Company Name

Project XYZ Vision

Issue 0.1

Revision History

Date	Issue	Description	Author
990106	0.1	This is the first version of the Vision for the project XYZ.	Håkan Dyrhage

Table of Contents

Project XYZ Vision

1. Objectives

The purpose of this document is to collect, analyze and define high-level user needs and features of the product. Focus on capabilities needed by the target users and why these needs exist. Record details of how the application fulfills these needs in the use-case specifications.

2. Scope

A brief description of what the Vision document applies to; what is affected or influenced by this document.

3. References

This subsection should:

- Provide 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.

4. Positioning

4.1 Business Opportunity

Briefly describe the business opportunity being met by this project.

4.2 Problem Statement

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

The problem of	(describe the problem)
affects	(the stakeholders affected by the problem).
The impact of which is	(what is the impact of the problem).
A successful solution would	(list some key benefits of a successful solution).

4.3 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	(target customer)
Who	(statement of the need or opportunity)
The (product name)	is a (product category)
That	(statement of key benefit - that is - compelling reason to buy)
Unlike	(primary competitive alternative)
Our product	(statement of primary differentiation)

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

5. User Description

To effectively provide products and services that meet your customers' needs, it is necessary to understand the challenges they confront when performing their jobs. This section should profile the intended users of the application and the key problems that limit their productivity. **It should not be used to state specific requirements**. Instead, provide the background and justification for why the requirements are needed.

5.1 User/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/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?

5.2 User Profiles

Describe each unique user of the system here. 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 should cover the following topics for each type of user:

- Technical background and degree of sophistication
- Key responsibilities
- Deliverables the user produces, and for whom
- Trends that make the user's job easier or harder
- Problems that interfere with success
- How does the target user define success? How is the user rewarded?

5.3 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, etc.?
- Which systems platforms are in use today? Future platforms?

- What other applications are in use? Does your application need to integrate with them?

5.4 Key User Needs

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

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

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

5.5 Alternatives and Competition

Identify alternatives the user 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 end user.

5.5.1 aCompetitor

5.5.2 anotherCompetitor

6. 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

6.1 Product Perspective

This subsection of the Vision document should put 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 should relate how these systems interact and should identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections and external interfaces is via a block diagram.

6.2 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:

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

6.3 Assumptions and Dependencies

List each of the factors that affect 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.

6.4 Cost and Pricing

For products sold to external customers and for many in house applications, cost and pricing issues can directly impact the applications definition and implementation. In this section, record any cost and pricing constraints that are relevant. For example, distribution costs, (# of diskettes, # 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.

6.5 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.

7. Feature Attributes

Features should be given attributes that can be used to evaluate, track, prioritize and manage the product items proposed for implementation. List and briefly describes the attributes you have chosen. Following subsections represent a set of suggested feature attributes.

7.1 Status

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

Proposed	Used to describe features that are under discussion but have 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.
Approved	Capabilities that are deemed useful and feasible and have been approved for implementation by the official channel.
Incorporated	Features incorporated into the product baseline at a specific point in time.

7.2 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	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.
Important	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.
Useful	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.

7.3 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.

7.4 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 sufficient, although finer gradations are possible. Risk can often be assessed indirectly by measuring the uncertainty (range) of the projects teams schedule estimate.

7.5 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.

7.6 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.

7.7 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 better understand responsibilities.

7.8 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 interview.

8. 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 should be general enough for everyone to understand. However, enough detail should 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 should 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 Requisite toolkit, all should be selected as requirements of type for easy reference and tracking

8.1 aFeature

8.2 anotherFeature

9. Constraints

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

10. Quality Ranges

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

11. Precedence and Priority

Define the priority of the different system features.

12. Other Product Requirements

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

12.1 Applicable Standards

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

12.2 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.

12.3 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, reliability or response times under a variety of loading conditions.

12.4 Environmental Requirements

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

13. Documentation Requirements

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

13.1 User Manual

Describe the purpose and contents of the User Manual. Discuss desired length, level of detail, need for index, glossary of terms, tutorial vs. reference manual strategy, etc. Formatting and printing constraints should also be identified.

13.2 Online Help

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

13.3 Installation Guides, Configuration, 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 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.

13.4 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, etc. 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, etc.