The Product Vision (The Vision) Document

The Product Vision is a very valuable document that summarizes the **high-level requirements of the software**. This document serves as a **guide** for stakeholders, developers, and other decision-makers, to understand the **purpose** and **objectives** of the software.

The Vision document is usually created by the **project/product owner** during the project **inception or initiation phase** and is **later updated** when more requirements are elaborated. In **Plan-Driven models**, this document is referenced during **project planning** to ensure that the development plan aligns with the vision. In Agile models, this document is usually created during project initiation which is sometimes called Sprint/Iteration 0 in a less formalized way. This document is usually refined during the **Product Backlog refinement sessions**, and is adjusted during the **Retrospective meetings**. This document is usually used as a guide for sprint planning to assure that short-term goals align with the overall vision. During **release planning**, which occurs at regular intervals, the team revisits and refines the product vision. Adjustments are made based on feedback, changing market conditions, or new insights. Since Agile development embraces change, the vision is **continuously adapted** based on customer feedback, market changes, or new insights gained during development.

In both traditional and Agile development, the key is to view the vision document as a **living document** that **evolves** throughout the project's lifecycle. Regular reviews and updates help ensure that the software development efforts are consistently directed toward the desired outcomes.

While the format and the details of this document may vary based on the process model being used and the specific needs of the project, this document usually includes some key components:

- **Revision History**: A table showing the version, date, the description of the change, and the author of the change.
- **Introduction:** An overview of the software project including the context, background, and the purpose of the document.
- **Business Opportunity:** An analysis of the market and the need for the product.
- **Problem Statement**: The justification of the need for the product.
- **Vision/Product Position Statement**: A concise and inspiring statement which is a summary of who the system is for, its outstanding features, what differentiates it from the competition. This statement should convey the long-term vision and expected impact of the software.
- **Scope:** A definition of the boundaries of the software project; what is included and excluded from the project scope.
- **Stakeholder and User Descriptions:** Identification of the primary users and non-user stakeholders of the software, their high-level goals (needs/expectations), their problems, and possibly the current solutions to their problems.
- **Key Features and Functionalities:** An at most two-level bulleted summary of thr high-level descriptions of the main features and functionalities of the software. Features are behavioural functions that a system do; features are usually written in the format of: The system does <feature X>. Anything that does not pass this

format, is not a feature (i.e. non-functional requirements). Vision document should not be too detailed. As a rule of thumb, less than 10 features is desirable in a vision document; if you have more, you may want to group and abstract them.

- **Constraints:** Any limitations or constraints that may impact the development of the software, including budget and time constraints, design constraints, usability, performance, documentation, packaging, etc.
- **Assumptions:** Any dependencies or conditions that are critical for the success of the project.
- **Risks:** Potential risks that may affect the project's success.

Glossary (Data Dictionary)

Glossary is a document that lists noteworthy terms and their definitions. This document is very useful in terms of creating a shared understanding of the concepts, that facilitates communications and the onboarding of new team members. This document is usually created **collaboratively**, and includes domain-specific, process-specific, team-specific, and any term that is useful to be clarified. Glossary should be started early and should be kept in a shared repository to be developed throughout the development.

There is no specific format for this document, however like any other document, teams should consider using a format that is easily maintainable. This document usually includes a sorted list of terms and their attributes including:

- **Aliases**
- **Description**
- **Format (Type, Length, Unit, etc.)**
- **Relationships to Other Terms**
- **Range of Values**
- **Validation Rules**

Business (Domain) Rules

This document captures all the rules that are mandated by law and policies (e.g. tax laws). It is very useful to record domain rules in a separate repository that is shared among all the company's analysts so it can be reused in similar projects. This document usually includes a **revision history** and a **list of rules**. Each rule is usually assigned a **unique id** that is used in references, a **description** of the rule, its **changeability** information, and its **source**.

Reference

Craig Larman, "*Applying UML and Patterns*", 3rd edition, Prentice Hall PTR.