

# Analysis Report

Author: Stephen Aranda

**Describe the process of requirements analysis for a software project in general. Address why requirements need to be analyzed after being elicited. Include information about requirements classification and conceptual modeling.**

The process of requirements analysis focuses on the tasks that are conducted to determine the needs or conditions of different stakeholders. Requirement Analysis is the process of defining the expectations of the users for a system that is to be developed or modified. So, requirements analysis means to analyze, document, validate and manage software or system requirements. There are many steps in this process and this report will go over all of them for general understanding. One of the steps in the process of requirement analysis is **eliciting requirements**, which is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders. Sometimes this practice is referred to as “Requirements gathering”, however, there is a difference. That difference is how a software product manager interacts with the client; When gathering requirements, the product manager typically asks the client for a list of things they want, compared to elicitation which is a much more involved, interactive, and investigative process that results in exponentially better requirements. There are several techniques in requirements elicitation which include interviews, scenarios, prototypes, facilitated meetings, observations, and user stories along with many more. Another step in the requirement analysis process is **recording requirement** which consists of the product manager documenting in various forms that usually includes a summary list and may include natural language documents like use cases, user stories, process specifications, and a variety of models including data models. The last step in this process is called **analyzing requirements** which consist of determining whether the requirements are clear, complete, unduplicated, concise, valid, consistent, unambiguous, and resolve any apparent conflicts. Analyzing requirements must be carried out after elicitation, this after elicitation, there is a huge amount of information in those requirements that must be inspected to ensure that they are done correctly. If analyzing was done before elicitation, then there would be nothing to analyze which is unproductive. Requirements can be classified in a multitude of different ways, one of which ways is as **functional and non-functional requirements**. A functional requirement is a behavior

that the product should do or support, while a non-functional requirement serves as a description of how well a product must perform. It is complementary to functional requirements. Requirements can also be classified by whether it was derived from one or more high-level requirements, an emergent property or is being imposed directly on the software by a stakeholder or another source. Another way to classify a requirement is by determining if the requirement is a **product or process requirement**. A product requirement is a need or constraint on the software to be developed, like “The system shall verify that the student has completed all prerequisites before registering for a class” for example. A process requirement is essentially a constraint on the development of software like “The system shall be developed using the waterfall model process” for example. Setting a priority level of how important that requirement is to the system is another way to classify requirements. The higher the priority level, the more essential the requirement is. Requirements can be classified by its scope and by its volatility/stability (How much the requirement is susceptible to change) as well. In software requirement analysis, real-world models are extremely important. The purpose of these models is to aid in understanding the problem at hand and depicting the solution to it. There are many kinds of models that can be developed, such as use case diagrams, data flow models, state models, goal-based models, user interactions, object models, data models, and many others. These modeling notations are part of UML (Unified Modeling Language).

**Describe the process you took in analyzing requirements for your project including classification and modeling.**

My process for analyzing requirements for “SwiftClock” was also one of many steps. After eliciting all of my requirements using methods such as interviews, prototypes, user stories, observations, and even analyzing competitor products, I looked at how I could classify each individual requirement. To classify the requirements, I set a priority level on each requirement and I gave each requirement a type that includes functional and non-functional requirements. In order to better understand the issue and the solution that “SwiftClock” deals with, a couple of models were developed such as a use-case diagram and a data flow model which is also known as a Software Context Model or a “Level 0”. Throughout the process of analyzing requirements, it was very important to be able to think critically about each individual to ensure that the correct product will be developed to satisfy the client. Each element of the “Elder Paul Critical Thinking

Model”, purpose, question at issue, information, interpretation and inference, concepts, assumptions, implications and consequences, and point of view were all used to when creating, finalizing, and analyzing each requirement. The goal of each requirement along with the problem or the issue both needed to be addressed first in order to form the requirement. Then data and observations were gathered together to come up with the solution. Once the solution was discovered, models were made to ensure that the requirement is understood throughout and assumptions were made to have something to base the reasoning of each requirement on. Next, the implications of each requirement were looked at to make sure that no surprises arise as a consequence of that requirement. Finally, the point of view from which the requirements were looked at differed from myself and each stakeholder that these requirements were shown to, in order to identify its strengths along with its weakness as a complete requirement.

**Describe how you recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.**

To recognize ethical responsibilities in engineering situations the Software Engineering Code of Ethics and Professional Practice is a key resource that ensures the products that is built by the software engineer adheres to the eight ethical principles. Professional responsibilities are recognized by inspecting and analyzing IEEE 29148’s international standard to ensure that the systems and software engineering life cycle processes are done accordingly, following the standard that is already set.