



Enterprise Architect

User Guide Series

Governance, Risk Management and Compliance

An introduction to using Enterprise Architect for modeling and overseeing the implementation of Governance, Risk Management and Conformance.

Author: Sparx Systems

Date: 7/08/2019

Version: 1.0

Table of Contents

Introduction	3
Governance and Methodologies	4
Governance Stages	4
Governance Definition	8
Value Delivery Metamodel Language	10
Governance Tools	13
Risk Analysis and Management	21
Risk Analysis	21
Risk Management Tools	23
Compliance	29
Defining Compliance	29
Compliance Tools	30
Requirements Definition	30
Conclusion	37

Introduction

Corporate Governance can help improve accountability, reduce the likelihood of systemic failure and ensure that policies and practices are put in place to protect people, shareholders, ethics and the corporation itself. If Governance is implemented well, it should weed out problems and identify conflicts without prejudice, providing transparency to all stakeholders and providing confidence to any entities that interact with the corporation. There are countless examples where a lack of corporate governance has led to failed projects or caused reputation risk to an individual or to organizations. These disasters can be costly, potentially impact market share and in extreme examples even eventuate in the loss of life. To avoid this strong governance is essential. It is important that all key stakeholders are kept informed and that an organization acts in an ethical and transparent manner, free from legal and civil liability.

Using Enterprise Architect can ensure that your internal and external policies and procedures are conformed to in the model by mandating their governance from inception, through the design and development phases, to the end of a project. Enterprise Architect allows you to map policies and lay out regulations required in the model to ensure conformance to these regulations both internal and external.

Setting the **Directives** of a new project involves outlining strategies, analyzing and managing the risks involved in the project and setting the standards to be conformed to.

With regulatory **Compliance**, once the standards are determined and documented these need to be implemented in the modeling process using features like Model Validation for validating the model design and Traceability for tracking where regulatory requirements are being implemented.

Risks must be defined and assessed, and policies written in order to reduce threats to the project completion. These Risks need to be detailed in the model, risk mitigation strategies can then be created and relationships set from these to the relevant sections of the model, ensuring that the risk mitigation strategy is traceable right through the design and implementation process.

Having risks and directives modeled in a transparent and open fashion, ensures that all stakeholders can not only see the risks, but they can understand those risks in context and ensure steps are taken to avoid and eliminate them altogether.

With a large system development, conveying and checking of conformance requires high level viewing via Dashboards as well as communication and feedback using features like Reviews, Model Mail, and Discussions for model collaboration. Combined, these features provide effective oversight in ensuring compliance to the directives set in place.

Directives

Defining and visualizing strategy within Enterprise Architect has numerous advantages. First and foremost, by modeling the strategy within a repository, you immediately link the strategic direction and executive thinking to the processes, architectures, software and capabilities that are being directed and then modeled, all within that same repository. This is a highly effective and powerful means of ensuring that strategic direction is followed and tied to the fundamental infrastructure and direction of an enterprise.

Governance and Methodologies

Good governance helps guide good decision making for public organizations and companies. This helps to provide transparent decision making, that adheres to relevant government legislation, providing a safe and fair environment for the end-users of the designed systems, the staff and key stakeholders. Good Governance should provide a framework for eliminating complacent error or corruption, as well as the establishment of workplace norms, goals, and policy guidelines for ensuring safe work practices and safe usability of a final product.

Implementing Governance within Enterprise Architect involves various stages, starting with establishing the guidelines of what is required in a formal governance definition, and then spelling out how these requirements will be applied when running through the various stages of implementing a project. These stages vary between different organizations, as well as between different methodologies. In this section we cover these stages and look at different popular methodologies and the implementation of governance, using Enterprise Architect.

One core principle with laying out the governance methodology is to ensure that this is defined in a single repository and that the implementation of these directives are traceable back to its source. This ensures that all parties are working on the same directive and not diverging with unrelated documents.

Governance Stages

Firstly we need to cover some basic principles of Project based governance. Governance can be seen as purely originating on the CxO level, however it does need a staged implementation down through the process. There are five core stages in this process:

1. **Establishing the principals** - A high level initiative to state what the governance framework should cover.
2. **Process planning**- Define the stakeholder engagement in the project. Set up the Risk analysis, Testing and Issue Management. Define any methodology required, and compliance to any standards.
3. **Implementing governance** - Setting up the tools to be used and implementing any boundaries defined in the initial scope.
4. **Monitoring the process** - define the monitoring that oversees the control of the process flow. Ensure the validation of the governance objectives.
5. **Completion review** - Review whether the directives have been met, any risks have been averted and that the outcome is compliant with the standards that were stipulated.

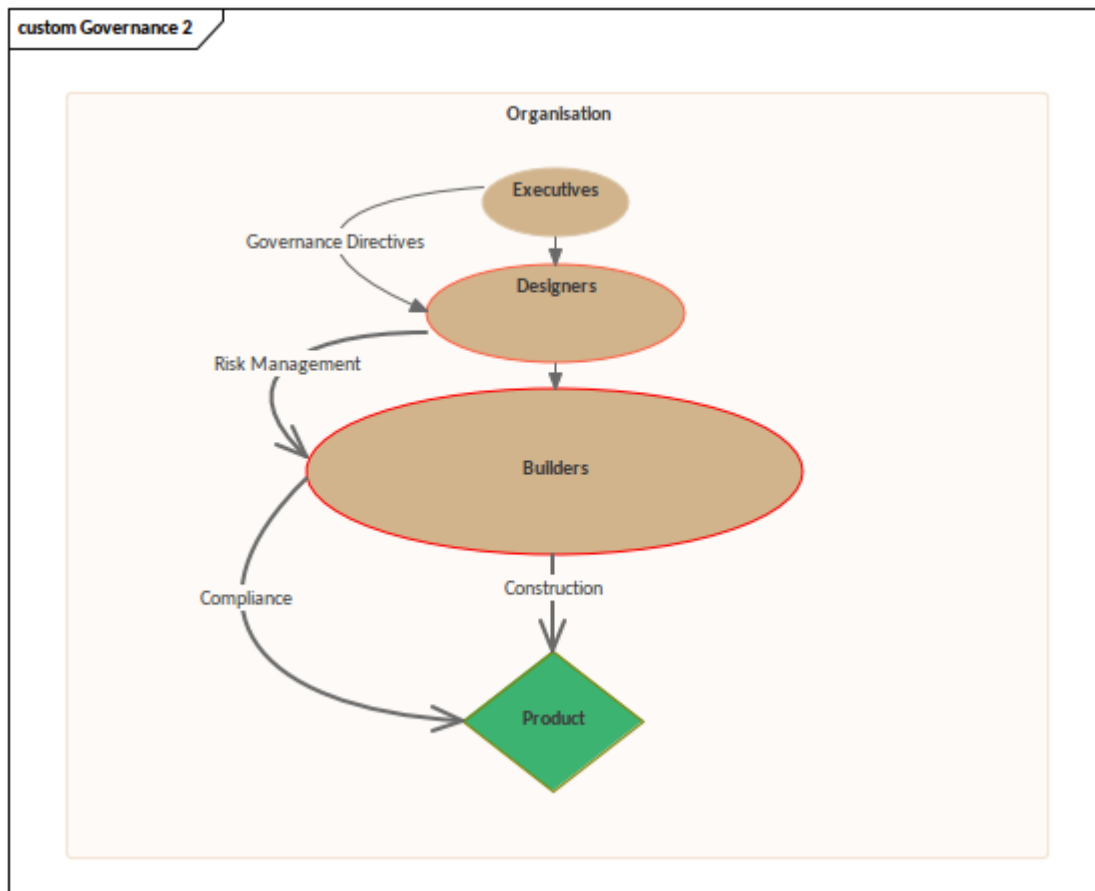


Figure 1.1: Getting an overview on the Governance, Risk and Compliance when starting a new project.

Setting the Principles

Define a Stakeholder section in the repository that includes a governance model that mandates the key principles to be implemented in the project. These principles can be outlined in a simple model document, modeled as a set of requirements, or modeled using the Business Motivation Modeling (BMM) notation.

Plan the Process

When formulating a plan of the process you need to define the tools to be used, the metrics used by these tools and the means to provide views for the stakeholders to engage in reviewing these metrics.

This includes decisions on:

- What modeling standards need to be applied?
 - Is there any need to create or alter an existing modeling language?
- Is Auditing required?
- Will Workflow be needed?
- What change management features are required?
- What level of security is required?
- What are the bounds of access to the model:
 - In-house
 - External via the cloud?

- What are the risk definition and taxonomy requirements?
- What model testing process is to be implemented?
- Is model validation required?
- Is Gap Analysis useful?
- What are the means of engaging the stakeholders:
 - Direct model access
 - Prolaborate
 - WebEA?

Implementing Governance

Laying out the process requires implementing the above sets of features that are mandated. This can include:

- Setting up the repository access (Enterprise Architect deployment)
- Defining the security of the model and the domain in which it is operating
- Setting up any MDG technologies
- Defining Workflow scripts
- Defining the risks
- Defining the test processes (implemented as part of the workflow process)
- Setting the stakeholder overview and engagement.

Monitoring and Controlling Compliance

For a complex project, that requires compliance to standards, a means of monitoring this compliance needs to be established for a number of roles. These roles range from the top-level stakeholders, through to the architects, project managers and the test managers. Core to monitoring this is the ability to gain an overview of the team's achievement of the goals, as well as the team's conformance to the standards.

The tools that can be used for the compliance review include:

- Reporting (documents and web page generation)
- Dashboard charts
- Baseline comparisons
- Review collaboration tools
- Workflow scripting.

The distribution of these controls can be undertaken directly in Enterprise Architect or via:

- User defined reports
- WeBEA
- Prolaborate
- HTML reporting

The means of conveying feedback on the conformance can range from:

- Model Mail
- Review posting
- Chat sessions.

Completion Review

On completion of short term iterative tasks, as well as on completion of any longer term core cycles, the outcomes must be reviewed and assessed to determine if they have addressed the initial requirements.

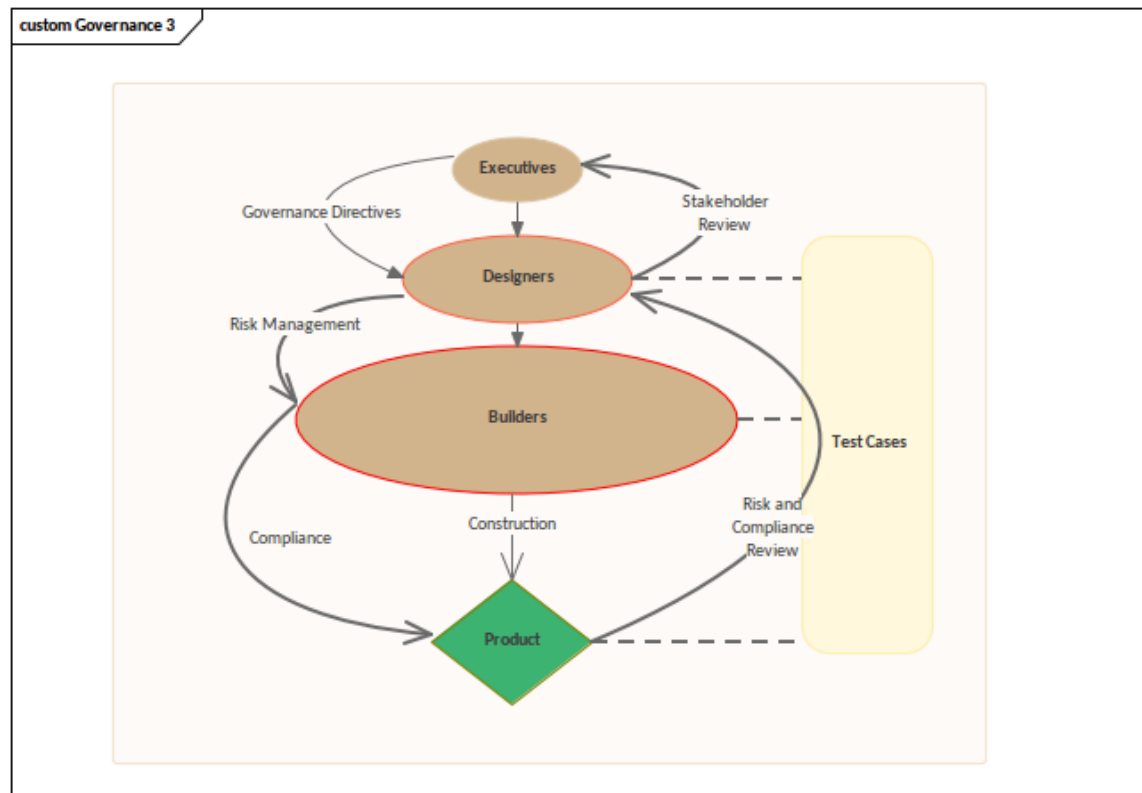


Figure 1.2: An overview of the completion and review process in line with the GRC for a new project.

Methodologies

Enterprise Architect is methodology agnostic. However it has a comprehensive set of features for implementing your chosen methodology. These methodologies range from simple, fast and low cost implementations like Agile, RAD, DevOps, XP and Scrum methodologies, through to more intense ones like Cascade Waterfall, V-Model, Spiral, RUP and Joint Application Development.

When dealing with the faster, less costly methods the governance, risk and compliance are only addressed with a minimalist approach that usually does not entail documentation or tight adherence to specifications.

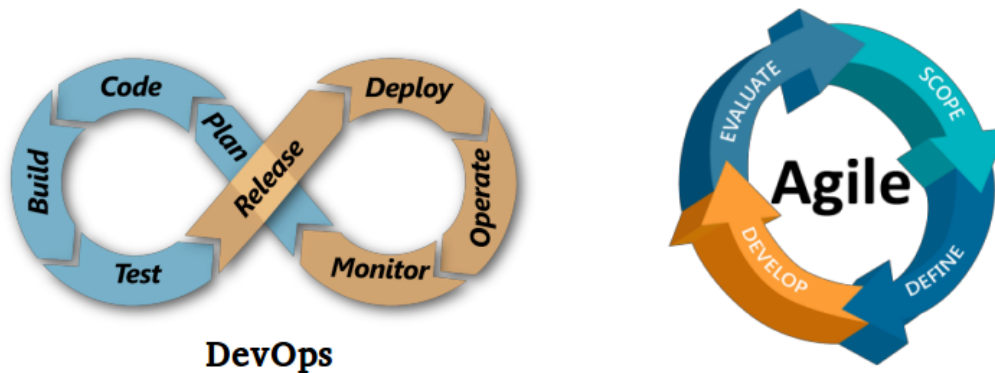


Figure 2: Two of the faster less costly methodologies - DevOps and Agile.

Accepted methodologies, based on best practice use, iterative approaches across a number of levels of abstraction and are more oriented towards tight adherence to specifications, as well as addressing risks and ensuring that there is compliance to the methods stipulated in the governance.

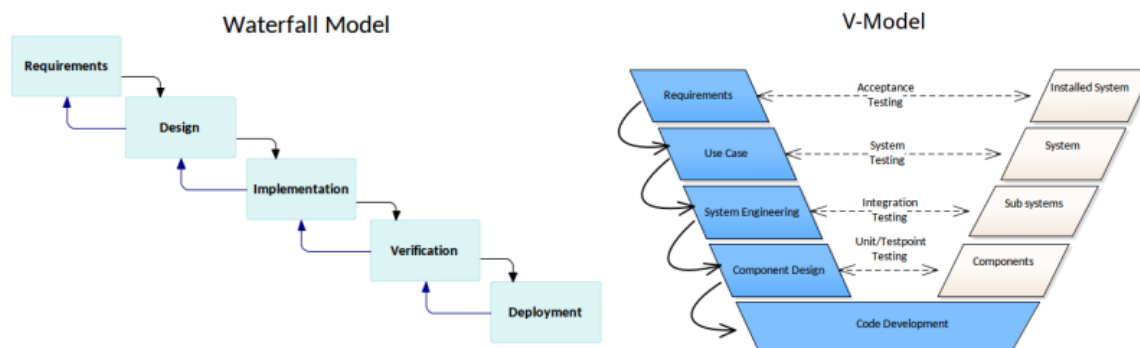


Figure 3: Two methodologies based on best practice - Waterfall and V-Model.

Governance Definition

For laying out the project's governance policies, Enterprise Architect provides a number of options which can be useful depending on the complexity, the budgeting and the how critical the application usage will be. For complex high-end projects a key tool for setting the core directives for the overall governance is to model this using the either the Business Motivation Modeling (BMM) notation or the Value Delivery Metamodel Language (VDML). Most projects would include stakeholder requirements, business analysis requirements and a domain model. For lower end projects the general governance directives can be simply defined in the Stakeholder requirements.

Business Motivation Modeling

BMM supports high level enterprise governance modeling. BMM is used for modeling processes and any influences involved in making governance decisions. The BMM modeling includes linking the governance to the operational

processes used to achieve the goals, as well as modeling the core influences and outlining the assessments of these influences.

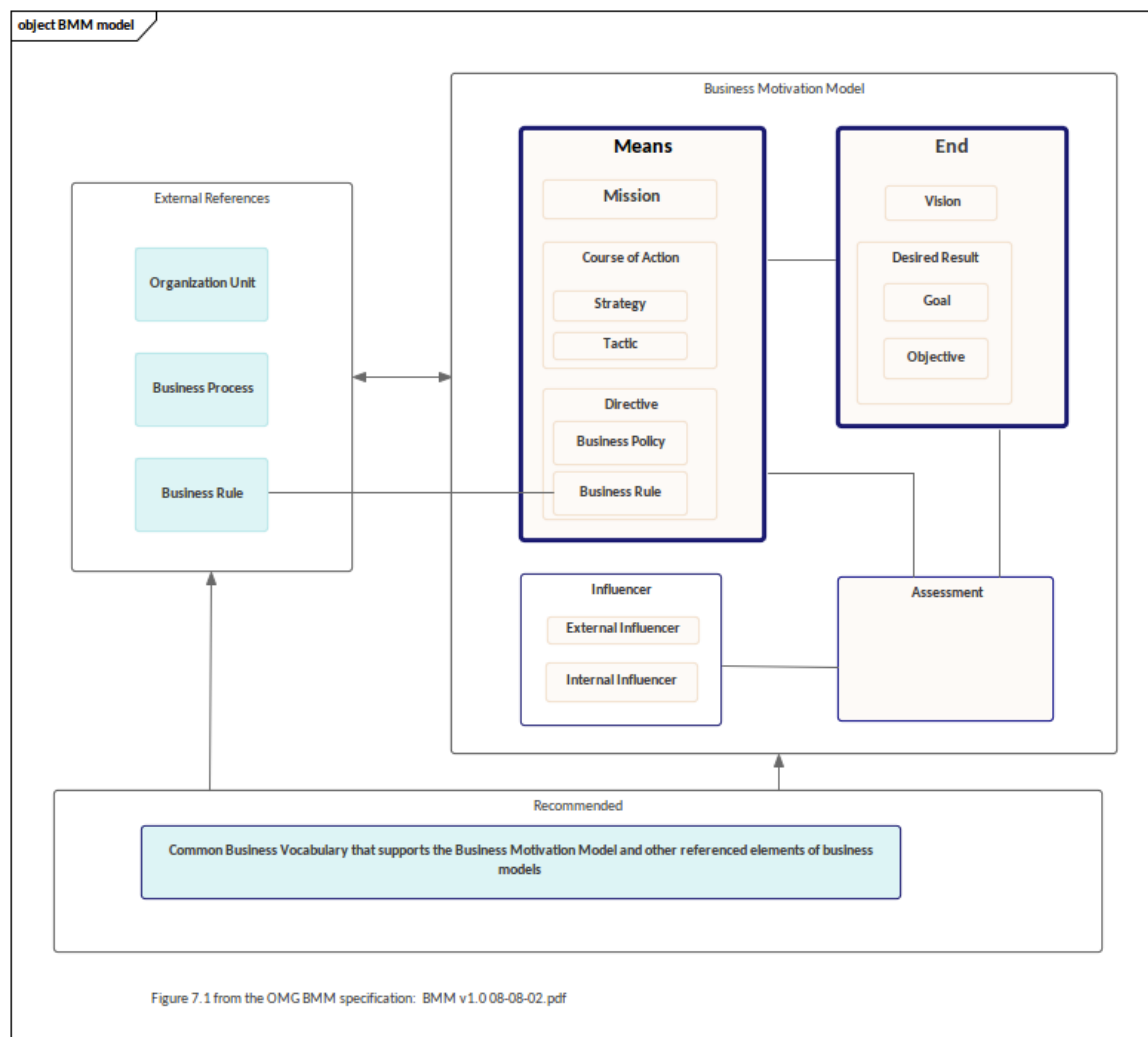


Figure 4: Overview of BMM modeling as outlined by the OMG BMM specification.

Figure 5 is a simple example using BMM to model the business motivation behind a proposed web-based business.

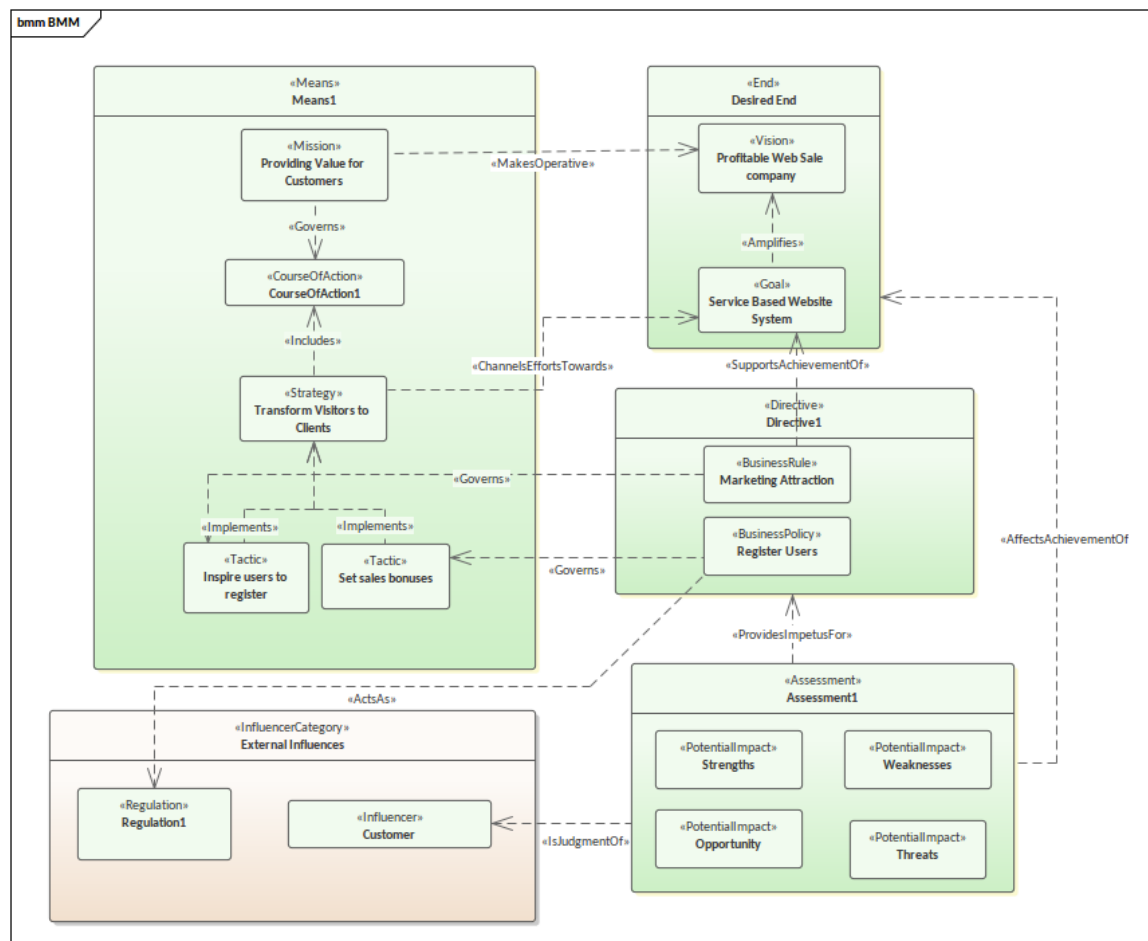


Figure 5: A BMM model of a web based business proposal.

Value Delivery Metamodel Language

In order to lay out a project's governance policies, you need to model abstractions of the business operations viewed from the executive level of the enterprise, in a form presentable to the CxO's. VDM is targeted at business executives and business analysts for modeling the connectivity between the business strategy and the business process models, including all activities, processes and roles that are involved in the general operation of the enterprise.

VDM Business Network models provide the capacity to model complex collaborations and business networks representing relationships with customers, suppliers and others in terms of collaborative activities such as a line of business, product development relationships, regulatory compliance relationships and more.

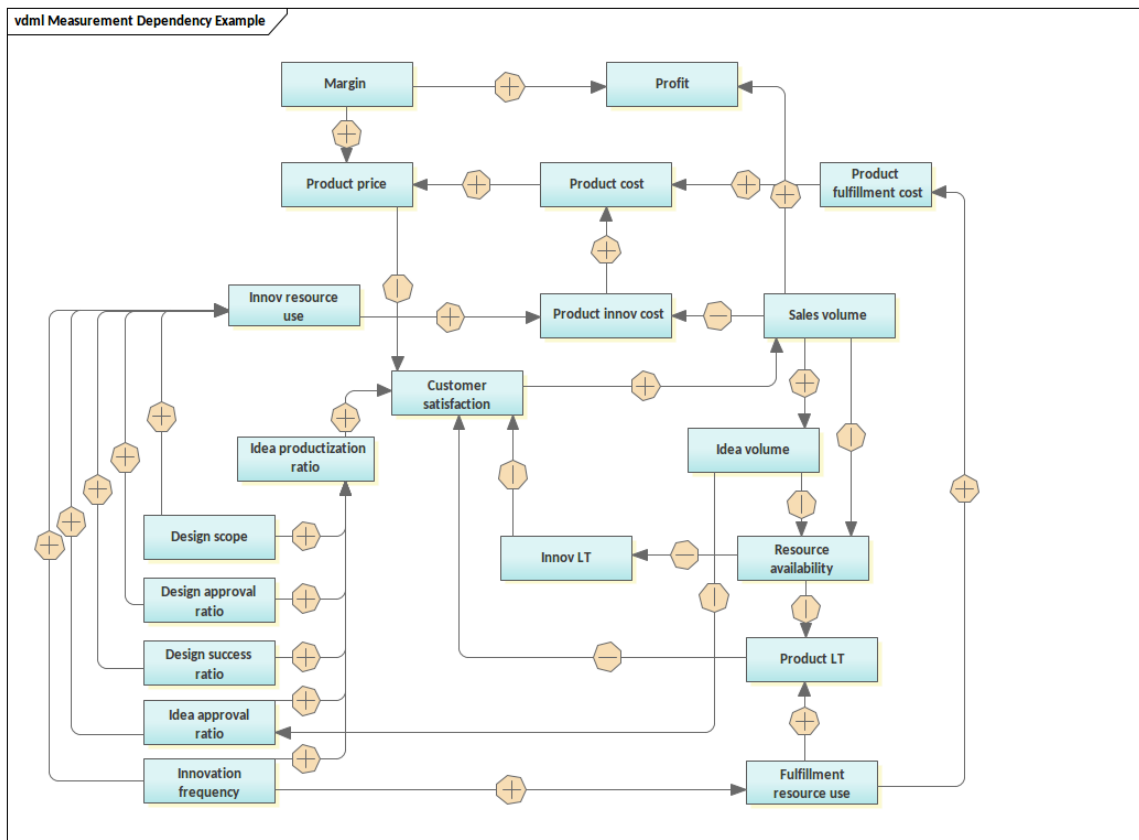


Figure 6: A VDM Measurement Dependency diagram (from the OMG VDM specification).

Stakeholder Definition

Stakeholder Requirements and compliance with them are critical in the solution development lifecycle and have a documented impact on the success of projects. Enterprise Architect has a wide range of tools and features to assist in the definition of requirements and to clearly trace adherence to these requirements, while viewing the impact that any change in strategic direction might have on their implementation.

Figure 7 is an example of an Opportunity Definition, which consists of statements on the general stakeholder positioning, an overview of the stakeholders in the organizational hierarchy and the core interests expressed by the stakeholders.

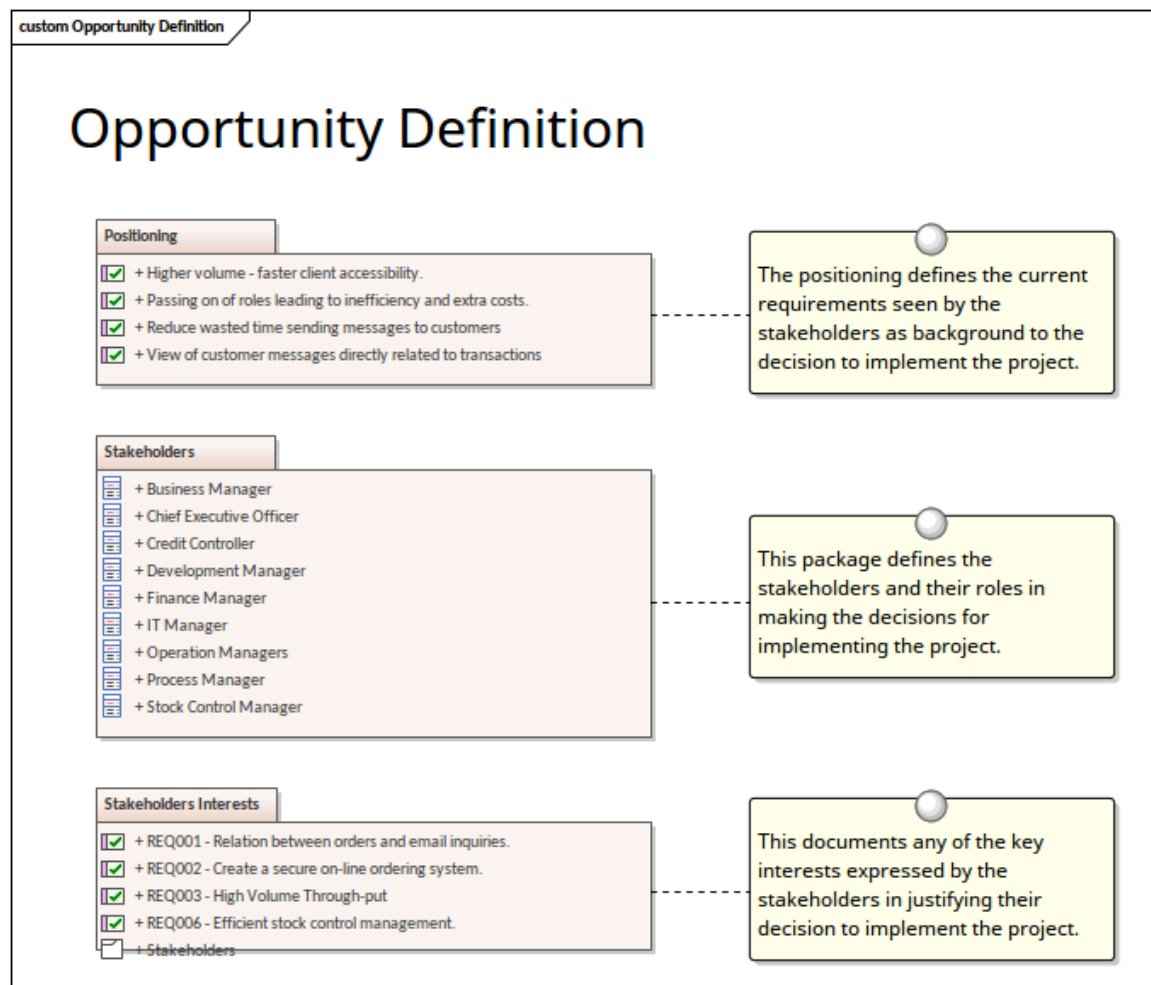


Figure 7: An Opportunity definition containing Stakeholder Positioning, Stakeholder Hierarchy and Stakeholder Interests.

Organization Hierarchy

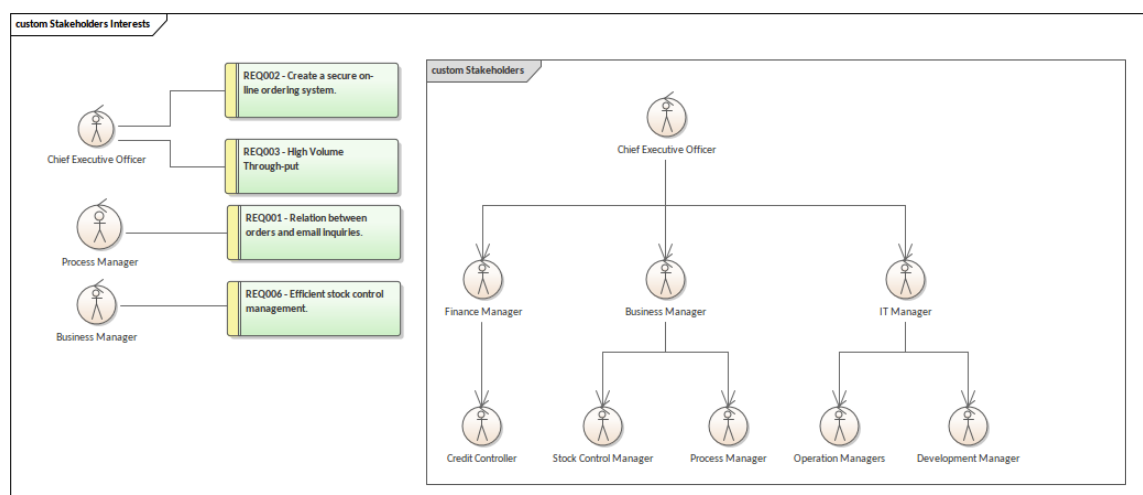


Figure 8: An overview of the organizational structure and the Stakeholder interests.

Positioning

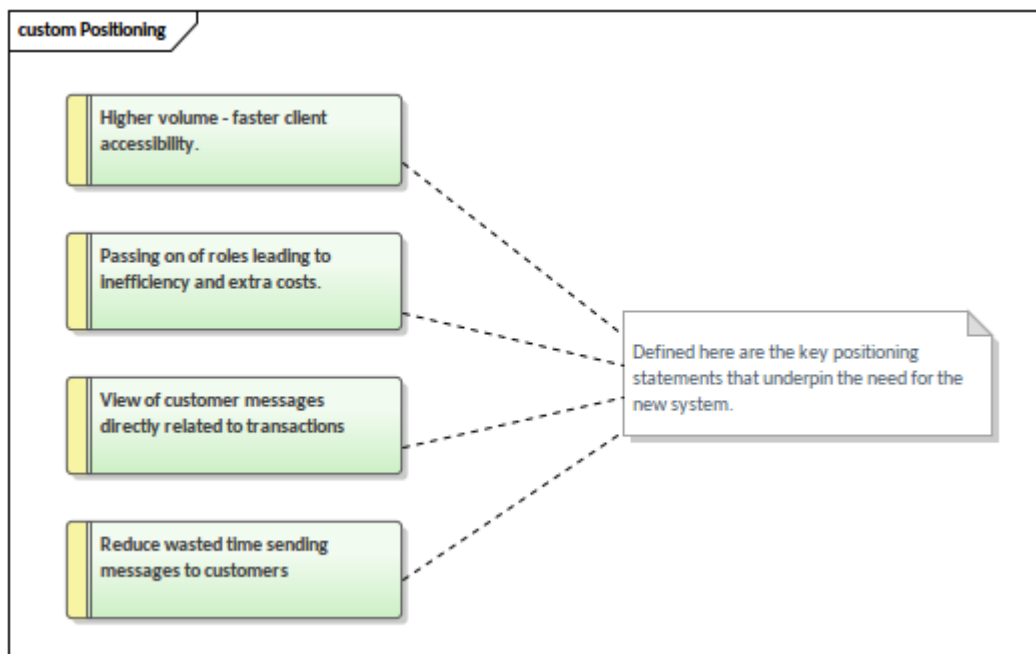


Figure 9: A diagram showing the key positioning statements.

Note: If these principles are supplied in a document, for traceability purposes, it is best to import the document as an artifact, highlight any core phrases and, from the context options, create elements from the selected text; in doing this it inherently hyperlinks the text as a reference to the created elements, that can be traceable throughout the model.

Domain Modeling

Using a Business Domain Model at a high level is a necessary part of the C-suite scoping of project. Using a Domain Model you can layout the bounds for the scope of the project, as well as the business entities to be modeled. In Enterprise Architect a Business Domain model is represented as a conceptual Class diagram.

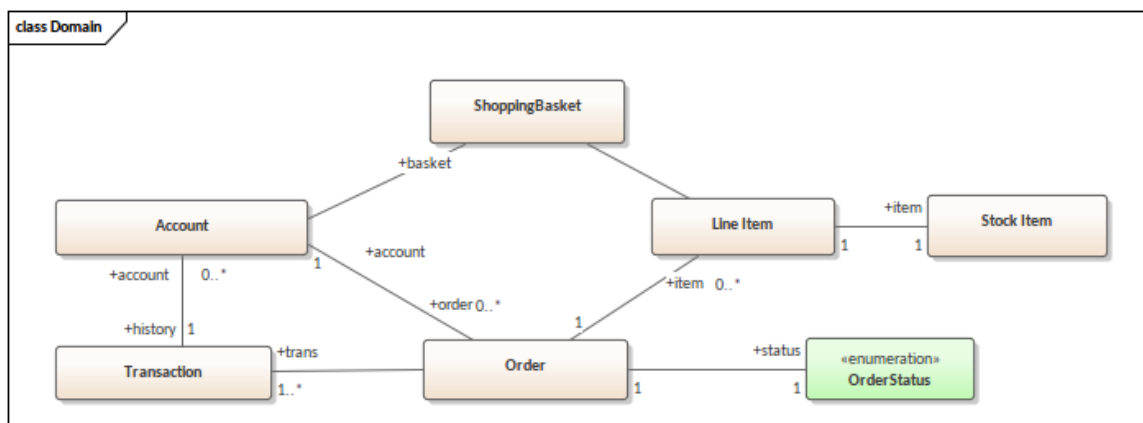


Figure 10: A Domain Model diagram showing the key objects defining the bounds on the scope.

Governance Tools

Governance action requires that the direction of the project is kept on track by delivering accurate appraisals of the

progress to ensure that the strategies are adhered to all the way through the design and development stages of the project. This can be implemented with tight adherence using a number of core features of Enterprise Architect.

Auditing

Enterprise Architect's detailed Auditing capabilities support the capturing of model changes, which can be used for examining who changed what and when. Maintaining a full audit log should be a mandatory part of any Governance strategy. Change management, governance and quality control are all aided by the use of Auditing.

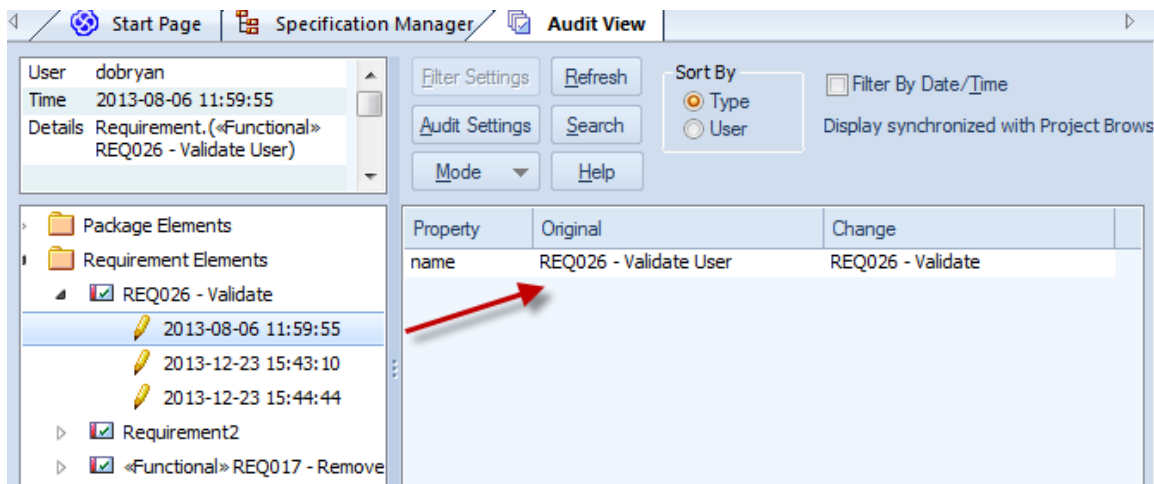


Figure 11: A simple example of the Auditing feature showing a logged change to a Requirement's name.

Workflow

Using Workflow scripting you can set a tight control on processes being performed in the model. Workflow scripts validate the user's work and actions against the policy and procedures within your model, providing a robust approach to applying company policy and strengthening project development guidelines.

Workflow - Logging Notes

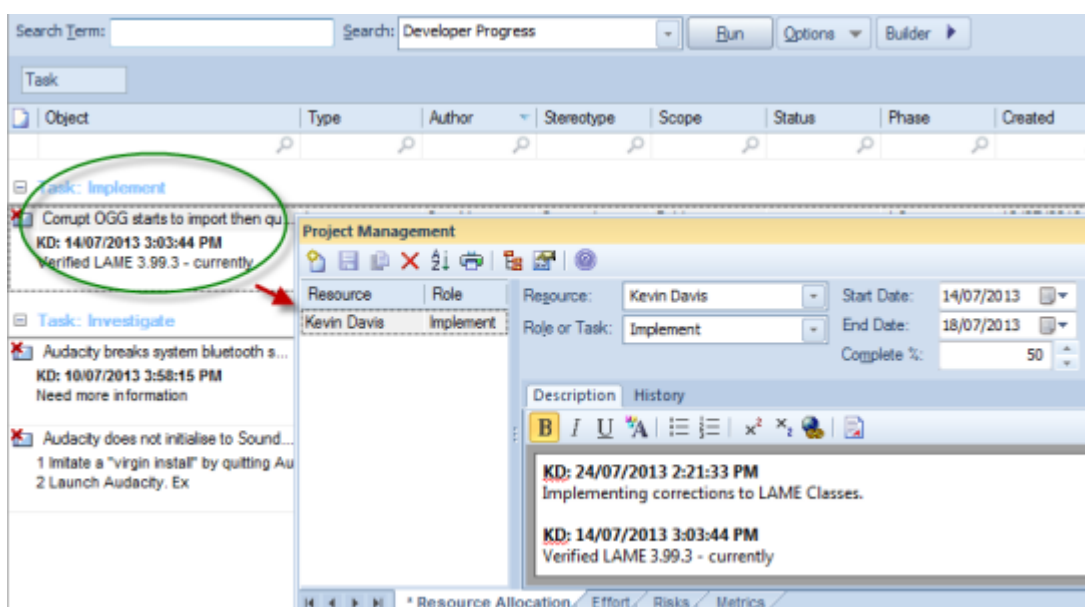


Figure 12: An example of notes covering tasks being performed in a Workflow cycle of events.

Workflow Process

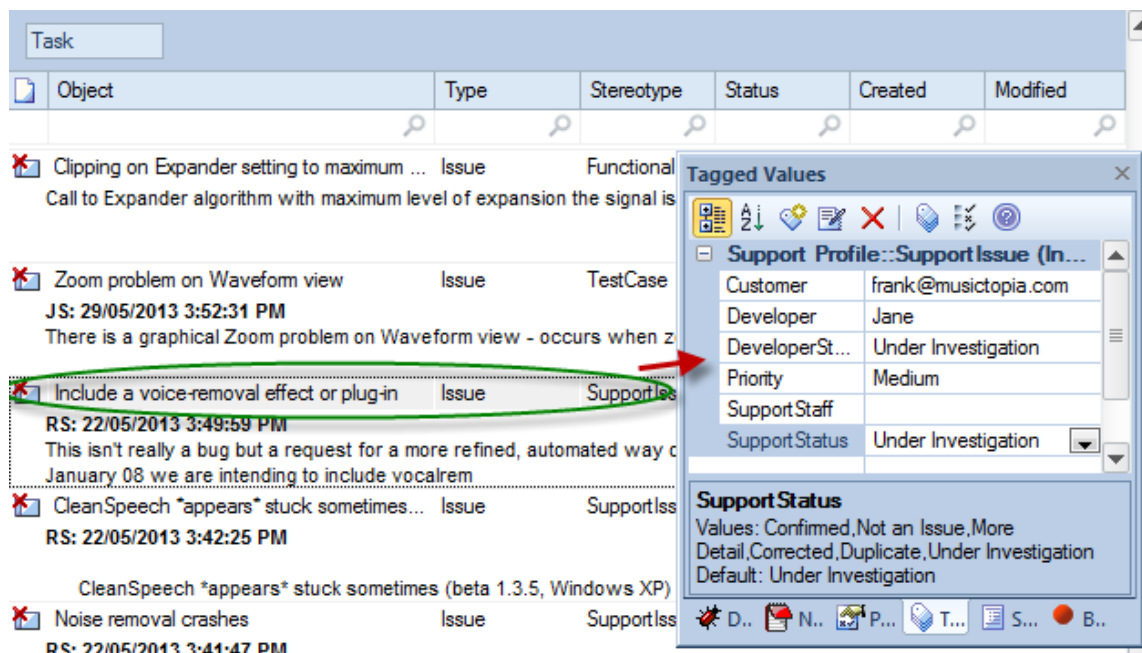


Figure 13: A example of setting a workflow process with user-defined fields (Tagged Values).

In the Figure 12 example, when setting a field status (in this case SupportStatus), the Workflow item is made visible to another user or team and the visibility for the current user is dropped, as the task is passed on.

Gap Analysis

When starting a new product version or migrating an existing system, Gap Analysis is an important step perform before a roadmap can be created, and solution options can be described and migration planning commenced. Typical gaps include issues 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.

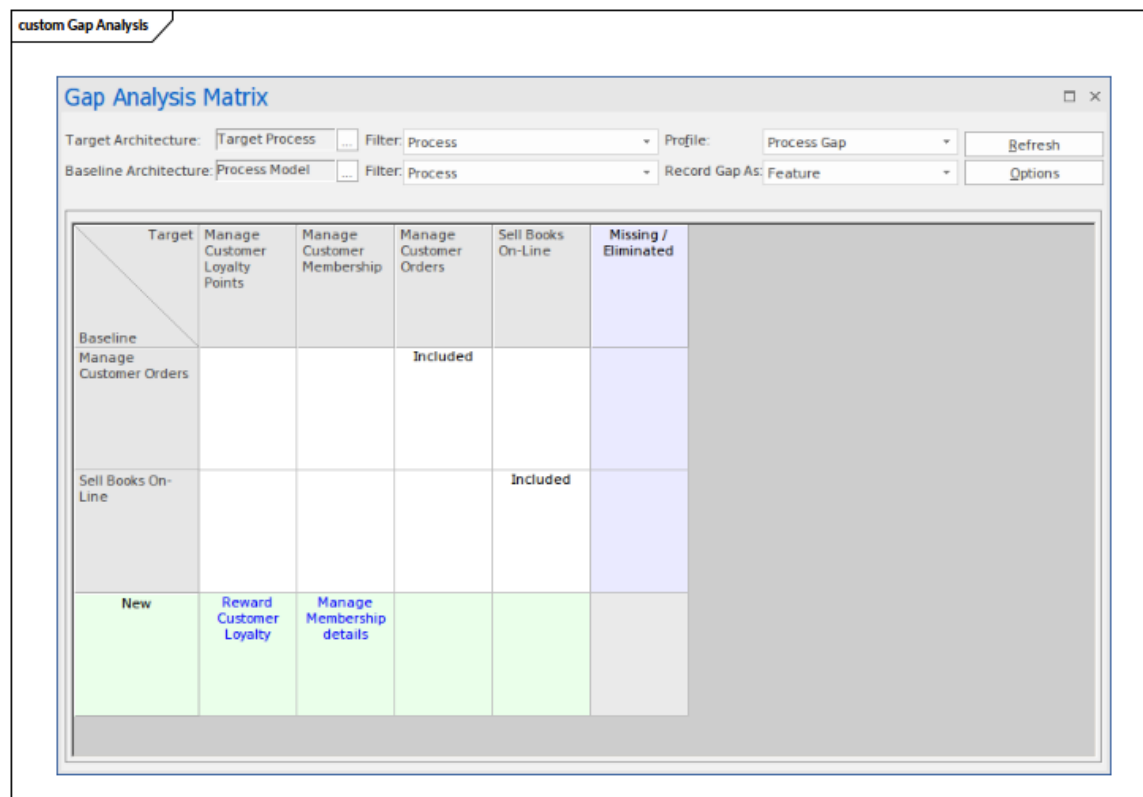


Figure 14: An example Gap Analysis Matrix mapping a baseline order process to a future target.

Monitoring the Process - Dashboards

High level reporting on the project status is essential for clear and simple viewing by C-suite executives who need to track the project's progress. The Dashboard features provide user-defined charts, Heat Maps and model views, that when combined, provide valuable summary information in a readily consumable format for a high level overview of the development process.

Standard Charts

Dashboards can include standards charts in a number of forms including pie charts and bar charts:

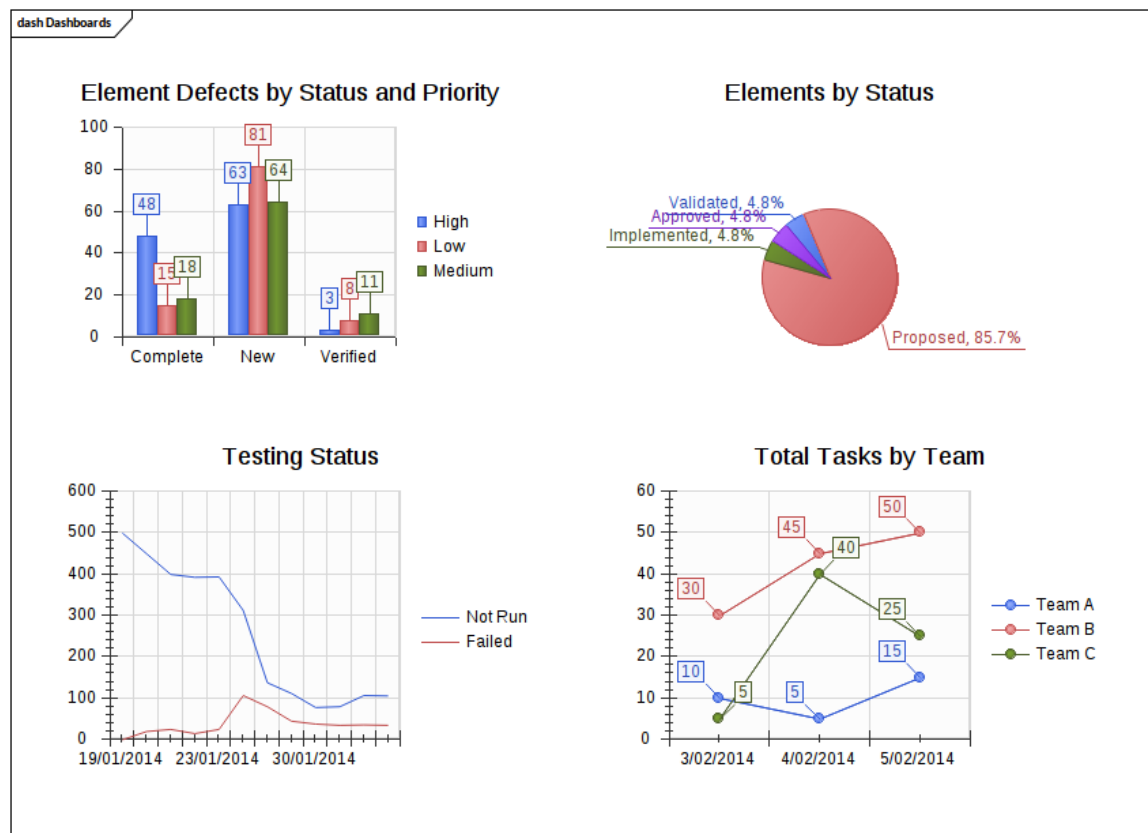


Figure 15: Examples of graphs used on dashboards.

Model Views

Model view provide a clear and simple query based, on-screen reporting of the progress of the project.

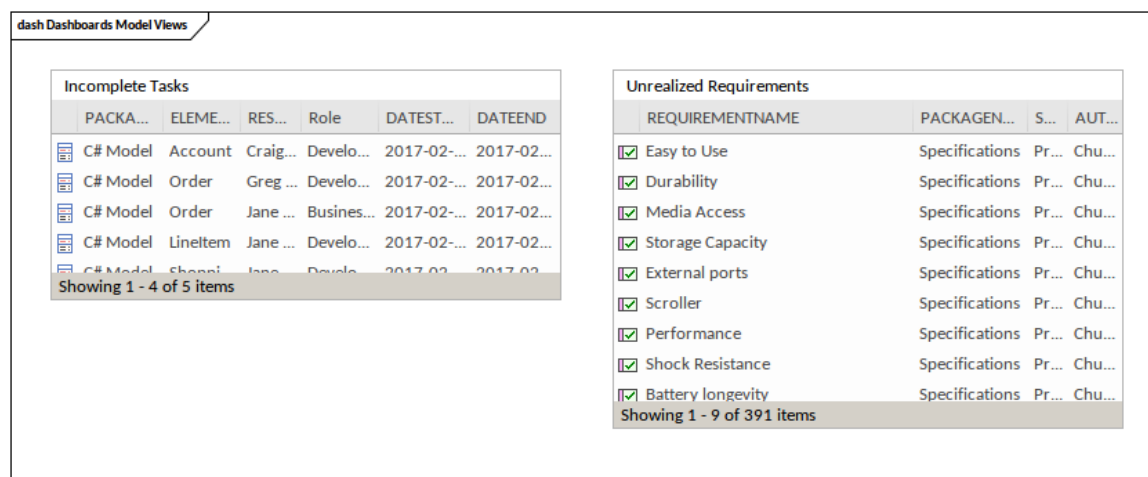


Figure 16: Examples of user-definable Model View tables used on dashboards.

Heat Maps

Heat maps are an essential tool for analyzing your data by aggregation, relevance or pattern detection, providing a visual relationship between operational and business activities for executive overview.

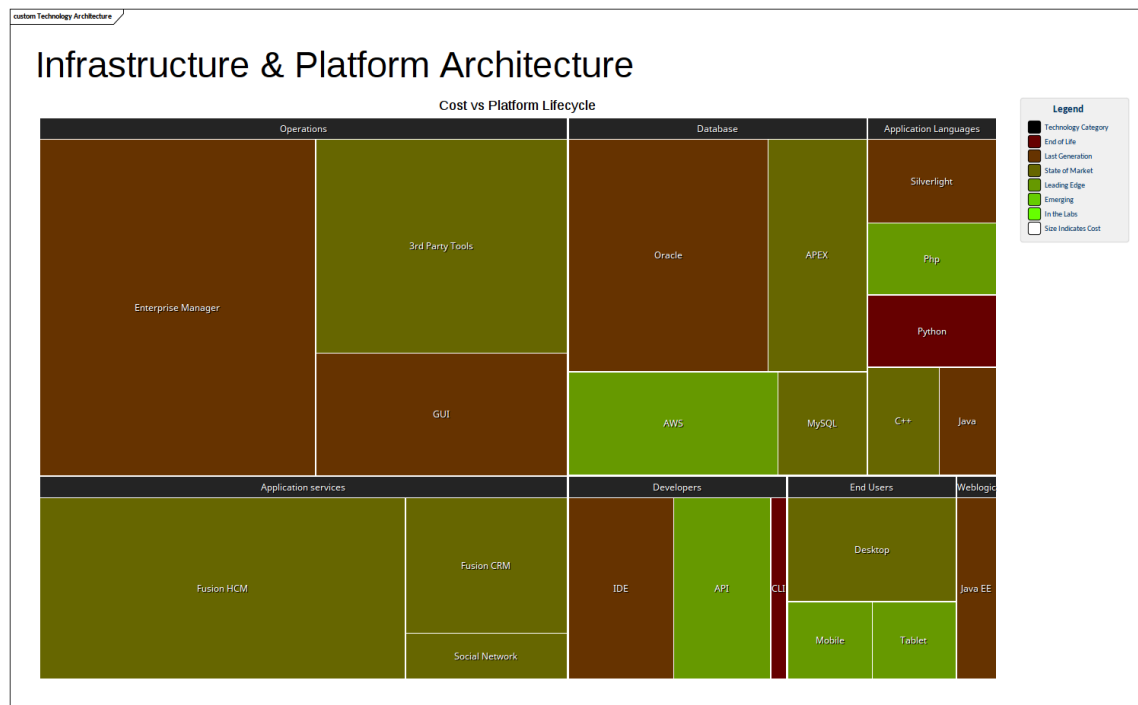


Figure 17: Examples of user-definable Heat maps for use on dashboards.

Stakeholder Engagement

With a large system development, conveying and checking of governance and conformance requires communication and feedback. The primary tools, for providing stakeholder engagement and communication, include the model collaboration features of Prolaborate and WebEA.

Prolaborate

Prolaborate provides a dashboard view for the C-suite executives to oversee the progress and ensure that there is adherence to the governance defined for a project. This includes graphs of the project progress and impact analysis graphs for an executive view of the implementation of the stakeholder requirements.



Figure 18: A dashboard view for C-suite executives using Prolaborate.

WebEA

One aspect of good governance is compliance and public disclosure. Using WebEa can ensure that all staff and stakeholders get access to the information when they need it. It provides a real-time view of content contained in an Enterprise Architect Cloud repository allowing you to enable a restricted group, or your entire organization, to read and consume models, and provide feedback to enforce the general guidelines defined in the governance of the project.



Figure 19: A real-time view of the model for C-suite executives using WebEA.

Metamodel Views

Metamodel Views provide a more focused view of diagrams by limiting the number of elements and connections

available to only the core required to achieve a specific task, to help ensure correct model creation. Enterprise Architect has a wide range of built-in diagram views, but you can also create your own Metamodels that define custom diagram views. For example, you might define a Metamodel for Requirements modeling in your organization, and then mandate that all Requirements diagrams use that diagram view as an alternative to the built-in Requirement diagram views. There is also optional security group control for diagram metamodel enforcement.

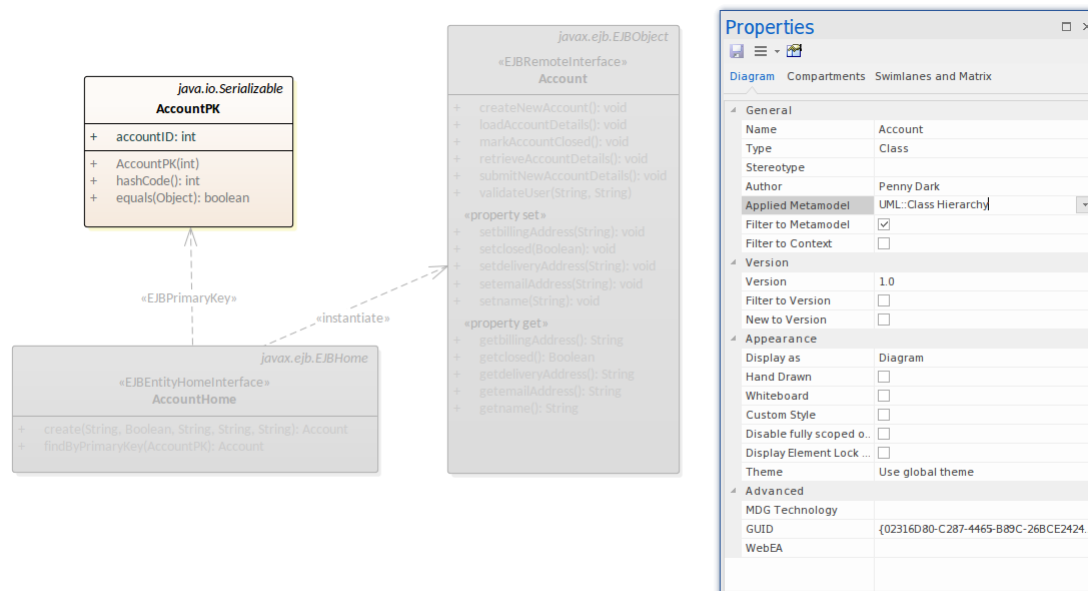


Figure 20: An example of applying a Metamodel view to narrow the focus on the diagram content.

Risk Analysis and Management

The goal of risk management is risk minimization. Operations in any reasonable sized business or organization involves risk, and often risk is related to return and liability, so by assessing potential risks and developing mitigation against these risks, you are ensuring a more consistent return and minimizing any liability.

For risk analysis and management, the risks need to be detailed in the model, risk mitigation strategies can then be defined and related to the relevant sections of the model to ensure that the risk mitigation strategy is traceable throughout the design and implementation process.

Enterprise Architect provides:

- Tools for modeling risk analysis
- Features for documenting risk management processes
- Tools used for the implementation of the risk mitigation strategies
- Features for linking the documentation and analysis to related elements in the model
- Ability to trace these links.

Risk Analysis

Risk analysis requires clear documentation of potential risks, as well as an outline of any processes used for risk minimization. The Risk analysis should clearly prioritize the risk reduction measures based on their likelihood, magnitude and ease of mitigation.

Risk Assessment

Risks can occur on different levels of the project and be anticipated by different roles. A senior analyst will typically define risks at the project level, whereas a business analyst might record risks at the level of requirements and modular units of the system such as components. When modeling the risk assessment it is necessary to try to identify:

- The likelihood that they will have a negative effect on a project and
- How large any effect is likely to be

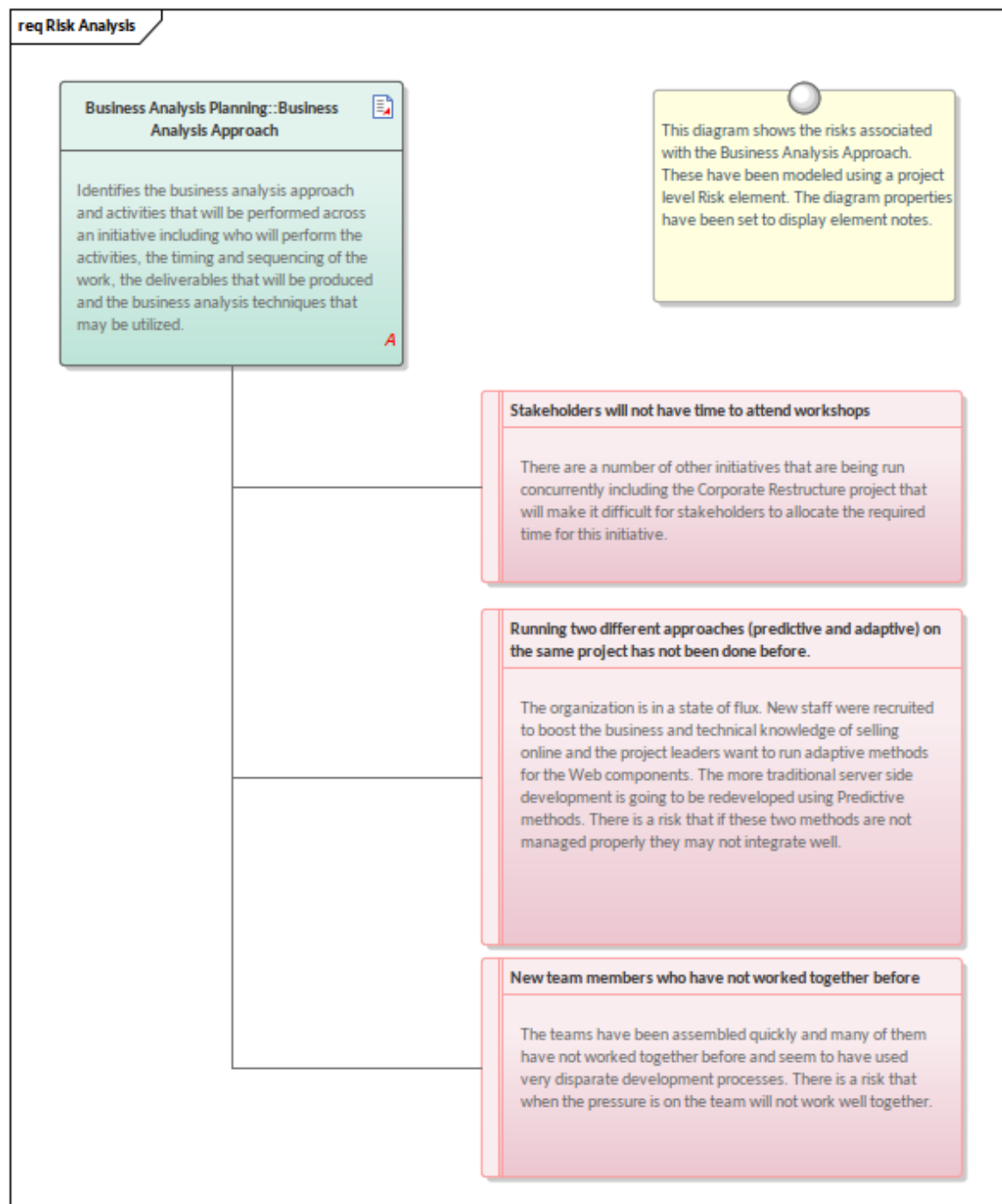


Figure 21: An example diagram showing Risk Analysis using the Business Analysis approach.

A core part of risk management is laying out Risk-Elements that identify a risk and set the magnitude and likelihood of the risk occurring, as well setting restrictions to ensure any potential risks are kept in check.

Risk Taxonomy

The 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.

This Risk Taxonomy Diagram shows the use of elements to model risk in a formal way including the Risk Scenario, Threat Communities, the frequency of the threat, the Threat Community's capability and the strength of the controls. Vulnerability can then be derived from these factors.

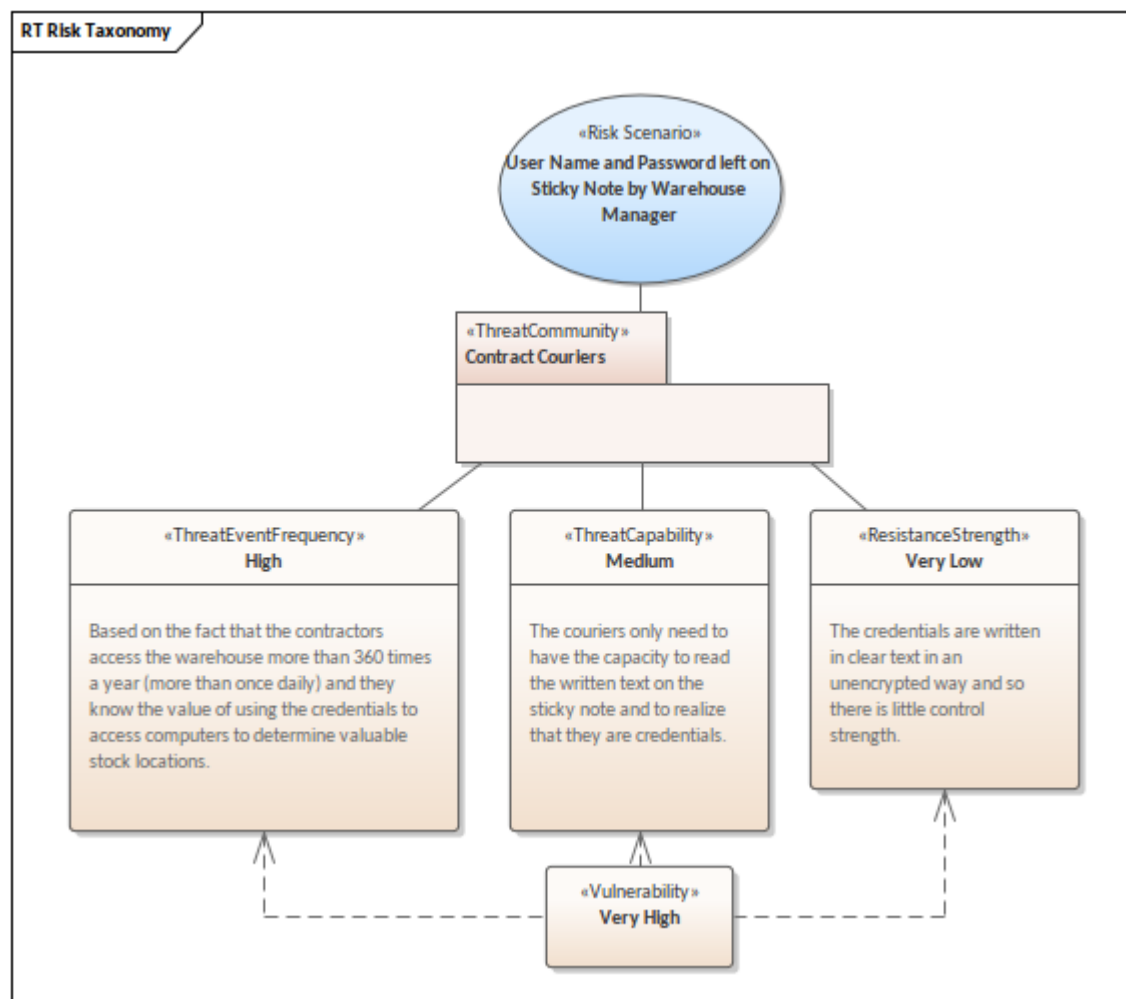


Figure 22: An example Risk Taxonomy diagram analyzing threat capability and vulnerability.

Risk Management Tools

The Traceability view is a key tool for viewing the impact of any risks defined in the model. To mitigate any risk of information loss or corruption in the model, during the development process, Enterprise Architect has a number of Change Management features, including the options to baseline or Version Control sections of the model. This can be further refined using the Time Aware modeling features, and tightly administered using the security and review features. When combined, these are a powerful means of implementing strong governance within a modeling team.

A strategy on the usage of these features might be required for a large complex project and it is necessary to define this in the project's risk analysis and stipulate the tool usage in terms of what mitigation they provide for a specific risk.

Risk Traceability

To ensure that the Risk Assessment and risk strategies are tightly controlled, any Elements defining a risk, as laid out in the Risk Assessment and Risk Taxonomy models, should be traceable to the specific objects where these policies need to be applied. The Traceability view provides a clear representation of the links to the source (Risk Assessment) and

where these need to be applied, for example Use Cases and Classes.

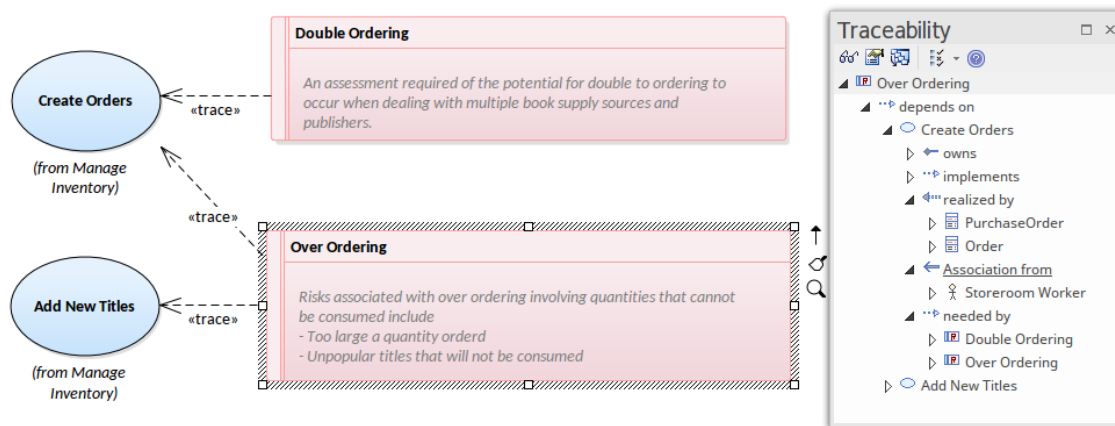


Figure 23: Tracing a risk, using Traceability view, through a Use Case and on to a Class.

Security

Some core factors in risk management are setting:

- Who accesses what
- Who changes specific areas of the model
- Who assigns tasks to be performed
- Who is assigned tasks to be performed
- Who reviews a process

All of the above factors are dependant on the security features defining who is logged in and what group they are assigned to. Enterprise Architect supports a comprehensive user security model that allows fine grained locking of individual elements, diagrams, Packages and other model constructs. These security features also govern identification in team work including workflows, task assignment and reviews.

Both group and individual permission sets are customizable by model administrators, providing a well regulated and controlled space in which competent and trusted modelers can work alongside less competent modelers or model consumers.

This can be further tightened to set restrictions on visibility of specific areas of the model using Visibility Levels which set restrictions that are strictly enforced on the DBMS (below the level of the application).

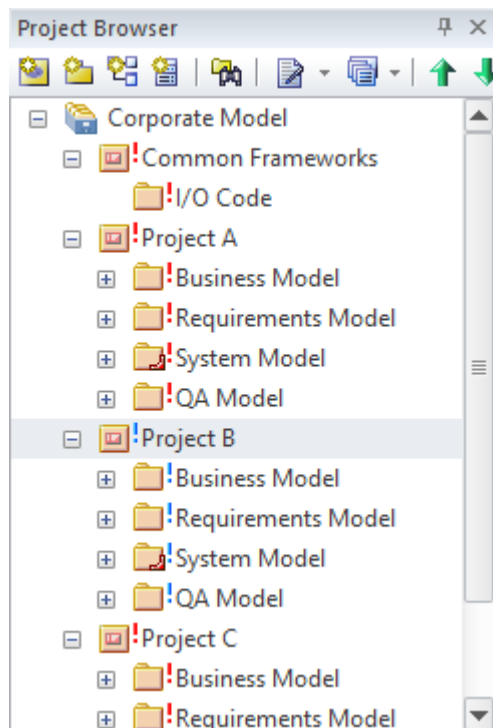


Figure 24: Example showing the security locking in the Browser window.

Package Baselines

Package Baselines are a core tool in the mitigation against loss of modeling data. Enterprise Architect provides a facility to 'Baseline' (snapshot) a model branch at a particular point in time. These baselines can be used for restoring the branch of the model back to a previous state.

Baselines also provide a tool for Branching the development where design and code implementation are staggered and the design needs to progress on to the next version while a stable model state is required for the developer to implement this. Baseline Branching minimizes the risk of version creep occurring in the implementation of a version.

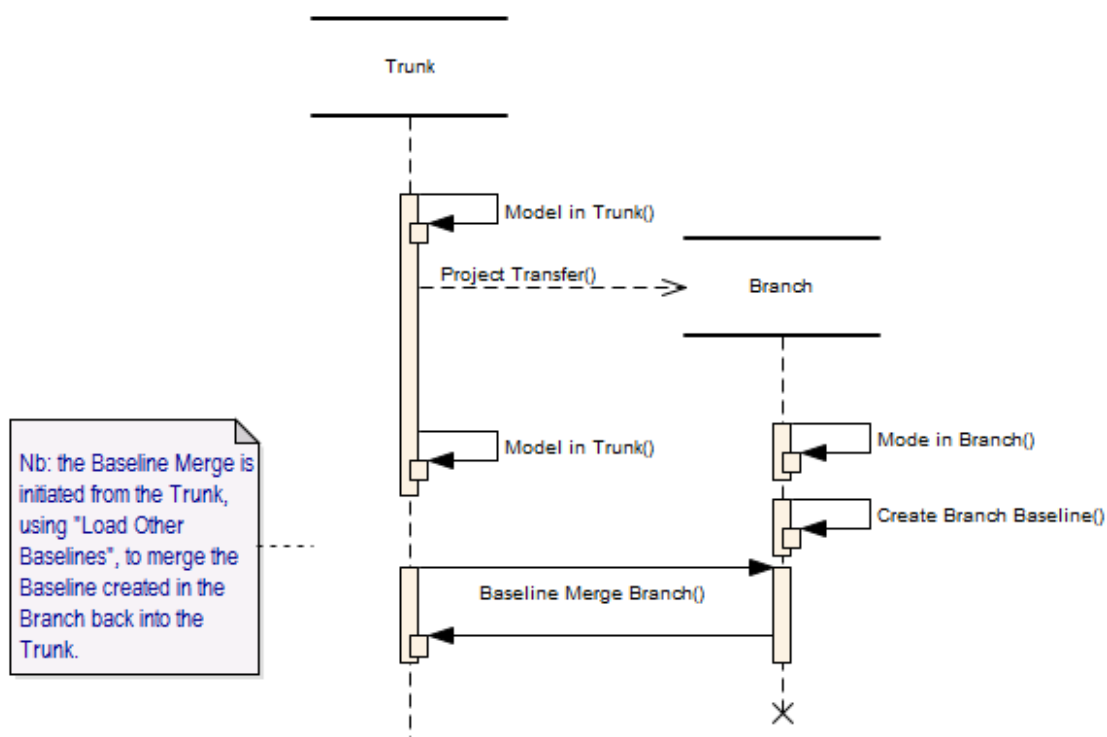


Figure 25: A Sequence diagram showing the process for Trunk and Branch type interchange between two models using the Baseline Merge features.

Version Control

When developing a complex application with a strong focus on a tight control over the design, version control of updates to the model can be a good tool for remediation of any errors or for any data-loss issues. By rolling back versioned sections of the model, you can reduce the risk of any small change causing a deviation that can amplify as the process progresses.

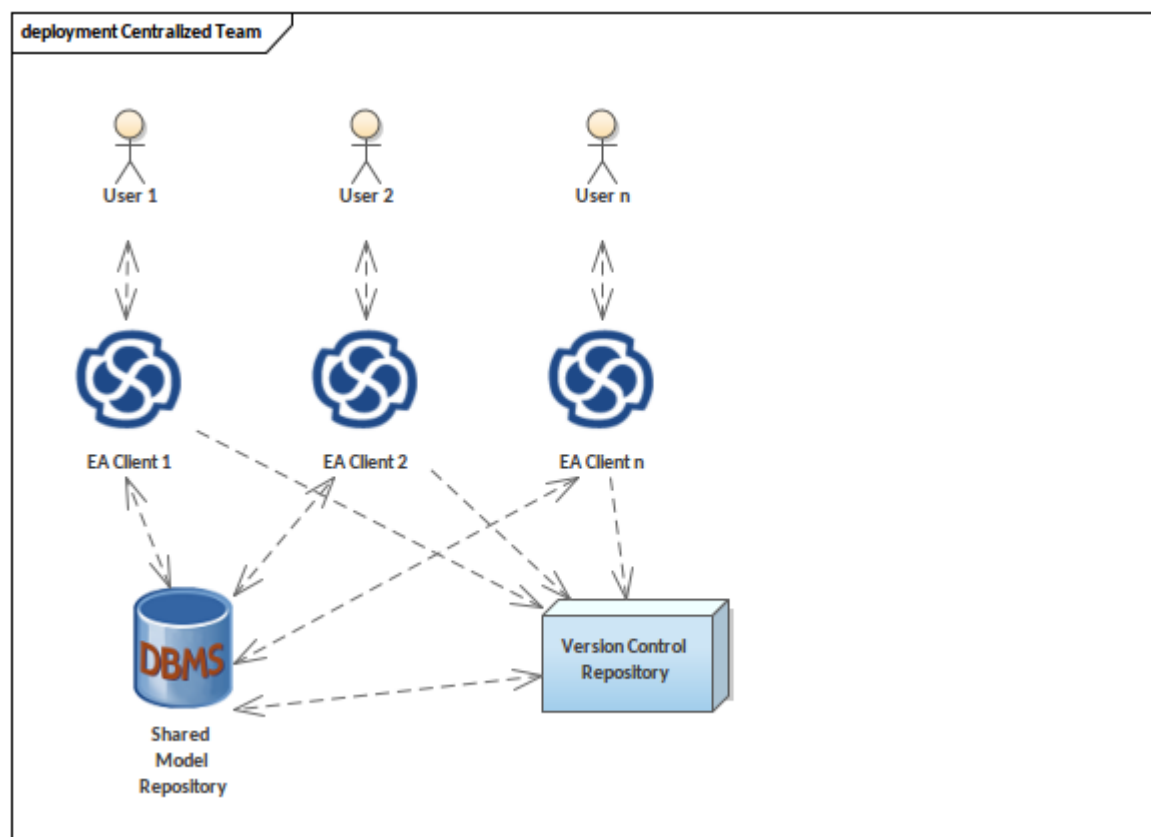
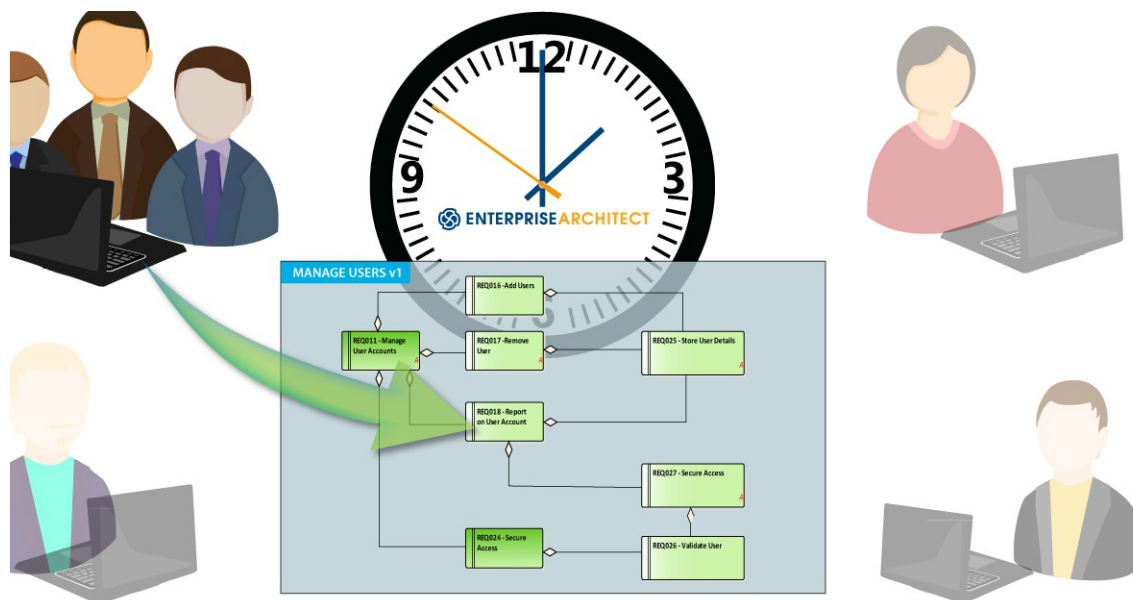


Figure 26: An example of one type of interchange using Version Control for versioning Packages.

Time Aware Modeling

Time aware modeling mitigates risk by benchmarking a current version state then allowing for detailed explorations of how processes and structures will evolve over time. 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.



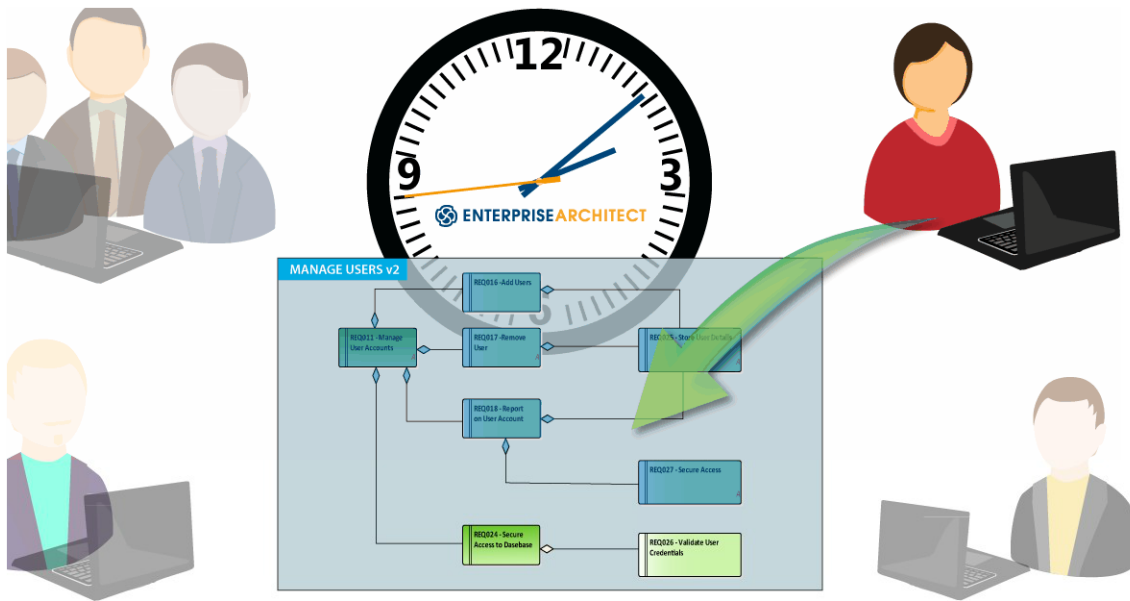


Figure 27: Using Time Aware modeling for evolving a model over time.

Compliance

Standards are important specifications with which all the roles in application development, from architects through to testers, need to comply. Conformance to international standards, as well as compliance with regulations require that these be mandated early in the design then properly overseen in the development cycle.

When overseeing the governance there are a range of standards to be considered, including:

- Organizational standards (internal to the organization)
- Industry Standards (governed by Industry bodies)
- Regulatory Standards (mandated by international, national or jurisdictional law).

Core to ensuring conformance when designing and developing a project is the stipulation of what standards must be adhered to.

Defining Compliance

All of these standards must be represented in the Repository and mapped to applicable parts of the architecture to indicate compliance or dispensation.

Documenting Standards

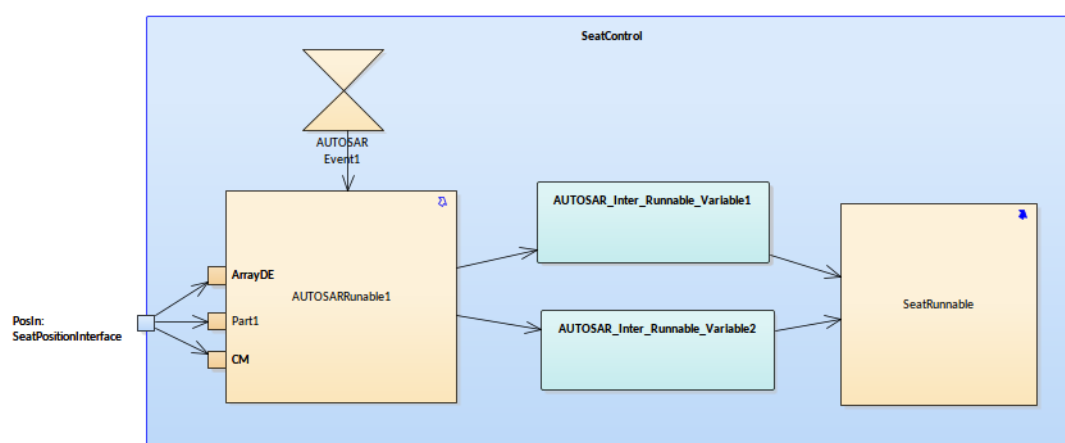
Where the governance stipulates compliance with a set of existing global standards that are mandatory in the implementation of the design, those standards must be referenced in the model and requirements created to define them.

For example, in the auto industry there are standards such as:

- ISO 26262 - Functional safety of electronic systems in production automobiles

There are also modeling frameworks that might be needed to ensure compatibility between manufacturers of parts and assemblers of parts. Again, these must be clearly mandated as requirements in high level governance documentation of the model. For example, in the automotive industry there is:

- AutoSAR - AUTomotive Open System ARchitecture



AutoSAR model of an automobile seat controller.

Figure 28: An example using the AutoSAR automotive standard for modeling a car seat controller.

Defining New Standards

Enterprise Architect not only covers a wide range of languages defined by the international bodies such as the Object Management Group and the Open Group, it also supports a comprehensive platform for institutions and organizations to define their own language or Framework. The support for the creation of your own language includes defining Profiles for Element-types and Diagram-types, as well as the option to enhance existing modeling languages like SysML, TOGAF and ArchiMate.

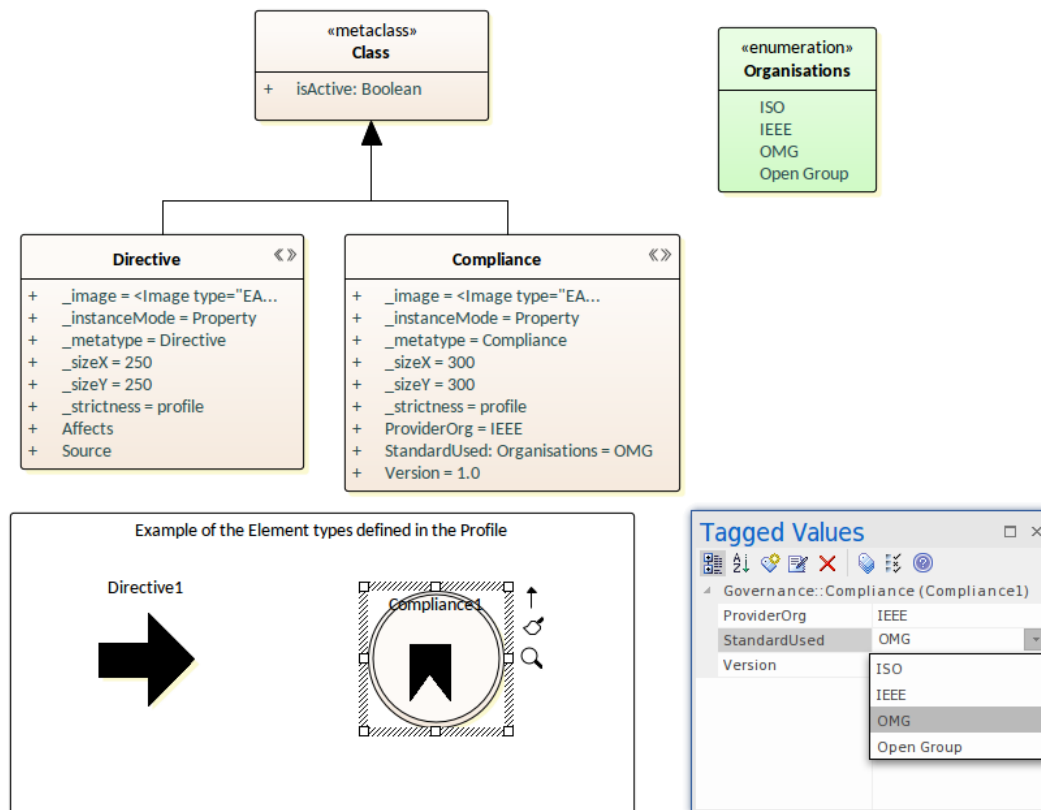


Figure 29: An example of setting a new standard as a modeling language using a Profile.

Compliance Tools

Enterprise Architect offers a range of tools for the implementation of compliance standards.

Requirements Definition

As compliance means conforming to a set of stated requirements, it is Enterprise Architect's requirements management features that are core to implementing this.

Any policies, national or international laws, regulations or standards that need to be complied to, can be documented using Requirement Elements in standard diagrams or using the Specification Manager ready to be linked to the next level that realizes these.

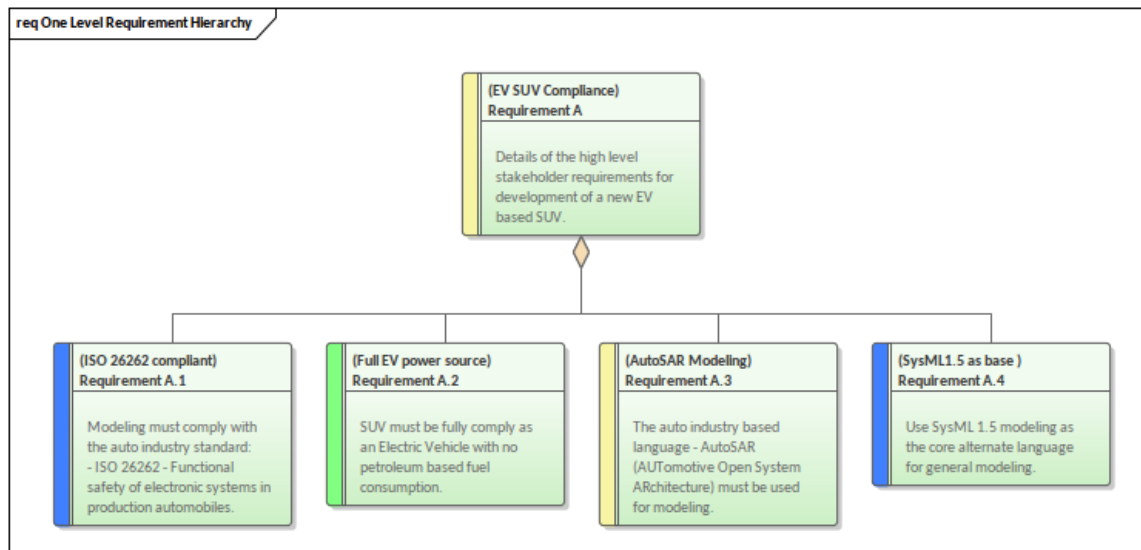


Figure 30: An example of Requirements setting the compliance to external specifications.

Traceability and Impact Analysis

Using Enterprise Architect's traceability features, a requirement, for example, could be traced from a stakeholders statements, which in turn could be traced to a set of drivers and goals, and these could be traced to a set of regulatory constraints. Enterprise Architect provides flexible and expressive features that help you to traverse these connections and analyze the impact of change. The prime features include the Traceability Window, the Relationship Matrix and Relationship view.

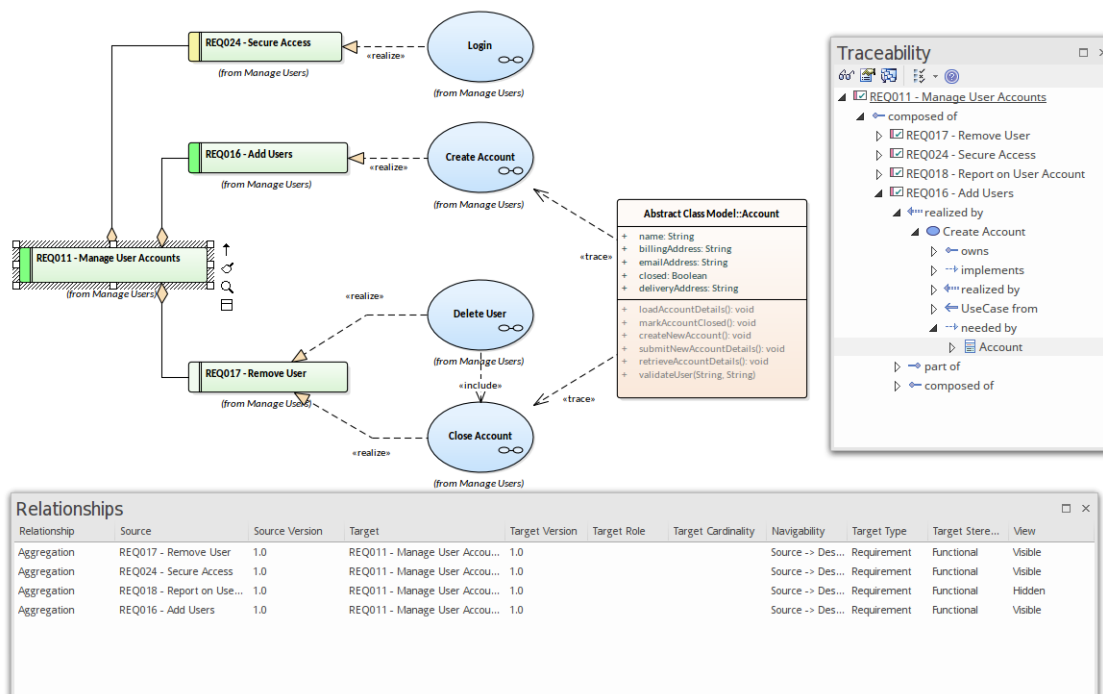


Figure 31: Using the Traceability view to track the impact of changes to a Requirement Element.

Model Validation

Model Validation is supported for base languages like UML, UPDM and Zachman framework. There is also

comprehensive support for defining your own model validation rules when working with a user-defined language or framework.

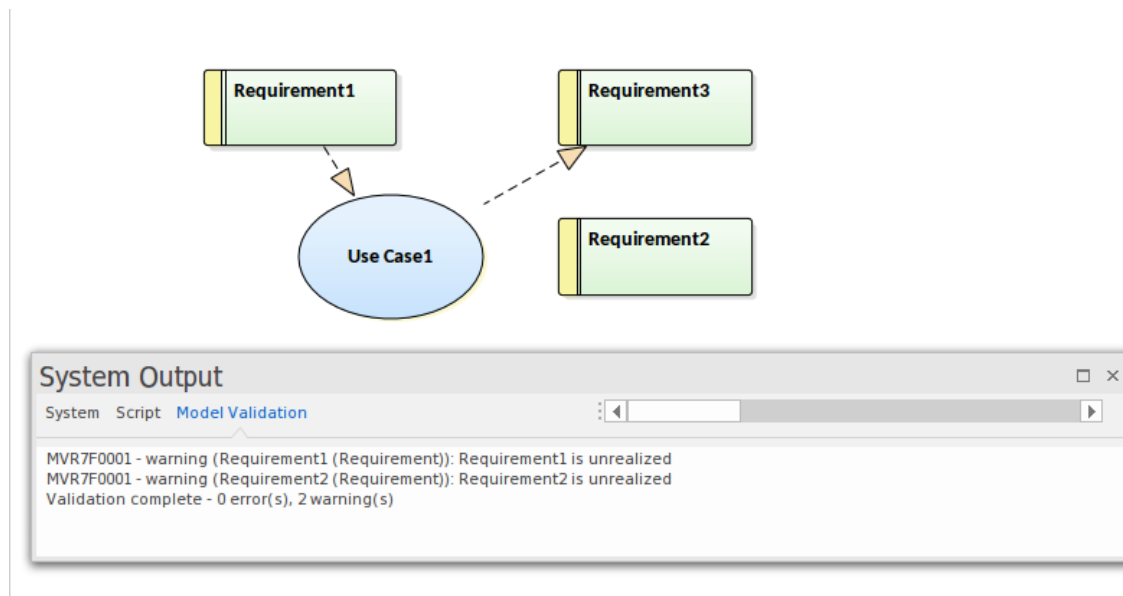


Figure 32: Model validation prompting where Requirement's are not connected or wrongly connected.

Model Testing on Conformance

Whether modeling requirements or a business process on a high level, down to developing code for an application, each step on the different levels of the process can be subject to testing to check whether there is conformance to the goals and the guidelines defined in the model. Enterprise Architect provides a comprehensive Test Management environment. This includes Test Case logging on the element level for design testing, as well as Testpoint and Unit testing on the coding level.

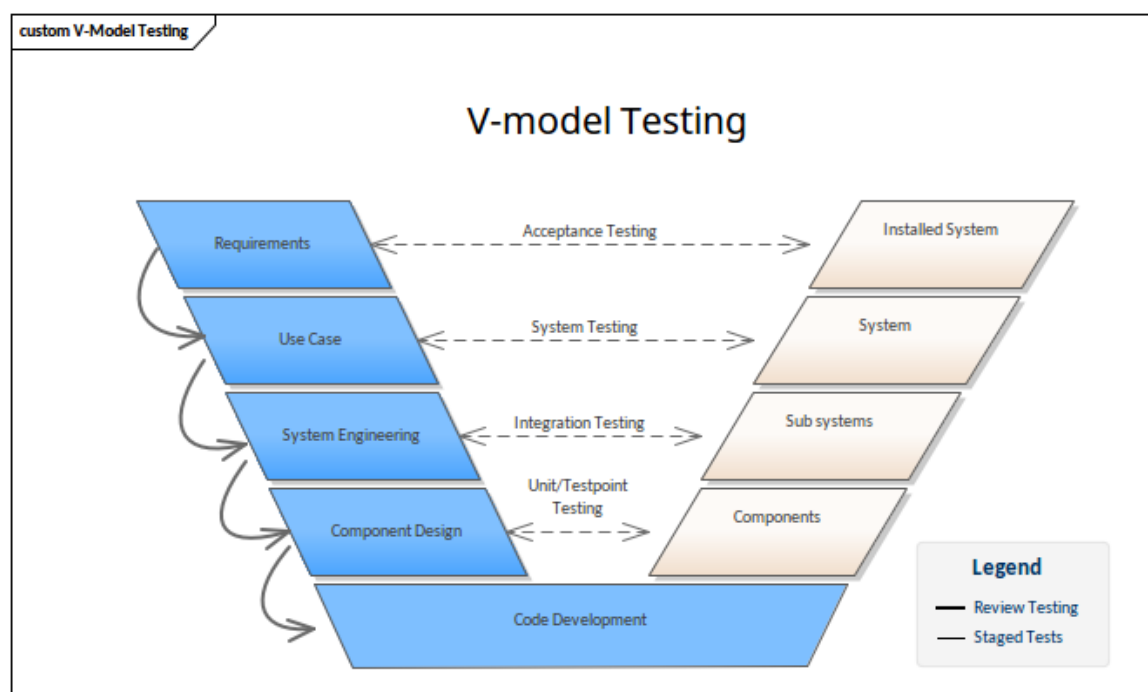


Figure 33: A diagram depicting the V-model Testing process that can be implemented using Element TestCases.

Test Cases

In a tightly controlled system, where the risk of application failure is high or the consequences can be catastrophic, an absolutely necessary modeling feature, for checking compliance, is the ability to set test cases on each level of the model. These test cases are used for verifying each layer of the model's compliance with the higher level modeling, all the way up to the source specifications. Verification of these test cases ensures that there is no error creeping from any one level of modeling down to the next.

This compliance can be tightly checked by linking Test Cases, containing a set of tests, to elements on each level of the design.

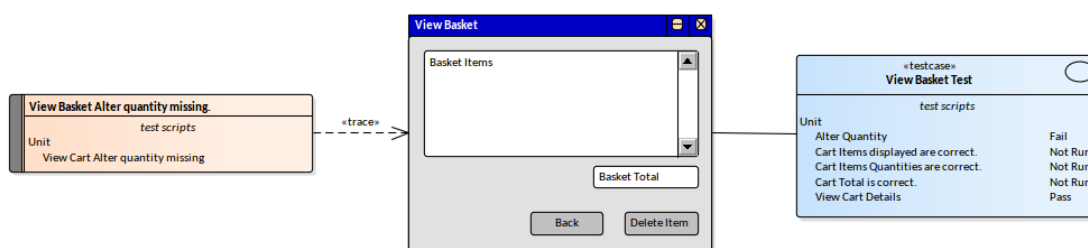


Figure 34: Test Case connectivity between an Issue element, a wireframe and a TestCase containing tests.

Figure 34 shows the clear logging of Test Case results:

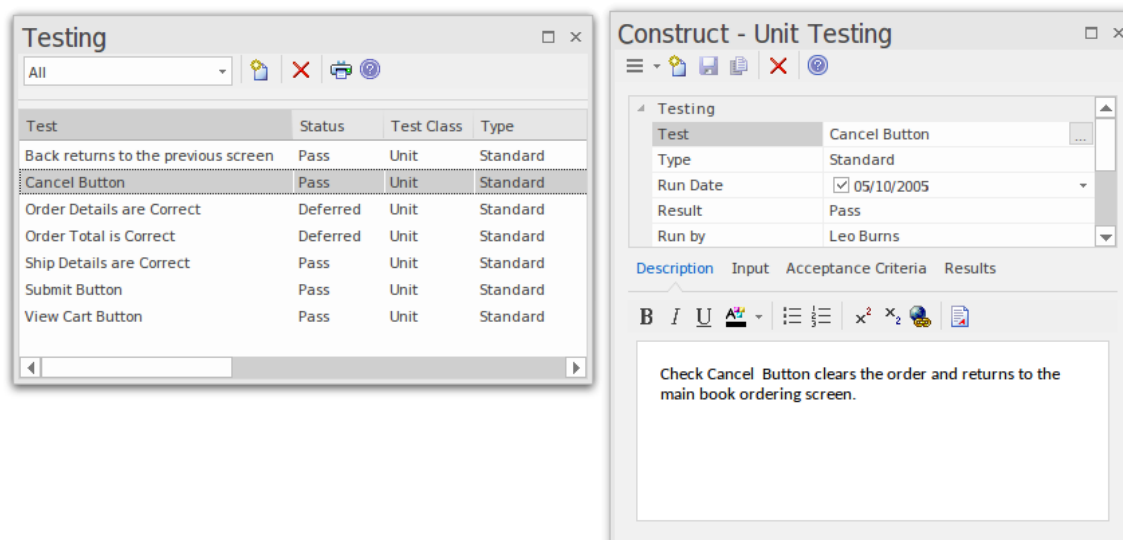


Figure 35: The Testing dialog showing a set of tests and the details of the one selected in the Construct view.

Task Management of Testing

Tools for project managing testing of the model include Gantt charts and Kanban diagrams.

Gantt charts provide a specific time related view of the progress of the items logged for testing.

Gantt Charts

A Gantt chart of a listing of Elements:

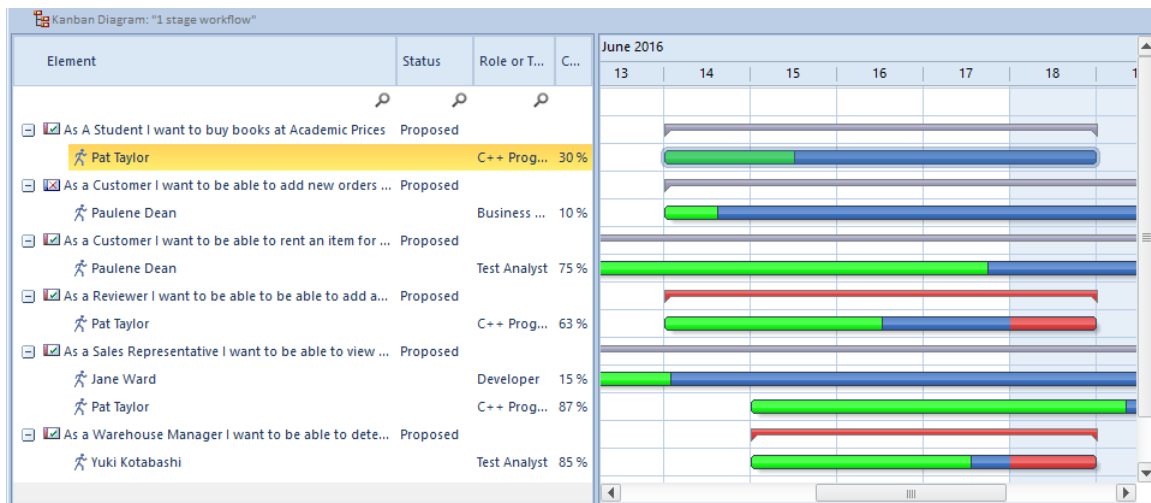


Figure 36: A Gantt chart, listed by Element, overseeing the assignment of test cases and showing the percentage of completion.

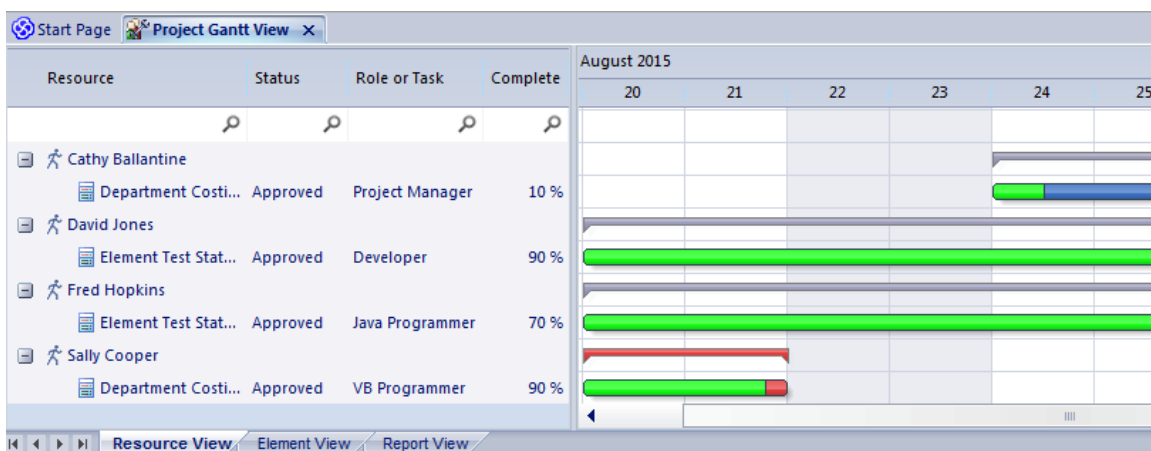


Figure 37: A Gantt chart, listed by users, showing Test Case assignments and the percentage of completion.

Kanban diagrams

An example of logging Test Cases using a Kanban diagram.



Figure 38: A Kanban diagram showing Test Case tasks mapped by their status.

Reviewing Compliance to Standards

The tools for reviewing compliance include setting up model reviews, setting profiles to log the review process and using Workflow script to tightly oversee the process for performing a review.

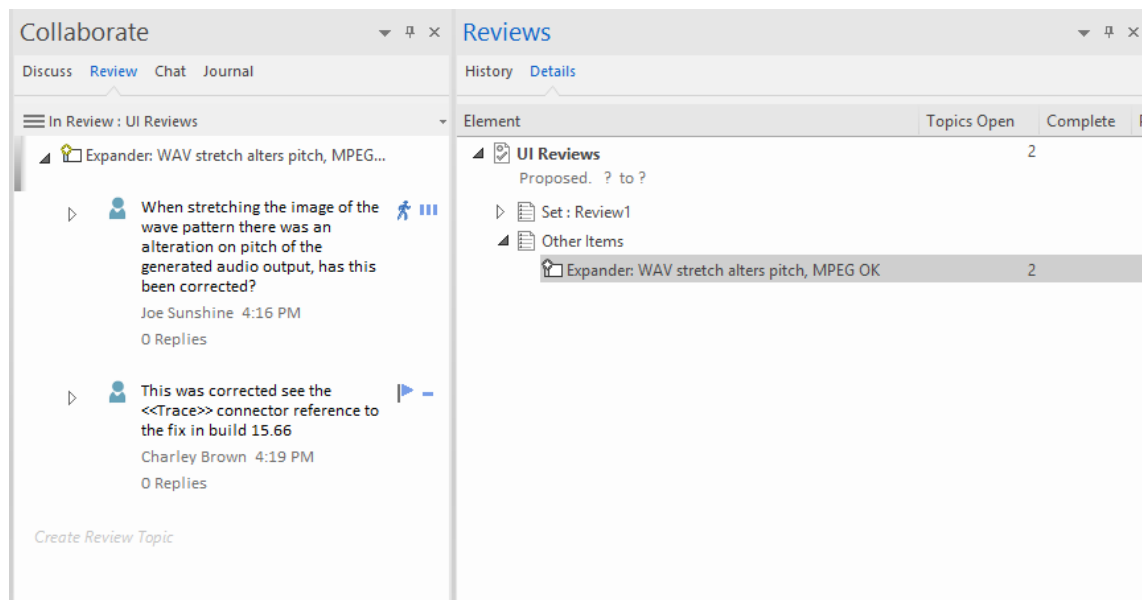


Figure 39: An example of a dialog on an Element reviewing the compliance to a data format (WAV).

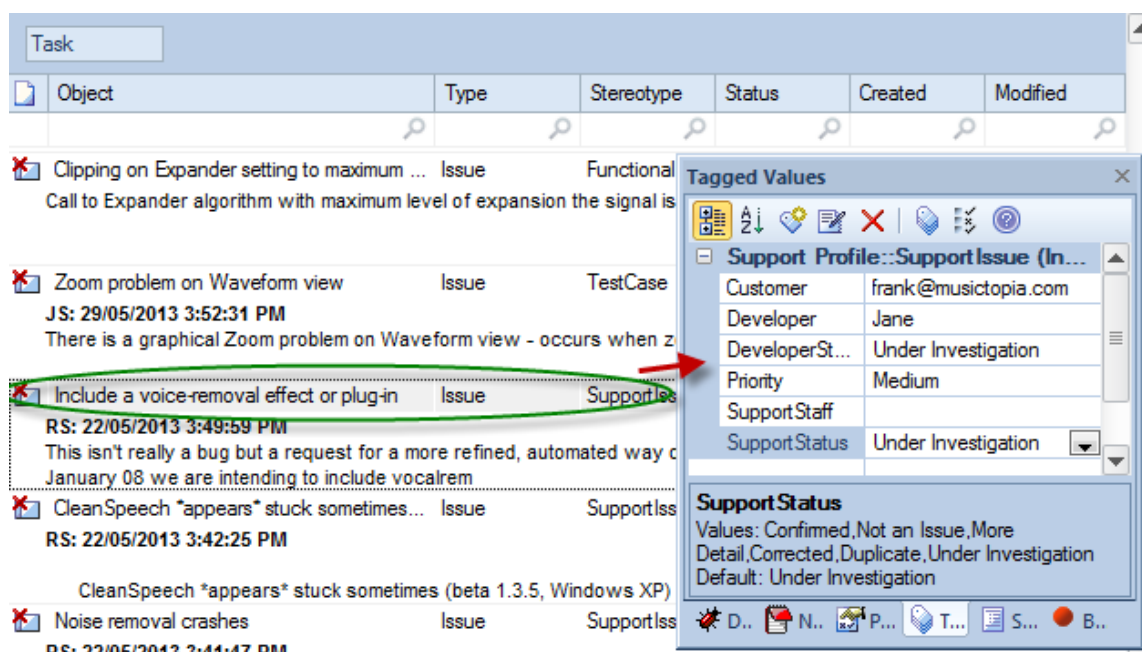


Figure 40: A example of setting a workflow process for test based workflow.

Change and Issue Tracking

For tight control over change and issue tracking, Enterprise Architect supports the linking of Issue Elements to the components that are causing an issue, as well as linking of Change Elements to components that will be affected by a proposed change request, providing clear traceability for Change and Issue management.

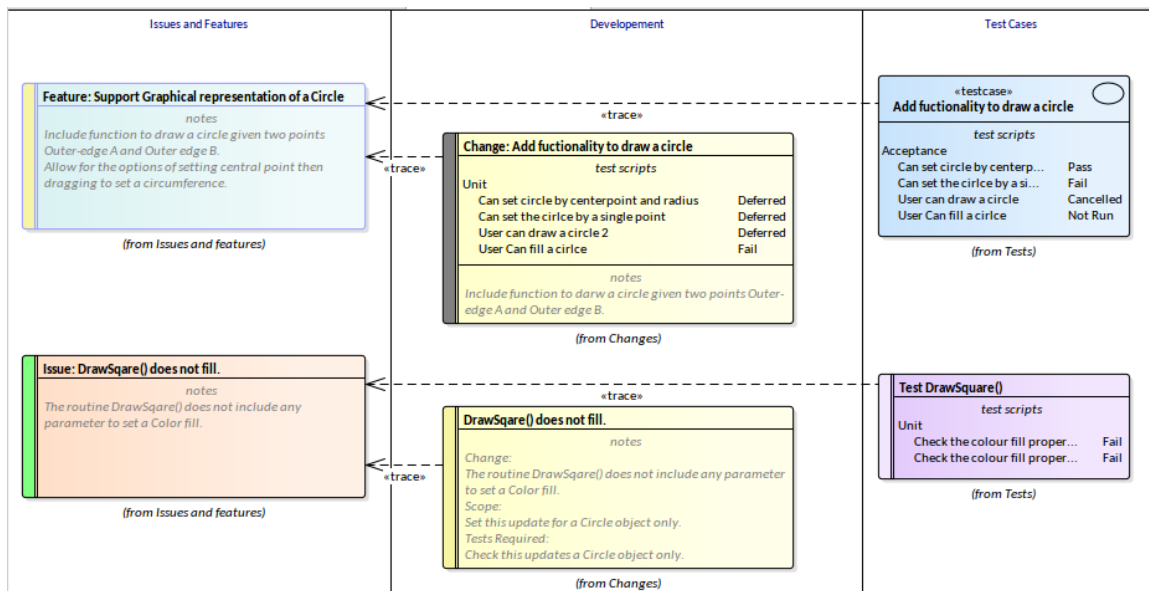


Figure 41: Typical model connectivity between an Issue Element, Change Elements and a Test Case.

WebEA

WebEA provides a real-time view of content contained in an Enterprise Architect Cloud repository allowing you to enable a small group or your entire organization to read and consume models, and provide feedback to enforce the compliance guidelines defined in the governance documentation of the project.



Figure 42: Using WebEA as a real-time view of the model for C-suite executives.

This can be highly productive for teams that are not directly involved in the modeling, but are:

- Checking compliance
- Adding issues (support staff)
- Testing changes which are required to be logged into to the model.

Plug-in capabilities for tight restrictions

Where very tight restrictions on model change are required there is the option to use simple Add-ins to restrict change on the model, Enterprise Architect supports comprehensive plug-in capabilities that can be used to approve and deny model changes.

Conclusion

Using the core Enterprise Architect features for modeling and implementing Governance, Risk and Compliance is a highly effective and powerful means of ensuring that strategic direction is correctly implemented and tied to the fundamental infrastructure and direction of an enterprise. This provides an effective means of ensuring that your policies and procedures are defined in a single model, and that the directives are mapped to related design and implementation, with tight linking for verifying that downstream actions conform to these guidelines, in all project phases, from the design and development through to the end of the lifecycle of the project.

