# Requirements Prioritization Techniques



**Nadina Martinez Carod**
*Universidad Nacional del Comahue, Argentina*

**Alejandra Cechich**
*Universidad Nacional del Comahue, Argentina*

## INTRODUCTION

As part of Requirements Engineering, "Elicitation" is the phase where an analyst collects information from the stakeholders, clarifies the problems and the needs of the customers and users, tries to find the best solutions, and makes its planning on what software system will be developed. During elicitation, to get well-defined requirements, a consensus among the different stakeholders is needed. There are several elicitation techniques in the literature; however every technique faces the same problem: each stakeholder has different requirements and priorities, which potentially produces conflicting situations. Therefore, this situation points out Requirements Prioritization as a relevant research area to define the requirements' level of importance.

Nevertheless, often the strategies implemented to solve conflicts among stakeholders are inadequate; for example, weighting requirements can be problematic because sometimes weights are inconsistent and lead to confusion about which are the most essential customer requirements. The prioritizing process must hold stakeholder satisfaction considering high-priority requirements first. However, practical experience shows that prioritizing requirements is not as straightforward task as the Literature suggests. In any case, clearly defining a way of balancing preferences on requirements is essential to the elicitation process.

The remainder of this chapter is structured as follows. Section 2 describes a conceptual framework to describe several prioritization proposals, which are characterized in Section 3. Future trends are presented afterwards.

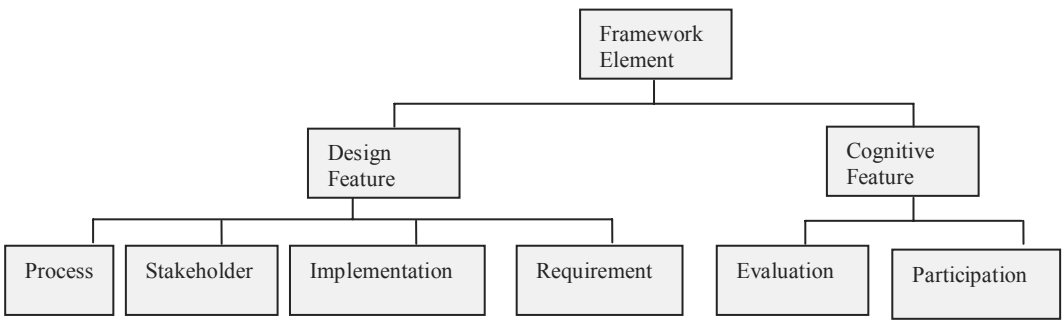
## BACKGROUND

Some comparisons of elicitation methods have clarified common features. Firstly, the comparative study by Thomas and Oliveros (2003) is centralized in properties and limitations of five of the most significant methods for eliciting requirements in goal-oriented requirements engineering. This comparison is organized from the viewpoint of goal acquisition with especial emphasis in goal elicitation. Secondly, based on an evaluation framework and influenced by an industrial application (Karlsson & Ryan, 1997), characterizes six different methods for prioritizing software requirements. The objective of Karlsson's evaluation is outlining the methods' behavior for a particular experience, thus the results obtained are not supposed to be generalized by any environment for any application. This evaluation framework is based on inherent characteristics, objective measures and subjective measures.

Our classification framework (Figure 1) is structured into two building blocks – *design features* and *cognitive features*.

The *Design category* is composed of four elements that consider different aspects: *Process, Stakeholders, Implementation* and *Requirements*. The specific features of each

*Figure 1. A conceptual framework for comparison*



prioritization method are categorized by the *Process* element. It considers answering some questions, such as: Does the process detect inconsistency? Is the process referred to as a systematic or a rigorous process? How we address the problem of dealing with different priorities? Conceptually, is it based on goal decomposition? Does it use a priority or an importance order? The framework also characterizes how prioritizing methods consider *stakeholders*. There are two parameters to be analyzed here: the former refers to the kind of information the method provides with respect to stakeholders. Does the method analyze which stakeholder prioritized a goal, and which priority degree was assigned? The second parameter considers stakeholders geographically distributed.

The *implementation category* depends on the method's scalability and dynamism, that is, usability. It is influenced by how many and which calculus the method uses, and by the performance of the method with a huge number of requirements. It is considerably important whether tools, as well as a reference to spread projects, were applied to support the method. The framework considers information that can demonstrate the method's success in pilot studies. *Requirements* analyze functional (FR) and nonfunctional requirements (NFR) as well as interactions among requirements – *interdependency* represents requirements interaction. Some methods calculate cost and benefit figures for individual requirements, but if there are significant interactions among requirements, the situation becomes more complex. As an example, if two requirements in a method can be achieved by sharing the same solutions to subproblems, then the cost of attaining both of them may be significantly less than the sum of their individual costs. Therefore, the main key is whether the method can handle requirements' interdependencies. The *requirement category* also analyses if the methods deal with functional—FR—and nonfunctional requirements—NFR.

*Cognitive aspects* cover the evaluation of cognitive features as participation and negotiation among stakeholders during the whole process (Chiew & Wang, 2003). *Evaluation* studies what personal characteristics serve to establish priorities. *Participation* includes defining how priorities were assigned (subjective or objective) from personal experiences and interviews to ensure the success of the developed method. The cognitive aspects no cover the cognitive techniques for knowledge acquisition of knowledge based system.

## FEATURES FOR COMPARISON

We can identify two kinds of features: those present in any strategy, and those that may be present or not. Although the first group of features is present in any method, the way these characteristics are maintained is specific of each method. Our appreciation focuses on the second group, which makes the comparison more interesting. By taking into account these concerns, we established three levels of increasing importance to analyze the features: desirable "D", highly desirable "HD" and mandatory "M".

### Simple Features

The simple features we considered to analyze processes are:

*Consistency*: Many times two stakeholders agree on requirements with opposite meanings, which turns impossible the implementation of those requirements. These requirements inconsistencies arise as a result of conflicts between requirements. We consider the action of *detect inconsistencies mandatory* because it is the key of a successful project.

R*igorous:* If a method is *rigorous* and *systematic* it provides robust and comprehensive steps and handling requirements consistently and effectively, which became this feature *highly desirable*. It aids in the validity and verification and it is related intimately to the consistency of requirements.

*Goal decomposition*: The process based on *goal decomposition* is *desirable* in a prioritization process. The reason is that goals help do not assure successfulness. Clarifying conflicting terms can reduce conflicts, even if the technique does not support goal decomposition.

*Priority:* Discussion of requirements priorities improves communication between the customer and the developer and helps resolve conflicts. Therefore, we consider *mandatory* the process of deriving an order relation on a given set of requirements, in order to assign a *priority order*, with the ultimate goal of obtaining a shared rationale for partitioning them into subsequent product releases.

*Requirements Interdependence:* The different occurrences of requirements changes throughout the life cycle points out some dependencies among functional requirements. Understanding these dependencies may improve the requirements process. An approach assumption implies that if two functions are modified due to the same fault report, then there are some *requirements interdependencies* between them. Thus an analysis of such identified fault reports is *desirable* as it may give additional information about requirements.

*Objective*: One disadvantage detected is that in many methods only one stakeholder has the responsibility of estimating the relative requirements value, which becomes the process subjective. We suppose as *desirable* that the process considers the search of solutions to be as *objective* as possible because the quality requirements always are influenced by analysts' opinions.