

Non-Functional Requirements Analysis

The *Non-Functional Requirements Analysis* pattern creates a series of packages, elements and a diagram that model non-functional requirements. The requirements are grouped based on the type of requirement including groups such as Availability, Comparability, Extensibility Scalability and more.

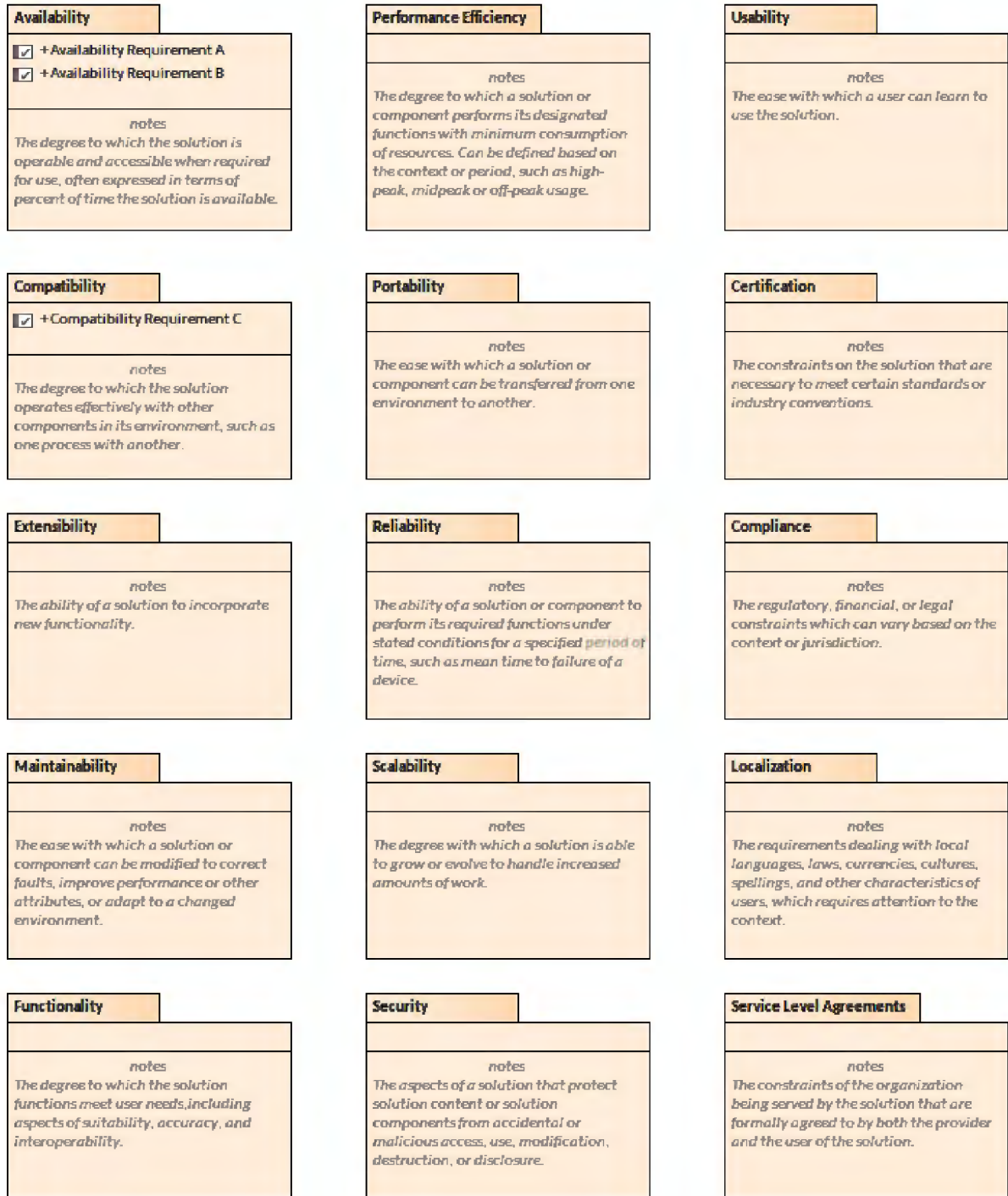


Figure 1. Shows a Package diagram with packages that group the Non-Functional Requirements by their type. The diagram properties have been configured to show the elements contained in the packages and their notes.

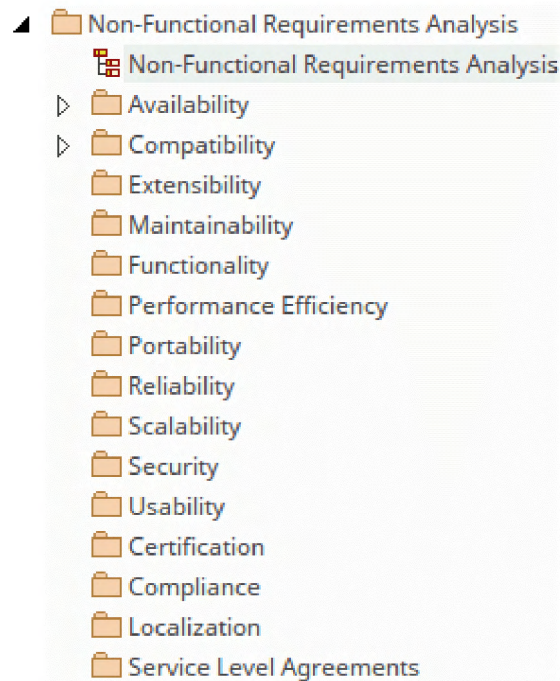


Figure 2. Shows the Project Browser with the Packages that group the Non-Functional Requirements.

Discussion

To provide a way of visualizing groups of Non-Functional Requirements in a single diagram or in the Project Browser. The predefined set of Packages helps to facilitate the identification of gaps in the Requirements Specification by allowing missing Requirements to be identified.

It is commonly used when a Requirements or Business Analyst wants to provide a visualization of the Non-Functional Requirements. Requirements that describe an entire system are typically created early on in a system description; the pattern can however be created at any time, particularly when the requirements describe a subsystem or a part of the system under focus.

The following is a list of some things you may want to do when working with this pattern.

- Rename the diagram.

- Rename the Requirements to suit the initiative.
- Add detailed notes that describe the business or system significance of the Requirement.
- Update the properties of the Requirements to suit the initiative.

The following is a list of some of the next steps available when applying the pattern.

- Define Trace relationships that indicate the relationship between Non-Functional and other requirements types.
- Define Trace relationships showing how the Requirements relate to up-process elements such as Strategies, Business Rules and other Requirements and down-process elements such as User Stories, Use Cases, Components, Artifacts and database tables.
- Create high quality documentation generated automatically from the model.
- Create Discussions and Reviews and engage in Chat to collaborate with team members, Requirement owners, Product Managers and other stakeholders.

Reference

The following help topics will assist you learn about how to work with this pattern.

[Non-Functional Requirements Analysis](#)

[Specification View](#)

[Business Analysis Body of Knowledge \(BABOK\)](#)

[Traceability Tools](#)

[Documentation](#)

The following are some of the tools that will be helpful when working with this pattern.

[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 and for analysts who are accustomed to working with requirements in a text based tool it will provide a

welcomed and compelling graphical representation of the requirements. For more details see the [Requirements Diagram](#) help topic.

Specification View

The Specification View can be used as a way of working with any element type in a spreadsheet or word process view. It is particularly useful when there are a large number of elements as is typically the case when describing a system of any appreciable size. For more details see the [Specification View](#) help topic.

Import and Export Spreadsheets

It is common for analyst to have started their modeling of Requirements including Non Functional Requirements 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 Requirements including Non Functional Requirements. Names, Description, 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 and each element is specified in a row. For more details see the [Import and Export Spreadsheet](#) help topic.

Document Generator

The Document Generator is a powerful facility in Enterprise Architect that allows a Database Engineer or other stakeholder to create high quality corporate or technical documentation directly from the model, suitable for internal or external audiences. For more details see the [Documentation](#) help topic or the more general topic on [Model Publishing](#).

Element Discussions

The Element Discussion facility is a fully featured collaboration tool allowing modelers and model viewers and reviewers to communicate with each other directly inside the repository. Modelers using the full client or occasional viewers using WebEA can both post and reply to discussions and communicate and engage in chat. For more details see the [Element Discussions](#) help topic.

Relationship Matrix

The Relationship Matrix provides a spreadsheet like view of two groups of elements and the relationships that exist between them. It can be used as a powerful analysis mechanism to visually indicate how elements are related to each other and to discover which elements are missing relationships. For more details see the [Relationship Matrix](#) help topic.

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. For more details see the [Traceability Window](#) help topic.