser

## Getting to know the Package Browser

### Introducing the Package Browser

The **Package Browser** is a powerful tool that provides a convenient list of the elements in a Package, displayed in a List or a Gantt view. The List view allows Modelers to work on the elements in the Package in a spreadsheet-like view, editing properties and notes inline and grouping and sorting the elements. The Gantt Chart view allows Analysts, Architects, Project Managers and others to view the resource allocation for an element in a convenient and familiar time-based view.



The screenshot shows the Package Browser interface with a breadcrumb trail: Babok Assets > Enterprise Architect BABOK Example Model > Enterprise > Information > Information Models > Book Store Domain. Below the breadcrumb, there are filters for Status and Type. The main table displays elements grouped by status (Approved and Proposed) and then by type (Class and Enumeration). Each element row includes a small icon, the element name, its status, type, and a modified date.

	Name	Status	Type	Modified
<b>Status: Approved</b>				
<b>Type: Class</b>				
	Item	Approved	Class	24/11/2020
	Review	Approved	Class	24/11/2020
	Reader Review	Approved	Class	24/11/2020
<b>Type: Enumeration</b>				
	Category	Approved	Enumeration	24/11/2020
<b>Status: Proposed</b>				
<b>Type: Class</b>				
	Shopping Cart	Proposed	Class	2/05/2017
	Address	Proposed	Class	24/11/2020
	Line Item	Proposed	Class	24/11/2020
	Catalog	Proposed	Class	24/11/2020

**Where to** Select the Package in a diagram or the

**find the  
Package  
Browser**

**Browser window**

Ribbon: Design > Model > List View  
Context Menu in diagram: Open Package  
in | List View  
Browser window Context Menu: Open  
Package in | List View

**Usage of the  
Package  
Browser**

The **Package Browser** provides a convenient and alternative view of the elements in a Package. The list format can be used as a workbench for viewing the elements in user defined order, changing the elements' properties and adding and deleting elements. The ability to display the elements grouped by various properties and Tagged Values allows analysis to be performed on groups of elements. The Gantt chart view provides a Project Management view of the elements, showing how resources are allocated to individual elements. Reports for all elements, or a selected group of elements, can be generated.

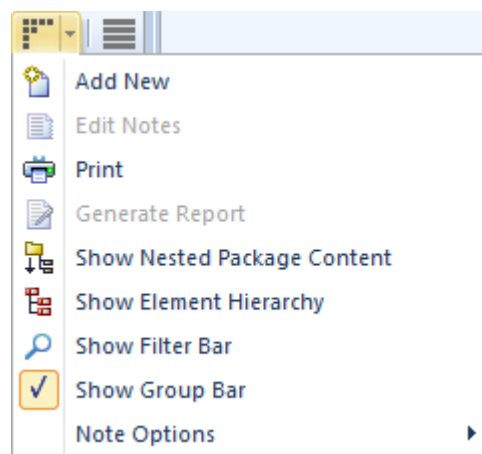
**Options for  
the Package  
Browser**

The **Package Browser** has two different formats or display options: a User Defined view and a Hierarchy view. The User Defined view displays all the



elements in the Package regardless of hierarchy. The Hierarchy view arranges the elements in accordance with the way they are grouped in the **Browser window**.

In the User Defined view, the items can be sorted in ascending or descending order by clicking in the column headers. The order of columns can be changed by dragging column headers left or right. A Modeler can drag column headers onto the View Header (above the column headers), which will group the items in the list by that property; column headers can be nested in a hierarchy to specify groups within groups.



**Learn more  
about the  
Package  
Browser**

[Package Browser](#)

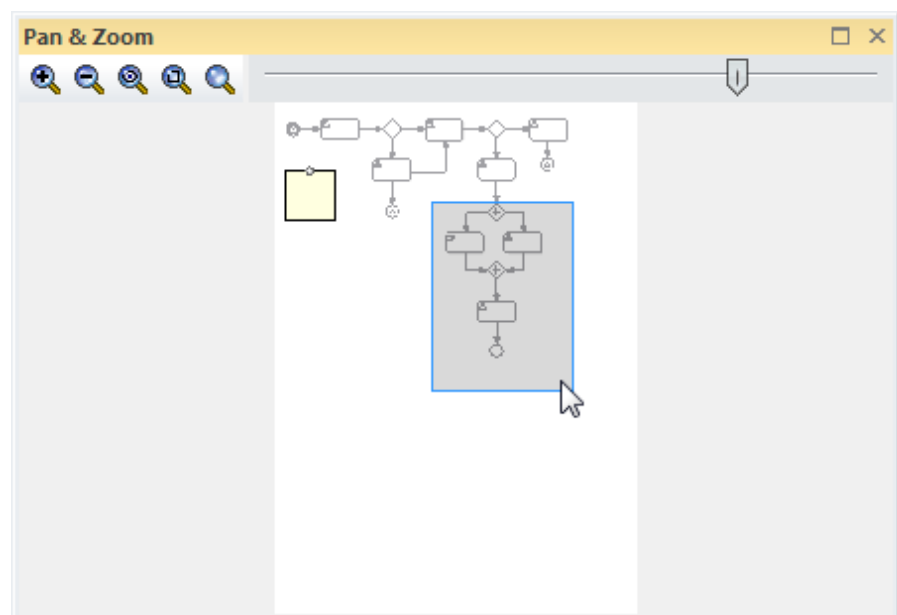


# Pan and Zoom

## Getting to know Pan and Zoom

### Introducing Pan and Zoom

The **Pan and Zoom window** is one of the tools that can be used to navigate around a large diagram. Often, the resolution of a diagram must be reduced to ensure it is wholly visible, but by using the Pan and Zoom window you can leave the diagram at a readable resolution and pan around to areas of interest, zooming in when necessary.



Many initiatives will be best described

by a number of large diagrams, ranging from A3 - A0. Even when you are fortunate enough to be using a large monitor, you will want to change the scale at which you are viewing the diagram and then pan around to find the section or element of interest in the diagram, and then zoom into that section. The Pan and Zoom window will allow you to do this for any size diagram, with tools for panning and zooming that are particularly useful during workshops or focus groups.

**Where to  
find Pan and  
Zoom**

Ribbon: Layout > Tools > **Pan and Zoom** (**Ctrl+Shift+N**)

**Usage of Pan  
and Zoom**

The **Pan and Zoom window** can be used for moving around and zooming into large diagrams. A typical scenario could be that an Analyst has created a diagram showing traceability from Solution Components back through detailed Requirements and User Stories and Processes up to the level of Business Goals. This diagram would typically be too large to view at the right resolution even on a large monitor. The Pan and Zoom window could be used to set the

diagram resolution to a scale that could be easily read and then to pan around the diagram to elements and areas of interest, zooming in and out as required.

### **Options for the Pan and Zoom**

The **Pan and Zoom window** has a number of options:

- You can pan across the diagram using your mouse pointer to drag the colored rectangle around to different sections of the diagram
- You can zoom into and out of the diagram either using the Slider control to change the scale of the diagram, or using the tool icons to:
  - Zoom In
  - Zoom Out
  - Zoom to fit diagram
  - Zoom to fit page
  - Zoom to 100%



### **Learn more about Pan and Zoom**

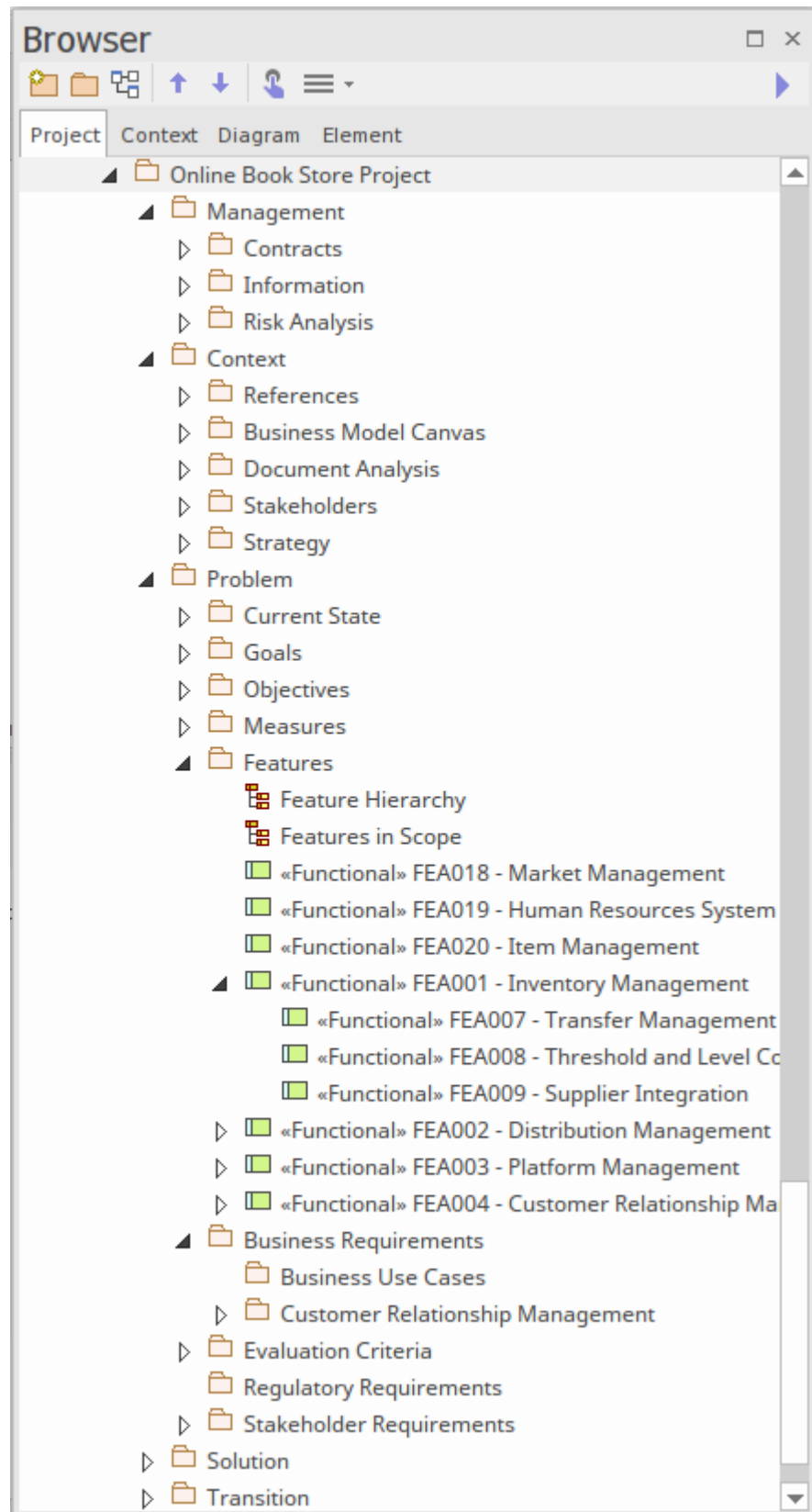
[Pan and Zoom](#)

# Browser Window

## Getting to know the Browser window

### Introducing the Browser window

The **Browser window** is the primary tool for structuring and navigating through the repository using expanding and collapsing tree nodes. The key structural element is the Package, which is a folder-like element that can contain other elements and diagrams, and other Packages. The elements in turn can contain other elements, features and diagrams. Root nodes are the highest nodes in the tree and these root Packages can contain views that in turn can contain any level of Packages and elements. Tree nodes including Packages, elements, Features and diagrams can be copied and pasted between locations or dragged and dropped to new locations. Many important tools, functions and windows are applied at the level of the Package, such as import or export of model content, documentation and **Package Control**, including **Baselines**.



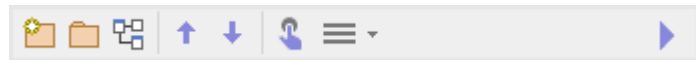
**Where to find the** Ribbon: Explore > Portals > Windows > Explore > Project Browser

**Browser window** Ribbon: Start > Desktop > Design > Browser

**Usage of the Browser window** The **Browser window** can be used to create and manage the structure of the repository by adding and deleting, moving and copying Packages, elements, Features and diagrams. It is often the primary tool for exploring and browsing through the elements in the repository and finding things of interest. A modeler working with an open diagram will often want to locate a diagram object in the Browser window as a way of finding out what Package it belongs to and what its peers are.

**Options for the Browser window** The **Browser window** has a context menu that contains the important functions that apply to each of the selected element types. Many of the functions are also available in the Browser header bar, including the ability to create models, Packages, diagrams, elements and documentation, and the ability to focus on specific Packages in the model by identifying them as 'Favorites'. The Browser window itself can be moved around the workspace as required.





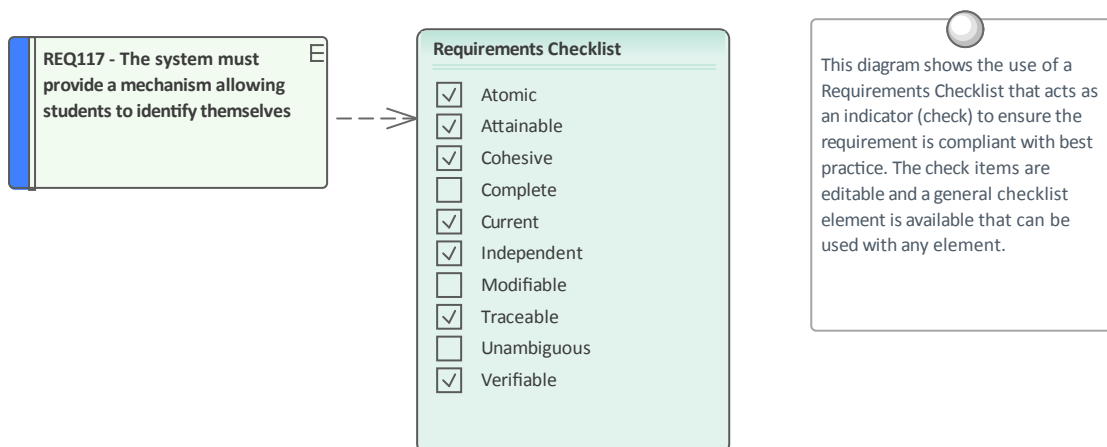
There are also several important ways the Browser window can be configured by using the 'Preferences' dialog. These include the ability to show or hide stereotypes in the name of a tree node, the ability to freely sort the tree nodes within a Package or element and whether to warn about deletions from the tree.

**Learn more  
about the  
Browser  
Window**


[Browser Window](#)

# Requirements Checklist

The Requirement Checklist is a convenient element that acts as a tally to indicate whether a Requirement complies with a set of predefined measures such as whether the Requirement is Atomic, Cohesive, Traceable and Verifiable. It can be assigned to any Requirement and the measures can be updated directly in the diagram. When working with Requirements it is sometimes very useful to refer to a common set of 'best practices' and qualities that help define the nature of a well formed specification. The Requirement Checklist element is designed to meet this need.



## Getting to know the Requirements Checklist

**Where to find the Requirements Checklist**      Toolbox :  to display the 'Find Toolbox Item' dialog and specify 'Requirements Checklist'

## Usage of the Requirements Checklist

Analysts and Requirements Managers can use the checklist to annotate whether one or more elements such as a Block or Activity or even a set of Requirements meet a set of predefined checks.

## Options for the Requirements Checklist

The list of measures is completely configurable and items can be added or removed from the list for each individual checklist by using the Checklist Tagged Value notes.

```
<Checklist>
  <Item Text="Atomic" Checked="True"/>
  <Item Text="Attainable" Checked="True"/>
  <Item Text="Cohesive" Checked="False"/>
  <Item Text="Complete" Checked="False"/>
  <Item Text="Current" Checked="True"/>
  <Item Text="Independent" Checked="False"/>
  <Item Text="Modifiable" Checked="True"/>
  <Item Text="Traceable" Checked="True"/>
  <Item Text="Unambiguous" Checked="True"/>
  <Item Text="Verifiable" Checked="True"/>
</Checklist>
```

## Learn more about the Requirements Checklist

[Using the Checklist and Audited Checklist Artifacts](#)

# Requirement Properties

## Getting to Know the Requirement Properties


### **Introducing Requirement Properties**

Requirement Properties define metadata about the Requirement that is useful for the management of Requirements for the purposes of prioritization and defining work Packages for the implementation teams. All Enterprise Architect elements have standard properties such as Status, Author and Phase, and the Requirement element has additional properties such as Difficulty and Priority. User-defined properties can also be created using Tagged Values.

**Properties** □ ×

Element Tags

---

Name REQ-022 

▾ **General**

Type	FunctionalRequirement
Stereotype	EAREQ::FunctionalRequirement
Alias	
Keywords	
Status	Proposed
Version	1.0

▾ **FunctionalRequirement ( from EAREQ )**

Priority	
dataDescription	<memo>
operationDescription	<memo>
workflowDescription	<memo>
reportDescription	<memo>

▾ **Requirement**

Abstract	<input type="checkbox"/>
Active	<input type="checkbox"/>
Difficulty	Medium
Final Specialization	<input type="checkbox"/>
Leaf	<input type="checkbox"/>
Priority	Medium
Visibility	Public

▾ **Project**

Author	hbritten
Package	
Phase	1.0
Complexity	Easy
Created	4/09/2019 4:31:58 PM
Modified	4/09/2019 4:31:58 PM
Language	<none>
Filename	
GUID	{B0170961-E1FE-4928-BDFE-8548E0ED6AD6}
WebEA	

**Where to** Ribbon: Design > Element > Editors >

**find** Properties  
**Requirement** Element Context Menu: Properties... |  
**Properties** Properties...

or

**Browser window** Context Menu:  
Properties | Properties...

**Usage of the** The Properties define the important meta  
**Requirement** information about a Requirement, for the  
**Properties** purposes of providing data to manage  
Requirements for prioritization,  
understanding which are the difficult  
Requirements, and managing the  
lifecycle by using Status to determine  
Requirements for implementation  
Packages.

**Options for** Enterprise Architect has a wide range of  
**Requirement** built-in properties for all elements, and a  
**Properties** number of additional Requirement  
Properties. If other properties are needed  
by a modeler or team, such as the  
volatility (stability) of a Requirement,  
these can be added using the  
general-purpose UML extension  
mechanism of Tagged Values.

REQ021 - List Stock Levels
<i>tags</i> Volatility = Medium
<i>notes</i> <i>A facility will exist to list current stock levels and to manually update stock quantities if physical checking reveals inconsistencies.</i>

**Learn more  
about  
Requirement  
Properties**

[Properties Dialog](#)

# Risk Taxonomy

## Getting to know Risk Taxonomy

**Introducing Risk Taxonomy** **Risk Taxonomy** is a facility to define a comprehensive, stable and reusable set of risk categories that can be applied universally across the system. This includes definitions of Threats, Loss type, Contact Frequency, Loss Magnitude, Risks and more. It is based on the Open Group Standard for Risk Taxonomy (OR-T) and provides Toolbox pages and diagrams for defining the Taxonomy.

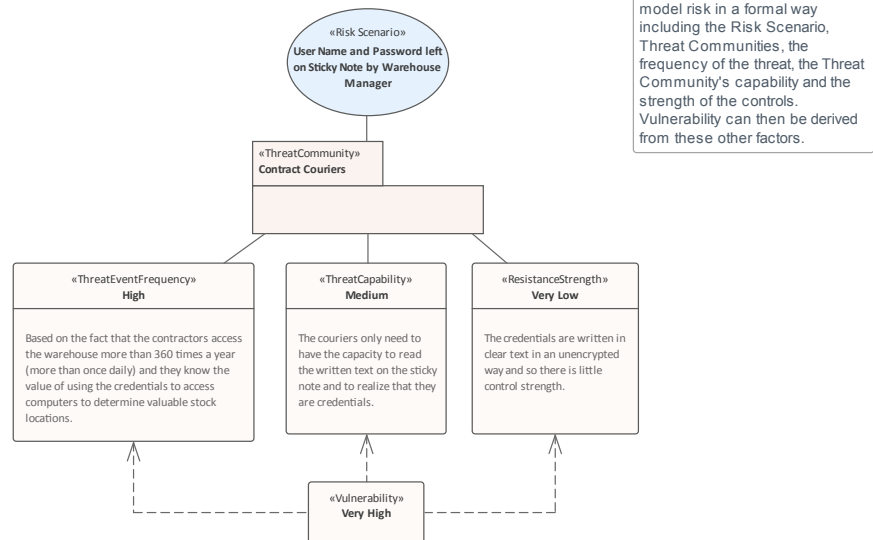


## Risk Taxonomy

The UML Profile for Risk Taxonomy provides support for modeling risk scenarios and analyzing risk conditions.

The Risk Taxonomy profile provided by Enterprise Architect supports the Open Group Standard for Risk Taxonomy (OR-T), version 2.0.

Risk Taxonomy diagrams and their associated toolboxes are available from the Risk Taxonomy category.



### Where to find the Taxonomy

Ribbon: Design > Diagram > Add > **Risk Taxonomy** > Risk Taxonomy  
 Browser window Toolbar : New Diagram icon > Risk Taxonomy > Risk Taxonomy  
 Browser window context menu | Add Diagram... > Risk Taxonomy > Risk Taxonomy

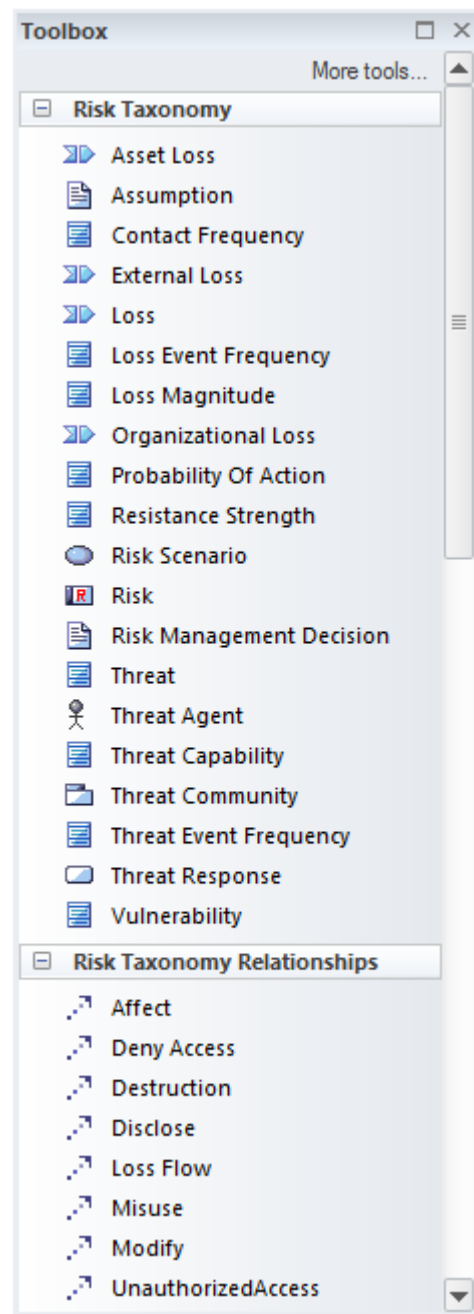
### Usage of Risk Taxonomy

**Risk Taxonomy** provides a common language and references for Security Analysts and Business Analysts who need to understand and analyze risk in a formal way. It allows Analysts to estimate the probable likelihood,

frequency and magnitude of future loss.

**Options for  
Risk  
Taxonomy**

**Risk Taxonomy** can be used at varying levels of formality depending on the initiatives, processes and requirements for risk assessment. The **Relationship Matrix** could also be used to record the relationships between the discrete values, Threat Capability and Resistance (Control) Strength to determine the derived Vulnerability. Toolbox pages of elements and relationships are available for the Risk Taxonomy diagram, allowing sophisticated models of risk to be created.



**Learn more  
about Risk  
Taxonomy**

[Risk Taxonomy](#)

# Security

## Getting to know Security

### **Introducing Security**

The Security system in Enterprise Architect is designed to facilitate collaboration, not as a barrier to incursion.

The information contained in the Repository is a valuable organizational asset that needs to be maintained and secured as such. The asset must be protected from both intentional and inadvertent compromises of content. The Security system allows update functions to be restricted to a set of users or groups with the appropriate defined permission. Packages, elements and diagrams can be locked by users, preventing others from updating them.

Surname	Firstname	Login
Administrator	The	admin
Dean	Pauline	paulinedean
Howard	Tim	timhoward
Nichols	Greg	gregnichols
Nielsen	Ken	kennielson

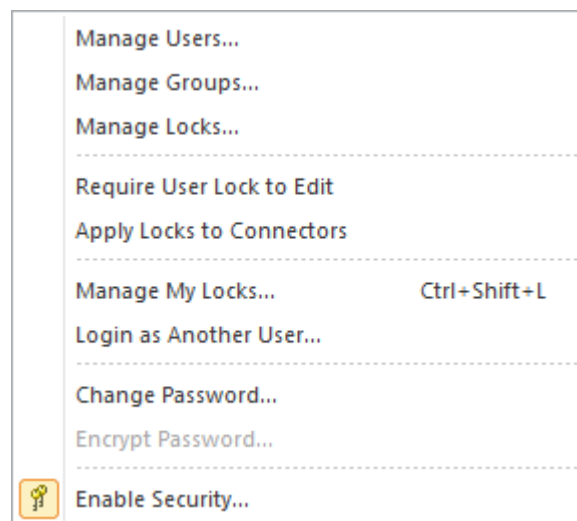
**Where to find Security** Ribbon: Configure > Security

**Usage of Security** Security has been designed to restrict access to update functions to groups and users who have been granted the access permissions to perform these functions. Packages, elements and diagrams can also be locked for change by a user under a single user ID or group user ID, preventing others from changing them. The Security system has been designed primarily to facilitate collaboration and cannot be used to restrict users from viewing parts of the model.

**Options for Security** Security itself is optional in Enterprise Architect, and by default is not enabled. If it is required, security can be enabled and a security policy can be set. There

are two policies that can be set to dictate the way security functions:

1. Require User Lock to Edit - (More Rigorous Policy) the whole project is locked against editing and the user must deliberately lock an element to be able to edit it.
2. User/group locking - (Less Rigorous Policy) the whole repository is unlocked. When a modeler edits an element or diagram the element or set of elements is automatically locked, preventing others from editing them.



Any number of users and groups can be defined. Users can be given individual permissions and also placed into one or more groups that have additional permissions defined. The set of permissions for a user is the sum of their individual permissions plus those of any groups they are assigned to. Users can be

defined manually or imported from Windows Active Directory, allowing single sign-on using Windows Authentication.

**Learn more  
about  
Security** [Security](#)

# Stereotyping

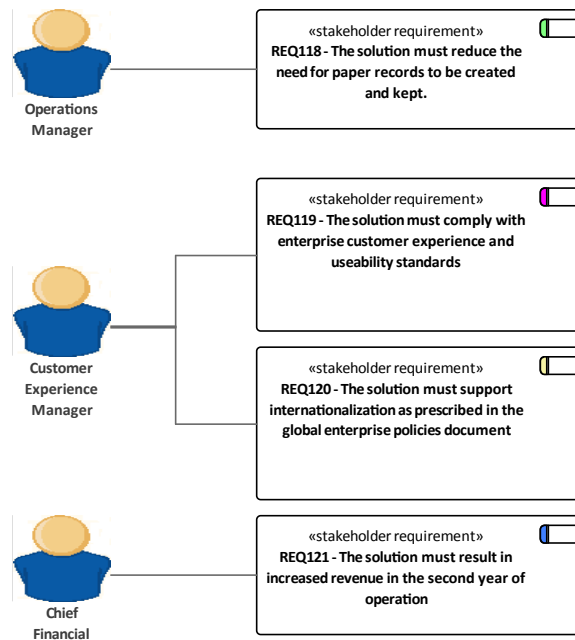
## Getting to know Stereotyping

**Introducing Stereotyping** Stereotypes are one of the Unified Modeling Language Extension Mechanism that can be used to create a new type of element based on an existing UML element type. There are many cases where things of interest in a particular industry or domain would benefit from being defined as a type. The stereotyped elements often add clarity to a model and are meaningful to a model audience who is familiar with their domain. Examples include a 'contract' stereotype in the legal profession, a 'policy' stereotype in the compliance profession or a 'meter' stereotype in the energy distribution industry.



# Stakeholder Requirements

This diagram shows a number of stakeholders and their needs (requirements). A stereotype has been created for the stakeholders, that has an alternate image assigned to it. The requirements are displayed using a rectangular presentation style, so as to display the stereotype <<stakeholder requirement>> in the diagram.



images and shapes can make the resulting models even more appealing and understandable to audiences. Care should be taken not to overuse the facility as it can lead to fragmentation of meaning and also can result in models that are not understood by external audiences.

### **Options for Stereotyping**

Stereotypes can have a number of appearance settings applied that help to distinguish the element type. These include changing the default colors of the element, or changing the appearance by assigning an image or a new shape to elements with the applied stereotype. The image is best defined in a vector format such as a metafile, so that when it is resized the image doesn't become pixilated. A Shape Script can be defined that uses a user defined script to draw the element, effectively allowing any style of element to be drawn and conditionally adding graphic features based on properties and Tagged Values.

The screenshot shows the 'Stereotype Settings' dialog for a stereotype named 'stakeholder'. The 'Group name' field is empty. The 'Base Class' is set to 'class'. The 'Notes' field contains the text: 'A stakeholder is a group or person who has interests that may be affected by an initiative or who have influence over the initiative.' Below the notes are 'New', 'Save', and 'Delete' buttons. The 'Override Appearance' section has three radio buttons: 'None', 'Metafile' (which is selected), and 'Shape Script'. To the right of these are 'Assign' and 'Remove' buttons. The 'Default Colors' section has three color pickers for 'Fill', 'Border', and 'Font', each with a 'De...' dropdown and a 'Reset' button. A 'Preview' window on the right shows a person icon with a yellow head and a blue body.

Stereotype: stakeholder

Group name:

Base Class: class

Notes: A stakeholder is a group or person who has interests that may be affected by an initiative or who have influence over the initiative.

New Save Delete

Override Appearance

☐ None

☒ Metafile

☐ Shape Script

Assign Remove

Default Colors

Fill: De...

Border: De...

Font: De...

Reset

Preview

Image of a person icon with a yellow head and a blue body.

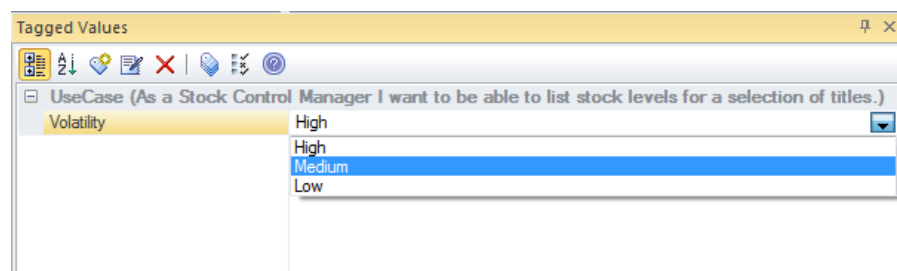
**Learn more** • [Stereotypes](#)  
**about** • [Stereotype Settings](#)  
**Stereotyping**

# Tagged Values

## Getting to know the Tagged Values

### Introducing the Tagged Values

Tagged Values are one of the Unified Modeling Language's built-in extension mechanisms. They provide a useful and productive way of adding extra properties to elements, attributes, operations or connectors; just as you would add an additional column in your favorite spreadsheet. They have a tag name and a value, and additional Notes. You can use a wide range of user interface controls to assist users when entering a value for the tag, such as drop downs, spin controls, date and color pickers, and more.



There are a variety of properties available for elements in the Repository, including name, notes, status, author and version. Even with these properties

available, it is likely that an analyst will need to add additional properties in some situations, and the Tagged Values facility is an easy to use and productive mechanism for working with these properties.

**Where to find Tagged Values**      Defining Tagged Values  
Ribbon: Configure > Reference Data > UML Types > **Tagged Value Types**

Using Tagged Values  
Ribbon: Start > Desktop > Design > Properties > Tags

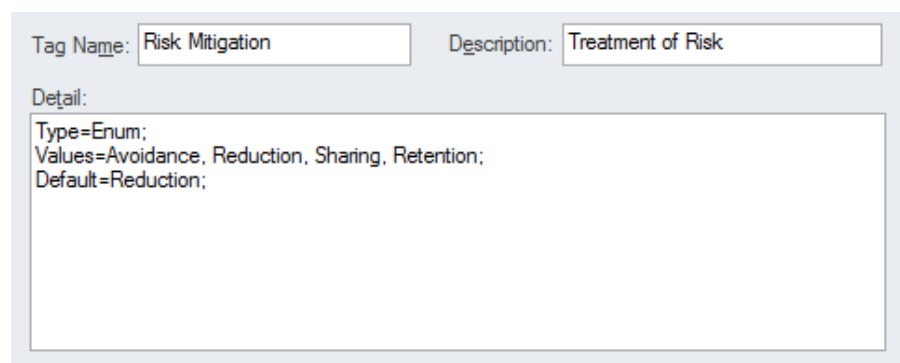
**Usage of the Tagged Values**      Tagged Values are additional properties (and their values) that can be set for any UML element, attribute, operation or connector. For example a Requirement element does not have a built in property for volatility and so a Tagged Value could be added providing the user with a drop down list of available values: such as High, Low, Medium.

Tagged Values are used extensively in profiles and technologies. For example you will use them when working in the BPMN Technology to set properties of Activities or in the Wire Framing

Technology to show or hide phone menu buttons and notification bars.

### Options for the Tagged Values

There are built in Structured **Tagged Value Types** that help the user when entering the value for a tag. The values can be restricted using user interface controls such as drop down lists, spin controls, date selectors and many more.



The screenshot shows a configuration window for a 'Tagged Value'. At the top, there are two input fields: 'Tag Name:' with the value 'Risk Mitigation' and 'Description:' with the value 'Treatment of Risk'. Below these is a section labeled 'Detail:' containing a text area with the following text: 'Type=Enum;', 'Values=Avoidance, Reduction, Sharing, Retention;', and 'Default=Reduction;'.

For example a date picker can be applied to allow a user to select the date a Change was authorized.

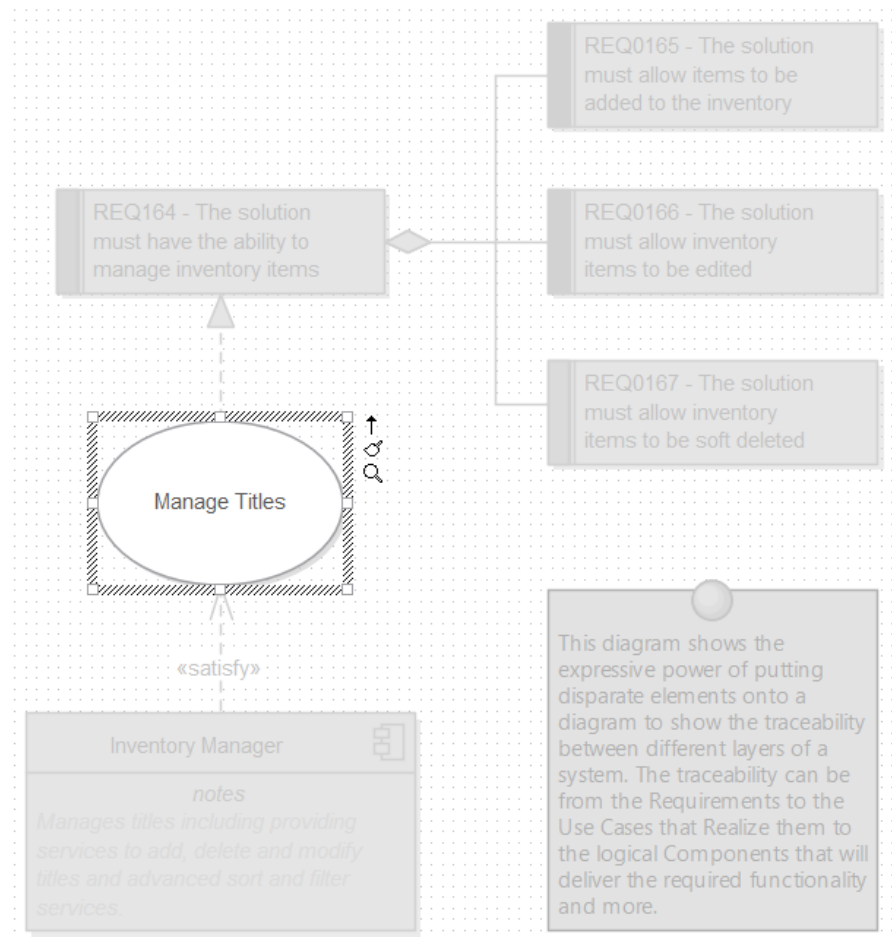
### Learn more about the Tagged Values

[Tagged Values](#)

# Visual Filters

## Getting to know Visual Filters

**Introducing the Visual Filters** Visual Filters provide a mechanism for filtering out parts of a diagram or list of elements that are not of interest, leaving just the elements and connectors that are relevant to the view. The filters can be defined for elements or connectors and there is a wide range of criteria that can be set, such as filter out all elements that don't have a status of 'Validated' and were created since a milestone date.



Model stakeholders typically have different interests and often only part of a diagram or element list will be relevant to them. Visual filters can assist by allowing you to filter out parts of a diagram or list of elements leaving just the elements and connectors that they want to see. This is a powerful tool in workshops, focus groups and meetings allowing a modeler to present a single diagram in many different ways.

**Where to** Ribbon: Layout > Tools > Filters and



## **find the Visual Filters**

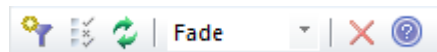
Layers

### **Usage of the Visual Filters**

Visual Filters can be defined at a Repository level making them available to all users. The application of the filter is however specific to the current user, so two analysts could apply different filters to the same diagram or list simultaneously.

### **Options for the Visual Filters**

There are a number of options available from the Visual Filters Toolbar including being able to change the way the element obscuration is presented from: Fade, Gray Scale, Hide and Select.



Context filtering can be applied so that only the element selected in the diagram and its directly connected elements will be enabled, this can be enabled from the Visual Filters window Context Menu. The filtering effect can be to hide, fade or gray scale the irrelevant elements; the Select option will conversely select the elements of interest in the diagram or list.

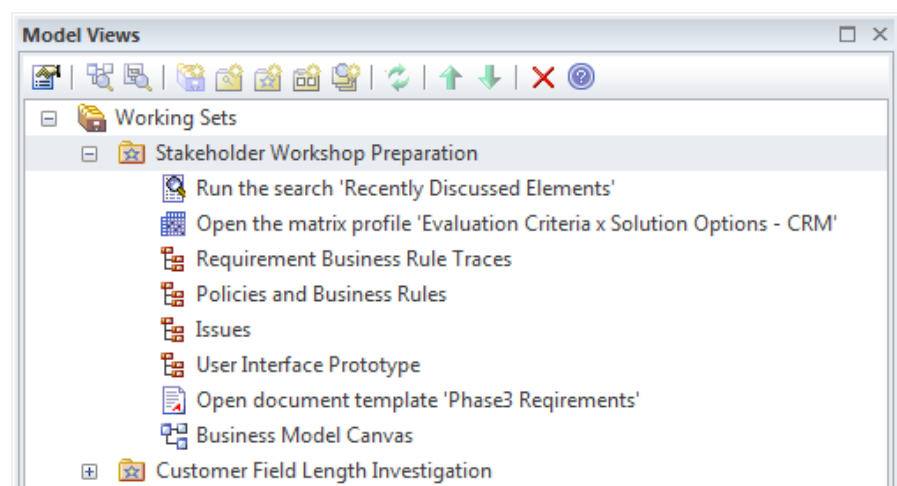
**Learn more  
about the  
Visual Filters**

[Visual Filters](#)

# Working Sets

## Getting to know Working Sets

**Introducing Working Sets** Working Sets are a powerful way of saving a group of Workspace items such as diagrams and matrices so they can be re-opened as a set at a later time. This allows a modeler to switch between multiple tasks without losing the context of the items they are working on.



A Business Analyst will often work on a number of tasks simultaneously and each task will typically require the application of a number of techniques and tools including diagrams, matrices, documents and more. Working Sets allow a user to

save groups of windows and diagrams they are working on as a set, giving the group a name so it can be easily recalled and the items opened as a set at a later time.

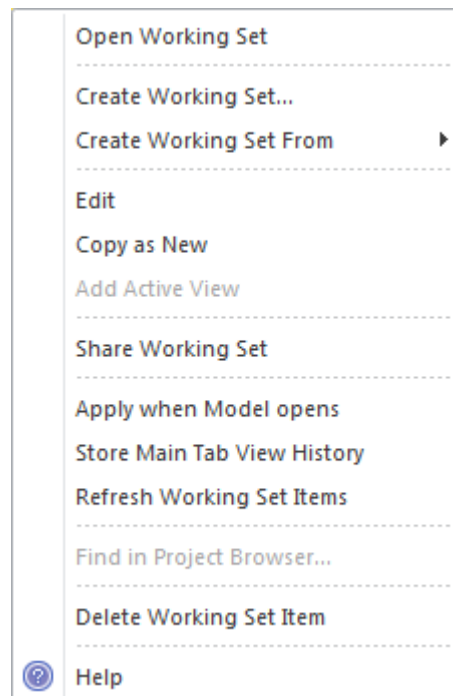
**Where to find Working Sets** Ribbon: Start > Desktop > **Working Sets**

**Usage of Working Sets** **Working Sets** can be used to store a group of items open in an Enterprise Architect workspace as a set that can be reopened at a later time. A typical scenario is that a Business Analyst is working on a particular task and has a number of relevant diagrams and matrices and documents open that define parts of a problem or solution for a particular initiative. Suddenly they receive an urgent request to complete an unrelated task. They could save the open items as a Working Set so that once the urgent matter had been attended to the group of items could be easily reopened.

**Options for Working Sets** **Working Sets** allow a number of different items to be added to a set including:

- Diagrams
- Matrix Profiles
- Searches
- **Team Library**
- Document Templates
- Resource Documents

Any number of each type of item can be added to a Working Set and the contents of the set can be edited and diagrams can be located in the **Browser window**.



There are a range of options available from the 'Working Set <name>' context menu including being able to make the working set available to other model users by choosing the 'Share' option.

**Learn more** [Working Sets](#)

# **about Working Sets**

