

## **Successful Software Projects Start with Requirements Management**

By Serge Charbonneau, Principal Consultant, Xelation Software Corporation ([www.xelation.com](http://www.xelation.com)).

Originally published in French in Direction Informatique on September 18, 2003

### **Introduction**

I have a question for you: Could you imagine asking a contractor to build you a house without specifying anything? Or buying a car without knowing its characteristics, or even its price? The answer to these questions is simple: No.

I have another question for you: Could you image asking a software development contractor to build you a pay management application without specifying its characteristics? Or buying a customer relationship management application without knowing its main features, or even the cost for deploying it? The answers to these questions should be as simple as the answers to the previous two questions: No.

However, how many software projects are never completed because requirements are not managed properly or are never captured explicitly? And how many applications are purchased but never deployed, or are deployed but never used because they don't meet the needs of the end users? The answer to these questions is the following: Way to many.

The best way to avoid these types of problems in the development and the acquisition of software is to implement a sound requirements management process. But what are the characteristics of a sound requirements management process? This article attempts to provide an answer to this question.

### **Characteristics of a Sound Requirements Management Process**

A sound requirements management process needs to have the following characteristics:

- Requirements expressed in a clear and concise manner;
- Attributes associated with each requirement;
- Explicit traceability between requirements that are related; and
- A sound change management process.

Let's look at each of these characteristics in details.

#### **Requirements Expressed in a Clear and Concise Manner**

A requirement is defined as a condition or capability a system must conform to.

The requirements for the development and the acquisition of a software application need to define the 'What' of the application as opposed to the 'How'. In the case of application development, it is necessary to clearly express what the application needs to do before expressing how it is going to do it. The 'What' is captured during the requirements definition while the 'How' is derived from the 'What' during the design. The design is then implemented using any programming language (C, C++, C# or Java). In the case of application acquisition, it is necessary to clearly express what the application needs to do before going forward with the selection process.

Each requirement for the development or the acquisition of an application must have the following qualities:

- Verifiable → Is testable.
- Modifiable → Is modifiable.
- Traceable → Is derived from one or many requirements, and/or one or many requirements are derived from it.
- Understandable → Is understandable by everyone technical and non-technical people alike.
- Correct → Is without errors.
- Complete → Describe by itself, or with other requirements, a complete set without any missing pieces.
- Consistent → Does not contradict other requirements.
- Unambiguous → Means the same thing for everybody.

## **Attributes Associated with Each Requirement**

The requirement attributes allow more efficient software development project management. Here is a sample of common requirement attributes:

- State: Identifies where the requirement is in its lifecycle. A requirement can be proposed, approved, incorporated or validated.
- Priority: Identifies the relative priority of a requirement. This attribute can be quantitative (with a value from 1 to 10, 1 being the highest priority and 10 being the lowest priority) or qualitative (high, medium, low).
- Cost or Effort: Identifies the cost (in dollars) or the effort (in person days) required to realize the requirement.
- Resource: Identifies the resource responsible for realizing the requirement.
- Target Iteration: Identifies when the requirement will be realized.

Other attributes can be associated with requirements.

The attributes allow the project managers to plan the effort required for implementing a software application in terms of time, cost and scope. If an application is developed using an iterative approach, the high priority requirements are usually implemented during the early iterations while the low

priority requirements are usually implemented during the late iterations. Furthermore, the cost or effort required to realize a set of requirements can be calculated from the cost or effort associated with each individual requirement.

## **Explicit Traceability Between Requirements**

Traceability allows the identification of the source the requirement is derived from and the destination the requirement is derived into. A software requirement can be derived from a higher-level requirement and/or can be derived into one or many lower-level requirements, one or many design elements (classes, attributes, operations, subsystems, etc.), or one or many test cases.

Traceability presents two main advantages:

1. Insures that all requirements of a certain type or level are being realized by requirements of another type or of a lower level; and
2. Allows more efficient and objective impact analysis from a project management perspective.

When a traceability strategy exists between software requirements and test cases, it becomes easier to insure that test cases exist to validate all requirements and that no requirement is left untested. Furthermore, if a traceability strategy exists between different levels of requirements and design elements, it becomes also easier to evaluate the impact of changing a requirement on all elements derived from it.

## **A Sound Change Management Process**

Why is it necessary to have a sound change management process in a software development project? The answer is simple: Because there is going to be change. Developing software is a very complex and risky endeavor in a very abstract field of activities. This is why it is almost impossible to perfectly implement an application in a single iteration. This implies that there are change requests in all software development projects whether they are explicitly tracked or not.

A sound change management process implies that each change request is formally and explicitly captured, analyzed in terms of its impact on the existing application and in terms of the effort required to implement it, and presented to an individual or group of individuals who decides what to do with it (approve it, reject it or postponed it). When it is approved, the change request is directly derived into changes to the baselined requirements, and indirectly derived into design and implementation elements.

## **Conclusion**

# ***Xelation***

■■■ Software Corporation ■■■

Software engineering is a discipline that is heavily based on 'common sense'. However, more often than not 'common sense' is very uncommon. Requirements management is one of the less technical aspects of software engineering that is almost entirely based on 'common sense'. Indeed, requirements management states that 'it is necessary to specify what the application needs to do before specifying how it is going to do it.' It is simple, no? So, next time you change your car, think about it.

## **About Xelation Software Corporation**

Xelation Software Corporation is a company founded in 2003 that provides its customers with solutions in the following areas of software engineering:

- Requirements Management;
- Architecture;
- Object-Oriented Analysis and Design;
- Development;
- Quality Control;
- Project Management;
- Configuration and Change Management; and
- Process Engineering.

Xelation Software Corporation's mission is to accelerate the success of its customers in improving their software engineering capabilities. Xelation Software Corporation's web site is found at [www.xelation.com](http://www.xelation.com).

## **About Serge Charbonneau**

Serge Charbonneau is a principal consultant for Xelation Software Corporation. Mr. Charbonneau has over 12 years of experience in the field of software engineering. He has played different roles in software organizations including analyst, designer, developer, team lead, technical representative and consultant. He has a solid expertise in software project management, requirements management, object-oriented analysis and design, and process engineering.