



Implementing the Rational Unified Process with S-Miner

# Vision Definition Baseline Plan





<b>INTRODUCTION</b>	<b>3</b>
REQUIREMENTS MANAGEMENT DISCIPLINE	3
S-MINER	3
RUP CONTEXT	3
REQUIREMENTS MANAGEMENT OVERVIEW	4
<b>DEVELOPING THE VISION</b>	<b>6</b>
INTRODUCTION	6
PROCESS	6
ERS ELICITATION	6
VISION DEFINITION	7
S-MINER SUPPORT FOR VISION DEFINITION	8
DERIVING S-MINER® VISION DEFINITION META-MODEL FOR YOUR OWN QUALITY PROGRAM	9
<b>CONTINUING TO NEXT PHASES</b>	<b>10</b>



## Introduction

This document focuses on requirements engineering in the early phase of a project which is commonly referred as “Definition of the vision” in the RUP (Rational Unified Process) methodology.

It documents a “generic foundation to vision definition” implemented as a S-Miner’s meta-model which can be used “as it is” to quickly initiate a RUP or which can serve as a customizable basis for specific instantiations in any company context. It is part of a more global “generic foundation to requirements management” which is also implemented as a S-Miner’s meta-model.

The overall “generic foundation to requirements management” is dedicated to “product’s engineering teams” or “IT systems architects” who have to identify, clarify, review, finalize and continually monitor the functions and features that make up “product/application requirements baselines” that they are maintaining.

### *Requirements Management Discipline*

Requirements management is a systematic approach to documenting, organizing and tracking changes to the requirements of a system.

A requirement is defined as a condition or capability to which the system must conform. Requirements management is a systematic approach to:

- Elicit, organize, and document the requirements of the system, and
- Establish and maintaining agreement between the customer and the project team on the changing requirements of the system.

“Product engineering teams” / “IT systems architects” must use a formal, standard approach for managing all requirements. This approach must be part of an overall quality assurance program which purpose is to promote, develop and deploy best practices over design phases of computer systems.

### *S-Miner*

S-Miner has been developed from our wide experience as consultant in research & development and specification phases of various kind of systems in various domains with various customers’ methods, practices, documents layouts, and quality assurance programs.

Since its development started in 1999, S-Miner has gained a capacity to virtually integrate any kind of engineering methods in any kind of domain.

This outstanding customization capacity is brought by a meta-model mechanism which allows to teach the tool with your own quality assurance program (method, vocabulary, formalism, ...) so that it strictly implements your quality assurance program and optimizes its deployment and dissemination among your teams.

### *RUP Context*

The following has been defined to specifically support the application of the RUP methodology which is based on the following key best practices:

- **Creating and maintaining a clear statement of the requirements through a formal, well-defined process.** Requirements are identified from several different sources, both external and internal. Stakeholders’ requests are initially captured as enhancement requests (ERs). This document details the process by which ERs are collected, organized and reviewed before being further analyzed and refined into features and other requirement types that serve as the contract with stakeholders as to the contents of the release of a product/application.



- **Defining and managing by requirement types** – Traditionally, requirements are looked upon as statements of text that describe “a condition or capability to which the system must conform”. In order to perform effective requirements management, it is important to extend what is maintained as requirements beyond only the detailed “software requirements”. Therefore, the concept of **requirements types** allows to separate the different levels of abstraction and purposes of the requirements.
- **Creating and maintaining applicable attributes for each requirement type.** In order to manage requirements more effectively, specific attributes are defined for each requirement type. Attributes are data elements that provide additional information about an item. Attributes are used to relate the feature or requirements data to other types of project information. The attributes are used to track, prioritize, and manage the requirements. For example, the attribute, “unique identifier” will help track the requirement, the “priority” or “risk” level may help to identify key requirements that may need to be addressed in an early iteration or a requirement that may require more intensive testing.
- **Maintaining and updating a requirements repository.** Requirements are maintained in a centralized requirements repository and kept current during the project.
- **Creating and maintaining traceability to other requirements and other project artifacts.** Traceability is a mechanism for ensuring requirements coverage from project start-up to project completion. Tracking key artifacts to the requirements that they realize or verify ensures that agreed-upon objectives and functionality are met and integrated into the solution as the project progresses. Furthermore, traceability provides with the ability to assess more accurately the impact of a change or enhancement request since the change can be traced to all impacted project artifacts.
- **Conducting coverage reviews during the project to verify that all requirements are being addressed in the solution.** These reviews identify and analyze requirements that are missing supporting artifact references in the requirements coverage reports.
- **Controlling changes to the system requirements.**

## ***Requirements Management Overview***

The requirements discipline revolves around understanding the **needs** of product **stakeholders**, analyzing those requirements, and organizing them in such a manner that they may be best understood and used by follow-on activities. A clear understanding and agreement on requirements is essential for the development of any product/application.

Throughout the lifecycle of a product/application, **enhancement requests** (ERs) are submitted by **stakeholders**. At pre-defined intervals, ERs are approved, rejected or delayed.

During the inception phase of a project, approved ERs are distilled into new product **features** or **enhancements to existing features**.

The selected features must be precisely identified / described and then serve as a basis for an impact analysis in order to understand the affect of the features on the existing product. A collection of features that are realizable within a given time frame is accumulated, reviewed, and approved as part of a **Vision document** for the project.



There are several different types of requirements. **Features** may be functional or non-functional in nature. Functional features will always be traced from one or more **use cases**. A use case represents the interaction between the system and any number of **actors**. Non-functional features will always be traced from one or more **supplementary specifications**. A use case is always composed of a series of flows, including a **basic flow** plus any number of **alternate flows**. The combination of a series of flows that results in a complete path through a use case is a **use case scenario**. Finer-grained requirements that specify the behavior of the system may be captured as **business rules** that trace to specific use case flows. Also any requirements pertaining to state of the system may be captured as **information needs**.

All of the above types of requirements must be managed in a requirements repository, and changes to them must be strictly controlled. All requirements types have an associated state and follow the same state transitions that occur at pre-defined points in the requirements management process. Once a requirement has been approved, it can only be changed as a result of an approved and scheduled ER. The project team uses requirements traceability to ensure that all associated project artifacts are modified to include the approved change to the requirement.

Periodically audits of the requirements data are conducted. Audits include such things as traceability and coverage reviews and reviews of the revision history of approved requirements to ensure that no changes have been made without approval.



## Developing the vision

### Introduction

In the inception phase for a project, a collection of **stakeholders'** requests in the form of **Enhancement Requests (ERs)** and their originating **needs** is considered for a **product release**.

ERs to be considered together with the associated new or impacted features will then serve to feed the Vision document for the release. This document will have to be agreed upon by all the stakeholders.

The Vision defines the stakeholders' view of the product to be developed, specified in terms of the stakeholders' key needs and features. Containing the proposed features for a product release, it serves as a **contract between the project team and the stakeholders**. It will be the basis for all subsequent work for the release.

The Vision sets the scope for a release. A completed Vision is a pre-requisite for exiting the inception phase and entering the elaboration phase for a project. **A project without a vision is a project with no direction, and as such is an unacceptable proposition.**

### Process

#### ERs elicitation

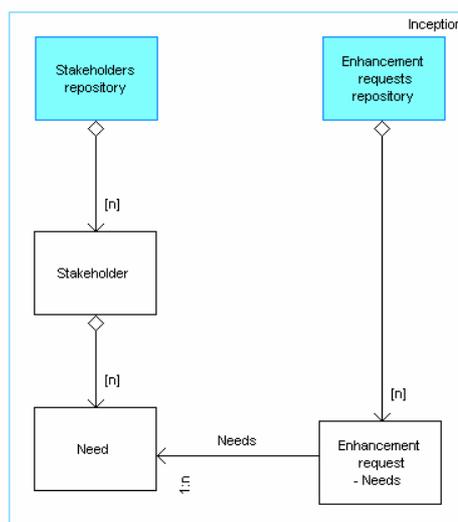
This first part of the process involves Business Analyst teams and Stakeholders.

On the basis of an identification of each **stakeholder** (S-Miner®) and of the **product(s)** (S-Miner®) or **application(s)** (S-Miner®)

- ↳ Stakeholders' **Needs** (new & evolution) are identified and their description is captured in a centralized repository (S-Miner®)

On the basis of a stakeholders' needs selection, analysis and consolidation

- ↳ **Enhancement Request** are then defined and their description is captured in a centralized repository (S-Miner®).





Providing this first set of vision elements, a first **documentation set is automatically produced** (S-Miner®). It contains :

- the pre-vision document describing stakeholders' needs and the proposed enhancement requests
- several matrixes to support coverage & impact analysis
  - Stakeholder / ER of interest
  - ER / Originating Needs

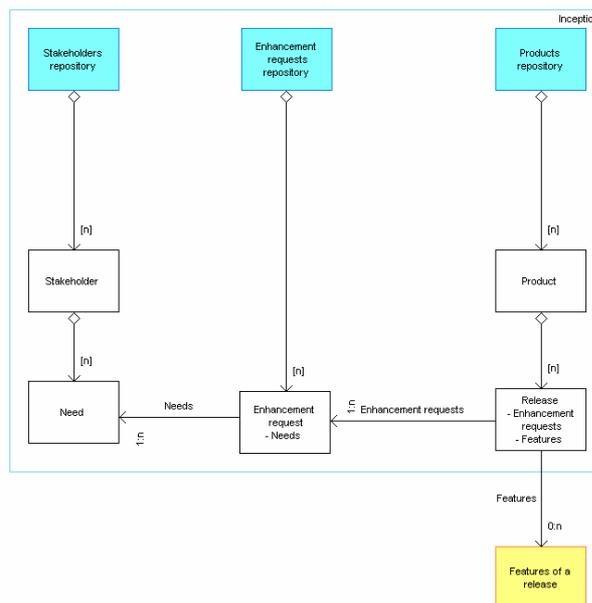
This first documentation set must then be validated by the stakeholders group before going on to next step.

### Vision definition

Given a set of ERs for a release, System Analyst teams distill a set of features that need to be included/modified in the product(s) / application(s) to satisfy the ERs. This process requires collaboration with the stakeholders to agree on the features to be implemented.

As features are identified and described, the System Analysts capture them in a **centralized repository** (S-Miner®). Each feature (new & evolution of existing ones) traces back to the ER from which it is derived.

In addition to capturing the features, the System Analysts builds the vision document by **extracting relevant information** (S-Miner®) from the **centralized repository** (S-Miner®) and by adding **additional prose to conform to a vision document template** (S-Miner®) issued from the global quality assurance program.



Providing this second set of vision elements, **the Vision Documentation set is automatically produced** (S-Miner®). It contains :

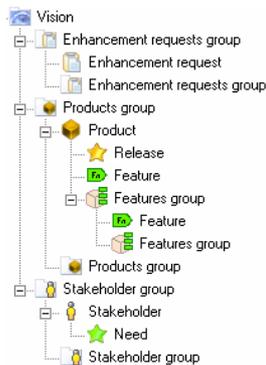
- the Vision document describing the features to be developed for each product / application impacted by the release. It also provides the full set of originating stakeholders' needs and enhancement requests in order to provide detailed description of the contract between the teams involved in next phases with stakeholders.



- several matrixes to support coverage & impact analysis
  - Stakeholder / ER of interest
  - ER / Originating Needs
  - Stakeholder / Features of interest
  - Stakeholder / Products of interest
  - Features / originating ERs
  - Features / originating Needs
  - Products / originating ERs
  - Products / originating Needs

This documentation set must then be validated by the stakeholders group before going on to next phase.

### ***S-Miner support for Vision Definition***



S-Miner® meta-model for Vision Definition provides the necessary concepts for managing a single centralized repository for :

- Stakeholders identification & description
- Stakeholders' needs identification & description
- Enhancement requests identification & description
- Products / Applications identification & description
- Product's features identification & description
- Product's release identification & description

It also provides all the necessary features to capture traceability and information's relationships :

- ERs / Stakeholders' Needs
- Features / ERs
- Release / ERs
- ...

For each kind of information managed, S-Miner® automatically provides :

- lifecycle support (draft, submitted, validated, ...),
- version management,
- changes monitoring,
- repository comparison and synchronization,
- ...

S-Miner® also provides advanced publishing and document modeling features. It allows to build any kind of documents by aggregating information extracted from the repository and external sources within your custom templates to fit your company's look'n'feel.



## ***Deriving S-Miner® Vision Definition meta-model for your own quality program***

S-Miner® Vision Definition meta-model can easily be customized to add / change concepts, attributes, ...

It can both provide a ready to use solution for implementing RUP process, or, a basis to be improved in order to take your own quality assurance program specificities.



## Continuing to next phases

S-Miner® Vision Definition meta-model integrates into a more global RUP meta-model which supports the whole RUP process.

When used with the full RUP meta-model it is extended with:

- Actors identification & description
- Use Cases identification & description
- Detailed Use cases definition
- Business Rules identification, description & allocation
- ...