

<Project Name> Vision

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
2. Positioning
 - 2.1 Business Opportunity
 - 2.2 Problem Statement
 - 2.3 Product Position Statement
3. Stakeholder and User Descriptions
 - 3.1 Market Demographics
 - 3.2 Stakeholder Summary
 - 3.3 User Summary
 - 3.4 User environment
 - 3.5 Stakeholder Profiles
 - 3.5.1 <Stakeholder Name>
 - 3.6 User Profiles
 - 3.6.1 <User Name>
 - 3.7 Key Stakeholder or User Needs
 - 3.8 Alternatives and Competition
 - 3.8.1 <aCompetitor>
 - 3.8.2 <anotherCompetitor>
4. Product Overview
 - 4.1 Product Perspective
 - 4.2 Summary of Capabilities
 - 4.3 Assumptions and Dependencies
 - 4.4 Cost and Pricing
 - 4.5 Licensing and Installation
5. Product Features
 - 5.1 <aFeature>
 - 5.2 <anotherFeature>
6. Constraints
7. Quality Ranges
8. Precedence and Priority
9. Other Product Requirements
 - 9.1 Applicable Standards
 - 9.2 System Requirements
 - 9.3 Performance Requirements
 - 9.4 Environmental Requirements
10. Documentation Requirements
 - 10.1 User Manual
 - 10.2 Online Help
 - 10.3 Installation Guides, Configuration, and Read Me File
 - 10.4 Labeling and Packaging

A. Feature Attributes

Vision

1. Introduction

*[The purpose of this document is to collect, analyze, and define high-level needs and features of the <<System Name>>. It focuses on the capabilities needed by the stakeholders, and the target users, and **why** these needs exist. The details of how the <<System Name>> fulfils these needs are detailed in the use-case and supplementary specifications.]*

*[The introduction of the **Vision** document provides an overview of the entire document. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **Vision** document.]*

1.1 Purpose

*[Specify the purpose of this **Vision** document.]*

1.2 Scope

*[A brief description of the scope of this **Vision** document; what Project(s) it is associated with and anything else that is affected or influenced by this document.]*

1.3 Definitions, Acronyms, and Abbreviations

*[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Vision** document. This information may be provided by reference to the project's Glossary.]*

1.4 References

*[This subsection provides a complete list of all documents referenced elsewhere in the **Vision** document. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

1.5 Overview

*[This subsection describes what the rest of the **Vision** document contains and explains how the document is organized.]*

2. Positioning

2.1 Business Opportunity

[Briefly describe the business opportunity being met by this project.]

2.2 Problem Statement

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

The problem of	<i>[describe the problem]</i>
affects	<i>[the stakeholders affected by the problem]</i>
the impact of which is	<i>[what is the impact of the problem?]</i>
a successful solution would be	<i>[list some key benefits of a successful solution]</i>

2.3 Product Position Statement

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

For	<i>[target customer]</i>
Who	<i>[statement of the need or opportunity]</i>
The (product name)	<i>is a [product category]</i>
That	<i>[statement of key benefit; that is, the compelling reason to buy]</i>
Unlike	<i>[primary competitive alternative]</i>
Our product	<i>[statement of primary differentiation]</i>

[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

3. Stakeholder and User Descriptions

[To effectively provide products and services that meet your stakeholders' and users' real needs, it is necessary to identify and involve all of the stakeholders as part of the Requirements Modeling process. You must also identify the users of the system and ensure that the stakeholder community adequately represents them. This section provides a profile of the stakeholders and users involved in the project, and the key problems that they perceive to be addressed by the proposed solution. It does not describe their specific requests or requirements as these are captured in a separate stakeholder requests artifact. Instead, it provides the background and justification for why the requirements are needed.]

3.1 Market Demographics

[Summarize the key market demographics that motivate your product decisions. Describe and position target market segments. Estimate the market's size and growth by using the number of potential users, or the amount of money your customers spend trying to meet needs that your product or enhancement would fulfill. Review major industry trends and technologies. Answer these strategic questions:

- *What is your organization's reputation in these markets?*
- *What would you like it to be?*
- *How does this product or service support your goals?]*

3.2 Stakeholder Summary

[There are a number of stakeholders with an interest in the development and not all of them are end users. Present a summary list of these non-user stakeholders. (The users are summarized in section 3.3.)]

Name	Description	Responsibilities
<i>[Name the stakeholder type.]</i>	<i>[Briefly describe the stakeholder.]</i>	<i>[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder:</i> <ul style="list-style-type: none"> - <i>ensures that the system will be maintainable</i> - <i>ensures that there will be a market demand for the product's features</i> - <i>monitors the project's progress</i> - <i>approves funding</i> - <i>and so forth]</i>

3.3 User Summary

[Present a summary list of all identified users.]

Name	Description	Responsibilities	Stakeholder
<i>[Name the user type.]</i>	<i>[Briefly describe what they represent with respect to the system.]</i>	<i>[List the user's key responsibilities with regard to the system being developed; for example:</i> <ul style="list-style-type: none"> - <i>captures details</i> - <i>produces reports</i> - <i>coordinates work</i> - <i>and so on]</i> 	<i>[If the user is not directly represented, identify which stakeholder is responsible for representing the user's interests.]</i>

3.4 User Environment

[Detail the working environment of the target user. Here are some suggestions:

- *Number of people involved in completing the task? Is this changing?*
- *How long is a task cycle? Amount of time spent in each activity? Is this changing?*
- *Any unique environmental constraints: mobile, outdoors, in-flight, and so on.?*
- *Which systems platforms are in use today? Future platforms?*
- *What other applications are in use? Does your application need to integrate with them?*

This is where extracts from the Business Model could be included to outline the task and business workers involved, and so on.]

3.5 Stakeholder Profiles

[Describe each stakeholder in the system here by filling in the following table for each stakeholder. Remember that stakeholder types can be as divergent as users, departments, and technical developers. A thorough profile would cover the following topics for each type of stakeholder.]

3.5.1 <Stakeholder Name>

Representative	<i>[Who is the stakeholder representative to the project? (Optional if documented elsewhere.) What we want here is names.]</i>
Description	<i>[Brief description of the stakeholder type.]</i>
Type	<i>[Qualify the stakeholder's expertise, technical background, and degree of sophistication—that is, guru, business, expert, casual user, and so on.]</i>
Responsibilities	<i>[List the stakeholder's key responsibilities with regards to the system being developed—that is, their interest as a stakeholder.]</i>
Success Criteria	<i>[How does the stakeholder define success? How is the stakeholder rewarded?]</i>
Involvement	<i>[How is the stakeholder involved in the project? Relate where possible to Rational Unified Process roles—that is, Requirements Reviewer, and so on.]</i>
Deliverables	<i>[Are there any additional deliverables required by the stakeholder? These could be project deliverables or outputs from the system under development.]</i>
Comments / Issues	<i>[Problems that interfere with success and any other relevant information go here.]</i>

3.6 User Profiles

[Describe each unique user of the system here by filling in the following table for each user type. Remember user types can be as divergent as gurus and novices. For example, a guru might need a sophisticated, flexible tool with cross-platform support, while a novice might need a tool that is easy to use and user-friendly. A thorough profile covers the following topics for each type of user.]

3.6.1 <User Name>

Representative	<i>[Who is the user representative to the project? (Optional if documented elsewhere.) This often refers to the Stakeholder that represents the set of users, for example, Stakeholder1.]</i>
Description	<i>[A brief description of the user type.]</i>
Type	<i>[Qualify the user's expertise, technical background, and degree of sophistication—that is, guru, casual user, and so on.]</i>
Responsibilities	<i>[List the user's key responsibilities with regard to the system being developed—that is, captures details, produces reports, coordinates work, and so on.]</i>
Success Criteria	<i>[How does the user define success? How is the user rewarded?]</i>
Involvement	<i>[How the user is involved in the project? Relate where possible to Rational Unified Process roles—that is, Requirements Reviewer, and the like.]</i>

Deliverables	<i>[Are there any deliverables the user produces and, if so, for whom?]</i>
Comments / Issues	<i>[Problems that interfere with success and any other relevant information go here. These would include trends that make the user's job easier or harder.]</i>

3.7 Key Stakeholder or User Needs

[List the key problems with existing solutions as perceived by the stakeholder. Clarify the following issues for each problem:

- *What are the reasons for this problem?*
- *How is it solved now?*
- *What solutions does the stakeholder or user want?]*

*[It is important to understand the **relative** importance the stakeholder or user places on solving each problem. Ranking and cumulative voting techniques indicate problems that **must** be solved versus issues they would like addressed.*

Fill in the following table—if using Rational RequisitePro to capture the Needs, this could be an extract or report from that tool.]

Need	Priority	Concerns	Current Solution	Proposed Solutions
Broadcast messages				

3.8 Alternatives and Competition

[Identify alternatives the stakeholder perceives as available. These can include buying a competitor's product, building a homegrown solution or simply maintaining the status quo. List any known competitive choices that exist or may become available. Include the major strengths and weaknesses of each competitor as perceived by the stakeholder or end user.]

3.8.1 <aCompetitor>

3.8.2 <anotherCompetitor>

4. Product Overview

[This section provides a high level view of the product capabilities, interfaces to other applications, and systems configurations. This section usually consists of three subsections, as follows:

- *Product perspective*
- *Product functions*
- *Assumptions and dependencies]*

4.1 Product Perspective

*[This subsection of the **Vision** document puts the product in perspective to other related products and the user's environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection relates how these systems interact and identifies the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]*

4.2 Summary of Capabilities

*[Summarize the major benefits and features the product will provide. For example, a **Vision** document for a customer support system may use this part to address problem documentation, routing, and status reporting without mentioning the amount of detail each of these functions requires.*

Organize the functions so the list is understandable to the customer or to anyone else reading the document for the first time. A simple table listing the key benefits and their supporting features might suffice. For example:]

Table 4-1 Customer Support System

Customer Benefit	Supporting Features
New support staff can quickly get up to speed.	Knowledge base assists support personnel in quickly identifying known fixes and workarounds.

Customer satisfaction is improved because nothing falls through the cracks.	Problems are uniquely itemized, classified and tracked throughout the resolution process. Automatic notification occurs for any aging issues.
Management can identify problem areas and gauge staff workload.	Trend and distribution reports allow high level review of problem status.
Distributed support teams can work together to solve problems.	Replication server allows current database information to be shared across the enterprise.
Customers can help themselves, lowering support costs and improving response time.	Knowledge base can be made available over the Internet. Includes hypertext search capabilities and graphical query engine.

4.3 Assumptions and Dependencies

*[List each of the factors that affects the features stated in the **Vision** document. List assumptions that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.]*

4.4 Cost and Pricing

[For products sold to external customers and for many in- house applications, cost and pricing issues can directly impact the application's definition and implementation. In this section, record any cost and pricing constraints that are relevant. For example, distribution costs (# of diskettes, # of CD-ROMs, CD mastering) or other cost of goods sold constraints (manuals, packaging) may be material to the projects success, or irrelevant, depending on the nature of the application.]

4.5 Licensing and Installation

[Licensing and installation issues can also directly impact the development effort. For example, the need to support serializing, password security or network licensing will create additional requirements of the system that must be considered in the development effort.

Installation requirements may also affect coding or create the need for separate installation software.]

5. Product Features

[List and briefly describe the product features. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users. Each feature is an externally desired service that typically requires a series of inputs to achieve the desired result. For example, a feature of a problem tracking system might be the ability to provide trending reports. As the use-case model takes shape, update the description to refer to the use cases.

*Because the **Vision** document is reviewed by a wide variety of involved personnel, the level of detail needs to be general enough for everyone to understand. However, enough detail must be available to provide the team with the information they need to create a use-case model.*

To effectively manage application complexity, we recommend for any new system, or an increment to an existing system, capabilities are abstracted to a high enough level so 25-99 features result. These features provide the fundamental basis for product definition, scope management, and project management. Each feature will be expanded in greater detail in the use-case model.

Throughout this section, each feature will be externally perceivable by users, operators or other external systems. These features should include a description of functionality and any relevant usability issues that must be addressed. The following guidelines apply:

- *Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented.*
- *If you are using the Rational RequisitePro toolkit, all need to be selected as requirements of type for easy reference and tracking.]*

5.1 <aFeature>

5.2 <anotherFeature>

6. Constraints

[Note any design constraints, external constraints or other dependencies.]

7. Quality Ranges

[Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.]

8. Precedence and Priority

[Define the priority of the different system features.]

9. Other Product Requirements

[At a high-level, list applicable standards, hardware or platform requirements, performance requirements, and environmental requirements.]

9.1 Applicable Standards

[List all standards with which the product must comply. These can include legal and regulatory (FDA, UCC) communications standards (TCP/IP, ISDN), platform compliance standards (Windows, UNIX, and so on), and quality and safety standards (UL, ISO, CMM).]

9.2 System Requirements

[Define any system requirements necessary to support the application. These can include the supported host operating systems and network platforms, configurations, memory, peripherals, and companion software.]

9.3 Performance Requirements

[Use this section to detail performance requirements. Performance issues can include such items as user load factors, bandwidth or communication capacity, throughput, accuracy, and reliability or response times under a variety of loading conditions.]

9.4 Environmental Requirements

[Detail environmental requirements as needed. For hardware-based systems, environmental issues can include temperature, shock, humidity, radiation, and so forth. For software applications, environmental factors can include usage conditions, user environment, resource availability, maintenance issues, and error handling and recovery.]

10. Documentation Requirements

[This section describes the documentation that must be developed to support successful application deployment.]

10.1 User Manual

[Describe the purpose and contents of the User Manual. Discuss desired length, level of detail, need for index, glossary of terms, tutorial versus reference manual strategy, and so on. Formatting and printing constraints must be identified also.]

10.2 Online Help

[Many applications provide an online help system to assist the user. The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, and so on) with aspects of technical writing such as organization and presentation. Many have found the development of online help system is a project within a project that benefits from up-front scope management and planning activity.]

10.3 Installation Guides, Configuration, and Read Me File

[A document that includes installation instructions and configuration guidelines is important to a full solution offering. Also, a Read Me file is typically included as a standard component. The Read Me file can include a “What’s New With This Release” section, and a discussion of compatibility issues with earlier releases. Most users also appreciate documentation defining any known bugs and workarounds in the Read Me file.]

10.4 Labeling and Packaging

[Today’s state-of-the-art applications provide a consistent look and feel that begins with product packaging and manifests through installation menus, splash screens, help systems, GUI dialogs, and so on. This section defines the needs and types of labeling to be incorporated into the code. Examples include copyright and patent notices, corporate logos, standardized icons and other graphic elements, and so forth.]

A. Feature Attributes

[Features are given attributes that can be used to evaluate, track, prioritize, and manage the product items proposed for implementation. All requirement types and attributes are outlined in the Requirements Management Plan, however you may wish to list and briefly describe the attributes for features that have been chosen. The following subsections represent a set of suggested feature attributes.]

A.1 Status

[Set after negotiation and review by the project management team. Tracks progress during definition of the project baseline.]

Proposed	<i>[Used to describe features that are under discussion but have</i>
----------	--

	<i>not yet been reviewed and accepted by the "official channel," such as a working group consisting of representatives from the project team, product management, and user or customer community.]</i>
Approved	<i>[Capabilities that are deemed useful and feasible, and have been approved for implementation by the official channel.]</i>
Incorporated	<i>[Features incorporated into the product baseline at a specific point in time.]</i>

A.2 Benefit

[Set by Marketing, the product manager or the business analyst. All requirements are not created equal. Ranking requirements by their relative benefit to the end user opens a dialogue with customers, analysts, and members of the development team. Used in managing scope and determining development priority.]

Critical	<i>[Essential features. Failure to implement means the system will not meet customer needs. All critical features must be implemented in the release or the schedule will slip.]</i>
Important	<i>[Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of an important feature may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of any important feature.]</i>
Useful	<i>[Features that are useful in less typical applications will be used less frequently or for which reasonably efficient workarounds can be achieved. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release.]</i>

A.3 Effort

[Set by the development team. Because some features require more time and resources than others, estimating the number of team or person-weeks, lines of code required or function points, for example, is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.]

A.4 Risk

[Set by development team based on the probability the project will experience undesirable events, such as cost overruns, schedule delays or even cancellation. Most project managers find categorizing risks as high, medium, and low is sufficient, although finer gradations are possible. Risk can often be indirectly assessed by measuring the uncertainty (range) of the projects team's schedule estimate.]

A.5 Stability

[Set by analyst and development team based on the probability the feature will change or the team's understanding of the feature will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.]

A.6 Target Release

*[Records the intended product version in which the feature will first appear. This field can be used to allocate features from a **Vision** document into a particular baseline release. When combined with the status field, your team can propose, record, and discuss various features of the release without committing them to development. Only features whose Status is set to Incorporated and whose Target Release is defined will be implemented. When scope management occurs, the Target Release Version Number can be increased so the item will remain in the **Vision** document but will be scheduled for a later release.]*

A.7 Assigned To

[In many projects, features will be assigned to "feature teams" responsible for further elicitation, writing the software requirements, and implementation. This simple pull-down list will help everyone on the project team to understand responsibilities better.]

A.8 Reason

[This text field is used to track the source of the requested feature. Requirements exist for specific reasons. This field records an explanation or a reference to an explanation. For example, the reference might be to a page and line number of a product requirement specification or to a minute marker on a video of an important customer review.]